

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до кваліфікаційного проєкту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПІДТРИМКИ ОРГАНІЗАЦІЇ ВИБЧЕННЯ ОНЛАЙН КУРСІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
SUPPORTING THE ONLINE COURSES

Виконав: студент(ка) 2 курсу, групи КНТ-220м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Лазарєв М.О.

(прізвище та ініціали)

Керівник Скрупський С.Ю.

(прізвище та ініціали)

Рецензент Голуб Т.В.

(прізвище та ініціали)

Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н И Я

НА КВАЛІФІКАЦІЙНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Лазарєва Микити Олександровича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів підтримки організації вивчення онлайн курсів
Research and Software Implementation of Methods for Supporting the Online Courses

керівник проєкту (роботи) Скрупський Степан Юрійович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “5” листопада 2021 року № 435

2. Строк подання студентом проекту (роботи) 13 грудня 2021 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 компонентів системи. 5. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
Слайди презентації _____

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Скрупський С.Ю., доцент		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “ 7 ” вересня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз та дослідження предметної області	1-2 тиждень	
2	Розробка та затвердження технічного завдання	3 тиждень	Технічне завдання
3	Дослідження методів розв’язання проблеми	4 тиждень	
4	Проектування програмного забезпечення	5 тиждень	
5	Розроблення модулів програми	6-7 тиждень	
6	Розроблення програмного коду	8 тиждень	Текст програми
7	Тестування та відлагодження програмного забезпечення	9 тиждень	Програма
8	Розробка розділів пояснювальної записки	10-11 тиждень	Пояснювальна записка
9	Розробка графічного матеріалу	12 тиждень	Графічний матеріал
10	Захист магістерської роботи	16 тиждень	

Студент(ка)

(підпис) Лазарєв М.О.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Скрупський С.Ю.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 91 с., 26 рис., 3 табл., 3 дод., 18 джерел.

ОНЛАЙН КУРСИ, НАВЧАННЯ, КЛАСИФІКАЦІЯ ДАНИХ, МАШИННЕ НАВЧАННЯ, НЕЙРОННА МЕРЕЖА ПРЯМОГО ПОШИРЕННЯ.

Об'єкт роботи – процес вивчення онлайн курсів.

Предмет роботи – методи класифікації для визначення успішності вивчення онлайн курсів.

Мета роботи – розроблення програмного забезпечення вивчення онлайн курсів для підтримки студентів на ранніх етапах.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C#, бібліотека Keras.NET.

Результати. Результатами роботи є створені діаграми прецедентів, схема бази даних, схема функціонування програми, її структурна схема, створені класи застосунку. Науковою новизною роботи є запропонований метод прогнозування успішності вивчення онлайн курсу студентом, який виконує прогнозування шляхом вирішення задачі класифікації на основі застосування багатошарової нейронної мережі прямого поширення, що дозволяє приймати рішення щодо необхідності додаткової підтримки вивчення курсів студентами на ранніх етапах.

Висновки. Розроблений програмний застосунок призначений для вивчення онлайн курсів і надає викладачам засоби роботи авторів курсів для публікації власних онлайн курсів і підтримки викладання цих курсів студентам, а студентам – засоби вивчення онлайн курсів і роботи з матеріалами цих курсів.

Галузь використання – навчання курсам онлайн в межах підприємств.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 91 pages, 26 figures, 3 tables, 3 appendixes, 18 sources.

ONLINE COURSES, TRAINING, DATA CLASSIFICATION, MACHINE LEARNING, FEED FORWARD NEURAL NETWORK.

The object of the work is the process of learning online courses.

Subject of work is classification methods to predict the success of online courses learning.

The aim of the work is to develop software for studying online courses to support students in the early learning stages.

Materials, methods and technical means: object-oriented programming, C# programming language, Keras.NET library.

Results. The results of the work are created precedent diagrams, database schema, program operation diagram, its structural diagram, application classes created. The scientific novelty of the work is the proposed method of predicting the success of online courses by students, which performs forecasting by solving the problem of classification based on the use of multilayer neural network direct propagation, which allows to decide on the need to support students in the early stages.

Conclusions. The developed software application is designed to study online courses and provides teachers with tools for course authors to publish their own online courses and support the teaching of these courses to students, and to provide students with tools for learning online courses and working with materials for these courses.

Field of use is learning of online courses inside enterprises.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Огляд сучасного стану проблеми онлайн навчання	11
1.2 Огляд програмних аналогів.....	12
1.3 Визначення сутностей предметної області.....	15
1.4 Висновки за розділом 1 та постановка завдань дослідження.....	16
2 Матеріали і методи.....	18
2.1 Методи класифікації для визначення успішності вивчення онлайн курсів	18
2.2 Метод прогнозування успішності вивчення онлайн курсу студентом .	21
2.3 Висновки за розділом 2	23
3 Проєктування програмного забезпечення	24
3.1 Вибір засобів веброзробки	24
3.2 Моделювання роботи користувачів з програмою	26
3.3 Проєктування бази даних	28
3.4 Структура програмної системи.....	36
3.5 Висновки за розділом 3	37
4 Основні рішення щодо реалізації компонентів системи.....	39
4.1 Алгоритм функціонування програми.....	39
4.2 Опис модулів реалізованої програмної системи.....	41
4.3 Висновки за розділом 4	46
5 Експлуатація, тестування та експериментальне дослідження програми	47
5.1 Призначення програми	47
5.2 Умови виконання програми	47
5.3 Виконання програми.....	47
5.4 Експериментальне дослідження	51
5.5 Висновки за розділом 5	51

Висновки	52
Перелік джерел посилання	54
Додаток А Технічне завдання	56
Додаток Б Текст програми	60
Додаток В Слайди презентації.....	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

KNN – k-nearest neighbor;

НМПП – нейронна мережа прямого поширення.

ВСТУП

Навчання все більше переміщується в Інтернет і базується на онлайн засобах. Онлайн курси є популярним інструментом отримання знань.

Онлайн-навчання продемонструвало значне зростання за останнє десятиліття, оскільки Інтернет та освіта поєднуються, щоб надати людям можливість отримати нові навички, до того ж після спалаху епідемії COVID-19 онлайн-навчання стало відігравати ще більш центральну роль у житті людей, так пандемія змусила школи, університети та компанії перейти на дистанційну роботу, що призвело до зростання використання засобів онлайн-навчання, при цьому ще до пандемії Research and Markets прогнозували, що ринок онлайн-освіти становитиме 350 мільярдів доларів до 2025 року, відповідно ці цифри можуть бути оновлені після аналізу впливу COVID-19 на зростання ринку онлайн-навчання [1].

Онлайн курси є гнучким засобом навчання, адже вони не вимагають вступу в університет для цього, не вимагають вивчення затвердженого набору дисциплін. При цьому кожен користувач відповідного програмного забезпечення може скористатися за потреби наявними на платформі онлайн курсами і розширити свої знання в потрібній області. Такі курси можуть бути безкоштовними, платними.

Онлайн курси сприяють впровадженню інформаційних технологій і інтелектуальних засобів у процес навчання. Однак не зважаючи на дуже активний розвиток даної сфери, слід констатувати, що вона тільки розвивається і є ще багато не вирішених питань. Однією з невирішених на даний момент проблем є прогнозування успішності навчання студента на ранніх етапах, щоби вплинути на цей процес ефективним шляхом для підтримки студентів. Розуміючи, що в даній конкретній ситуації студент швидше за все не зможе завершити курс успішно, викладачі можуть акцентувати на ньому увагу і шукати потрібні для нього ресурси. У такому випадку, наприклад, можуть використовуватися додаткові матеріали для

вивчення. Можуть надаватися додаткові відеоматеріали, текстові матеріали, які дозволять розглянути незрозумілі питання студенту додатково.

Тому метою роботи є розроблення програмного забезпечення вивчення онлайн курсів для підтримки студентів на ранніх етапах.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд сучасного стану проблеми онлайн навчання

Електронне навчання – це формалізована система навчання за допомогою електронних ресурсів, при цьому використання Інтернету та комп'ютерів є важливими компонентами електронного навчання, які дозволяють проводити навчання в класі або поза ним, окрім того електронне навчання також можна описати як мережеву передачу навичок і знань багатьом одержувачам одночасно [2].

Онлайн навчання є частиною електронного навчання і передбачає навчання саме в Інтернеті, тобто в такому випадку не передбачається використання інших засобів окрім електронних ресурсів.

До переваг онлайн навчання належать:

- свобода вивчати будь-який курс: можна підписатися на будь-який курс за бажанням, до того ж передумови проходження курсів достатньо спрощені, так не потрібно бути фахівцем з науки чи бізнесу, щоб пройти поглиблений курс фізики, онлайн навчання дозволяє збільшити різноманітність у тому, який варіант продовження кар'єри обрати, на відміну від традиційного навчання, так розширюються напрями навчання, до того ж багато університетів і коледжів розширюють сфери навчання, дозволяючи студентам проходити онлайн курси, у свою чергу заклади освіти виявили, що, надаючи студентам широкий спектр курсів, вони збільшують різноманітність студентів, які вивчають ці курси;

- гнучкість: люди, які працюють повний робочий день, можуть отримати освіту, не звільняючись з роботи, окрім гнучкого графіку ще й не потрібно використовувати власний час на поїздку на навчання;

- можливості для самостійного навчання: онлайн навчання дуже зручне людям, які найкраще навчаються у своєму власному темпі, при цьому немає тиску стосовно того, щоб не відставати від однокласників, можна займатися стільки, скільки потрібно, виконуючи завдання, це може бути зручним

інструментом не тільки для тих, хто мислить повільніше, але і для тих, хто хоче не тільки вивчити, але і розібратися, перш ніж рухатися далі, окрім того можна робити у власному темпі перерви у навчанні;

- зручність для студентів з обмеженими можливостями: у випадку онлайн навчання такі студенти мають більше можливостей, ніж будь-коли раніше, здобути освіту;

- комфорт навчання вдома: у світі, де люди постійно використовують свої телефони і ноутбуки, не дивно, що онлайн навчання стає все більш популярним, завдяки онлайн курсам можна навчатися, не виходячи з дому, при цьому вивчати можна все, що бажає студент, коли захоче, не турбуючись ні про що інше;

- розвиток технічних навичок: онлайн курси чудово підходять для людей, які хочуть навчитися новим технічним навичкам у сучасному сучасному світі, технології відіграють все більш важливу роль у нашому житті, і студентам важливо знати, як ефективно використовувати технології, онлайн навчання дає можливість ознайомитися з новими технологіями, а також отримати диплом або працювати над сертифікатом [3].

На даний момент за даними LinkedIn 90% корпоративних компаній пропонують можливості електронного навчання співробітникам [4].

Результатом всіх останніх подій у світі є те, що 33% вищих навчальних закладів вказують, що продовжуватимуть пропонувати свої курси частково або повністю онлайн після пандемії [4].

1.2 Огляд програмних аналогів

Для того щоб зрозуміти наявний інструментар у сфері онлайн навчання, потрібно проаналізувати засоби, які надаються програмними аналогами розроблюваної в роботі програмної системи.

Найвідомішим на даний момент аналогом виконуваної розробки в Україні є Prometheus (рис. 1.1) [5].

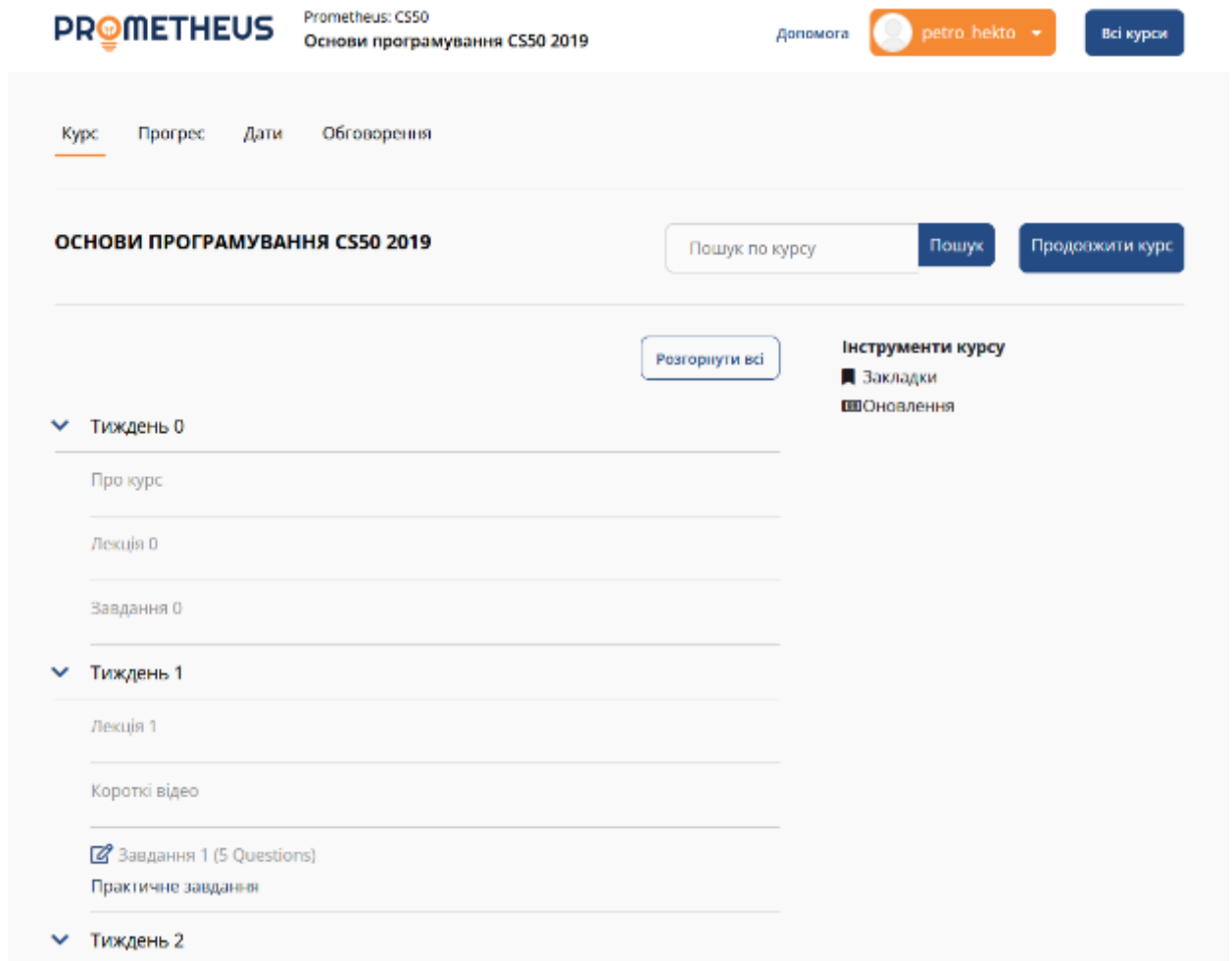


Рисунок 1.1 – Інтерфейс перегляду курсу на платформі Prometheus

На платформі Prometheus розташовуються онлайн курси, створені спеціально для цієї платформи, а також відомі в світі курси, перекладені українською.

Система дозволяє виконувати пошук нових курсів з наявної бази, підписуватися на їх проходження, в межах оформленої підписки переглядати надані матеріали для вивчення, проходити тести. При цьому як в Prometheus, так і в інших системах можливість створення курсів не надається кожному користувачу, тобто базовий сценарій роботи передбачає роботу студента. Викладачі працюють з системою на основі іншого набору сценаріїв.

Міжнародна система Coursera (рис. 1.2) надає доступ до авторських курсів в тому числі від відомих світових університетів. Coursera пропонує різні

навчальні програми, починаючи від окремих курсів до отримання спеціальностей та ступенів [6].

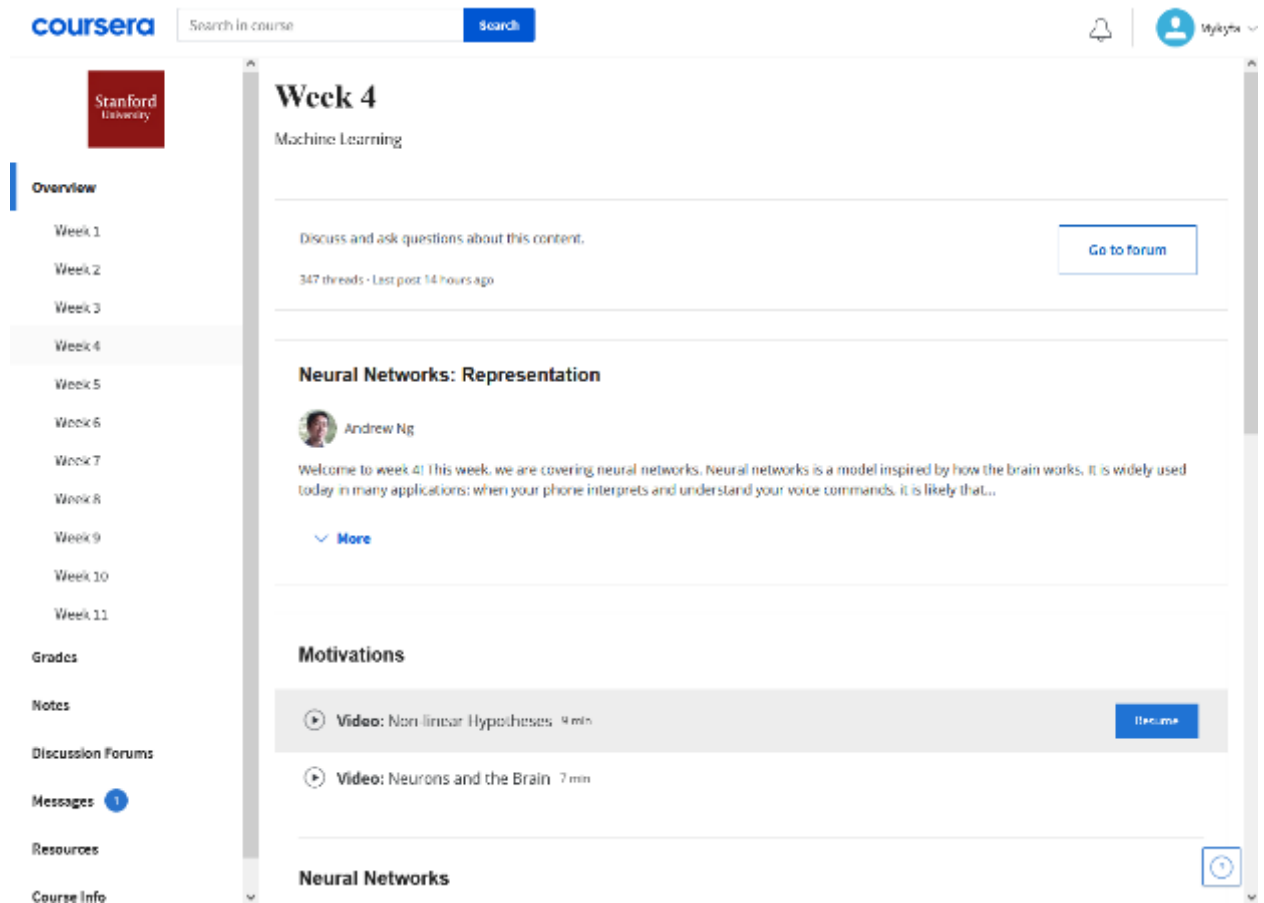


Рисунок 1.2 – Інтерфейс перегляду курсу на платформі Coursera

Система Coursera передбачає окремий інтерфейс для представників підприємств. Для багатьох курсів передбачається формат безкоштовного проходження і платного у разі необхідності отримання відповідних офіційних підтверджуючих документів. Загалом на платформі доступно приблизно 2000 безкоштовних курсів.

Важливо відзначити, що Coursera окрім підтримки роботи звичайних бажаючих навчатися за курсами, якими можуть стати будь-які бажаючі, надає окремі наступні варіанти роботи:

- підсистема Coursera для бізнесу: окрім індивідуального навчання, Coursera для бізнесу пропонує план Enterprise для великих організацій, при цьому надається необмежений доступ до курсів платформи, показників

навичок, аналітики користувачів та інтеграції для досягнення успіху, на даний момент Coursera працює з понад 15 000 підприємствами;

- підсистема Coursera для команд: як і Coursera для бізнесу, цей варіант для команд, який орієнтований на меншу кількість працівників, його розроблено, щоб надати команді доступ до понад 4000 курсів, спеціалізацій та керованих проєктів, цей план розрахований на оплату за одного користувача протягом 1 року, і гарантується 14-денна політика відшкодування;

- підсистема Coursera для кампуса: Coursera для кампуса спрямована на те, щоб надати будь-якому університету можливість запропонувати онлайн освіту, яка відповідає вакансіям, студентам, викладачам і співробітникам, крім того Coursera пропонує цей план безкоштовно для студентів, які бажають безкоштовно опанувати навички, готові до роботи, цей план дає доступ до необмеженої кількості керованих проєктів і 1 безкоштовного курсу на рік [6].

1.3 Визначення сутностей предметної області

Виконавши аналіз основних функціональних можливостей, які надаються подібними програмними продуктами, і є важливими для реалізації нової програмної системи, яка може використовуватися для створення окремого навчального центру в межах підприємства, було визначено основні сутності предметної області, до яких віднесено:

- Course – онлайн курс;
- Module – модуль цього онлайн курсу;
- Topic – тема онлайн курсу, що існує в межах модулю;
- Group – група, в яку об'єднуються онлайн курси;
- Material – матеріал, що визначений для вивчення теми модулю;
- Task – завдання, визначене в межах теми модулю, для відповіді на яке потрібно вислати результат у вигляді окремого файлу, який потім перевірить викладач і виставить оцінку;

- Test – тест, який потрібно пройти в межах вивчення теми (фінальний тест може бути відділений окремо в разі, якщо він стосується всього курсу загалом, тоді він виноситься в окрему тему);
- TestTask – запитання в межах тесту;
- TestTaskAnswer – відповідь на запитання в межах тесту;
- Account – обліковий запис користувача програми, який може бути викладачем або студентом;
- Student – студент (повинен містити дані саме про особливості навчання студента);
- Subscription – підписка студента на вивчення онлайн курсу;
- Comment – запитання, відповідь або коментар на онлайн курс, створений студентом або викладачем;
- TopicResult – результат вивчення теми за онлайн курсом, який визначає, чи було вивчено основний матеріал теми (переглянуто відео);
- MaterialResult – результат вивчення матеріала курсу студентом (необхідний для того, щоб визначати, чи переглянув статтю або якесь інше посилання студент під час вивчення курсу);
- TestResult – результат проходження тесту студентом;
- TestTaskResult – відповіді на запитання тесту студентом;
- TaskResult – результат виконання завдання за онлайн курсом студентом.

1.4 Висновки за розділом 1 та постановка завдань дослідження

Розглянуто основні характеристики сучасного стану проблеми онлайн навчання. Розглянуто основні програмні аналоги програмної системи, що розробляється, до яких віднесено Coursera та Prometheus. Система, що створюється, повинна бути направлена на підтримку навчання в обособлених групах. Це можуть бути окремі підприємства, курси в межах класів для забезпечення розмаїття формальної освіти тощо.

Завданнями кваліфікаційної роботи є:

- створити новий метод прогнозування успішності вивчення онлайн курсу студентом;
- виконати проєктування програмного забезпечення;
- реалізувати програмне забезпечення;
- виконати експериментальне дослідження проблеми.

2 МАТЕРІАЛИ І МЕТОДИ

Прогнозування успішності вивчення онлайн курсів є задачею персоналізації навчання. Отримуючи результати її розв'язання, викладачі можуть реагувати на поточний стан навчання, сприяючи досягненню успішних результатів. Таким чином цей підхід може застосовуватися для змінювання курсу під потреби студента. При чому не використання або пропущення певних матеріалів, а накопичення бази матеріалів, яка має відкоригувати підготовлений курс для найкращого впровадження на практиці.

Задача прогнозування успішності вивчення онлайн курсу студентом є задачею класифікації. Тому в даному розділі необхідно спочатку розглянути методи класифікації, які можуть використовуватися для вирішення цієї проблеми, а на основі вивченого досвіду створити власний метод.

2.1 Методи класифікації для визначення успішності вивчення онлайн курсів

До основних методів, які на даний момент використовуються на практиці для розв'язання задач класифікації, належать:

- метод k найближчих сусідів або k -nearest neighbor (KNN);
- нейронна мережа прямого поширення (НМПП).

Метод KNN – це контрольований навчальний і непараметричний алгоритм, який можна використовувати для вирішення як класифікаційних, так і регресійних задач:

- використовує дані, в яких присутня цільова ознака, тобто дані з мітками, щоб моделювати функцію, щоб створити вихід для невидимих даних, він використовує формулу евклідової відстані для обчислення відстані між точками даних для класифікації або передбачення, основна мета цього алгоритму полягає в тому, що подібні точки даних повинні бути близькими

одна до одної, тому він використовує відстань для обчислення подібних точок, які знаходяться близько одна до одної;

- непараметричність алгоритму означає відсутність будь-яких припущень щодо розподілу базових даних, ці методи не мають фіксованого числа параметрів у моделі, аналогічно в KNN параметри моделі зростають разом із навчальними даними, розглядаючи кожен навчальний випадок як параметр моделі, тому KNN є непараметричним алгоритмом;

- значення K в даному методі представляє кількість найближчих сусідів, які потрібно вибрати, щоб передбачити клас даного елемента, який надходить як невидимий набір даних для моделі [7]-[8].

Відносно побудови НМПП можна виділити наступні основні характеристики:

- кожен вхід до штучного нейрона має пов'язану з ним вагу, при цьому вхідні дані спочатку помножуються на відповідні ваги, а до результату додається зміщення, тому результат попередньої роботи окремого нейрона можна назвати зваженою сумою, потім зважена сума проходить через функцію активації, яка в основному є нелінійною функцією, таким чином, штучний нейрон можна розглядати як просту або множинну модель лінійної регресії з функцією активації в кінці;

- нейронна мережа зазвичай має декілька шарів, кожен з яких має декілька нейронів, де всі нейрони одного шару з'єднані з усіма нейронами наступного шару тощо: перший шар – це вхідний шар, який містить деяку кількість нейронів, що визначається даними, які подаються як вхід до нейронної мережі, останній шар є вихідним шаром, кількість нейронів в останньому шарі та першому шарі заздалегідь визначається набором даних та типом проблеми (кількістю вихідних класів тощо);

- нейрон із поточного рівня бере вихідні дані всіх нейронів із попереднього шару як вхідні дані та обчислює зважену суму, додає до неї зміщення, а потім відправляє отримане значення через функцію активації;

– перший нейрон з першого прихованого шару буде підключений до всіх входів з попереднього шару (вхідного шару), аналогічно другий нейрон з першого прихованого шару також буде підключений до всіх входів з попереднього шару, і так далі для всіх нейронів першого прихованого шару;

– для нейронів другого прихованого шару вихідні дані попереднього прихованого шару вважаються входами, і кожен із цих нейронів так само з'єднаний з усіма попередніми нейронами;

– ваги нейронів під час навчання зазвичай ініціалізуються випадковим чином, але в подальшому вони досягають свого оптимального значення, щоб зменшити функцію втрат [9]-[10].

Основні методи, які використовуються для визначення успішності вивчення онлайн курсів на даний момент, зведені до табл. 2.1.

Таблиця 2.1 – Порівняння існуючих методів визначення успішності вивчення онлайн курсів

Метод	Основа оцінки	Модель	Етап застосування	Вихідна ознака
Метод [11]	Опитувальник студентів	НМПП	До початку навчання	Клас успішності
Метод [12]	Загальні характеристики і поточні результати навчання	KNN	У процесі навчання	Клас оцінки
Метод [13]	Загальні характеристики	Дерева рішень	До початку навчання	Клас успішності
Пропонований метод	Загальні і поточні результати навчання	НМПП	Перший етап навчання	Клас успішності

2.2 Метод прогнозування успішності вивчення онлайн курсу студентом

Метод прогнозування успішності вивчення онлайн курсу студентом повинен застосовуватися на ранніх етапах вивчення курсу, коли студент вже отримав перші результати. Такими першими результатами є перші отримані оцінки. Наприклад, у випадку 4 модулів, це може бути оцінка за перший модуль. Окрім того це може бути оцінка за проміжний тест або виконане завдання. Після отримання такої оцінки можна виконати прогнозування, за яким приймати рішення про роботу зі студентом на наступних етапах навчання.

Важливим для реалізації методу прогнозування успішності вивчення онлайн курсу студентом є якісна підготовка вибірки даних, яка має використовуватися для навчання та тестування моделі прогнозування успішності вивчення онлайн курсу студентом.

В якості ознак, що використовуються для побудови моделі, було виділено:

- поточний рівень освіти студента;
- вік студента;
- кількість курсів, яку студент вивчав на даний момент, – курси, за якими було отримано оцінку, тобто час вивчення яких минув або за якими було отримано підсумкову оцінку;
- кількість курсів, яку студент успішно завершив;
- отримана оцінка за проходження курсу, для якого відбувається визначення успішності, на поточний момент, – проміжна оцінка за модуль, тест або завдання, яке є попереднім етапом вивчення курсу, за яким вирішено виконувати прогноз;
- вага отриманої оцінки – кількість балів, яку максимально міг отримати студент на даний момент;

- відсоток вивчених матеріалів – кількість матеріалів за курсом (відеоматеріалів, текстових матеріалів, посилань, завдань, тестів), яку вивчив або виконав студент на даний момент, відносно кількості матеріалів, яку мав би вивчити студент до цього моменту;

- кількість інших курсів, які вивчає студент в цей же момент.

Перші дві ознаки характеризуються студента загалом. Наступні дві ознаки визначають те, як студент працював на всіма попередніми курсами в межах системи. Останні ознаки визначають поточний рівень вивчення даного курсу студентом.

Вибірка даних у свою чергу має складатися з екземплярів, які відповідають значенням перелічених ознак, а також класу успішності. Клас успішності може бути мати значення або 0, або 1. Одиниця відповідає успіху за результатами навчання, тобто коли студент набрав кількість балів за курс, що дорівнює або перевищує встановленому порогу за курсом. Якщо ж студент набрав меншу кількість балів, то йому встановлюється клас успішності 0.

Сформовану таким чином вибірку даних потрібно на наступному етапі передобробити, для чого реалізується заповнення пропущених значень (нульовими значеннями або значеннями за замовчуванням) або відкидання певних екземплярів у випадку відсутності необхідних ознак (наприклад, отримана оцінка або її вага), нормалізація отриманих даних.

Подальшим етапом є створення моделі на основі багатошарової НМПП.

У відповідності з описаним набором ознак НМПП повинна містити 8 нейронів на вхідному шарі для отримання значень вхідних ознак, що і було реалізовано в процесі експериментального дослідження в розділі 5.

У процесі подальшого дослідження запропоновано використовувати НМПП з 2 прихованими шарами. Для досягнення кращих результатів навчання потрібно використовувати глибокі алгоритми навчання.

За результатами створення моделі відбувається її навчання за передобробленою навчальною вибіркою даних з оцінюванням результатів навчання за передобробленою тестувальною вибіркою даних.

Після цього навчена модель може використовуватися для класифікації траєкторій навчання студентів за отриманими значеннями ознак, описаних для першого етапу. Значення мають так само передоброблятися, як було описано для другого етапу.

2.3 Висновки за розділом 2

Виконано аналіз методів класифікації даних, які використовуються для прогнозування успішності вивчення онлайн курсу студентом. У роботі запропоновано власний метод прогнозування успішності вивчення онлайн курсу студентом, який виконує прогнозування шляхом вирішення задачі класифікації на основі застосування багат шарової нейронної мережі прямого поширення, що дозволяє приймати рішення щодо необхідності додаткової підтримки вивчення курсів студентами на ранніх етапах.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір засобів веброзробки

Основний вибір засобів веброзробки зводився до вибору фреймворку, який має забезпечити якісну і зручну роботу щодо реалізації саме вебзастосунку. Двома популярними на даний момент фреймворками є ASP.NET, який працює в основному з мовою C#, та Node.js (табл. 3.1) [14]-[16].

Таблиця 3.1 – Вибір фреймворку розробки

Характеристика	ASP.NET	Node.js
Розробка сучасних вебзастосунків	Співмірно	Співмірно
Підтримка сучасних бібліотек машинного навчання	Співмірно	Співмірно
Рівень безпеки	Краще	
Зручність масштабування	Дещо краще	
Якість коду	Краще	
Мови програмування	C# (можливо F#)	JavaScript

Порівняння фреймворків виконувалось на основі наступних характеристик:

- розробка сучасних вебзастосунків: важливо, щоб створена система була зручною у використанні, відповідала основним можливостям, які повинні мати сучасні вебзастосунки, адже без цього використовувати таку систему в даній області користувачі не будуть, за даною характеристикою обидва фреймворки дозволяють досягти високих результатів;

- підтримка сучасних бібліотек машинного навчання: реалізації нейронної мережі прямого поширення є дуже важливим елементом побудови програмної системи, адже на цій моделі ґрунтується представлений метод, а він по суті забезпечує основну функціональну особливість програми, що є

дуже важливим, при цьому за даною характеристикою порівняння необхідно зазначити, що у обидвох фреймворків є можливість використання прикладних інтерфейсів для роботи з бібліотекою Tensorflow, тому за двома першими характеристиками, які є базовими, забезпечується повна відповідність створюваних рішень;

- рівень безпеки: за цією характеристикою немає просідань у фреймворків, але C#, платформа .NET і фреймворк ASP.NET більше мають засобів підтримки безпеки в застосунках (зокрема автоматизовані засоби авторизації, захист від атак);

- зручність масштабування: зі збільшенням кількості користувачів, які працюють з системою, важливо забезпечити якісну підтримку їх роботи, а якщо вебсервер і система загалом не зможуть це зробити відповідним чином, то користувачі не будуть використовувати її і знайдуть інший варіант організації платформи, тому за даним показником слід відзначити, що загалом обидва фреймворки придатні до масштабування, але зважаючи на те, що ASP.NET надає можливість створювати багатопотокові застосунки, то це в перспективі може принести користь для застосунку, коли він отримає широке практичне впровадження і буде працювати під великим навантаженням;

- якість коду: в основі C# традиційна статична типізація, що призводить до того, що помилки виявляються швидше і не відбувається накопичення, з таким кодом простіше працювати в довгій перспективі.

Використання мови C# в основі фреймворку ASP.NET також дозволяє зважати на те, що на основі виконаної реалізації предметної області можна побудувати застосунок для операційної системи Windows, не маючи необхідності реалізовувати все наново. Тому в результаті розробку було вирішено вести мовою C# на основі фреймворку ASP.NET. Відповідно системою керування базами даних обрано Microsoft SQL Server, що є базовим інструментом у цьому стеку технологій.

3.2 Моделювання роботи користувачів з програмою

Для моделювання роботи користувачів з програмою на основі виконаного огляду програмних аналогів визначено перелік функціональних вимог до програми:

- перегляд найрейтинговіших онлайн курсів;
- перегляд курсів за автором;
- перегляд найновіших курсів;
- пошук курсів;
- визначення тільки тих курсів, на які можна підписатися;
- публікація курсів;
- перегляд курсів за групою;
- комунікація щодо онлайн курсу;
- перегляд онлайн курсу;
- перегляд курсів, на які підписався студент;
- деактивація підписки на курс;
- оцінювання курсу студентом;
- визначення викладачів, які працюють з курсом;
- структуризація курсу;
- оформлення підписки на курс;
- відслідковування перегляду теми студентом;
- виконання завдання за курсом;
- перевірювання завдання викладачем;
- проходження тесту за курсом;
- керування авторами своїми курсами;
- обчислення оцінки студенту за курс;
- отримання аналітики за курсом, що включає відсоток успішного проходження курсу, відсоток неуспішного проходження курсу, відсоток студентів, які приступили до вивчення курсу.

На основі визначених функціональних вимог визначено сценарії роботи користувачів з програмою.

Користувачів було розділено на два окремі групи: викладачі і студенти. Вони передбачають окрему роботу з програмою. Моделювання роботи їх з програмою окремо виконувалось для забезпечення підтримки безпеки роботи з програмою. Кожен з них повинен мати свій інтерфейс і свої сценарії.

Студенту надаються сценарії навчання за курсом та пошуку курсів для підписки на навчання (рис. 3.1).

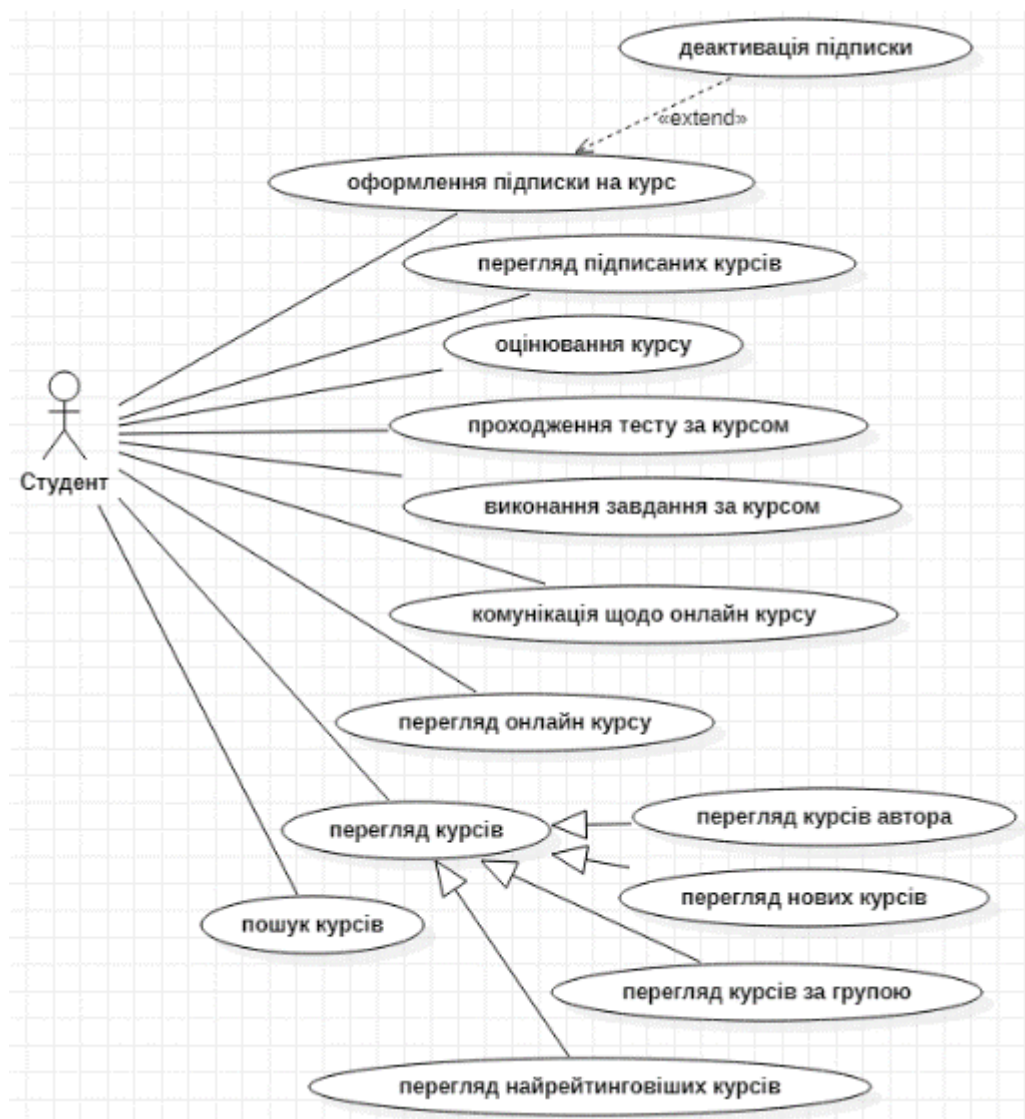


Рисунок 3.1 – Діаграма прецедентів студента

Викладачу надаються сценарії пошуку курсів на платформі та роботи з власними курсами (рис. 3.2).

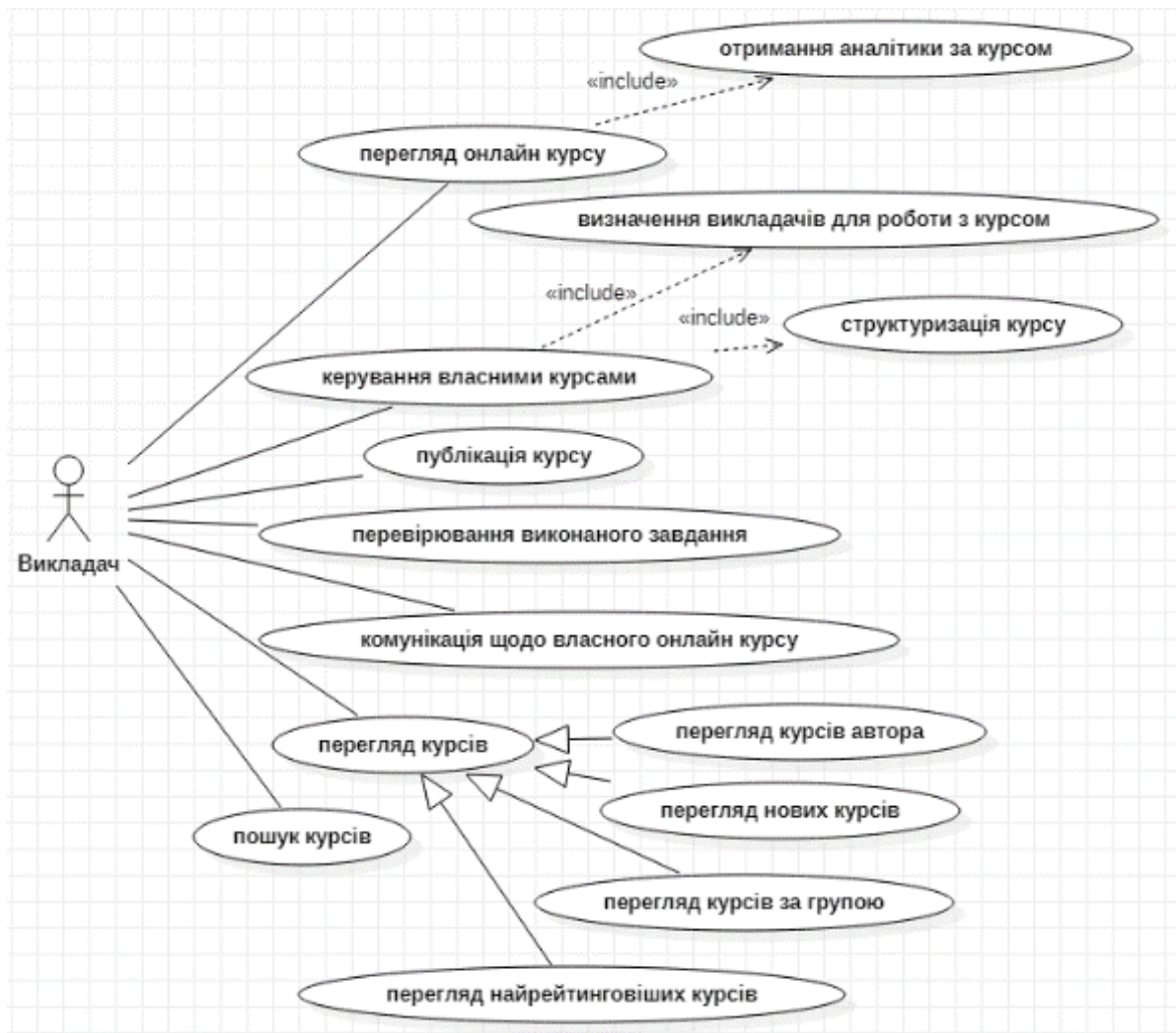


Рисунок 3.2 – Діаграма викладача

3.3 Проєктування бази даних

Використовуючи визначені в розділі 1 основні сутності предметної області, було виконано їх проєктування для визначення структури бази даних.

Таблиця Group містить наступні поля:

- GroupId – номер групи;
- Title – назва групи.

Таблиця Course містить наступні поля:

- CourseId – номер онлайн курсу;
- GroupId – група, до якої належить цей онлайн курс;
- Title – назва онлайн курсу;
- Description – повний опис онлайн курсу;
- Prerequisite – передумови проходження онлайн курсу;
- Cover – зображення, яке представляє онлайн курсу;
- Published – дата публікації онлайн курсу;
- Score – середня оцінка онлайн курсу;
- Subscribers – кількість підписників на цей онлайн курс;
- MinimumScore – мінімальна оцінка, за якої проходження онлайн курсу буде вважатися успішним;

- StartDate – дата, з якої онлайн курс буде доступним;
- EndDate – дата, до якої онлайн курс буде доступним.

Таблиця Module містить наступні поля:

- ModuleId – номер модуля;
- CourseId – онлайн курс, до якого належить модуль;
- Order – порядок цього модуля серед всіх модулів онлайн курсу;
- Title – назва модуля.

Поле CourseId є зовнішнім ключем з таблиці Course і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця Topic містить наступні поля:

- TopicId – номер теми;
- ModuleId – модуль, до якого входить ця тема;
- Order – порядок теми серед всіх тем, завдань і тестів модуля;
- Title – формулювання теми;
- Video – відеоматеріал за темою;
- Description – опис теми (текстова інформація до неї);
- Published – дата публікації теми;
- StartDate – дата, з якої ця тема в онлайн курсі буде доступною;
- EndDate – дата, до якої ця тема в онлайн курсі буде доступною.

Поле `ModuleId` є зовнішнім ключем з таблиці `Module` і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця `Material` містить наступні поля:

- `MaterialId` – номер матеріалу;
- `Source` – джерело матеріалу;
- `Title` – назва матеріалу;
- `TopicId` – тема, до якої належить цей матеріал;
- `Published` – дата публікації матеріалу;
- `StartDate` – дата, з якої матеріал в онлайн курсі буде доступним;
- `EndDate` – дата, до якої матеріал в онлайн курсі буде доступним.

Поле `TopicId` є зовнішнім ключем з таблиці `Topic` і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця `Task` містить наступні поля:

- `TaskId` – номер завдання;
- `ModuleId` – модуль, до якого належить завдання;
- `Description` – текст завдання;
- `Published` – дата публікації завдання;
- `StartDate` – дата, з якої завдання в онлайн курсі буде доступним;
- `EndDate` – дата, до якої завдання в онлайн курсі буде доступним;
- `Order` – порядок завдання серед всіх тем, завдань і тестів модуля;
- `Score` – максимальна оцінка, яку можна отримати за це завдання;
- `CourseAuthorId` – викладач, який може перевіряти це завдання.

Поле `ModuleId` є зовнішнім ключем з таблиці `Module` і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле `CourseAuthorId` є зовнішнім ключем з таблиці `CourseAuthor` і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця `Test` містить наступні поля:

- `TestId` – номер тесту;
- `Order` – порядок тесту серед всіх тем, завдань і тестів модуля;
- `ModuleId` – модуль, до якого належить цей тест;

- StartDate – дата, з якої тест в онлайн курсі буде доступним;
- EndDate – дата, до якої тест в онлайн курсі буде доступним;
- Type – тип тесту (проміжний, фінальний).

Поле ModuleId є зовнішнім ключем з таблиці Module і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця TestTask містить наступні поля:

- TestTaskId – номер завдання в тесті;
- TaskQuestion – текст завдання;
- Order – порядок завдання серед всіх завдань тесту;
- TestId – тест, до якого належить завдання;
- Score – оцінка, яку можна отримати за правильну відповідь на завдання тесту.

Поле TestId є зовнішнім ключем з таблиці Test і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця TestTaskAnswer містить наступні поля:

- TestTaskAnswerId – номер відповіді на завдання тесту;
- TestTaskId – завдання, до якого належить відповідь;
- TaskAnswer – текст відповіді;
- Right – поле, що визначає правильність відповіді на запитання.

Поле TestTaskId є зовнішнім ключем з таблиці TestTask і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця Account містить наступні поля:

- AccountId – номер облікового запису;
- Login – логін облікового запису;
- Password – пароль облікового запису;
- Description – опис користувачу (обов'язково заповнюється викладачем і може заповнюватися студентом);
- Photo – фотографія облікового запису;
- Role – роль користувача у програмі.

Таблиця Student містить наступні поля:

- AccountId – номер облікового запису студента;
- BirthDay – день народження студента;
- Education – рівень освіти студента на поточний момент;
- CoursesSubscribed – кількість курсів, на вивчення яких підписувався студент загалом;
- CoursesSuccess – кількість курсів, яку загалом успішно завершив студент.

Поле AccountId є зовнішнім ключем з таблиці Account і визначає зв'язок типу «один-до-одного», проведений до цієї таблиці.

Таблиця Subscription містить наступні поля:

- SubscriptionId – номер підписки студента на онлайн курс;
- CourseId – онлайн курс, на який підписався студент;
- AccountId – номер облікового запису студента;
- Score – загальна оцінка студента за вивчення цього онлайн курсу;
- SubDate – дата підписки на проходження онлайн курсу;
- EndDate – дата завершення проходження онлайн курсу (дата, коли був пройдений останній елемент курсу, зазвичай – фінальний тест);
- Rating – оцінка, виставлена студентом курсу;
- Active – поле, яке визначає, чи активною є підписка;
- Success – поле, яке визначає, чи успішно було завершено вивчення онлайн курсу;
- PredictedSuccess – поле, яке визначає результат прогнозованого визначення успішності вивчення курсу студентом;
- Learned – частка матеріалів курсу, яку опрацював студент на даний момент.

Поле CourseId є зовнішнім ключем з таблиці Course і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле AccountId є зовнішнім ключем з таблиці Account і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця CourseAuthor містить наступні поля:

- CourseAuthorId – номер;
- CourseId – онлайн курс, який веде цей викладач;
- AccountId – викладач, який веде курс;
- Role – роль викладача в курсі (автор чи просто учасник).

Поле CourseId є зовнішнім ключем з таблиці Course і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле AccountId є зовнішнім ключем з таблиці Account і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця Comment містить наступні поля:

- CommentId – номер коментаря;
- CourseId – онлайн курс, до якого належить це запитання або коментар;
- AccountId – користувач, який залишив коментар;
- CommentText – текст коментаря;
- Published – дата публікації коментаря.

Поле CourseId є зовнішнім ключем з таблиці Course і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле AccountId є зовнішнім ключем з таблиці Account і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця TopicResult містить наступні поля:

- TopicResultId – результат проходження онлайн курсу;
- TopicId – тема, до якої належить результат;
- SubscriptionId – підписка на онлайн курс, в межах якої досягнуто цей результат;
- VideoPublished – дата перегляду відео студентом;
- Published – дата отримання цього результату.

Поле TopicId є зовнішнім ключем з таблиці Topic і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле SubscriptionId є зовнішнім ключем з таблиці Subscription і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця TestResult містить наступні поля:

- TestResultId – результат проходження тесту;
- SubscriptionId – підписка на онлайн курс, за якою було отримано цей результат тесту;
- TestId – тест, результат якого отримано;
- Published – дата проходження тесту;
- Score – оцінка за тест.

Поле TestId є зовнішнім ключем з таблиці Test і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле SubscriptionId є зовнішнім ключем з таблиці Subscription і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця TestTaskResult містить наступні поля:

- TestTaskResultId – номер створеної відповіді на запитання тесту;
- TestTaskAnswerId – відповідь на запитання, обрана студентом;
- SubscriptionId – підписка на онлайн курс, в межах якої обрано цю відповідь;
- Published – дата внесення відповіді на запитання тесту;
- Score – оцінка, що відповідає цій відповіді.

Поле TestTaskAnswerId є зовнішнім ключем з таблиці TestTaskAnswer і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле SubscriptionId є зовнішнім ключем з таблиці Subscription і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Таблиця MaterialResult містить наступні поля (рис. 3.3):

- MaterialResultId – номер перегляду матеріалу;
- SubscriptionId – підписка на онлайн курс, за якою було переглянуто матеріал онлайн курсу;
- MaterialId – матеріал курсу, який було переглянуто;
- Published – дата проходження тесту.

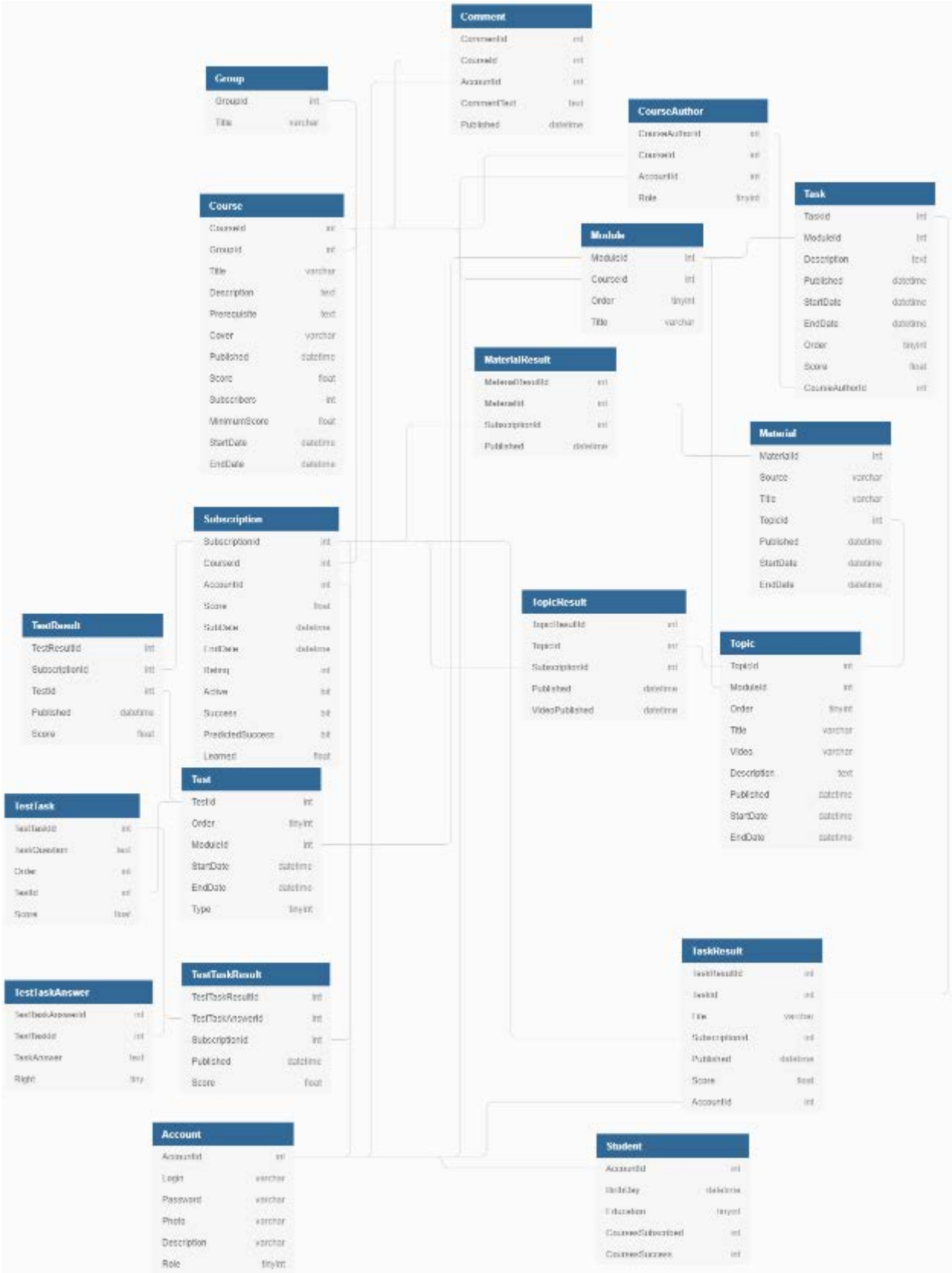


Рисунок 3.3 – Схема бази даних

Таблиця TaskResult містить наступні поля:
– TaskResultId – номер результату виконання завдання;

- TaskId – завдання, до якого належить цей результат;
- File – файл, завантажений студентом;
- SubscriptionId – підписка на онлайн курс, за якою виконано завдання;
- Published – дата публікації завдання;
- Score – оцінка за виконання завдання;
- AccountId – викладач, який перевіряв завдання.

Поле TaskId є зовнішнім ключем з таблиці Task і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле AccountId є зовнішнім ключем з таблиці Account і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

Поле SubscriptionId є зовнішнім ключем з таблиці Subscription і визначає зв'язок типу «один-до-багатьох», проведений до цієї таблиці.

3.4 Структура програмної системи

В основі розробленої структури програмної системи (рис. 3.4) лежить спільно використовувана бібліотека, яка може використовуватися для моделювання даної предметної області і реалізації методу прогнозування успішності вивчення онлайн курсу студентом.

Моделювання предметної області виконується на основі реалізованих моделей застосунку та контексту бази даних DataBaseContext і міграцій Entity Framework, які забезпечують підтримку створення і збереження стану бази даних. Ці засоби взаємодіють напямую з базою даних Microsoft SQL Server і вони створюють інфраструктуру програмної системи.

Класи, які реалізують метод прогнозування успішності вивчення онлайн курсу студентом, виокремлені в підсистему LearningSuccessPredictor. Ця підсистема через спільно використовувану бібліотеку взаємодії з Keras.NET API, який забезпечує програмний інтерфейс до бібліотеки Tensorflow, необхідної для реалізації машинного навчання.

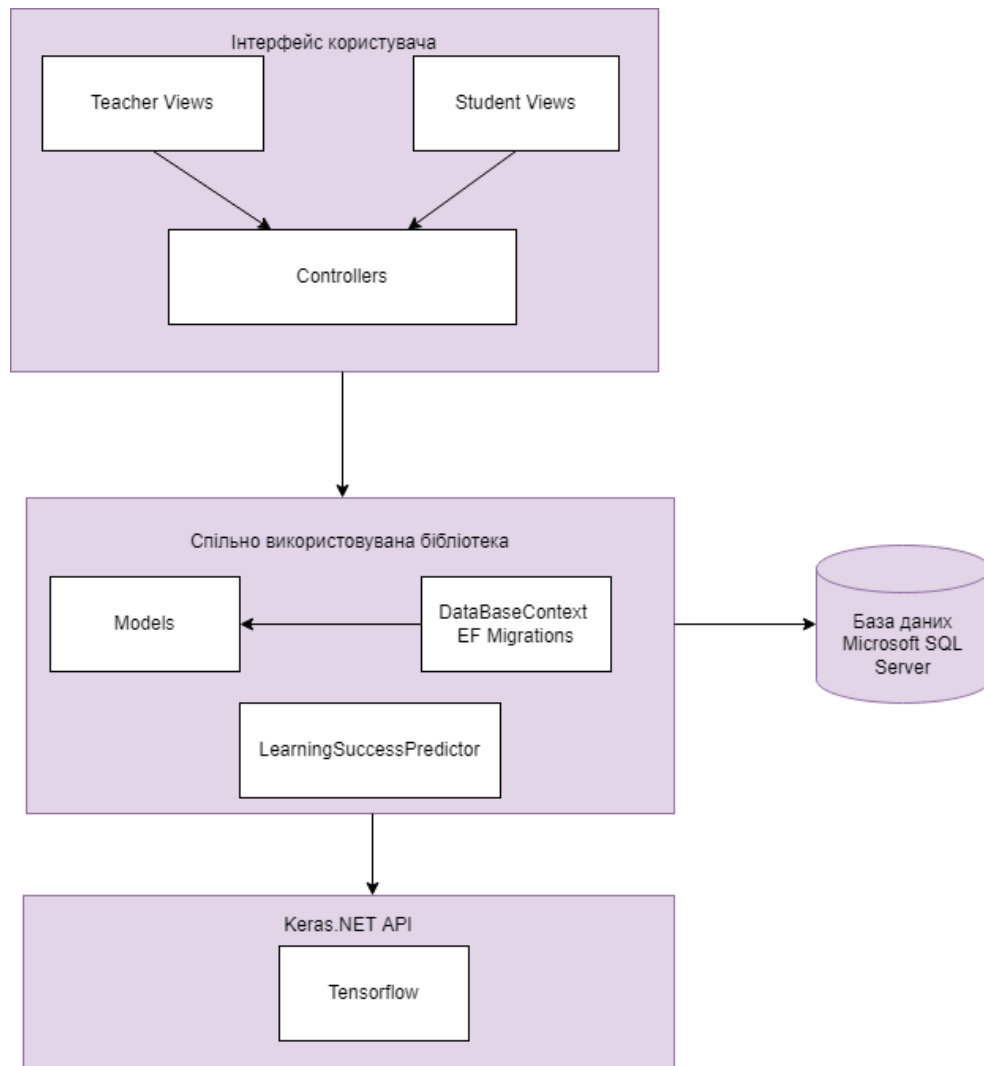


Рисунок 3.4 – Структурна схема програмної системи

На основі спільно використовуваної бібліотеки побудований інтерфейс користувача. В основі інтерфейсу користувача лежать створені контролери вебзастосунку. Саме контролери безпосередньо взаємодіють зі спільно використовуваною бібліотекою. Усі уявлення вебзастосунку розділені на уявлення викладача та уявлення студента. І уявлення викладача, і уявлення студента взаємодіють з контролерами вебзастосунку.

3.5 Висновки за розділом 3

Розробку вебзастосунку вирішено виконувати за допомогою фреймворку ASP.NET, мови програмування C#, системи управління базами

даних Microsoft SQL Server. Проєктування програми виконано за допомогою створених діаграм прецедентів викладача і студента. Структуру бази даних побудовано на основі виділених раніше сутностей предметної області. Структуру програмної системи запропоновано побудувати з виділенням спільно використовуваної бібліотеки та засобів реалізації інтерфейсу користувача.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Алгоритм функціонування програми

Алгоритм функціонування програми побудований на двох окремих гілках, перша з яких призначена для роботи викладачів, а інша – для роботи студентів (рис. 4.1). Для роботи вебзастосунку він спочатку має бути запущений на сервері, а далі користувач має звернутися до відповідного вебінтерфейсу.

Якщо користувач звернувся до вебінтерфейсу викладача, то йому відкривається форма авторизації. Після проходження авторизації викладач може переглядати курси, доступні на платформі, якщо йому потрібно вивчити досвід інших викладачів або якщо він хоче переглянути, як виглядає його курс для інших викладачів. При цьому він не може підписатися на проходження курсу, для цього йому потрібно завести окремий обліковий запис студента. Окрім того викладач може переглянути власні курси викладача, звідки він може або опублікувати новий курс, або обрати існуючий власний курс. Обравши один з таких курсів, викладач може відредагувати його опис, перевірити подані студентами завдання, або працювати над матеріалами, тестами, завданнями, модулями, темами курсу, комунікувати зі студентами, переглядати результати навчання студентів, які підписалися на його проходження, а також переглядати загалом дані власного курсу.

Якщо користувач звернувся до вебінтерфейсу студента, то йому відтворюється список найрейтинговіших курсів. Після цього студент може обрати один з варіантів доступу до списку курсів на платформі: переглянути найрейтинговіші курси, найновіші курси, виконати пошук курсів, переглянути курси за групою або виконувати авторизацію. Після авторизації студент може скористатися одним з двох варіантів роботи – з курсами на платформі або над оформленими підписками.

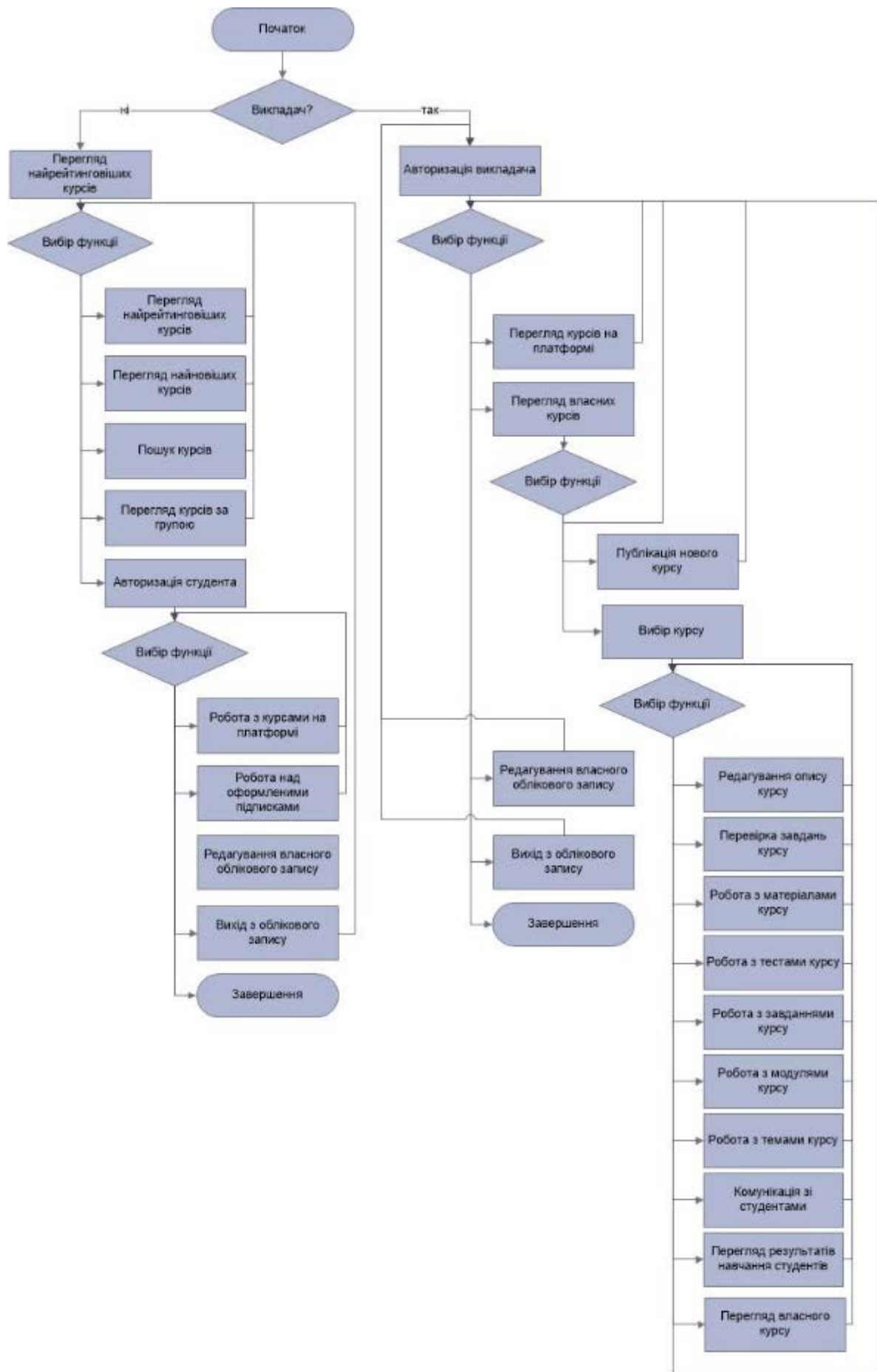


Рисунок 4.1 – Схема функціонування програми

4.2 Опис модулів реалізованої програмної системи

Класи моделей створені для забезпечення підтримки аналітичних засобів роботи з курсами.

Клас Course надає такі атрибути:

- int CourseId – унікальний номер онлайн курсу;
- string Title – назва онлайн курсу;
- string Description – повний опис онлайн курсу;
- string Prerequisite – передумови вивчення курсу;
- string Cover – зображення для опису курсу;
- DateTime Published – дата публікації курсу;
- float Score – середня оцінка, виставлена студентами курсу;
- float MinimumScore – мінімальна кількість балів, яку необхідно отримати для успішного проходження курсу;
- DateTime StartDate – дата, з якої починається вивчення курсу;
- DateTime EndDate – дата, до якої має бути завершено вивчення курсу;
- int GroupId – номер групи, до якої належить курс;
- virtual Group Group – повні дані групи курсів з номером GroupId;
- virtual ICollection<Topic> Topics – теми, які вивчаються за цим курсом;
- virtual ICollection<Subscription> Subscriptions – підписки студентів на вивчення цього курсу;
- virtual ICollection<CourseAuthor> CourseAuthors – автори і викладачі, відповідальні за цей курс;
- virtual ICollection<Comment> Comments – запитання і коментарі студентів і викладачів за курсом;
- ICollection<Topic> OrderedTopics – теми курсу, впорядковані за значенням Order кожної теми;

- `ICollection<Topic> OrderedComments` – запитання, коментарі і відповіді викладачів і студентів за курсом, впорядковані за часом створення, починаючи з останніх;
- `ICollection<Topic> OrderedAuthors` – викладачі курсу, впорядковані за їх роллю (спочатку автори, потім викладачі);
- `string UnSuccessLevel` – відсоток студентів, які неуспішно завершили вивчення курсу, відносно всіх студентів, для яких вивчення курсу мало завершитися на поточний момент;
- `string SuccessLevel` – відсоток студентів, які успішно завершили вивчення курсу, відносно всіх студентів, для яких вивчення курсу мало завершитися на поточний момент;
- `string TextStartDate` – текстове представлення дати, з якої починається вивчення курсу;
- `string TextEndDate` – текстове представлення дати, до якої має бути завершено вивчення курсу;
- `string TextPublished` – текстове представлення дати публікації курсу;
- `int Subscribers` – сумарна кількість підписників на вивчення курсу;
- `string TextScore` – середня оцінка, виставлена студентами курсу, в текстовій формі.

Відповідно на основі описаного набору атрибутів викладач може отримати аналітику щодо вивчення курсу різними студентами, і відповідним чином реагувати на цю аналітику, змінюючи за потреби курс або залишаючи його таким самим.

Контролером, який базується на використанні класів моделей `Course` і `CourseAuthor`, є клас `CourseController`, що має такий набір методів:

- `ActionResult BestCourses(int n, bool all)` – метод, який звертається до уявлення, заповнюючи його списком онлайн курсів з найвищим рейтингом, параметри впливають на створення списку наступним чином: `n` – кількість курсів, яку необхідно повернути (наприклад, 10 найкращих курсів), істинне

значення параметра `all` вказує на те, що потрібно повернути всі курси, а хибне – на те, що повертаються тільки курси, строк вивчення яких ще не завершився;

- `ActionResult CourseComments(int n)` – метод, який звертається до уявлення, заповнюючи його коментарями, запитаннями і відповідями щодо курсу з номером `n`;

- `ActionResult NewCourses(int n, bool all)` – метод, який звертається до уявлення, заповнюючи його списком останніх опублікованих курсів, параметри впливають на створення списку наступним чином: `n` – кількість курсів, яку необхідно повернути, істинне значення параметра `all` вказує на те, що потрібно повернути всі курси, а хибне – на те, що повертаються тільки курси, строк вивчення яких ще не завершився;

- `ActionResult AuthorCourses(int n, int m)` – метод, який звертається до уявлення, заповнюючи його списком курсів, автором яких є викладач з номером облікового рахунку `n`, повертаючи в результаті тільки `m` останніх опублікованих курсів;

- `ActionResult Details(int n)` – метод, який звертається до уявлення, заповнюючи його даними онлайн курсу з номером `n`;

- `ActionResult AuthorDetails(int n)` – метод, який звертається до уявлення, заповнюючи його даними онлайн курсу з номером `n` і додаючи аналітичні дані, що стосуються викладача (рівень успішного вивчення, рівень неуспішного вивчення курсу);

- `ActionResult SearchCourses(string name, bool all)` – метод, який звертається до уявлення, заповнюючи його результатами пошуку курсу, до назви якого входить рядок `name`, при цьому істинне значення параметра `all` вказує на те, що потрібно повернути всі курси, а хибне – на те, що повертаються тільки курси, строк вивчення яких ще не завершився;

- `ActionResult Create()` – метод, який обробляє GET-форму публікації нового курсу;

- `ActionResult Create([Bind("Title, Description, Prerequisite, Cover, MinimumScore, GroupId")] Course course, string sdate, string edate)` – метод, який

обробляє POST-форму публікації курсу, створюючи курс з назвою Title, описом Description, передумовами вивчення Prerequisite, зображенням Cover, мінімальним балом для отримання успішного результату MinimumScore, номером групи GroupId;

- ActionResult Edit(int? n) – метод, який обробляє GET-форму редагування курсу з номером n;

- ActionResult Edit(int n, [Bind("CourseId, Title, Description, Prerequisite, Cover, MinimumScore, GroupId")] Course course, string sdate, string edate) – метод, який обробляє POST-форму редагування курсу з номером CourseId, оновлюючи в базі даних інформацію про курс, включаючи його назву Title, опис Description, передумови вивчення Prerequisite, зображення Cover, мінімальний бал для отримання успішного результату MinimumScore, номер групи GroupId;

- ActionResult GroupCourses(int n, int m) – метод, який звертається до уявлення, заповнюючи його списком курсів, які належать до групи з номером n, повертаючи в результаті тільки m найкращих курсів за цією групою.

Для реалізації вебзастосунку було створено уявлення за допомогою вільного шаблону [17]:

- newcourse.cshtml – уявлення публікації нового онлайн курсу;
- managecourse.cshtml – уявлення редагування онлайн курсу;
- bestcourses.cshtml – уявлення перегляду найрейтинговіших курсів;
- courseforum.cshtml – уявлення комунікації між студентами і викладачами щодо курсу;
- newcourses.cshtml – уявлення перегляду останніх за часом публікації курсів;
- authorcourses.cshtml – уявлення перегляду курсів автора або доступу з можливістю керування до власних курсів;
- course.cshtml – уявлення вебсторінки онлайн курсу;
- detcourse.cshtml – уявлення вебсторінки онлайн курсу для викладача;
- search.cshtml – уявлення пошуку курсів;

- groupcourses.cshtml – уявлення перегляду курсів, що належать заданій групі;
- subscriptions.cshtml – уявлення перегляду студентом курсів, на які він підписався;
- coursesubscriptions.cshtml – уявлення перегляду підписок на курс для роботи викладача зі студентами;
- owncourses.cshtml – уявлення перегляду власних курсів викладачем як автором з можливістю їх редагування;
- topic.cshtml – уявлення перегляду інформації за темою курсу;
- results.cshtml – уявлення роботи викладача з результатами студента за курсом;
- courseauthors.cshtml – уявлення керування переліком викладачів, що працюють з курсом;
- task.cshtml – уявлення роботи студента з завданням курсу;
- test.cshtml – уявлення доступу студента до проходження тесту за курсом;
- testtask.cshtml – уявлення для перегляду і відповіді студентом на запитання тесту;
- checktaskresults.cshtml – уявлення для перевірки завдання, виконаного студентом за курсом, викладачем;
- newtask.cshtml – уявлення створення нового завдання за курсом;
- newmodule.cshtml – уявлення створення нового модуля за курсом;
- newmaterial.cshtml – уявлення створення нового матеріалу за курсом;
- newtest.cshtml – уявлення створення нового тесту за курсом;
- managetask.cshtml – уявлення редагування модуля за курсом;
- managemodule.cshtml – уявлення редагування даних модулю;
- managematerial.cshtml – уявлення редагування даних матеріалу;
- managetest.cshtml – уявлення редагування тесту;
- tasks.cshtml – уявлення перегляду всіх завдань за курсом з можливістю доступу до їх редагування;

- modules.cshtml – уявлення перегляду всіх модулів за курсом з можливістю доступу до їх редагування;
- topics.cshtml – уявлення перегляду всіх тем курсу з можливістю доступу до їх редагування;
- tests.cshtml – уявлення перегляду всіх тестів курсу з можливістю доступу до їх редагування;
- login.cshtml – уявлення авторизації користувача в програмі;
- register.cshtml – уявлення реєстрації студента.

4.3 Висновки за розділом 4

У четвертому розділі представлено результати реалізації програми. Розроблені класи реалізують сутності предметної області, необхідні для підтримки створення бази даних, функціональні можливості предметної області. Реалізовані уявлення визначають користувацький інтерфейс і забезпечують взаємодію користувача з тими функціональними можливостями, які йому надаються.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Призначення програми

Програма призначена для вивчення онлайн курсів. Для різних сторін даного процесу програма може бути призначена для:

- використання її авторами курсів для публікації власних онлайн курсів і підтримки викладання цих курсів студентам;
- використання її студентами для вивчення онлайн курсів.

5.2 Умови виконання програми

Для виконання програми на сервері повинні забезпечуватися наступні вимоги:

- процесор: чотири ядра, тактова частота – 2,9 ГГц;
- оперативна пам'ять: 10 Гб;
- жорсткий диск: 300 Мб доступного простору.

5.3 Виконання програми

Під час тестування програми було перевірено виконання всіх функціональних вимог програмою.

Для звернення до програми потрібно задати в браузері адресу, за якою розташовано інтерфейс студента.

Приклад відображення доступних курсів у вигляді рейтингу цих курсів для авторизованого студента представлено на рис. 5.1. Курси відображаються через меню «Курси». На цій сторінці доступне переключення між режимами відображення списків курсів: «Рейтинг», «Нові», «Групи», «Пошук». Для цих позицій доступний прапорець «доступні для підписки», який дозволяє

відобразити або всі курси за критерієм, або тільки ті, на які ще за часом можна оформити підписку.

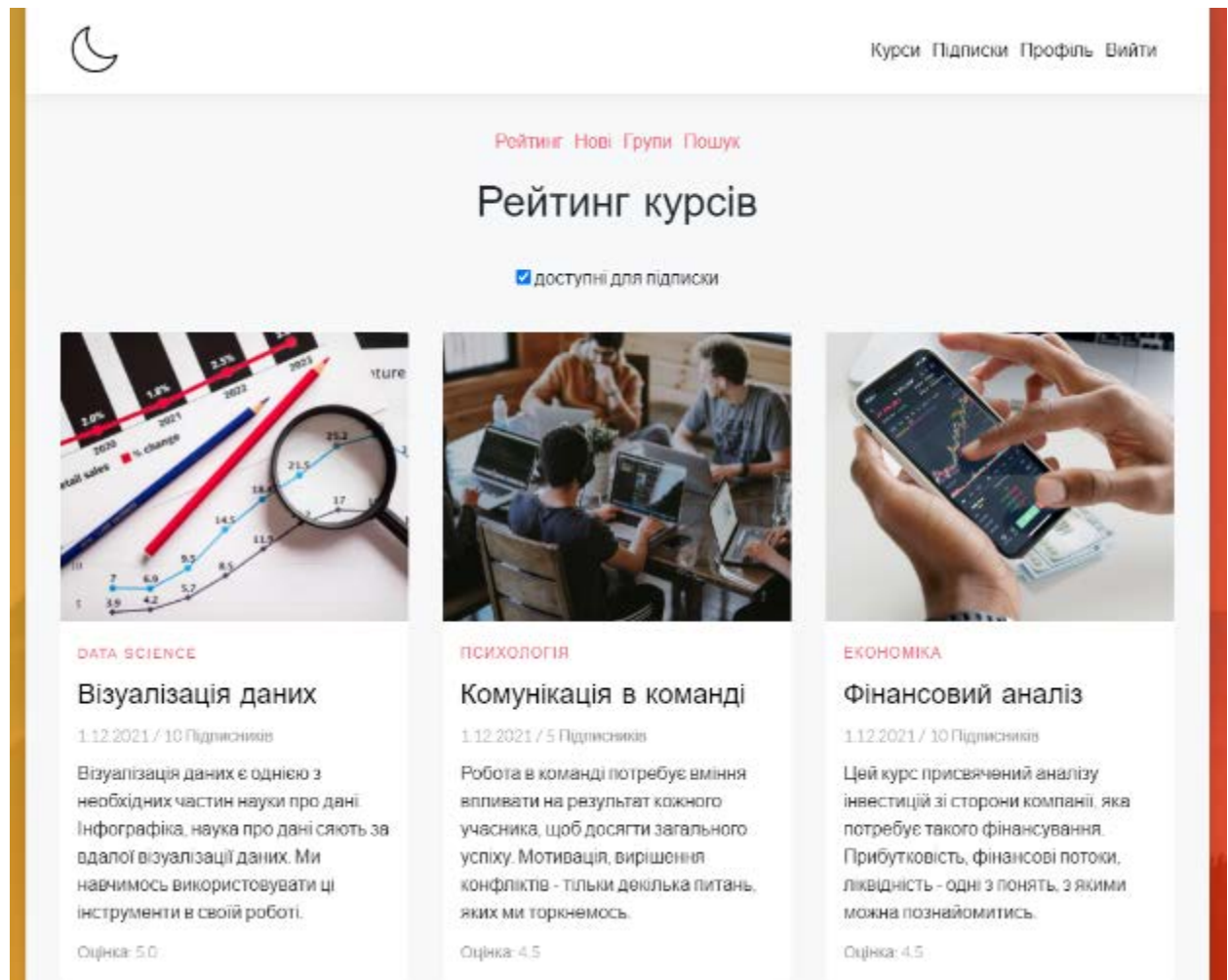


Рисунок 5.1 – Перегляд найрейтинговіших курсів

За кожним курсом відображається його зображення, назву, групу, опис (перший абзац опису), середню оцінку, виставлену студентами. Якщо натиснути на назву курсу або зображення, то відкриється сторінка курсу. На цій сторінці можна переглянути опис та оформити підписку.

Після оформлення підписки можна з цієї сторінки отримати також перелік тем, завдань, тестів. Якщо перейти на сторінку окремої теми, то буде відображено відеоматеріал, назву теми, її текстовий опис, список матеріалів (рис. 5.2). Можна перейти за посиланням для перегляду матеріалу.



Візуалізація даних



Тема 1: Вступ до візуалізації даних

Володіти даними в сучасному світі недостатньо, оскільки цих даних дуже багато. Важливо вміти їх аналізувати.

Аналіз даних - це основа прийняття рішень.

Візуалізація даних - це крок до аналізу даних. Якщо зробити його ефективно, то рішення з'являться перед очима і не будуть потребувати додаткових кроків.

Доступно до: 12.12.2021

Матеріали

Приклади візуалізації

До 12.12.2021

Текст лекції

До 11.12.2021

Рисунок 5.2 – Перегляд однієї з тем онлайн курсів

У випадку відкриття тесту відображається сторінка з першим запитанням (рис. 5.3). Потрібно обрати відповідь, натиснути на кнопку «Наступне запитання» або «Попереднє запитання» для попереднього запитання, або «Завершити тест» для останнього запитання.

Викладачу доступ надається через окремий інтерфейс. Список курсів викладача доступний через меню «Кабінет». Зі сторінки курсу можна перейти до списку підписок, де відображаються всі студенти (рис. 5.4).

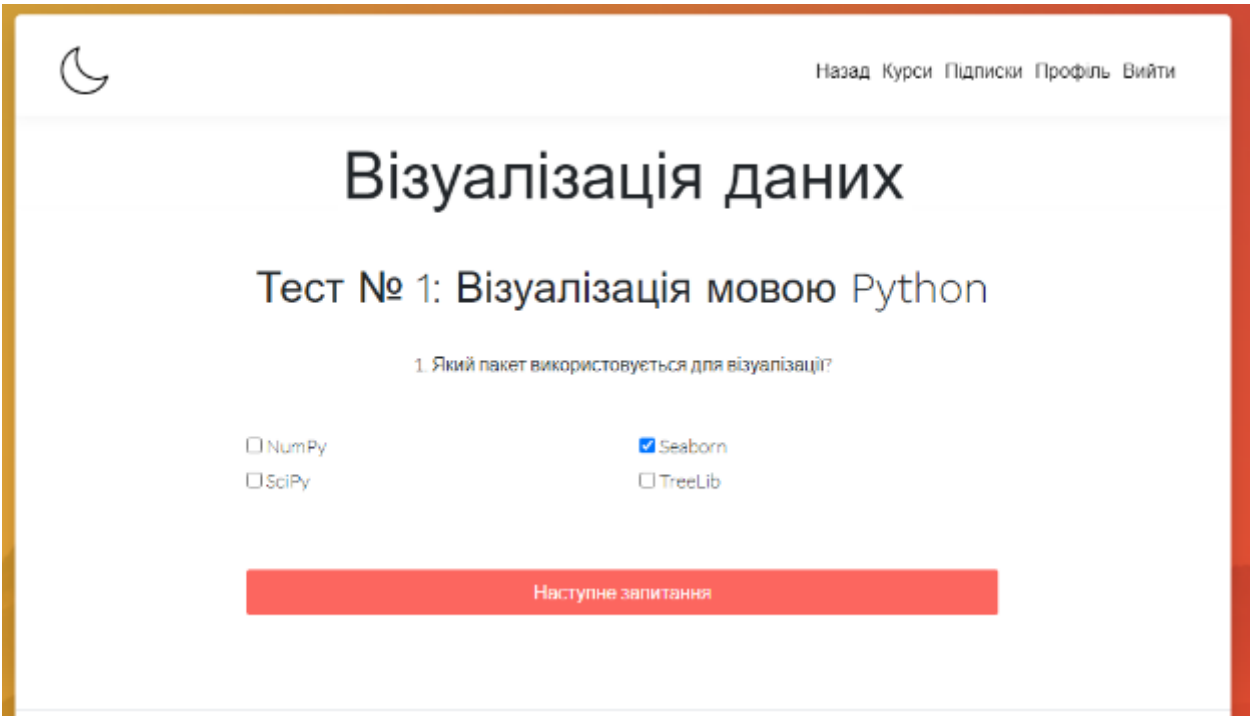


Рисунок 5.3 – Проходження тесту за курсом

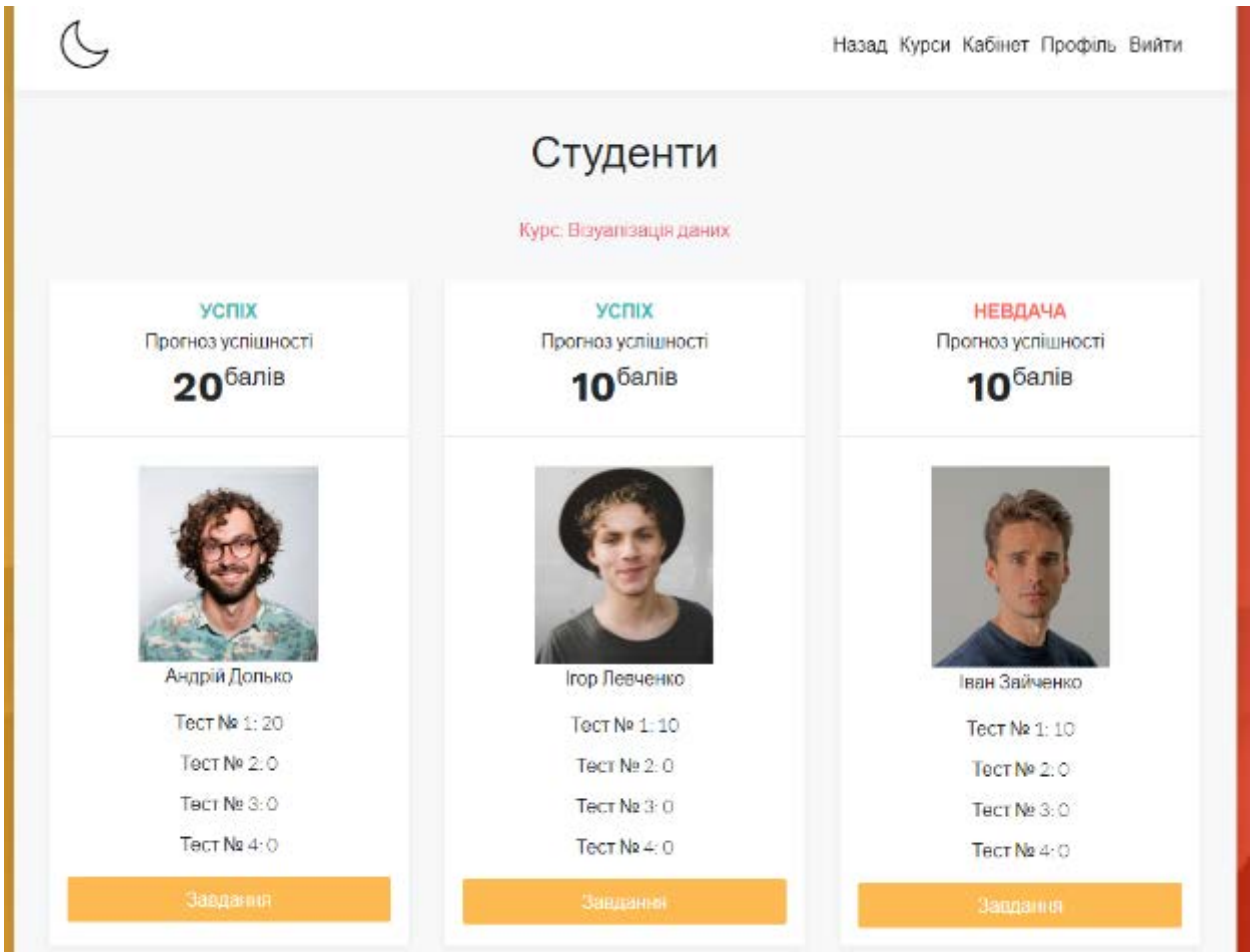


Рисунок 5.4 – Керування роботою зі студентами

Для кожного студента відображається прогнозований результат успішності вивчення курсу. Звідси можна перейти до перевірки завдання.

5.4 Експериментальне дослідження

Експериментальне дослідження виконано за допомогою вибірки [18], за якою було сформовано вибірку даних, що відповідає пропонованому методу. Відповідні результати експериментального дослідження представлені в табл. 5.1.

Таблиця 5.1 – Результати експериментального дослідження

Метод	Точність результатів
KNN	0,67
Пропонований метод	0,84

Метод KNN призвів до отримання точності прогнозування у 0,67 на сформованій вибірці, а пропонований метод – 0,84. Таким чином, застосування моделі на основі НМПП виявився кращим при вирішенні даної задачі.

5.5 Висновки за розділом 5

Проведено тестування роботи програмного забезпечення з приведенням основних інтерфейсів програми, отриманими за результатами тестування. Виконані дії підтвердили реалізацію всіх запланованих функцій програми. Експериментальне дослідження підтвердило покращення результатів класифікації за використання пропонованого методу.

ВИСНОВКИ

За результатами проведеного аналізу предметної області виділено основні сутності предметної області і визначено функціональні особливості програми.

Для вибору засобів проектування і розробки було проаналізовано фреймворки ASP.NET та Node.js. Розробку вебзастосунку вирішено виконувати за допомогою фреймворку ASP.NET, мови програмування C#, системи управління базами даних Microsoft SQL Server. Проектування програми виконано за допомогою створених діаграм прецедентів викладача і студента. Структуру бази даних визначено у відповідності з виділеними сутностями предметної області. Структуру програмної системи запропоновано побудувати з виділенням спільно використовуваної бібліотеки, що дозволить використовувати результати проектування предметної області для вирішення подібних задач за потреби або при розширенні системи, та засобів реалізації інтерфейсу користувача.

Реалізовано програмне забезпечення. Розроблені класи реалізують сутності предметної області, необхідні для підтримки створення бази даних, функціональні можливості предметної області. Реалізовані уявлення визначають користувацький інтерфейс і забезпечують взаємодію користувача з тими функціональними можливостями, які йому надаються.

Розроблений програмний застосунок призначений для вивчення онлайн курсів і надає викладачам засоби роботи авторів курсів для публікації власних онлайн курсів і підтримки викладання цих курсів студентам, а студентам – засоби вивчення онлайн курсів і роботи з матеріалами цих курсів. У межах курсів підтримуються модулі, теми, матеріали, у тому числі відеоматеріали, завдання і тести.

Науковою новизною роботи є запропонований метод прогнозування успішності вивчення онлайн курсу студентом, який виконує прогнозування шляхом вирішення задачі класифікації на основі застосування багатосарової

нейронної мережі прямого поширення, що дозволяє приймати рішення щодо необхідності додаткової підтримки вивчення курсів студентами на ранніх етапах.

Завдання кваліфікаційної роботи виконані повністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Koksai, I. The Rise Of Online Learning [Electronic resource] / I. Koksai. – Access mode : <https://www.forbes.com/sites/ilkerkoksai/2020/05/02/the-rise-of-online-learning/?sh=3704f8d72f3c>.
2. Steward, J. The Ultimate List of eLearning Statistics for 2022 [Electronic resource] / J. Steward. – Access mode : <https://findstack.com/elearning-statistics/>.
3. Brown, A. Here's Why Online Learning is Popular and Here to Stay [Electronic resource] / A. Brown. – Access mode : <https://techdayhq.com/community/articles/here-s-why-online-learning-is-popular-and-here-to-stay>.
4. Moran, M. 30+ eLearning Statistics 2021 [Electronic resource] : Facts, Trends, Demographics, And More / M. Moran. – Access mode : <https://startupbonsai.com/elearning-statistics/>.
5. Найкращі онлайн-курси України та світу [Електрон. ресурс]. – Режим доступу : <https://prometheus.org.ua/>.
6. Cooke, C. Coursera Review [Electronic resource] : How Popular Is This Elearning Platform? / C. Cooke. – Access mode : <https://upskillwise.com/reviews/coursera/>.
7. Goyal, C. 20 Questions to Test your Skills on KNN Algorithm [Electronic resource] / C. Goyal. – Access mode : <https://www.analyticsvidhya.com/blog/2021/05/20-questions-to-test-your-skills-on-k-nearest-neighbour/>.
8. Implementation of K Nearest Neighbors [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/implementation-k-nearest-neighbors/>.
9. Aggarwal, C. Neural Networks and Deep Learning [Text] : A Textbook / C. Aggarwal. – Berlin : Springer, 2018. – 520 p.
10. Prasad, A. Feed Forward Neural Networks. Intuition on Forward

Propagation [Electronic resource] / A. Prasad. – Access mode : <https://medium.com/analytics-vidhya/feed-forward-neural-networks-intuition-on-forward-propagation-f77468fad625>.

11. Hamoud, A. K. Student's Success Prediction Model Based on Artificial Neural Networks (ANN) and A Combination of Feature Selection Methods [Electronic resource] / A. K. Hamoud, A. M. Humadi // Journal of Southwest Jiaotong University. – 2019. – Volume 54, No. 3. – Access mode : <http://jsju.org/index.php/journal/article/view/305>.

12. Emirtekin, E. Online Course Success Prediction of Students with Machine Learning Methods [Text] / E. Emirtekin, M. Karatay, T. Kışla // Journal of Modern Technology and Engineering. – 2020. – Vol. 5, No. 3. – Pp. 271-282.

13. Kovacic, Z. J. Early Prediction of Student Success : Mining Students Enrolment Data [Electronic resource] / Z. J. Kovacic // Proceedings of Informing Science & IT Education Conference (InSIT) 2010. – Access mode : <https://proceedings.informingscience.org/InSITE2010/InSITE10p647-665Kovacic873.pdf>.

14. ASP.NET Core. Полное руководство [Электрон. ресурс]. – Режим доступа : <https://metanit.com/sharp/aspnet5/>.

15. Lock, A. ASP.NET Core in Action [Text] / A. Lock. – New York : Manning, 2021. – 832 p.

16. Zammetti, F. Modern Full-Stack Development [Text] : Using TypeScript, React, Node.js, Webpack, and Docker / F. Zammetti. – New York : Apress, 2020. – 395 p.

17. Moon Free Website Template. Free CSS Templates. Free CSS [Electronic resource]. – Access mode : <https://www.free-css.com/free-css-templates/page273/moon>.

18. Open Learning Analytics. OU Analyse. Knowledge Media Institute. The Open University [Electronic resource]. – Access mode : https://analyse.kmi.open.ac.uk/open_dataset.

ДОДАТОК А
Технічне завдання

Вступ

Програма має задовольняти вимогам усіх сторін навчального процесу. Для підтримки цього процесу мають надаватися аналітичні засоби щодо вивчення курсів студентами для викладачів та засоби прогнозування успішності вивчення курсів студентами на ранньому етапі.

A.1 Підстави для розробки

Підставою для розробки є завдання на кваліфікаційну роботу магістра, визначене за темою «Дослідження та програмна реалізація методів підтримки організації вивчення онлайн курсів», яку затверджено наказом № 435 від 5 листопада 2021 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Розробка, яка виконується, призначена для вивчення онлайн курсів. Для різних сторін даного процесу розробка може бути призначена для:

- використання її авторами курсів для публікації власних онлайн курсів і підтримки викладання цих курсів студентам;
- використання її студентами для вивчення онлайн курсів.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Функціональні характеристики програмного забезпечення повинні включати:

- перегляд найрейтинговіших онлайн курсів;
- перегляд курсів за автором;
- перегляд найновіших курсів;

- пошук курсів;
- визначення тільки тих курсів, на які можна підписатися;
- публікація курсів;
- перегляд курсів за групою;
- комунікація щодо онлайн курсу;
- перегляд онлайн курсу;
- перегляд курсів, на які підписався студент;
- деактивація підписки на курс;
- оцінювання курсу студентом;
- визначення викладачів, які працюють з курсом;
- структуризація курсу;
- оформлення підписки на курс;
- відслідковування перегляду теми студентом;
- виконання завдання за курсом;
- перевірювання завдання викладачем;
- проходження тесту за курсом;
- керування авторами своїми курсами;
- обчислення оцінки студенту за курс;
- отримання аналітики за курсом, що включає відсоток успішного проходження курсу, відсоток неуспішного проходження курсу, відсоток студентів, які приступили до вивчення курсу.

А.3.2 Вимоги до складу та параметрів технічних засобів

Апаратне забезпечення роботи вебсервера повинно відповідати вимогам роботи з щонайменше 200 користувачами одночасно. Для підтримки цих вимог апаратне забезпечення повинно мінімально включати:

- процесор: чотири ядра, тактова частота – 2,9 ГГц;
- оперативна пам'ять: 10 Гб;
- жорсткий диск: 300 Мб доступного простору.

А.4 Порядок контролю та приймання

Календарний план є основою для контролю виконання роботи, у свою чергу приймання роботи здійснюється під час її захисту перед екзаменаційною комісією.

ДОДАТОК Б
Текст програми

Б.1 Текст файла Course.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace OnlineEducator.Models
{
    public class Course
    {
        [Key]
        public int CourseId {get; set;}
        public string Title {get; set;}
        public string Description {get; set;}
        public string Prerequisite {get; set;}
        public string Cover {get; set;}
        public DateTime Published {get; set;}
        public float Score {get; set;}
        public float MinimumScore {get; set;}
        public DateTime StartDate {get; set;}
        public DateTime EndDate {get; set;}
        public int GroupId {get; set;}
        public virtual Group Group {get; set;}
        public virtual ICollection<Topic> Topics {get; set;}
        public virtual ICollection<Subscription> Subscriptions {get; set;}
        public virtual ICollection<CourseAuthor> CourseAuthors {get; set;}
        public virtual ICollection<Comment> Comments {get; set;}
        [NotMapped]
        public ICollection<Topic> OrderedTopics
        {
```

```

    get
    {
        return this.Topics.OrderByDescending(t => t.Order);
    }
}

[NotMapped]
public ICollection<Topic> OrderedComments
{
    get
    {
        return this.Comments.OrderByDescending(t => t.Published);
    }
}

[NotMapped]
public ICollection<Topic> OrderedAuthors
{
    get
    {
        var cauthors = this.CourseAuthors;
        cauthors.OrderByDescending(c => c.Role);
        return cauthors;
    }
}

[NotMapped]
public string UnSuccessLevel
{
    get
    {
        DateTime currdate = DateTime.Now;

```

```

        var subscriptions = this.Subscriptions. Where(c =>
c.Course.EndDate <= currdate && c.Success == 0);

        var unsubscriptions = this.Subscriptions. Where(c =>
c.Course.EndDate <= currdate);

        float sl = subscriptions.Count()/unsubscriptions.Count();
        return Math.Round(sl*100, 0).ToString();
    }
}

[NotMapped]
public string SuccessLevel
{
    get
    {
        DateTime currdate = DateTime.Now;

        var subscriptions = this.Subscriptions. Where(c =>
c.Course.EndDate <= currdate && c.Success == 1);

        var unsubscriptions = this.Subscriptions. Where(c =>
c.Course.EndDate <= currdate);

        float sl = subscriptions.Count()/unsubscriptions.Count();
        return Math.Round(sl*100, 0).ToString();
    }
}

[NotMapped]
public string TextStartDate
{
    get
    {
        return this.StartDate.ToString("dd.MM.yyyy");
    }
}

```

```
[NotMapped]
public string TextEndDate
{
    get
    {
        return this.EndDate.ToString("dd.MM.yyyy");
    }
}

[NotMapped]
public string TextPublished
{
    get
    {
        return this.Published.ToString("dd.MM.yyyy");
    }
}

[NotMapped]
public int Subscribers
{
    return this.Subscriptions.Count();
}

[NotMapped]
public string TextScore
{
    get
    {
        return Math.Round(this.Score, 1).ToString();
    }
}
}
```

```
}
```

Б.2 Текст файла Topic.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace OnlineEducator.Models
{
    public class Topic
    {
        [Key]
        public int TopicId {get; set;}
        public string Title {get; set;}
        public string Description {get; set;}
        public DateTime Published {get; set;}
        public DateTime StartDate {get; set;}
        public DateTime EndDate {get; set;}
        public int ModuleId {get; set;}
        public int Order {get; set;}
        public string Video {get; set;}
        public virtual Module Module {get; set;}
        public virtual ICollection<Material> Materials {get; set;}
        [NotMapped]
        public string TextStartDate
        {
            get
            {
                return this.StartDate.ToString("dd.MM.yyyy");
            }
        }
    }
}
```

```

        }
    }
    [NotMapped]
    public string TextEndDate
    {
        get
        {
            return this.EndDate.ToString("dd.MM.yyyy");
        }
    }
    [NotMapped]
    public string TextPublished
    {
        get
        {
            return this.Published.ToString("dd.MM.yyyy");
        }
    }
}

```

Б.3 Текст файла Material.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace OnlineEducator.Models
{
    public class Material

```

```

{
    [Key]
    public int MaterialId {get; set;}
    public string Source {get; set;}
    public string Title {get; set;}
    public int TopicId {get; set;}
    public DateTime Published {get; set;}
    public DateTime StartDate {get; set;}
    public DateTime EndDate {get; set;}
    public virtual Topic Topic {get; set;}
    [NotMapped]
    public string TextStartDate
    {
        get
        {
            return this.StartDate.ToString("dd.MM.yyyy");
        }
    }
    [NotMapped]
    public string TextEndDate
    {
        get
        {
            return this.EndDate.ToString("dd.MM.yyyy");
        }
    }
    [NotMapped]
    public string TextPublished
    {
        get

```

```

        {
            return this.Published.ToString("dd.MM.yyyy");
        }
    }
}

```

Б.4 Текст файла Task.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace OnlineEducator.Models
{
    public class Task
    {
        [Key]
        public int TaskId {get; set;}
        public string Description {get; set;}
        public DateTime Published {get; set;}
        public DateTime StartDate {get; set;}
        public DateTime EndDate {get; set;}
        public int ModuleId {get; set;}
        public int Order {get; set;}
        public float Score {get; set;}
        public int CourseAuthorId {get; set;}
        public virtual Module Module {get; set;}
        public virtual CourseAuthor CourseAuthor {get; set;}
        public virtual ICollection<TaskResult> TaskResults {get; set;}
    }
}

```

```
[NotMapped]
public string TextStartDate
{
    get
    {
        return this.StartDate.ToString("dd.MM.yyyy");
    }
}
```

```
[NotMapped]
public string TextEndDate
{
    get
    {
        return this.EndDate.ToString("dd.MM.yyyy");
    }
}
```

```
[NotMapped]
public string TextPublished
{
    get
    {
        return this.Published.ToString("dd.MM.yyyy");
    }
}
```

```
[NotMapped]
public string TextScore
{
    get
    {
        return Math.Round(this.Score, 2).ToString();
    }
}
```

```

    }
}
}
}

```

Б.5 Текст файла CourseController.cs

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using OnlineEducator.Data;
using OnlineEducator.Models;
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;

namespace OnlineEducator.Controllers
{
    public class CourseController : Controller
    {
        private readonly DataBaseContext _dbcontext;

        public CourseController (DataBaseContext dbc)
        {
            _dbcontext = dbc;
        }

        public ActionResult BestCourses(int n, bool all)

```

```

    {
        var courses = _dbcontext.Courses.Include(c =>
c.CourseId).Include(c => c.Title).Include(c => c.Cover).Include(c =>
c.Score).Include(c => c.Subscribers).Include(c =>
c.TextStartDate).OrderByDescending (c => c.Score);
        if (!all)
        {
            DateTime currdate = DateTime.Now;
            courses.Where(c => c.EndDate <= currdate);
        }
        var clist = courses.ToList();
        return View(clist.Take(n));
    }

```

```

public ActionResult CourseComments(int n)
{
    var courses = _dbcontext.Courses.Include(c =>
c.CourseId).Include(c => c.OrderedComments);
    var clist = courses.ToList();
    return View(clist);
}

```

```

public ActionResult NewCourses(int n, bool all)
{
    var courses = _dbcontext.Courses.Include(c =>
c.CourseId).Include(c => c.Title).Include(c => c.Cover).Include(c =>
c.Score).Include(c => c.Subscribers).Include(c =>
c.TextStartDate).OrderByDescending (c => c.TextPublished);
    if (!all)
    {

```

```

        DateTime currdate = DateTime.Now;
        courses.Where(c => c.EndDate <= currdate);
    }
    var clist = courses.ToList();
    return View(clist.Take(n));
}

```

```

public ActionResult AuthorCourses(int n, int m)
{
    var courses = _dbcontext.CourseAuthors.Include(c =>
c.CourseAuthorId).Include(c => c.CourseId).Include(c => c.AccountId).Include(c
=> c.Role).Where(c => c.AccountId == n).OrderByDescending (c =>
c.TextPublished);
    var clist = courses.ToList();
    if (m > 0)
    {
        return View(clist.Take(m));
    }
    else
    {
        return View(clist);
    }
}

```

```

public ActionResult Details(int n)
{
    var fullinfo = _dbcontext.Courses.Include(c =>
c.CourseId).Include(c => c.Group).Include(c => c.Title).Include(c =>
c.Description).Include(c => c.Prerequisite).Include(c => c.Cover).Include(c =>
c.Score).Include(c => c.TextPublished).Include(c => c.Subscribers).Include(c =>

```

```

c.TextStartDate).Include(c => c.TextEndDate).Include(c =>
c.MinimumScore).Include(c => c.OrderedTopics).Include(c =>
c.OrderedAuthors).FirstOrDefault(a => a.CourseId == n);

```

```

    return View(fullinfo);

```

```

}

```

```

public ActionResult AuthorDetails(int n)

```

```

{

```

```

    var fullinfo = _dbcontext.Courses.Include(c =>

```

```

c.CourseId).Include(c => c.Group).Include(c => c.Title).Include(c =>

```

```

c.Description).Include(c => c.Prerequisite).Include(c => c.Cover).Include(c =>

```

```

c.Score).Include(c => c.TextPublished).Include(c => c.Subscribers).Include(c =>

```

```

c.TextStartDate).Include(c => c.TextEndDate).Include(c =>

```

```

c.MinimumScore).Include(c => c.OrderedTopics).Include(c =>

```

```

c.OrderedAuthors).Include(c => c.UnSuccessLevel).Include(c =>

```

```

c.SuccessLevel).FirstOrDefault(a => a.CourseId == n);

```

```

    return View(fullinfo);

```

```

}

```

```

public ActionResult SearchCourses(string name, bool all)

```

```

{

```

```

    var courses = _dbcontext.Courses.Include(c =>

```

```

c.CourseId).Include(c => c.Title).Include(c => c.Cover).Include(c =>

```

```

c.Score).Include(c => c.Subscribers).Include(c => c.TextStartDate).Where(c =>

```

```

c.Title.Contains(name)).OrderByDescending (c => c.Score);

```

```

    if (!all)

```

```

    {

```

```

        DateTime currdate = DateTime.Now;

```

```

        courses.Where(c => c.EndDate <= currdate);

```

```

    }

```

```

var clist = courses.ToList();
return View(clist.Take(n));
}

```

```

[Authorize]
public ActionResult Create()
{
    ViewData["Groups"] = new SelectList(_dbcontext.Groups,
"GroupId", "Title");
    return View();
}

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind("Title, Description, Prerequisite,
Cover, MinimumScore, GroupId")] Course course, string sdate, string edate)
{
    if (ModelState.IsValid)
    {
        DateTime d1 = DateTime.Parse(sdate);
        DateTime d2 = DateTime.Parse(edate);
        course.StartDate = d1;
        course.EndDate = d2;
        course.Published = DateTime.Now;
        course.Score = 0;
        _dbcontext.Add(course);
        _dbcontext.SaveChanges();
        var teachers = _dbcontext.Accounts.Include(a =>
a.Login).Where(a => a.Role == 2);
        var login = User.Identity.Name;

```

```

        var t = teachers.FirstOrDefault(a => a.Login == login);
        CourseAuthor ca = new CourseAuthor();
        ca.CourseId = course.CourseId;
        ca.AccountId = t.AccountId;
        ca.Role = 2;
        _dbcontext.Add(ca);
        _dbcontext.SaveChanges();
        return RedirectToAction(nameof(Index));
    }
    return View(course);
}

```

[Authorize]

```

public ActionResult Edit(int? n)
{
    var course = _dbcontext.Courses.Find(n);
    if (course == null)
    {
        return NotFound();
    }
    ViewData["Groups"] = new SelectList(_dbcontext.Groups,
"GroupId", "Title");
    return View(course);
}

```

[HttpPost]

[ValidateAntiForgeryToken]

```

public ActionResult Edit(int n, [Bind("CourseId, Title, Description,
Prerequisite, Cover, MinimumScore, GroupId")] Course course, string sdate, string
edate)

```

```

    {
        if (ModelState.IsValid)
        {
            DateTime d1 = DateTime.Parse(sdate);
            DateTime d2 = DateTime.Parse(edate);
            course.StartDate = d1;
            course.EndDate = d2;
            _dbcontext.Update(course);
            _dbcontext.SaveChanges();
        }
        ViewData["Groups"] = new SelectList(_dbcontext.Groups,
"GroupId", "Title");
        return View(course);
    }

    public ActionResult GroupCourses(int n, int m)
    {
        var courses = _dbcontext.Courses.Include(c =>
c.CourseId).Include(c => c.Title).Include(c => c.Cover).Include(c =>
c.Score).Include(c => c.Subscribers).Include(c => c.TextStartDate) .Include(c =>
c.Group).Where(c => c.GroupId == n).OrderByDescending (c => c.Score);
        if (!all)
        {
            DateTime currdate = DateTime.Now;
            courses.Where(c => c.EndDate <= currdate);
        }
        var clist = courses.ToList();
        if (m > 0)
        {
            return View(clist.Take(m));
        }
    }

```

```

    }
    else
    {
        return View(clist);
    }
}
}
}

```

Б.6 Текст файла SubscriptionController.cs

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using OnlineEducator.Data;
using OnlineEducator.Models;
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;

namespace OnlineEducator.Controllers
{
    public class SubscriptionController : Controller
    {
        private readonly DataBaseContext _dbcontext;

        public SubscriptionController (DataBaseContext dbc)
        {

```

```

        _dbcontext = dbc;
    }

```

```

    public ActionResult UserSubscriptions(int n, int m)
    {
        var subscriptions = _dbcontext.Subscriptions.Include(c =>
c.SubscriptionId).Include(c => c.Course).Include(c => c.Account).Include(c =>
c.TextScore).Include(c => c.TextSubDate).Include(c => c.TextEndDate).Include(c
=> c.Rating).Include(c => c.Active).Include(c => c.Success).Include(c =>
c.PredictedSuccess).Include(c => c.Learned).Where(c => c.Account.AccountId ==
n).OrderByDescending (c => c.SubscriptionId);

        if (m==1)
        {
            subscriptions = subscriptions.Where(c => c.Active == 0)
        }
        else if (m==2)
        {
            subscriptions = subscriptions.Where(c => c.Active == 1)
        }
        return View(subscriptions.ToList());
    }

```

```

    public ActionResult ProcessedSubscriptions(int n)
    {
        DateTime currrdate = DateTime.Now;
        var subscriptions = _dbcontext.Subscriptions.Include(c =>
c.SubscriptionId).Include(c => c.Course).Include(c => c.Account).Include(c =>
c.TextScore).Include(c => c.TextSubDate).Include(c => c.TextEndDate).Include(c
=> c.Rating).Include(c => c.Active).Include(c => c.Success).Include(c =>

```

```

c.PredictedSuccess).Include(c => c.Learned).Where(c => c.Account.AccountId ==
n && c.Course.EndDate <= currdate).OrderByDescending (c => c.SubscriptionId);
        return View(subscriptions.ToList());
    }

```

```

public ActionResult UnSuccessSubscriptions(int n)
{
    DateTime currdate = DateTime.Now;
    var subscriptions = _dbcontext.Subscriptions.Include(c =>
c.SubscriptionId).Include(c => c.Course).Include(c => c.Account).Include(c =>
c.TextScore).Include(c => c.TextSubDate).Include(c => c.TextEndDate).Include(c
=> c.Rating).Include(c => c.Active).Include(c => c.Success).Include(c =>
c.PredictedSuccess).Include(c => c.Learned).Where(c => c.Account.AccountId ==
n && c.Course.EndDate <= currdate && c.Success == 0).OrderByDescending (c
=> c.SubscriptionId);
        return View(subscriptions.ToList());
    }

```

```

[Authorize]
public ActionResult Delete(int? n)
{
    if (n == null)
    {
        return NotFound();
    }
    var sub = _dbcontext.Subscriptions.Include(c =>
c.SubscriptionId).Include(c => c.Course).Include(c => c.Account).Include(c =>
c.TextScore).Include(c => c.TextSubDate).Include(c => c.TextEndDate).Include(c
=> c.Rating).FirstOrDefault(c => c.SubscriptionId == n);
    if (sub == null)

```

```

    {
        return NotFound();
    }
    return View(sub);
}

```

```

[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int n)
{
    var sub = _dbcontext.Subscriptions.Find(n);
    _dbcontext.Subscriptions.Remove(sub);
    _dbcontext.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult StopSubscription(int n)
{
    var sub = _dbcontext.Subscriptions.Find(n);
    sub.Active = 0;
    _dbcontext.Update(sub);
    _dbcontext.SaveChanges();
}
return View(sub);
}

```

```

public ActionResult ActivateSubscription(int n)
{

```

```

var sub = _dbcontext.Subscriptions.Find(n);
sub.Active = 1;
_dbcontext.Update(sub);
_dbcontext.SaveChanges();
}
return View(sub);
}

```

```

public ActionResult RateSubscription(int n, int r)
{
    var sub = _dbcontext.Subscriptions.Find(n);
    sub.Rating = r;
    _dbcontext.Update(sub);
    _dbcontext.SaveChanges();
}
return View(sub);
}

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind("CourseId, AccountId")]
Subscription sub)
{
    if (ModelState.IsValid)
    {
        DateTime currdate = DateTime.Now;
        sub.StartDate = currdate;
        sub.Rating = 0;
        sub.Score = 0;
        sub.Active = 1;
    }
}

```

```
        _dbcontext.Add(sub);  
        _dbcontext.SaveChanges();  
    }  
    return View(sub);  
}
```

```
public ActionResult Details(int n)  
{  
    var fullinfo = _dbcontext.Subscriptions.Include(c =>  
c.SubscriptionId).Include(c => c.Course).Include(c => c.Account).Include(c =>  
c.TextScore).Include(c => c.TextSubDate).Include(c => c.TextEndDate).Include(c  
=> c.Rating).Include(c => c.Active).Include(c => c.Success).Include(c =>  
c.PredictedSuccess).FirstOrDefault(c => c.SubscriptionId == n);  
    return View(fullinfo);  
}  
}  
}
```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд 1

Об'єкт, предмет, мета роботи

Об'єкт роботи – процес вивчення онлайн курсів.

Предмет роботи – методи класифікації для визначення успішності вивчення онлайн курсів.

Мета роботи – розроблення програмного забезпечення вивчення онлайн курсів для підтримки студентів на ранніх етапах.

Рисунок В.2 – Слайд 2

Завдання роботи

- ✓ створити новий метод прогнозування успішності вивчення онлайн курсу студентом;
- ✓ виконати проєктування програмного забезпечення;
- ✓ реалізувати програмне забезпечення;
- ✓ виконати експериментальне дослідження проблеми.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих методів визначення успішності вивчення онлайн курсів

Метод	Основа оцінки	Модель	Етап застосування	Вихідна ознака
Метод [A. K. Hamoud, A. M. Humadi]	Опитувальник студентів	Нейронна мережа прямого поширення	До початку навчання	Клас успішності
Метод [E. Emirtekin, M. Karatay, T. Kislal]	Загальні характеристики і поточні результати навчання	KNN	У процесі навчання	Клас оцінки
Метод [Z. J. Kovacic]	Загальні характе-ристичи	Дерева рішень	До початку навчання	Клас успішності
Пропонований метод	Загальні і поточні результати навчання	Нейронна мережа прямого поширення	Перший етап навчання	Клас успішності

4

Рисунок В.4 – Слайд 4

Метод прогнозування успішності вивчення онлайн курсу студентом

1. Підготувати вибірку даних.
2. Передобробити підготовлену вибірку даних.
3. Створити модель на основі багатошарової нейронної мережі прямого поширення для обробки наступних входних ознак:
 - поточний рівень освіти студента;
 - вік студента;
 - кількість курсів, яку студент вивчав на даний момент;
 - кількість курсів, яку студент успішно завершив;
 - отримана оцінка за проходження курсу, для якого відбувається визначення успішності, на поточний момент;
 - вага отриманої оцінки;
 - відсоток вивчених матеріалів;
 - кількість інших курсів, на які відписався студент.
4. Навчити модель за передобробленою навчальною вибіркою даних з оцінюванням результатів навчання за передобробленою тестувальною вибіркою даних.

5

Рисунок В.5 – Слайд 5

Вибір фреймворку розробки

Характеристика	ASP.NET	Node.js
Розробка сучасних вебзастосунків	Співмірно	Співмірно
Підтримка сучасних бібліотек машинного навчання	Співмірно	Співмірно
Рівень безпеки	Краще	
Зручність масштабування	Дещо краще	
Якість коду	Краще	
Мови програмування	C# (можливо F#)	JavaScript

6

Рисунок В.6 – Слайд 6



Рисунок В.7 – Слайд 7

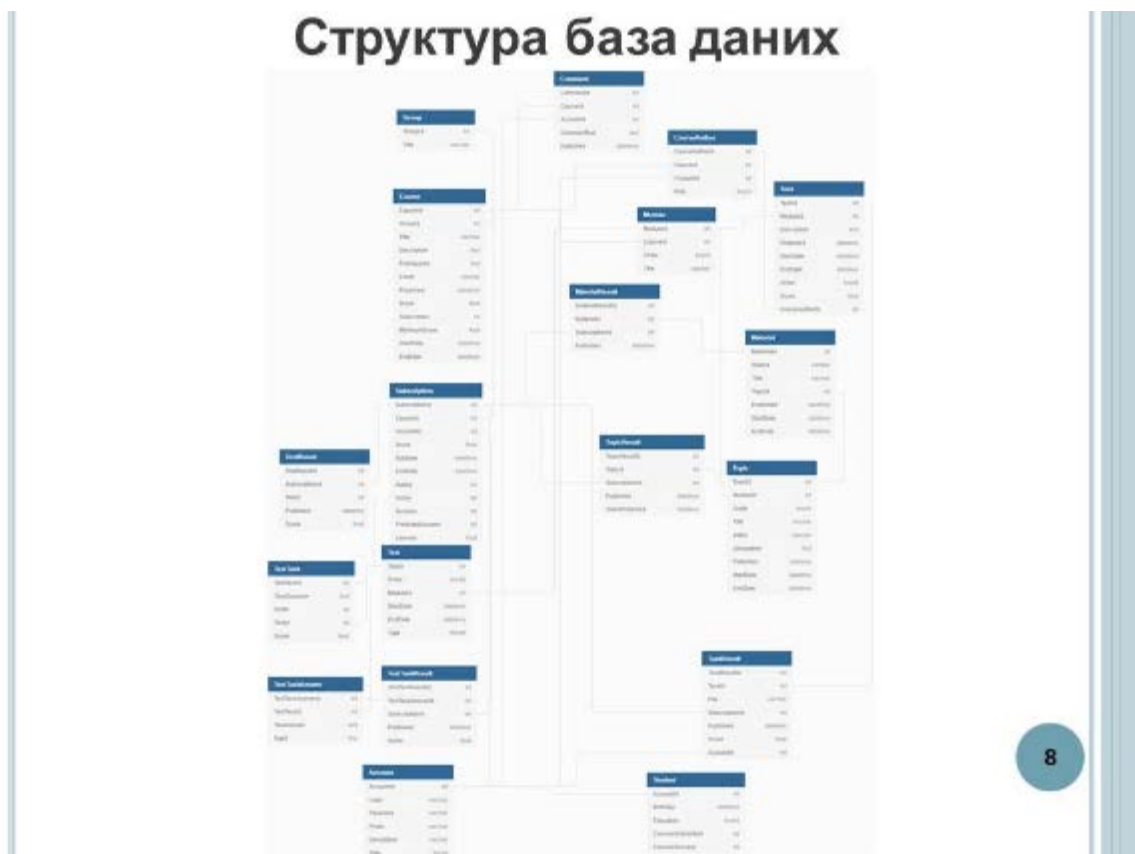


Рисунок В.10 – Слайд 10

Знаходження онлайн курсів для підписки

Курси Підписки Профіль Вийти

Рейтинг Нові Групи Пошук

Рейтинг курсів

доступне для підписки



DATA SCIENCE

Візуалізація даних

1.12.2021 / 10 Підписників

Візуалізація даних є однією з необхідних частин науки про дані. Інфографіка, наука про дані сяють, за вдалої візуалізації даних. Ми навчимося використовувати ці інструменти в своїй роботі.

Оцінка: 5.0



ПСИХОЛОГІЯ

Комунікація в команді

1.12.2021 / 5 Підписників

Робота в команді потребує визначити впливати на результат кожного учасника, щоб досягти загального успіху. Мотивація, вирішення конфліктів - тільки деякі з питань, яких ми торнемося.

Оцінка: 4.5



ЕКОНОМІКА

Фінансовий аналіз

1.12.2021 / 30 Підписників

Цей курс присвячений аналізу інвестицій зі сторони компанії, яка потребує такого фінансування. Прибутковість, фінансові потоки, ліквідність - одні з понять, з якими можна познайомитись.

Оцінка: 4.5

11

Рисунок В.11 – Слайд 11

Вивчення онлайн курсів

Нова Курси Підписки Профіль Online

Візуалізація даних

Тест № 1: Візуалізація мовою Python

1. Який пакет використовується для візуалізації?

☐ NumPy

☒ Seaborn

☐ SciPy

☐ Treemap

Наступна запитання

Візуалізація даних

The beauty of data visualization - David McDermott



Тема 1: Вступ до візуалізації даних

Вступ до візуалізації даних є однією з необхідних частин науки про дані. Інфографіка, наука про дані сяють, за вдалої візуалізації даних. Ми навчимося використовувати ці інструменти в своїй роботі.

Автори: Даниєл - це особа, яка працює над цим курсом.

Візуалізація даних - це процес аналізу даних з метою роботи з ними ефективно, тобто зрозуміти, як вони пов'язані між собою, і як це впливає на результати дослідження.

Додаток до: 12.12.2021

Матеріали

Посилання на матеріали

12.12.2021

12.12.2021

12

Рисунок В.12 – Слайд 12

Керування роботою зі студентами

Назад Курси Кабінет Профіль Вийти

Студенти

Курс: Визначення даних

успіх

Прогноз успішності

20 балів



Андрій Долько

Тест № 1: 20

Тест № 2: 0

Тест № 3: 0

Тест № 4: 0

Завдання

успіх

Прогноз успішності

10 балів



Ігор Певченко

Тест № 1: 10

Тест № 2: 0

Тест № 3: 0

Тест № 4: 0

Завдання

НЕВДАЧА

Прогноз успішності

10 балів



Іван Зайченко

Тест № 1: 10

Тест № 2: 0

Тест № 3: 0

Тест № 4: 0

Завдання

13

Рисунок В.13 – Слайд 13

Експериментальне дослідження

Метод	Точність результатів
Метод k-найближчих сусідів	0,67
Пропонований метод	0,84

Рисунок В.14 – Слайд 14

Висновки

Розроблений програмний застосунок призначений для вивчення онлайн курсів і надає викладачам засоби роботи авторів курсів для публікації власних онлайн курсів і підтримки викладання цих курсів студентам, а студентам – засоби вивчення онлайн курсів і роботи з матеріалами цих курсів. У межах курсів підтримуються модулі, теми, матеріали, у тому числі відеоматеріали, завдання і тести.

Науковою новизною роботи є запропонований метод прогнозування успішності вивчення онлайн курсу студентом, який виконує прогнозування шляхом вирішення задачі класифікації на основі застосування багатосарової нейронної мережі прямого поширення, що дозволяє приймати рішення щодо необхідності додаткової підтримки вивчення курсів студентами на ранніх етапах.

15

Завдання дипломної роботи виконані повністю.

Рисунок В.15 – Слайд 15