

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ

(назва теми)

ПІДТРИМКИ ПЕРСОНАЛІЗОВАНОГО НАВЧАННЯ ДИЗАЙНЕРІВ-
МОДЕЛЬЄРІВ. RESEARCH AND SOFTWARE IMPLEMENTATION
OF METHODS FOR PERSONALIZED LEARNING SUPPORT FOR
FASHION DESIGNERS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-114м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЛАВРЕНЧУК С.О.

(ПРІЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРІЗВИЩЕ та ініціали)

2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет _____ КНТ _____
Кафедра _____ програмних засобів _____
Ступінь вищої освіти _____ магістр _____
Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код і найменування)
Освітня програма (спеціалізація) _____ Інженерія програмного забезпечення _____
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф. _____
Сергій СУББОТІН
« _____ » _____ 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЛАВРЕНЧУКА Степана Олеговича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів підтримки персоналізованого навчання дизайнерів-модельєрів. Research and Software Implementation of Methods for Personalized Learning Support for Fashion Designers

керівник проєкту (роботи) к.т.н., доцент ЛЬОВКІН Валерій Миколайович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 30 » вересня 2025 року № 447

2. Строк подання студентом проєкту (роботи) 08 грудня 2025 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Проєктування програмного забезпечення. 4. Основні рішення щодо реалізації програмного забезпечення. 5. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 30 » вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Дослідження проблеми	3-5 тижні	Розділ 2
4	Проектування та розробка архітектури програми	6 тиждень	Розділ 3
5	Розробка програми	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження	9-10 тижні	Розділ 5
7	Оформлення пояснювальної записки та документів до неї	11 тиждень	Додатки
8	Нормоконтроль та рецензування	12 тиждень	
9	Захист роботи	12 тиждень	

Студент(ка)

_____ Степан ЛАВРЕНЧУК
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Валерій ЛЬОВКІН
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
145 с., 38 рис., 3 табл., 3 дод., 30 джерел.

ОНЛАЙН НАВЧАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
ДИЗАЙНЕР-МОДЕЛЬЄР, ТЕХНІЧНИЙ ПАКЕТ, ПРОГНОЗУВАННЯ,
RANDOM FOREST.

Об'єкт роботи – процес розроблення програмного забезпечення для персоналізованого навчання дизайнерів-модельєрів. Предмет роботи – алгоритми і програмні засоби підтримки персоналізованого навчання дизайнерів-модельєрів. Мета роботи – зменшення часових витрат на організацію персоналізованого навчання дизайнерів-модельєрів на основі програмного засобу за рахунок відбору кандидатів з більшим потенційним підвищенням вмінь у результаті.

Матеріали, методи та технічні засоби: мова Python, фреймворк Django, засоби машинного навчання, зокрема метод Random Forest, система MySQL.

Результати. Створена програма призначена для використання дизайнерами-модельєрами для вивчення курсів, які надаються майстрами, представлення колекцій одягу за технічними пакетами, ознайомлення з ідеями інших дизайнерів-модельєрів, майстрами з дизайну одягу для навчання дизайнерів-модельєрів за своїми курсами, пошуку талантів серед них.

Висновки. Наукова новизна роботи полягає в методі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання, який характеризується використанням методу Random Forest, що дозволяє виконувати прогнозування зміни рівня знань і вмінь дизайнера-модельєра відносно його минулого середнього рівня.

Галузь використання. Надання послуг навчання дизайнерів-модельєрів, створення платформи презентації ідей дизайнерів-модельєрів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 145 pages, 38 figures, 3 tables, 3 appendixes, 30 sources.

ONLINE LEARNING, SOFTWARE, FASHION DESIGNER, TECHNICAL PACKAGE, FORECASTING, RANDOM FOREST.

The object of the work is the process of developing software for personalized training of fashion designers. The subject of the work is algorithms and software tools to support personalized training of fashion designers. The purpose of the work is to reduce the time spent on organizing personalized training of fashion designers based on a software tool by selecting candidates with a greater potential for improving skills as a result.

Materials, methods and technical means: Python language, Django framework, machine learning tools, in particular the Random Forest method, MySQL system.

Results. The created program is intended for usage by fashion designers to study courses provided by masters, present clothing collections according to technical packages, get acquainted with the ideas of other fashion designers, by fashion design masters to train fashion designers based on their courses, search for talents among them.

Conclusions. The scientific novelty of the work is in the method of forecasting the results of increasing the level of skills of a fashion designer based on the results of personalized training, which is characterized by the usage of the Random Forest method, which allows predicting changes in the level of knowledge and skills of a fashion designer relative to past average level.

Field of use. Provision of training services for fashion designers, creation of a platform for presenting the ideas of fashion designers.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Огляд вебзастосунків підтримки навчання дизайнерів-модельєрів	10
1.2 Порівняння аналогів розробки вебзастосунку персоналізованого навчання дизайнерів-модельєрів.....	12
1.3 Висновки за розділом 1	13
2 Матеріали і методи.....	15
2.1 Задача прогнозування результатів підвищення рівня вмінь дизайнера- модельєра за результатами персоналізованого навчання.....	15
2.2 Метод прогнозування результатів підвищення рівня вмінь дизайнера- модельєра за результатами персоналізованого навчання.....	16
2.3 Висновки за розділом 2	19
3 Проектування програмного забезпечення	21
3.1 Вибір мови та засобів розробки.....	21
3.2 Моделювання роботи з програмою дизайнера-модельєра, майстра дизайну одягу, адміністратора, неавторизованого користувача	22
3.3 Проектування структури бази даних.....	39
3.4 Висновки за розділом 3	56
4 Основні рішення щодо реалізації програмного забезпечення	57
4.1 Послідовність взаємодії користувача з програмою персоналізованого навчання дизайнерів-модельєрів.....	57
4.2 Рішення з реалізації вебзастосунку	58
4.3 Висновки за розділом 4	84
5 Експлуатація, тестування та експериментальне дослідження програми	85
5.1 Опис програми.....	85
5.2 Експлуатація програми.....	85
5.3 Експериментальне дослідження	97

5.4 Висновки за розділом 5	98
Висновки	99
Перелік джерел посилання	100
Додаток А Технічне завдання	103
Додаток Б Текст програми	110
Додаток В Слайди презентації.....	137

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MAE – Mean Absolute Error;

MSE – Mean Squared Error;

RMSE – Root Mean Squared Error;

БД – база даних;

ПЗ – програмне забезпечення;

ВСТУП

Навчання дизайнерів-модельєрів на цифровій платформі потребує наявності спеціальних інструментів, які повинні гарантувати зв'язок між майстрами у цій сфері та користувачами, які тільки навчаються.

Персоналізоване навчання передбачає з одного боку використання персоналізованих матеріалів для користувачів, тобто коли для окремих користувачів можуть надаватися майстром спеціальні матеріали. З іншого боку це потребує розуміння того, кого таким чином будуть навчати. Тобто персоналізовані матеріали можуть потребувати кропіткої роботи, а тому необхідно розуміти, на кого ця робота направлена. Тому дизайнер-модельєр повинен мати можливість продемонструвати свої вміння в програмі, а з іншого боку необхідно застосовувати інструменти прогнозування, які б дозволили передбачити те, на скільки такий курс буде персонально цьому дизайнеру-модельєру корисним.

Необхідно розуміти, що без наявності таких інструментів майстер з дизайну одягу буде витрачати власний час на велику кількість дизайнерів-модельєрів, які фактично не зацікавлені в навчанні, не здатні прогресувати під керівництвом майстра. Такий варіант взаємодії потребує від майстра додаткової роботи, при цьому вкладення ресурсів будуть неефективними. Відповідно майстру необхідно знайти такого дизайнера-модельєра, який буде зацікавлений у навчанні, здатен розвиватись, і на нього витрачати додаткові ресурси, яких потребує персоналізоване навчання.

Саме на це направлена розробка програмного забезпечення (ПЗ) персоналізованого навчання дизайнерів-модельєрів і саме такі інструменти мають використовуватися в ньому в основі.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд вебзастосунків підтримки навчання дизайнерів-модельєрів

За результатами огляду існуючих вебзастосунків підтримки навчання дизайнерів-модельєрів було з'ясовано, що цілісного програмного засобу, який би використовувався для навчання саме спеціалізовано дизайнерів-модельєрів, пошуку дизайнерів-модельєрів, не існує на даний момент.

Існуючі програмні засоби для підтримки дизайнерів-модельєрів можна розділити на дві групи.

Прикладом першої групи є LinkedIn (рис. 1.1) [1].

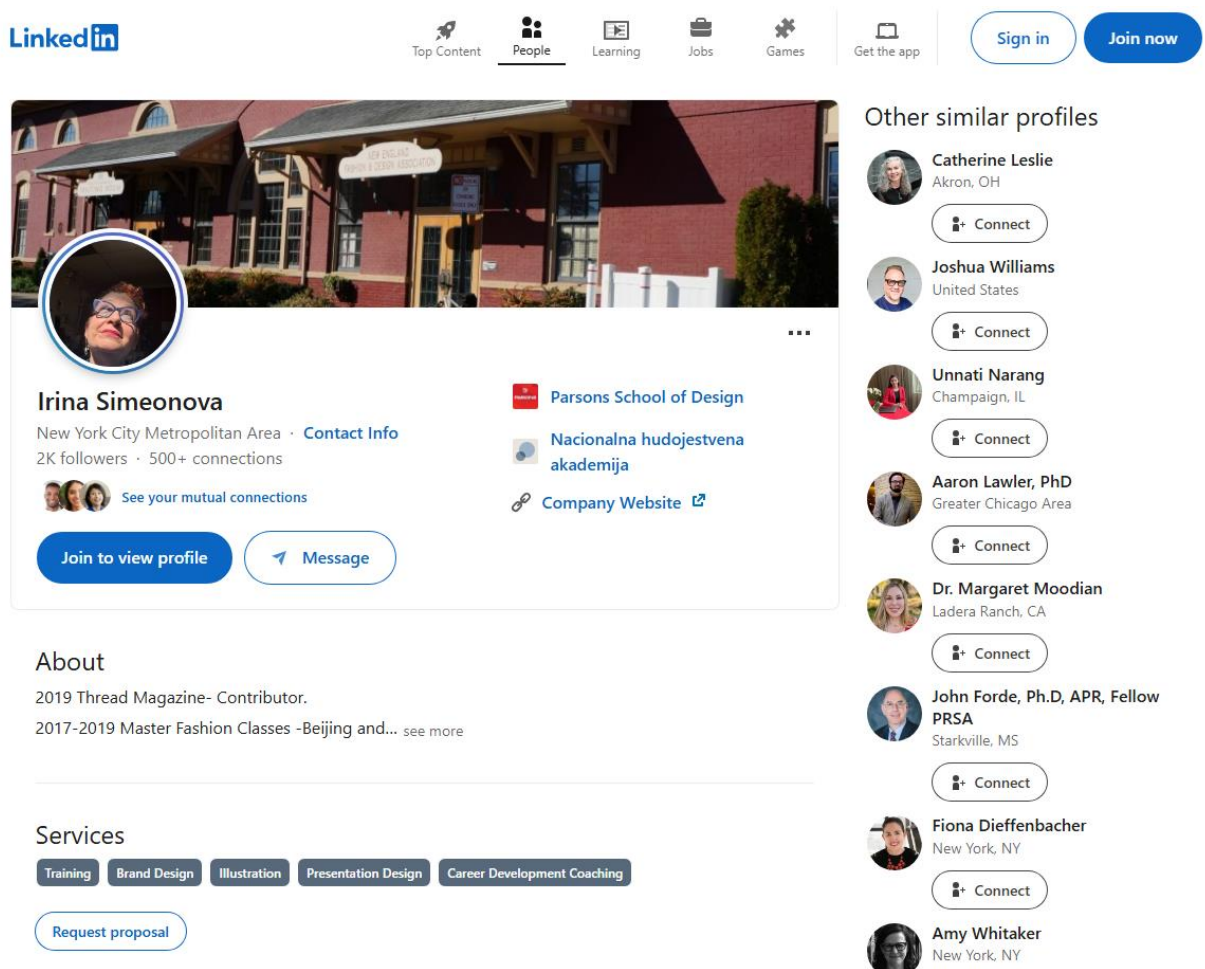


Рисунок 1.1 – Сторінка користувача в LinkedIn

LinkedIn є соціальною мережею, де представлені користувачі з різними професіями, але серед них є і дизайнери-модельєри (рис. 1.2). За такими користувачами може бути приведена інформація про курси, які ці користувачі вивчали, досвід роботи таких користувачів, опубліковані ними статті. Такі статті можуть містити різну інформацію, проте вони не призначені для виділення структури технічних пакетів з виготовлення одягу.

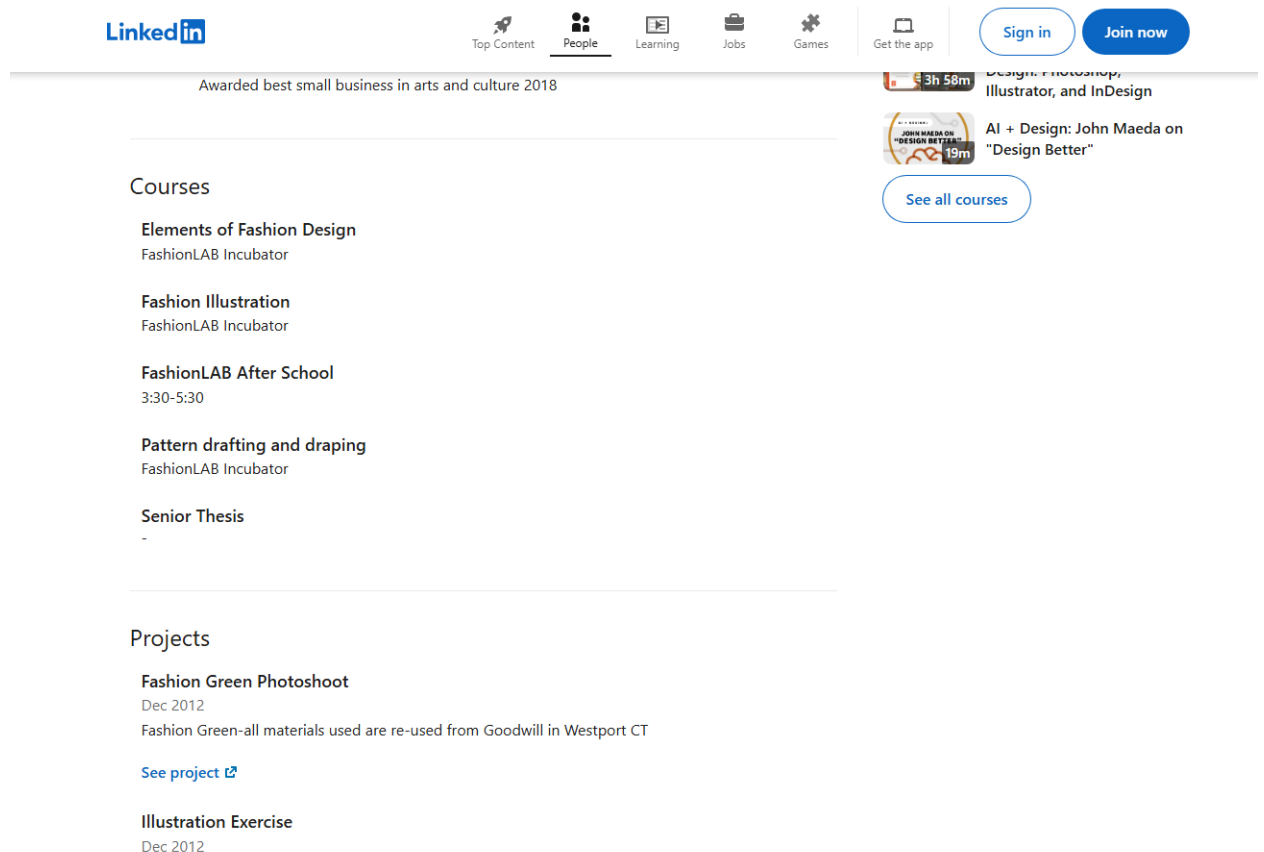


Рисунок 1.2 – Курси користувача в LinkedIn

Techracker (рис. 1.3) представляє другу групу програмних засобів, які дозволяють створювати технічні пакети для дизайнерів-модельєрів [2]. У результаті формується представлення технічного пакету в програмі та формується pdf-файл (рис. 1.4) [3]-[4].

Технічні пакети є специфікаціями одягу, що створює дизайн-модельєр, вираженими кресленнями одягу в різних ракурсах [5].

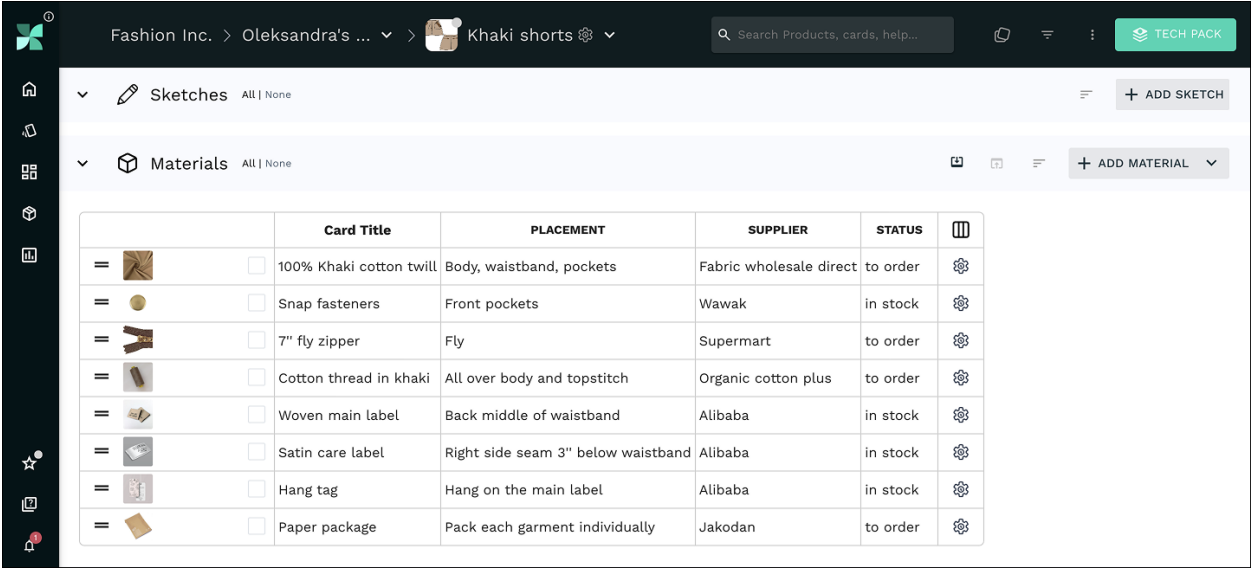


Рисунок 1.3 – Інтерфейс програми Techpacker [3]

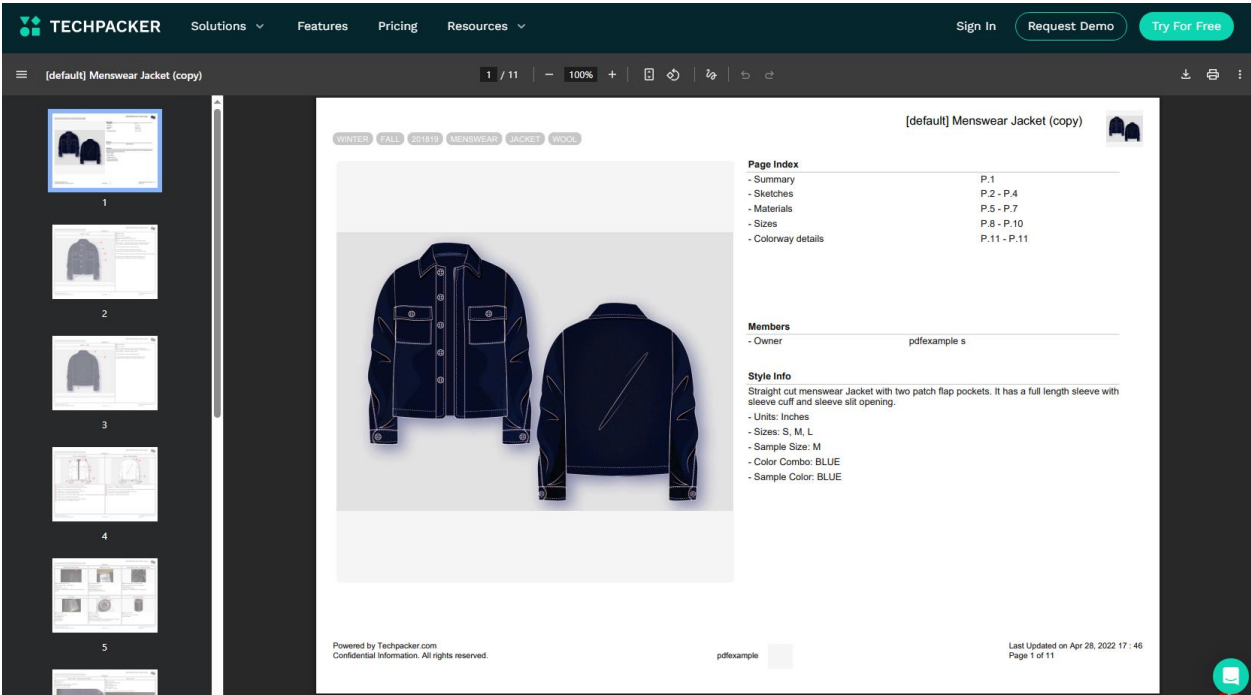


Рисунок 1.4 – Технічний пакет, створений у Techpacker [4]

1.2 Порівняння аналогів розробки вебзастосунку персоналізованого навчання дизайнерів-модельєрів

До порівняння за результатами огляду вебзастосунків підтримки навчання дизайнерів-модельєрів внесено програми LinkedIn і Techpacker та FasDeApp, що є розробкою цієї роботи (табл. 1.1).

Таблиця 1.1 – Порівняння аналогів вебзастосунку персоналізованого навчання дизайнерів-модельєрів

Показник	LinkedIn	Techpacker	FasDeApp
Створення технічних пакетів	—	+	+
Перегляд технічних пакетів	Як загальні публікації	+	+
Пошук дизайнерів-модельєрів, майстрів	+	—	+
Матеріали для навчання дизайнерів-модельєрів	Загальні, неперсоналізовані	—	Персоналізовані
Прогнозування результатів підвищення рівня вмінь дизайнера-модельєра	—	—	+

Програма FasDeApp, яка створюється в роботі, надає можливості створення технічних пакетів, які доступні тільки в Techpacker, пошуку дизайнерів-модельєрів і майстрів як у LinkedIn, але зі спеціалізованими інструментами для дизайнерів-модельєрів, надає можливість створювати і переглядати матеріали для навчання, які є персоналізованими на відміну від LinkedIn. Окрім того розробка містить можливості прогнозування результатів підвищення рівня вмінь дизайнера-модельєра, які відсутні у всіх аналогів.

1.3 Висновки за розділом 1

Метою роботи є зменшення часових витрат на організацію персоналізованого навчання дизайнерів-модельєрів на основі програмного

засобу за рахунок відбору кандидатів з більшим потенційним підвищенням вмінь у результаті. Для виконання мети завданнями є:

- аналіз програм персоналізованого навчання дизайнерів-модельєрів;
- розробити метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра;
- проєктування програми персоналізованого навчання дизайнерів-модельєрів;
- реалізація програми персоналізованого навчання дизайнерів-модельєрів;
- експериментальне дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра.

У розділі виконано аналіз програм персоналізованого навчання дизайнерів-модельєрів на основі порівняння розробки з LinkedIn і Techpacker. Розробка об'єднує можливості аналогів, містить можливості прогнозування результатів підвищення рівня вмінь дизайнера-модельєра, які відсутні у всіх аналогів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Задача прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Персоналізоване навчання одним з важливих вимірів результатів має підвищення рівня вмінь. Рівень вмінь дизайнера-модельєра в подібних програмних системах визначається балом, який отримує цей дизайнер-модельєр загалом або за окремими критеріями. Це дозволяє характеризувати вміння дизайнера-модельєра.

У даному випадку результатом підвищення рівня вмінь дизайнера-модельєра розглядається показник, який визначає зміну балів, які має дизайнер-модельєр за результатами навчання. Ця зміна балів розраховується як різниця між тим балом, який дизайнер-модельєр отримує за вивчення курсу, та тим балом, який обчислюється як усереднений бал за вивчення курсів, які цей дизайнер-модельєр опанував раніше. Таким чином вихідним показником є те, на скільки знання та вміння дизайнера-модельєра за результатами вивчення персоналізованого курсу відрізняються від його середнього рівня знань та вмінь за результатами вивчення попередніх курсів.

Описаний показник є вихідною ознакою для задачі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання.

Вхідними ознаками задачі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання, які характеризують дизайнера-студента за його попереднім досвідом, є наступні.

Перша ознака – середня оцінка останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр, виражена значенням в діапазоні від 0 до 100.

Друга ознака – середня оцінка цифрових компетентностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

Третя ознака – середня оцінка рівня підприємницьких вмінь дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

Четверта ознака – середня оцінка здібностей вирішувати комплексні проблеми дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

П'ята ознака – середня оцінка здібностей до креативності та іноваційності дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

Шоста ознака – середня оцінка комунікаційних здібностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

Сьома ознака – середня оцінка залученості дизайнера-модельєра до виконання завдань курсів навчання, яку було виставлено за останні рази за вивченими курсами, виражена значенням в діапазоні від 0 до 100.

Увесь цей набір ознак сукупно характеризує дизайнера-модельєра і є вхідними даними для вирішення задачі, яка описується.

Таким чином, задача прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання полягає в тому, щоб для сукупності даних про минулий досвід дизайнерів-модельєрів, який характеризується набором значень за кожною вхідною ознакою кожного екземпляру, визначити спосіб перетворення цих значень на вихідну ознаку, якою є зміна рівня знань та вмінь дизайнера-модельєра за результатами вивчення персоналізованого курсу.

2.2 Метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Для розв'язання задачі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

запропоновано метод (рис. 2.1).

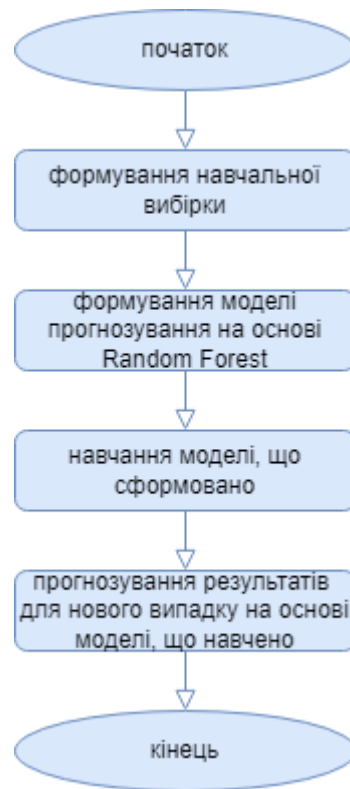


Рисунок 2.1 – Метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Перший етап методу прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання передбачає формування навчальної вибірки.

Для формування навчальної вибірки необхідно визначити сукупність екземплярів, за якими відомо значення вихідної ознаки та всі значення набору вхідних ознак, включаючи такі:

- середня оцінка останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр.
- середня оцінка цифрових компетентностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами;
- середня оцінка рівня підприємницьких вмінь дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами;

- середня оцінка здібностей вирішувати комплексні проблеми дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами;

- середня оцінка здібностей до креативності та інноваційності дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами;

- середня оцінка комунікаційних здібностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами;

- середня оцінка залученості дизайнера-модельєра до виконання завдань курсів навчання, яку було виставлено за останні рази за вивченими курсами;

- оцінка результату вивчення персоналізованого курсу за поточного стану знань та вмінь дизайнера-модельєра (вихідна ознака).

Виділяючи за всіма екземплярами значення всіх вхідних ознак, вони мають сформувати навчальну вибірку розміром кількість екземплярів на кількість вхідних ознак та окремо вектор з вихідною ознакою. Значення у векторі вихідної ознаки мають бути нормалізовані (приведені до інтервалу від -1 до 1).

Якщо у навчальній вибірці зустрічаються екземпляри, які характеризують результати навчання того самого дизайнера-модельєра за тим самим курсом, то такі записи мають бути вилучені.

На другому етапі має бути сформовано модель для прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання. Ця модель представляє собою ансамбль дерев рішень, які утворюються на основі методу Random Forest [6]-[8]. Метод Random Forest дозволяє створювати моделі для розв'язання задачі прогнозування у різних сферах [9]-[12].

Дерев у створеному ансамблі має бути виділено 100. Дерев використовуються для досягнення спеціалізації у вивченні результатів навчання дизайнерів-модельєрів.

На третьому етапі всі дані навчальної вибірки мають бути використані для навчання ансамблю дерев рішень.

На четвертому етапі має забезпечуватися прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання для тих випадків, які виникли в подальшому. Це ситуації, коли ті самі або інші дизайнери-модельєри проходять нові курси. За таким дизайнером-модельєром визначаються значення ознак, що охоплюють на момент початку його навчання середню оцінку останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр, середню оцінку цифрових компетентностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, середню оцінку рівня підприємницьких вмінь дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, середню оцінку здібностей вирішувати комплексні проблеми дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, середню оцінку здібностей до креативності та інноваційності дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, середню оцінку комунікаційних здібностей дизайнера-модельєра, яку було виставлено за останні рази за вивченими курсами, середню оцінку залученості дизайнера-модельєра до виконання завдань курсів навчання, яку було виставлено за останні рази за вивченими курсами.

Ансамбль дерев рішень, який було навчено, має виконати прогнозування результатів підвищення рівня вмінь цього дизайнера-модельєра за результатами персоналізованого навчання. Отримане значення і є результатом роботи методу.

2.3 Висновки за розділом 2

Другий розділ містить формулювання задачі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання та запропонованого методу для її розв'язання.

Наукова новизна роботи полягає в методі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання, який характеризується використанням методу Random Forest, що дозволяє виконувати прогнозування зміни рівня знань і вмінь дизайнера-модельєра відносно його минулого середнього рівня.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір мови та засобів розробки

Розробку ПЗ персоналізованого навчання дизайнерів-модельєрів вирішено вести на основі використання наступних засобів:

– мови програмування Python [13]-[17] та фреймворку Django [18]-[20], забезпечуючи для Python засоби створення вебзастосунків: вибір здійснений після порівняння мов Python і PHP [21] (табл. 3.1);

– шаблону створення макету вебзастосунку [22], що є вільнорозповсюджуваним;

– системи MySQL для управління базами даних.

Таблиця 3.1 – Вибір мови для програмування ПЗ персоналізованого навчання дизайнерів-модельєрів

	Python	PHP
Напрямок застосування для розробки ПЗ	Обробка даних, машинне навчання, розробка вебзастосунків	Розробка вебзастосунків
Підтримка ПЗ	Менше ресурсів	Більше ресурсів
Витрати ресурсів на роботу з даними, моделювання	Менше ресурсів	Більше ресурсів
Витрати ресурсів на розробку вебзастосунків	Співмірний обсяг ресурсів	Співмірний обсяг ресурсів

Вибір мови для програмування ПЗ персоналізованого навчання дизайнерів-модельєрів здійснений на користь Python, оскільки витрати ресурсів на роботу з даними, моделювання в таких проєктах є меншими, а на

веброзробку – співмірні, при тому подальша підтримка ПЗ зазвичай вимагає також менше ресурсів.

3.2 Моделювання роботи з програмою дизайнера-модельєра, майстра дизайну одягу, адміністратора, неавторизованого користувача

Користувачі ПЗ розподілені за ролями:

- неавторизований користувач;
- дизайнер-модельєр – користувач, який головним чином може виставляти свої технічні пакети з виготовлення одягу, навчатися за курсами, які надаються майстрами;
- майстер – користувач, який може надавати навчання дизайнерам-модельєрам за власними курсами, в тому числі персоналізовано, шукати талановитих дизайнерів-модельєрів;
- адміністратор – користувач, який може керувати процесами взаємодії між дизайнерами-модельєрами та майстрами, моделлю прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання.

Неавторизований користувач має забезпечуватися програмою сценаріями:

- відображення інформації стосовно дизайнера-модельєра;
- авторизація;
- відображення інформації стосовно майстра дизайну одягу;
- відображення інформації стосовно курсу для навчання дизайнерів-модельєрів;
- відображення курсів для навчання дизайнерів-модельєрів за майстром;
- пошук серед курсів для навчання дизайнерів-модельєрів;
- пошук серед майстрів дизайну одягу;
- пошук серед технічних пакетів зі створення одягу;

- відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом;
- відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- пошук серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів;
- відображення всіх оцінювань курсу дизайнерами-модельєрами.

Реєстрація є альтернативним для авторизації дизайнера-модельєра, коли дизайнер-модельєр не має акаунта для реєстрації (рис. 3.1).

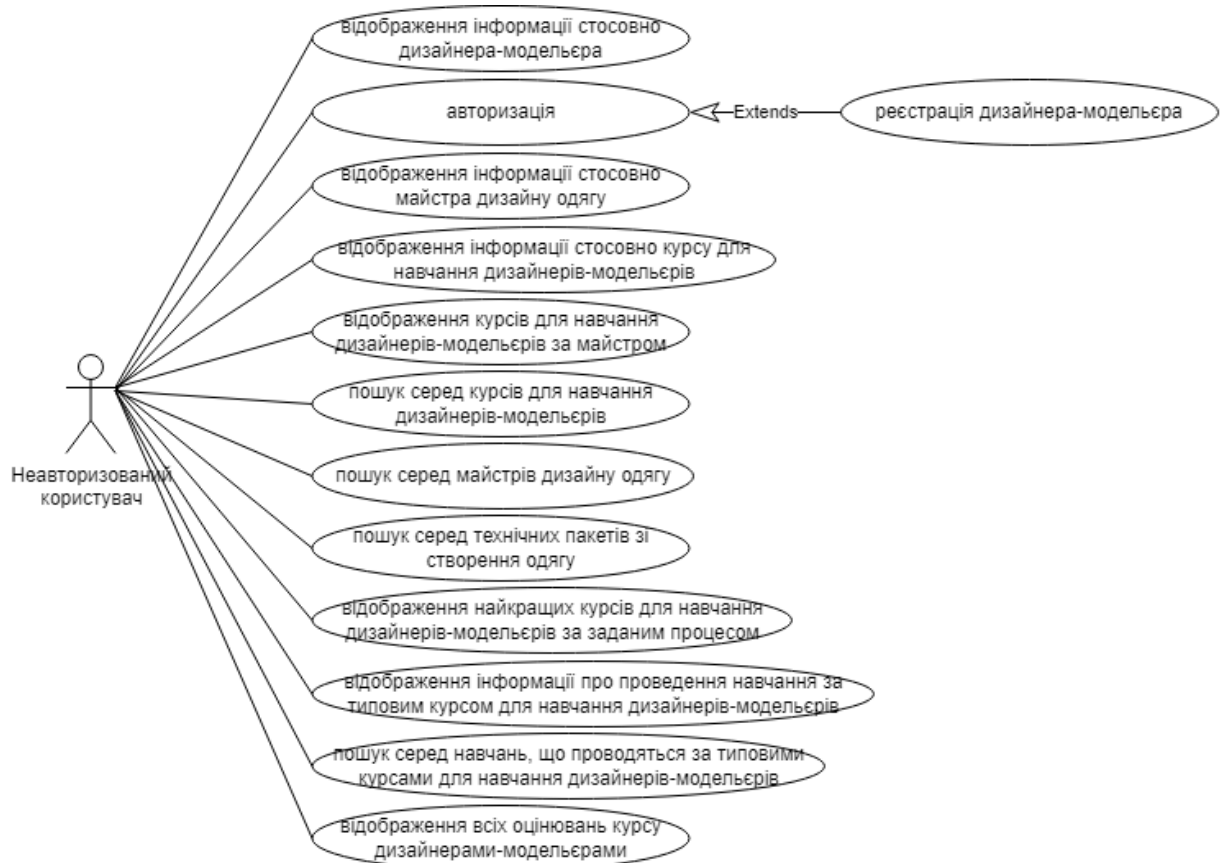


Рисунок 3.1 – Діаграма прецедентів неавторизованого користувача

Дизайнер-модельєр має забезпечуватися програмою сценаріями:

- відображення інформації стосовно дизайнера-модельєра;
- додавання фотографії дизайнера-модельєра;
- додавання програмних вмінь дизайнера-модельєра;

- додавання дизайнерських вмінь дизайнера-модельєра;
- додавання досвіду роботи дизайнера-модельєра;
- додавання сильних сторін дизайнера-модельєра;
- авторизація дизайнера-модельєра;
- підписування на колекції робіт дизайнера-модельєра;
- відображення колекцій дизайнерів-модельєрів з підписок;
- відображення дизайнерів-модельєрів з підписок;
- створення колекції одягу дизайнера-модельєра;
- відображення всіх колекцій дизайнера-модельєра;
- створення технічного пакету за колекцією одягу;
- відображення повного вмісту технічного пакету дизайнера-модельєра;
- відображення всіх технічних пакетів за колекцією одягу;
- збереження технічного пакету за колекцією одягу у вкладку;
- відображення технічних пакетів зі вкладок;
- збереження колекції одягу у вкладку;
- відображення колекцій одягу зі вкладок;
- відображення інформації стосовно майстра дизайну одягу;
- відображення інформації стосовно курсу для навчання дизайнерів-модельєрів;
- відображення курсів для навчання дизайнерів-модельєрів за майстром;
- пошук серед курсів для навчання дизайнерів-модельєрів;
- пошук серед майстрів дизайну одягу;
- пошук серед технічних пакетів зі створення одягу;
- відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом;
- відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;

- пошук серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів;
- відображення всіх матеріалів за курсом для навчання дизайнерів-модельєрів;
- відображення матеріалу за курсом для навчання дизайнерів-модельєрів;
- подання заявки на навчання за курсом для дизайнерів-модельєрів, що включає сценарій прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання;
- відображення всіх курсів, які відвідує дизайнер-модельєр;
- оцінювання дизайнером-модельєром курсу;
- відображення всіх оцінювань курсу дизайнерами-модельєрами;
- відображення всіх оцінювань цьому дизайнеру-модельєру.

Альтернативні сценарії дизайнера-модельєра:

- вилучення фотографії дизайнера-модельєра, що є альтернативним для додавання фотографії дизайнера-модельєра та реалізується за умов вже наявності доданої фотографії дизайнера-модельєра, яка більше не має використовуватись;
- вилучення програмних вмінь дизайнера-модельєра, що є альтернативним для додавання програмних вмінь дизайнера-модельєра та реалізується за умов вже наявності доданого цього програмного вміння для цього дизайнера-модельєра, яке більше не має використовуватись;
- вилучення дизайнерських вмінь дизайнера-модельєра, що є альтернативним для додавання дизайнерських вмінь дизайнера-модельєра та реалізується за умов вже наявності доданого цього дизайнерського вміння для цього дизайнера-модельєра, яке більше не має використовуватись;
- вилучення досвіду роботи дизайнера-модельєра, що є альтернативним для додавання досвіду роботи для дизайнера-модельєра та реалізується за умов вже наявності цього досвіду роботи, внесеного для цього дизайнера-модельєра та який більше не має використовуватись;

– вилучення сильних сторін дизайнера-модельєра, що є альтернативним для додавання сильних сторін дизайнера-модельєра та реалізується за умов вже наявності внесеної цієї сильної сторони для цього дизайнера-модельєра, яка більше не має використовуватись;

– модифікування загальних даних дизайнера-модельєра, що є альтернативним для відображення інформації стосовно дизайнера-модельєра та реалізується за умов некоректності цих даних дизайнера-модельєра на поточний момент;

– реєстрація, що є альтернативним для авторизації дизайнера-модельєра, коли дизайнер-модельєр не має акаунта для реєстрації;

– скасування підписування на колекції робіт дизайнера-модельєра, що є альтернативним для підписування на колекції робіт дизайнера-модельєра та реалізується за умов наявності підписування цього дизайнера-модельєра на цю колекцію робіт іншого дизайнера-модельєра, коли таке підписування вже не релевантне;

– редагування колекції одягу дизайнера-модельєра, що є альтернативним для створення колекції одягу дизайнера-модельєра та реалізується за умов попереднього створення цієї колекції одягу дизайнера-модельєра та наявності некоректної інформації щодо неї;

– вилучення колекції одягу дизайнера-модельєра, що є альтернативним для створення колекції одягу дизайнера-модельєра та реалізується за умов попереднього створення цієї колекції одягу дизайнера-модельєра і відсутності в необхідності її представлення в програмі на думку цього дизайнера-модельєра;

– редагування технічного пакету за колекцією одягу, що є альтернативним для створення технічного пакету за колекцією одягу та реалізується за умов наявності такого технічного пакету і наявності некоректної інформації щодо нього за даними самого дизайнера-модельєра;

– вилучення технічного пакету за колекцією одягу, що є альтернативним для створення технічного пакету за колекцією одягу та

реалізується за умов наявності такого технічного пакету і відсутності необхідності представлення такого технічного пакету в програмі на думку цього дизайнера-модельєра;

- скасування збереження технічного пакету за колекцією одягу у вкладку, що є альтернативним для збереження технічного пакету за колекцією одягу у вкладку та реалізується за умов попереднього збереження цим дизайнером-модельєром цього технічного пакету і відсутності потреби в такій вкладці у цього дизайнера-модельєра;

- скасування збереження колекції одягу у вкладку, що є альтернативним для збереження колекції одягу у вкладку та реалізується за умов попереднього збереження цим дизайнером-модельєром цієї колекції одягу у вкладку і відсутності потреби в такій вкладці у цього дизайнера-модельєра;

- скасування заявки на навчання за курсом для дизайнерів-модельєрів, що є альтернативним для подання заявки на навчання за курсом для дизайнерів-модельєрів та реалізується за умов попередньо поданої і непідтвердженої заявки на навчання за цим курсом для дизайнерів-модельєрів цим дизайнером-модельєром та відсутності бажання у нього навчатися за цим курсом на поточний момент;

- вилучення оцінювання дизайнером-модельєром курсу, що є альтернативним для оцінювання дизайнером-модельєром курсу та реалізується за умов створеної попередньо оцінки за цим курсом цим дизайнером-модельєром та розумінні некоректних внесених ним результатів (рис. 3.2).

Майстер дизайну одягу має забезпечуватися програмою сценаріями:

- відображення інформації стосовно дизайнера-модельєра;
- підписування на колекції робіт дизайнера-модельєра;
- відображення колекцій дизайнерів-модельєрів з підписок;
- відображення повного вмісту технічного пакету дизайнера-модельєра.

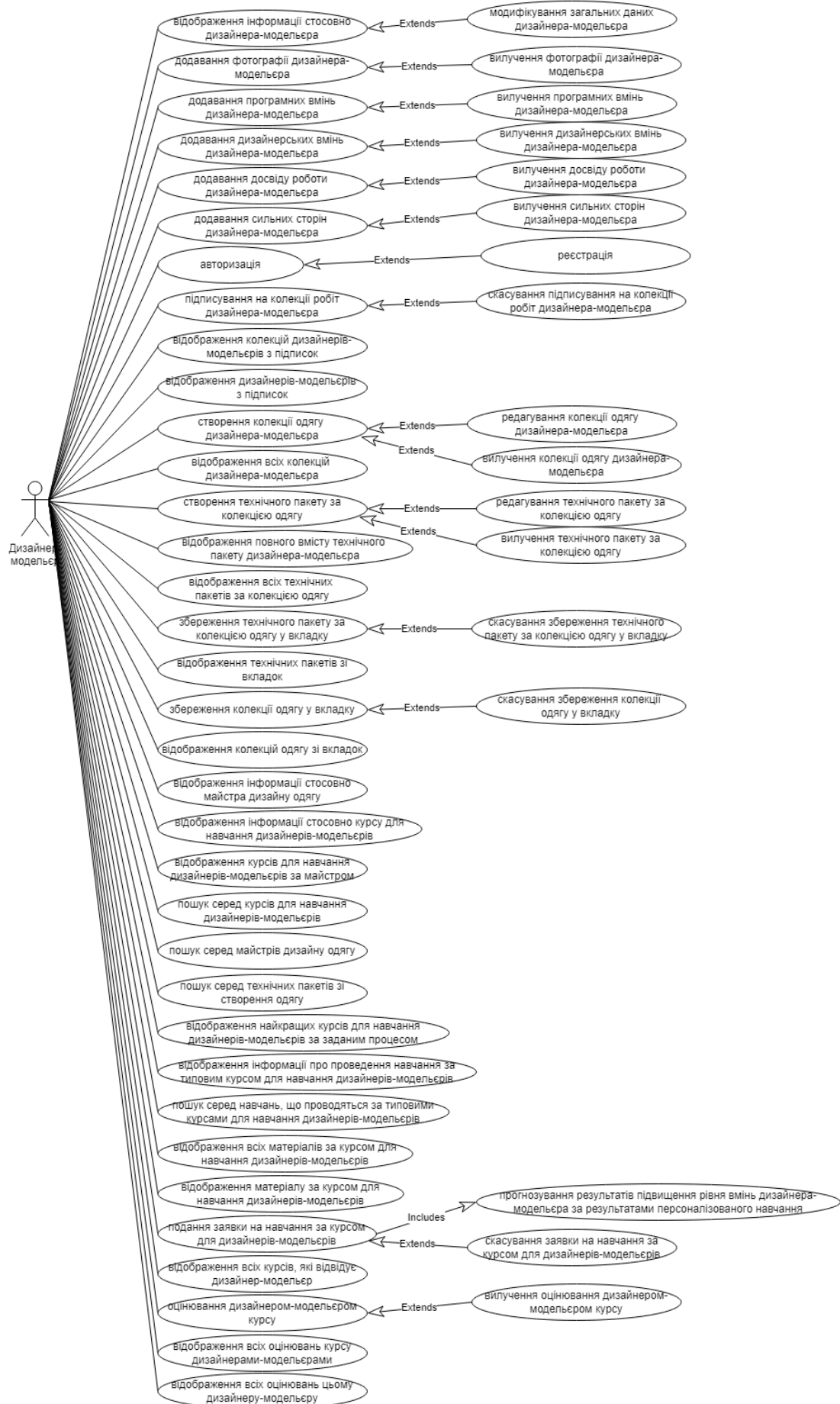


Рисунок 3.2 – Діаграма прецедентів дизайнера-модельєра

Також майстер дизайну одягу має забезпечуватися програмою сценаріями:

- відображення дизайнерів-модельєрів з підписок;
- відображення всіх колекцій дизайнера-модельєра;
- відображення всіх технічних пакетів за колекцією одягу;
- збереження технічного пакету за колекцією одягу у вкладку;
- відображення технічних пакетів зі вкладок;
- збереження колекції одягу у вкладку;
- відображення колекцій одягу зі вкладок;
- авторизація майстрів дизайну одягу;
- відображення інформації стосовно майстра дизайну одягу;
- оцінювання рівня дизайнера-модельєра;
- відображення всіх оцінювань курсу дизайнерами-модельєрами;
- відображення всіх оцінювань, виставлених дизайнеру-модельєру;
- додавання типового курсу для навчання дизайнерів-модельєрів;
- відображення інформації стосовно курсу для навчання дизайнерів-модельєрів;
- відображення курсів для навчання дизайнерів-модельєрів за майстром;
- пошук серед курсів для навчання дизайнерів-модельєрів;
- пошук серед майстрів дизайну одягу;
- пошук серед технічних пакетів зі створення одягу;
- відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом;
- створення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- пошук серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів;

- відображення всіх матеріалів за власним курсом для навчання дизайнерів-модельєрів;
- відображення матеріалу за власним курсом для навчання дизайнерів-модельєрів;
- публікація матеріалу за курсом для навчання дизайнерів-модельєрів;
- оцінювання результатів навчання дизайнера-модельєра за курсом;
- висунення заявки-пропозиції на навчання за курсом для дизайнерів-модельєрів, що включає сценарій прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання;
- підтвердження заявки на навчання за курсом для дизайнерів-модельєрів;
- відображення поданих заявок на навчання за курсом для дизайнерів-модельєрів;
- відображення всіх дизайнерів-модельєрів, які відвідують курс.

Альтернативні сценарії майстра дизайну одягу:

- скасування підписування на колекції робіт дизайнера-модельєра, що є альтернативним для підписування на колекції робіт дизайнера-модельєра та реалізується за умов наявності й неактуальності такого підписування на колекції робіт дизайнера-модельєра;
- скасування збереження технічного пакету за колекцією одягу у вкладку, що є альтернативним для збереження технічного пакету за колекцією одягу у вкладку та реалізується за умов вже збереженого технічного пакету за колекцією одягу у вкладку цим майстром і відсутності необхідності в такій вкладці на поточний момент;
- скасування збереження колекції одягу у вкладку, що є альтернативним для збереження колекції одягу у вкладку та реалізується за умов вже збереженої колекції одягу у вкладку цим майстром і відсутності необхідності в такій вкладці на поточний момент;
- скасування оцінювання рівня дизайнера-модельєра, що є альтернативним для оцінювання рівня дизайнера-модельєра та реалізується за

умов виконаного раніше такого оцінювання цим майстром цього дизайнера-модельєра і визнання некоректності такого оцінювання цим майстром з дизайну одягу;

- модифікування типового курсу для навчання дизайнерів-модельєрів, що є альтернативним для додавання типового курсу для навчання дизайнерів-модельєрів та реалізується за умов створення такого типового курсу для навчання дизайнерів-модельєрів цим майстром і наявності некоректної інформації щодо нього за даними самого майстра;

- вилучення типового курсу для навчання дизайнерів-модельєрів, що є альтернативним для додавання типового курсу для навчання дизайнерів-модельєрів та реалізується за умов створення такого типового курсу для навчання дизайнерів-модельєрів цим майстром і відсутності необхідності в представленні такого курсу з точки зору майстра;

- модифікування публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, що є альтернативним для створення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів та реалізується за умов виконаної цієї публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів цим майстром і наявності некоректної інформації щодо неї за даними самого майстра;

- вилучення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, що є альтернативним для створення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів та реалізується за умов виконаної цієї публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів цим майстром і відсутності необхідності в представленні такого навчання з точки зору майстра (рис. 3.3).

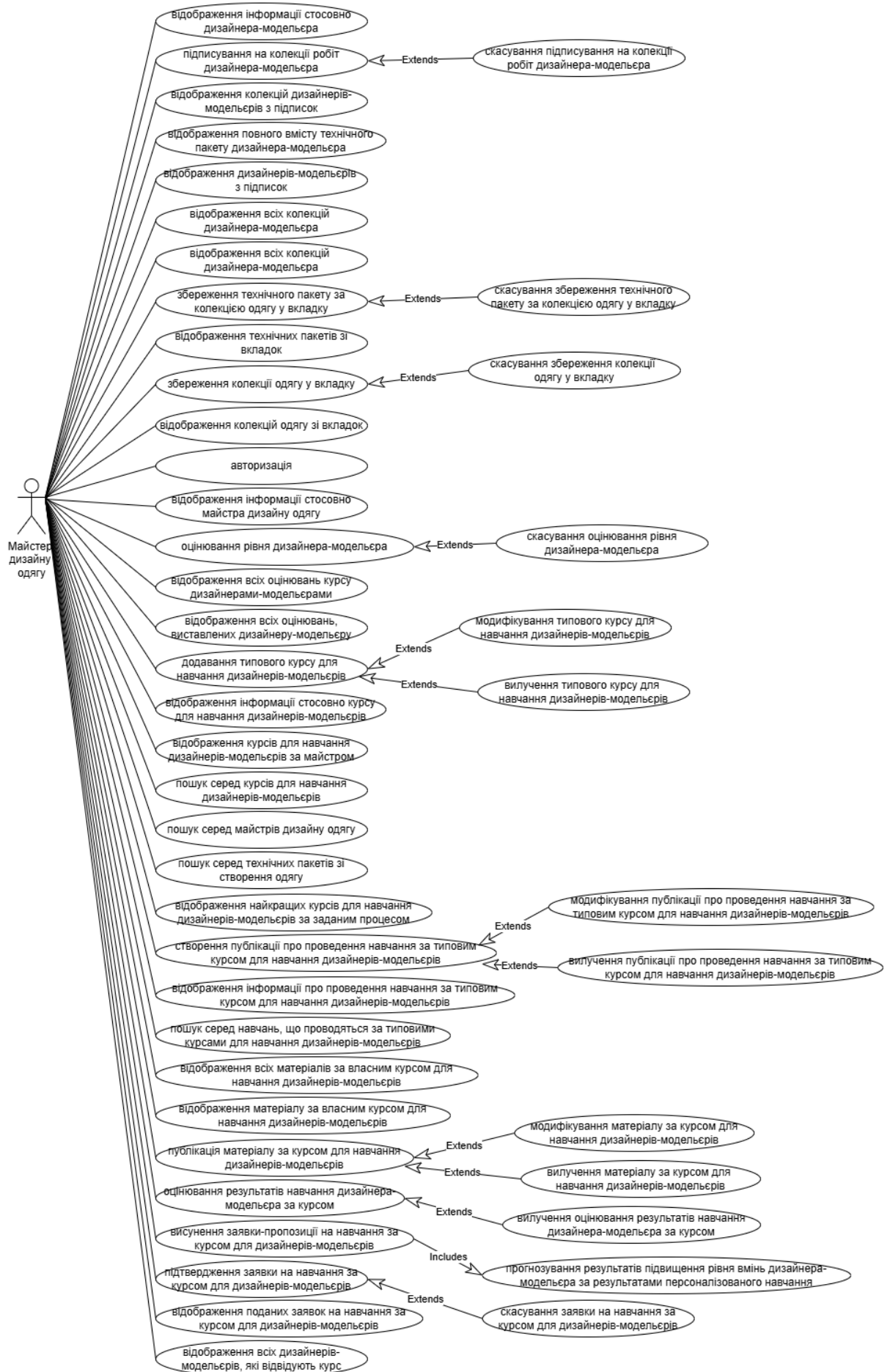


Рисунок 3.3 – Діаграма прецедентів майстра дизайну одягу

Також альтернативними сценаріями для майстра дизайну одягу є:

- модифікування матеріалу за курсом для навчання дизайнерів-модельєрів, що є альтернативним для публікації матеріалу за курсом для навчання дизайнерів-модельєрів та реалізується за умов виконаного збереження цього матеріалу за курсом для навчання дизайнерів-модельєрів цим майстром і наявності некоректної інформації щодо нього за даними самого майстра;

- вилучення матеріалу за курсом для навчання дизайнерів-модельєрів, що є альтернативним для публікації матеріалу за курсом для навчання дизайнерів-модельєрів та реалізується за умов виконаного збереження цього матеріалу за курсом для навчання дизайнерів-модельєрів цим майстром і відсутності необхідності в наданні такого матеріалу з точки зору майстра;

- вилучення оцінювання результатів навчання дизайнера-модельєра за курсом, що є альтернативним для оцінювання результатів навчання дизайнера-модельєра за курсом та реалізується за умов виконаного цього оцінювання результатів навчання дизайнера-модельєра за курсом цим майстром і некоректності внесених даних;

- скасування заявки на навчання за курсом для дизайнерів-модельєрів, що є альтернативним для підтвердження заявки на навчання за курсом для дизайнерів-модельєрів та реалізується за умов подання такої заявки дизайнером-модельєром на курс цього майстра або подання заявки-пропозиції цим майстром і відсутності необхідності в реалізації такого навчання на думку майстра.

Адміністратор має забезпечуватися програмою сценаріями:

- відображення повного вмісту технічного пакету дизайнера-модельєра;

- авторизація;

- відображення інформації стосовно дизайнера-модельєра;

- вилучення фотографії дизайнера-модельєра;

- вилучення програмних вмінь дизайнера-модельєра;

- вилучення дизайнерських вмінь дизайнера-модельєра;
- вилучення досвіду роботи дизайнера-модельєра;
- вилучення сильних сторін дизайнера-модельєра;
- відображення всіх колекцій дизайнера-модельєра;
- відображення всіх технічних пакетів за колекцією одягу;
- реєстрація майстрів дизайну одягу;
- відображення інформації стосовно майстра дизайну одягу;
- відображення всіх оцінювань курсу дизайнерами-модельєрами;
- відображення всіх оцінювань, виставлених дизайнеру-модельєру;
- відображення інформації стосовно курсу для навчання дизайнерів-модельєрів;
- відображення курсів для навчання дизайнерів-модельєрів за майстром;
- пошук серед курсів для навчання дизайнерів-модельєрів;
- пошук серед майстрів дизайну одягу;
- пошук серед технічних пакетів зі створення одягу;
- відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом;
- відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- пошук серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів;
- відображення всіх матеріалів за курсом для навчання дизайнерів-модельєрів;
- відображення матеріалу за курсом для навчання дизайнерів-модельєрів;
- підтвердження заявки на навчання за курсом для дизайнерів-модельєрів;
- відображення поданих заявок на навчання за курсом для дизайнерів-модельєрів;

- відображення всіх дизайнерів-модельєрів, які відвідують курс;
- відображення всіх курсів, які відвідує дизайнер-модельєр;
- створення моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання;
- навчання моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання.

Альтернативні сценарії адміністратора:

- модифікування загальних даних дизайнера-модельєра, що є альтернативним для відображення інформації стосовно дизайнера-модельєра та реалізується за умов наявності даних дизайнера-модельєра, які порушують правила;
- редагування колекції одягу дизайнера-модельєра, що є альтернативним для відображення всіх технічних пакетів за колекцією одягу та реалізується за умов наявності даних колекції одягу дизайнера-модельєра, які порушують правила та які можуть бути усунені шляхом внесення змін;
- вилучення колекції одягу дизайнера-модельєра, що є альтернативним для відображення всіх технічних пакетів за колекцією одягу та реалізується за умов наявності даних колекції одягу дизайнера-модельєра, які порушують правила та які не можуть бути усунені шляхом внесення змін;
- редагування технічного пакету за колекцією одягу, що є альтернативним для відображення повного вмісту технічного пакету дизайнера-модельєра та реалізується за умов наявності даних технічного пакету за колекцією одягу, які порушують правила та які можуть бути усунені шляхом внесення змін;
- вилучення технічного пакету за колекцією одягу, що є альтернативним для відображення повного вмісту технічного пакету дизайнера-модельєра та реалізується за умов наявності даних технічного пакету за колекцією одягу, які порушують правила та які не можуть бути усунені шляхом внесення змін;

– модифікування загальних даних майстра дизайну одягу, що є альтернативним для відображення інформації стосовно майстра дизайну одягу та реалізується за умов наявності даних майстра дизайну одягу, які порушують правила;

– скасування оцінювання рівня дизайнера-модельєра майстром, що є альтернативним для відображення всіх оцінювань, виставлених дизайнеру-модельєру, та реалізується за умов наявності такого оцінювання рівня дизайнера-модельєра майстром, дані в якому порушують правила;

– вилучення оцінювання дизайнером-модельєром курсу, що є альтернативним для відображення всіх оцінювань курсу дизайнерами-модельєрами та реалізується за умов наявності такого оцінювання дизайнером-модельєром курсу, дані в якому порушують правила;

– вилучення оцінювання результатів навчання дизайнера-модельєра за курсом, що є альтернативним для відображення всіх оцінювань, виставлених дизайнеру-модельєру та реалізується за умов наявності такого оцінювання результатів навчання дизайнера-модельєра за курсом, дані в якому порушують правила;

– модифікування типового курсу для навчання дизайнерів-модельєрів, що є альтернативним для відображення інформації стосовно курсу для навчання дизайнерів-модельєрів та реалізується за умов наявності даних типового курсу для навчання дизайнерів-модельєрів, які порушують правила та які можуть бути усунені шляхом внесення змін;

– вилучення типового курсу для навчання дизайнерів-модельєрів, що є альтернативним для відображення інформації стосовно курсу для навчання дизайнерів-модельєрів та реалізується за умов наявності даних типового курсу для навчання дизайнерів-модельєрів, які порушують правила та які не можуть бути усунені шляхом внесення змін;

– модифікування публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, що є альтернативним для відображення інформації про проведення навчання за типовим курсом для

навчання дизайнерів-модельєрів та реалізується за умов наявності даних публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, які порушують правила та які можуть бути усунені шляхом внесення змін;

- вилучення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, що є альтернативним для відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів та реалізується за умов наявності даних публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, які порушують правила та які не можуть бути усунені шляхом внесення змін;

- модифікування матеріалу за курсом для навчання дизайнерів-модельєрів, що є альтернативним для відображення матеріалу за курсом для навчання дизайнерів-модельєрів та реалізується за умов наявності даних матеріалу за курсом для навчання дизайнерів-модельєрів, які порушують правила та які можуть бути усунені шляхом внесення змін;

- вилучення матеріалу за курсом для навчання дизайнерів-модельєрів, що є альтернативним для відображення матеріалу за курсом для навчання дизайнерів-модельєрів та реалізується за умов наявності даних матеріалу за курсом для навчання дизайнерів-модельєрів, які порушують правила та які не можуть бути усунені шляхом внесення змін;

- скасування заявки на навчання за курсом для дизайнерів-модельєрів, що є альтернативним для підтвердження заявки на навчання за курсом для дизайнерів-модельєрів та реалізується за умов необхідності виконання такої дії, оскільки вона не може бути виконана самим майстром з дизайну одягу (рис. 3.4).

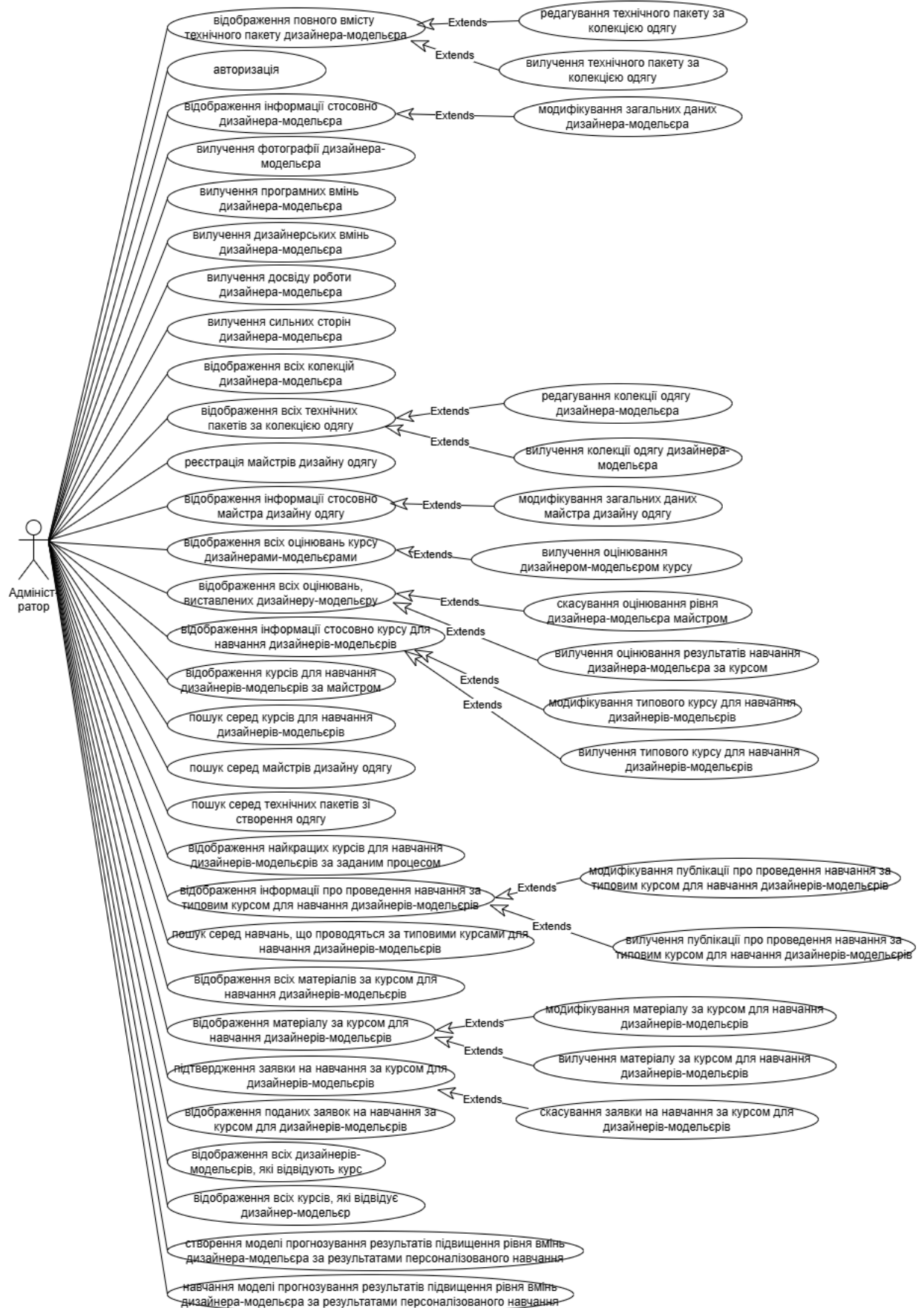


Рисунок 3.4 – Діаграма прецедентів адміністратора

3.3 Проектування структури бази даних

Структура бази даних (БД) ПЗ персоналізованого навчання дизайнерів-модельєрів визначена таким чином, щоб забезпечити підтримку всіх функціональних можливостей, що у розрізі сценаріїв використання приведені в підрозділі 3.2.

Таблиця `fashiondesigner` зберігає дані дизайнерів-модельєрів за характеристиками:

- `id` – ідентифікатор дизайнера-модельєра;
- `designeruser` – акаунт Django, що застосовується дизайнером-модельєром (зв'язок «один-до-одного» до таблиці `user`);
- `fdartname` – творчий псевдонім дизайнера-модельєра;
- `fddob` – дата народження дизайнера-модельєра;
- `fdphone` – номер телефону дизайнера-модельєра;
- `fdinstagramprofile` – вебадреса профілю дизайнера-модельєра в мережі Instagram;
- `fdwebportfolio` – вебадреса портфоліо дизайнера-модельєра;
- `designdescription` – текстова інформація про дизайнера-модельєра (біографія, опис особливостей);
- `fdstrengths` – текстове представлення сильних сторін дизайнера-модельєра;
- `fdeducationlevel` – рівень освіти дизайнера-модельєра;
- `fdeducationtext` – текстове представлення отриманої дизайнером-модельєром освіти;
- `fdactivities` – текстове представлення активностей, в яких брав участь дизайнер-модельєр;
- `foundationalscore` – середня оцінка максимум 3 останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр;
- `digitalcompetencescore` – середня оцінка цифрових компетентностей дизайнера-модельєра, яку було виставлено максимум 3 останні рази за

вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником);

– *initiativesscore* – середня оцінка рівня підприємницьких вмінь дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником);

– *problemsolvingscore* – середня оцінка здібностей вирішувати комплексні проблеми дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником);

– *creativityscore* – середня оцінка здібностей до креативності та іноваційності дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником);

– *communicationscore* – середня оцінка комунікаційних здібностей дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником);

– *engagementscore* – середня оцінка залученості дизайнера-модельєра до виконання завдань курсів навчання, яку було виставлено максимум 3 останні рази за вивченими курсами (3 останні курси або більше, якщо за якісь з останніх курсів не було виставлено оцінку дизайнеру-модельєру за цим показником).

Таблиця *fashiondesignerimage* зберігає дані фотографій дизайнерів-модельєрів за характеристиками:

– *id* – ідентифікатор фотографії дизайнера-модельєра;

– *fashiondesigner* – дизайнер-модельєр, який показаний на фотографії (зв'язок «один-до-багатьох» до таблиці *fashiondesigner*);

– `designerimage` – дані фотографії, на якій показаний цей дизайнер-модельєр;

– `coverness` – визначення того, що фотографія є обкладинкою (основною для представлення дизайнера-модельєра).

Таблиця `fashiondesignerconnection` зберігає дані підписувань на колекції робіт дизайнерів-модельєрів за характеристиками:

– `id` – ідентифікатор підписування на колекції робіт дизайнера-модельєра;

– `fashiondesignerstarter` – дизайнер-модельєр, який підписався на колекцію робіт іншого дизайнера-модельєра (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`);

– `fashiondesignerreceiver` – дизайнер-модельєр, на колекції робіт якого підписався інший дизайнер-модельєр (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`);

– `connectiondt` – час і дата підписування на колекції робіт дизайнера-модельєра.

Таблиця `fashiondesignerssoftwareskill` зберігає дані програмних вмінь дизайнера-модельєра за характеристиками:

– `id` – ідентифікатор представленого програмного вміння дизайнера-модельєра;

– `fashiondesigner` – дизайнер-модельєр, який має ці програмні вміння (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`);

– `designersoftware` – програмне забезпечення, яке вміє застосовувати в своїй роботі дизайнер-модельєр (зв'язок «один-до-багатьох» до таблиці `designersoftware`);

– `softwareskilllevel` – рівень цього програмного вміння у цього дизайнера-модельєра (оцінка, від 0 до 5);

– `softwareskillleveltext` – текстове представлення рівня цього програмного вміння у цього дизайнера-модельєра.

Таблиця `designersoftware` зберігає дані програмного забезпечення, яке використовується дизайнерами-модельєрами в їх роботі, за характеристиками:

- `id` – ідентифікатор програмного забезпечення, яке використовується дизайнерами-модельєрами в їх роботі;
- `softwarefullcaption` – повна назва програмного забезпечення, яке використовується дизайнерами-модельєрами в їх роботі.

Таблиця `designskill` зберігає дані можливих дизайнерських вмінь за характеристиками:

- `id` – ідентифікатор дизайнерського вміння;
- `designskilltype` – назва дизайнерського вміння.

Таблиця `fashiondesignerskill` зберігає дані дизайнерських вмінь дизайнерів-модельєрів за характеристиками:

- `id` – ідентифікатор дизайнерського вміння дизайнера-модельєра;
- `designskill` – дизайнерське вміння, яким характеризується цей дизайнер-модельєр (зв'язок «один-до-багатьох» до таблиці `designskill`);
- `fashiondesigner` – дизайнер-модельєр, який має це дизайнерське вміння (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`).

Таблиця `designcompany` зберігає дані компаній з дизайну одягу за характеристиками:

- `id` – ідентифікатор компанії, що працює в сфері дизайну, виробництва одягу;
- `designlogo` – логотип компанії, що працює в сфері дизайну, виробництва одягу;
- `designcompanytitle` – назва компанії, що працює в сфері дизайну, виробництва одягу.

Таблиця `designexperience` зберігає дані досвіду роботи дизайнерів-модельєрів за характеристиками:

- `id` – ідентифікатор запису досвіду роботи дизайнера-модельєра;

– designcompany – компанія, що працює в сфері дизайну, виробництва одягу та в якій працював або працює цей дизайнер-модельєр (зв'язок «один-до-багатьох» до таблиці designcompany);

– fashiondesigner – дизайнер-модельєр, який працював в цій компанії з дизайну або виробництва одягу (зв'язок «один-до-багатьох» до таблиці fashiondesigner);

– designposition – посада, на якій працював цей дизайнер-модельєр у цій компанії з дизайну або виробництва одягу (зв'язок «один-до-багатьох» до таблиці designposition);

– designjobstartdate – дата початку роботи цього дизайнера-модельєра у цій компанії з дизайну або виробництва одягу;

– designjobenddate – дата закінчення роботи цього дизайнера-модельєра у цій компанії з дизайну або виробництва одягу;

– designactivities – текстовий опис відповідальності за робочі завдання цього дизайнера-модельєра у цій компанії з дизайну та виробництва одягу.

Таблиця designposition зберігає дані можливих посад, на яких працюють дизайнери-модельєри, за характеристиками:

– id – ідентифікатор можливої посади, на якій може працювати дизайнер-модельєр;

– designpositiontitle – назва посади, на якій може працювати дизайнер-модельєр.

Таблиця typicalfashionstrength зберігає дані типових сильних сторін дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор типової сильної сторони дизайнера-модельєра;

– fashionstrengthtype – назва типової сильної сторони дизайнера-модельєра.

Таблиця fashiondesignertypicalstrength зберігає дані сильних сторін конкретних дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор сильної сторони дизайнера-модельєра;

– `typicalfashionstrength` – типова сильна сторона, яка наявна у цього дизайнера-модельєра (зв'язок «один-до-багатьох» до таблиці `typicalfashionstrength`);

– `fashiondesigner` – дизайнер-модельєр, який має цю сильну сторону (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`).

Таблиця `fashionstyle` зберігає дані можливих стилів в одязі за характеристиками:

– `id` – ідентифікатор можливого стилю в одязі;

– `fashionstyletitle` – назва можливого стилю в одязі.

Таблиця `garment` зберігає дані можливих видів одягу (пальта, сукні, футболки тощо) за характеристиками:

– `id` – ідентифікатор можливого виду одягу;

– `garname` – назва можливого виду одягу.

Таблиця `fashioncollection` зберігає дані колекцій одягу дизайнерів-модельєрів за характеристиками:

– `id` – ідентифікатор колекції одягу дизайнера-модельєра;

– `colcaption` – назва колекції одягу дизайнера-модельєра;

– `fashiondesigner` – дизайнер-модельєр, який створив цю колекцію одягу (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`);

– `colldescription` – текстовий опис колекції одягу дизайнера-модельєра;

– `collstyle` – основний стиль, до якого належить ця колекція одягу дизайнера-модельєра (зв'язок «один-до-багатьох» до таблиці `fashionstyle`);

– `collpublicavailability` – можливість публічного перегляду цієї колекції одягу дизайнера-модельєра: якщо доступ не публічний, то тільки майстри, які є авторами курсів з навчання, на які самі дизайнери-модельєри подали заявки, можуть переглядати такі колекції одягу;

– `collpublicationdt` – час і дата публікації колекції одягу дизайнера-модельєра.

Таблиця `fashioncollectiontechpack` зберігає дані технічних пакетів зі створення одягу за характеристиками:

- id – ідентифікатор технічного пакета зі створення одягу;
- fashioncollection – колекція одягу, до якої належить цей технічний пакет зі створення одягу (зв'язок «один-до-багатьох» до таблиці fashioncollection);
- garment – вид одягу, який показаний на цьому технічному пакеті зі створення одягу в цій колекції одягу (зв'язок «один-до-багатьох» до таблиці garment);
- fashionseason – сезон, для якого призначено одяг з цього технічного пакету;
- packdescription – текстовий опис технічного пакету зі створення одягу;
- packpublicationdt – час і дата публікації цього технічного пакету зі створення одягу;
- garmentpattern – основний візерунок, який використовується в одязі за цим технічним пакетом;
- garmentshouldertype – тип рукава у одязі за цим технічним пакетом;
- garmentwaisttype – тип талії у одязі за цим технічним пакетом;
- garmentlength – довжина речі, створеної за цим технічним пакетом у базовому розмірі;
- garmentbuttons – текстовий опис ґудзиків, які використовуються у одязі за цим технічним пакетом;
- garmentthread – нитка, яка має використовуватися при створенні одягу за цим технічним пакетом (зв'язок «один-до-багатьох» до таблиці fashionthread).

Таблиця fashionthread зберігає дані ниток, які можуть використовуватися у виробництві одягу, за характеристиками:

- id – ідентифікатор нитки, що може використовуватися у виробництві одягу;
- threadtype – тип нитки, яка використовується у виробництві одягу;
- threadcolor – колір нитки, яка використовується у виробництві одягу.

Таблиця `fashioncollectiontechpackcolor` зберігає дані особливостей кольору технічного пакету зі створення одягу (кольорових варіацій) за характеристиками:

- `id` – ідентифікатор кольорової варіації технічного пакету зі створення одягу;
- `fashioncollectiontechpack` – технічний пакет зі створення одягу, до якого відноситься ця кольорова варіація (зв'язок «один-до-багатьох» до таблиці `fashioncollectiontechpack`);
- `colormark` – додатковий колір цієї варіації технічного пакету зі створення одягу;
- `maincolormark` – основний колір цієї варіації технічного пакету зі створення одягу;
- `usedescription` – текстовий опис особливостей цього технічного пакету зі створення одягу.

Таблиця `fashioncollectiontechpackimage` зберігає дані зображень креслень (малюнків, набросків) технічних пакетів зі створення одягу за характеристиками:

- `id` – ідентифікатор зображення креслення технічного пакету зі створення одягу;
- `fashioncollectiontechpackcolor` – кольорова варіація технічного пакету зі створення одягу, яка представлена на цьому кресленні (зв'язок «один-до-багатьох» до таблиці `fashioncollectiontechpackcolor`);
- `techpackimage` – самі дані зображення креслення технічного пакету зі створення одягу;
- `imagedescription` – текстовий опис зображення креслення технічного пакету зі створення одягу;
- `imagepackposition` – ракурс одягу, який показаний на цьому кресленні технічного пакету (спиною, обличчям, збоку).

Таблиця `fashioncollectiontechpackdetail` зберігає дані додаткового опису особливостей, співставлених з технічним пакетом, за характеристиками:

– `id` – ідентифікатор додаткового опису особливостей, співставлених з технічним пакетом;

– `fashioncollectiontechpack` – технічний пакет зі створення одягу, до якого відносяться ці особливості (зв'язок «один-до-багатьох» до таблиці `fashioncollectiontechpack`);

– `garmentpart` – частина одягу, до якої відноситься ця особливість (наприклад, комір, манжета тощо);

– `itemdescription` – додатковий опис особливостей, співставлених з технічним пакетом.

Таблиця `fashionmaterial` зберігає дані матеріалів тканин, які можуть використовуватися для виробництва одягу (вовна, бавовна тощо), за характеристиками:

– `id` – ідентифікатор матеріалу тканини, що може використовуватися для виробництва одягу;

– `fabricname` – назва матеріалу тканини, що може використовуватися для виробництва одягу.

Таблиця `fashioncollectiontechpackfabric` зберігає дані складу тканини, яку необхідно використовувати для виробництва одягу за технічним пакетом, за характеристиками:

– `id` – ідентифікатор складу тканини, що необхідно використовувати для виробництва одягу за технічним пакетом;

– `fashioncollectiontechpack` – технічний пакет, за яким для виробництва одягу використовується ця тканина (зв'язок «один-до-багатьох» до таблиці `fashioncollectiontechpack`);

– `fashionmaterial` – матеріал зі складу тканини, яка використовується для виробництва одягу за цим технічним пакетом (зв'язок «один-до-багатьох» до таблиці `fashionmaterial`);

– `fabricproportion` – відсоток вмісту цього матеріалу в тканині.

Таблиця `fashioncollectiontechpackfabricpart` зберігає дані співвідношення складу тканини з частиною одягу, для виготовлення якої вона використовується, за характеристиками:

- `id` – ідентифікатор співвідношення складу тканини з частиною одягу, для виготовлення якої вона використовується;
- `fashioncollectiontechpackfabric` – склад тканини, яка використовується для виготовлення цієї частини одягу (зв'язок «один-до-багатьох» до таблиці `fashioncollectiontechpack`);
- `garmentpart` – частина одягу, для виготовлення якої використовується ця тканина.

Таблиця `fashioncollectiontechpacksaving` зберігає дані вкладок, збережених за технічними пакетами одягу, за характеристиками:

- `id` – ідентифікатор вкладки, збереженої за технічним пакетом за одягом;
- `fashioncollectiontechpack` – технічний пакет для виготовлення одягу, який збережено у вкладку користувачем;
- `designeruser` – користувач, який зберіг собі у вкладку цей технічний пакет з виготовлення одягу (зв'язок «один-до-багатьох» до таблиці `user`).

Таблиця `fashioncollectionsaving` зберігає дані колекцій одягу дизайнерів-модельєрів за характеристиками:

- `id` – ідентифікатор збереженої вкладки колекції одягу дизайнера-модельєра;
- `designeruser` – користувач, який зберіг собі у вкладку цю колекцію одягу (зв'язок «один-до-багатьох» до таблиці `user`);
- `fashioncollection` – колекція одягу, яку у вкладку зберіг собі цей користувач (зв'язок «один-до-багатьох» до таблиці `fashioncollection`).

Таблиця `fashionmaster` зберігає дані майстрів дизайну одягу, які є користувачами програми, за характеристиками:

- `id` – ідентифікатор майстра дизайну одягу;

- masteruser – користувацький акаунт, який використовується цим майстром з дизайну одягу (зв'язок «один-до-одного» до таблиці user);
- fmartname – творчий псевдонім цього майстра з дизайну одягу;
- fmwebhub – вебсайт, де представлена інформація про цього майстра з дизайну одягу;
- fmadditionalurl – додаткова вебадреса, де представлена інформація про цього майстра з дизайну одягу;
- fashionachievements – досягнення, представлені текстом, цього майстра з дизайну одягу;
- fmdesignbio – біографія цього майстра з дизайну одягу;
- fmbasecountry – основна країна, в якій знаходиться цей майстер з виробництва одягу;
- fmexpertlevel – галузі дизайну, виробництва одягу, в яких є експертом цей майстер;
- fmtextonstyle – опис особливостей стилю цього майстра з виробництва одягу;
- fmtextoncolors – опис особливостей застосування кольорів у одязі цим майстром;
- fmtextonpatterns – опис особливостей застосування візерунків цим майстром з виробництва одягу.

Таблиця fashionmasterlevel зберігає дані оцінювання рівня дизайнера-модельєра майстром за характеристиками:

- id – ідентифікатор оцінювання рівня дизайнера-модельєра майстром;
- fashionmaster – майстер, який оцінив дизайнера-модельєра (зв'язок «один-до-багатьох» до таблиці fashionmaster);
- fashiondesigner – дизайнер-модельєр, здібності якого оцінив майстер (зв'язок «один-до-багатьох» до таблиці fashiondesigner);
- fashionlevelpoint – числове оцінювання рівня дизайнера-модельєра майстром;

– `fashionleveltext` – текстове оцінювання рівня дизайнера-модельєра майстром;

– `actualdt` – час і дата оцінювання рівня дизайнера-модельєра майстром.

Такі оцінювання відбуваються за результатами перегляду технічних пакетів дизайнерів-модельєрів майстрами, а не після проведення курсів.

Таблиця `fashionmasterimage` зберігає дані зображень майстрів з дизайну одягу за характеристиками:

– `id` – ідентифікатор фотографії майстра з дизайну одягу;

– `fashionmaster` – майстер з дизайну одягу, який показаний на фотографії (зв'язок «один-до-багатьох» до таблиці `fashionmaster`);

– `fmpersonimage` – дані фотографії майстра з дизайну одягу;

– `fmcoverness` – використання фотографії в якості основної для майстра з дизайну одягу (обкладинка).

Таблиця `fashionmastercourse` зберігає дані типових курсів для навчання дизайнерів-модельєрів за характеристиками:

– `id` – ідентифікатор типового курсу для навчання дизайнерів-модельєрів;

– `fashionmaster` – майстер з дизайну одягу, який проводить цей типовий курс для навчання дизайнерів-модельєрів (зв'язок «один-до-багатьох» до таблиці `fashionmaster`);

– `fashioncoursefulltitle` – повна назва типового курсу для навчання дизайнерів-модельєрів;

– `fashioncourseshorttitle` – коротка назва типового курсу для навчання дизайнерів-модельєрів для використання на сторінках у позиціях, де є обмежена кількість символів;

– `fcavailableplaces` – максимальна кількість місць, доступна для навчання за цим курсом для дизайнерів-модельєрів одночасно;

– `fccoverimage` – зображення, яке використовується як обкладинка типового курсу для навчання дизайнерів-модельєрів;

– videoforfcpresentation – відео для презентації типового курсу для навчання дизайнерів-модельєрів;

– purposedesignerlevel – рівень мистецтва, на заволодіння яким направлено цей типовий курс для навчання дизайнерів-модельєрів;

– fashioncoursetype – тип курсів для навчання дизайнерів-модельєрів, до якого цей належить (зв'язок «один-до-багатьох» до таблиці fashioncoursetype);

– fashioncourselevel – оцінка, яку було виставлено курсу дизайнерами-модельєрами за результатами навчання.

Таблиця fashioncoursetype зберігає дані типів курсів для навчання дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор типу курсів для навчання дизайнерів-модельєрів;

– fashionprocessname – назва типу курсів для навчання дизайнерів-модельєрів (відповідного процесу в дизайні або виготовленні одягу).

Таблиця fashionmastercoursemeeting зберігає дані публікацій про проведення навчання за типовими курсами для навчання дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;

– fashionmastercourse – типовий курс для навчання дизайнерів-модельєрів, за яким проводиться це навчання (зв'язок «один-до-багатьох» до таблиці fashionmastercourse);

– coursemeetingstart – час і дата початку навчання за цим курсом для дизайнерів-модельєрів;

– coursemeetingend – час і дата закінчення навчання за цим курсом для дизайнерів-модельєрів;

– meetavailableplaces – кількість місць, яка доступна для цього навчання за цим курсом для дизайнерів-модельєрів;

– meetoccupiedplaces – кількість місць, яка зайнята за цим навчанням за цим курсом для дизайнерів-модельєрів;

– meetcoverimage – зображення, яке використовується для цього навчання за цим курсом для дизайнерів-модельєрів;

– videoformeetpresentation – відео, яке використовується як презентація для цього навчання за цим курсом для дизайнерів-модельєрів;

– meetingtype – тип зустрічі (онлайн чи офлайн) для цього навчання за цим курсом для дизайнерів-модельєрів;

– hiddenmeetinginfo – прихована інформація для цього навчання за цим курсом для дизайнерів-модельєрів, яка надається тільки в тому випадку, якщо користувач отримав підтверджене місце для навчання;

– coursevenue – місце проведення зустрічі для навчання за курсом.

Таблиця coursevenue зберігає дані можливих місць проведення навчання дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор можливого місця проведення навчання дизайнерів-модельєрів;

– venue name – назва можливого місця проведення навчання дизайнерів-модельєрів;

– venue phone – номер телефону можливого місця проведення навчання дизайнерів-модельєрів;

– venue address – поштова адреса можливого місця проведення навчання дизайнерів-модельєрів;

– venue url – веб адреса можливого місця проведення навчання дизайнерів-модельєрів;

– venue city – місто, в якому знаходиться це можливе місце проведення навчання дизайнерів-модельєрів.

Таблиця fashionmastercoursestuff зберігає дані матеріалів за курсами для навчання дизайнерів-модельєрів за характеристиками:

– id – ідентифікатор матеріалу курсу для навчання дизайнерів-модельєрів;

– `fashionmastercourse` – типовий курс для навчання дизайнерів-модельєрів, за яким використовується цей матеріал (зв'язок «один-до-багатьох» до таблиці `fashionmastercourse`);

– `fashionstuffurl` – вебадреса матеріалу курсу для навчання дизайнерів-модельєрів;

– `fashionstuffdata` – дані матеріалу курсу для навчання дизайнерів-модельєрів;

– `fashionstuffshorttitle` – повна назва матеріалу курсу для навчання дизайнерів-модельєрів;

– `fashionstufffulltitle` – коротка назва матеріалу курсу для навчання дизайнерів-модельєрів;

– `activeness` – визначає, чи цей матеріал ще використовується при вивченні цього курсу для навчання дизайнерів-модельєрів, чи вже ні;

– `accesstype` – тип доступу до матеріалу курсу для навчання дизайнерів-модельєрів: загальний або персоналізований;

– `fashionstufftype` – тип матеріалу курсу для навчання дизайнерів-модельєрів;

– `fashionstuffindex` – порядковий номер даного матеріалу курсу для навчання дизайнерів-модельєрів серед всіх матеріалів за цим курсом;

– `activationstartdt` – час і дата, з яких цей матеріал курсу для навчання дизайнерів-модельєрів доступний;

– `activationenddt` – час і дата, до яких цей матеріал курсу для навчання дизайнерів-модельєрів доступний.

Таблиця `fashionmastercourseapplication` зберігає дані заявок на вивчення курсів від дизайнерів-модельєрів (або майстрів) за характеристиками:

– `id` – ідентифікатор заявки на вивчення курсу від дизайнера-модельєра (або майстра);

– `fashiondesigner` – дизайнер-модельєр, який створив цю заявку на вивчення курсу або якому адресована заявка від майстра (зв'язок «один-до-багатьох» до таблиці `fashiondesigner`);

– `fashionmastercoursemeeting` – конкретне проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, за яким подав цю заявку дизайнер-модельєр або майстер (зв'язок «один-до-багатьох» до таблиці `fashionmastercoursemeeting`);

– `fashionapplicationdt` – час і дата, коли ця заявка на вивчення курсу від дизайнера-модельєра (або майстра) була подана;

– `initializer` – сторона-ініціатор подання заявки (майстер або дизайнер-модельєр, оскільки майстер за результатами роботи з програмою може знайти сам талановитих дизайнерів-модельєрів та запропонувати навчання);

– `fashionapplicationstatus` – стан розгляду заявка на вивчення курсу від дизайнера-модельєра (або майстра);

– `fashiondesignerlevelpoint` – чисельна оцінка, виставлена загалом за вивчення курсу дизайнеру-модельєру;

– `fashiondesignerleveltext` – текстова оцінка, виставлена загалом за вивчення курсу дизайнеру-модельєру;

– `fashioncourselevelpoint` – чисельна оцінка, виставлена загалом курсу дизайнером-модельєром;

– `fashioncourseleveltext` – текстова оцінка, виставлена загалом курсу дизайнером-модельєром;

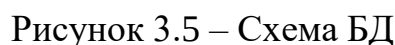
– `foundationalscorebefore` – середня оцінка максимум 3 останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр (до моменту початку вивчення цього курсу);

– `foundationalscore` – результат у балах дизайнера-модельєра за вивчення курсу;

– `predictedfoundationalscore` – прогнозований моделлю результат у балах дизайнера-модельєра за вивчення курсу;

– `digitalcompetencescore` – оцінка цифрових компетентностей дизайнера-модельєра за результатами вивчення курсу;

– `initiativescore` – оцінка рівня підприємницьких вмінь дизайнера-модельєра за результатами вивчення курсу;



Таблиця `fashionmastercoursestuffapplication` зберігає дані щодо персоналізованого доступу до матеріалів курсів для навчання дизайнерів-модельєрів за характеристиками:

- `id` – ідентифікатор визначання персоналізованого доступу до матеріалів курсів для навчання дизайнерів-модельєрів;

- `fashionmastercoursestuff` – матеріал курсу для навчання дизайнерів-модельєрів, який доступний для персоналізованого доступу за цією заявкою на вивчення курсу від дизайнера-модельєра (зв'язок «один-до-багатьох» до таблиці `fashionmastercoursestuff`);

- `fashionmastercourseapplication` – заявка на вивчення курсу від дизайнера-модельєра, за якою визначено доступ до цього матеріалу (зв'язок «один-до-багатьох» до таблиці `fashionmastercourseapplication`).

3.4 Висновки за розділом 3

Вибір мови програмування Python, фреймворку Django, шаблону створення макету вебзастосунку, системи MySQL для управління БД для забезпечення в подальшому реалізації вебзастосунку виконано в цьому розділі. Виконано моделювання роботи дизайнера-модельєра, неавторизованого користувача, майстра дизайну одягу, адміністратора з програмою. Описано структуру БД.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Послідовність взаємодії користувача з програмою персоналізованого навчання дизайнерів-модельєрів

Функціональна схема програми персоналізованого навчання дизайнерів-модельєрів (рис. 4.1) побудована на основі логіки утворення меню, оскільки це потрібно було для організації шаблонів вебсторінок, які розроблялись у процесі створення програмного проєкту.

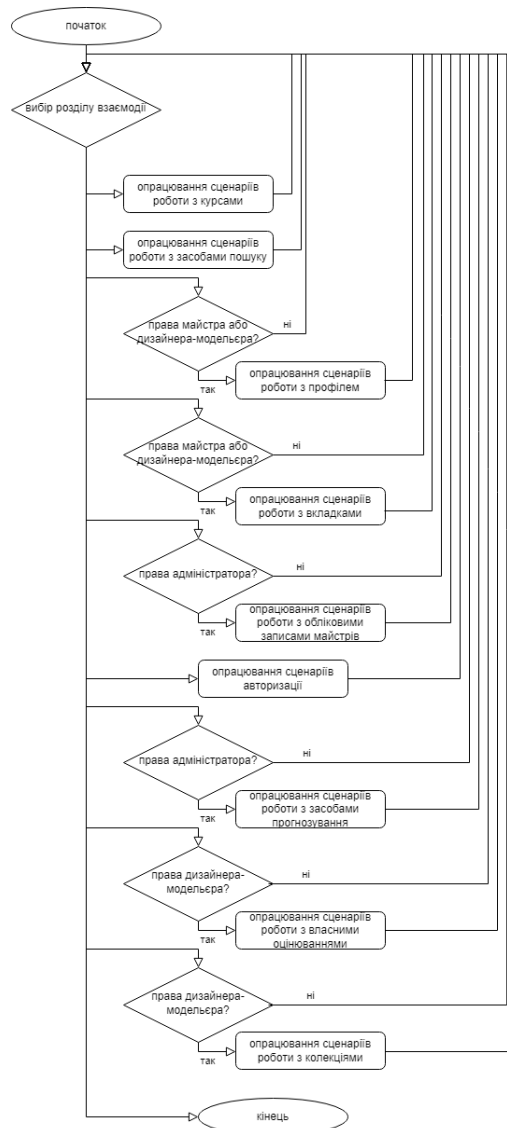


Рисунок 4.1 – Функціональна схема програми персоналізованого навчання дизайнерів-модельєрів

4.2 Рішення з реалізації вебзастосунку

Реалізація проєкту, що відповідає вебзастосунку персоналізованого навчання дизайнерів-модельєрів, виконана у чіткій відповідності з тією структурою, що має будь-який проєкт у Django.

Моделі проєкту персоналізованого навчання дизайнерів-модельєрів реалізовані в файлі `models.py` за структурою БД, що була описана в розділі 3, на основі підходу `code-first`, тобто вже структура таблиць автоматично Django при запуску відповідних команд виконання міграцій формувалась автоматично під представлені нижче класи моделей.

Модель `Fashiondesigner` реалізує клас дизайнерів-модельєрів з такими полями.

Поле `designeruser` реалізує визначення акаунта Django, що застосовується дизайнером-модельєром, як об'єкт класу `models.OneToOneField`, забезпечуючи зв'язок зі стандартною Django-моделлю `User`.

Поле `fdartname` реалізує творчий псевдонім дизайнера-модельєра як об'єкт класу `models.CharField`.

Поле `fddob` реалізує дату народження дизайнера-модельєра як об'єкт класу `models.DateField`.

Поле `fdphone` реалізує номер телефону дизайнера-модельєра як об'єкт класу `models.CharField`.

Поле `fdinstagramprofile` реалізує вебадресу профілю дизайнера-модельєра в мережі Instagram як об'єкт класу `models.URLField`.

Поле `fdwebportfolio` реалізує вебадресу портфоліо дизайнера-модельєра як об'єкт класу `models.URLField`.

Поле `designdescription` реалізує текстову інформацію про дизайнера-модельєра як об'єкт класу `models.TextField`.

Поле `fdstrengths` реалізує текстове представлення сильних сторін дизайнера-модельєра як об'єкт класу `models.TextField`.

Поле `fdeducationlevel` реалізує рівень освіти дизайнера-модельєра як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `EDUCATION_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного рівня освіти.

Поле `fdeducationtext` реалізує текстове представлення отриманої дизайнером-модельєром освіти як об'єкт класу `models.TextField`.

Поле `fdactivities` реалізує текстове представлення активностей, в яких брав участь дизайнер-модельєр, як об'єкт класу `models.TextField`.

Поле `foundationalscore` реалізує середню оцінку максимум 3 останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `digitalcompetencescore` реалізує середню оцінку цифрових компетентностей дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `initiativescore` реалізує середню оцінку рівня підприємницьких вмінь дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `problemsolvingscore` реалізує середню оцінку здібностей вирішувати комплексні проблеми дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `creativityscore` реалізує середню оцінку здібностей до креативності та інноваційності дизайнера-модельєра, яку було виставлено максимум 3 останні рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `communicationscore` реалізує середню оцінку комунікаційних здібностей дизайнера-модельєра, яку було виставлено максимум 3 останні

рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `engagementscore` реалізує середню оцінку залученості дизайнера-модельєра до виконання завдань курсів навчання, яку було виставлено максимум 3 останні рази за вивченими курсами, як об'єкт класу `models.DecimalField` з одним знаком після коми.

Поле `fashiondesignerskills` реалізує визначення дизайнерських вмінь цього дизайнера-модельєра як об'єкт класу `models.ManyToManyField` через зв'язок з класом `Designskill` («багато-до-багатьох»), таким чином реалізуючи таблицю `fashiondesignerskill` з БД, описану в розділі 3.

Поле `dashiondesignertypicalstrengths` реалізує визначення сильних сторін дизайнера-модельєра як об'єкт класу `models.ManyToManyField` через зв'язок з класом `Typicalfashionstrength` («багато-до-багатьох»), таким чином реалізуючи таблицю `fashiondesignertypicalstrength` з БД, описану в розділі 3.

Модель `Fashiondesignerimage` реалізує клас фотографій дизайнерів-модельєрів з такими полями.

Поле `fashiondesigner` реалізує визначення дизайнера-модельєра, який показаний на фотографії, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `designerimage` реалізує дані фотографії, на якій показаний цей дизайнер-модельєр, як об'єкт класу `models.ImageField`.

Поле `coverness` реалізує дані визначення того, що фотографія є обкладинкою, як об'єкт класу `models.BooleanField`.

Модель `Fashiondesignerconnection` реалізує клас підписувань на колекції робіт дизайнерів-модельєрів з такими полями.

Поле `fashiondesignerstarter` реалізує визначення дизайнера-модельєра, який підписався на колекцію робіт іншого дизайнера-модельєра, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner` зі зворотнім викликом колекції через ім'я `starters`.

Поле `fashiondesignerreceiver` реалізує визначення дизайнера-модельєра, на колекції робіт якого підписався інший дизайнер-модельєр, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner` зі зворотнім викликом колекції через ім'я `receivers`.

Поле `connectiondt` реалізує час і дату підписування на колекції робіт дизайнера-модельєра як об'єкт класу `models.DateTimeField`.

Модель `Fashiondesignersoftwareskill` реалізує клас програмних вмінь дизайнера-модельєра з такими полями.

Поле `fashiondesigner` реалізує визначення дизайнера-модельєра, який має ці програмні вміння, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `designersoftware` реалізує програмне забезпечення, яке вміє застосовувати в своїй роботі дизайнер-модельєр, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Designersoftware`.

Поле `softwareskilllevel` реалізує рівень цього програмного вміння у цього дизайнера-модельєра як об'єкт класу `models.IntegerField`.

Поле `softwareskillleveltext` реалізує текстове представлення рівня цього програмного вміння у цього дизайнера-модельєра як об'єкт класу `models.CharField`.

Модель `Designersoftware` реалізує клас програмного забезпечення, яке використовується дизайнерами-модельєрами в їх роботі, з полем `softwarefullcaption`, що реалізує повну назву програмного забезпечення, яке використовується дизайнерами-модельєрами в їх роботі, як об'єкт класу `models.CharField`.

Модель `Designskill` реалізує клас можливих дизайнерських вмінь з полем `designskilltype`, що реалізує назву дизайнерського вміння як об'єкт класу `models.CharField`.

Модель `Designcompany` реалізує клас компаній з дизайну одягу з такими полями.

Поле `designlogo` реалізує логотип компанії, що працює в сфері дизайну, виробництва одягу, як об'єкт класу `models.ImageField`.

Поле `designcompanytitle` реалізує назву компанії, що працює в сфері дизайну, виробництва одягу, як об'єкт класу `models.CharField`.

Модель `Designexperience` реалізує клас досвіду роботи дизайнерів-модельєрів з такими полями.

Поле `designcompany` реалізує визначення компанії, що працює в сфері дизайну, виробництва одягу та в якій працював або працює цей дизайнер-модельєр, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Designcompany`.

Поле `fashiondesigner` реалізує визначення дизайнера-модельєра, який працював в цій компанії з дизайну або виробництва одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `designposition` реалізує посаду, на якій працював цей дизайнер-модельєр у цій компанії з дизайну або виробництва одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Designposition`.

Поле `designjobstartdate` реалізує дату початку роботи цього дизайнера-модельєра у цій компанії з дизайну або виробництва одягу, як об'єкт класу `models.DateField`.

Поле `designjobenddate` реалізує дату закінчення роботи цього дизайнера-модельєра у цій компанії з дизайну або виробництва одягу, як об'єкт класу `models.DateField`.

Поле `designactivities` реалізує текстовий опис відповідальності за робочі завдання цього дизайнера-модельєра у цій компанії з дизайну та виробництва одягу як об'єкт класу `models.CharField`.

Модель `Designposition` реалізує клас можливих посад, на яких працюють дизайнери-модельєри, з полем `designpositiontitle`, яке реалізує назву посади, на якій може працювати дизайнер-модельєр, як об'єкт класу `models.CharField`.

Модель `Typicalfashionstrength` реалізує клас типових сильних сторін дизайнерів-модельєрів з полем `fashionstrengthtype`, який реалізує назву типової сильної сторони дизайнера-модельєра як об'єкт класу `models.CharField`.

Модель `Fashionstyle` реалізує клас можливих стилів в одязі з полем `fashionstyletitle`, яке реалізує назву можливого стилю в одязі як об'єкт класу `models.CharField`.

Модель `Garment` реалізує клас можливих видів одягу полем `garname`, яке реалізує назву можливого виду одягу як об'єкт класу `models.CharField`.

Модель `Fashioncollection` реалізує клас колекцій одягу дизайнерів-модельєрів з такими полями.

Поле `collcaption` реалізує назву колекції одягу дизайнера-модельєра як об'єкт класу `models.CharField`.

Поле `fashiondesigner` реалізує визначення дизайнера-модельєра, який створив цю колекцію одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `colldescription` реалізує текстовий опис колекції одягу дизайнера-модельєра як об'єкт класу `models.TextField`.

Поле `collstyle` реалізує визначення основного стилю, до якого належить ця колекція одягу дизайнера-модельєра, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionstyle`.

Поле `collpublicavailability` реалізує можливість публічного перегляду цієї колекції одягу дизайнера-модельєра як об'єкт класу `models.BooleanField`.

Поле `collpublicationdt` реалізує час і дату публікації колекції одягу дизайнера-модельєра як об'єкт класу `models.DateTimeField`.

Модель `Fashioncollectiontechpack` реалізує клас технічних пакетів зі створення одягу з такими полями.

Поле `fashioncollection` реалізує визначення колекції одягу, до якої належить цей технічний пакет зі створення одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollection`.

Поле `garment` реалізує визначення виду одягу, який показаний на цьому технічному пакеті зі створення одягу в цій колекції одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Garment`.

Поле `fashionseason` реалізує сезон, для якого призначено одяг з цього технічного пакету, як об'єкт класу `models.CharField`.

Поле `packdescription` реалізує текстовий опис технічного пакету зі створення одягу як об'єкт класу `models.TextField`.

Поле `packpublicationdt` реалізує час і дату публікації цього технічного пакету зі створення одягу як об'єкт класу `models.DateTimeField`.

Поле `garmentpattern` реалізує основний візерунок, який використовується в одязі за цим технічним пакетом, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `PATTERN_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного візерунку.

Поле `garmentshouldertype` реалізує тип рукава у одязі за цим технічним пакетом як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `SHOULDER_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного типу рукава.

Поле `garmentwaisttype` реалізує тип талії у одязі за цим технічним пакетом як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `WAISTLINE_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного типу талії.

Поле `garmentlength` реалізує довжину речі, створеної за цим технічним пакетом у базовому розмірі як об'єкт класу `models.DecimalField` з 2 знаками після коми.

Поле `garmentbuttons` реалізує текстовий опис гудзиків, які використовуються у одязі за цим технічним пакетом, як об'єкт класу `models.CharField`.

Поле `garmentthread` реалізує визначення нитки, яка має використовуватися при створенні одягу за цим технічним пакетом, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionthread`.

Моделю `Fashionthread` реалізує клас ниток, які можуть використовуватися у виробництві одягу, з такими полями.

Поле `threadtype` реалізує тип нитки, яка використовується у виробництві одягу, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `THREAD_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного типу нитки.

Поле `threadcolor` реалізує колір нитки, яка використовується у виробництві одягу, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `COLORS`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного кольору.

Моделю `Fashioncollectiontechpackcolor` реалізує клас особливостей кольору технічного пакету зі створення одягу (кольорових варіацій) з такими полями.

Поле `fashioncollectiontechpack` реалізує технічний пакет зі створення одягу, до якого відноситься ця кольорова варіація, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpack`.

Поле `colormark` реалізує додатковий колір цієї варіації технічного пакету зі створення одягу як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `COLORS`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного кольору.

Поле `maincolormark` реалізує основний колір цієї варіації технічного пакету зі створення одягу як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `COLORS`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного кольору.

Поле `usagedescription` реалізує текстовий опис особливостей цього технічного пакету зі створення одягу як об'єкт класу `models.CharField`.

Модель `Fashioncollectiontechpackimage` реалізує клас зображень креслень технічних пакетів зі створення одягу з такими полями.

Поле `fashioncollectiontechpackcolor` реалізує визначення кольорової варіації технічного пакету зі створення одягу, що представлена на цьому кресленні, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpackcolor`.

Поле `techpackimage` реалізує самі дані зображення креслення технічного пакету зі створення одягу як об'єкт класу `models.ImageField`.

Поле `imagedescription` реалізує текстовий опис зображення креслення технічного пакету зі створення одягу як об'єкт класу `models.CharField`.

Поле `imagetechpackposition` реалізує ракурс одягу, який показаний на цьому кресленні технічного пакету, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `POSITION_TYPES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідного ракурсу одягу.

Модель `Fashioncollectiontechpackdetail` реалізує клас додаткового опису особливостей, співставлених з технічним пакетом, з такими полями.

Поле `fashioncollectiontechpack` реалізує визначення технічного пакету зі створення одягу, до якого відносяться ці особливості, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpack`.

Поле `garmentpart` реалізує частину одягу, до якої відноситься ця особливість, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `GARMENT_PARTS`, реалізованого для створення

відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідної частини одягу.

Поле `itemdescription` реалізує додатковий опис особливостей, співставлених з технічним пакетом, як об'єкт класу `models.TextField`.

Модель `Fashionmaterial` реалізує клас матеріалів тканин, які можуть використовуватися для виробництва одягу, з полем `fabricname`, яке реалізує назву матеріалу тканини, що може використовуватися для виробництва одягу, як об'єкт класу `models.CharField`.

Модель `Fashioncollectiontechpackfabric` реалізує клас складу тканини, яку необхідно використовувати для виробництва одягу за технічним пакетом, з такими полями.

Поле `fashioncollectiontechpack` реалізує визначення технічного пакету, за яким для виробництва одягу використовується ця тканина, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpack`.

Поле `fashionmaterial` реалізує визначення матеріалу зі складу тканини, яка використовується для виробництва одягу за цим технічним пакетом, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmaterial`.

Поле `fabricproportion` реалізує відсоток вмісту цього матеріалу в тканині як об'єкт класу `models.DecimalField`.

Модель `Fashioncollectiontechpackfabricpart` реалізує клас співвідношення складу тканини з частиною одягу, для виготовлення якої вона використовується, з такими полями.

Поле `fashioncollectiontechpackfabric` реалізує визначення складу тканини, яка використовується для виготовлення цієї частини одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpackfabric`.

Поле `garmentpart` реалізує частину одягу, для виготовлення якої використовується ця тканина, як об'єкт класу `models.CharField`.

Модель `Fashioncollectiontechpacksaving` реалізує клас вкладок, збережених за технічними пакетами одягу, з такими полями.

Поле `fashioncollectiontechpack` реалізує визначення технічного пакету для виготовлення одягу, який збережено у вкладку користувачем, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollectiontechpack`.

Поле `designeruser` реалізує визначення користувача, який зберіг собі у вкладку цей технічний пакет з виготовлення одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `User`.

Модель `Fashioncollectionsaving` реалізує клас колекцій одягу дизайнерів-модельєрів з такими полями.

Поле `designeruser` реалізує визначення користувача, який зберіг собі у вкладку цю колекцію одягу, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `User`.

Поле `fashioncollection` реалізує визначення колекції одягу, яку у вкладку зберіг собі цей користувач, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncollection`.

Модель `Fashionmaster` реалізує клас майстрів дизайну одягу, які є користувачами програми, з такими полями.

Поле `masteruser` реалізує визначення користувацького акаунту, який використовується цим майстром з дизайну одягу, як об'єкт класу `models.OneToOneField` зі зв'язком з моделлю `User`.

Поле `fmartname` реалізує творчий псевдонім цього майстра з дизайну одягу як об'єкт класу `models.CharField`.

Поле `fmwebhub` реалізує вебсайт, де представлена інформація про цього майстра з дизайну одягу, як об'єкт класу `models.URLField`.

Поле `fmadditionalurl` реалізує додаткову вебадресу, де представлена інформація про цього майстра з дизайну одягу, як об'єкт класу `models.URLField`.

Поле `fashionachievements` реалізує досягнення, представлені текстом, цього майстра з дизайну одягу, як об'єкт класу `models.TextField`.

Поле `fmdesignbio` реалізує біографію цього майстра з дизайну одягу як об'єкт класу `models.TextField`.

Поле `fmbasecountry` реалізує основну країну, в якій знаходиться цей майстер з виробництва одягу, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `COUNTRIES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням відповідної країни.

Поле `fmexpertlevel` реалізує галузі дизайну, виробництва одягу, в яких є експертом цей майстер, як об'єкт класу `models.CharField`.

Поле `fmtextonstyle` реалізує опис особливостей стилю цього майстра з виробництва одягу як об'єкт класу `models.CharField`.

Поле `fmtextoncolors` реалізує опис особливостей застосування кольорів у одязі цим майстром як об'єкт класу `models.CharField`.

Поле `fmtextonpatterns` реалізує опис особливостей застосування візерунків цим майстром з виробництва одягу як об'єкт класу `models.CharField`.

Модель `Fashionmasterlevel` реалізує клас оцінювання рівня дизайнера-модельєра майстром з такими полями.

Поле `fashionmaster` реалізує визначення майстра, який оцінив дизайнера-модельєра, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmaster`.

Поле `fashiondesigner` реалізує визначення дизайнера-модельєра, здібності якого оцінив майстер, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `fashionlevelpoint` реалізує числове оцінювання рівня дизайнера-модельєра майстром як об'єкт класу `models.IntegerField`.

Поле `fashionleveltext` реалізує текстове оцінювання рівня дизайнера-модельєра майстром як об'єкт класу `models.CharField`.

Поле `actualdt` реалізує час і дату оцінювання рівня дизайнера-модельєра майстром як об'єкт класу `models.DateTimeField`.

Модель `Fashionmasterimage` реалізує клас зображень майстрів з дизайну одягу з такими полями.

Поле `fashionmaster` реалізує визначення майстра з дизайну одягу, який показаний на фотографії, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmaster`.

Поле `fmpersonimage` реалізує дані фотографії майстра з дизайну одягу як об'єкт класу `models.ImageField`.

Поле `fmcoverness` реалізує використання фотографії в якості основної для майстра з дизайну одягу як об'єкт класу `models.BooleanField`.

Модель `Fashionmastercourse` реалізує клас типових курсів для навчання дизайнерів-модельєрів з такими полями.

Поле `fashionmaster` реалізує визначення майстра з дизайну одягу, який проводить цей типовий курс для навчання дизайнерів-модельєрів, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmaster`.

Поле `fashioncoursefulltitle` реалізує повну назву типового курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `fashioncourseshorttitle` реалізує коротку назву типового курсу для навчання дизайнерів-модельєрів для використання на сторінках у позиціях, де є обмежена кількість символів, як об'єкт класу `models.CharField`.

Поле `fcavailableplaces` реалізує максимальну кількість місць, доступну для навчання за цим курсом для дизайнерів-модельєрів одночасно, як об'єкт класу `models.IntegerField`.

Поле `fccoverimage` реалізує зображення, яке використовується як обкладинка типового курсу для навчання дизайнерів-модельєрів, як об'єкт класу `models.ImageField`.

Поле `videoforfcpresentation` реалізує відео для презентації типового курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.URLField`.

Поле `purposedesignerlevel` реалізує рівень мистецтва, на заволодіння яким направлено цей типовий курс для навчання дизайнерів-модельєрів, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `LEVELS`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням рівня мистецтва.

Поле `fashioncoursetype` реалізує тип курсів для навчання дизайнерів-модельєрів, до якого цей належить, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashioncoursetype`.

Поле `fashioncourselevel` реалізує оцінку, яку було виставлено курсу дизайнерів-модельєрів за результатами навчання, як об'єкт класу `models.DecimalField`.

Моделю `Fashioncoursetype` реалізує клас типів курсів для навчання дизайнерів-модельєрів з полем `fashionprocessname`, яке реалізує назву типу курсів для навчання дизайнерів-модельєрів як об'єкт класу `CharField`.

Моделю `Fashionmastercoursemeeting` реалізує клас публікацій про проведення навчання за типовими курсами для навчання дизайнерів-модельєрів з такими полями.

Поле `fashionmastercourse` реалізує типовий курс для навчання дизайнерів-модельєрів, за яким проводиться це навчання, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmastercourse`.

Поле `coursemeetingstart` реалізує час і дату початку навчання за цим курсом для дизайнерів-модельєрів як об'єкт класу `models.DateTimeField`.

Поле `coursemeetingend` реалізує час і дату закінчення навчання за цим курсом для дизайнерів-модельєрів як об'єкт класу `models.DateTimeField`.

Поле `meetavailableplaces` реалізує кількість місць, яка доступна для цього навчання за цим курсом для дизайнерів-модельєрів, як об'єкт класу `models.IntegerField`.

Поле `meetoccupiedplaces` реалізує кількість місць, яка зайнята за цим навчанням за цим курсом для дизайнерів-модельєрів, як об'єкт класу `models.IntegerField`.

Поле `meetcoverimage` реалізує зображення, яке використовується для цього навчання за цим курсом для дизайнерів-модельєрів, як об'єкт класу `models.ImageField`.

Поле `videoformmeetpresentation` реалізує відео, яке використовується як презентація для цього навчання за цим курсом для дизайнерів-модельєрів, як об'єкт класу `models.URLField`.

Поле `meetingtype` реалізує тип зустрічі (онлайн чи офлайн) для цього навчання за цим курсом для дизайнерів-модельєрів як об'єкт класу `models.BooleanField`.

Поле `hiddenmeetinginfo` реалізує приховану інформацію для цього навчання за цим курсом для дизайнерів-модельєрів як об'єкт класу `models.TextField`.

Поле `coursevenue` реалізує місце проведення зустрічі для навчання за курсом як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Coursevenue`.

Модель `Coursevenue` реалізує клас можливих місць проведення навчання дизайнерів-модельєрів з такими полями.

Поле `venueename` реалізує назву можливого місця проведення навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `venuephone` реалізує номер телефону можливого місця проведення навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `venueaddress` реалізує поштову адресу можливого місця проведення навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `venueurl` реалізує веб адресу можливого місця проведення навчання дизайнерів-модельєрів як об'єкт класу `models.URLField`.

Поле `venuecity` реалізує місто, в якому знаходиться це можливе місце проведення навчання дизайнерів-модельєрів, як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `CITIES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням міста.

Модель `Fashionmastercoursestuff` реалізує клас матеріалів за курсами для навчання дизайнерів-модельєрів з такими полями.

Поле `fashionmastercourse` реалізує типовий курс для навчання дизайнерів-модельєрів, за яким використовується цей матеріал як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmastercourse`.

Поле `fashionstuffurl` реалізує вебадресу матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.URLField`.

Поле `fashionstuffdata` реалізує дані матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.TextField`.

Поле `fashionstuffshorttitle` реалізує повну назву матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `fashionstufffulltitle` реалізує коротку назву матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`.

Поле `activeness` реалізує визначення, чи цей матеріал ще використовується при вивченні цього курсу для навчання дизайнерів-модельєрів, чи вже ні, як об'єкт класу `models.BooleanField`.

Поле `accesstype` реалізує тип доступу до матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.BooleanField`.

Поле `fashionstufftype` реалізує тип матеріалу курсу для навчання дизайнерів-модельєрів як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `STUFF_TYPE`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням типу матеріалу курсу.

Поле `fashionstuffindex` реалізує порядковий номер даного матеріалу курсу для навчання дизайнерів-модельєрів серед всіх матеріалів за цим курсом як об'єкт класу як об'єкт класу `models.IntegerField`.

Поле `activationstartdt` реалізує час і дату, з яких цей матеріал курсу для навчання дизайнерів-модельєрів доступний, як об'єкт класу `models.DateTimeField`.

Поле `activationenddt` реалізує час і дату, до яких цей матеріал курсу для навчання дизайнерів-модельєрів доступний, як об'єкт класу `models.DateTimeField`.

Поле `fashionmastercoursestuffapplications` реалізує визначення сильних сторін дизайнера-модельєра як об'єкт класу `models.ManyToManyField` через зв'язок з класом `Fashionmastercourseapplication` («багато-до-багатьох»), таким чином реалізуючи таблицю `fashionmastercoursestuffapplication` з БД, описану в розділі 3.

Модель `Fashionmastercourseapplication` реалізує клас заявок на вивчення курсів від дизайнерів-модельєрів (або майстрів) з такими полями.

Поле `fashiondesigner` реалізує дизайнера-модельєра, який створив цю заявку на вивчення курсу або якому адресована заявка від майстра, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashiondesigner`.

Поле `fashionmastercoursemeeting` реалізує конкретне проведення навчання за типовим курсом для навчання дизайнерів-модельєрів, за яким подав цю заявку дизайнер-модельєр або майстер, як об'єкт класу `models.ForeignKey` зі зв'язком з моделлю `Fashionmastercoursemeeting`.

Поле `fashionapplicationdt` реалізує час і дату, коли ця заявка на вивчення курсу від дизайнера-модельєра (або майстра) була подана, як об'єкт класу `models.DateTimeField`.

Поле `initializer` реалізує сторону-ініціатора подання заявки (майстер або дизайнер-модельєр) як об'єкт класу `models.BooleanField`.

Поле `fashionapplicationstatus` реалізує стан розгляду заявка на вивчення курсу від дизайнера-модельєра (або майстра) як об'єкт класу `models.CharField`, вибір значень забезпечуючи на основі словника `APPLICATION_STATUSES`, реалізованого для створення відповідності між одним символом, який шифрує значення, та повним текстовим значенням стану розгляду заявки на вивчення курсу дизайнера-модельєра (або майстра).

Поле `fashiondesignerlevelpoint` реалізує чисельну оцінку, виставлену загалом за вивчення курсу дизайнеру-модельєру, як об'єкт класу `models.IntegerField`.

Поле `fashiondesignerleveltext` реалізує текстову оцінку, виставлену загалом за вивчення курсу дизайнеру-модельєру, як об'єкт класу `models.CharField`.

Поле `fashioncourselevelpoint` реалізує чисельну оцінку, виставлену загалом курсу дизайнером-модельєром, як об'єкт класу `models.IntegerField`.

Поле `fashioncourseleveltext` реалізує текстову оцінку, виставлену загалом курсу дизайнером-модельєром, як об'єкт класу `models.CharField`.

Поле `foundationalscorebefore` реалізує середню оцінку максимум 3 останніх курсів, за навчання за якими отримав оцінку дизайнер-модельєр, як об'єкт класу `models.DecimalField`.

Поле `foundationalscore` реалізує результат у балах дизайнера-модельєра за вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `predictedfoundationalscore` реалізує прогнозований моделлю результат у балах дизайнера-модельєра за вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `digitalcompetencescore` реалізує оцінку цифрових компетентностей дизайнера-модельєра за результатами вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `initiativescore` реалізує оцінку рівня підприємницьких вмінь дизайнера-модельєра за результатами вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `problemsolvingscore` реалізує оцінку здібностей вирішувати комплексні проблеми дизайнера-модельєра за результатами вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `creativityscore` реалізує оцінку здібностей до креативності та інноваційності дизайнера-модельєра за результатами вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `communicationscore` реалізує оцінку комунікаційних здібностей дизайнера-модельєра за результатами вивчення курсу як об'єкт класу `models.DecimalField`.

Поле `engagementscore` реалізує оцінку залученості дизайнера-модельєра до виконання завдань цього курсу як об'єкт класу `models.DecimalField`.

Представлення реалізовані окремими функціями, що забезпечують сценарії роботи, об'єднуючи відповідні сценарії різних користувачів у одну функцію.

Функція `fashiondesigner_view` реалізує представлення відображення інформації стосовно дизайнера-модельєра з параметром `fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashiondesigner_collections_view` реалізує представлення відображення всіх колекцій дизайнера-модельєра з параметром `page`, що задає номер сторінки колекцій, та `fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashiondesigner_levels_view` реалізує представлення відображення всіх оцінювань, виставлених дизайнеру-модельєру з параметром `page`, що задає номер сторінки оцінювань, `course_level`, що задає відображення оцінювань за результатами навчання за курсами або за прямим оцінюванням майстром після перегляду технічних пакетів, `fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashiondesigner_courses_view` реалізує представлення відображення всіх курсів, які відвідує дизайнер-модельєр з параметром `fashioncourse_status`, що задає стан заявок (всі або заданий стан), `designeruser_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashionmaster_view` реалізує представлення відображення інформації стосовно майстра дизайну одягу з параметром `fashionmaster_username`, що задає логін майстра дизайну одягу, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як майстер дизайну одягу.

Функція `fashionmaster_courses_view` реалізує представлення відображення курсів для навчання дизайнерів-модельєрів за майстром з параметром `page`, що задає номер сторінки курсів, `fashionmaster_username`, що задає логін майстра дизайну одягу, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як майстер дизайну одягу.

Функція `fashion_course_stuffs_view` реалізує представлення відображення всіх матеріалів за курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор курсу для навчання дизайнерів-модельєрів, `page`, що задає номер сторінки матеріалів.

Функція `fashioncourse_applications` реалізує представлення відображення поданих заявок на навчання за курсом для дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор курсу для дизайнерів-модельєрів, `page`, що задає номер сторінки заявок, `status`, що задає стан заявок.

Функція `fashioncollectiontechpack_view` реалізує представлення відображення повного вмісту технічного пакету дизайнера-модельєра з параметром `fashioncollectiontechpack_id`, що задає ідентифікатор цього технічного пакету.

Функція `saved_fashioncollections_view` реалізує представлення відображення колекцій одягу зі вкладок.

Функція `connected_fashiondesigner_collections_view` реалізує представлення відображення колекцій дизайнерів-модельєрів з підписок.

Функція `fashiondesigner_change_view` реалізує представлення модифікування загальних даних дизайнера-модельєра з параметром

`fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashiondesignerimage_add_view` реалізує представлення додавання фотографії дизайнера-модельєра з роботою на основі класу відповідної форми як і наступні функції, які такі дії виконують без параметрів.

Функція `fashiondesignerimage_delete_view` реалізує представлення вилучення фотографії дизайнера-модельєра з параметром `fashiondesignerimage_id`, що задає ідентифікатор фотографії.

Функція `fashiondesignersoftwareskill_add_view` реалізує представлення додавання програмних вмінь дизайнера-модельєра.

Функція `fashiondesignersoftwareskill_delete_view` реалізує представлення вилучення програмних вмінь дизайнера-модельєра з параметром `fashiondesignersoftwareskill_id`, що задає ідентифікатор програмного вміння дизайнера-модельєра.

Функція `fashiondesignerskill_add_view` реалізує представлення додавання дизайнерських вмінь дизайнера-модельєра.

Функція `fashiondesignerskill_delete_view` реалізує представлення вилучення дизайнерських вмінь дизайнера-модельєра з параметром `designkill_id`, що задає ідентифікатор дизайнерського вміння, `fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `designexperience_add_view` реалізує представлення додавання досвіду роботи дизайнера-модельєра.

Функція `designexperience_delete_view` реалізує представлення вилучення досвіду роботи дизайнера-модельєра з параметром `designexperience_id`, що задає ідентифікатор досвіду роботи дизайнера-модельєра.

Функція `fashiondesignertypicalstrength_add_view` реалізує представлення додавання сильних сторін дизайнера-модельєра.

Функція `fashiondesignertypicalstrength_delete_view` реалізує представлення видалення сильних сторін дизайнера-модельєра з параметром `typicalfashionstrength_id`, що задає ідентифікатор сильної сторони, `fashiondesigner_username`, що задає логін дизайнера-модельєра, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як дизайнер-модельєр.

Функція `fashiondesignerconnection_add_view` реалізує представлення підписування на колекції робіт дизайнера-модельєра та скасування за його наявності з параметром `fashiondesignerreceiver_username`, що задає логін дизайнера-модельєра.

Функція `fashiondesigner_connections_view` реалізує представлення відображення дизайнерів-модельєрів з підписок.

Функція `fashioncollection_add_view` реалізує представлення створення колекції одягу дизайнера-модельєра.

Функція `fashioncollection_change_view` реалізує представлення редагування колекції одягу дизайнера-модельєра з параметром `fashioncollection_id`, що задає ідентифікатор колекції.

Функція `fashioncollection_delete_view` реалізує представлення видалення колекції одягу дизайнера-модельєра з параметром `fashioncollection_id`, що задає ідентифікатор колекції.

Функція `fashioncollectiontechpack_add_view` реалізує представлення створення технічного пакету за колекцією одягу.

Функція `fashioncollectiontechpack_change_view` реалізує представлення редагування технічного пакету за колекцією одягу з параметром `fashioncollectiontechpack_id`, що задає ідентифікатор колекції одягу.

Функція `fashioncollectiontechpack_delete_view` реалізує представлення видалення технічного пакету за колекцією одягу з параметром `fashioncollectiontechpack_id`, що задає ідентифікатор колекції одягу.

Функція `fashioncollection_techpacks_view` реалізує представлення відображення всіх технічних пакетів за колекцією одягу з параметром `fashioncollection_id`, що задає ідентифікатор колекції.

Функція `fashioncollectiontechpack_tosave_view` реалізує представлення збереження технічного пакету за колекцією одягу у вкладку та скасування наявності з параметром `fashioncollectiontechpack_id`, що задає ідентифікатор технічного пакету.

Функція `fashioncollectiontechpacksavings_view` реалізує представлення відображення технічних пакетів зі вкладок.

Функція `fashioncollection_tosave_view` реалізує представлення збереження колекції одягу у вкладку та скасування з параметром `fashioncollection_id`, що задає ідентифікатор колекції.

Функція `fashionmaster_change_view` реалізує представлення модифікування загальних даних майстра дизайну одягу з параметром `fashionmaster_username`, що задає логін майстра дизайну одягу, а якщо його значення не задано, то логін користувача, який повинен бути в цей момент авторизований як майстер дизайну одягу.

Функція `fashionmastercourse_add_view` реалізує представлення додавання типового курсу для навчання дизайнерів-модельєрів.

Функція `fashionmastercourse_change_view` реалізує представлення модифікування типового курсу для навчання дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор типового курсу для навчання дизайнерів-модельєрів.

Функція `fashionmastercourse_delete_view` реалізує представлення вилучення типового курсу для навчання дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор типового курсу для навчання дизайнерів-модельєрів.

Функція `fashionmastercourse_view` реалізує представлення відображення інформації стосовно курсу для навчання дизайнерів-модельєрів

з параметром `fashionmastercourse_id`, що задає ідентифікатор курсу для навчання дизайнерів-модельєрів.

Функція `fashionmastercourse_search_view` реалізує представлення пошуку серед курсів для навчання дизайнерів-модельєрів з параметром `page`, що задає номер сторінки результатів пошуку, а самі параметри передаються через структуру GET-запиту.

Функція `fashionmaster_search_view` реалізує представлення пошуку серед майстрів дизайну одягу з параметром `page`, що задає номер сторінки результатів пошуку, а самі параметри передаються через структуру GET-запиту.

Функція `techpack_search_view` реалізує представлення пошуку серед технічних пакетів зі створення одягу з параметром `page`, що задає номер сторінки результатів пошуку, а самі параметри передаються через структуру GET-запиту.

Функція `fashionmastercourses_best_view` реалізує представлення відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом з параметром `fashioncoursetype_id`, що задає ідентифікатор типу курсів (процесу дизайну).

Функція `fashionmastercoursemeeting_add_view` реалізує представлення створення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор типового курсу.

Функція `fashionmastercoursemeeting_change_view` реалізує представлення модифікування публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercoursemeeting_id`, що задає ідентифікатор навчання за типовим курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursemeeting_delete_view` реалізує представлення вилучення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів з параметром

`fashionmastercoursemeeting_id`, що задає ідентифікатор навчання за типовим курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursemeeting_view` реалізує представлення відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercoursemeeting_id`, що задає ідентифікатор навчання за типовим курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursemeetings_search_view` реалізує представлення пошуку серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів з параметром `page`, що задає номер сторінки результатів пошуку, а самі параметри передаються через структуру GET-запиту.

Функція `fashionmastercoursestuff_view` реалізує представлення відображення матеріалу за курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercoursestuff_id`, що задає ідентифікатор матеріалу за курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursestuff_add_view` реалізує представлення публікації матеріалу за курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercourse_id`, що задає ідентифікатор курсу для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursestuff_edit_view` реалізує представлення модифікування матеріалу за курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercoursestuff_id`, що задає ідентифікатор матеріалу за курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercoursestuff_delete_view` реалізує представлення вилучення матеріалу за курсом для навчання дизайнерів-модельєрів з параметром `fashionmastercoursestuff_id`, що задає ідентифікатор матеріалу за курсом для навчання дизайнерів-модельєрів.

Функція `fashionmastercourseapplication_add_view` реалізує представлення подання заявки на навчання за курсом для дизайнерів-

модельєрів з параметром `fashionmastercoursemeeting_id`, що задає ідентифікатор зустрічі для проведення навчання.

Функція `fashionmastercourseapplication_cancel_view` реалізує представлення скасування заявки на навчання за курсом для дизайнерів-модельєрів з параметром `fashionmastercourseapplication_id`, що задає ідентифікатор заявки на навчання за курсом для дизайнерів-модельєрів.

Функція `fashionmastercourseapplication_approve_view` реалізує представлення підтвердження заявки на навчання за курсом для дизайнерів-модельєрів з параметром `fashionmastercourseapplication_id`, що задає ідентифікатор заявки на навчання за курсом для дизайнерів-модельєрів.

Функція `fashionmastercourseapplication_level_add_view` реалізує представлення оцінювання дизайнером-модельєром курсу з параметром `fashionmastercourse_id`, що задає номер курсу.

Функція `fashionmastercourseapplication_level_delete_view` реалізує представлення вилучення оцінювання дизайнером-модельєром курсу з параметром `fashionmastercourse_id`, що задає номер курсу.

Функція `fashionmasterlevel_add_view` реалізує представлення оцінювання рівня дизайнера-модельєра майстром з параметром `fashiondesigner_username`, що задає логін дизайнера-модельєра.

Функція `fashionmasterlevel_delete_view` реалізує представлення скасування оцінювання рівня дизайнера-модельєра майстром з параметром `fashionmasterlevel_id`, що задає ідентифікатор оцінювання рівня дизайнера-модельєра майстром.

Функція `master_fashionmastercourseapplication_level_add_view` реалізує представлення оцінювання результатів навчання дизайнера-модельєра за курсом з параметром `fashionmastercourseapplication_id`, що задає номер заявки на навчання.

Функція `master_fashionmastercourseapplication_level_delete_view` реалізує представлення вилучення оцінювання результатів навчання

дизайнера-модельєра за курсом з параметром `fashionmastercourseapplication_id`, що задає номер заявки на навчання.

Функція `fashion_model_create_view` реалізує представлення створення моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання. Для реалізації представлення використано клас `RandomForestRegressor` з підпаketу `ensemble` в складі пакету `scikit-learn` [23].

Функція `fashion_model_train_view` реалізує представлення навчання моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання. Зі збереженого об'єкту класу `RandomForestRegressor` викликається метод `fit` для навчання моделі. Прогнозування відбувається в представленнях подання заявок на основі методу `predict`.

Функція `fashionmastercourse_levels_view` реалізує представлення відображення всіх оцінювань курсу дизайнерами-модельєрами з параметром `fashionmastercourse_id`, що задає ідентифікатор курсу.

4.3 Висновки за розділом 4

Реалізовано сукупність класів і функцій для забезпечення роботи вебзастосунку персоналізованого навчання дизайнерів-модельєрів, які описано в розділі. Приведено схему функціонування створеної програми.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Опис програми

ПЗ персоналізованого навчання дизайнерів-модельєрів призначене для його використання:

- дизайнерами-модельєрами для вивчення курсів, які надаються майстрами, представлення себе у вигляді колекцій одягу за технічними пакетами, ознайомлення з ідеями інших дизайнерів-модельєрів;
- майстрами з дизайну одягу для навчання дизайнерів-модельєрів за своїми курсами, пошуку талантів серед них.

5.2 Експлуатація програми

Експлуатація ПЗ персоналізованого навчання дизайнерів-модельєрів може виконуватись:

- неавторизованим користувачем;
- дизайнером-модельєром;
- майстром з дизайну одягу;
- адміністратором.

Від конкретної ролі залежить те, який набір розділів основного меню доступний користувачу, та те, як виглядають певні сценарії в інтерфейсі відповідного користувача.

До прикладу можна розглянути відображення курсів для навчання дизайнерів-модельєрів за майстром. У випадку, якщо сам майстер з дизайну одягу працює зі своєю такою сторінкою (рис. 5.1), то йому доступні розділи меню «Профіль», «Курси», «Пошук», «Вкладки» та «Вийти». Останній елемент є посиланням, яке доступне всім користувачам після авторизації. Майстру за власним набором курсів доступна можливість створити новий курс

за допомогою відповідного посилання, посилання на модифікування та вилучення курсів з переліку.

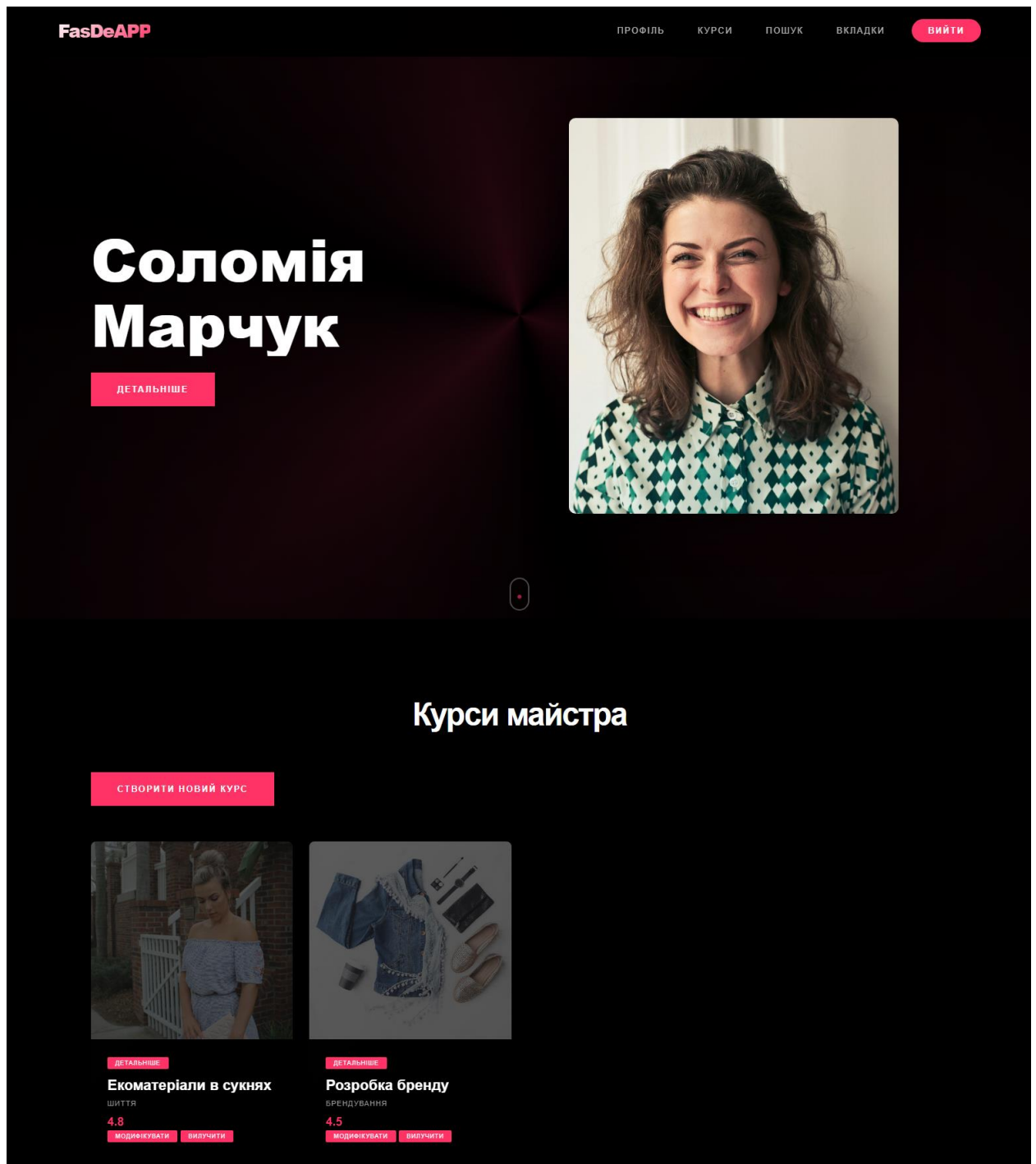


Рисунок 5.1 – Відображення курсів для навчання дизайнерів-модельєрів за майстром (при відкритті майстром-власником)

Якщо таку сторінку відкриває не власник, а просто інший майстер з дизайну одягу, то посилання для внесення змін йому вже не доступні (рис. 5.2).

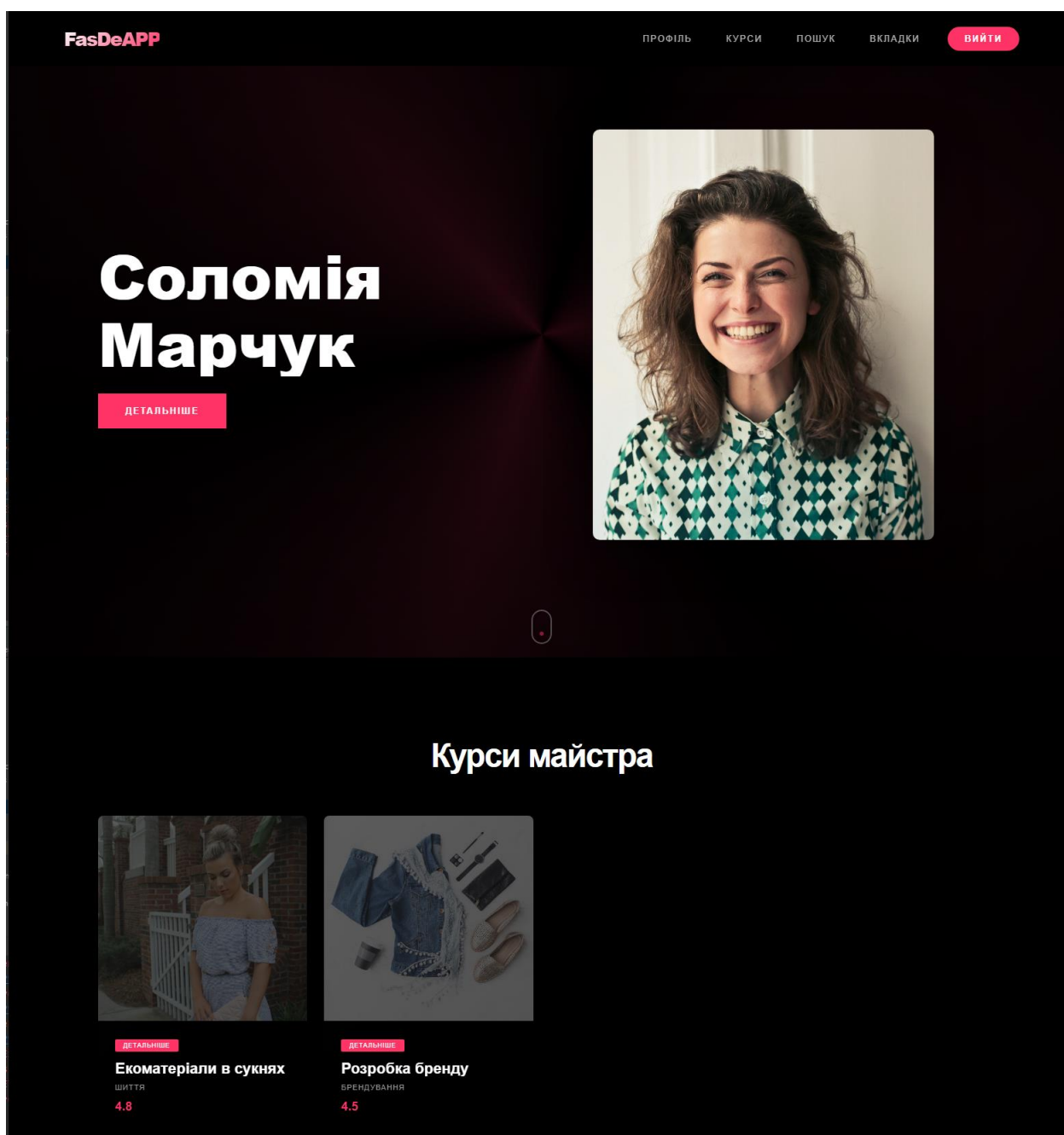


Рисунок 5.2 – Відображення курсів для навчання дизайнерів-модельєрів за майстром (при відкритті майстром, який не є власником)

Якщо сторінку відображення курсів для навчання дизайнерів-модельєрів за майстром відкриває адміністратор, то він не може створити новий курс, але йому доступна можливість модифікувати і вилучати курси майстра (рис. 5.3).

За кожним курсом демонструється його основне ображення, назва, тип процесів, навчання за якими відбувається, середня оцінка. Можна також перейти до сторінки самого курсу з навчання дизайнерів-модельєрів.

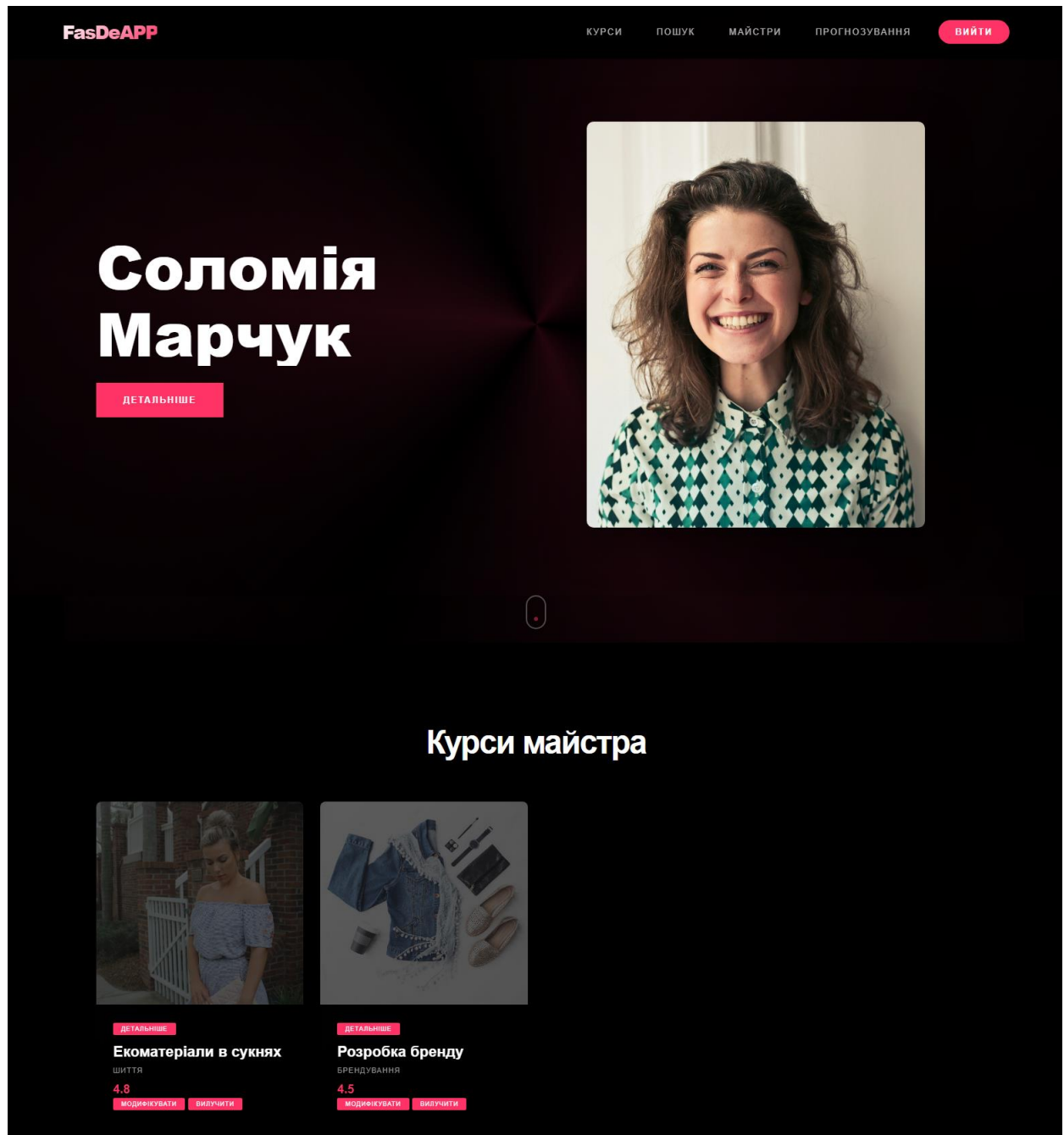


Рисунок 5.3 – Відображення курсів для навчання дизайнерів-модельєрів за майстром для адміністратора

Якщо сторінку відображення курсів для навчання дизайнерів-модельєрів за майстром відкриває дизайнер-модельєр (рис. 5.4), то йому

доступні засоби майстра, який не є власником. При цьому він має відкрити сторінку курсу, щоб з неї обрати потрібне конкретне навчання і подати заявку на таке навчання.

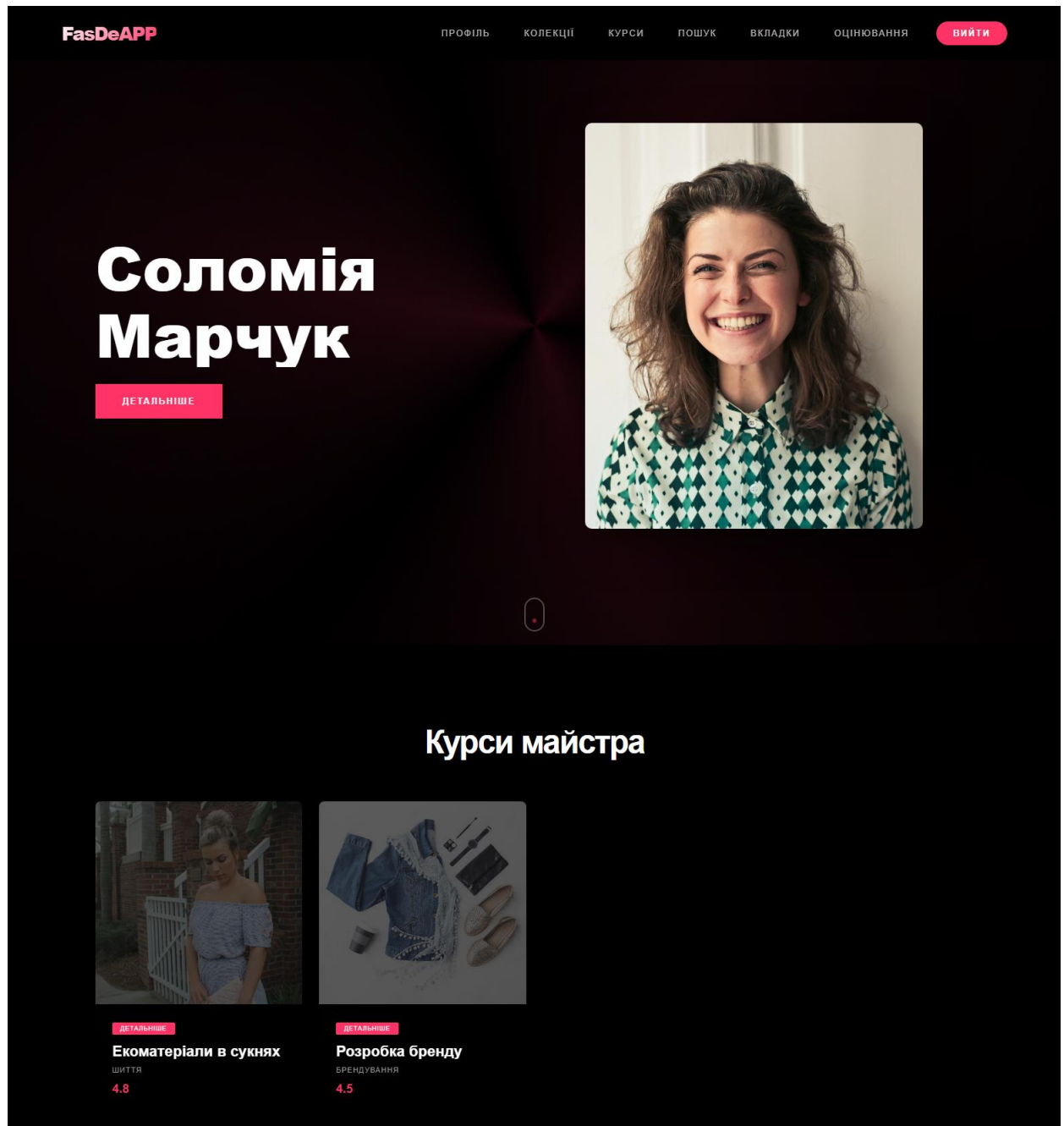


Рисунок 5.4 – Відображення курсів для навчання дизайнерів-модельєрів за майстром для дизайнера-модельєра

Дизайнеру-модельєру доступні розділи меню «Профіль», «Колекції», «Курси», «Пошук», «Вкладки», «Оцінювання».

Якщо користувач не авторизувався (рис. 5.5), то йому доступні тільки розділи меню «Курси», звідки доступний рейтинг найкращих курсів, «Пошук» та посилання «Увійти» для авторизації та перемикання на реєстрацію.

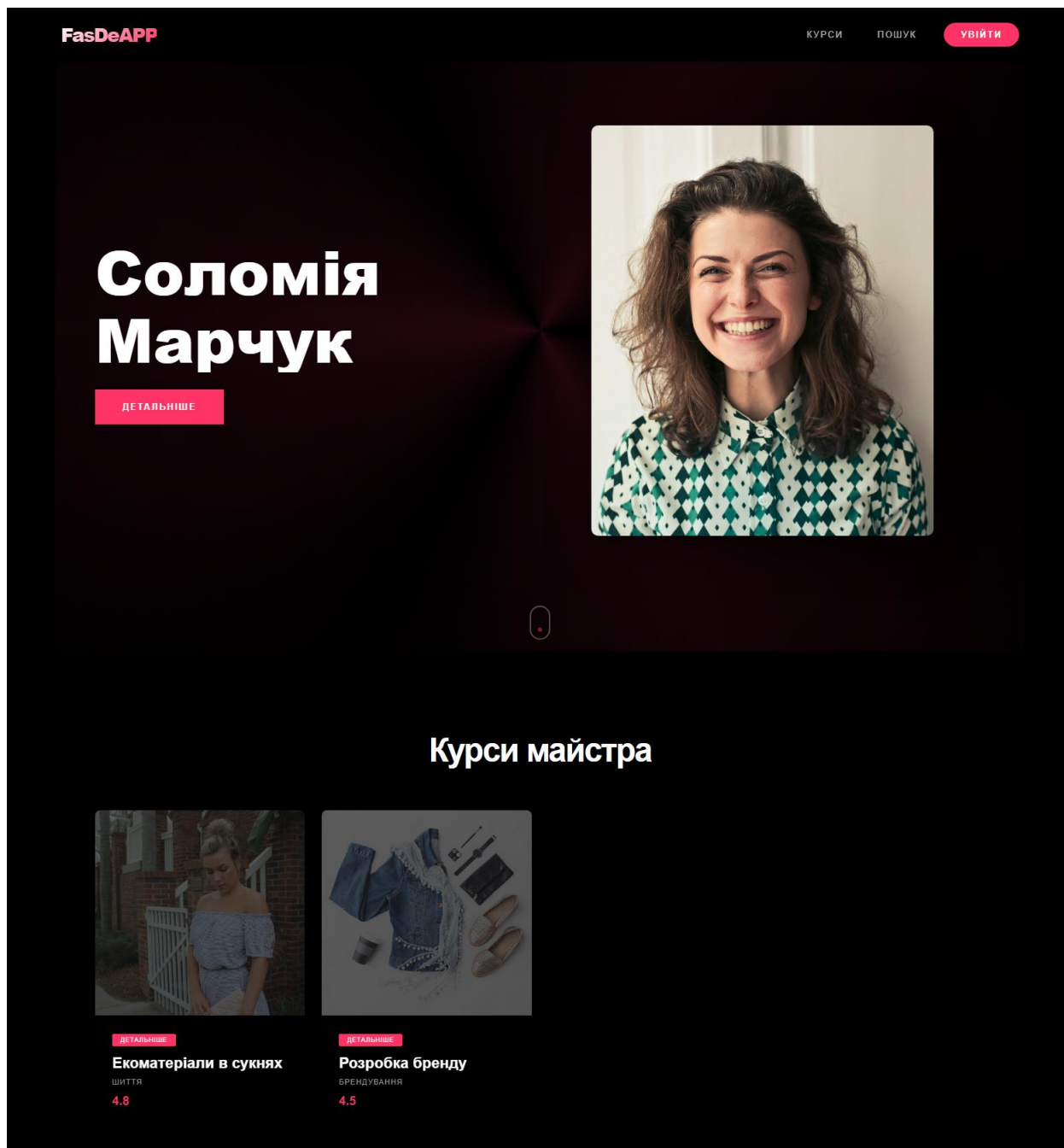


Рисунок 5.5 – Відображення курсів для навчання дизайнерів-модельєрів за майстром для користувача без авторизації

Розділ меню «Пошук» дозволяє виконувати пошук типових курсів для навчання дизайнерів-модельєрів, майстрів, технічних пакетів, конкретних

варіантів навчання (зустрічей за курсами), перемикаючись за ними за допомогою відповідних посилань на сторінці, задаючи далі значення критеріїв та виконуючи пошук.

Усім користувачам також доступний розділ меню «Курси», де доступний рейтинг, але за певними ролями доступні за допомогою відповідних посилань засоби перемикання на інші варіанти роботи з переліками курсів – власними для майстрів з дизайну одягу або ті, за якими є учасниками дизайнери-модельєри.

У випадку дизайнера-модельєра йому також доступний розділ меню «Оцінювання» з оцінками, які йому виставили майстри з дизайну одягу окремо або за результатами навчання за курсами. Дизайнеру-модельєру доступний також розділ з його власними колекціями.

Розділ «Вкладки» зі збереженими колекціями та технічними пакетами з переключенням за допомогою відповідних посилань доступний для дизайнерів-модельєрів та майстрів з дизайну одягу.

Для адміністратора доступний розділ меню «Майстри», звідки він має перелік майстрів з дизайну одягу та можливість зареєструвати нового майстра з дизайну одягу.

При відкритті сторінки дизайнера-модельєра неавторизованому користувачу (рис. 5.6) доступна основна інформація про дизайнера-модельєра, включаючи ім'я, оцінки за різними критеріями, дата народження (та відповідні контактні дані, включаючи посилання, за наявності), сильні сторони, рівень освіти, деталізація освіти, програмні вміння. Також нижче на сторінці продемонстровано перелік наявних колекцій дизайнера-модельєра. При цьому подивитись колекцію користувач може тільки тоді, коли він є майстром, адміністратором, самим дизайнером-модельєром-власником або коли дизайнер-модельєр відкрив публічний доступ за колекцією.

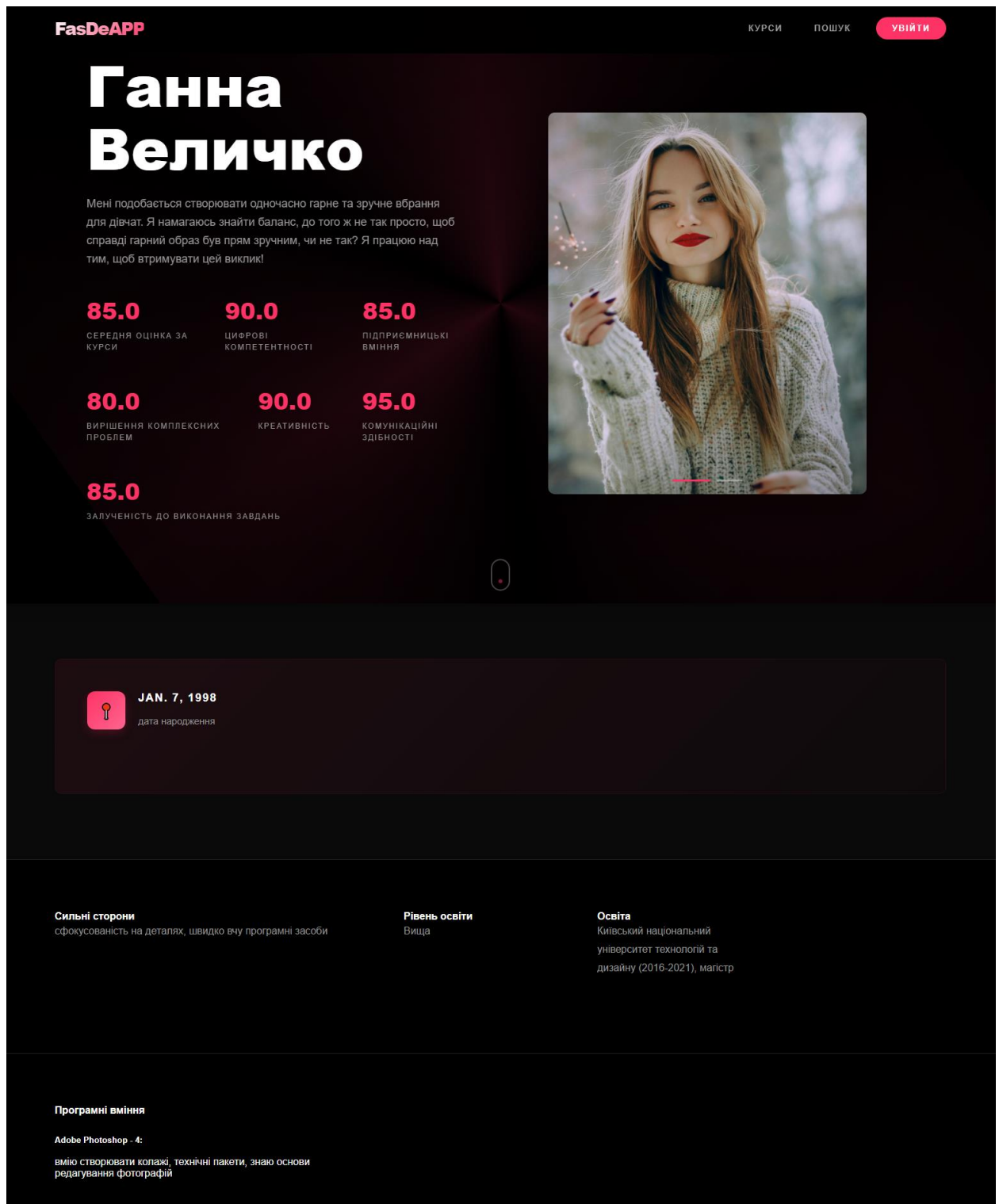


Рисунок 5.6 – Відображення інформації стосовно дизайнера-модельєра для користувача без авторизації

Ця ж сторінка для іншого дизайнера-модельєра (рис. 5.7) дає можливість підписатись на дизайнера-модельєра. Її варіація для того ж самого

дизайнера-модельєра та для адміністратора дозволяє перейти до модифікування даних за допомогою посилання «Модифікувати» (рис. 5.8).

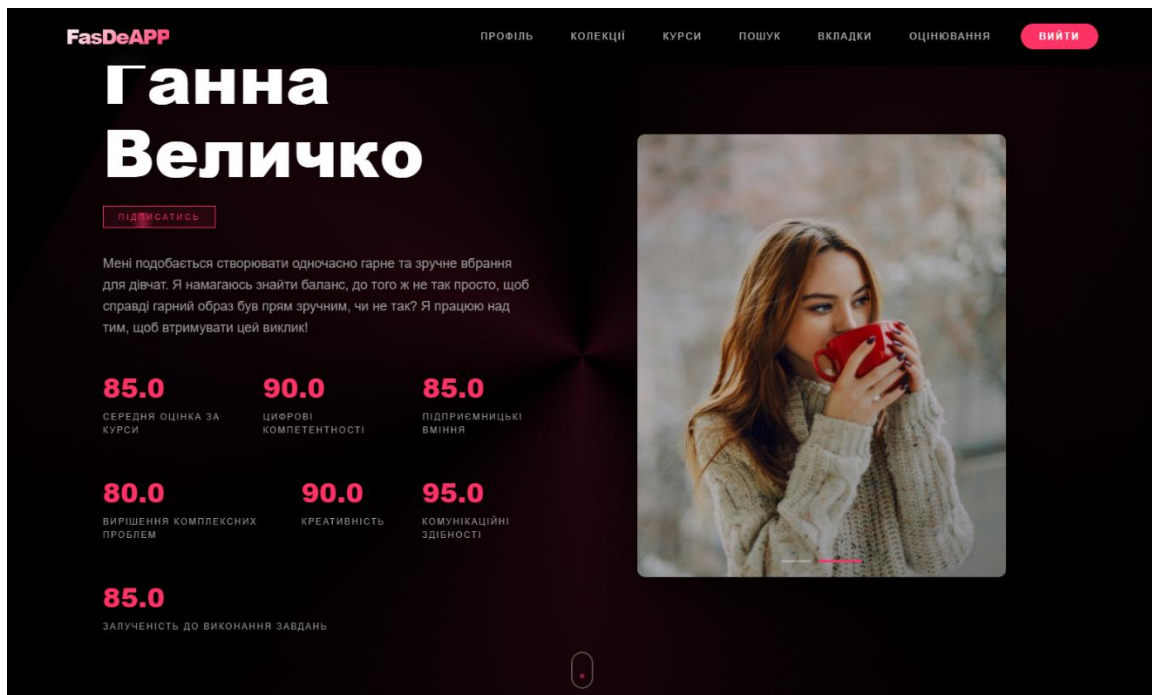


Рисунок 5.7 – Відображення інформації стосовно дизайнера-модельєра для іншого дизайнера-модельєра

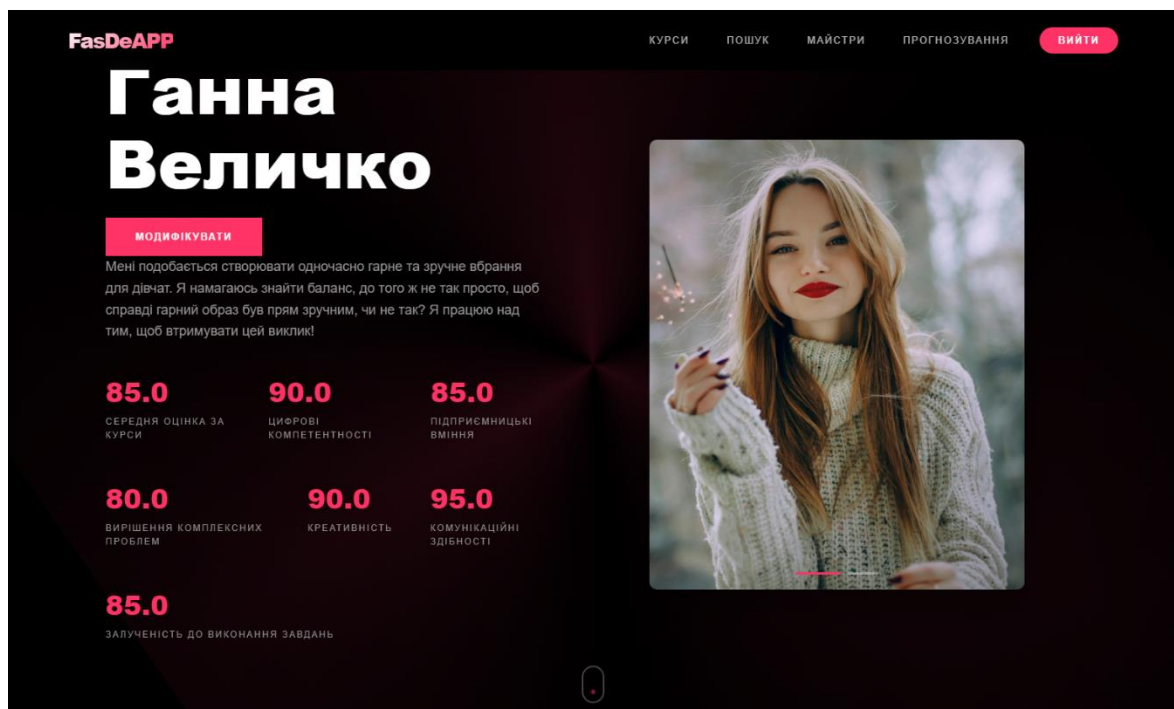


Рисунок 5.8 – Відображення інформації стосовно дизайнера-модельєра для адміністратора

Майстер з дизайну одягу може запросити зі сторінки профілю дизайнера-модельєра на один зі своїх курсів (рис. 5.9).

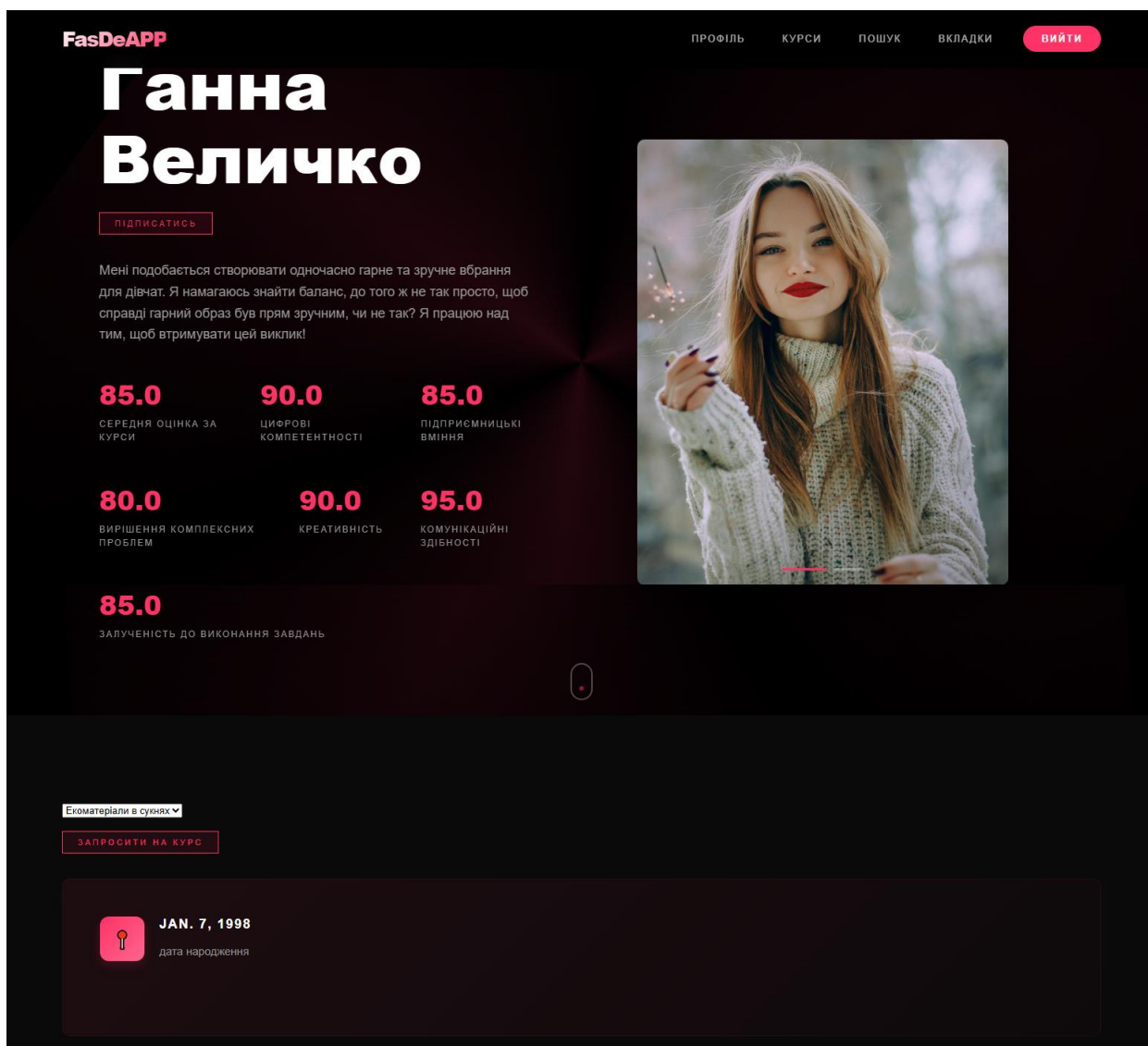


Рисунок 5.9 – Відображення інформації стосовно дизайнера-модельєра для майстра

Майстер з дизайну одягу за своїм курсом може переглянути подані заявки, керуючи ними через підтвердження або скасування заявок (рис. 5.10). При цьому майстер спочатку може вибрати стан заявок (всі, підтверджені, скасовані, подані).

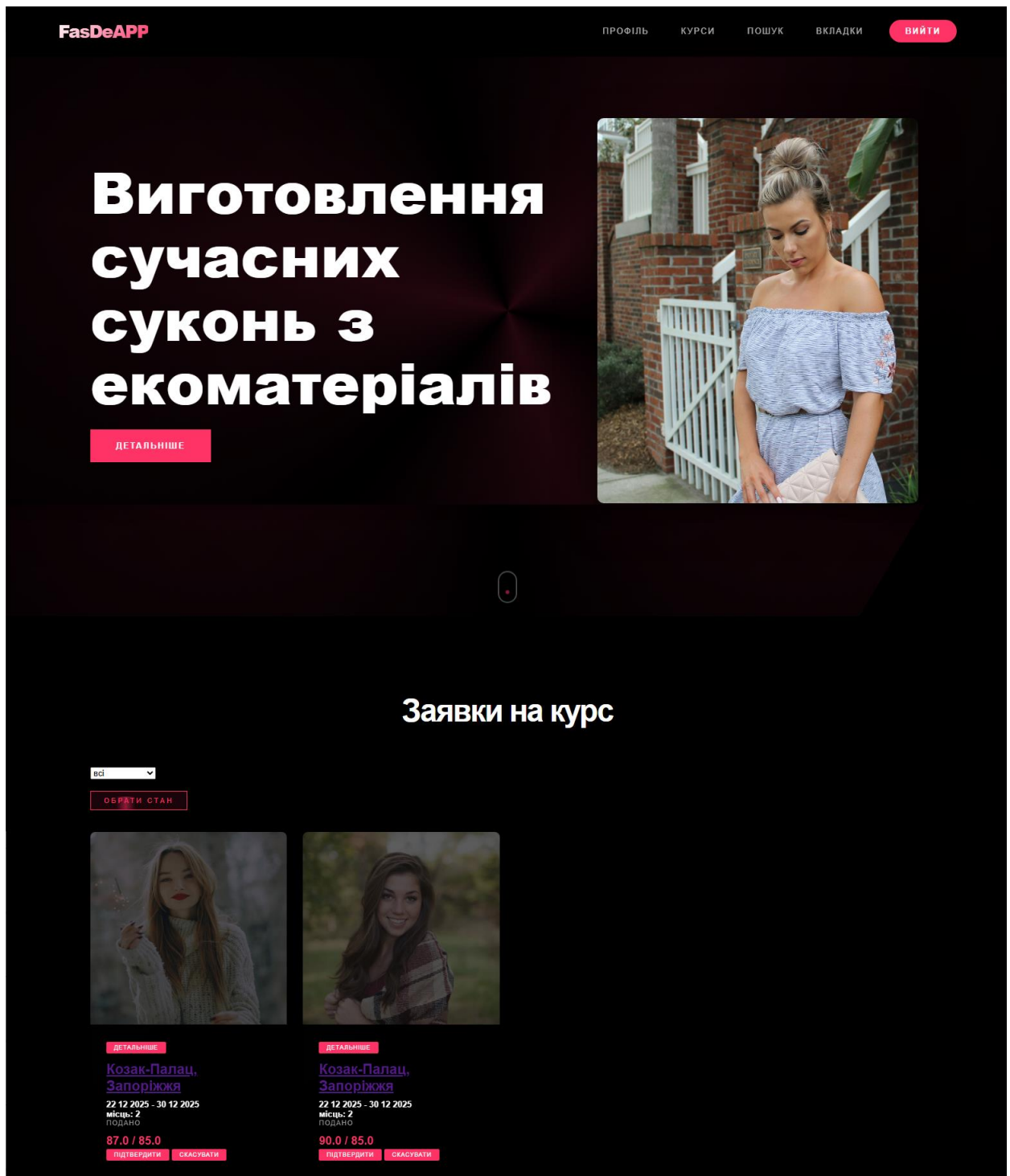


Рисунок 5.10 – Відображення поданих заявок на навчання за курсом для навчання дизайнерів-модельєрів

Сторінка відображення повного вмісту технічного пакету дизайнера-модельєра (рис. 5.11) містить інформацію про колекцію, опис технічного пакету, різні параметри стосовно нього в лівій частині сторінки (склад тканини для виготовлення різних частин одягу, наприклад). У правій частині сторінки

можна обрати конкретну кольорову варіацію з випадуючого переліку, після чого відобразяться зображення технічного пакету, між якими можна перемикатись.

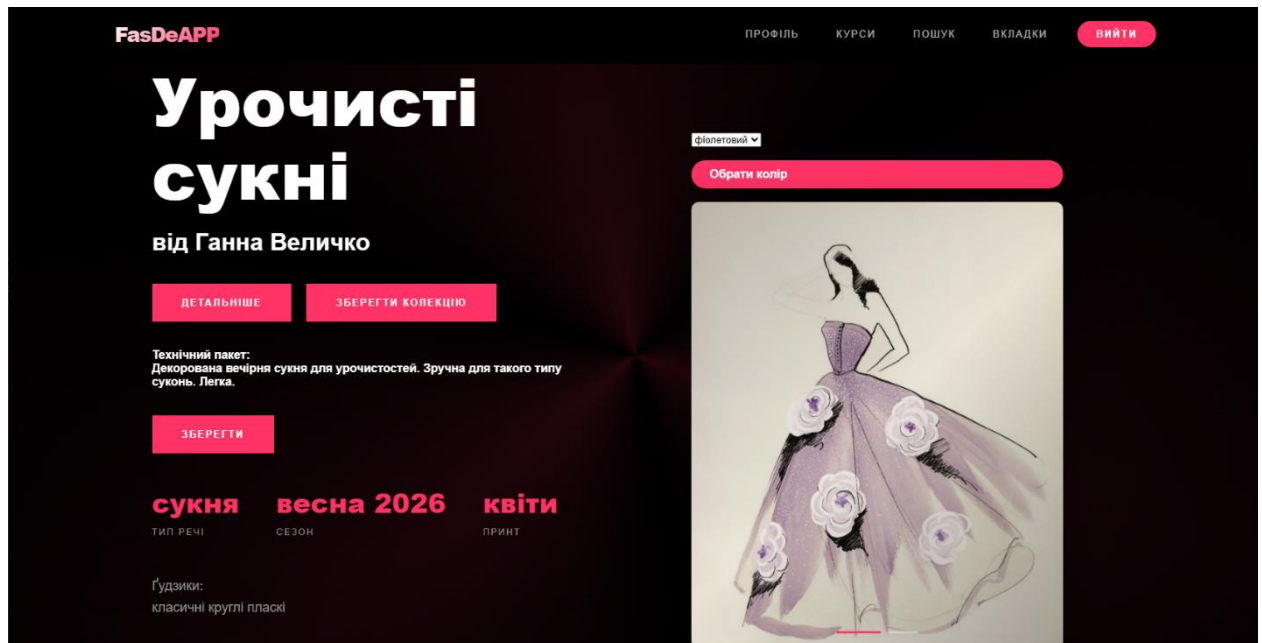


Рисунок 5.11 – Відображення повного вмісту технічного пакету дизайнера-модельєра

Адміністратор має розділ «Прогнозування» в меню, який відкриває відповідну сторінку (рис. 5.12). З цієї сторінки адміністратор може перейти до створення нової моделі за допомогою посилання «Створити модель». За допомогою посилання «Навчити модель» адміністратор може працювати з моделлю, яку було тільки створено. Тоді завантажуються дані з БД, які необхідні для навчання моделі, реалізується саме навчання моделі.

Результати використання моделі демонструються під час подання заявки-пропозиції майстру з дизайну одягу, коли він надає пропозицію дизайнеру-модельєру навчатися за його курсом, а також на сторінці майстра з його власним курсом, де він бачить перелік всіх заявок (рис. 5.10). Ця ж сторінка доступна адміністратору. На цій сторінці за кожним дизайнером-модельєром через дріб демонструється 2 значення. Перше значення – прогнозоване, друге – усереднене за минулими курсами.

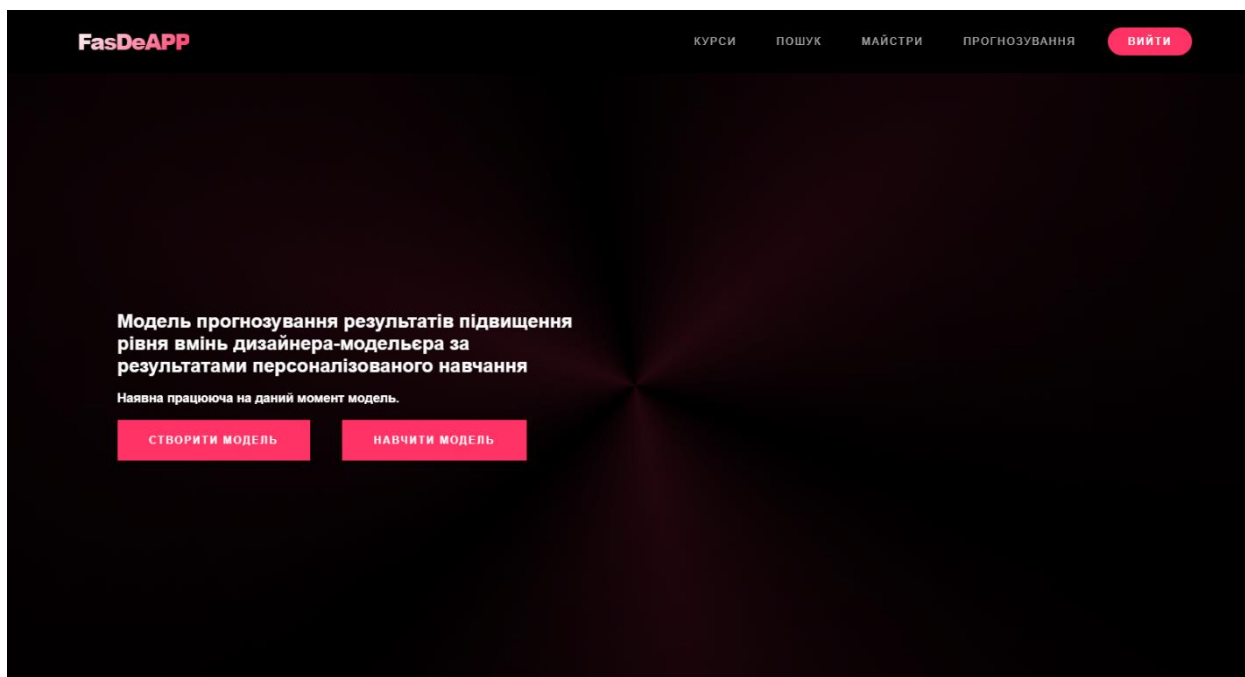


Рисунок 5.12 – Керування моделями прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

5.3 Експериментальне дослідження

Експериментальне дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання виконано за допомогою вибірки даних [24].

Дана вибірка була розподілена на 70 % екземплярів, які випадковим чином виділені та віднесені до навчальної вибірки, та на 30 %, які випадковим чином віднесені до тестувальної вибірки.

Оцінювання результатів прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання виконувалось через показники Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE).

Для порівняння окрім запропонованого методу з використанням методу Random Forest розглянуто метод опорних векторів [25]-[27] та індивідуальні дерева рішень [28]-[30] (табл. 5.1).

Таблиця 5.1 – Результати прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Метод / модель	MAE	RMSE	MSE
Дерево рішень	0,00357646	0,00447873	0,00002006
Метод опорних векторів	0,00572777	0,00678967	0,0000461
Random Forest (запропонований метод)	0,00235198	0,00294896	0,00000869

Запропонований у роботі метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання характеризується покращенням показника MAE у 1,5 рази порівняно з деревами рішень та у 2,4 рази порівняно з методом опорних векторів. У той же час показник RMSE покращено у 2,3 рази порівняно з методом опорних векторів та у 1,5 рази порівняно з деревами рішень. Значення MSE покращено порівняно з деревами рішень у 2,3 рази та у 5,3 рази порівняно з методом опорних векторів.

5.4 Висновки за розділом 5

Описано варіанти роботи з програмою дизайнера-модельєра, майстра з дизайну одягу, адміністратора, користувача без авторизації. Представлено результати проведеного експериментального дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання. Запропонований метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання дозволив щонайменше у 1,5 рази покращити результати методу опорних векторів та дерев рішень.

ВИСНОВКИ

У роботі досягнуто мету зменшення часових витрат на організацію персоналізованого навчання дизайнерів-модельєрів на основі програмного засобу за рахунок відбору кандидатів з більшим потенційним підвищенням вмінь у результаті.

Для виконання мети у роботі було розроблено програмний засіб, який дозволяє виконувати прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання. Результати прогнозування дозволяють приймати майстрам дизайну одягу обґрунтовані рішення та обирати серед кандидатів, які подали заявки на навчання, тих, хто більш ймовірно досягне більш значного підвищення власного рівня. Таким чином майстер, розробляючи персоналізовані матеріали, може витратити час більш результативно.

Розроблене ПЗ персоналізованого навчання дизайнерів-модельєрів призначене для його використання дизайнерами-модельєрами для вивчення курсів, які надаються майстрами, представлення себе у вигляді колекцій одягу за технічними пакетами, ознайомлення з ідеями інших дизайнерів-модельєрів, майстрами з дизайну одягу для навчання дизайнерів-модельєрів за своїми курсами, пошуку талантів серед них.

Наукова новизна роботи полягає в методі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання, який характеризується використанням методу Random Forest, що дозволяє виконувати прогнозування зміни рівня знань і вмінь дизайнера-модельєра відносно його минулого середнього рівня.

Запропонований метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання дозволив щонайменше у 1,5 рази покращити результати методу опорних векторів та дерев рішень, з якими відбувалось порівняння під час проведення експериментального дослідження.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. LinkedIn Ukraine [Electronic resource] : Log In or Sign Up. – Access mode : <https://ua.linkedin.com/>.
2. Techpacker [Electronic resource] : Clothing Design & Product Development Software. – Access mode : <https://techpacker.com/>.
3. How to Create a Tech Pack for Shorts [Electronic resource]. – Access mode : <https://techpacker.com/blog/design/how-to-create-a-tech-pack-for-shorts/>.
4. Tech pack Example. Menswear-Jacket [Electronic resource]. – Access mode : <https://techpacker.com/pdf/menswear-jacket/>.
5. Kochar, S. The Ultimate Guide to Tech Packs in Fashion [Electronic resource] / S. Kochar. – Access mode : <https://techpacker.com/blog/design/what-is-a-tech-pack/>.
6. Donges, N. Random Forest [Electronic resource] : A Complete Guide for Machine Learning / N. Donges. – Access mode : <https://builtin.com/data-science/random-forest-algorithm>.
7. Kavlakoglu, E. What is random forest? [Electronic resource] / E. Kavlakoglu. – Access mode : <https://www.ibm.com/think/topics/random-forest>.
8. Random Forest Algorithm in Machine Learning [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/>.
9. Study on prediction model of nitrogen oxide concentration in reprocessing plant based on random forest [Text] / Xuankun Wei, Yan Xu, Xiaomeng Li, Gengxin Fan, Xuening Cheng, Tiantian Yu, Baihua Jiang // International Journal of Advanced Nuclear Reactor Design and Technology. – 2025. – Volume 7, Issue 2. – Pp. 63-69.
10. Prediction of body weight in indigenous sheep using random forest regression, support vector regression, and classification and regression trees algorithms [Electronic resource] / Ashenafi Getachew Megersa, Fikrineh Negash, Abebe Hailu, Awoke Melak, Abraham Assefa, Tesfalem Aseged, Seble Sinkie //

Veterinary and Animal Science. – 2025. – Volume 30. – Article 100517. – Access mode : <https://www.sciencedirect.com/science/article/pii/S2451943X25000894>.

11. Probabilistic prediction of operable sea states with a random forest ensemble [Electronic resource] / Oystein Dorskland, Svein Savik, Zhen Gao, Petter Moen // Marine Structures. – 2025. – Volume 103. – Article 103850. – Access mode : <https://www.sciencedirect.com/science/article/pii/S0951833925000735>.

12. Study on prediction models of oxygenated components content in biomass pyrolysis oil based on neural networks and random forests [Electronic resource] / Yuqian Zou, Hong Tian, Zhangjun Huang, Lei Liu, Yanni Xuan, Jingchao Dai, Liubao Nie // Biomass and Bioenergy. – 2025. – Volume 193. – Article 107601. – Access mode : <https://www.sciencedirect.com/science/article/abs/pii/S0961953425000121>.

13. The Python Tutorial – Python 3.14.2 documentation [Electronic resource]. – Access mode : <https://docs.python.org/3/tutorial/index.html>.

14. Matthes, E. Python Crash Course, 3rd Edition [Text] : A Hands-On, Project-Based Introduction to Programming / E. Matthes. – San Francisco : No Starch Press, 2023. – 511 p.

15. Bader, D. Python Tricks [Text] : A Buffet of Awesome Python Features / D. Bader. – Vancouver : Dan Bader (dbader.org), 2017. – 326 p.

16. Python Basics [Text] : A Practical Introduction to Python 3 / David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler. – Vancouver : Real Python (realpython.com), 2022. – 770 p.

17. Ramalho, L. Fluent Python [Text] : Clear, Concise, and Effective Programming / L. Ramalho. – Sebastopol, CA : O'Reilly Media, 2022. – 1012 p.

18. Using Django. Django documentation. Django [Electronic resource]. – Access mode : <https://docs.djangoproject.com/en/6.0/topics/>.

19. Mele, A. Django 5 By Example [Text] : Build powerful and reliable Python web applications from scratch / A. Mele. – Birmingham : Packt Publishing, 2024. – 820 p.

20. Saha, A. Django in Production [Text] : Expert tips, strategies, and

essential frameworks for writing scalable and maintainable code in Django / A. Saha. – Birmingham : Packt Publishing, 2024. – 554 p.

21. PHP [Electronic resource] : Documentation. – Access mode : <https://www.php.net/docs.php>.

22. Free Template 599 Noir Fashion [Electronic resource]. – Access mode : <https://templatemo.com/tm-599-noir-fashion>.

23. Ensembles [Electronic resource] : Gradient boosting, random forests, bagging, voting, stacking – scikit-learn 1.8.0 documentation. – Access mode : <https://scikit-learn.org/stable/modules/ensemble.html>.

24. Professional Talent in Fashion Design [Electronic resource]. – Access mode : <https://www.kaggle.com/datasets/zara2099/professional-talent-in-fashion-design>.

25. Kavlakoglu, E. What are support vector machines (SVMs)? [Electronic resource] / E. Kavlakoglu. – Access mode : <https://www.ibm.com/think/topics/support-vector-machine>.

26. Support Vector Machine (SVM) Algorithm [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/>.

27. Support Vector Machines – scikit-learn 1.8.0 documentation [Electronic resource]. – Access mode : <https://scikit-learn.org/stable/modules/svm.html>.

28. Kavlakoglu, E. What is a decision tree? [Electronic resource] / E. Kavlakoglu. – Access mode : <https://www.ibm.com/think/topics/decision-trees>.

29. Decision Trees – scikit-learn 1.8.0 documentation [Electronic resource]. – Access mode : <https://scikit-learn.org/stable/modules/tree.html>.

30. Implementing Decision Tree Regression using Scikit-Learn [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/machine-learning/python-decision-tree-regression-using-sklearn/>.

ДОДАТОК А
Технічне завдання

Вступ

Навчання дизайнерів-модельєрів на цифровій платформі потребує наявності спеціальних інструментів, які повинні гарантувати зв'язок між майстрами у цій сфері та користувачами, які тільки навчаються.

Персоналізоване навчання передбачає з одного боку використання персоналізованих матеріалів для користувачів, тобто коли для окремих користувачів можуть надаватися майстром спеціальні матеріали. З іншого боку це потребує розуміння того, кого таким чином будуть навчати. Тобто персоналізовані матеріали можуть потребувати кропіткої роботи, а тому необхідно розуміти, на кого ця робота направлена. Тому дизайнер-модельєр повинен мати можливість продемонструвати свої вміння в програмі, а з іншого боку необхідно застосовувати інструменти прогнозування, які б дозволили передбачити те, на скільки такий курс буде персонально цьому дизайнеру-модельєру корисним.

Саме на це направлена розробка ПЗ персоналізованого навчання дизайнерів-модельєрів і саме такі інструменти мають використовуватися в ньому в основі.

A.1 Підстави для розробки

Наказ № 447 від 30.09.2025 року за Національним університетом «Запорізька політехніка» використовується для даної дипломної кваліфікаційної роботи як підстава для розробки за темою «Дослідження та програмна реалізація методів підтримки персоналізованого навчання дизайнерів-модельєрів».

A.2 Призначення розробки

ПЗ персоналізованого навчання дизайнерів-модельєрів призначене для

його використання:

- дизайнерами-модельєрами для вивчення курсів, які надаються майстрами, представлення себе у вигляді колекцій одягу за технічними пакетами, ознайомлення з ідеями інших дизайнерів-модельєрів;
- майстрами з дизайну одягу для навчання дизайнерів-модельєрів за своїми курсами, пошуку талантів серед них.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Функціональні характеристики ПЗ персоналізованого навчання дизайнерів-модельєрів мають включати:

- відображення інформації стосовно дизайнера-модельєра;
- модифікування загальних даних дизайнера-модельєра;
- додавання та вилучення фотографії дизайнера-модельєра;
- додавання та вилучення програмних вмінь дизайнера-модельєра;
- додавання та вилучення дизайнерських вмінь дизайнера-модельєра;
- додавання та вилучення досвіду роботи дизайнера-модельєра;
- додавання та вилучення сильних сторін дизайнера-модельєра;
- реєстрація та авторизація дизайнера-модельєра;
- підписування на колекції робіт дизайнера-модельєра та скасування;
- відображення колекцій дизайнерів-модельєрів з підписок;
- відображення дизайнерів-модельєрів з підписок;
- створення колекції одягу дизайнера-модельєра;
- редагування колекції одягу дизайнера-модельєра;
- вилучення колекції одягу дизайнера-модельєра;
- відображення всіх колекцій дизайнера-модельєра;
- створення технічного пакету за колекцією одягу;
- редагування технічного пакету за колекцією одягу;
- вилучення технічного пакету за колекцією одягу;

- відображення всіх технічних пакетів за колекцією одягу;
- збереження технічного пакету за колекцією одягу у вкладку;
- скасування збереження технічного пакету за колекцією одягу у вкладку;
- відображення технічних пакетів зі вкладок;
- збереження колекції одягу у вкладку;
- скасування збереження колекції одягу у вкладку;
- відображення колекцій одягу зі вкладок;
- відображення повного вмісту технічного пакету дизайнера-модельєра;
- авторизація майстрів дизайну одягу;
- відображення інформації стосовно майстра дизайну одягу;
- модифікування загальних даних майстра дизайну одягу;
- оцінювання рівня дизайнера-модельєра майстром;
- скасування оцінювання рівня дизайнера-модельєра майстром;
- додавання типового курсу для навчання дизайнерів-модельєрів;
- модифікування типового курсу для навчання дизайнерів-модельєрів;
- вилучення типового курсу для навчання дизайнерів-модельєрів;
- відображення інформації стосовно курсу для навчання дизайнерів-модельєрів;
- відображення курсів для навчання дизайнерів-модельєрів за майстром;
- пошук серед курсів для навчання дизайнерів-модельєрів;
- пошук серед майстрів дизайну одягу;
- пошук серед технічних пакетів зі створення одягу;
- відображення найкращих курсів для навчання дизайнерів-модельєрів за заданим процесом;
- створення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;

- модифікування публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- вилучення публікації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- відображення інформації про проведення навчання за типовим курсом для навчання дизайнерів-модельєрів;
- пошук серед навчань, що проводяться за типовими курсами для навчання дизайнерів-модельєрів;
- відображення всіх матеріалів за курсом для навчання дизайнерів-модельєрів;
- відображення матеріалу за курсом для навчання дизайнерів-модельєрів;
- публікація матеріалу за курсом для навчання дизайнерів-модельєрів;
- модифікування матеріалу за курсом для навчання дизайнерів-модельєрів;
- вилучення матеріалу за курсом для навчання дизайнерів-модельєрів;
- оцінювання результатів навчання дизайнера-модельєра за курсом;
- вилучення оцінювання результатів навчання дизайнера-модельєра за курсом;
- подання заявки на навчання за курсом для дизайнерів-модельєрів;
- скасування заявки на навчання за курсом для дизайнерів-модельєрів;
- підтвердження заявки на навчання за курсом для дизайнерів-модельєрів;
- відображення поданих заявок на навчання за курсом для дизайнерів-модельєрів;
- відображення всіх дизайнерів-модельєрів, які відвідують курс;
- відображення всіх курсів, які відвідує дизайнер-модельєр;
- оцінювання дизайнером-модельєром курсу;
- вилучення оцінювання дизайнером-модельєром курсу;

- створення моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання;
- навчання моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання;
- відображення всіх оцінювань курсу дизайнерами-модельєрами;
- відображення всіх оцінювань, виставлених дизайнеру-модельєру;
- прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання.

А.3.2 Умови експлуатації

Для запуску ПЗ персоналізованого навчання дизайнерів-модельєрів на сервері (на відміну від клієнта, де наявний сучасний браузер та підключення до інтернет є достатніми) потрібно мати:

- сучасний вебсервер (наприклад, Apache);
- засоби підтримки розробки у відповідності з вибором, який має бути зроблений в роботі.

А.3.3 Вимоги до складу та параметрів технічних засобів

Сервер підтримки ПЗ персоналізованого навчання дизайнерів-модельєрів повинен мати такі технічні засоби:

- багатоядерний процесор з частотою щонайменше 3 ГГц;
- 16 Гб оперативної пам'яті мінімально для підтримки повноцінної роботи зі створенням моделі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання (або щонайменше 4 Гб без використання цієї моделі);
- 1 Тб вільного для використання ПЗ обсягу жорсткого диску, який мінімально може бути обмежений 10 Гб початково, але передбачає роботу невеликої групи користувачів або в режимі ознайомлення, або зі значним

обмеженням на кількість та обсяг зображень для технічних пакетів. У випадку обмеження в 1 Тб з них 10 Гб використовується на збереження моделей, витягнутих даних для навчання моделі, даних БД без зображень. Відповідно 90 Гб, що залишаються, за умов максимального використання обсягу в 100 Мб одним користувачем достатньо для роботи мінімум 900 користувачів. Доступний обсяг може бути зменшено, а кількість користувачів збільшена.

A.4 Порядок контролю та приймання

Календарний план на дипломну кваліфікаційну роботу, що є частиною завдання, містить вимоги до контролю роботи з поетапним виконанням. Приймання роботи відбувається керівником, нормоконтролерами, завідувачем кафедри та в підсумку екзаменаційною комісією під час захисту.

ДОДАТОК Б
Текст програми

```

from django.shortcuts import render
from fasdeapp.models import Designskill,\
    Typicalfashionstrength,\
    Fashiondesigner,\
    Fashiondesignerimage,\
    Fashiondesignerconnection,\
    Designersoftware,\
    Fashiondesignersoftwareskill,\
    Designcompany,\
    Designposition,\
    Designexperience,\
    Fashionstyle,\
    Garment,\
    Fashioncollection,\
    Fashionthread,\
    Fashioncollectiontechpack,\
    Fashioncollectiontechpackcolor,\
    Fashioncollectiontechpackimage,\
    Fashioncollectiontechpackdetail,\
    Fashionmaterial,\
    Fashioncollectiontechpackfabric,\
    Fashioncollectiontechpackfabricpart,\
    Fashioncollectiontechpacksaving,\
    Fashioncollectionsaving,\
    Fashionmaster,\
    Fashionmasterlevel,\
    Fashionmasterimage,\
    Fashioncoursetype,\
    Fashionmastercourse,\
    Fashionmastercoursemeeting,\
    Coursevenue,\
    Fashionmastercourseapplication,\
    Fashionmastercoursestuff

from django.core.paginator import Paginator
from django.conf import settings
import datetime

def fashiondesigner_view(request, fashiondesigner_username
= None):
    fashion_master = None
    mode = 0
    fashion_designer_auth = None
    fashion_master_auth = None
    if request.user.is_authenticated:
        fashion_designer = Fashiondesigner.objects. \
            select_related("designeruser"). \
            filter(designeruser = request.user)
        if fashion_designer:
            fashion_designer_auth = fashion_designer[0]
            mode = 3
            if fashion_designer_auth and (not
fashiondesigner_username or fashion_designer_auth.designeruser
== fashiondesigner_username):

```

```

        mode = 4
    elif not fashion_designer_auth:
        return render(
            request,
            "fashiondesigner.html",
            {
                "title": "Профіль дизайнера-
модельєра",
                "mode": -1,
                "error": "Помилка авторизації",
                "fashiondesigner": None,
                "images": None,
                "softwareskills": None,
                "experiences": None,
                "collections": None,
            }
        )
    elif request.user.is_staff:
        mode = 2
    else:
        fashion_master_auth = Fashionmaster.objects.\
filter(masteruser=request.user)
        if fashion_master_auth:
            mode = 1
            fashion_master_auth =
fashion_master_auth[0]
            fashion_course_time =
datetime.datetime.now()
            fashion_master_course_meetings =
Fashionmastercoursemeeting.objects.\
select_related("fashionmastercourse",\
"coursevenue").\
filter(coursemeetingstart__gte = fashion_course_time,\
meetavailableplaces__gte = 1).\
values_list("fashionmastercourse__id")
            fashion_master_courses =
Fashionmastercourse.objects.filter(id__in =
fashion_master_course_meetings)
        else:
            return render(
                request,
                "fashiondesigner.html",
                {
                    "title": "Профіль дизайнера-
модельєра",
                    "mode": -1,
                    "error": "Помилка авторизації",

```

```

        "fashiondesigner": None,
        "images": None,
        "softwareskills": None,
        "experiences": None,
        "collections": None,
    }
    )
    if fashiondesigner_username:
        fashion_designer =
Fashiondesigner.objects.prefetch_related("fashiondesignerskills"
,\
"dashiondesignnertypicalstrengths").\
select_related("designeruser").\
filter(designeruser__username = fashiondesigner_username)
    else:
        fashion_designer =
Fashiondesigner.objects.prefetch_related("fashiondesignerskills"
, \
"dashiondesignnertypicalstrengths"). \
        select_related("designeruser"). \
        filter(designeruser=request.user)
    if fashion_designer:
        fashion_designer = fashion_designer[0]
    else:
        return render(
            request,
            "fashiondesigner.html",
            {
                "title": "Профіль дизайнера-модельєра",
                "mode": mode,
                "error": "Немає такого дизайнера-
модельєра",
                "fashiondesigner": None,
                "images": None,
                "softwareskills": None,
                "experiences": None,
                "collections": None,
            }
        )
    designer_connection = False
    if mode == 3 or mode == 4 or mode == 1:
        fashion_designer_connection =
Fashiondesignerconnection.objects.\
filter(fashiondesignerstarter = fashion_designer,\
fashiondesignerreceiver = fashion_designer_auth)
        if not fashion_designer_connection:

```

```

        fashion_designer_connection =
Fashiondesignerconnection.objects.\

filter(fashiondesignerreceiver=fashion_designer, \

fashiondesignerstarter =fashion_designer_auth)
        if fashion_designer_connection:
            designer_connection = True
            fashion_designer_images =
Fashiondesignerimage.objects.\
                                filter(fashiondesigner =
fashion_designer,\
                                coverness =
False).\
                                order_by("-id")
            fashion_designer_cover = Fashiondesignerimage.objects.\
                                filter(fashiondesigner =
fashion_designer, coverness = True)
            if fashion_designer_cover:

fashion_designer.cover=fashion_designer_cover[0].designerimage.u
rl
                else:
                    fashion_designer_cover = None
                    fashion_designer_softwareskills =
Fashiondesignerssoftwareskill.objects.select_related("designersof
tware")
                    fashion_designer_experiences =
Designexperience.objects.select_related("designcompany").\

filter(fashiondesigner = fashion_designer).\
                                order_by("-
designjobenddate")
                    fashion_designer_collections =
Fashioncollection.objects.select_related("collstyle").\

filter(fashiondesigner = fashion_designer).\
                                order_by("-
collpublicationdt")[:5]
                    auth_state = False
                    if request.user.is_authenticated:
                        auth_state = True
                    for fashion_designer_collection in
fashion_designer_collections:
                        fashion_designer_collection.image =
Fashioncollectiontechpackimage.objects.\

filter(fashioncollectiontechpackcolor__fashioncollectiontechpack
__fashioncollection = fashion_designer_collection).\

order_by("id")[0]
                        fashion_designer_collection.saved = False
                        if auth_state:

```

```

        fashion_collection_saved =
Fashioncollectionsaving.objects.\

filter(designeruser = request.user,\

fashioncollection = fashion_designer_collection).\
        order_by("-
collpublicationdt")
        if fashion_collection_saved:
            fashion_designer_collection.saved = True
        fashion_master = False
        fashion_designer_levels_by_master = None
        fashion_designer_levels = None
        if request.user.is_authenticated:
            if fashion_master_auth:
                fashion_designer_levels_by_master =
Fashionmasterlevel.objects.\

filter(fashionmaster = fashion_master_auth,\

fashiondesigner = fashion_designer)\

.order_by("-actualdt")
        fashion_designer_levels =
Fashionmasterlevel.objects.\

filter(fashiondesigner = fashion_designer).\

exclude(fashionmaster = fashion_master_auth).\
        order_by("-
actualdt")[:5]
        fashion_designer_course_levels =
Fashionmastercourseapplication.objects.\

filter(fashiondesigner = fashion_designer,\

fashionapplicationstatus = 'A').\

exclude(fashiondesignerlevelpoint__isnull = True).\
        order_by("-
fashionapplicationdt")[:5]
        if not fashion_master_auth and
(request.user.is_staff or (not fashiondesigner_username and
fashion_designer_auth)):
            fashion_designer_levels =
Fashionmasterlevel.objects.\

filter(fashiondesigner = fashion_designer).\
        order_by("-
actualdt")[:5]
        fashion_designer_course_levels =
Fashionmastercourseapplication.objects.\

```

```

        filter(fashiondesigner =
fashion_designer,\
fashionapplicationstatus = 'A').\
exclude(fashiondesignerlevelpoint__isnull = True).\
                                                order_by("-
fashionapplicationdt")[:5]

        if fashion_master_auth:
            fashion_designer_applications =
Fashionmastercourseapplication.objects.\
filter(fashiondesigner = fashion_designer,\

fashionmastercoursemeeting__fashionmastercourse__fashionmaster =
fashion_master_auth).\

exclude(fashionapplicationstatus = 'A').\
                                                order_by("-
fashionapplicationdt")[:5]
            fashion_designer_courses =
Fashionmastercourseapplication.objects.\

filter(fashiondesigner = fashion_designer,\

fashionmastercoursemeeting__fashionmastercourse__fashionmaster =
fashion_master_auth,\

fashionapplicationstatus = 'A').\
                                                order_by("-
fashionapplicationdt")[:5]
            if mode == 1:
                return render(
                    request,
                    "fashiondesigner.html",
                    {
                        "title": "Профіль дизайнера-модельєра",
                        "mode": mode,
                        "fashiondesigner": fashion_designer,
                        "images":fashion_designer_images,

"softwareskills":fashion_designer_softwareskills,
                        "experiences":fashion_designer_experiences,
                        "collections":fashion_designer_collections,
                        "levels": fashion_designer_levels,

"courselevels":fashion_designer_course_levels,
                        "masterlevels":
fashion_designer_levels_by_master,
                        "masterapplications":
fashion_designer_applications,
                        "mastercourses": fashion_designer_courses,

```

```

        "connection": designer_connection,
        "courses": fashion_master_courses,
    }
)
if mode == 0:
    return render(
        request,
        "fashiondesigner.html",
        {
            "title": "Профіль дизайнера-модельєра",
            "mode": mode,
            "fashiondesigner": fashion_designer,
            "images":fashion_designer_images,

"softwareskills":fashion_designer_softwareskills,
            "experiences":fashion_designer_experiences,
            "collections":fashion_designer_collections,
        }
    )
elif mode == 4:
    return render(
        request,
        "fashiondesigner.html",
        {
            "title": "Профіль дизайнера-модельєра",
            "mode": mode,
            "fashiondesigner": fashion_designer,
            "images":fashion_designer_images,

"softwareskills":fashion_designer_softwareskills,
            "experiences":fashion_designer_experiences,
            "collections":fashion_designer_collections,
            "levels": fashion_designer_levels,

"courselevels":fashion_designer_course_levels,
        }
    )
elif mode == 3:
    return render(
        request,
        "fashiondesigner.html",
        {
            "title": "Профіль дизайнера-модельєра",
            "mode": mode,
            "fashiondesigner": fashion_designer,
            "images":fashion_designer_images,

"softwareskills":fashion_designer_softwareskills,
            "experiences":fashion_designer_experiences,
            "collections":fashion_designer_collections,
            "connection": designer_connection,
        }
    )

```

```

elif mode == 2:
    return render(
        request,
        "fashiondesigner.html",
        {
            "title": "Профіль дизайнера-модельєра",
            "mode": mode,
            "fashiondesigner": fashion_designer,
            "images": fashion_designer_images,

"softwareskills":fashion_designer_softwareskills,
            "experiences":fashion_designer_experiences,
            "collections":fashion_designer_collections,
            "levels": fashion_designer_levels,

"courselevels":fashion_designer_course_levels,
        }
    )

    from django.core.paginator import Paginator

    def fashiondesigner_collections_view(request, page,
fashiondesigner_username = None):
        fashion_designer = None
        mode = 0
        fashion_designer_auth = None
        fashion_master_auth = None
        if request.user.is_authenticated:
            fashion_designer = Fashiondesigner.objects. \

select_related("designeruser"). \

filter(designeruser = request.user)
            if fashion_designer:
                fashion_designer_auth = fashion_designer[0]
                mode = 3
                if fashion_designer_auth and (not
fashiondesigner_username or fashion_designer_auth.designeruser
== fashiondesigner_username):
                    mode = 4
                elif not fashion_designer_auth:
                    return render(
                        request,
                        "fashiondesignercollections.html",
                        {
                            "title": "Колекції дизайнера-
модельєра",

                            "error": "Помилка авторизації",
                            "mode": -1,
                            "fashiondesigner": None,
                            "collections": None,
                        }
                    )

```

```

        elif request.user.is_staff:
            mode = 2
        else:
            fashion_master_auth = Fashionmaster.objects.\

filter(masteruser=request.user)
        if fashion_master_auth:
            mode = 1
            fashion_master_auth =
fashion_master_auth[0]
        else:
            return render(
                request,
                "fashiondesignercollections.html",
                {
                    "title": "Колекції дизайнера-
моделєра",
                    "error": "Помилка авторизації",
                    "mode": -1,
                    "fashiondesigner": None,
                    "collections": None,
                }
            )
        if not fashiondesigner_username and not
request.user.is_authenticated:
            return render(
                request,
                "fashiondesignercollections.html",
                {
                    "title": "Колекції дизайнера-моделєра",
                    "error": "Ви не є дизайнером-моделєром для
доступу до цієї сторінки",
                    "mode": mode,
                    "fashiondesigner": None,
                    "collections": None,
                }
            )
        elif not fashiondesigner_username:
            fashion_designer = fashion_designer_auth
        else:
            fashion_designer = Fashiondesigner.objects.\

select_related("designeruser").\

filter(designeruser__username = fashiondesigner_username)
        if fashion_designer:
            fashion_designer = fashion_designer[0]
        else:
            return render(
                request,
                "fashiondesignercollections.html",
                {

```

```

        "title": "Колекції дизайнера-
модельєра",
        "error": "Немає такого дизайнера-
модельєра",
        "mode": mode,
        "fashiondesigner": None,
        "collections": None,
    }
)
fashion_designer.cover = Fashiondesignerimage.objects.\
    filter(fashiondesigner =
fashion_designer, coverness = True)
fashion_designer_collections =
Fashioncollection.objects.select_related("collstyle").\
filter(fashiondesigner = fashion_designer).\
    order_by("-
collpublicationdt")
fashion_collections_paginator = Paginator
(fashion_designer_collections, 10)
fashion_designer_collections =
fashion_collections_paginator.get_page(int(page))
for fashion_designer_collection in
fashion_designer_collections:
    fashion_designer_collection.image =
Fashioncollectiontechpackimage.objects.\
filter(fashioncollectiontechpackcolor__fashioncollectiontechpack
__fashioncollection = fashion_designer_collection).\
order_by("id")[0]
    fashion_designer_collection.saved = False
    if mode == 3 or mode == 1:
        fashion_collection_saved =
Fashioncollectionsaving.obects.\
filter(designeruser = request.user,\
fashioncollection = fashion_designer_collection).\
    order_by("-
collpublicationdt")
        if fashion_collection_saved:
            fashion_designer_collection.saved = True
return render(
    request,
    "fashiondesignercollections.html",
    {
        "title": "Колекції дизайнера-модельєра",
        "fashiondesigner": fashion_designer,
        "mode": mode,
        "collections":fashion_designer_collections,
    }
)

```

```

def fashiondesigner_levels_view(request, page, course_level
= True, fashiondesigner_username = None):
    fashion_designer = None
    mode = 0
    fashion_designer_auth = None
    fashion_master_auth = None
    if request.user.is_authenticated:
        fashion_designer = Fashiondesigner.objects. \
            select_related("designeruser"). \
            filter(designeruser=request.user)
    if fashion_designer:
        fashion_designer_auth = fashion_designer[0]
        mode = 3
        if fashion_designer_auth and (
            not fashiondesigner_username or
fashion_designer_auth.designeruser == fashiondesigner_username):
            mode = 4
        elif not fashion_designer_auth:
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-
модельєра",
                    "mode": -1,
                    "error": "Помилка авторизації",
                    "fashiondesigner": None,
                    "levels": None,
                }
            )
    elif request.user.is_staff:
        mode = 2
    else:
        fashion_master_auth = Fashionmaster.objects. \
            filter(masteruser=request.user)
        if fashion_master_auth:
            mode = 1
            fashion_master_auth =
fashion_master_auth[0]
        else:
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-
модельєра",
                    "mode": -1,
                    "error": "Помилка авторизації",
                    "fashiondesigner": None,
                    "levels": None,
                }
            )

```

```

        if not fashiondesigner_username and not
request.user.is_authenticated:
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-модельєра",
                    "mode": mode,
                    "error": "Ви не є дизайнером-модельєром для
доступу до цієї сторінки",
                    "fashiondesigner": None,
                    "levels": None,
                }
            )
        elif not fashiondesigner_username:
            fashion_designer = Fashiondesigner.objects.\
prefetch_related("fashiondesignerskills",\
"dashiondesignertypicalstrengths").\
select_related("designeruser").\
                                filter(designeruser =
request.user)
            if fashion_designer:
                fashion_designer = fashion_designer[0]
            else:
                return render(
                    request,
                    "fashiondesignerlevels.html",
                    {
                        "title": "Оцінка дизайнера-модельєра",
                        "mode": mode,
                        "error": "Ви не є дизайнером-модельєром
для доступу до цієї сторінки",
                        "fashiondesigner": None,
                        "levels": None,
                    }
                )
            else:
                fashion_designer =
Fashiondesigner.objects.prefetch_related("fashiondesignerskills"
,\
"dashiondesignertypicalstrengths").\
select_related("designeruser").\
filter(designeruser__username = fashiondesigner_username)
                if fashion_designer:
                    fashion_designer = fashion_designer[0]
                else:
                    return render(

```

```

        request,
        "fashiondesignerlevels.html",
        {
            "title": "Оцінка дизайнера-модельєра",
            "mode": mode,
            "error": "Немає такого дизайнера-
модельєра",
            "fashiondesigner": None,
            "levels": None,
        }
    )
    fashion_designer.cover = Fashiondesignerimage.objects.\
        filter(fashiondesigner =
fashion_designer, coverness = True)
    fashion_master = False
    fashion_designer_levels_by_master = None
    fashion_designer_levels = None
    if request.user.is_authenticated and
fashiondesigner_username:
        if fashion_master_auth:
            if not course_level:
                fashion_designer_levels_by_master =
Fashionmasterlevel.objects.\

filter(fashionmaster = fashion_master_auth,\

fashiondesigner = fashion_designer)\

.order_by("-actualdt")
            if fashion_master_auth or (not
fashiondesigner_username and fashion_designer) or
request.user.is_staff:
                if not course_level:
                    fashion_designer_levels =
Fashionmasterlevel.objects.\

filter(fashiondesigner=fashion_designer).\

exclude(fashionmaster=fashion_master_auth).\
                    order_by("-actualdt")
            else:
                fashion_designer_levels =
Fashionmastercourseapplication.objects.\

filter(fashiondesigner=fashion_designer,\
                    fashionapplicationstatus='A').\

exclude(fashiondesignerlevelpoint__isnull=True).\
                    order_by("-fashionapplicationdt")
        fashion_designer_levels_paginator = Paginator
(fashion_designer_levels, 10)
        fashion_designer_levels =
fashion_designer_levels_paginator.get_page(int(page))

```

```

        if not fashion_master_auth and (not
fashiondesigner_username and fashion_designer):
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-модельєра",
                    "mode": mode,
                    "fashiondesigner": fashion_designer,
                    "levels":fashion_designer_levels,
                }
            )
        elif request.user.is_staff:
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-модельєра",
                    "mode": mode,
                    "fashiondesigner": fashion_designer,
                    "levels":fashion_designer_levels,
                }
            )
        elif fashion_master_auth:
            return render(
                request,
                "fashiondesignerlevels.html",
                {
                    "title": "Оцінка дизайнера-модельєра",
                    "mode": mode,
                    "fashiondesigner": fashion_designer,
                    "levels": fashion_designer_levels,
                    "masterlevels":
fashion_designer_levels_by_master,
                }
            )

    def fashiondesigner_courses_view(request,
fashioncourse_status = 0, designeruser_username = None):
        fashion_designer = None
        mode = 0
        fashion_designer_auth = None
        fashion_master_auth = None
        if request.user.is_authenticated:
            fashion_designer = Fashiondesigner.objects. \
                select_related("designeruser"). \
                filter(designeruser=request.user)
            if fashion_designer:
                fashion_designer_auth = fashion_designer[0]
                mode = 3
                if fashion_designer_auth and (
                    not fashiondesigner_username or
fashion_designer_auth.designeruser == fashiondesigner_username):

```

```

        mode = 4
    elif not fashion_designer_auth:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Ваші курси",
                "mode": -1,
                "error": "Помилка авторизації",
                "fashiondesigner": None,
                "courses": None,
            }
        )
    elif request.user.is_staff:
        mode = 2
    else:
        fashion_master_auth = Fashionmaster.objects.\
            filter(masteruser=request.user)
        if fashion_master_auth:
            mode = 1
            fashion_master_auth =
fashion_master_auth[0]
        else:
            return render(
                request,
                "fashiondesignercourses.html",
                {
                    "title": "Ваші курси",
                    "mode": -1,
                    "error": "Помилка авторизації",
                    "fashiondesigner": None,
                    "courses": None,
                }
            )
        if mode > 0 and not designeruser_username:
            fashion_designer = Fashiondesigner.objects.\
prefetch_related("fashiondesignerskills",\
"dashiondesignertypicalstrengths").\
select_related("designeruser").\
                                filter(designeruser =
request.user)
        if fashion_designer:
            fashion_designer = fashion_designer[0]
        else:
            return render(
                request,
                "fashiondesignercourses.html",
                {
                    "title": "Ваші курси",

```

```

        "error": "Немає такого дизайнера-
модельєра",
        "mode": mode,
        "fashiondesigner": None,
        "courses": None,
    }
)
elif mode == 2 and designeruser_username:
    fashion_designer = Fashiondesigner.objects. \
        prefetch_related("fashiondesignerskills", \
"dashiondesignertypicalstrengths"). \
        select_related("designeruser"). \
filter(designeruser__username=designeruser_username)
    if fashion_designer:
        fashion_designer = fashion_designer[0]
    else:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Курси дизайнера-модельєра",
                "error": "Для отримання доступу
потрібно бути дизайнером-модельєром",
                "mode": mode,
                "fashiondesigner": None,
                "courses": None,
            }
        )
    elif mode == 2:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Курси дизайнера-модельєра",
                "error": "Не задано дизайнера-модельєра",
                "mode": mode,
                "fashiondesigner": None,
                "courses": None,
            }
        )
    else:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Ваші курси",
                "error": "Немає прав доступу",
                "mode": mode,
                "fashiondesigner": None,
                "courses": None,
            }
        )

```

```

    )
    fashion_designer.cover = Fashiondesignerimage.objects.\
        filter(fashiondesigner =
fashion_designer, coverness = True)
    fashion_designer_courses = None
    if fashioncourse_status == 0:
        fashion_designer_courses =
Fashionmastercourseapplication.objects.\

filter(fashiondesigner=fashion_designer).\
        order_by("-fashionapplicationdt")
    elif fashioncourse_status == 1:
        fashion_designer_courses =
Fashionmastercourseapplication.objects. \
        filter(fashiondesigner=fashion_designer,\
            fashionapplicationstatus='A'). \
        order_by("-fashionapplicationdt")
    elif fashioncourse_status == 2:
        fashion_designer_courses =
Fashionmastercourseapplication.objects. \
        filter(fashiondesigner=fashion_designer,\
            fashionapplicationstatus='F'). \
        order_by("-fashionapplicationdt")
    elif fashioncourse_status == 2:
        fashion_designer_courses =
Fashionmastercourseapplication.objects. \
        filter(fashiondesigner=fashion_designer,\
            fashionapplicationstatus='C'). \
        order_by("-fashionapplicationdt")
    fashion_designer_courses.paginator = Paginator
(fashion_designer_courses, 10)
    fashion_designer_courses =
fashion_designer_courses.paginator.get_page(int(page))
    if mode == 2:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Курси дизайнера-модельєра",
                "mode": mode,
                "fashiondesigner": fashion_designer,
                "courses": fashion_designer_courses,
            }
        )
    else:
        return render(
            request,
            "fashiondesignercourses.html",
            {
                "title": "Ваші курси",
                "mode": mode,
                "fashiondesigner": fashion_designer,
                "courses":fashion_designer_courses,

```

```

        }
    )

    def fashionmaster_view(request, fashionmaster_username =
None):
        fashion_master = None
        mode = 0
        fashion_designer_auth = None
        fashion_master_auth = None
        if request.user.is_authenticated:
            fashion_designer = Fashiondesigner.objects. \
                select_related("designeruser"). \
                filter(designeruser=request.user)
            if fashion_designer:
                fashion_designer_auth = fashion_designer[0]
                mode = 1
            elif request.user.is_staff:
                mode = 2
            else:
                fashion_master_auth = Fashionmaster.objects. \
                    filter(masteruser=request.user)
                if fashion_master_auth and (
                    not fashionmaster_username or
fashion_master_auth[0].masteruser == fashionmaster_username):
                    mode = 4
                    fashion_master_auth =
fashion_master_auth[0]
                elif fashion_master_auth:
                    mode = 3
                    fashion_master_auth =
fashion_master_auth[0]
                else:
                    return render(
                        request,
                        "fashionmaster.html",
                        {
                            "title": "Профіль майстра з дизайну
одягу",
                            "error": "Помилка авторизації",
                            "mode": -1,
                            "fashionmaster": None,
                            "images": None,
                            "courses": None,
                        }
                    )
            if fashionmaster_username:
                fashion_master = Fashionmaster.objects.\
                    filter(masteruser__username = fashionmaster_username)
            if fashion_master:
                fashion_master = fashion_master[0]
            else:
                return render(

```

```

        request,
        "fashionmaster.html",
        {
            "title": "Профіль майстра з дизайну
одягу",
            "error": "Такий майстер з дизайну одягу
невідомий.",
            "mode": mode,
            "fashionmaster": None,
            "images": None,
            "courses": None,
        }
    )
else:
    fashion_master = Fashionmaster.objects.\
        filter(masteruser =
request.user)
    if fashion_master:
        fashion_master = fashion_master[0]
    else:
        return render(
            request,
            "fashionmaster.html",
            {
                "title": "Профіль майстра з дизайну
одягу",
                "error": "Ви не є майстром, ваш профіль
не може бути продемонстрований.",
                "mode": mode,
                "fashionmaster": None,
                "images": None,
                "courses": None,
            }
        )
    fashion_master.cover = Fashionmasterimage.objects.\
filter(fashionmaster=fashion_master, fmcoverness=True)
    fashion_master_images = Fashionmasterimage.objects.\
filter(fashionmaster=fashion_master, fmcoverness=False)
    fashion_master_courses = Fashionmastercourse.objects.\
filter(fashionmaster=fashion_master).\
        order_by("-
fashioncourselevel")[:5]
    if request.user.is_authenticated:
        fashion_designer = Fashiondesigner.objects.\
            prefetch_related("fashiondesignerskills", \
"dashiondesignnertypicalstrengths").\
                select_related("designeruser").\
                    filter(designeruser=request.user)
        if fashion_designer:

```

```

        fashion_designer = fashion_designer[0]
        for fashion_master_course in
fashion_master_courses:
            fashion_master_course.application =
Fashionmastercourseapplication.objects.\

filter(fashiondesigner=fashion_designer,\

fashion_master_course = fashion_master_course).\

order_by("-fashionapplicationdt")
    return render(
        request,
        "fashionmaster.html",
        {
            "title": "Профіль майстра з дизайну одягу",
            "mode": mode,
            "fashionmaster": fashion_master,
            "images": fashion_master_images,
            "courses": fashion_master_courses,
        }
    )

    def fashionmaster_courses_view(request, page,
fashionmaster_username=None):
        fashion_master = None
        mode = 0
        fashion_designer_auth = None
        fashion_master_auth = None
        if request.user.is_authenticated:
            fashion_designer = Fashiondesigner.objects. \

select_related("designeruser"). \

filter(designeruser=request.user)
        if fashion_designer:
            fashion_designer_auth = fashion_designer[0]
            mode = 1
        elif request.user.is_staff:
            mode = 2
        else:
            fashion_master_auth = Fashionmaster.objects. \

filter(masteruser=request.user)
            if fashion_master_auth and (not
fashionmaster_username or fashion_master_auth[0].masteruser ==
fashionmaster_username):
                mode = 4
                fashion_master_auth =
fashion_master_auth[0]
            elif fashion_master_auth:
                mode = 3

```

```

        fashion_master_auth =
fashion_master_auth[0]
        else:
            return render(
                request,
                "fashionmastercourses.html",
                {
                    "title": "Курси майстра з дизайну
одягу",
                    "error": "Помилка авторизації.",
                    "fashionmaster": None,
                    "courses": None,
                    "mode": -1,
                }
            )
        if fashionmaster_username:
            fashion_master = Fashionmaster.objects. \
filter(masteruser__username=fashionmaster_username)
            if fashion_master:
                fashion_master = fashion_master[0]
            else:
                return render(
                    request,
                    "fashionmastercourses.html",
                    {
                        "title": "Курси майстра з дизайну
одягу",
                        "error": "Такий майстер з дизайну
одягу невідомий.",
                        "fashionmaster": None,
                        "courses": None,
                        "mode": mode,
                    }
                )
            fashion_master_cover = Fashionmasterimage.objects. \
filter(fashionmaster=fashion_master, fmcoverness=True)[0]
            fashion_master.cover =
fashion_master_cover.fmpersonimage.url
            fashion_master_courses = Fashionmastercourse.objects. \
filter(fashionmaster=fashion_master). \
order_by("-fashioncourselevel")
            fashion_course_time = datetime.datetime.now()
            fashion_master_courses_paginator = Paginator
(fashion_master_courses, 10)
            fashion_master_courses =
fashion_master_courses_paginator.get_page(int(page))
            for fashion_master_course in fashion_master_courses:
                fashion_master_course.meetings =
Fashionmastercoursemeeting.objects.\

```

```

filter(fashionmastercourse
= fashion_master_course,\
coursemeetingstart__gte = fashion_course_time).\
order_by("coursemeetingstart")
        if request.user.is_authenticated and
fashion_designer_auth:
            for fashion_master_course in
fashion_master_courses:
                fashion_master_course.applications =
Fashionmastercourseapplication.objects. \
filter(fashiondesigner=fashion_designer_auth, \
fashionmastercoursemeeting__fashionmastercourse=fashion_master_c
ourse).\
order_by("-fashionapplicationdt")
        if mode == 4:
            return render(
                request,
                "fashionmastercourses.html",
                {
                    "title": "Ваші курси",
                    "fashionmaster": fashion_master,
                    "courses": fashion_master_courses,
                    "mode": mode,
                }
            )
        else:
            return render(
                request,
                "fashionmastercourses.html",
                {
                    "title": "Курси майстра з дизайну
одягу",
                    "fashionmaster": fashion_master,
                    "courses": fashion_master_courses,
                    "mode": mode,
                }
            )

from django.db.models import Q

def fashion_course_stuffs_view(request,
fashionmastercourse_id, page):
    if not request.user.is_authenticated:
        return render(
            request,
            "fashionmastercoursestuffs.html",
            {
                "title": "Матеріали курсу",

```

```

        "mode": 0,
        "error": "Ви не є майстром з дизайну одягу
або дизайнером-модельєром",
        "fashioncourse": None,
        "meeting": None,
        "stuffs": None,
    }
)
fashion_master = Fashionmaster.objects.\
    filter(masteruser =
request.user)
mode = 0
if fashion_master:
    fashion_master = fashion_master[0]
    mode = 3
if request.user.is_staff:
    mode = 2
if mode == 2 or mode == 3:
    if fashion_master:
        fashionmaster_course =
Fashionmastercourse.objects.\

filter(fashionmaster = fashion_master,\

id = fashionmastercourse_id)
        if fashionmaster_course:
            fashionmaster_course =
fashionmaster_course[0]
        else:
            return render(
                request,
                "fashionmastercoursestuffs.html",
                {
                    "title": "Матеріали курсу",
                    "mode": 3,
                    "error": "Це не ваш курс",
                    "fashioncourse": None,
                    "meeting": None,
                    "stuffs": None,
                }
            )
    else:
        fashionmaster_course =
Fashionmastercourse.objects. \

filter(id=fashionmastercourse_id)
        if fashionmaster_course:
            fashionmaster_course =
fashionmaster_course[0]
        else:
            return render(
                request,
                "fashionmastercoursestuffs.html",

```

```

        {
            "title": "Матеріали курсу",
            "mode": 2,
            "error": "Немає такого курсу",
            "fashioncourse": None,
            "meeting": None,
            "stuffs": None,
        }
    )
    fashion_course_time = datetime.datetime.now()
    fashion_course_stuffs =
Fashionmastercoursestuff.objects.\

filter(fashionmastercourse__id = int(fashionmastercourse_id)).\

order_by("activationstartdt")
    fashion_course_stuffs_paginator =
Paginator(fashion_course_stuffs, 10)
    fashion_course_stuffs =
fashion_course_stuffs_paginator.get_page(int(page))
    fashion_course_meetings =
Fashionmastercoursemeeting.objects.\

select_related("coursevenue").\

filter(fashionmastercourse = fashionmaster_course,\

coursemeetingstart__gte = fashion_course_time).\

order_by("coursemeetingstart__gte")
    if fashion_master:
        return render(
            request,
            "fashionmastercoursestuffs.html",
            {
                "title": "Матеріали курсу",
                "mode": mode,
                "fashioncourse": fashionmaster_course,
                "meeting": fashion_course_meetings,
                "stuffs": fashion_course_stuffs,
            }
        )
    else:
        return render(
            request,
            "fashionmastercoursestuffs.html",
            {
                "title": "Матеріали курсу",
                "mode": mode,
                "fashioncourse": fashionmaster_course,
                "meeting": fashion_course_meetings,
                "stuffs": fashion_course_stuffs,
            }
        )

```

```

    )
    else:
        fashion_designer = Fashiondesigner.objects. \
select_related("designeruser"). \
filter(designeruser=request.user)
        fashionmaster_course = Fashionmastercourse.objects. \
            filter(id=fashionmastercourse_id)
        if fashionmaster_course:
            fashionmaster_course = fashionmaster_course[0]
            mode = 1
        else:
            return render(
                request,
                "fashionmastercoursestuffs.html",
                {
                    "title": "Матеріали курсу",
                    "mode": mode,
                    "error": "Немає такого курсу",
                    "fashioncourse": None,
                    "meeting": None,
                    "stuffs": None,
                }
            )
        fashionmastercourse_application =
Fashionmastercourseapplication.objects.\
select_related("fashionmastercoursemeeting"). \

select_related("fashionmastercoursemeeting__coursevenue"). \

filter(fashionmastercoursemeeting__fashionmastercourse=fashionma
ster_course,\

fashiondesigner = fashion_designer,

fashionapplicationstatus = 'A'). \

order_by("fashionmastercoursemeeting__coursemeetingstart")
        if not fashionmastercourse_application:
            return render(
                request,
                "fashionmastercoursestuffs.html",
                {
                    "title": "Матеріали курсу",
                    "mode": mode,
                    "error": "Немає прав доступу до такого
курс",
                    "fashioncourse": None,
                    "meeting": None,
                    "stuffs": None,

```

```

        }
    )
    else:
        fashion_course_time = datetime.datetime.now()
        fashion_course_stuffs =
Fashionmastercoursestuff.objects. \

filter(Q(fashionmastercourse__id=int(fashionmastercourse_id)),\

Q(activeness = True),\

Q(accesstype = True) | Q(accesstype = False,
fashionmastercoursestuffapplications__in =
fashionmastercourse_application),\

Q(activationstartdt__gte = fashion_course_time),\

Q(activationenddt__lte = fashion_course_time)). \

order_by("activationstartdt")
        fashion_course_stuffs_paginator =
Paginator(fashion_course_stuffs, 10)
        fashion_course_stuffs =
fashion_course_stuffs_paginator.get_page(int(page))
        fashion_course_meetings =
Fashionmastercoursemeeting.objects. \

select_related("coursevenue"). \

filter(fashionmastercourse=fashionmaster_course, \

coursemeetingstart__gte=fashion_course_time). \

order_by("coursemeetingstart")
        return render(
            request,
            "fashionmastercoursestuffs.html",
            {
                "title": "Матеріали курсу",
                "mode": mode,
                "fashioncourse": fashionmaster_course,
                "application":
fashionmastercourse_application,
                "meeting": fashion_course_meetings,
                "stuffs": fashion_course_stuffs,
            }
        )

```

ДОДАТОК В
Слайди презентації

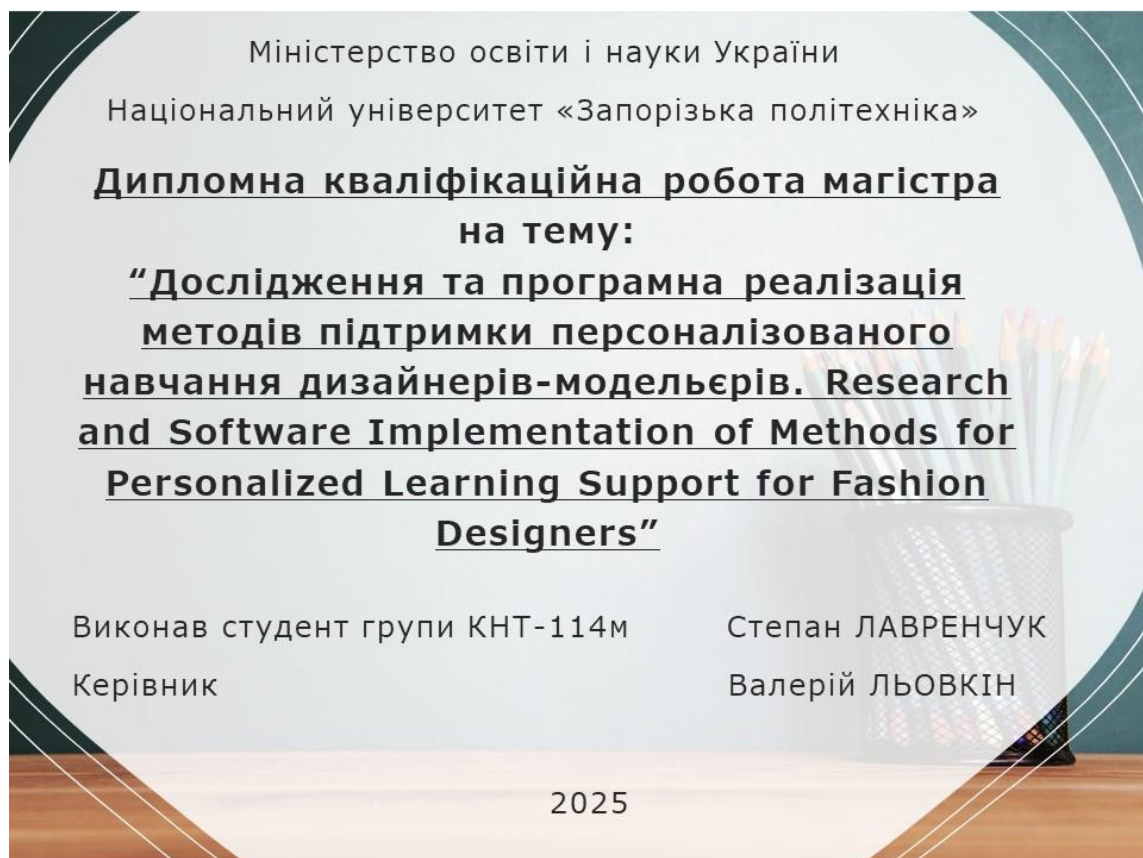


Рисунок В.1 – Початковий слайд

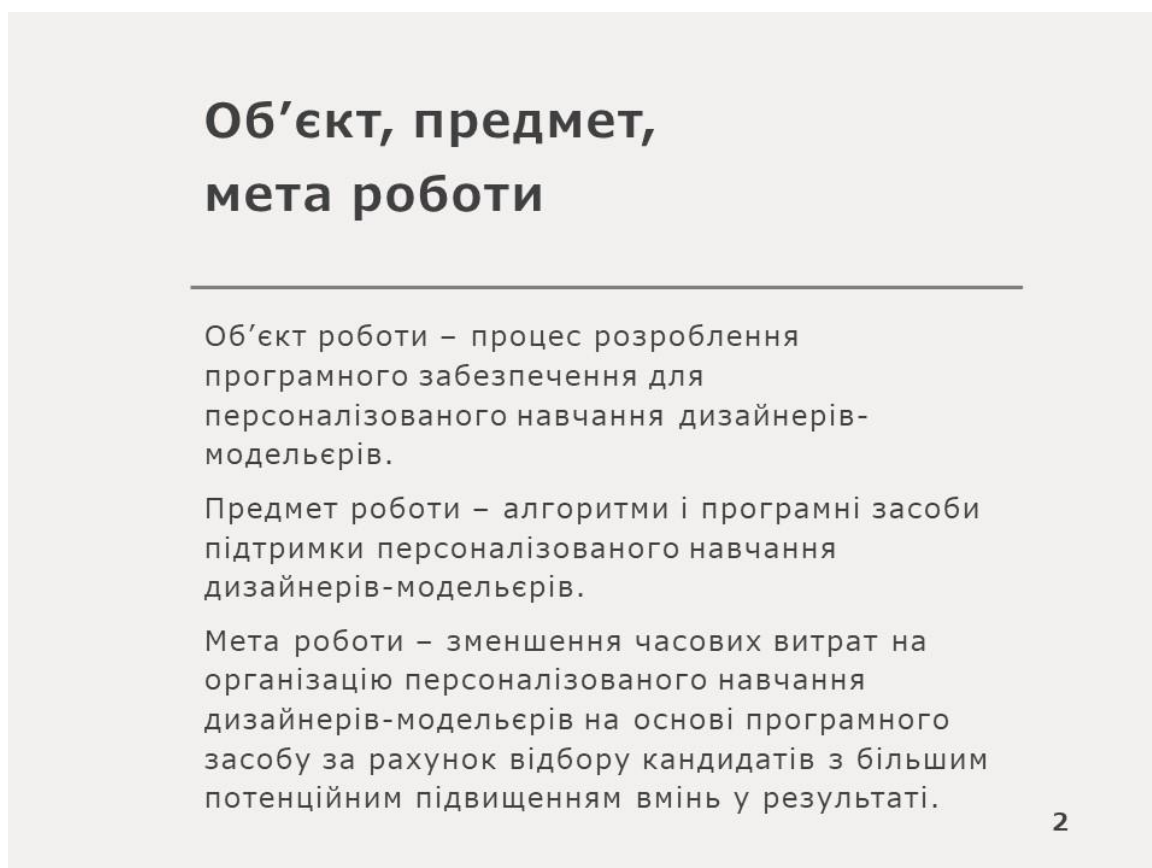


Рисунок В.2 – Об'єкт, предмет, мета роботи

Завдання роботи

- аналіз програм персоналізованого навчання дизайнерів-модельєрів;
- розробити метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра;
- проектування програми персоналізованого навчання дизайнерів-модельєрів;
- реалізація програми персоналізованого навчання дизайнерів-модельєрів;
- експериментальне дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра.

3

Рисунок В.3 – Завдання роботи

Порівняння аналогів вебзастосунку персоналізованого навчання дизайнерів-модельєрів

Показник	LinkedIn	Techpacker	FasDeApp
Створення технічних пакетів	–	+	+
Перегляд технічних пакетів	Як загальні публікації	+	+
Пошук дизайнерів-модельєрів, майстрів	+	–	+
Матеріали для навчання дизайнерів-модельєрів	Загальні, неперсоналізовані	–	Персоналізовані
Прогнозування результатів підвищення рівня вмінь дизайнера-модельєра	–	–	+

4

Рисунок В.4 – Порівняння аналогів вебзастосунку персоналізованого навчання дизайнерів-модельєрів

Метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання



5

Рисунок В.5 – Метод прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Вибір мови для програмування програмного забезпечення персоналізованого навчання дизайнерів-модельєрів

	Python	PHP
Напрямок застосування для розробки ПЗ	Обробка даних, машинне навчання, розробка вебзастосунків	Розробка вебзастосунків
Підтримка ПЗ	Менше ресурсів	Більше ресурсів
Витрати ресурсів на роботу з даними, моделювання	Менше ресурсів	Більше ресурсів
Витрати ресурсів на розробку вебзастосунків	Співмірний обсяг ресурсів	Співмірний обсяг ресурсів

6

Рисунок В.6 – Вибір мови для програмування програмного забезпечення персоналізованого навчання дизайнерів-модельєрів

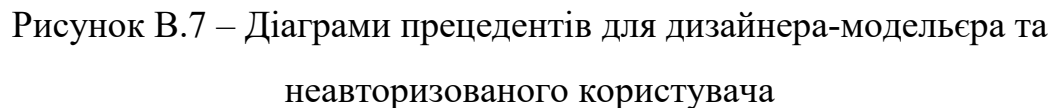
[illegible]

Рисунок В.8 – Діаграми прецедентів для майстра дизайну одягу та адміністратора

Схема бази даних

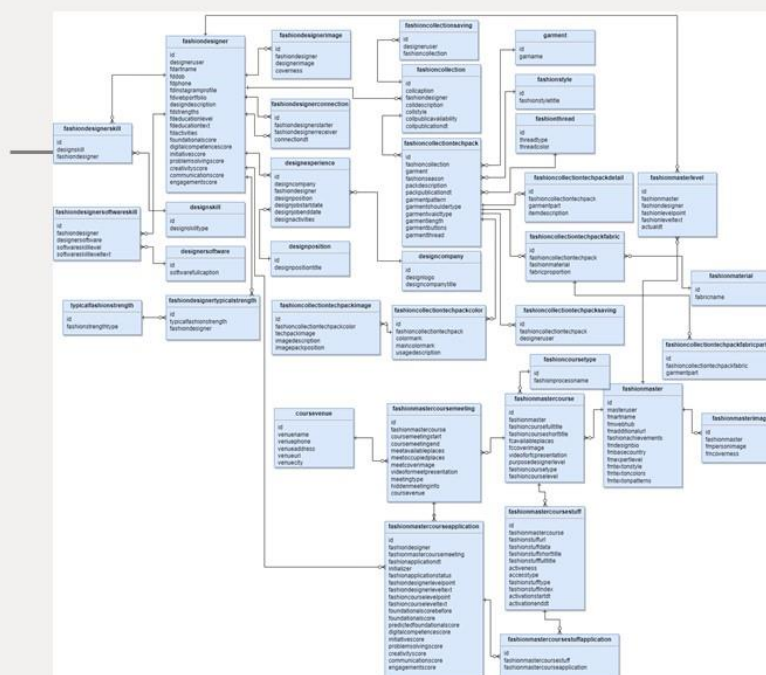


Рисунок В.9 – Схема БД

Функціональна схема програми персоналізованого навчання дизайнерів-модельєрів

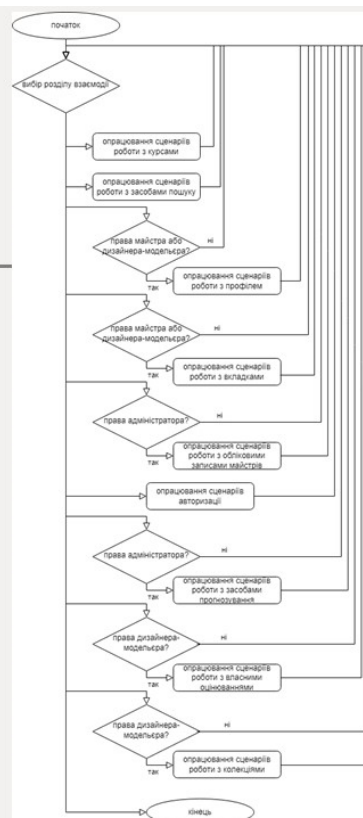


Рисунок В.10 – Функціональна схема програми персоналізованого навчання дизайнерів-модельєрів

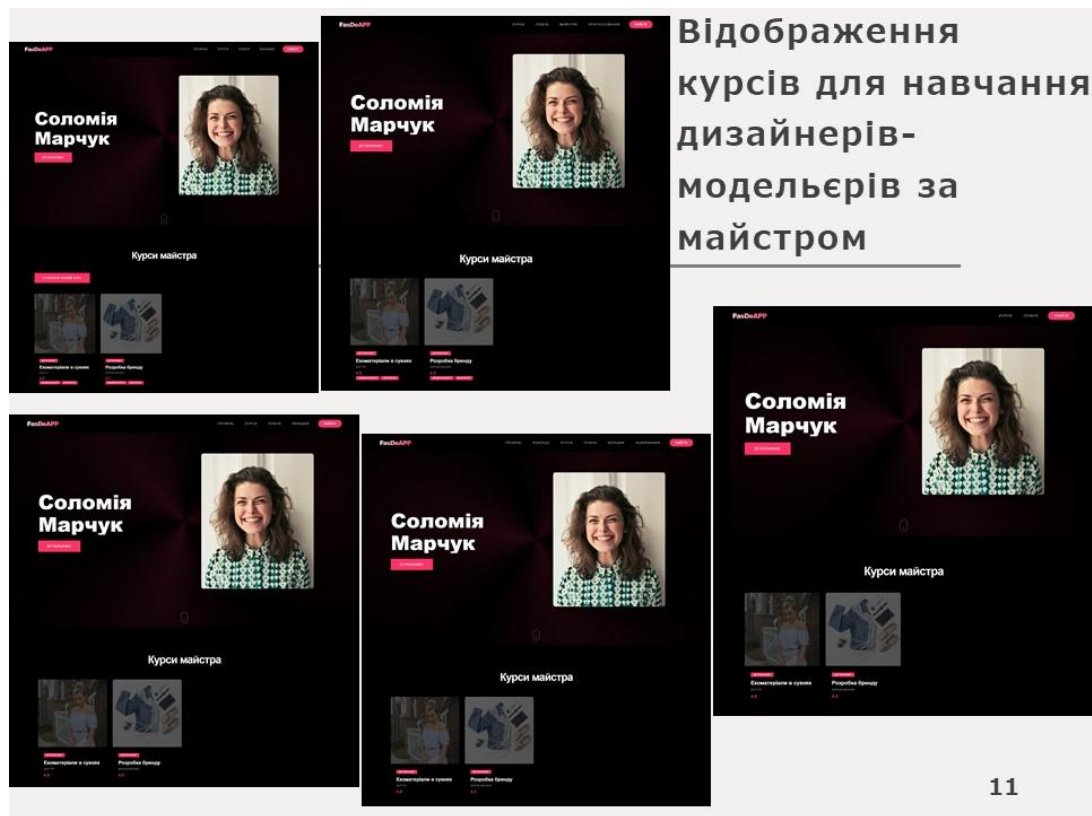


Рисунок В.11 – Відображення курсів для навчання дизайнерів-модельєрів за майстром

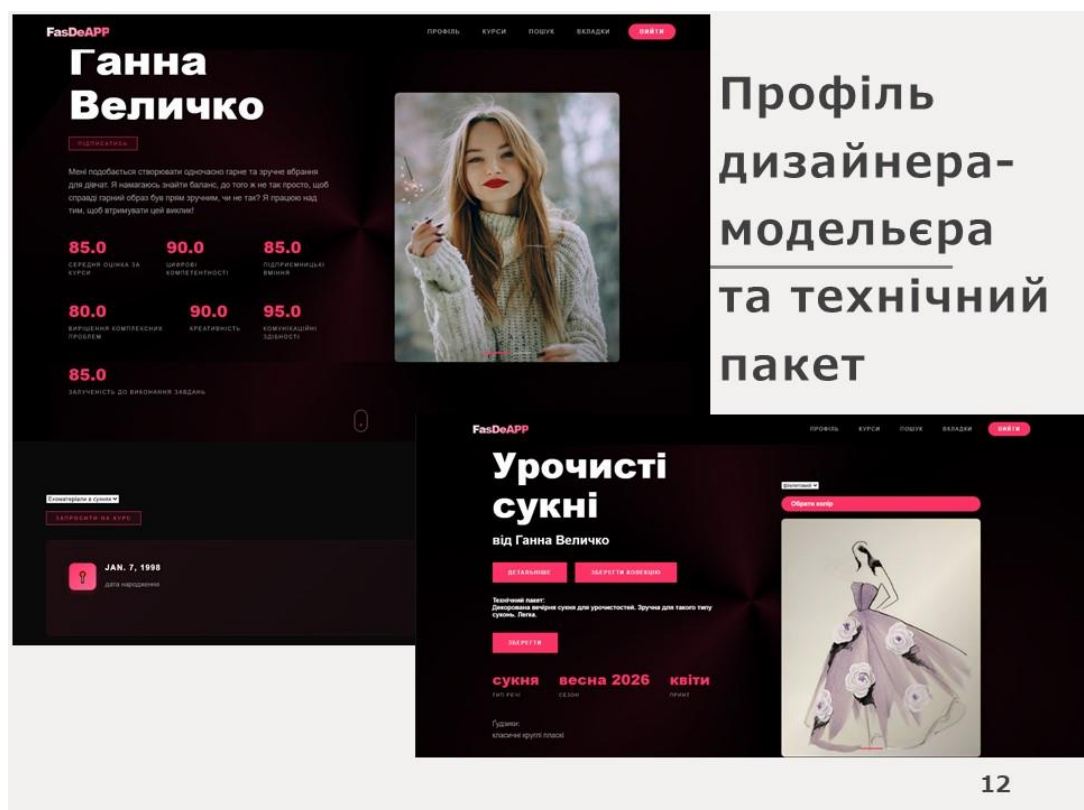


Рисунок В.12 – Профіль дизайнера-модельєра та технічний пакет

Прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

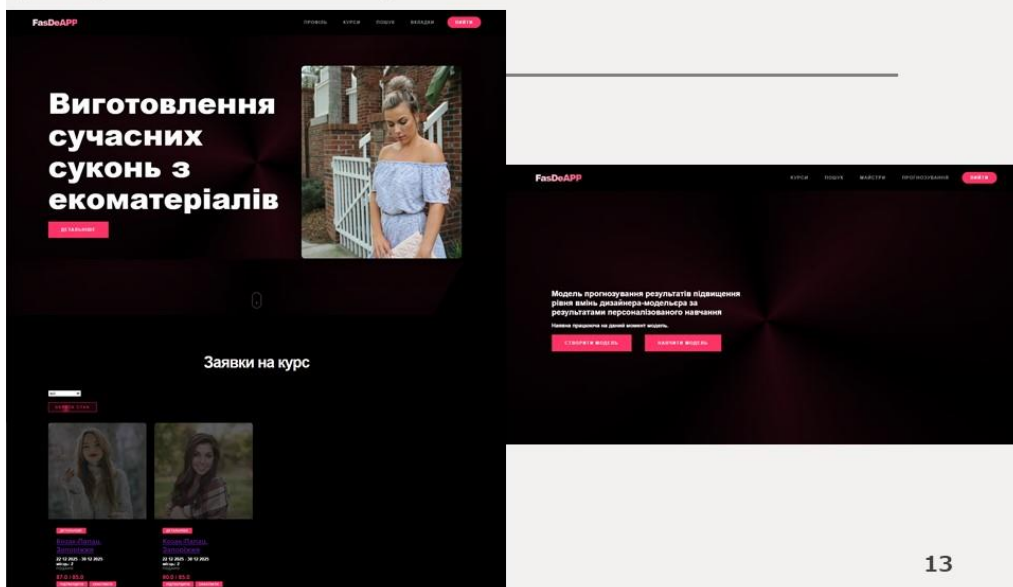


Рисунок В.13 – Прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Експериментальне дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Метод / модель	MAE	RMSE	MSE
Дерево рішень	0,00357646	0,00447873	0,00002006
Метод опорних векторів	0,00572777	0,00678967	0,0000461
Random Forest (запропонований метод)	0,00235198	0,00294896	0,00000869

14

Рисунок В.14 – Експериментальне дослідження прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання

Висновки

У роботі досягнуто мету зменшення часових витрат на організацію персоналізованого навчання дизайнерів-модельєрів на основі програмного засобу за рахунок відбору кандидатів з більшим потенційним підвищенням вмінь у результаті.

Розроблене програмне забезпечення персоналізованого навчання дизайнерів-модельєрів призначене для його використання дизайнерами-модельєрами для вивчення курсів, які надаються майстрами, представлення себе у вигляді колекцій одягу за технічними пакетами, ознайомлення з ідеями інших дизайнерів-модельєрів, майстрами з дизайну одягу для навчання дизайнерів-модельєрів за своїми курсами, пошуку талантів серед них.

Наукова новизна роботи полягає в методі прогнозування результатів підвищення рівня вмінь дизайнера-модельєра за результатами персоналізованого навчання, який характеризується використанням методу Random Forest, що дозволяє виконувати прогнозування зміни рівня знань і вмінь дизайнера-модельєра відносно його минулого середнього рівня.

15

Рисунок В.15 – Висновки