

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ
ПРОВЕДЕННЯ МАЙСТЕР-КЛАСІВ З МАЛЮВАННЯ
SOFTWARE FOR DRAWING WORKSHOP MANAGEMENT

Виконав(ла): студент(ка) 4 курсу, групи КНТ-112

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

НІКІТИНА А.А.

(ПРИЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

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

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

НІКІТИНОЇ Анастасії Андріївни
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення підтримки проведення майстер-класів з малювання. Software for Drawing Workshop Management

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

затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 01 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програми. 3. Розробка програми. 4. Експлуатація та тестування програми.

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	БСЛОВА А.В., асистент		

7. Дата видачі завдання « 20 » квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 2
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Анастасія НІКІТІНА
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Валерій ЛЬОВКІН
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 95 с., 3 табл., 25 рис., 3 дод., 10 джерел.

МАЙСТЕР-КЛАС, МАЛЮВАННЯ, ОЦІНЮВАННЯ, ЗВІТ, ВЕБЗАСТОСУНОК, ФРЕЙМВОРК.

Об'єкт роботи – процес розробки програмного забезпечення підтримки проведення майстер-класів з малювання. Предмет роботи – програмне забезпечення підтримки проведення майстер-класів з малювання.

Мета роботи – підвищення залученості відвідувачів до проведення майстер-класів з малювання за рахунок інформування та вивчення зацікавленості в таких майстер-класах на основі розробки програмного забезпечення підтримки проведення майстер-класів з малювання.

Матеріали, методи та технічні засоби: мова Python та фреймворк для веброзробки Django як базові технології, система PostgreSQL для роботи з даними.

Результати. Програмне забезпечення підтримки проведення майстер-класів з малювання було порівняно з аналогами з виділенням особливостей, реалізовано проєктування з моделюванням і визначенням структури бази даних, програмно реалізовано та протестовано.

Висновки. Програма, яка створена в роботі, може використовуватися для надання інформації про майстер-класи з малювання, реєстрації на проведення цих майстер-класів, вимірювання зацікавленості потенційних відвідувачів у проведенні конкретних подій майстер-класів з малювання перед безпосередньо відкриттям реєстрації на такі майстер-класи з малювання, для обміну інформацією про досвід участі в таких майстер-класах.

Галузь використання. Майстер-класи з малювання, які організовуються як індивідуальними майстрами, так і більш централізовано на певних конкретних локаціях.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 95 pages, 3 tables, 25 figures, 3 appendixes, 10 sources.

WORKSHOP, DRAWING, EVALUATION, REPORT, WEB APPLICATION, FRAMEWORK.

The object of the work is the process of developing software to support drawing workshops. The subject of the work is software to support drawing workshops.

The purpose of the work is to increase the involvement of visitors in drawing workshops by informing and studying interest in such workshops based on the development of software to support drawing workshops.

Materials, methods and technical means: Python language and Django web development framework as basic technologies, PostgreSQL system for working with data.

Results. The software to support drawing workshops was compared with analogues with the selection of features, was designed with software modeling and database structuring, implemented and tested.

Conclusions. The program created in the work can be used to provide information about drawing workshops, register for these workshops, measure the interest of potential visitors in holding specific drawing workshop events before directly opening registration for such drawing workshops, and to exchange information about the experience of participating in such workshops.

Scope of use. Drawing workshops organized both by individual masters and more centrally at certain specific locations.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд програмних засобів підтримки проведення майстер-класів з малювання	10
1.2 Порівняльний аналіз програмного забезпечення підтримки проведення майстер-класів з малювання.....	11
1.3 Висновки за розділом 1	12
2 Проєктування програми.....	14
2.1 Вибір засобів проєктування і розробки	14
2.2 Моделювання роботи з програмою підтримки проведення майстер- класів з малювання	15
2.3 Визначення структури бази даних програми підтримки проведення майстер-класів з малювання.....	21
2.4 Висновки за розділом 2	31
3 Розробка програми	32
3.1 Послідовність роботи з програмою підтримки проведення майстер- класів з малювання	32
3.2 Рішення з реалізації програми підтримки проведення майстер-класів з малювання	33
3.3 Висновки за розділом 3	49
4 Експлуатація та тестування програми	51
4.1 Призначення програми	51
4.2 Умови виконання програми	51
4.3 Експлуатація програми підтримки проведення майстер-класів з малювання	51

4.4 Тестування програми підтримки проведення майстер-класів з малювання	55
4.5 Висновки за розділом 4	59
Висновки	60
Перелік джерел посилання	61
Додаток А Технічне завдання	62
Додаток Б Текст програми	67
Додаток В Слайди презентації.....	88

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД — база даних;

ФДВР — фреймворк для веброзробки.

ВСТУП

Майстер-класи з малювання вимагають обов'язкової реєстрації участі відвідувачів у них, щоб контролювати кількість учасників, але окрім даного процесу виникає ще одна проблема. Вона пов'язана з тим, що кількість учасників таких майстер-класів з малювання може бути невеликою, тоді проведення їх може бути неефективним. Для підвищення ефективності потрібно з одного боку надавати достатню кількість інформації про досвід участі конкретних людей у таких майстер-класах з малювання в минулому, але також і вивчати готовність людей до відвідування майстер-класів, які можуть бути проведені. Саме реалізації таких можливостей потрібно приділити увагу при розробці програмного забезпечення підтримки проведення майстер-класів з малювання.

Тому мета роботи полягає в підвищенні залученості відвідувачів до проведення майстер-класів з малювання за рахунок інформування та вивчення зацікавленості в таких майстер-класах на основі розробки програмного забезпечення підтримки проведення майстер-класів з малювання. Мета повинна бути досягнута за виконання завдань:

- аналіз проблеми розробки програми підтримки проведення майстер-класів з малювання;
- проєктування програми підтримки проведення майстер-класів з малювання;
- розробка програми підтримки проведення майстер-класів з малювання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд програмних засобів підтримки проведення майстер-класів з малювання

Оскільки основною функціональністю програми, що має розроблятися в роботі для підтримки проведення майстер-класів з малювання, є бронювання участі в цих майстер-класах, то саме такі можливості спершу були розглянуті в якості пріоритету. Застосунком, який забезпечує ці можливості, є YouCanBookMe (рис. 1.1) [1], який не призначений безпосередньо для майстер-класів з малювання, але дозволяє універсально розробити систему бронювання місць, яка і я для них може бути застосована. При цьому слід розуміти, що програма не забезпечує створення звітів, оцінювань за результатами проведення майстер-класів, тобто не сприяє додатковому інформуванню відвідувачів, також не доступна можливість голосування про відкриття нових майстер-класів.

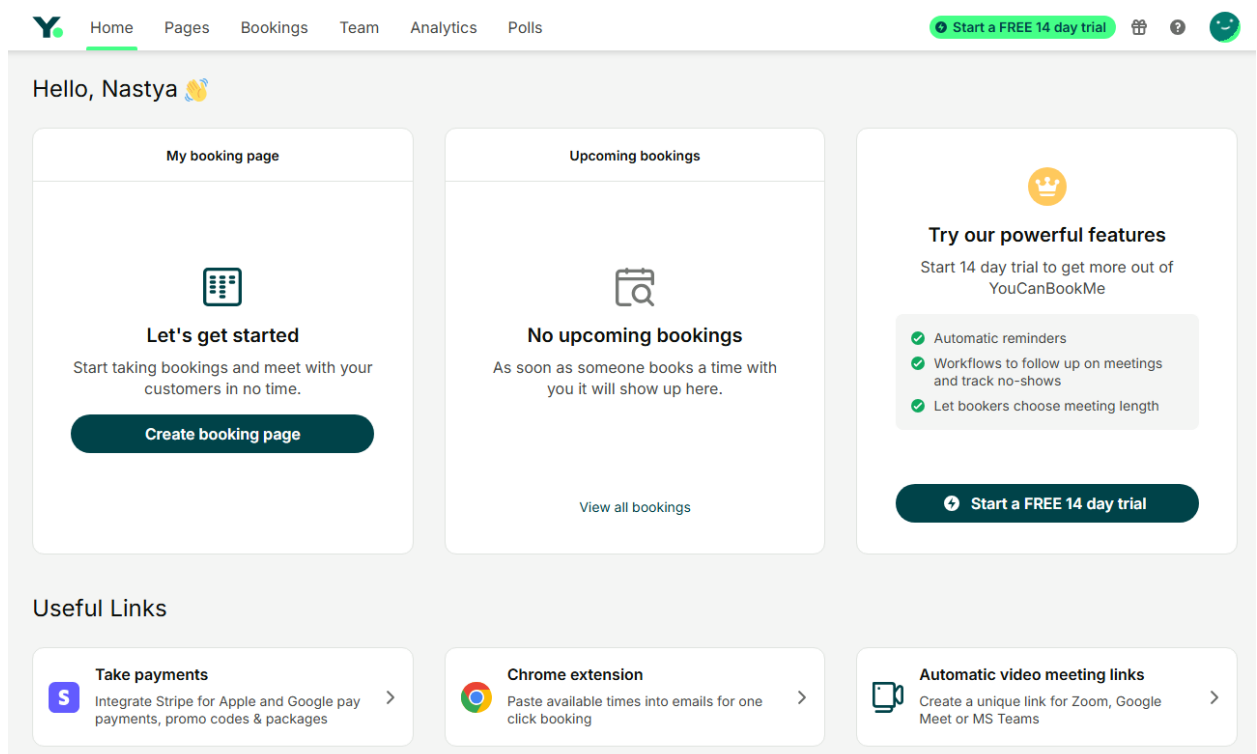


Рисунок 1.1 – Інтерфейс програми YouCanBookMe

Ще однією альтернативою застосунку, що розробляється, націленою в першу чергу на підтримку бронювання, але з врахуванням особливостей творчих майстер-класів, є програма Bookedin [2].

Bookedin надає можливості бронювання участі в майстер-класах, при цьому надаючи можливості попереднього знайомства з відвідувачами шляхом запитування від них додаткової інформації, нагадування про події майстер-класів з малювання зокрема, а також будь-яких інших творчих майстер-класів, щоб не пропустити ці події. Також доступна можливість сплати за бронювання, що може бути корисно в певних ситуаціях. З іншого боку програма не забезпечує створення звітів, оцінювань за результатами проведення майстер-класів, голосування про відкриття нових майстер-класів.

Окрім цього напрямку існують також застосунки, які дозволяють серед їх основних можливостей малювати. Це може бути корисною функцією під час організації майстер-класів з малювання. Проте якщо майстер-клас передбачає використання звичайних засобів для малювання офлайн, то ця можливість не буде корисною для користувачів.

Програма ArtWorkout [3] надає можливості малювання за заданими завданнями, одночасної участі в таких завданнях з іншими користувачами, таким чином створюючи певну подобу майстер-класам онлайн. Проте програма не підтримує логіку звичайних майстер-класів. Відповідно виконати бронювання, навести інформацію про такі майстер-класи, підтримувати супутні інструменти в програмі неможливо, що не дозволяє вирішувати мету роботи за її допомогою.

1.2 Порівняльний аналіз програмного забезпечення підтримки проведення майстер-класів з малювання

До порівняння програмного забезпечення підтримки проведення майстер-класів з малювання винесено власну розробку – програму

DrawWorkshopApp та ті рішення, які частково здатні задовольняти мету та завдання роботи: застосунки YouCanBookMe та Bookedin (табл. 1.1).

Таблиця 1.1 – Порівняння програмних засобів підтримки проведення майстер-класів з малювання

Критерій	YouCan-BookMe	Bookedin	DrawWorkshopApp
Бронювання місць на майстер-класах з малювання	Так	Так	Так
Надання даних профілів відвідувачів	Ні	Так	Так
Створення звітів про майстер-класи з малювання	Ні	Ні	Так
Голосування за відкриття подій майстер-класів з малювання	Ні	Ні	Так
Параметризований пошук	Ні	Ні	Так

Розробка програми DrawWorkshopApp відрізняється від аналогів наявністю можливостей створення звітів про майстер-класи з малювання, голосування за відкриття подій майстер-класів з малювання, параметризованого пошуку, що відсутні в аналогів повністю.

1.3 Висновки за розділом 1

За результатами розділу 1 програма підтримки проведення майстер-класів з малювання була концептуалізована на основі виділення її відмінностей порівняно з аналогами, до яких віднесено в першу чергу YouCanBookMe та Bookedin. Розробка повинна дозволити створювати звіти про майстер-класи з малювання, сприяючи додатковому інформуванню

потенційних відвідувачів про досвід, який вже сформований, голосувати за відкриття подій майстер-класів з малювання, окрім того зручності роботи також має сприяти параметризований пошук серед локацій для проведення майстер-класів, самих запланованих подій майстер-класів та майстрів з малювання. Тож у цьому розділі виконано аналіз проблеми розробки програми підтримки проведення майстер-класів з малювання.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Вибір засобів проєктування і розробки

Розробка мовою Python є виграшною з точки зору швидкості написання програмного коду, гнучкості його подальшої модифікації, готовності до веброзробки, що призвело до вибору саме цієї мови для розробки програми підтримки проведення майстер-класів з малювання, однак веброзробка потребує застосування готового рішення – фреймворку для веброзробки (ФДВР), щоб цей процес пришвидшити. Вибір ФДВР включав ті, що найбільш активно використовуються для Python-розробки: Django [4] і Flask [5] (табл. 2.1).

Як ФДВР обрано Django, оскільки він має більше вбудованих засобів, більше готових можливостей, які можуть бути додані, більше вбудованих засобів захисту, де Flask є значно слабшою альтернативою, окрім того Django пропонує вбудоване об'єктно-реляційне відображення для роботи з даними, що також пришвидшує розробку [6]-[7].

Таблиця 2.1 – Порівняння ФДВР для програми підтримки проведення майстер-класів з малювання

Критерій	Django	Flask
Основні можливості	ФДВР повного стеку	Мікрофреймворк
Додавання можливостей	Велика кількість при великій кількості вбудованих засобів	Достатня кількість при малій кількості вбудованих засобів
Наявність засобів захисту	Велика кількість вбудованих	Мала кількість вбудованих, далі потрібно розширювати
Робота з даними	На основі вбудованого об'єктно-реляційного відображення	Потребує підключення або розробки з нуля

2.2 Моделювання роботи з програмою підтримки проведення майстер-класів з малювання

Неавторизований користувач програми підтримки проведення майстер-класів з малювання має основний потік сценаріїв:

- перегляд базового інформування про майстер-клас з малювання;
- перегляд сторінки майстра з базовими інформуваннями про майстер-класи з малювання;
- перегляд публікації проведення майстер-класу з малювання;
- перегляд голосування за проведення майстер-класу з малювання;
- перегляд даних локації для проведення майстер-класів з малювання;
- авторизація майстра з малювання;
- авторизація відвідувача;
- авторизація менеджера;
- перегляд запланованих для проведення майстер-класів з малювання за майстром;
- перегляд проведених майстер-класів з малювання за майстром;
- перегляд запланованих для проведення майстер-класів з малювання за локацією;
- перегляд проведених майстер-класів з малювання за локацією;
- перегляд звіту за проведеним майстер-класом з малювання;
- перегляд оцінювань майстер-класу з малювання;
- перегляд оцінювань відвідувача майстер-класів з малювання;
- перегляд сторінки відвідувача майстер-класів з малювання;
- пошук серед локацій для проведення майстер-класів з малювання;
- пошук серед запланованих для проведення майстер-класів з малювання;
- пошук серед майстрів з малювання;
- перегляд оцінювань за майстер-класами майстра з малювання;
- перегляд звітів за майстер-класом з малювання;

– перегляд звітів за майстер-класами з малювання відвідувача (рис. 2.1).

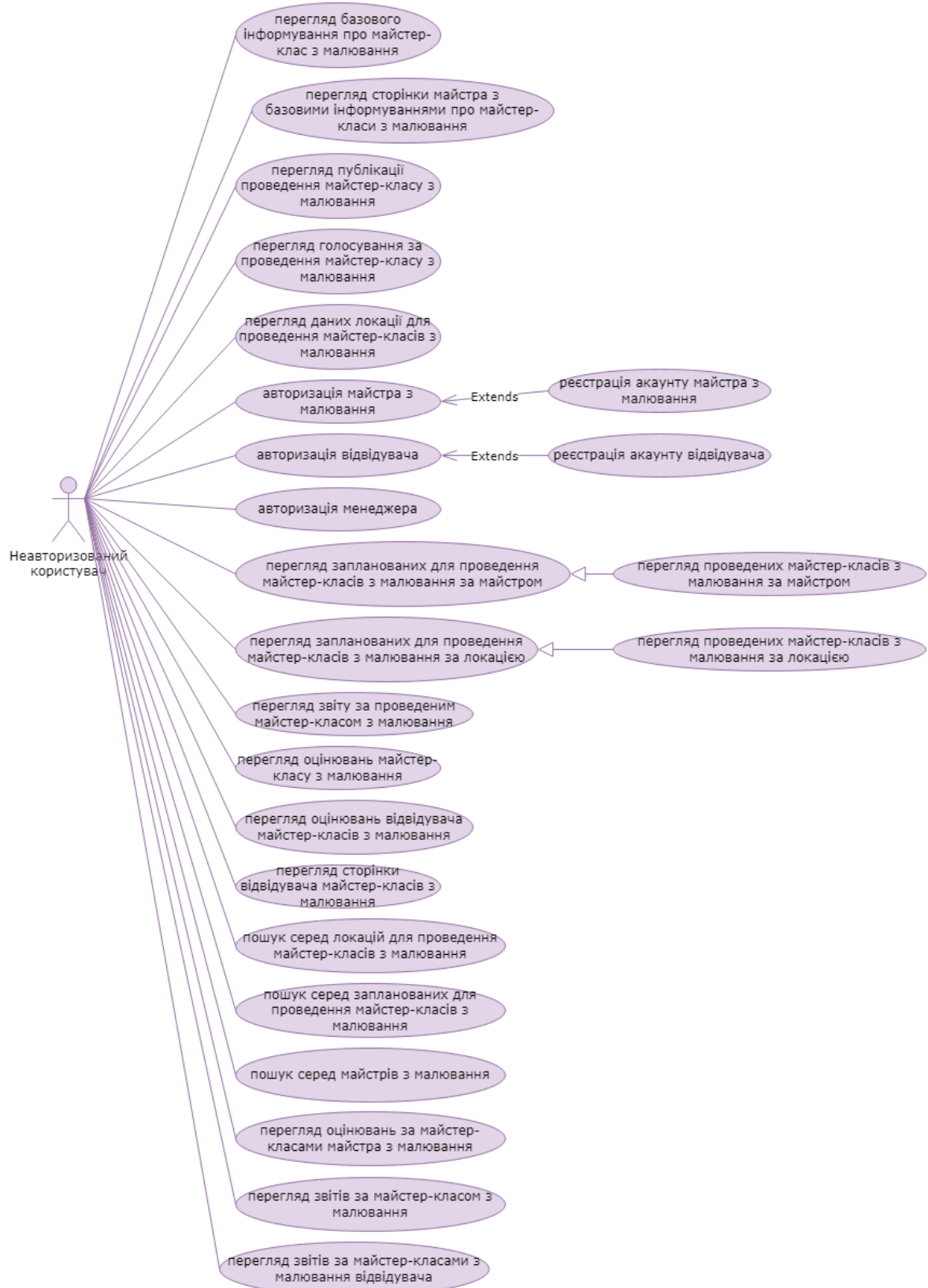


Рисунок 2.1 – Діаграма прецедентів неавторизованого користувача програми підтримки проведення майстер-класів з малювання

Альтернативні сценарії неавторизованого користувача:

- реєстрація акаунту майстра з малювання як альтернатива до авторизації майстра з малювання, коли користувач не має акаунту майстра з малювання в програмі, але фактично ним є та потребує такої роботи з програмою;

- реєстрація акаунту відвідувача як альтернатива до авторизації відвідувача, коли користувач не має акаунту відвідувача, але потребує виконання відповідних сценаріїв авторизованого відвідувача, які надає йому програма.

Відвідувач програми підтримки проведення майстер-класів з малювання має основний потік сценаріїв додатково до неавторизованого користувача:

- голосування за проведення майстер-класу з малювання;
- перегляд майстер-класів, за проведення яких проголосував користувач;
- додавання звіту за проведеним майстер-класом з малювання;
- оцінювання проведеного майстер-класу з малювання;
- оновлення даних відвідувача майстер-класів з малювання;
- бронювання місця на майстер-класі з малювання;
- перегляд власних бронювань місць на майстер-класах з малювання;
- підтвердження запрошення відвідувача на безкоштовне відвідування майстер-класу;
- перегляд адресованих для безкоштовного відвідування запрошень на майстер-класи з малювання (рис. 2.2).

Альтернативні сценарії відвідувача:

- скасування голосування за проведення майстер-класу з малювання альтернативно до голосування за проведення майстер-класу з малювання, коли таке голосування за проведення цього майстер-класу з малювання було виконано помилково цим відвідувачем;

- оновлення звіту за проведеним майстер-класом з малювання альтернативно до додавання звіту за проведеним майстер-класом з малювання, коли такий звіт за проведеним майстер-класом з малювання був створений, але містить неточності;
- скасування бронювання місця на майстер-класі з малювання альтернативно до бронювання місця на майстер-класі з малювання, коли таке бронювання місця на майстер-класі з малювання було виконано відвідувачем помилково або коли воно більше не відповідає його інтересам;
- скасування запрошення відвідувача на безкоштовне відвідування майстер-класу альтернативно до підтвердження запрошення відвідувача на безкоштовне відвідування майстер-класу, коли таке запрошення не відповідає інтересам цього відвідувача.

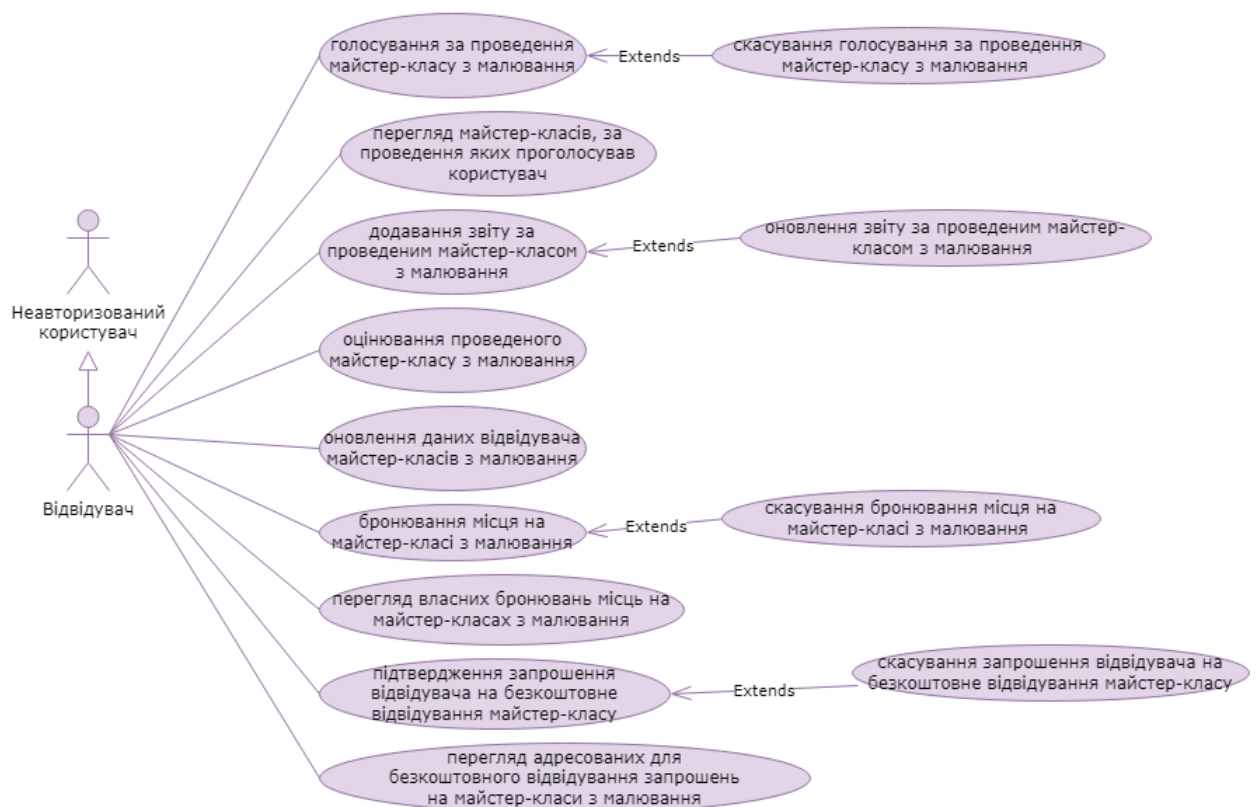


Рисунок 2.2 – Діаграма прецедентів відвідувача програми підтримки проведення майстер-класів з малювання

Майстер програми підтримки проведення майстер-класів з малювання має основний потік сценаріїв додатково до неавторизованого користувача:

- створення базового інформування про майстер-клас з малювання;
- перегляд власних базових інформувачів про майстер-класи з малювання;
- публікація проведення майстер-класу з малювання;
- відкриття голосування за проведення майстер-класу з малювання;
- додавання звіту за проведеним майстер-класом з малювання;
- оцінювання відвідувача майстер-класу з малювання;
- перегляд бронювань за майстер-класом з малювання;
- запрошення відвідувача на безкоштовне відвідування майстер-класу;
- перегляд оформлених запрошень на безкоштовне відвідування власного майстер-класу (рис. 2.3).

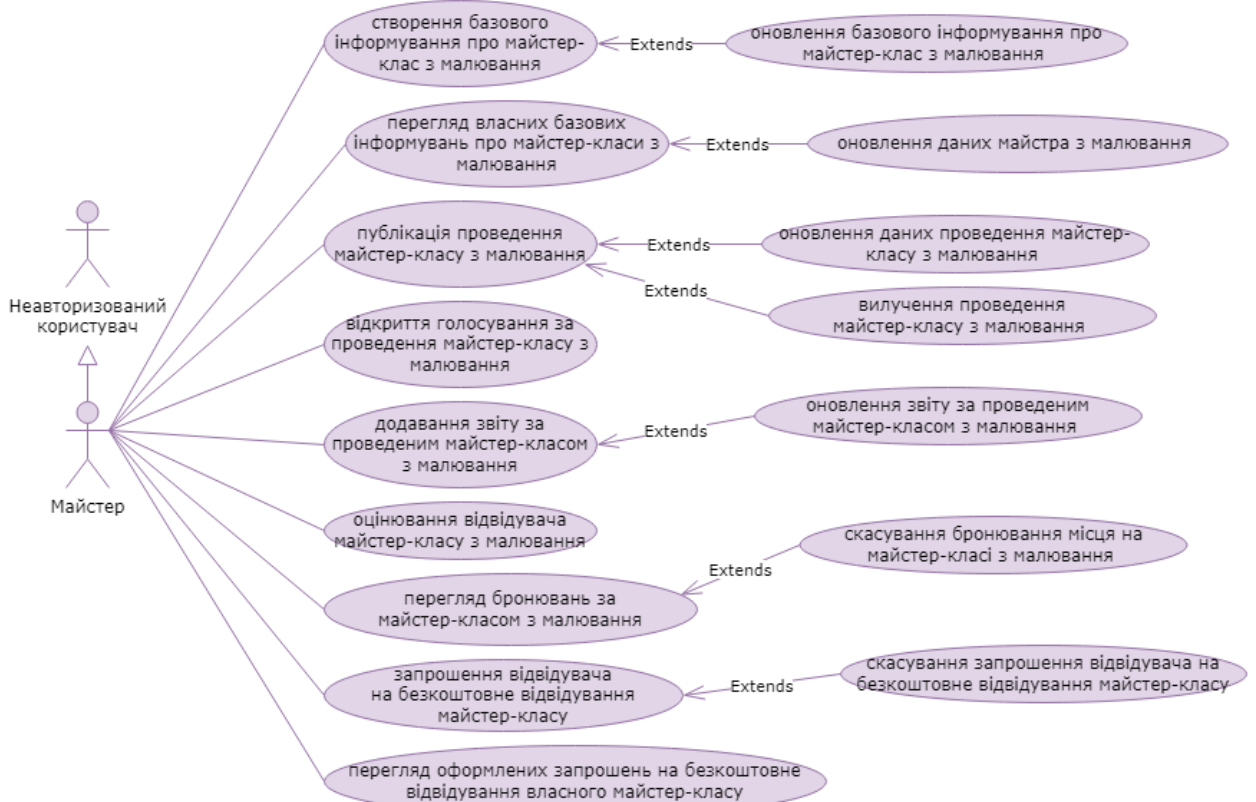


Рисунок 2.3 – Діаграма прецедентів майстра програми підтримки проведення майстер-класів з малювання

Альтернативні сценарії майстра:

- оновлення базового інформування про майстер-клас з малювання альтернативно до створення базового інформування про майстер-клас з малювання, коли таке базове інформування про майстер-клас з малювання містить неточні або неактуальні дані;
- оновлення даних майстра з малювання альтернативно до перегляду власних базових інформувальних про майстер-класи з малювання, коли самі дані майстра з малювання є некоректними;
- оновлення даних проведення майстер-класу з малювання альтернативно до публікації проведення майстер-класу з малювання, коли таке проведення майстер-класу з малювання містить неактуальні або неточні дані;
- вилучення проведення майстер-класу з малювання альтернативно до публікації проведення майстер-класу з малювання, коли таке проведення майстер-класу з малювання взагалі більше не передбачається, зокрема бо було створено помилково;
- оновлення звіту за проведеним майстер-класом з малювання альтернативно до додавання звіту за проведеним майстер-класом з малювання, коли такий звіт за проведеним майстер-класом з малювання був створений, але містить некоректності;
- скасування бронювання місця на майстер-класі з малювання альтернативно до перегляду бронювань за майстер-класом з малювання, коли майстер не бажає дозволяти саме певному відвідувачу відвідати його майстер-клас з малювання;
- скасування запрошення відвідувача на безкоштовне відвідування майстер-класу альтернативно до запрошення відвідувача на безкоштовне відвідування майстер-класу, коли це запрошення відвідувача на безкоштовне відвідування майстер-класу більше не є актуальним.

Менеджер програми підтримки проведення майстер-класів з малювання має основний потік сценаріїв додатково до неавторизованого

користувача, що складається зі сценарію додавання локації для проведення майстер-класів з малювання.

Альтернативним сценарієм для менеджера програми підтримки проведення майстер-класів з малювання є оновлення даних локації для проведення майстер-класів з малювання альтернативно до додавання локації для проведення майстер-класів з малювання, коли така локація для проведення майстер-класів з малювання вже внесена в програму, але наявні дані є неточними (рис. 2.4).

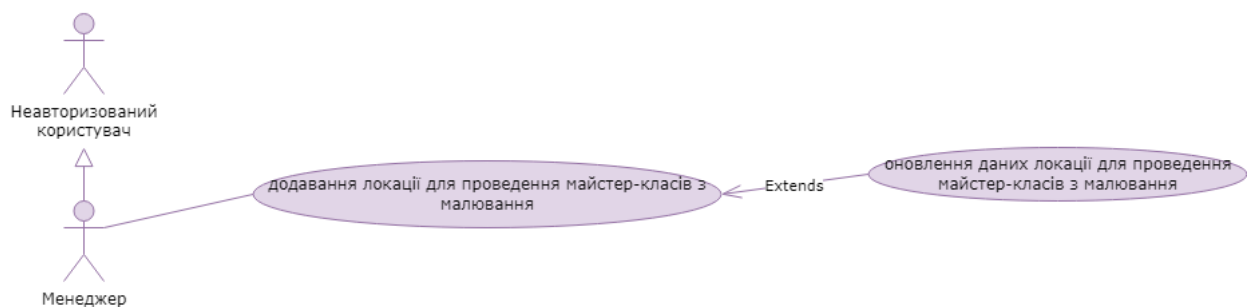


Рисунок 2.4 – Діаграма прецедентів менеджера програми підтримки проведення майстер-класів з малювання

2.3 Визначення структури бази даних програми підтримки проведення майстер-класів з малювання

База даних (БД) була розроблена таким чином, щоб у складі програми підтримки проведення майстер-класів з малювання забезпечувати наповнення та доступ до тих даних, які необхідні в світлі тих функцій, які програма повинна надавати користувачам різних ролей. В якості базової системи використано PostgreSQL.

БД має таблицю Drawstyle, яка визначає стилі малювання через поля:

- id – ідентифікатор стилю малювання;
- drawstyle_name – назва стилю малювання.

БД має таблицю Drawworkshop, яка визначає базове інформування про майстер-класи з малювання через поля:

– id – ідентифікатор базового інформування про майстер-клас з малювання;

– drawworkshop_title – загальна назва майстер-класу з малювання;

– drawmaster – майстер, який проводить цей майстер-клас з малювання, визначений зв'язком з таблицею Drawmaster через зовнішній ключ;

– draw_style – стиль малювання, який використовується на цьому майстер-класі з малювання, визначений зв'язком з таблицею Drawstyle через зовнішній ключ;

– drawworkshop_meta – текстовий опис мети проведення майстер-класу з малювання;

– drawworkshop_text – текстовий рекламний опис майстер-класу з малювання;

– drawworkshop_metaguests – цільова кількість учасників, на яку розраховане проведення таких майстер-класів з малювання.

БД має таблицю Drawworkshopphoto, яка визначає фотографії для ілюстрації майстер-класів з малювання через поля:

– id – ідентифікатор фотографії для ілюстрації майстер-класів з малювання;

– drawworkshop – майстер-клас з малювання, який ілюструє ця фотографія, визначений зв'язком з таблицею Drawworkshop через зовнішній ключ;

– photo_data – фотографія, яка ілюструє майстер-клас з малювання (самі дані фотографії);

– photo_poster – позначення того, чи ця фотографія, яка ілюструє майстер-клас з малювання, використовується як основна (перша фотографія для ілюстрації, чи ні).

БД має таблицю Drawworkshopvideo, яка визначає відео для ілюстрації майстер-класів з малювання через поля:

– id – ідентифікатор відео для ілюстрації майстер-класів з малювання;

- drawworkshop – майстер-клас з малювання, який ілюструє це відео, визначений зв'язком з таблицею Drawworkshop через зовнішній ключ;

- video_data – відео, яке ілюструє майстер-клас з малювання (самі дані відео);

- video_text – текстова примітка до відео, яке ілюструє майстер-клас з малювання.

БД має таблицю Drawworkshopevent, яка визначає конкретне проведення майстер-класу з малювання через поля:

- id – ідентифікатор конкретного проведення майстер-класу з малювання;

- drawworkshop – базове інформування про проведення майстер-класу з малювання, до якого відноситься це конкретне проведення майстер-класу з малювання, визначене зв'язком з таблицею Drawworkshop через зовнішній ключ;

- workshoplocation – місце проведення цього майстер-класу з малювання, визначене зв'язком з таблицею Workshoplocation через зовнішній ключ;

- drawworkshopevent_startdate – дата і час початку проведення цього майстер-класу з малювання;

- drawworkshopevent_enddate – дата і час завершення проведення цього майстер-класу з малювання;

- drawworkshopevent_maxguests – максимальна кількість відвідувачів цього майстер-класу з малювання;

- drawworkshopevent_currguests – поточна кількість заброньованих місць на цьому майстер-класі з малювання;

- drawworkshopevent_text – текстова презентація конкретного проведення майстер-класу з малювання;

- drawworkshopevent_state – стан підтримки проведення цього майстер-класу з малювання: запланований, заповнений, скасований, проведений.

БД має таблицю Drawworkshopeventrequest, яка визначає запрошення відвідувача на безкоштовне відвідування майстер-класу з малювання через поля:

- id – ідентифікатор запрошення відвідувача на безкоштовне відвідування майстер-класу з малювання;
- drawworkshopevent – проведення майстер-класу з малювання, на безкоштовне відвідування якого майстер запросив відвідувача, визначене зв'язком з таблицею Drawworkshopevent через зовнішній ключ;
- workshopguest – відвідувач, якого майстер запросив на безкоштовне відвідування майстер-класу з малювання, визначений зв'язком з таблицею Workshopguest через зовнішній ключ;
- reaction – відповідь відвідувача на запрошення на безкоштовне відвідування майстер-класу з малювання: погодження, відхилення або відсутність відповіді.

БД має таблицю Drawworkshopeventbooking, яка визначає бронювання місць на майстер-класах з малювання через поля:

- id – ідентифікатор бронювання місця на майстер-класі з малювання;
- drawworkshopevent – проведення майстер-класу з малювання, на яке забронював місце відвідувач, визначене зв'язком з таблицею Drawworkshopevent через зовнішній ключ;
- workshopguest – відвідувач, який забронював місце на майстер-класі з малювання, визначений зв'язком з таблицею Workshopguest через зовнішній ключ;
- drawworkshopeventbooking_state – стан бронювання місця на майстер-класі з малювання: оформлене, скасоване відвідувачем, скасоване майстром, скасоване менеджером.

БД має таблицю Drawworkshopeventphoto, яка визначає фотографії для ілюстрації конкретного майстер-класу з малювання через поля:

- id – ідентифікатор фотографії для ілюстрації конкретного майстер-класу з малювання;

- drawworkshopevent – проведення майстер-класу з малювання, яке ілюструє ця фотографія, визначене зв'язком з таблицею Drawworkshopevent через зовнішній ключ;

- photo_data – фотографія, яка ілюструє це проведення майстер-класу з малювання (самі дані фотографії);

- photo_poster – позначення того, чи ця фотографія, яка ілюструє проведення майстер-класу з малювання, використовується як основна (перша фотографія для ілюстрації, чи ні).

БД має таблицю Drawworkshopeventreport, яка визначає звіти про проведення майстер-класів з малювання через поля:

- id – ідентифікатор звіту про проведення майстер-класу з малювання;

- drawworkshopevent – проведення майстер-класу з малювання, за яким оформлено звіт, визначене зв'язком з таблицею Drawworkshopevent через зовнішній ключ;

- author_djangoaccount – автор звіту про проведення майстер-класу з малювання, визначений своїм акаунтом в ФДВР Django (це може бути і майстер, і відвідувач) через зовнішній ключ до таблиці User;

- drawworkshopeventreport_date – дата створення звіту про проведення майстер-класу з малювання;

- drawworkshopeventreport_text – текст звіту про проведення майстер-класу з малювання.

БД має таблицю Drawworkshopeventreportphoto, яка визначає фотографії до звіту про проведення майстер-класу з малювання через поля:

- id – ідентифікатор фотографії до звіту про проведення майстер-класу з малювання;

- drawworkshopeventreport – звіт про проведення майстер-класу з малювання, до якого прикріплено цю фотографію, визначений зв'язком з таблицею Drawworkshopeventreport через зовнішній ключ;

- photo_data – фотографія до звіту про проведення майстер-класу з малювання (самі дані фотографії).

БД має таблицю Drawworkshopeventreportvideo, яка визначає відео для ілюстрації звіту конкретного майстер-класу з малювання через поля:

- id – ідентифікатор відео для ілюстрації конкретного майстер-класу з малювання;
- drawworkshopeventreport – звіт про проведення майстер-класу з малювання, до якого прикріплено це відео, визначене зв'язком з таблицею Drawworkshopeventreport через зовнішній ключ;
- video_data – відео до звіту про проведення майстер-класу з малювання;
- video_text – текстова примітка до відео до звіту про проведення майстер-класу з малювання.

БД має таблицю Drawmaster, яка визначає дані майстра з малювання через поля:

- id – ідентифікатор майстра з малювання;
- user_djangoaccount – акаунт користувача в ФДВР Django, що використовується цим майстром з малювання, визначений зв'язком з таблицею User через зовнішній ключ;
- drawmaster_text – текстове представлення майстра з малювання;
- workshoplocationcity – місто, в якому знаходиться цей майстер з малювання, визначене зв'язком з таблицею Workshoplocationcity через зовнішній ключ.

БД має таблицю Drawmasterphoto, яка визначає фотографії майстра з малювання через поля:

- id – ідентифікатор фотографії майстра з малювання;
- drawmaster – майстер з малювання, який показаний на фотографії, визначений зв'язком з таблицею Drawmaster через зовнішній ключ;
- photo_data – фотографія майстра з малювання (самі дані фотографії);
- photo_poster – позначення того, чи ця фотографія майстра з малювання використовується як основна (перша фотографія для ілюстрації, чи ні).

БД має таблицю Drawworkshopvoting, яка визначає голосування за проведення майстер-класу з малювання через поля:

- id – ідентифікатор голосування за проведення майстер-класу з малювання;
- drawworkshop – майстер-клас з малювання, визначений своїм базовим інформуванням, за яким відкрито це голосування за його проведення, встановлений зв'язком з таблицею Drawworkshop через зовнішній ключ;
- workshoplocation – місце проведення майстер-класу з малювання, яке пропонується за цим голосуванням, встановлене зв'язком з таблицею Workshoplocation через зовнішній ключ;
- drawworkshopvoting_text – текст, який додається до голосування за проведення майстер-класу з малювання;
- drawworkshopvoting_date – дата і час пропонованого проведення майстер-класу з малювання, за яким відбувається голосування;
- drawworkshopvoting_minvotes – мінімальна кількість голосів, яку має отримати майстер-клас для його підтримки на голосуванні;
- drawworkshopvoting_stopvotes – кількість голосів, за досягнення якої за проведення цього майстер-класу з малювання голосування закривається;
- drawworkshopvoting_stopdate – дата і час, коли має бути завершено голосування за проведення майстер-класу з малювання.

БД має таблицю Drawworkshopvotingmark, яка визначає самі голоси в межах голосування за проведення майстер-класу з малювання через поля:

- id – ідентифікатор голосу в межах голосування за проведення майстер-класу з малювання;
- drawworkshopvoting – голосування за проведення майстер-класу з малювання, за яким віддано цей голос, визначений зв'язком з таблицею Drawworkshopvoting через зовнішній ключ;
- workshopguest – відвідувач, який віддав свій голос за проведення цього майстер-класу з малювання в межах голосування, визначений зв'язком з таблицею Workshopguest через зовнішній ключ.

БД має таблицю Workshoplocationcity, яка визначає міста, в яких можуть проводитись майстер-класи з малювання, через поля:

- id – ідентифікатор міста, в якому можуть проводитись майстер-класи з малювання;
- city_name – назва міста, в якому може проводитись майстер-клас з малювання.

БД має таблицю Workshoplocation, яка визначає місця проведення майстер-класів з малювання через поля:

- id – ідентифікатор місця проведення майстер-класу з малювання;
- workshoplocation_city – місто, в якому знаходиться це місце проведення майстер-класу з малювання, визначене зв'язком з таблицею Workshoplocationcity через зовнішній ключ;
- workshoplocation_title – назва місця проведення майстер-класу з малювання;
- workshoplocation_address – адреса, за якою знаходиться це місце проведення майстер-класу з малювання;
- workshoplocation_url – вебадреса представлення цього місця для проведення майстер-класу з малювання;
- workshoplocation_phone – номер телефону для зв'язку з цим місцем проведення майстер-класу з малювання.

БД має таблицю Workshoplocationphoto, яка визначає фотографії місця проведення майстер-класу з малювання через поля:

- id – ідентифікатор фотографії місця проведення майстер-класу з малювання;
- workshoplocation – місце проведення майстер-класу з малювання, яке показано на фотографії, визначене зв'язком з таблицею Workshoplocation через зовнішній ключ;
- photo_data – фотографія місця проведення майстер-класу з малювання.

БД має таблицю Workshopguest, яка визначає відвідувачів майстер-класів з малювання через поля:

- id – ідентифікатор відвідувача майстер-класів з малювання;
- user_djangoaccount – відвідувач майстер-класів з малювання, визначений своїм акаунтом в ФДВР Django через зовнішній ключ до таблиці User;
- workshopguest_phone – номер телефону для зв'язку з відвідувачем майстер-класів з малювання;
- workshopguest_photo – фотографія відвідувача майстер-класів з малювання;
- workshopguest_estimationtotal – середня оцінка, виставлена цьому відвідувачу майстер-класів з малювання;
- workshopguest_text – текстове представлення відвідувача майстер-класу з малювання.

БД має таблицю Drawworkshopeventestimation, яка визначає оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання через поля:

- id – ідентифікатор оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання;
- drawworkshopeventbooking – бронювання місця на майстер-класі з малювання, за яким виконано це оцінювання, визначене зв'язком з таблицею Drawworkshopeventbooking через зовнішній ключ;
- drawworkshopeventestimation_date – дата оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання;
- drawworkshopeventestimation_text – текст оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання;

2.4 Висновки за розділом 2

За результатами розділу 2 програма підтримки проведення майстер-класів з малювання була запроєктована, включаючи прийняття рішень щодо вибору основних технологій, включаючи мову програмування Python, ФДВР Django, систему керування БД PostgreSQL, щодо моделювання вимог до програми підтримки проведення майстер-класів з малювання, структури БД.

3 РОЗРОБКА ПРОГРАМИ

3.1 Послідовність роботи з програмою підтримки проведення майстер-класів з малювання

Робота з програмою підтримки проведення майстер-класів з малювання розпочинається з пошуку серед майстрів з малювання (рис. 3.1).

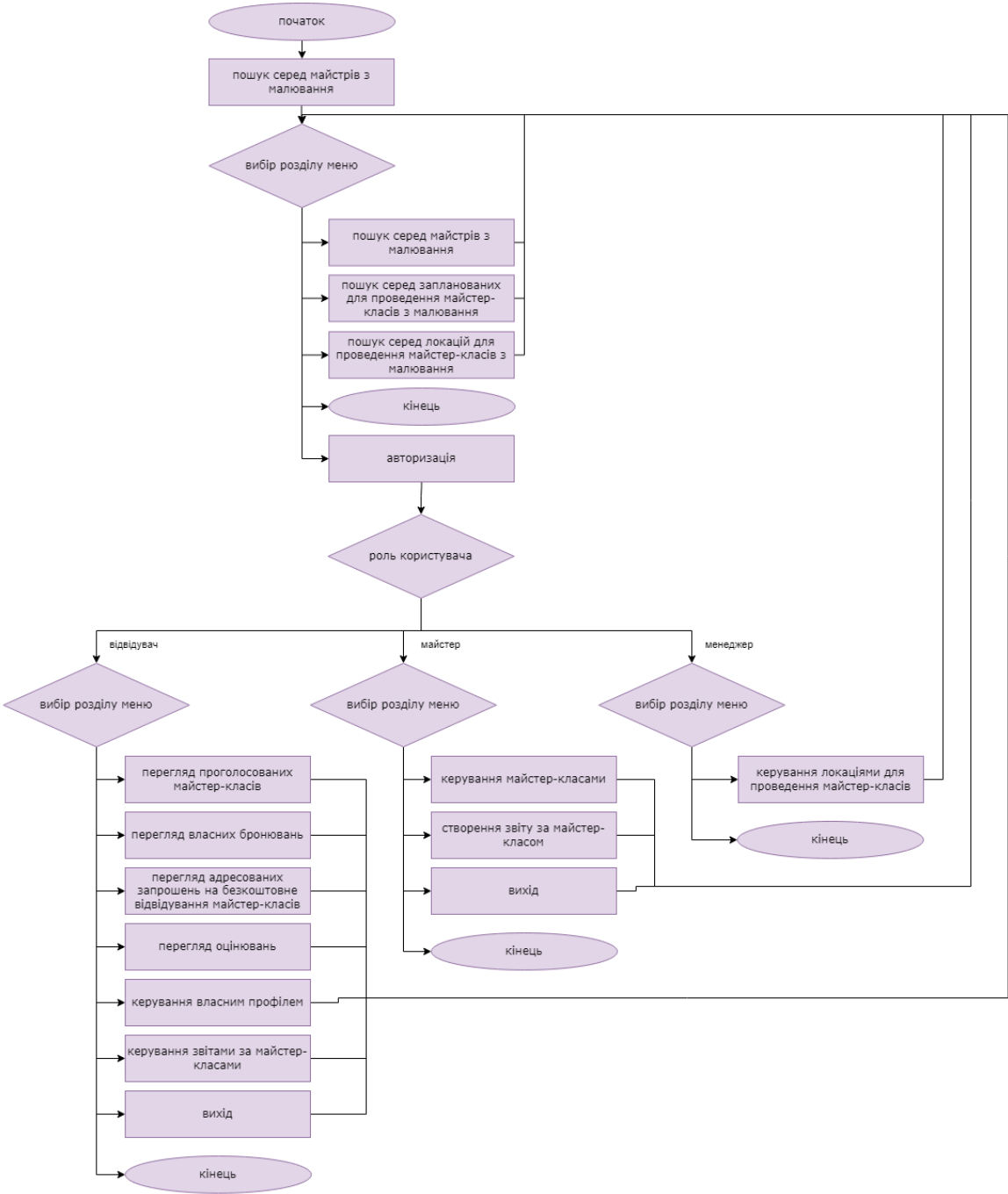


Рисунок 3.1 – Схема функціонування програми підтримки проведення майстер-класів з малювання

Після цього користувач може обрати потрібний розділ меню з доступних або виконати авторизацію. Після авторизації йому доступні розділи меню в залежності від ролі, що включає менеджера, майстра з малювання та відвідувача, а також ті розділи, які доступні неавторизованому користувачу.

3.2 Рішення з реалізації програми підтримки проведення майстер-класів з малювання

У розробці вебзастосунку забезпечено певну етапність у відповідності зі структурою застосунку на ФДБР Django. Опис розпочато зі створених моделей [8].

Модель Drawstyle визначає стилі малювання, включаючи до реалізації тільки поле drawstyle_name, яке задає назву стилю малювання за типом CharField.

Модель Drawworkshop визначає базове інформування про майстер-класи з малювання, включаючи до реалізації наступні поля.

Поле drawworkshop_title задає загальну назву майстер-класу з малювання за типом CharField.

Поле drawmaster задає майстра, який проводить цей майстер-клас з малювання, визначений типом ForeignKey зі зв'язком з моделлю Drawmaster.

Поле draw_style задає стиль малювання, який використовується на цьому майстер-класі з малювання, визначений типом ForeignKey зі зв'язком з моделлю Drawstyle.

Поле drawworkshop_meta задає текстовий опис мети проведення майстер-класу з малювання за типом CharField.

Поле drawworkshop_text задає текстовий рекламний опис майстер-класу з малювання за типом TextField.

Поле drawworkshop_metaguests задає цільову кількість учасників, на яку розраховане проведення таких майстер-класів з малювання, за типом PositiveSmallIntegerField.

Модель Drawworkshopphoto визначає фотографії для ілюстрації майстер-класів з малювання, включаючи до реалізації наступні поля.

Поле drawworkshop задає майстер-клас з малювання, який ілюструє ця фотографія, визначений типом ForeignKey зі зв'язком з моделлю Drawworkshop.

Поле photo_data задає фотографію, яка ілюструє майстер-клас з малювання (самі дані фотографії), за типом ImageField.

Поле photo_poster задає позначення того, чи ця фотографія, яка ілюструє майстер-клас з малювання, використовується як основна (перша фотографія для ілюстрації, чи ні), за типом BooleanField.

Модель Drawworkshopvideo визначає відео для ілюстрації майстер-класів з малювання, включаючи до реалізації наступні поля.

Поле drawworkshop задає майстер-клас з малювання, який ілюструє це відео, визначений типом ForeignKey зі зв'язком з моделлю Drawworkshop.

Поле video_data задає відео, яке ілюструє майстер-клас з малювання (самі дані відео), за типом URLField.

Поле video_text задає текстову примітку до відео, яке ілюструє майстер-клас з малювання, за типом CharField.

Модель Drawworkshopevent визначає конкретне проведення майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле drawworkshop задає базове інформування про проведення майстер-класу з малювання, до якого відноситься це конкретне проведення майстер-класу з малювання, визначене типом ForeignKey зі зв'язком з моделлю Drawworkshop.

Поле workshoplocation задає місце проведення цього майстер-класу з малювання, визначене типом ForeignKey зі зв'язком з моделлю Workshoplocation.

Поле drawworkshopevent_startdate задає дату і час початку проведення цього майстер-класу з малювання за типом DateTimeField.

Поле `drawworkshopevent_enddate` задає дату і час завершення проведення цього майстер-класу з малювання за типом `DateTimeField`.

Поле `drawworkshopevent_maxguests` задає максимальну кількість відвідувачів цього майстер-класу з малювання за типом `PositiveSmallIntegerField`.

Поле `drawworkshopevent_currguests` задає поточну кількість заброньованих місць на цьому майстер-класі з малювання за типом `PositiveSmallIntegerField`.

Поле `drawworkshopevent_text` задає текстову презентацію конкретного проведення майстер-класу з малювання за типом `TextField`.

Поле `drawworkshopevent_state` задає стан підтримки проведення цього майстер-класу з малювання за типом `CharField`.

Модель `Drawworkshopeventrequest` визначає запрошення відвідувача на безкоштовне відвідування майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshopevent` задає проведення майстер-класу з малювання, на безкоштовне відвідування якого майстер запросив відвідувача, визначене типом `ForeignKey` зі зв'язком з моделлю `Drawworkshopevent`.

Поле `workshopguest` задає відвідувача, якого майстер запросив на безкоштовне відвідування майстер-класу з малювання, визначеного типом `ForeignKey` зі зв'язком з моделлю `Workshopguest`.

Поле `reaction` задає відповідь відвідувача на запрошення на безкоштовне відвідування майстер-класу з малювання за типом `PositiveSmallIntegerField`.

Модель `Drawworkshopeventbooking` визначає бронювання місць на майстер-класах з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshopevent` задає проведення майстер-класу з малювання, на яке забронював місце відвідувач, визначене типом `ForeignKey` зі зв'язком з моделлю `Drawworkshopevent`.

Поле `workshopguest` задає відвідувача, який забронював місце на майстер-класі з малювання, визначеного типом `ForeignKey` зі зв'язком з моделлю `Workshopguest`.

Поле `drawworkshopeventbooking_state` задає стан бронювання місця на майстер-класі з малювання за типом `CharField`.

Модель `Drawworkshopeventphoto` визначає фотографії для ілюстрації конкретного майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshopevent` задає проведення майстер-класу з малювання, яке ілюструє ця фотографія, визначене типом `ForeignKey` зі зв'язком з моделлю `Drawworkshopevent`.

Поле `photo_data` задає фотографію, яка ілюструє це проведення майстер-класу з малювання (самі дані фотографії), за типом `ImageField`.

Поле `photo_poster` задає позначення того, чи ця фотографія, яка ілюструє проведення майстер-класу з малювання, використовується як основна (перша фотографія для ілюстрації, чи ні), за типом `BooleanField`.

Модель `Drawworkshopeventreport` визначає звіти про проведення майстер-класів з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshopevent` задає проведення майстер-класу з малювання, за яким оформлено звіт, визначене типом `ForeignKey` зі зв'язком з моделлю `Drawworkshopevent`.

Поле `author_djangoaccount` задає автора звіту про проведення майстер-класу з малювання, визначеного своїм акаунтом в ФДБР Django (це може бути і майстер, і відвідувач) за типом `ForeignKey` зі зв'язком з моделлю `User`.

Поле `drawworkshopeventreport_date` задає дату створення звіту про проведення майстер-класу з малювання за типом `DateTimeField`.

Поле `drawworkshopeventreport_text` задає текст звіту про проведення майстер-класу з малювання за типом `TextField`.

Модель Drawworkshopeventreportphoto визначає фотографії до звіту про проведення майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле drawworkshopeventreport задає звіт про проведення майстер-класу з малювання, до якого прикріплено цю фотографію, визначений типом ForeignKey зі зв'язком з моделлю Drawworkshopeventreport.

Поле photo_data задає фотографію до звіту про проведення майстер-класу з малювання (самі дані фотографії) за типом ImageField.

Модель Drawworkshopeventreportvideo визначає відео для ілюстрації звіту конкретного майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле drawworkshopeventreport задає звіт про проведення майстер-класу з малювання, до якого прикріплено це відео, визначений типом ForeignKey зі зв'язком з моделлю Drawworkshopeventreport.

Поле video_data задає відео до звіту про проведення майстер-класу з малювання за типом URLField.

Поле video_text задає текстову примітку до відео до звіту про проведення майстер-класу з малювання за типом CharField.

Модель Drawmaster визначає дані майстра з малювання, включаючи до реалізації наступні поля.

Поле user_djangoaccount задає акаунт користувача в ФДБР Django, що використовується цим майстром з малювання, визначений типом OneToOneField зі зв'язком з моделлю User.

Поле drawmaster_text задає текстове представлення майстра з малювання за типом TextField.

Поле workshoplocationcity задає місто, в якому знаходиться цей майстер з малювання, визначене типом ForeignKey зі зв'язком з моделлю Workshoplocationcity.

Модель Drawmasterphoto визначає фотографії майстра з малювання, включаючи до реалізації наступні поля.

Поле `drawmaster` задає майстра з малювання, який показаний на фотографії, визначеного типом `ForeignKey` зі зв'язком з моделлю `Drawmaster`.

Поле `photo_data` задає фотографію майстра з малювання (самі дані фотографії) за типом `ImageField`.

Поле `photo_poster` задає позначення того, чи ця фотографія майстра з малювання використовується як основна (перша фотографія для ілюстрації, чи ні), за типом `BooleanField`.

Модель `Drawworkshopvoting` визначає голосування за проведення майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshop` задає майстер-клас з малювання, визначений своїм базовим інформуванням, за яким відкрито це голосування за його проведення, встановлений типом `ForeignKey` зі зв'язком з моделлю `Drawworkshop`.

Поле `workshoplocation` задає місце проведення майстер-класу з малювання, яке пропонується за цим голосуванням, визначене типом `ForeignKey` зі зв'язком з моделлю `Workshoplocation`.

Поле `drawworkshopvoting_text` задає текст, який додається до голосування за проведення майстер-класу з малювання, за типом `TextField`.

Поле `drawworkshopvoting_date` задає дату і час пропонованого проведення майстер-класу з малювання, за яким відбувається голосування, за типом `DateTimeField`.

Поле `drawworkshopvoting_minvotes` задає мінімальну кількість голосів, яку має отримати майстер-клас для його підтримки на голосуванні, за типом `PositiveSmallIntegerField`.

Поле `drawworkshopvoting_stopvotes` задає кількість голосів, за досягнення якої за проведення цього майстер-класу з малювання голосування закривається, за типом `PositiveSmallIntegerField`.

Поле `drawworkshopvoting_stopdate` задає дату і час, коли має бути завершено голосування за проведення майстер-класу з малювання, за типом `DateTimeField`.

Модель Drawworkshopvotingmark визначає самі голоси в межах голосування за проведення майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле drawworkshopvoting задає голосування за проведення майстер-класу з малювання, за яким віддано цей голос, визначене типом ForeignKey зі зв'язком з моделлю Drawworkshopvoting.

Поле workshopguest задає відвідувача, який віддав свій голос за проведення цього майстер-класу з малювання в межах голосування, визначеного типом ForeignKey зі зв'язком з моделлю Workshopguest.

Модель Workshoplocationcity визначає міста, в яких можуть проводитись майстер-класи з малювання, включаючи до реалізації наступні поля.

Поле city_name задає назву міста, в якому може проводитись майстер-клас з малювання, за типом CharField.

Модель Workshoplocation визначає місця проведення майстер-класів з малювання, включаючи до реалізації наступні поля.

Поле workshoplocation_city задає місто, в якому знаходиться це місце проведення майстер-класу з малювання, визначене типом ForeignKey зі зв'язком з моделлю Workshoplocationcity.

Поле workshoplocation_title задає назву місця проведення майстер-класу з малювання за типом CharField.

Поле workshoplocation_address задає адресу, за якою знаходиться це місце проведення майстер-класу з малювання, за типом CharField.

Поле workshoplocation_url задає вебадресу представлення цього місця для проведення майстер-класу з малювання за типом URLField.

Поле workshoplocation_phone задає номер телефону для зв'язку з цим місцем проведення майстер-класу з малювання за типом CharField.

Модель Workshoplocationphoto визначає фотографії місця проведення майстер-класу з малювання, включаючи до реалізації наступні поля.

Поле `workshoplocation` задає місце проведення майстер-класу з малювання, яке показано на фотографії, визначене типом `ForeignKey` зі зв'язком з моделлю `Workshoplocation`.

Поле `photo_data` задає фотографію місця проведення майстер-класу з малювання за типом `ImageField`.

Модель `Workshopguest` визначає відвідувачів майстер-класів з малювання, включаючи до реалізації наступні поля.

Поле `user_djangoaccount` задає відвідувача майстер-класів з малювання, визначеного своїм акаунтом в ФДБП Django та типом `OneToOneField` зі зв'язком з моделлю `User`.

Поле `workshopguest_phone` задає номер телефону для зв'язку з відвідувачем майстер-класів з малювання за типом `CharField`.

Поле `workshopguest_photo` задає фотографію відвідувача майстер-класів з малювання за типом `ImageField`.

Поле `workshopguest_estimationtotal` задає середню оцінку, виставлену цьому відвідувачу майстер-класів з малювання, за типом `DecimalField`.

Поле `workshopguest_text` задає текстове представлення відвідувача майстер-класу з малювання за типом `CharField`.

Модель `Drawworkshopeventestimation` визначає оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання, включаючи до реалізації наступні поля.

Поле `drawworkshopeventbooking` задає бронювання місця на майстер-класі з малювання, за яким виконано це оцінювання, визначене типом `ForeignKey` зі зв'язком з моделлю `Drawworkshopeventbooking`.

Поле `drawworkshopeventestimation_date` задає дату оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання за типом `DateTimeField`.

Поле `drawworkshopeventestimation_text` задає текст оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання за типом `CharField`.

Поле `drawworkshopeventestimation_score` задає бали оцінювання майстер-класу з малювання відвідувачем або відвідувача майстром з малювання за результатами участі в майстер-класі з малювання за типом `PositiveSmallIntegerField`.

Поле `drawworkshopeventestimation_type` задає тип оцінювання майстер-класу з малювання: оцінювання відвідувачем майстер-класу з малювання або оцінювання відвідувача майстром з малювання за результатами участі в майстер-класі з малювання за типом `BooleanField`.

Для реалізації вебзастосунку як Django-застосунку було створено ряд функцій, які відповідають представленням [9], що реалізують функції програми на основі застосування шаблонів вебсторінок, збудованим за допомогою шаблону [10].

Перегляд власних базових інформувань про майстер-класи з малювання реалізовано функцією `OwnDrawworkshopsView`. Усі дані про майстер-клас знаходяться за ключем `owndrawworkshops` у контексті даних, переданому в шаблон вебсторінки `owndrawworkshops.html`.

Перегляд сторінки майстра з базовими інформуваннями про майстер-класи з малювання реалізовано функцією `MasterDrawworkshopsView` з параметром `drawmaster_login`, що задає логін майстра з малювання. Дані майстер-класів, сформовані на сторінках порціями по 10 елементів, передаються в шаблон вебсторінки `masterdrawworkshops.html` через ключ `masterdrawworkshops`.

Перегляд майстер-класів, за проведення яких проголосував користувач, реалізовано функцією `OwnDrawworkshopVotesView`. Логін користувача-відвідувача майстер-класів з малювання передається через параметр `request` у складі атрибуту `user`, що є стандартним для всіх функцій представлень.

Результати майстер-класів, за проведення яких проголосував користувач, передаються в сторінку `owndrawworkshopvotes.html` через ключ `owndrawworkshopvotes`.

Перегляд запланованих для проведення майстер-класів з малювання за майстром реалізовано функцією `PlannedMasterDrawworkshopeventsView`. Параметр `drawmaster_login` задає логін майстра. Результати передаються в шаблон вебсторінки `planmasterdrawworkshopevents.html` через ключ `masterdrawworkshopevents` з майстер-класами з малювання, які заплановані, у контексті даних.

Перегляд проведених майстер-класів з малювання за майстром реалізовано функцією `FinishedMasterDrawworkshopeventsView` з параметром `drawmaster_login`, який задає логін майстра з малювання. Результати передаються в шаблон вебсторінки `finishmasterdrawworkshopevents.html` через ключ `masterdrawworkshopevents`, який задає проведені майстер-класи з малювання, тобто такі, які знаходяться в стані проведених та за часом завершення мають задане значення, яке є ранішим за поточний час.

Перегляд запланованих для проведення майстер-класів з малювання за локацією реалізовано функцією `PlannedLocationDrawworkshopeventsView` з параметром `workshoplocation_id`, який задає ідентифікатор локації проведення майстер класів з малювання. Результати передаються в шаблон вебсторінки `planlocationdrawworkshopevents.html` через ключ `masterdrawworkshopevents`, який задає заплановані майстер-класи з малювання, тобто такі, які знаходяться в стані запланованих або заповнених та за часом початку мають задане значення, яке є пізнішим за поточний час.

Перегляд проведених майстер-класів з малювання за локацією реалізовано функцією `FinishedLocationDrawworkshopeventsView` з параметром `workshoplocation_id`, який задає ідентифікатор локації проведення майстер класів з малювання. Результати передаються в шаблон вебсторінки `finishedlocationdrawworkshopevents.html` через ключ `masterdrawworkshopevents`, який задає проведені майстер-класи з малювання,

тобто такі, які знаходяться в стані проведених та за часом завершення мають задане значення, яке є ранішим за поточний час.

Перегляд оцінювань майстер-класу з малювання реалізовано функцією `DrawworkshopeventEstimationsView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання. Результати передаються в шаблон вебсторінки `workshopestimations.html` через ключ `estimations`, куди формуються 10 оцінювань майстер-класу з малювання, які розділяються порціями за значенням, заданим у ключі `page` GET-запиту.

Перегляд оцінювань за майстер-класами майстра з малювання реалізовано функцією `MasterDrawworkshopeventEstimationsView` з параметром `drawmaster_login`, який задає логін майстра з малювання. Результати передаються в шаблон вебсторінки `masterworkshopestimations.html` через ключ `estimations`, куди формуються 10 оцінювань майстер-класу з малювання, які розділяються порціями за значенням, заданим у ключі `page` GET-запиту.

Перегляд звітів за майстер-класом з малювання реалізовано функцією `DrawworkshopeventReportsView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання. Результати передаються в шаблон вебсторінки `workshopeventreports.html` через ключ `reports`, куди формуються 10 звітів за майстер-класом з малювання, які розділяються порціями за значенням, заданим у ключі `page` GET-запиту.

Перегляд звітів за майстер-класами з малювання відвідувача реалізовано функцією `WorkshopguestReportsView` з параметром `workshopguest_login`, який задає логін відвідувача. Результати передаються в шаблон вебсторінки `drawworkshopevent_reports` через ключ `reports`, куди формуються 10 звітів за майстер-класом з малювання, які розділяються порціями за значенням, заданим у ключі `page` GET-запиту.

Перегляд оцінювань відвідувача майстер-класів з малювання реалізовано функцією `WorkshopguestEstimationsView` з параметром `workshopguest_login`, який задає логін відвідувача. Результати передаються в шаблон вебсторінки `guestworkshopestimations.html` через ключ `estimations`,

куди формуються 10 оцінювань відвідувача майстер-класів, які розділяються порціями за значенням, заданим у ключі page GET-запиту.

Перегляд сторінки відвідувача майстер-класів з малювання реалізовано функцією `WorkshopguestView` з параметром `workshopguest_login`, який задає логін відвідувача. Результати передаються в шаблон вебсторінки `guest.html` через ключ `estimations` з 10 останніми оцінюваннями відвідувача та `reports` з 10 останніми звітами за майстер-класами з малювання цього відвідувача.

Перегляд власних бронювань місць на майстер-класах з малювання реалізовано функцією `OwnBookedDrawworkshopeventsView`. Результати передаються в шаблон вебсторінки `bookedworhsopevents.html`, в який в контексті даних за ключем `drawworkshopevents` передаються дані проведення майстер-класів з малювання, за якими виконав бронювання відвідувач майстер-класів з малювання, який авторизувався.

Перегляд бронювань за майстер-класом з малювання реалізовано функцією `DrawworkshopeventBookingsView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання. Результати передаються в шаблон вебсторінки `workshopeventbookings.html` в складі контексту даних з ключем `drawworkshopevents`, за яким містяться бронювання за цим майстер-класом з малювання.

Перегляд публікації проведення майстер-класу з малювання реалізовано функцією `DrawworkshopEventView` з параметром `drawworkshopevent_id`, який задає ідентифікатор цього проведення майстер-класу з малювання. Результати передаються в шаблон вебсторінки `workshopevent.html` з ключем `drawworkshopevent`, за яким знаходяться всі дані публікації проведення майстер-класу з малювання.

Перегляд оформлених запрошень на безкоштовне відвідування власного майстер-класу реалізовано функцією `DrawworkshopeventrequestsView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання. Результати

передаються в шаблон вебсторінки `drawworkshopeventrequests.html` в складі контексту даних з ключем `drawworkshopevents`, за яким містяться запрошення за цим майстер-класом з малювання.

Перегляд адресованих для безкоштовного відвідування запрошень на майстер-класи з малювання реалізовано функцією `OwnDrawworkshopeventrequestsView`. Результати передаються в шаблон вебсторінки `owndrawworkshopeventrequests.html` в складі контексту даних з ключем `drawworkshopevents`, за яким містяться запрошення на відвідування майстер-класів з малювання, адресовані до відвідувача, який авторизувався.

Перегляд базового інформування про майстер-клас з малювання реалізовано функцією `DrawworkshopView` з параметром `drawworkshop_id`, який задає ідентифікатор базового інформування про майстер-клас з малювання. Результати передаються в шаблон вебсторінки `drawworkshop.html` за ключем `drawworkshop`, де містяться дані базового інформування про майстер-клас з малювання.

Перегляд голосування за проведення майстер-класу з малювання реалізовано функцією `DrawworkshopvotingView` з параметром `drawworkshopvoting_id`, який задає ідентифікатор голосування за проведення майстер-класу з малювання. Результати передаються в шаблон вебсторінки `drawworkshopvoting.html` за ключем `drawworkshopvoting`, де містяться дані голосування за проведення майстер-класу з малювання.

Перегляд даних локації для проведення майстер-класів з малювання реалізовано функцією `WorkshoplocationView` з параметром `workshoplocation_id`, який задає ідентифікатор локації для проведення майстер-класів з малювання. Результати передаються в шаблон вебсторінки `workshoplocation.html` за ключем `workshoplocation`, де містяться дані локації для проведення майстер-класів з малювання.

Перегляд звіту за проведеним майстер-класом з малювання реалізовано функцією `DrawworkshopeventreportView` з параметром `drawworkshopeventreport_id`, який задає ідентифікатор звіту за проведеним

майстер-класом з малювання. Результати передаються в шаблон вебсторінки `drawworkshopeventreport.html` за ключем `drawworkshopeventreport`, де містяться дані звіту за проведенням майстер-класом з малювання.

Створення базового інформування про майстер-клас з малювання реалізовано функцією `DrawworkshopeventreportCreateView`, яка головним чином працює за допомогою класу `DrawworkshopeventreportForm`, який опрацьовує форму з даними базового інформування про майстер-клас з малювання. Оновлення базового інформування про майстер-клас з малювання реалізовано функцією `DrawworkshopeventreportEditView`, яка містить параметр `drawworkshopeventreport_id`, який задає ідентифікатор звіту за проведенням майстер-класом з малювання, але головним чином працює на основі класу `DrawworkshopeventreportForm`.

Публікація проведення майстер-класу з малювання реалізовано функцією `DrawworkshopeventCreateView`, яка головним чином працює за допомогою класу `DrawworkshopeventForm`, який опрацьовує форму з даними проведення майстер-класу з малювання. Оновлення даних проведення майстер-класу з малювання реалізовано функцією `DrawworkshopeventEditView`, яка містить параметр `drawworkshopevent_id`, який задає ідентифікатор майстер-класу з малювання, але головним чином працює на основі класу `DrawworkshopeventForm`.

Вилучення проведення майстер-класу з малювання реалізовано функцією `DrawworkshopeventDeleteView`, яка містить параметр `drawworkshopevent_id`, який задає ідентифікатор майстер-класу з малювання.

Оновлення даних майстра з малювання реалізовано функцією `DrawmasterView`, яка головним чином працює за допомогою класу `DrawmasterForm`, який опрацьовує форму з даними майстра з малювання.

Відкриття голосування за проведення майстер-класу з малювання реалізовано функцією `DrawworkshopvotingCreateView`, яка головним чином працює за допомогою класу `DrawworkshopvotingForm`, який опрацьовує форму з даними голосування за проведення майстер-класу з малювання.

Голосування за проведення майстер-класу з малювання реалізовано функцією `DrawworkshopvotingmarkCreateView` з параметром `drawworkshopvoting_id`, який задає ідентифікатор голосування за проведення майстер-класу з малювання.

Скасування голосування за проведення майстер-класу з малювання реалізовано функцією `DrawworkshopvotingmarkCancelView` з параметром `drawworkshopvoting_id`, який задає ідентифікатор голосування за проведення майстер-класу з малювання.

Додавання локації для проведення майстер-класів з малювання реалізовано функцією `WorkshoplocationCreateView`, яка головним чином працює за допомогою класу `WorkshoplocationForm`, який опрацьовує форму з даними локації для проведення майстер-класів з малювання. У функції відбувається перевірка того, що користувач, який авторизувався, є менеджером. Оновлення даних локації для проведення майстер-класів з малювання реалізовано функцією `WorkshoplocationEditView`, яка містить параметр `workshoplocation_id`, який задає ідентифікатор локації для проведення майстер-класів з малювання, але головним чином працює на основі класу `WorkshoplocationForm`.

Додавання звіту за проведеним майстер-класом з малювання реалізовано функцією `DrawworkshopeventreportCreateView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання. Ця функція працює за допомогою класу `DrawworkshopeventreportForm`, який опрацьовує форму з даними звіту за проведеним майстер-класом з малювання. Оновлення звіту за проведеним майстер-класом з малювання реалізовано функцією `DrawworkshopeventreportEditView`, яка містить параметр `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання, але головним чином працює на основі класу `DrawworkshopeventreportForm`.

Оцінювання проведеного майстер-класу з малювання реалізовано функцією `DrawworkshopeventestimationView`, яка має параметр `drawworkshopeventbooking_id`, який задає ідентифікатор бронювання місця на майстер-класі з малювання. В основі роботи знаходиться форма `DrawworkshopeventestimationForm`, заповнення даних бронювання, дати оцінювання, типу оцінювання відбувається програмно.

Оцінювання відвідувача майстер-класу з малювання реалізовано функцією `GuestDrawworkshopeventestimationView`, яка має параметр `drawworkshopeventbooking_id`, який задає ідентифікатор бронювання місця на майстер-класі з малювання. В основі роботи знаходиться форма `DrawworkshopeventestimationForm`, заповнення даних бронювання, дати оцінювання, типу оцінювання відбувається програмно.

Оновлення даних відвідувача майстер-класів з малювання реалізовано функцією `WorkshopguestView`, яка головним чином працює за допомогою класу `WorkshopguestForm`, який опрацьовує форму з даними відвідувача майстер-класів з малювання.

Бронювання місця на майстер-класі з малювання реалізовано функцією `DrawworkshopeventbookingCreateView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання, якого стосується це бронювання.

Скасування бронювання місця на майстер-класі з малювання реалізовано функцією `DrawworkshopeventbookingCancelView` з параметром `drawworkshopeventbooking_id`, який задає ідентифікатор бронювання місця на майстер-класі з малювання.

Запрошення відвідувача на безкоштовне відвідування майстер-класу реалізовано функцією `DrawworkshopeventrequestCreateView` з параметром `drawworkshopevent_id`, який задає ідентифікатор проведення майстер-класу з малювання, а `workshopguest_login` задає логін відвідувача.

Скасування запрошення відвідувача на безкоштовне відвідування майстер-класу реалізовано функцією `DrawworkshopeventrequestCancelView` з

параметром `drawworkshopeventrequest_id`, який задає ідентифікатор запрошення відвідувача на безкоштовне відвідування майстер-класу.

Підтвердження запрошення відвідувача на безкоштовне відвідування майстер-класу реалізовано функцією `DrawworkshopeventrequestApproveView` з параметром `drawworkshopeventrequest_id`, який задає ідентифікатор запрошення відвідувача на безкоштовне відвідування майстер-класу.

Пошук серед локацій для проведення майстер-класів з малювання реалізовано функцією `WorkshoplocationSearchView`, де параметри самого пошуку передаються ключами GET-запиту, включаючи назву або частину назви локації в параметрі `workshoplocationtitle`, назву міста в параметрі `workshoplocationcity`.

Пошук серед запланованих для проведення майстер-класів з малювання реалізовано функцією `DrawworkshopeventSearchView`, де параметри самого пошуку передаються ключами GET-запиту, включаючи назву або частину назви локації в параметрі `workshoplocationtitle`, назву міста в параметрі `workshoplocationcity`, назву або частину назви майстер-класу з малювання `drawworkshoptitle`, ідентифікатор стилю малювання в параметрі `drawstyle_id`, мінімальну кількість відвідувачів в параметрі `drawworkshopguests_min`, максимальну кількість відвідувачів в параметрі `drawworkshopguests_max`, період проведення в параметрах `drawworkshopevent_startdate` та `drawworkshopevent_enddate`.

Пошук серед майстрів з малювання реалізовано функцією `DrawmasterSearchView`, де параметри самого пошуку передаються ключами GET-запиту, включаючи назву міста в параметрі `workshoplocationcity`, ім'я майстра або його частину `drawmaster_name`.

3.3 Висновки за розділом 3

За результатами розділу 3 виконано реалізацію програми підтримки проведення майстер-класів з малювання, забезпечуючи всі структурні одиниці

в програмі у відповідності зі структурою Django-застосунку, приводячи в цьому розділі пояснювальної записки моделі, представлення, шаблони вебсторінок, які було створено.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Програмне забезпечення підтримки проведення майстер-класів з малювання призначене для надання інформації про майстер-класи з малювання, реєстрації на проведення цих майстер-класів, вимірювання зацікавленості потенційних відвідувачів у проведенні конкретних подій майстер-класів з малювання перед безпосередньо відкриттям реєстрації на такі майстер-класи з малювання, для обміну інформацією про досвід участі в таких майстер-класах.

4.2 Умови виконання програми

Мінімально на сервері для роботи програми підтримки проведення майстер-класів з малювання мають бути виділені процесор мінімально з 2 ядрами, 4 Гб оперативної пам'яті, 100 Гб сховища на жорсткому диску, має бути забезпечено підтримку мови програмування Python та системи керування базами даних PostgreSQL.

4.3 Експлуатація програми підтримки проведення майстер-класів з малювання

Експлуатація програми підтримки проведення майстер-класів з малювання розпочинається зі звернення до сервера, на якому має бути розгорнуто попередньо застосунок.

Якщо під час першого звернення до застосунку користувач ще не був авторизований за цим веббраузером, то йому доступні з меню розділи пошуку, який включає пошук за локаціями, майстрами з малювання та майстер-класами з малювання, для чого далі на сторінці вибирається кнопками один з

цих режимів, та авторизації, звідки також можна перейти до реєстрації акаунту майстра або відвідувача, якщо його ще немає.

Якщо відкрити одну з публікацій про конкретне проведення майстер-класу (рис. 4.1), то з цієї сторінки доступна його назва, можливість забронювати за допомогою відповідної кнопки, опис. При цьому у випадку неавторизованого користувача кнопка бронювання призведе до необхідності виконання авторизації спершу, про що буде повідомлено відповідною сторінкою.

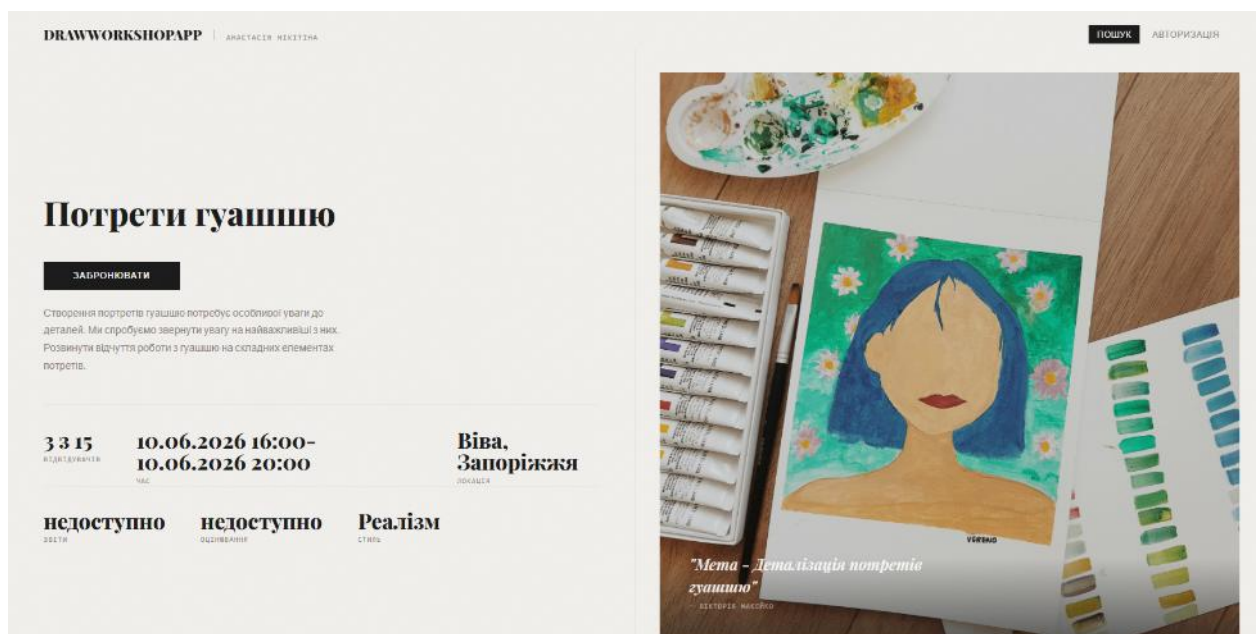


Рисунок 4.1 – Перегляд публікації про проведення майстер-класу з малювання

З цієї сторінки публікації про проведення майстер-класу з малювання доступна також основна фотографія, а весь перелік фотографій доступний нижче на сторінці з перемикачами між ними. Звідки також доступна можливість переходу до базового інформування про майстер-клас, де міститься тільки загальна інформація без уточнення локації та часу проведення, які під описом містяться на сторінці публікації.

Якщо цей майстер-клас вже проведений, то також містяться останні оцінювання, заголовки звітів та можливість відкриття сторінок зі всіма

оцінювання та звітами. У випадку звернення до сторінки самого майстра йому також доступний перехід до всіх бронювань за майстер-класом, звідки він може ними керувати, за необхідності скасовуючи бронювання певним відвідувачам. Кількість заброньованих місць та максимальна кількість місць на майстер-класі відображається під описом на цій сторінці.

З цієї сторінки також доступні під основною фотографією інформацію про мету проведення, майстра, що дозволяє перейти до сторінки самого майстра з малювання.

Якщо з програмою працює вже користувач як відвідувач після авторизації, то йому з меню доступний широкий набір розділів, включаючи:

- розділ для пошуку, аналогічний пошуку для неавторизованого користувача, включаючи засоби пошуку за локаціями, майстрами з малювання та майстер-класами з малювання;
- розділ з бронюваннями, звідки доступні всі виконані бронювання участі в майстер-класах відвідувача (рис. 4.2);
- розділ з виконаними голосуваннями за проведення майстер-класів;
- розділ з запрошеннями для безкоштовної участі в майстер-класах, адресованими авторизованому відвідувачу;
- розділ з оцінюваннями, звідки доступні оцінювання цього відвідувача за результатами відвідування майстер-класів з малювання;
- розділ зі звітами, звідки доступні створені звіти за відвіданими майстер-класами відвідувача, а також можливість їх оновлювати за необхідності;
- розділ з профілем з даними відвідувача, звідки він може зокрема оновити ці дані;
- можливість виходу з акаунта.

У випадку, якщо серед бронювань доступний майстер-клас, який вже проведений, то під фотографією доступні посилання для переходу до його оцінювання, створення звіту (рис. 4.3).

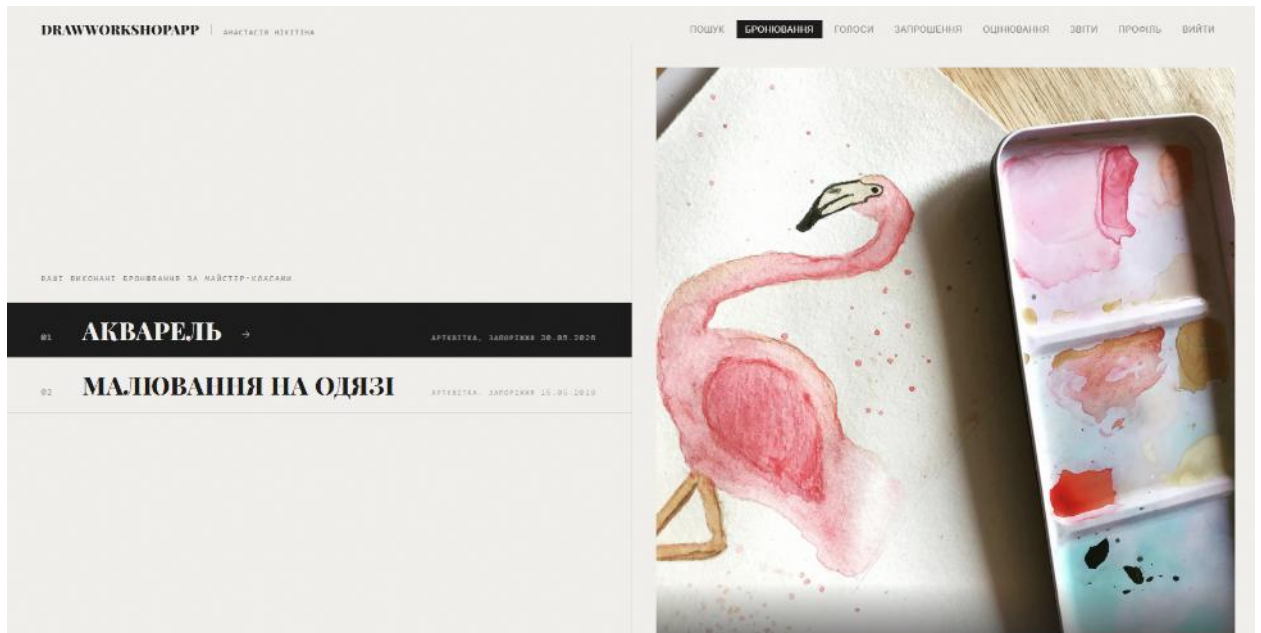


Рисунок 4.2 – Перегляд власних бронювань майстер-класів відвідувачем з деталізацією майстер-класу, який ще не відбувся

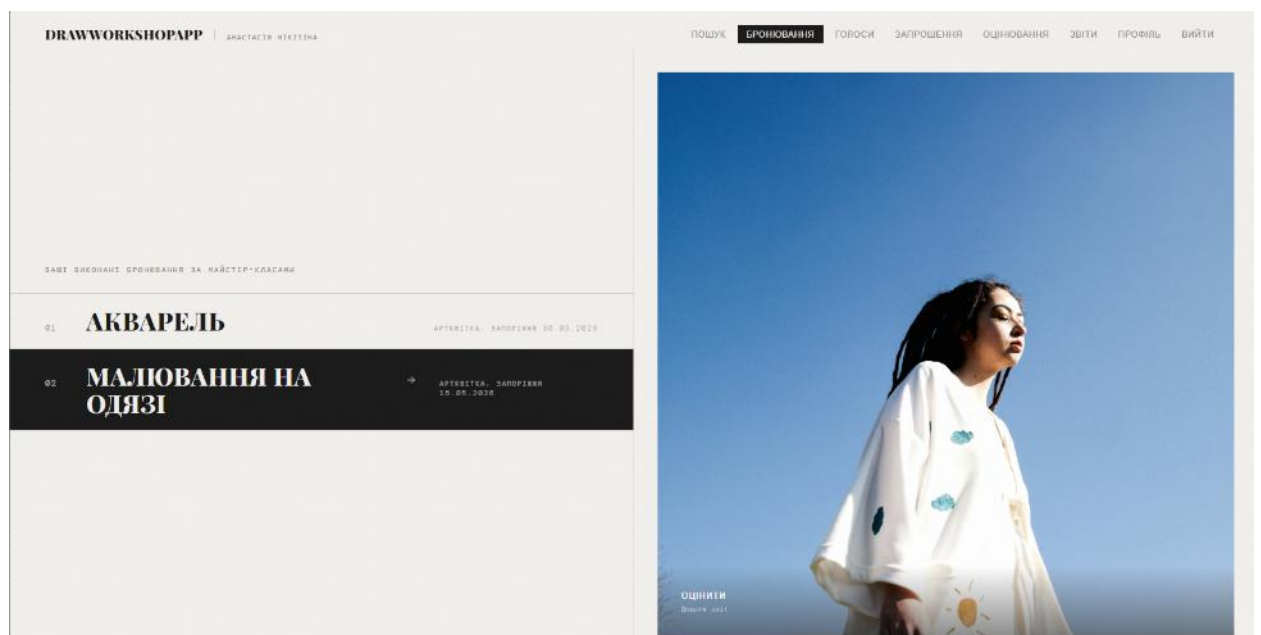


Рисунок 4.3 – Перегляд власних бронювань майстер-класів відвідувачем з деталізацією майстер-класу, який вже відбувся

Якщо після авторизації користувач має акаунт майстра з малювання, то йому також як і іншим користувачам доступний розділ пошуку, але також і розділ з майстер-класами, звідки доступний перелік майстер-класів авторизованого майстра з малювання, розділ зі створення звіту за

результатами проведеного майстер-класу з малювання, розділ з профілем майстра (рис. 4.4).

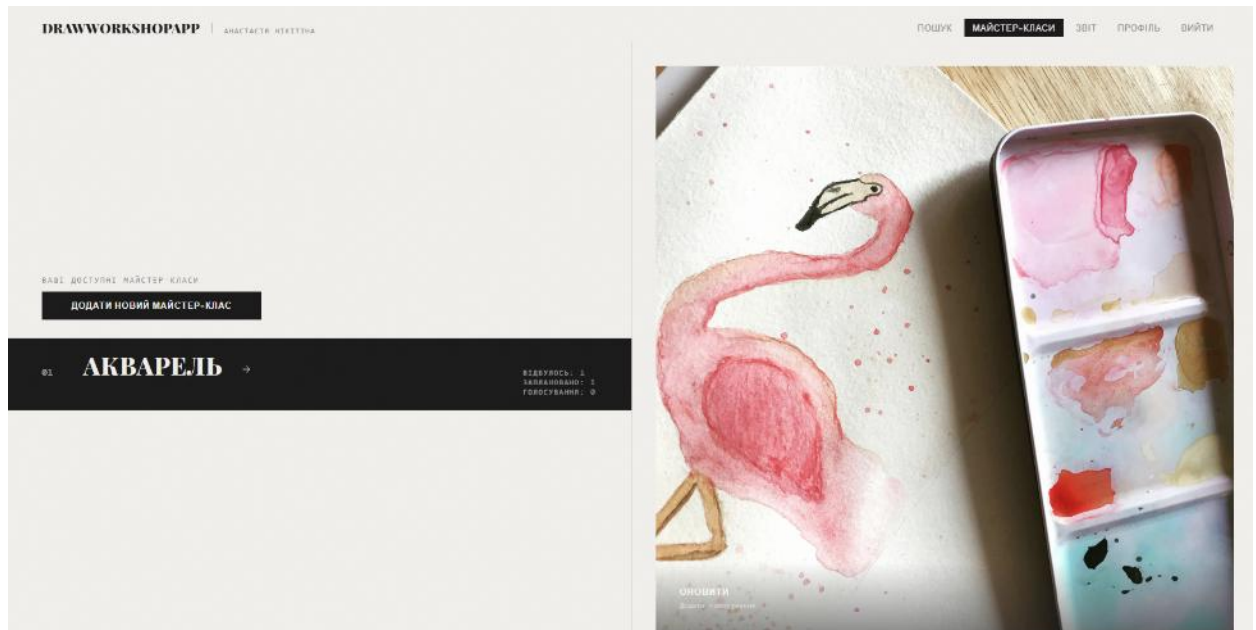


Рисунок 4.4 – Перегляд власних майстер-класів майстром з малювання

За своїми майстер-класами майстер може переглянути загальну інформацію, включаючи назву, кількість конкретних подій за майстер-класом, які були проведені, які заплановані до проведення, голосувань. Він може також перейти до відкриття голосування щодо проведення нової події за цим майстер-класом, оновлення даних майстер-класу. Також майстер може натиснути на саму фотографію і відкрити окрему сторінку з повними даними базового інформування про майстер-клас з малювання. З цієї сторінки вже доступна можливість створити публікацію про проведення події за майстер-класом з малювання.

4.4 Тестування програми підтримки проведення майстер-класів з малювання

Тестування програми підтримки проведення майстер-класів з малювання було реалізовано з перевіркою всіх функцій програми (табл. 4.1) з

успішним завершенням у підсумку.

Таблиця 4.1 – Результати тестування програми підтримки проведення майстер-класів з малювання

Функція	Результат
1	2
Створення базового інформування про майстер-клас з малювання	+
Оновлення базового інформування про майстер-клас з малювання	+
Перегляд власних базових інформувачів про майстер-класи з малювання	+
Перегляд сторінки майстра з базовими інформуваннями про майстер-класи з малювання	+
Оновлення даних майстра з малювання	+
Публікація проведення майстер-класу з малювання	+
Оновлення даних проведення майстер-класу з малювання	+
Вилучення проведення майстер-класу з малювання	+
Відкриття голосування за проведення майстер-класу з малювання	+
Голосування за проведення майстер-класу з малювання	+
Скасування голосування за проведення майстер-класу з малювання	+
Перегляд майстер-класів, за проведення яких проголосував користувач	+
Додавання локації для проведення майстер-класів з малювання	+
Оновлення даних локації для проведення майстер-класів з малювання	+

Продовження таблиці 4.1

1	2
Реєстрація акаунту відвідувача	+
Авторизація майстра з малювання	+
Авторизація відвідувача	+
Авторизація менеджера	+
Перегляд запланованих для проведення майстер-класів з малювання за майстром	+
Перегляд проведених майстер-класів з малювання за майстром	+
Перегляд запланованих для проведення майстер-класів з малювання за локацією	+
Перегляд проведених майстер-класів з малювання за локацією	+
Додавання звіту за проведеним майстер-класом з малювання	+
Оновлення звіту за проведеним майстер-класом з малювання	+
Оцінювання проведеного майстер-класу з малювання	+
Перегляд оцінювань майстер-класу з малювання	+
Оцінювання відвідувача майстер-класу з малювання	+
Перегляд оцінювань відвідувача майстер-класів з малювання	+
Перегляд сторінки відвідувача майстер-класів з малювання	+
Оновлення даних відвідувача майстер-класів з малювання	+
Пошук серед локацій для проведення майстер-класів з малювання	+
Пошук серед запланованих для проведення майстер-класів з малювання	+

Продовження таблиці 4.1

1	2
Пошук серед майстрів з малювання	+
Бронювання місця на майстер-класі з малювання	+
Скасування бронювання місця на майстер-класі з малювання	+
Перегляд власних бронювань місць на майстер-класах з малювання	+
Перегляд бронювань за майстер-класом з малювання	+
Запрошення відвідувача на безкоштовне відвідування майстер-класу	+
Скасування запрошення відвідувача на безкоштовне відвідування майстер-класу	+
Перегляд оформлених запрошень на безкоштовне відвідування власного майстер-класу	+
Перегляд адресованих для безкоштовного відвідування запрошень на майстер-класи з малювання	+
Перегляд оцінювань за майстер-класами майстра з малювання	+
Перегляд звітів за майстер-класом з малювання	+
Перегляд звітів за майстер-класами з малювання відвідувача	+
Перегляд базового інформування про майстер-клас з малювання	+
Перегляд публікації проведення майстер-класу з малювання	+
Перегляд голосування за проведення майстер-класу з малювання	+
Реєстрація акаунту майстра з малювання	+

Кінець таблиці 4.1

1	2
Перегляд даних локації для проведення майстер-класів з малювання	+
Перегляд звіту за проведеним майстер-класом з малювання	+
Підтвердження запрошення відвідувача на безкоштовне відвідування майстер-класу	+

4.5 Висновки за розділом 4

За результатами розділу 4 програма підтримки проведення майстер-класів з малювання була протестована, що засвідчило виконання всіх функціональних вимог до програми в повній мірі, проводячи перевірку за кожною функцією та проводячи опис основної логіки роботи користувача з програмою.

ВИСНОВКИ

Виконання роботи безпосередньо сприяло наданню можливостей підвищення залученості відвідувачів до проведення майстер-класів з малювання. Для цього розроблено програмне забезпечення підтримки проведення майстер-класів з малювання, яке надає інструменти для того, щоб інформувати відвідувачів про потенційно цікаві їм майстер-класи, щоб вивчати зацікавленість потенційних відвідувачів у проведенні таких майстер-класів.

Розробку програмного забезпечення підтримки проведення майстер-класів з малювання реалізовано з використанням мови Python та ФДБП Django, системи PostgreSQL для роботи з даними.

Програмне забезпечення підтримки проведення майстер-класів з малювання було порівняно з аналогами з виділенням особливостей, реалізовано проектування з моделюванням роботи відвідувачів майстер-класів з малювання, майстрів з малювання, менеджерів і неавторизованих користувачів та визначенням структури бази даних. Виконано реалізацію програми, забезпечуючи всі структурні одиниці в програмі у відповідності зі структурою Django-застосунку, включаючи зокрема моделі, представлення, шаблони вебсторінок.

Створена програма була протестована, що засвідчило виконання всіх функціональних вимог до програми в повній мірі, проводячи перевірку за кожною функцією та проводячи опис основної логіки роботи користувача з програмою.

Програма, яка створена в роботі, може використовуватися для надання інформації про майстер-класи з малювання, реєстрації на проведення цих майстер-класів, вимірювання зацікавленості потенційних відвідувачів у проведенні конкретних подій майстер-класів з малювання перед безпосередньо відкриттям реєстрації на такі майстер-класи з малювання, для обміну інформацією про досвід участі в таких майстер-класах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Powerful Online Scheduling Software. YouCanBookMe. URL: <https://youcanbook.me/features> (date of access: 11.05.2026).
2. Bookedin Online Appointment Scheduling App – Features. URL: <https://bookedin.com/online-scheduling-payment-system-features/> (date of access: 11.05.2026).
3. ArtWorkout. URL: <https://artworkout.app/> (date of access: 11.05.2026).
4. Writing your first Django app, part 1. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/intro/tutorial01/> (date of access: 12.05.2026).
5. Quickstart – Flask Documentation (3.1.x). URL: <https://flask.palletsprojects.com/en/stable/quickstart/> (date of access: 12.05.2026).
6. Verbina E. Which Is the Best Python Web Framework: Django, Flask, or FastAPI? *JetBrains Blog*. URL: <https://blog.jetbrains.com/pycharm/2025/02/django-flask-fastapi/> (date of access: 12.05.2026).
7. Flask vs Django – GeeksForGeeks. URL: <https://www.geeksforgeeks.org/python/differences-between-django-vs-flask/> (date of access: 12.05.2026).
8. Models. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/topics/db/models/> (date of access: 12.05.2026).
9. Writing views. Django documentation. Django. URL: <https://docs.djangoproject.com/en/6.0/topics/http/views/> (date of access: 12.05.2026).
10. Free Template 616 Split Index. URL: <https://templatemo.com/tm-616-split-index> (date of access: 15.05.2026).

ДОДАТОК А
Технічне завдання

Вступ

Майстер-класи з малювання вимагають обов'язкової реєстрації участі відвідувачів у них, щоб контролювати кількість учасників, але окрім даного процесу виникає ще одна проблема. Вона пов'язана з тим, що кількість учасників таких майстер-класів з малювання може бути невеликою, тоді проведення їх може бути неефективним. Для підвищення ефективності потрібно з одного боку надавати достатню кількість інформації про досвід участі конкретних людей у таких майстер-класах з малювання в минулому, але також і вивчати готовність людей до відвідування майстер-класів, які можуть бути проведені. Саме реалізації таких можливостей потрібно приділити увагу при розробці програмного забезпечення підтримки проведення майстер-класів з малювання.

A.1 Підстави для розробки

Робота виконана за темою «Програмне забезпечення підтримки проведення майстер-класів з малювання» у відповідності з даними Наказу № 139 від 7.04.2026 року в Національному університеті «Запорізька політехніка».

A.2 Призначення розробки

Програмне забезпечення підтримки проведення майстер-класів з малювання може використовуватися для надання інформації про майстер-класи з малювання, реєстрації на проведення цих майстер-класів, вимірювання зацікавленості потенційних відвідувачів у проведенні конкретних подій майстер-класів з малювання перед безпосередньо відкриттям реєстрації на такі майстер-класи з малювання, для обміну інформацією про досвід участі в таких майстер-класах.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма підтримки проведення майстер-класів з малювання має відповідати функціональним характеристикам, серед яких:

- створення базового інформування про майстер-клас з малювання;
- оновлення базового інформування про майстер-клас з малювання;
- перегляд власних базових інформувальних про майстер-класи з малювання;
- перегляд сторінки майстра з базовими інформуваннями про майстер-класи з малювання;
- оновлення даних майстра з малювання;
- публікація проведення майстер-класу з малювання;
- оновлення даних проведення майстер-класу з малювання;
- вилучення проведення майстер-класу з малювання;
- відкриття голосування за проведення майстер-класу з малювання;
- голосування за проведення майстер-класу з малювання;
- скасування голосування за проведення майстер-класу з малювання;
- перегляд майстер-класів, за проведення яких проголосував користувач;
- додавання локації для проведення майстер-класів з малювання;
- оновлення даних локації для проведення майстер-класів з малювання;
- реєстрація акаунту майстра з малювання, відвідувача;
- авторизація майстра з малювання, відвідувача, менеджера;
- перегляд запланованих для проведення майстер-класів з малювання за майстром;
- перегляд проведених майстер-класів з малювання за майстром;
- перегляд запланованих для проведення майстер-класів з малювання за локацією;
- перегляд проведених майстер-класів з малювання за локацією;

- додавання звіту за проведенням майстер-класом з малювання;
- оновлення звіту за проведенням майстер-класом з малювання;
- оцінювання проведеного майстер-класу з малювання;
- перегляд оцінювань майстер-класу з малювання;
- оцінювання відвідувача майстер-класу з малювання;
- перегляд оцінювань відвідувача майстер-класів з малювання;
- перегляд сторінки відвідувача майстер-класів з малювання;
- оновлення даних відвідувача майстер-класів з малювання;
- пошук серед локацій для проведення майстер-класів з малювання;
- пошук серед запланованих для проведення майстер-класів з малювання;
- пошук серед майстрів з малювання;
- бронювання місця на майстер-класі з малювання;
- скасування бронювання місця на майстер-класі з малювання;
- перегляд власних бронювань місць на майстер-класах з малювання;
- перегляд бронювань за майстер-класом з малювання;
- запрошення відвідувача на безкоштовне відвідування майстер-класу;
- скасування запрошення відвідувача на безкоштовне відвідування майстер-класу;
- перегляд оформлених запрошень на безкоштовне відвідування власного майстер-класу;
- перегляд адресованих для безкоштовного відвідування запрошень на майстер-класи з малювання;
- перегляд оцінювань за майстер-класами майстра з малювання;
- перегляд звітів за майстер-класом з малювання;
- перегляд звітів за майстер-класами з малювання відвідувача;
- перегляд базового інформування про майстер-клас з малювання;
- перегляд публікації проведення майстер-класу з малювання;
- перегляд голосування за проведення майстер-класу з малювання;
- перегляд даних локації для проведення майстер-класів з малювання;

– перегляд звіту за проведеним майстер-класом з малювання.

A.3.2 Умови експлуатації

Експлуатаційні вимоги до програми підтримки проведення майстер-класів з малювання включають виділення серверу для роботи з програмою шляхом використання сервера підприємства або замовлення хостингу. У обох випадках сервер повинен забезпечувати роботу з системою керування базами даних, мовою програмування, що застосовуватимуться для роботи з програмою після прийняття цих рішень в даній роботі.

A.3.3 Вимоги до складу та параметрів технічних засобів

Для забезпечення хостингу шляхом оренди сервера або використання власного серверу необхідно враховувати, що мінімальними вимогами, які на сервері виділені виключно під роботу з застосунком підтримки проведення майстер-класів з малювання передбачають наявність процесора мінімально з 2 ядрами, 4 Гб оперативної пам'яті, 100 Гб сховища на жорсткому диску.

A.4 Порядок контролю та приймання

Робота над дипломною кваліфікаційною роботою має виконуватися протягом 8-тижневого циклу, під час якого має бути виконано саму роботу, проведено перевірку роботи на нормоконтроль, проведено процедури рецензування та захисту роботи.

ДОДАТОК Б
Текст програми

```

from django.shortcuts import render
from django.db.models import Avg
from .models import Drawstyle,
    Drawworkshop,
    Drawworkshopphoto,
    Drawworkshopvideo,
    Drawworkshopevent,
    Drawworkshopeventrequest,
    Drawworkshopeventbooking,
    Drawworkshopeventphoto,
    Drawworkshopeventreport,
    Drawworkshopeventreportphoto,
    Drawworkshopeventreportvideo,
    Drawmaster,
    Drawmasterphoto,
    Drawworkshopvoting,
    Drawworkshopvotingmark,
    Workshoplocationcity,
    Workshoplocation,
    Workshoplocationphoto,
    Workshopguest,
    Drawworkshopeventestimation
from django.core.paginator import Paginator

def OwnDrawworkshopsView (request):
    drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount = request.user)
    if not drawmaster_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Ви повинні авторизуватись спершу,
а тоді запитувати майстер-класи"
            }
        )
    else:
        import datetime
        own_drawworkshops =
Drawworkshop.objects.select_related("drawmaster").select_related("draw
master__user_djangoaccount").select_related("draw_style").filter(drawm
aster__user_djangoaccount = request.user)

```

```

        current_workshop_time = datetime.datetime.now()
        own_drawworkshops = Paginator(own_drawworkshops, 10)
        own_drawworkshops_for_page = request.GET.get('page')
        own_drawworkshops =
own_drawworkshops.get_page(own_drawworkshops_for_page)
        for own_drawworkshop in own_drawworkshops:
            own_drawworkshop.photo =
Drawworkshopphoto.objects.filter(drawworkshop = own_drawworkshop,
photo_poster = True)
            own_drawworkshop.archive_event_count =
Drawworkshopevent.objects.filter(drawworkshop = own_drawworkshop,
drawworkshopevent_startdate__lt = current_workshop_time).count()
            own_drawworkshop.next_event_count =
Drawworkshopevent.objects.filter(drawworkshop = own_drawworkshop,
drawworkshopevent_startdate__gte = current_workshop_time).count()
            own_drawworkshop.votings_count =
Drawworkshopvoting.objects.filter(drawworkshop =
own_drawworkshop).count()
        return render(
            request,
            "drawworkshopapp/owndrawworkshops.html",
            {
                "user": drawmaster_user,
                "owndrawworkshops": own_drawworkshops
            }
        )

def MasterDrawworkshopsView (request, drawmaster_login = None):
    if drawmaster_login:
        drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount__username =
drawmaster_login)
    else:
        drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount = request.user)
    if not drawmaster_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такого майстра з малювання"
            }
        )

```

```

        else:
            drawmaster_user.photos =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = False)
            drawmaster_user.photo =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = True)
            import datetime
            master_drawworkshops =
Drawworkshop.objects.select_related("drawmaster").select_related("draw
master__user_djangoaccount").select_related("draw_style").filter(drawm
aster__user_djangoaccount = drawmaster_user)
            current_workshop_time = datetime.datetime.now()
            master_drawworkshops = Paginator(master_drawworkshops,
10)
            master_drawworkshops_for_page = request.GET.get('page')
            master_drawworkshops =
master_drawworkshops.get_page(master_drawworkshops_for_page)
            for master_drawworkshop in master_drawworkshops:
                master_drawworkshop.photo =
Drawworkshopphoto.objects.filter(drawworkshop = master_drawworkshop,
photo_poster = True)
                master_drawworkshop.archive_event_count =
Drawworkshopevent.objects.filter(drawworkshop = master_drawworkshop,
drawworkshopevent_startdate__lt = current_workshop_time).count()
                master_drawworkshop.next_event_count =
Drawworkshopevent.objects.filter(drawworkshop = master_drawworkshop,
drawworkshopevent_startdate__gte = current_workshop_time).count()
                master_drawworkshop.votings_count =
Drawworkshopvoting.objects.filter(drawworkshop =
master_drawworkshop).count()
                master_drawworkshop.votings =
Drawworkshopvoting.objects.filter(drawworkshop = master_drawworkshop,
drawworkshopvoting_stopdate__gt = current_workshop_time)
            return render(
                request,
                "drawworkshopapp/masterdrawworkshops.html",
                {
                    "user": drawmaster_user,
                    "masterdrawworkshops":
master_drawworkshops
                }
            )

```

```

def OwnDrawworkshopVotesView (request):
    workshopguest_user =
Workshopguest.objects.filter(user_djangoaccount = request.user)
    if not workshopguest_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Ви повинні авторизуватись спершу,
а тоді запитувати майстер-класи"
            }
        )
    else:
        import datetime
        own_drawworkshop_votes = Drawworkshopvotingmark.objects.
select_related("drawworkshopvoting").select_related("drawworkshopvotin
g__drawworkshop").select_related("drawworkshopvoting__drawworkshop__dr
awmaster").select_related("drawworkshopvoting__drawworkshop__drawmaste
r__user_djangoaccount").select_related("drawworkshopvoting__drawworksh
op__draw_style").filter(drawworkshopvoting__drawworkshop__drawmaster__
user_djangoaccount = request.user)
        current_workshop_time = datetime.datetime.now()
        own_drawworkshop_votes =
Paginator(own_drawworkshop_votes, 10)
        own_drawworkshop_votes_for_page =
request.GET.get('page')
        own_drawworkshop_votes =
own_drawworkshop_votes.get_page(own_drawworkshop_votes_for_page)
        for own_drawworkshop_vote in own_drawworkshop_votes:

own_drawworkshop_vote.drawworkshopvoting.drawworkshop.photo =
Drawworkshopphoto.objects.filter(drawworkshop =
own_drawworkshop_vote.drawworkshopvoting.drawworkshop, photo_poster =
True)

own_drawworkshop_vote.drawworkshopvoting.drawworkshop.archive_event_co
unt = Drawworkshopevent.objects.filter(drawworkshop =
own_drawworkshop_vote.drawworkshopvoting.drawworkshop,
drawworkshopevent_startdate__lt = current_workshop_time).count()

own_drawworkshop_vote.drawworkshopvoting.drawworkshop.next_event_count
= Drawworkshopevent.objects.filter(drawworkshop =

```

```

own_drawworkshop_vote.drawworkshopvoting.drawworkshop,
drawworkshopevent_startdate__gte = current_workshop_time).count()

own_drawworkshop_vote.drawworkshopvoting.drawworkshop.votings_count =
Drawworkshopvoting.objects.filter(drawworkshop =
own_drawworkshop_vote.drawworkshopvoting.drawworkshop).count()
        return render(
            request,
            "drawworkshopapp/owndrawworkshopvotes.html",
            {
                "user": workshopguest_user,
                "owndrawworkshopvotes":
own_drawworkshop_votes
            }
        )

def PlannedMasterDrawworkshopeventsView (request,
drawmaster_login):
    drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount__username =
drawmaster_login)
    if not drawmaster_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такого майстра з малювання"
            }
        )
    else:
        drawmaster_user.photo =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = True)
        import datetime
        current_workshop_time = datetime.datetime.now()
        master_drawworkshop_events =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").sele
ct_related("workshoplocation").select_related("workshoplocation__works
hoplocation_city").filter(drawworkshop__drawmaster__user_djangoaccount
= drawmaster_user, drawworkshopevent_startdate__gte =
current_workshop_time, drawworkshopevent_state__in = ["p", "f"])

```

```

        master_drawworkshop_events =
Paginator(master_drawworkshop_events, 10)
        master_drawworkshop_events_for_page =
request.GET.get('page')
        master_drawworkshop_events =
master_drawworkshops.get_page(master_drawworkshop_events_for_page)
        for master_drawworkshop_event in
master_drawworkshop_events:
            master_drawworkshop_event.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
master_drawworkshop_event, photo_poster = True)
            master_drawworkshop_event.estimation =
Drawworkshopeventestimation.objects.filter(drawworkshopeventbooking__d
rawworkshopevent = master_drawworkshop_event,
drawworkshopeventestimation_type =
True).aggregate(Avg('drawworkshopeventestimation_score'))
            return render(
                request,

"drawworkshopapp/planmasterdrawworkshopevents.html",
                {
                    "user": drawmaster_user,
                    "masterdrawworkshopevents":
master_drawworkshop_events
                }
            )

def FinishedMasterDrawworkshopeventsView (request,
drawmaster_login):
    drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount__username =
drawmaster_login)
    if not drawmaster_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такого майстра з малювання"
            }
        )
    else:

```

```

        drawmaster_user.photo =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = True)
        import datetime
        current_workshop_time = datetime.datetime.now()
        master_drawworkshop_events =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").sele
ct_related("workshoplocation").select_related("workshoplocation__works
hoplocation_city").filter(drawworkshop__drawmaster__user_djangoaccount
= drawmaster_user, drawworkshopevent_enddate__lt =
current_workshop_time, drawworkshopevent_state__in = ["d"])
        master_drawworkshop_events =
Paginator(master_drawworkshop_events, 10)
        master_drawworkshop_events_for_page =
request.GET.get('page')
        master_drawworkshop_events =
master_drawworkshops.get_page(master_drawworkshop_events_for_page)
        for master_drawworkshop_event in
master_drawworkshop_events:
            master_drawworkshop_event.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
master_drawworkshop_event, photo_poster = True)
            master_drawworkshop_event.estimation =
Drawworkshopeventestimation.objects.filter(drawworkshopeventbooking__d
rawworkshopevent = master_drawworkshop_event,
drawworkshopeventestimation_type =
True).aggregate(Avg('drawworkshopeventestimation_score'))
            return render(
                request,

"drawworkshopapp/finishmasterdrawworkshopevents.html",
                {
                    "user": drawmaster_user,
                    "masterdrawworkshopevents":
master_drawworkshop_events
                }
            )

def PlannedLocationDrawworkshopeventsView (request,
workshoplocation_id):

```

```

        workshoplocation =
Workshoplocation.objects.select_related("workshoplocation_city").filter(
    id = workshoplocation_id)
        if not workshoplocation:
            return render(
                request,
                "drawworkshopapp/error.html",
                {
                    "error": "Немає такої локації для проведення
майстер-класів"
                }
            )
        else:
            workshoplocation.photo =
Workshoplocationphoto.objects.filter(workshoplocation =
workshoplocation)[0]
            import datetime
            current_workshop_time = datetime.datetime.now()
            workshoplocation_drawworkshop_events =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").sele
ct_related("workshoplocation").select_related("workshoplocation__works
hoplocation_city").filter(workshoplocation = workshoplocation,
drawworkshopevent_startdate__gte = current_workshop_time,
drawworkshopevent_state__in = ["p", "f"])
            workshoplocation_drawworkshop_events =
Paginator(workshoplocation_drawworkshop_events, 10)
            workshoplocation_drawworkshop_events_for_page =
request.GET.get('page')
            workshoplocation_drawworkshop_events =
workshoplocation_drawworkshop_events.get_page(master_drawworkshop_even
ts_for_page)
            for workshoplocation_drawworkshop_event in
workshoplocation_drawworkshop_events:
                workshoplocation_drawworkshop_event.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
workshoplocation_drawworkshop_event, photo_poster = True)
                workshoplocation_drawworkshop_event.estimation =
Drawworkshopeventestimation.objects.filter(drawworkshopeventbooking__d
rawworkshopevent = workshoplocation_drawworkshop_event,
drawworkshopeventestimation_type =
True).aggregate(Avg('drawworkshopeventestimation_score'))

```

```

        return render(
            request,

            "drawworkshopapp/planlocationdrawworkshopevents.html",
            {
                "workshoplocation": workshoplocation,
                "masterdrawworkshopevents":
master_drawworkshop_events
            }
        )

    def FinishedLocationDrawworkshopeventsView (request,
workshoplocation_id):
        workshoplocation =
Workshoplocation.objects.select_related("workshoplocation_city").filter(id = workshoplocation_id)
        if not workshoplocation:
            return render(
                request,
                "drawworkshopapp/error.html",
                {
                    "error": "Немає такої локації для проведення
майстер-класів"
                }
            )
        else:
            workshoplocation.photo =
Workshoplocationphoto.objects.filter(workshoplocation =
workshoplocation)[0]
            import datetime
            current_workshop_time = datetime.datetime.now()
            workshoplocation_drawworkshop_events =
Drawworkshopevent.objects.select_related("drawworkshop").select_related("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster__user_djangoaccount").select_related("drawworkshop__draw_style").select_related("workshoplocation").select_related("workshoplocation__workshoplocation_city").filter(workshoplocation = workshoplocation, drawworkshopevent_enddate__lt = current_workshop_time, drawworkshopevent_state__in = ["d"])
            workshoplocation_drawworkshop_events =
Paginator(workshoplocation_drawworkshop_events, 10)
            workshoplocation_drawworkshop_events_for_page =
request.GET.get('page')

```

```

        workshoplocation_drawworkshop_events =
workshoplocation_drawworkshop_events.get_page(master_drawworkshop_events_for_page)
        for workshoplocation_drawworkshop_event in
workshoplocation_drawworkshop_events:
            workshoplocation_drawworkshop_event.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
workshoplocation_drawworkshop_event, photo_poster = True)
            workshoplocation_drawworkshop_event.estimation =
Drawworkshopeventestimation.objects.filter(drawworkshopeventbooking__d
rawworkshopevent = workshoplocation_drawworkshop_event,
drawworkshopeventestimation_type =
True).aggregate(Avg('drawworkshopeventestimation_score'))
            return render(
                request,

"drawworkshopapp/finishedlocationdrawworkshopevents.html",
                {
                    "workshoplocation": workshoplocation,
                    "masterdrawworkshopevents":
master_drawworkshop_events
                }
            )

```

```

def DrawworkshopeventEstimationsView (request,
drawworkshopevent_id):
    drawworkshopevent =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").
select_related("workshoplocation").
select_related("workshoplocation__workshoplocation_city").filter(id =
drawworkshopevent_id)
    if not drawworkshopevent:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такої події з проведення
майстер-класу з малювання"
            }
        )
    else:

```

```

        drawworkshopevent.photo =
Drawworkshopevent.objects.filter(drawworkshopevent =
drawworkshopevent, photo_poster = True)[0]
        drawworkshopevent_estimations =
Drawworkshopeventestimation.objects.select_related("drawworkshopeventb
ooking").select_related("drawworkshopeventbooking__drawworkshopevent")
.select_related("drawworkshopeventbooking__workshopguest").select_rela
ted("drawworkshopeventbooking__workshopguest__user_djangoaccount").fil
ter(drawworkshopeventbooking__drawworkshopevent = drawworkshopevent,
drawworkshopeventestimation_type = True).order_by("-
drawworkshopeventestimation_date")
        drawworkshopevent_estimations =
Paginator(drawworkshopevent_estimations, 10)
        drawworkshopevent_estimations_for_page =
request.GET.get('page')
        drawworkshopevent_estimations =
workshoplocation_drawworkshop_events.get_page(drawworkshopevent_estima
tions_for_page)
        return render(
            request,
            "drawworkshopapp/workshopestimations.html",
            {
                "event": drawworkshopevent,
                "estimations":
drawworkshopevent_estimations
            }
        )

```

```

def MasterDrawworkshopeventEstimationsView (request,
drawmaster_login):
    drawmaster_user =
Drawmaster.objects.filter(user_djangoaccount__username =
drawmaster_login)
    if not drawmaster_user:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такого майстра з малювання"
            }
        )
    else:

```

```

        drawmaster_user.photos =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = False)
        drawmaster_user.photo =
Drawmasterphoto.objects.filter(drawmaster = drawmaster_user,
photo_poster = True)
        import datetime
        master_drawworkshops =
Drawworkshop.objects.select_related("drawmaster").select_related("draw
master__user_djangoaccount").select_related("draw_style").filter(drawm
aster__user_djangoaccount = drawmaster_user)
        drawworkshopevent_estimations =
Drawworkshopeventestimation.objects.select_related("drawworkshopeventb
ooking").select_related("drawworkshopeventbooking__drawworkshopevent")
.select_related("drawworkshopeventbooking__drawworkshopevent__drawwork
shop").select_related("drawworkshopeventbooking__workshopguest").selec
t_related("drawworkshopeventbooking__workshopguest__user_djangoaccount
").filter(drawworkshopeventbooking__drawworkshopevent__drawworkshop__i
n = master_drawworkshops, drawworkshopeventestimation_type =
True).order_by("-drawworkshopeventestimation_date")
        drawworkshopevent_estimations =
Paginator(drawworkshopevent_estimations, 10)
        drawworkshopevent_estimations_for_page =
request.GET.get('page')
        drawworkshopevent_estimations =
workshoplocation_drawworkshop_events.get_page(drawworkshopevent_estima
tions_for_page)
        return render(
            request,

"drawworkshopapp/masterworkshopestimations.html",
            {
                "user": drawmaster_user,
                "estimations":
drawworkshopevent_estimations
            }
        )

def DrawworkshopeventReportsView (request,
drawworkshopevent_id):
    drawworkshopevent =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster

```

```

__user_djangoaccount").select_related("drawworkshop__draw_style").
select_related("workshoplocation").
select_related("workshoplocation__workshoplocation_city").filter(id =
drawworkshopevent_id)
    if not drawworkshopevent:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такої події з проведення
майстер-класу з малювання"
            }
        )
    else:
        drawworkshopevent.photo =
Drawworkshopevent.objects.filter(drawworkshopevent =
drawworkshopevent, photo_poster = True)[0]
        drawworkshopevent_reports =
Drawworkshopeventreport.objects.select_related("drawworkshopevent").se
lect_related("author_djangoaccount").filter(drawworkshopevent =
drawworkshopevent).order_by("-drawworkshopeventreport_date")
        drawworkshopevent_reports =
Paginator(drawworkshopevent_reports, 10)
        drawworkshopevent_reports_for_page =
request.GET.get('page')
        drawworkshopevent_reports =
workshoplocation_drawworkshop_events.get_page(drawworkshopevent_report
s_for_page)
        for drawworkshopevent_report in
drawworkshopevent_reports:
            drawworkshopevent_report.photo =
Drawworkshopeventreportphoto.objects.filter(drawworkshopeventreport =
drawworkshopevent_report)[0]
            return render(
                request,
                "drawworkshopapp/workshopeventreports.html",
                {
                    "event": drawworkshopevent,
                    "reports": drawworkshopevent_reports
                }
            )

def WorkshopguestReportsView (request, workshopguest_login =
None):

```

```

        if not workshopguest_login:
            workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount = request.user)
        else:
            workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount__username = workshopguest_login)
            if not workshop_guest:
                return render(
                    request,
                    "drawworkshopapp/error.html",
                    {
                        "error": "Немає такого відвідувача майстер-
класів з малювання"
                    }
                )
            else:
                drawworkshopevent_reports =
Drawworkshopeventreport.objects.select_related("drawworkshopevent").se
lect_related("author_djangoaccount").filter(author_djangoaccount =
workshop_guest.username).order_by("-drawworkshopeventreport_date")
                drawworkshopevent_reports =
Paginator(drawworkshopevent_reports, 10)
                drawworkshopevent_reports_for_page =
request.GET.get('page')
                drawworkshopevent_reports =
workshoplocation_drawworkshop_events.get_page(drawworkshopevent_report
s_for_page)
                for drawworkshopevent_report in
drawworkshopevent_reports:
                    drawworkshopevent_report.photo =
Drawworkshopeventreportphoto.objects.filter(drawworkshopeventreport =
drawworkshopevent_report)[0]
                return render(
                    request,
                    "drawworkshopapp/workshopguestreports.html",
                    {
                        "event": drawworkshopevent,
                        "reports": drawworkshopevent_reports
                    }
                )

```

```

def WorkshopguestEstimationsView (request, workshopguest_login =
None):
    if not workshopguest_login:
        workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount = request.user)
    else:
        workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount__username = workshopguest_login)
        if not workshop_guest:
            return render(
                request,
                "drawworkshopapp/error.html",
                {
                    "error": "Немає такого відвідувача майстер-
класів з малювання"
                }
            )
        else:
            drawworkshopevent_estimations =
Drawworkshopeventestimation.objects.select_related("drawworkshopeventb
ooking").select_related("drawworkshopeventbooking__drawworkshopevent")
.select_related("drawworkshopeventbooking__drawworkshopevent__drawwork
shop").select_related("drawworkshopeventbooking__workshopguest").selec
t_related("drawworkshopeventbooking__workshopguest__user_djangoaccount
").filter(drawworkshopeventbooking__workshopguest = workshop_guest,
drawworkshopeventestimation_type = False).order_by("-
drawworkshopeventestimation_date")
            drawworkshopevent_estimations =
Paginator(drawworkshopevent_estimations, 10)
            drawworkshopevent_estimations_for_page =
request.GET.get('page')
            drawworkshopevent_estimations =
workshoplocation_drawworkshop_events.get_page(drawworkshopevent_estima
tions_for_page)
            return render(
                request,
                "drawworkshopapp/guestworkshopestimations.html",
                {
                    "user": workshop_guest,
                    "estimations":
drawworkshopevent_estimations

```

```

        }
    )

    def WorkshopguestView (request, workshopguest_login = None):
        if not workshopguest_login:
            workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount = request.user)
        else:
            workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount__username = workshopguest_login)
            if not workshop_guest:
                return render(
                    request,
                    "drawworkshopapp/error.html",
                    {
                        "error": "Немає такого відвідувача майстер-
класів з малювання"
                    }
                )
            else:
                drawworkshopevent_estimations =
Drawworkshopeventestimation.objects.select_related("drawworkshopeventb
ooking").select_related("drawworkshopeventbooking__drawworkshopevent")
.select_related("drawworkshopeventbooking__drawworkshopevent__drawwork
shop").select_related("drawworkshopeventbooking__workshopguest").selec
t_related("drawworkshopeventbooking__workshopguest__user_djangoaccount
").filter(drawworkshopeventbooking__workshopguest = workshop_guest,
drawworkshopeventestimation_type = False).order_by("-
drawworkshopeventestimation_date")[:10]
                drawworkshopevent_reports =
Drawworkshopeventreport.objects.select_related("drawworkshopevent").se
lect_related("author_djangoaccount").filter(author_djangoaccount =
workshop_guest.username).order_by("-
drawworkshopeventreport_date")[:10]
                for drawworkshopevent_report in
drawworkshopevent_reports:
                    drawworkshopevent_report.photo =
Drawworkshopeventreportphoto.objects.filter(drawworkshopeventreport =
drawworkshopevent_report)[0]
                return render(
                    request,

```

```

        "drawworkshopapp/guest.html",
        {
            "user": workshop_guest,
            "estimations":
drawworkshopevent_estimations,
            "reports": drawworkshopevent_reports
        }
    )

    def OwnBookedDrawworkshopeventsView (request):
        workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount = request.user)
        if not workshop_guest:
            return render(
                request,
                "drawworkshopapp/error.html",
                {
                    "error": "Треба авторизуватись спершу"
                }
            )
        else:
            workshoplocation_drawworkshop_event_bookings =
Drawworkshopeventbooking.objects.select_related("drawworkshopevent").s
elect_related("drawworkshopevent__drawworkshop").select_related("draww
orkshopevent__drawworkshop__drawmaster").select_related("drawworkshope
vent__drawworkshop__drawmaster__user_djangoaccount").select_related("d
rawworkshopevent__drawworkshop__draw_style").select_related("drawworks
hopevent__workshoplocation").select_related("drawworkshopevent__worksh
oplocation__workshoplocation_city").filter(workshopguest =
workshop_guest).order_by("-
drawworkshopevent__drawworkshopevent_startdate")
            workshoplocation_drawworkshop_event_bookings =
Paginator(workshoplocation_drawworkshop_event_bookings, 10)
            workshoplocation_drawworkshop_event_bookings_for_page =
request.GET.get('page')
            workshoplocation_drawworkshop_event_bookings =
workshoplocation_drawworkshop_events.get_page(workshoplocation_drawwor
kshop_event_bookings_for_page)
            for workshoplocation_drawworkshop_event_booking in
workshoplocation_drawworkshop_event_bookings:
                workshoplocation_drawworkshop_event.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =

```

```
workshoplocation_drawworkshop_event.drawworkshopevent, photo_poster =
True)
```

```
        return render(
            request,
            "drawworkshopapp/bookedworkshopevents.html",
            {
                "drawworkshopevents":
workshoplocation_drawworkshop_event_bookings
            }
        )
```

```
    def DrawworkshopeventBookingsView (request,
drawworkshopevent_id):
        drawworkshopevent =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").
select_related("workshoplocation").
select_related("workshoplocation__workshoplocation_city").filter(id =
drawworkshopevent_id)
        if not drawworkshopevent:
            return render(
                request,
                "drawworkshopapp/error.html",
                {
                    "error": "Немає такої події з проведення
майстер-класу з малювання"
                }
            )
        else:
            drawworkshopeventbookings =
Drawworkshopeventbooking.objects.select_related("drawworkshopevent").s
elect_related("drawworkshopevent__drawworkshop").select_related("draww
orkshopevent__drawworkshop__drawmaster").select_related("drawworkshope
vent__drawworkshop__drawmaster__user_djangoaccount").select_related("d
rawworkshopevent__drawworkshop__draw_style").select_related("drawworks
hopevent__workshoplocation").select_related("drawworkshopevent__worksh
oplocation__workshoplocation_city").filter(drawworkshopevent =
drawworkshopevent)
            drawworkshopeventbookings =
Paginator(drawworkshopeventbookings, 10)
            drawworkshopeventbookings_for_page =
request.GET.get('page')
```

```

        drawworkshopeventbookings =
drawworkshopeventbookings.get_page(drawworkshopeventbookings_for_page)
        for drawworkshopeventbooking in
drawworkshopeventbookings:
            drawworkshopeventbooking.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
drawworkshopeventbooking.drawworkshopevent, photo_poster = True)
        return render(
            request,
            "drawworkshopapp/workshopeventbookings.html",
            {
                "drawworkshopevents":
drawworkshopeventbookings
            }
        )

```

```

def DrawworkshopEventView (request, drawworkshopevent_id):
    drawworkshopevent =
Drawworkshopevent.objects.select_related("drawworkshop").select_relate
d("drawworkshop__drawmaster").select_related("drawworkshop__drawmaster
__user_djangoaccount").select_related("drawworkshop__draw_style").
select_related("workshoplocation").
select_related("workshoplocation__workshoplocation_city").filter(id =
drawworkshopevent_id)
    if not drawworkshopevent:
        return render(
            request,
            "drawworkshopapp/error.html",
            {
                "error": "Немає такої події з проведення
майстер-класу з малювання"
            }
        )
    else:
        drawworkshopevent.photo =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
drawworkshopevent, photo_poster = True)
        drawworkshopevent.photos =
Drawworkshopeventphoto.objects.filter(drawworkshopevent =
drawworkshopevent, photo_poster = False)
        drawworkshopevent.reports_count =
Drawworkshopeventreport.objects.filter(drawworkshopevent =
drawworkshopevent).order_by("-drawworkshopeventreport_date").count()

```

```

        drawworkshopevent.reports =
Drawworkshopeventreport.objects.filter(drawworkshopevent =
drawworkshopevent).order_by("-drawworkshopeventreport_date")[:10]
        for drawworkshopevent_report in
drawworkshopevent.reports:
            drawworkshopevent_report.photo =
Drawworkshopeventreportphoto.objects.filter(drawworkshopeventreport =
drawworkshopevent_report)[0]
            workshop_guest =
Workshopguest.objects.select_related("user_djangoaccount").filter(user
_djangoaccount = request.user)
            if workshop_guest:
                booking =
Drawworkshopeventbooking.objects.filter(drawworkshopevent =
drawworkshopevent, workshopguest = workshop_guest,
drawworkshopeventbooking_state = "S")
                if booking:
                    drawworkshopevent.booked = True
                else:
                    drawworkshopevent.booked = False
            else:
                drawworkshopevent.booked = False
    return render(
        request,
        "drawworkshopapp/workshopevent.html",
        {
            "drawworkshopevent": drawworkshopevent
        }
    )

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
кафедра програмних засобів

Тема дипломної кваліфікаційної роботи бакалавра
Програмне забезпечення підтримки
проведення майстер-класів з малювання
Software for Drawing Workshop Management

Виконавець:

Анастасія НІКІТИНА

Керівник: к.т.н., доцент

Валерій ЛЬОВКІН

2026

Рисунок В.1 – Початковий слайд

Об'єкт, предмет, мета роботи

Об'єкт роботи – процес розробки програмного забезпечення підтримки проведення майстер-класів з малювання.

Предмет роботи – програмне забезпечення підтримки проведення майстер-класів з малювання.

Мета роботи – підвищення залученості відвідувачів до проведення майстер-класів з малювання за рахунок інформування та вивчення зацікавленості в таких майстер-класах на основі розробки програмного забезпечення підтримки проведення майстер-класів з малювання.

2

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

Завдання роботи

- аналіз проблеми розробки програми підтримки проведення майстер-класів з малювання;
- проєктування програми підтримки проведення майстер-класів з малювання;
- розробка програми підтримки проведення майстер-класів з малювання.

3

Рисунок В.3 – Завдання роботи

Порівняння програмних засобів підтримки проведення майстер-класів з малювання

Критерій	YouCanBookMe	Bookedin	DrawWorkshopApp
Бронювання місць на майстер-класах з малювання	Так	Так	Так
Надання даних профілів відвідувачів	Ні	Так	Так
Створення звітів про майстер-класи з малювання	Ні	Ні	Так
Голосування за відкриття подій майстер-класів з малювання	Ні	Ні	Так
Параметризований пошук	Ні	Ні	Так

4

Рисунок В.4 – Порівняння програмних засобів підтримки проведення майстер-класів з малювання

Порівняння фреймворків для веброзробки програми підтримки проведення майстер-класів з малювання

Критерій	Django	Flask
Основні можливості	Фреймворк для веброзробки повного стеку	Мікрофреймворк
Додавання можливостей	Велика кількість при великій кількості вбудованих засобів	Достатня кількість при малій кількості вбудованих засобів
Наявність засобів захисту	Велика кількість вбудованих	Мала кількість вбудованих, далі потрібно розширювати
Робота з даними	На основі вбудованого об'єктно-реляційного відображення	Потребує підключення або розробки з нуля

Рисунок В.5 – Порівняння фреймворків для веброзробки програми підтримки проведення майстер-класів з малювання

Діаграми прецедентів відвідувача програми підтримки проведення майстер-класів з малювання та неавторизованого користувача

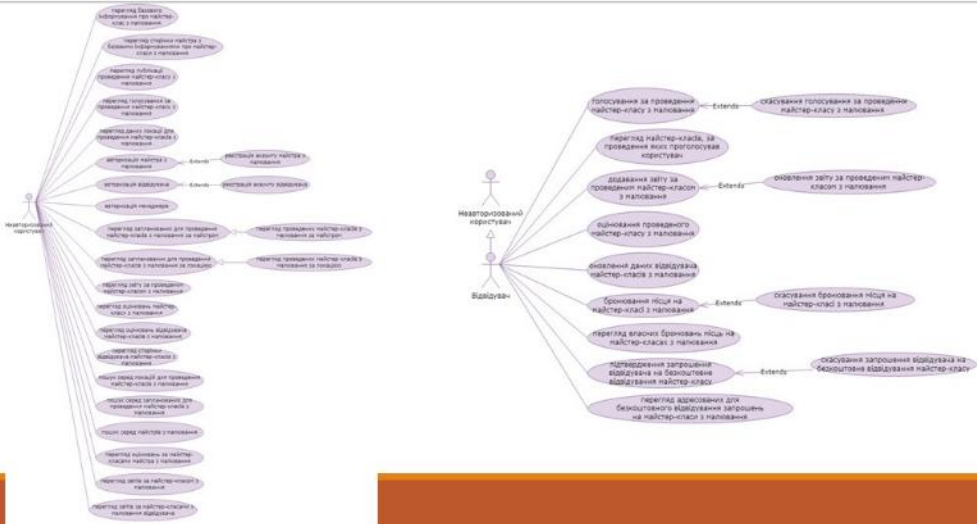
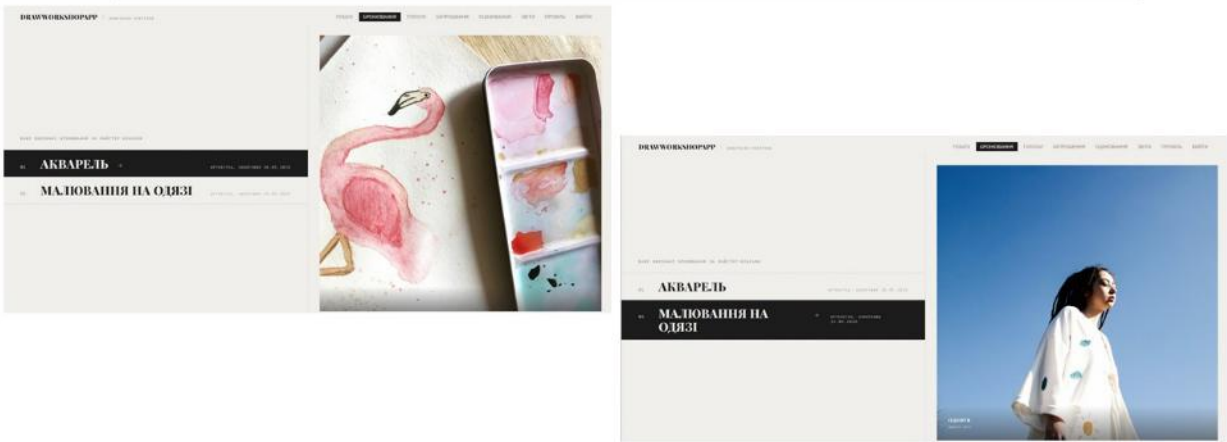


Рисунок В.6 – Діаграми прецедентів відвідувача програми підтримки проведення майстер-класів з малювання та неавторизованого користувача

Інтерфейси програми підтримки проведення майстер-класів з малювання для відвідувача



11

Рисунок В.11 – Інтерфейси програми підтримки проведення майстер-класів з малювання для відвідувача

Результати тестування програми підтримки проведення майстер-класів з малювання

		Функція	Результат
		Регістрація акаунту відвідувача	+
		Авторизація майстра з малювання	+
		Авторизація відвідувача	+
		Авторизація менеджера	+
Створення базового інформування про майстер-клас з малювання	+	Перегляд запланованих для проведення майстер-класів з малювання за майстром	+
Оновлення базового інформування про майстер-клас з малювання	+	Перегляд проведених майстер-класів з малювання за майстром	+
Перегляд власних базових інформувачів про майстер-класи з малювання	+	Перегляд запланованих для проведення майстер-класів з малювання за локацією	+
Перегляд сторінки майстра з базовими інформуваннями про майстер-класи з малювання	+	Перегляд проведених майстер-класів з малювання за локацією	+
Оновлення даних майстра з малювання	+	Додавання звіту за проведеним майстер-класом з малювання	+
Публікація проведення майстер-класу з малювання	+	Оновлення звіту за проведеним майстер-класом з малювання	+
Оновлення даних проведення майстер-класу з малювання	+	Оцінювання проведеного майстер-класу з малювання	+
Вилучення проведення майстер-класу з малювання	+	Перегляд оцінювань майстер-класу з малювання	+
Відкриття голосування за проведення майстер-класу з малювання	+	Оцінювання відвідувача майстер-класу з малювання	+
Голосування за проведення майстер-класу з малювання	+	Перегляд оцінювань відвідувача майстер-класів з малювання	+
Скасування голосування за проведення майстер-класу з малювання	+	Перегляд сторінки відвідувача майстер-класів з малювання	+
Перегляд майстер-класів, за проведення яких проголосував користувач	+	Оновлення даних відвідувача майстер-класів з малювання	+
Додавання локації для проведення майстер-класів з малювання	+	Пошук серед локацій для проведення майстер-класів з малювання	+
Оновлення даних локації для проведення майстер-класів з малювання	+	Пошук серед запланованих для проведення майстер-класів з малювання	+

12

Рисунок В.12 – Результати тестування програми підтримки проведення майстер-класів з малювання

Результати тестування програми підтримки проведення майстер-класів з малювання

Функція	Результат	Функція	Результат
Пошук серед майстрів з малювання	+	Перегляд даних локації для проведення майстер-класів з малювання	+
Бронювання місця на майстер-класі з малювання	+	Перегляд звіту за проведенням майстер-класом з малювання	+
Скасування бронювання місця на майстер-класі з малювання	+	Підтвердження запрошення відвідувача на безкоштовне відвідування майстер-класу	+
Перегляд власних бронювань місць на майстер-класах з малювання	+		
Перегляд бронювань за майстер-класом з малювання	+		
Запрошення відвідувача на безкоштовне відвідування майстер-класу	+		
Скасування запрошення відвідувача на безкоштовне відвідування майстер-класу	+		
Перегляд оформлених запрошень на безкоштовне відвідування власного майстер-класу	+		
Перегляд адресованих для безкоштовного відвідування запрошень на майстер-класи з малювання	+		
Перегляд оцінювань за майстер-класами майстра з малювання	+		
Перегляд звітів за майстер-класом з малювання	+		

13

Рисунок В.13 – Результати тестування програми підтримки проведення майстер-класів з малювання

Висновки

Виконання роботи безпосередньо сприяло наданню можливостей підвищення залученості відвідувачів до проведення майстер-класів з малювання. Для цього розроблено програмне забезпечення підтримки проведення майстер-класів з малювання, яке надає інструменти для того, щоб інформувати відвідувачів про потенційно цікаві їм майстер-класи, щоб вивчати зацікавленість потенційних відвідувачів у проведенні таких майстер-класів.

Програма, яка створена в роботі, може використовуватися для надання інформації про майстер-класи з малювання, реєстрації на проведення цих майстер-класів, вимірювання зацікавленості потенційних відвідувачів у проведенні конкретних подій майстер-класів з малювання перед безпосередньо відкриттям реєстрації на такі майстер-класи з малювання, для обміну інформацією про досвід участі в таких майстер-класах.

14

Рисунок В.14 – Висновки