

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)

бакалавра

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РЕЦЕНЗУВАННЯ

(назва теми)

ВУЛИЧНИХ АКТИВНОСТЕЙ

SOFTWARE FOR REVIEWING OPEN STREET ACTIVITIES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-110

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ДУБОВИК М.А.

(ПРІЗВИЩЕ та ініціали)

Керівник СУББОТІН С.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

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
« _____ » _____ 2024 року

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

ДУБОВИКА Микити Андрійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення рецензування вуличних активностей. Software for Reviewing Open Street Activities

керівник проєкту (роботи) д.т.н., професор СУББОТІН Сергій Олександрович ,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 24 » квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 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 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	СУББОТІН С.О., зав. кафедри		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 15 » квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

_____ Микита ДУБОВИК
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Сергій СУББОТІН
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 89 с., 26 рис., 2 табл., 3 дод., 7 джерел.

ВУЛИЧНА АКТИВНІСТЬ, РЕЦЕНЗУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, БАЗА ДАНИХ, ОБ'ЄКТНО-РЕЛЯЦІЙНЕ ВІДОБРАЖЕННЯ.

Об'єкт роботи – процес виступу виконавців у формі вуличних активностей. Предмет роботи – програмне забезпечення рецензування вуличних активностей. Мета роботи – розробити програмне забезпечення рецензування вуличних активностей для підвищення інформованості користувачів.

Матеріали, методи та технічні засоби: мова Python, мультипарадигмальне програмування, фреймворк Django, система MySQL.

Результати. Розроблена програма надає можливість отримувати інформацію про вуличні активності, які проводяться у місті, час таких виступів, про фестивалі, в межах яких такі вуличні активності відбуваються, рецензії від інших користувачів, які стали свідками таких активностей, та створювати власні рецензії.

Висновки. Створення програми як вебдодатку сприяє широті доступу користувачів з різних пристроїв, з різних локацій, що дозволяє охоплювати ширшу аудиторію, яка може стати відвідувачами фестивалів вуличних активностей, глядачами виступів вуличних артистів, долучатися до розвитку вуличних активностей як бізнесу, докладатися до розвитку вулиць міста, на яких такі активності відбуваються.

Галузь використання. Підтримка інформування щодо вуличних активностей, що включають виступи музикантів, театралізовані виступи, виступи інших вуличних артистів та інші типи вуличних активностей.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 89 pages, 26 figures, 2 tables, 3 appendixes, 7 sources.

STREET ACTIVITY, REVIEWING, SOFTWARE, DATABASE, OBJECT-RELATIONAL MAPPING.

The object of the work is the performance process by street artists in the form of street activities. The subject of the work is software for reviewing street activities. The goal of the work is to develop software for reviewing street activities to increase user awareness.

Materials, methods and technical means: Python language, multiparadigm programming, Django framework, MySQL system.

The results. The developed program provides an opportunity to receive information about street activities which take place in the city, the time of such performances, about festivals within which such street activities take place, reviews from other users who witnessed such activities, and to create own reviews.

Conclusions. The creation of the program as a web application contributes to the breadth of access of users from different devices, from different locations, which allows to reach a wider audience, who can become visitors to street activity festivals, spectators of performances by street artists, join the development of street activities as a business, engage in the development of city streets, on which such activities take place.

Field of use. Support awareness of street activities including performances by musicians, theatrical performances, performances by other street artists and other types of street activities.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Огляд програмних засобів рецензування вуличних активностей.....	9
1.2 Особливості розробки.....	11
1.3 Висновки за розділом 1	12
2 Проєктування програми.....	14
2.1 Вибір засобів проєктування і розробки	14
2.2 Визначення функціональних вимог до програмного забезпечення	15
2.3 Проєктування структури бази даних.....	22
2.4 Висновки за розділом 2	32
3 Розробка програми	33
3.1 Порядок реалізації функцій програми	33
3.2 Реалізація функцій програми	34
3.3 Висновки за розділом 3	48
4 Експлуатація та тестування програми	49
4.1 Призначення програми	49
4.2 Умови виконання програми	49
4.3 Виконання і тестування програми.....	49
4.4 Висновки за розділом 4	54
Висновки	55
Перелік джерел посилення	56
Додаток А Технічне завдання	57
Додаток Б Текст програми	63
Додаток В Слайди презентації.....	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД – база даних;

ОРВ – об’єктно-реляційне відображення.

ВСТУП

Вуличні активності є однією з форм розваг, яка тісно інтегрована з життям вулиці, допомагаючи в підсумку розвивати вулицю, привертати до неї увагу, а одночасно за рахунок розвитку, створення певної історії вулиці, привертати додаткову увагу до самих вуличних активностей, до виконавців цих активностей.

Прикладами вуличних активностей є виступи музикантів, при чому і індивідуальні, і в групах, театралізовані виступи, інші типи виступів вуличних активностей, але окрім того це може бути і проведення фестивалів, у яких беруть участь продавців різних товарів, зокрема власноруч виготовлених, суміщення такого продажу з певними виступами.

Розповсюдження інформації про такі події, які можуть відбуватися регулярно або одноразово, є важливим, адже це не концерти, на які продаються квитки, створюються афіші. Зазвичай інформація про такі вуличні активності розповсюджується від людей до людей. Тож створення вебдодатку має сприяти розвитку цієї сфери.

Важливим питанням при створенні такого додатку є рецензування вуличних активностей, адже лише наявності інформації про вуличні активності може бути недостатньо. Так велика частина вуличних активностей може бути неякісною, що буде сприяти розчаруванню відвідувачів та зменшенню уваги до інших виступів, які тільки втратять в увазі від інформування без рецензування. Тому наявність механізму рецензування є обов'язковою і має надаватися як стосовно виступів груп (або індивідуальних) виконавців вуличних активностей, так і стосовно цілих фестивалів вуличних активностей.

Таким чином, у роботі було поставлено мету – розробити програмне забезпечення рецензування вуличних активностей для підвищення інформованості користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд програмних засобів рецензування вуличних активностей

Необхідно одразу зазначити, що як такого програмного рішення, яке на даний момент і охоплювало би інформацію про вуличні активності в місті або містах, і надавало можливості рецензування таких вуличних активностей цілісно немає.

Зазвичай інформація про вуличні активності проводиться у різних соціальних мережах, відповідно є дуже фрагментованою, оскільки передбачає, що потрібно слідкувати за відповідними сторінками, а сторінок може бути багато і різних. Тобто інформація є фрагментованою, знайти її стає не просто. До того ж зазвичай це інформація про певні фестивалі або інші більш масштабні події, а виступи окремих музикантів залишаються без уваги. Тобто знайти інформацію про виступ окремого музиканта на площі або біля будинку на вулиці можна швидше у вигляді відео, де фрагмент цього виступу записаний. Однак такий варіант знову ж таки ускладнює пошук інформації. До того ж нерідко незрозуміло навіть, де цей виступ відбувся, не кажучи вже про те, що не можна ідентифікувати, тобто отримати інформацію про самого виконавця вуличних активностей. Усе це ускладнює роботу.

Під час пошуку програм, які надають інформацію про вуличні активності, серед централізованих додатків (не соціальних мереж) вдалось знайти Open Streets Day [1] (рис. 1.1), який характеризується як глобальна світова ініціатива щодо проведення днів вулиць. Це цілісні події, які можна розглядати як фестивалі, відповідно знайти інформацію про окремих вуличних виконавців, які традиційно виступають фрагментовано містом, неможливо. Проте отримати інформацію про подію загалом можливо, хоча і не можна поставити запитання щодо таких подій. Надається інструмент для реєстрації власної такої події. Тобто загалом з ініціатива з залучення громадських активістів для проведення таких подій у власному місті.

Partners | Contact us | Log in | Sign up


OPEN STREETS DAY
22 September

HOW TO | EVENTS | RESOURCES | NEWS

REGISTER EVENT

CROATIA 24.09



OPEN STREETS DAY
OPEN STREETS, OPEN MOVEMENT, OPEN MINDS

CROATIA 24.09.

Institute of Public Health County of Međimurje is organizing an Open Streets Day - event as a part of the Security and Prevention Fair in the City of Čakovec. A wide variety of sport-recreational activities will be organized on the closures road and green spaces.

We have arranged a 50 km long bike tour on the **24th of September** for public health workers, police officers and Red Cross employees. The Cycling tour will connect all three cities in Međimurje County – **Prelog, Mursko Središće and Čakovec**. Finish point of the race will be on the Security and Prevention Fair which will take place on 24.09.2019.

On the same day - **24th of September**, children from kindergartens and schools as well as people from different vulnerable groups will have the opportunity to participate in children's cycling races and outdoor sport activities in all the three cities.

Contact person:	Renata Kutnjak Kiš, MD
------------------------	------------------------

The concept of Open Streets Day is built into the name of the project: on 22 September 2019 public streets and urban areas of European cities will be transformed into venues for recreational physical activity. Local community members are invited to take the initiative and organise events that are relevant and interesting to residents of their area. As an organiser, you are free to choose an event theme ranging from simple ideas, like rope-skipping contests, to something more elaborate like a mass participation cycling ride.

With the Car-Free Day also scheduled on 22 September as a part of EUROPEANMOBILITYWEEK, Open Streets Day events may take advantage of the already existing road closures and make the experience even more memorable by letting residents discover their neighbourhood from a different perspective.

Serving as a bridge between European Mobility Week and European Week of Sport, Open Streets Day will promote environmentally-friendly modes of active transport, as well as encourage citizens to celebrate the joy of being active. Open Streets means opening up the public space to movement which also means open minds – and that is why the events are meant to be accessible and enjoyable for everyone.

Рисунок 1.1 – Сторінка події у додатку Open Streets Day

Окрім опису ініціативи, сторінок самих окремих подій додаток Open Streets Day дозволяє також виконувати пошук подій (рис. 1.2), обираючи рік проведення події, країну, місто, виконуючи пошук за полями подій (назва події, опис події, назва організатора, адреса проведення події).

Тож загалом можна вказати на те, що програмні інструменти для цієї сфери існують, однак вони не спрямовані на рецензування і не зовсім охоплюють інформацію про окремих виконавців.

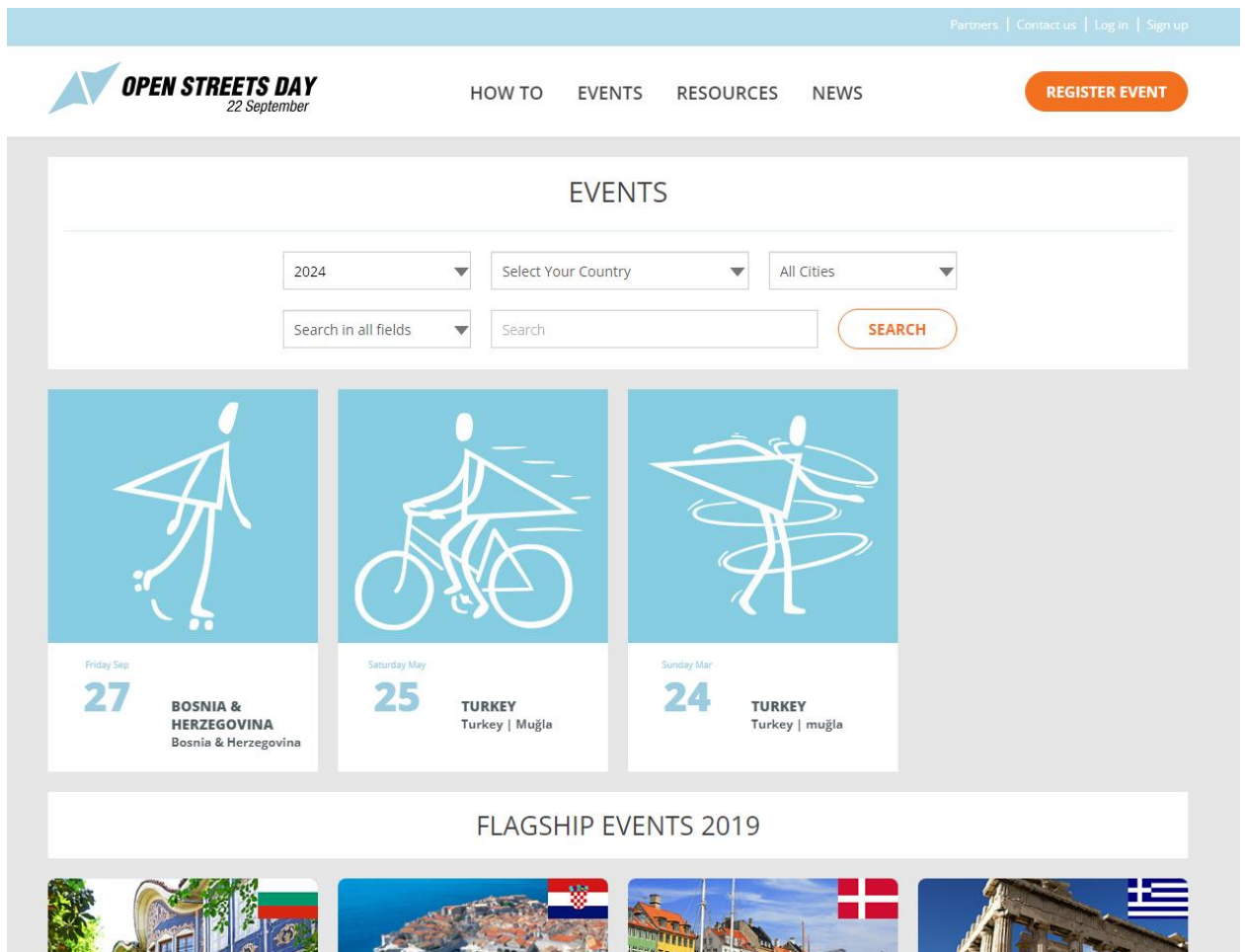


Рисунок 1.2 – Пошук подій у додатку Open Streets Day

1.2 Особливості розробки

Провівши огляд, результати якого були представлені в підрозділі 1.1, далі необхідно було додатково проаналізувати виділений аналог – Open Streets Day та виділити з нього найважливіше для реалізації у розробці цієї роботи, додавши ті можливості, які будуть її відрізняти від аналога та сприяти реалізації власних особливостей.

Виконувана розробка надає можливості реєстрації окремих виконавців вуличних активностей або їх групи, а не тільки фестивалів (подій), а також представлення інформації користувачам про такі вуличні активності, дозволяє створювати запитання за фестивалями або окремими виконавцями, рецензувати фестивалі вуличних активностей, окремі виступи, що відсутнє в додатку Open Streets Day (табл. 1.1).

Таблиця 1.1 – Порівняння розробки з прототипом

Характеристика	Open Streets Day	Розробка
Пошук вуличних активностей	+	+
Представлення інформації про фестивалі вуличних активностей	+	+
Реєстрація власного фестивалю вуличних активностей	+	+
Реєстрація окремих виконавців вуличних активностей або їх групи	—	+
Представлення інформації про окремих виконавців вуличних активностей або їх групи	—	+
Створення запитань за фестивалями або окремими виконавцями	—	+
Рецензування фестивалів вуличних активностей, окремих виступів	—	+

1.3 Висновки за розділом 1

Реалізацію розробки та всієї дипломної кваліфікаційної роботи бакалавра слід розглядати як реалізацію набору завдань:

- аналіз проблеми розробки програми рецензування вуличних активностей;
- визначення вимог до програми рецензування вуличних активностей;
- проєктування програми рецензування вуличних активностей;
- реалізація програми рецензування вуличних активностей.

У розділі проаналізовано програмні інструменти, які стосуються інформування про вуличні активності, серед яких як аналог виділено додаток Open Streets Day. У підсумку вирішено, що виконувана розробка повинна надавати можливості реєстрації окремих виконавців вуличних активностей або їх групи, а також представлення інформації користувачам про такі вуличні активності, дозволяти створювати запитання за фестивалями або окремими виконавцями, рецензувати фестивалі вуличних активностей, окремі виступи, що відсутнє в аналога.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Вибір засобів проєктування і розробки

Порівняння засобів розробки відбувалось як порівняння Django [2]-[3] і WordPress [4]-[5], адже WordPress є однією з найвідоміших систем керування вмістом, а Django – одним з найвідоміших вебфреймворків (табл. 2.1). Однак зважаючи на те, що програма має не тільки стандартну функціональність з додавання і відтворення інформації про вуличні активності, але і пошуку, який потрібно зробити таким, щоб враховував багато параметрів, що зокрема передбачають і обчислення (наприклад, графіку активностей), а також на те, що Django має кращу безпеку роботи, забезпечує роботу з ширшим набором баз даних, надає об'єктно-реляційне відображення (ОРВ) для роботи з базою даних, то обрано було Django.

Таблиця 2.1 – Порівняння рішень для створення програми рецензування вуличних активностей

Показник	Django	WordPress
Тип інструменту	Фреймворк для веброзробки	Система керування вмістом
Доступні інструменти для організації баз даних	PostgreSQL, MariaDB, MySQL, Oracle, SQLite (офіційно) з підтримкою ОРВ	MySQL, MariaDB
Мова	Python	PHP
Безпека роботи	Відповідає дуже високому рівню, забезпечуючи обробку багатьох ситуацій	Може мати проблеми зокрема за рахунок необхідності встановлення рішень третіх сторін для реалізації
Ліцензія	Відкрита	Відкрита

2.2 Визначення функціональних вимог до програмного забезпечення

Функціональні вимоги до програми визначались для наступних типів користувачів:

- неавторизований користувач;
- авторизований користувач, який авторизувався, але не має додаткових прав;
- виконавець групи виконавців вуличних активностей, який є авторизованим користувачем, але до того ж має права представляти групу виконавців вуличних активностей;
- представник організатора вуличних активностей, який є авторизованим користувачем, але до того ж має права представляти організатора вуличних активностей;
- модератор, який є авторизованим користувачем і може керувати правами інших користувачів на представлення організатора вуличних активностей або групи виконавців вуличних активностей.

Неавторизованому користувачу має бути доступним наступний основний потік сценаріїв:

- відтворення сторінки організатора вуличних активностей;
 - пошук організаторів вуличних активностей;
 - відтворення переліку фестивалів вуличних активностей організатора;
 - відтворення сторінки фестивалю вуличних активностей;
 - пошук фестивалів вуличних активностей;
 - відтворення сторінки групи виконавців вуличних активностей;
 - пошук груп виконавців вуличних активностей;
 - відтворення переліку рецензій фестивалю вуличних активностей
- (рис. 2.1).

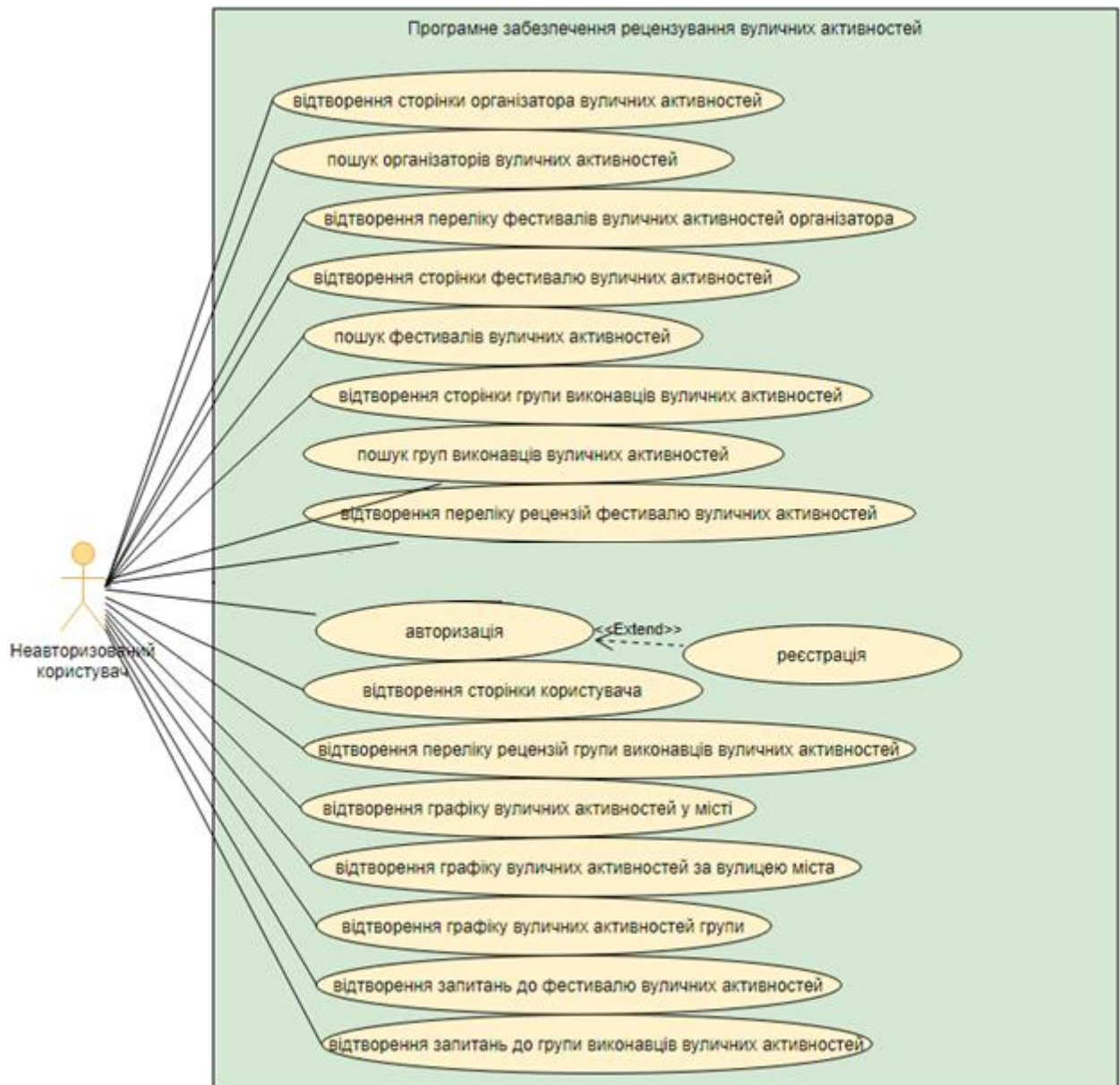


Рисунок 2.1 – Діаграма прецедентів для неавторизованого користувача в програмі рецензування вуличних активностей

Також основний потік формують:

- відтворення переліку рецензій користувача;
- авторизація;
- відтворення сторінки користувача;
- відтворення переліку рецензій групи виконавців вуличних активностей;
- відтворення графіку вуличних активностей у місті;
- відтворення графіку вуличних активностей за вулицею міста;

- відтворення графіку вуличних активностей групи;
- відтворення запитань до фестивалю вуличних активностей;
- відтворення запитань до групи виконавців вуличних активностей.

Сценарій реєстрації є альтернативою авторизації на випадок, коли неавторизований користувач взагалі не має акаунту в програмі.

Авторизованому користувачу має бути доступним наступний основний потік сценаріїв (додатково до неавторизованого):

- внесення даних групи виконавців вуличних активностей, включаючи внесення типів вуличних активностей групи;
- подання заявки на відповідальність за групу виконавців вуличних активностей;
- рецензування фестивалю вуличних активностей;
- рецензування групи виконавців вуличних активностей;
- редагування даних користувача;
- відтворення власних запитань;
- збереження фестивалю вуличних активностей;
- збереження групи виконавців вуличних активностей;
- збереження організатора вуличних активностей;
- створення запитання до фестивалю вуличних активностей;
- створення запитання до групи виконавців вуличних активностей;
- відтворення переліку збережених фестивалів вуличних активностей;
- відтворення переліку збережених груп виконавців вуличних активностей;
- відтворення переліку збережених організаторів вуличних активностей (рис. 2.2).

Альтернативні сценарії:

- сценарій видалення збереження фестивалю вуличних активностей є альтернативою збереження фестивалю вуличних активностей на випадок, коли ці дані збережено, а потрібно змінити це;
- сценарій видалення збереження групи виконавців вуличних активностей є альтернативою збереження групи виконавців вуличних активностей на випадок, коли ці дані збережено, а потрібно змінити це;

– сценарій видалення збереження організатора вуличних активностей є альтернативою збереження організатора вуличних активностей на випадок, коли ці дані збережено, а потрібно змінити це.

Виконавцю групи виконавців вуличних активностей має бути доступним наступний основний потік сценаріїв:

- редагування даних групи виконавців вуличних активностей;
- додавання відповідального за групу виконавців вуличних активностей;
- відтворення переліку груп виконавців вуличних активностей, за які відповідальний користувач;
- створення відповіді на запитання до групи виконавців вуличних активностей (рис. 2.3).

Сценарій вилучення відповідального за групу виконавців вуличних активностей є альтернативою додавання відповідального за групу виконавців вуличних активностей у випадку, коли цей відповідальний вже був доданий, але він більше не є відповідальним за саме цю групу виконавців вуличних активностей.

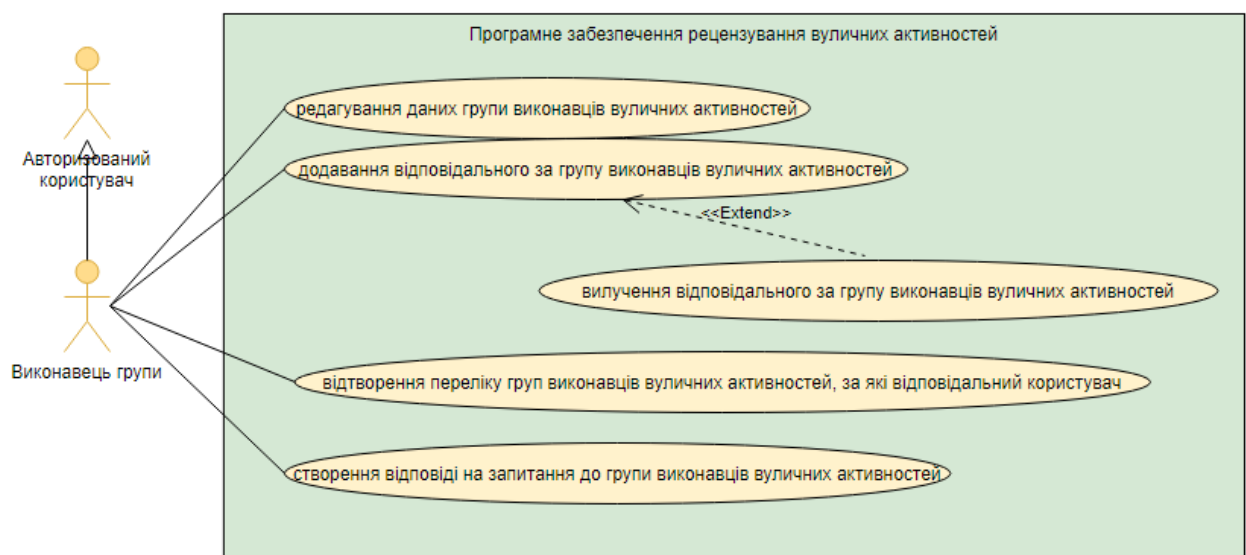


Рисунок 2.3 – Діаграма прецедентів для виконавця групи виконавців вуличних активностей в програмі рецензування вуличних активностей

Представнику організатора вуличних активностей має бути доступним наступний основний потік сценаріїв:

- відтворення переліку організаторів вуличних активностей, які підтримує користувач;
- внесення даних фестивалю вуличних активностей;
- редагування даних організатора вуличних активностей;
- додавання користувача до підтримки організатора вуличних активностей;
- створення відповіді на запитання до фестивалю вуличних активностей.

Альтернативні сценарії:

- сценарій редагування даних фестивалю вуличних активностей є альтернативою для внесення даних фестивалю вуличних активностей у випадку, коли дані змінились;
- сценарій скасування участі користувача в підтримці організатора вуличних активностей є альтернативою додавання користувача до підтримки організатора вуличних активностей у випадку, коли користувач доданий до підтримки організатора вуличних активностей, але це було зроблено випадково (помилково);
- сценарій деактивація участі користувача в підтримці організатора вуличних активностей є альтернативою додавання користувача до підтримки організатора вуличних активностей, коли користувач доданий до підтримки організатора вуличних активностей, але це вже застарілі дані (рис. 2.4).

Модератору має бути доступним наступний основний потік сценаріїв:

- підтримання заявки на відповідальність за групу виконавців вуличних активностей;
- внесення даних організатора вуличних активностей.

Сценарій скасування заявки на відповідальність за групу виконавців вуличних активностей є альтернативою підтримання заявки на

відповідальність за групу виконавців вуличних активностей у випадку, коли така заявка не відповідає дійсності (рис. 2.5).

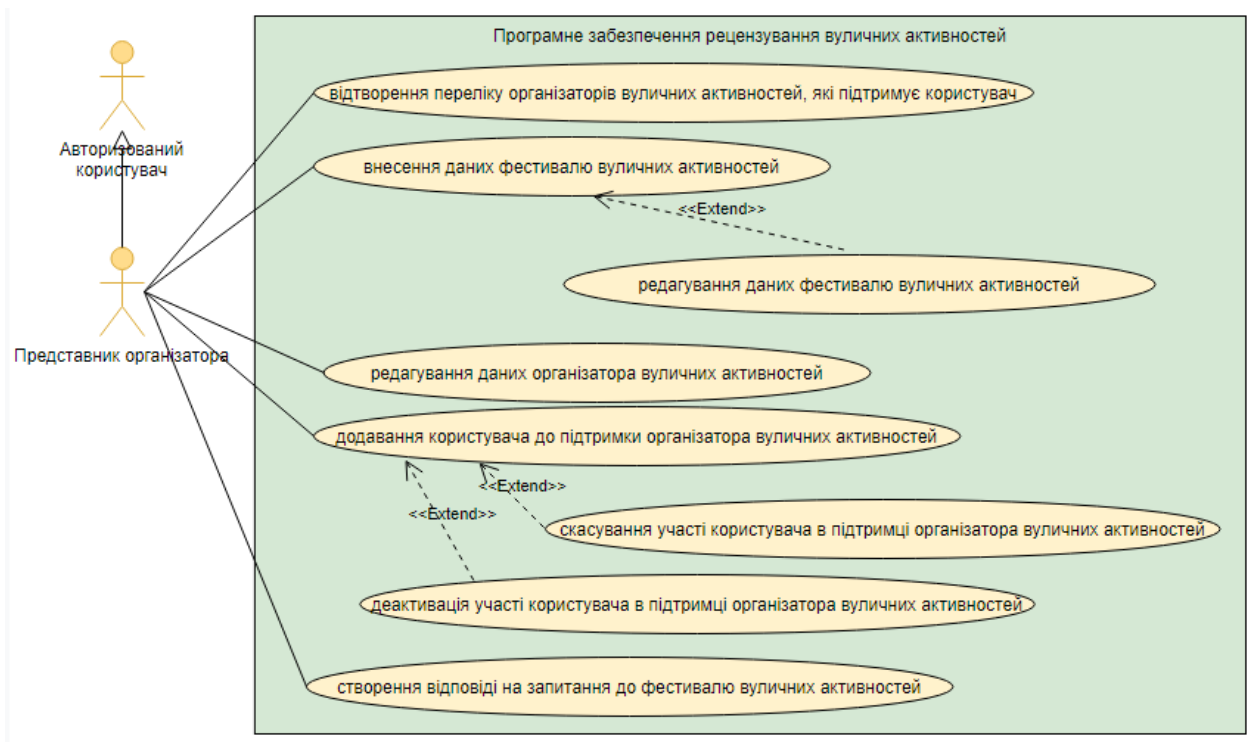


Рисунок 2.4 – Діаграма прецедентів для представника організатора вуличних активностей в програмі рецензування вуличних активностей

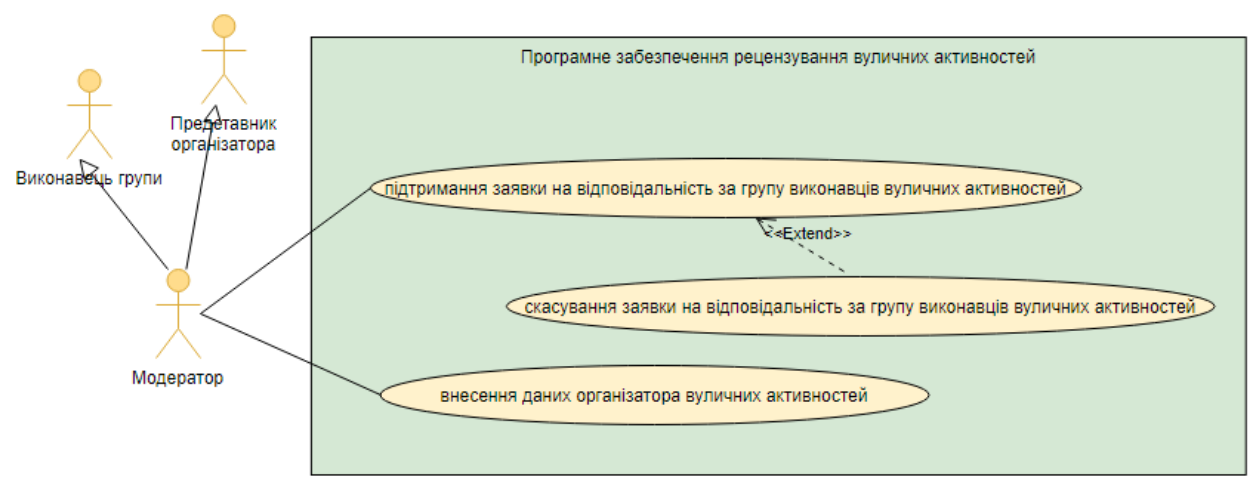


Рисунок 2.5 – Діаграма прецедентів для модератора в програмі рецензування вуличних активностей

2.3 Проектування структури бази даних

MySQL [6] забезпечує підтримку роботи з базою даних (БД), яка описана за структурою нижче. Вибір обумовлений високою продуктивністю, простотою використання, відкритістю, в тому числі й безкоштовністю, та масштабованістю.

Організатори вуличних активностей виділені в таблицю Stractivityorganizer з полями:

- id – ідентифікатор організатора вуличних активностей;
- org_name – іменування організатора вуличних активностей;
- main_city – місто, в якому відбувається основна діяльність організатора вуличних активностей;
- org_address – адреса організатора вуличних активностей в цьому місті;
- org_contact – контактний номер організатора вуличних активностей (за наявності);
- org_about – представлення організатора вуличних активностей текстом;
- org_photo – ілюстративна фотографія організатора вуличних активностей;
- org_website – вебсайт організатора вуличних активностей.

Користувачі, які працюють над функціями організатора вуличних активностей в програмі, виділені в таблицю Stractivityorganizerappuser з полями:

- id – ідентифікатор користувача, який працює над функціями організатора вуличних активностей в програмі;
- stractivityorganizer – організатор вуличних активностей, з функціями якого може працювати цей користувач;
- appuser – користувач, який може працювати над функціями цього організатора вуличних активностей;

– `isactive` – позначення того, чи на даний момент цей користувач все ще може працювати над функціями цього організатора вуличних активностей, чи це вже застарілі дані.

Типи вуличних активностей виділені в таблицю `Stractivitytype` з полями:

- `id` – ідентифікатор типу вуличної активності;
- `activity_name` – назва типу вуличної активності.

Фестивалі вуличних активностей виділені в таблицю `Stractivityfestival` з полями:

- `id` – ідентифікатор фестивалю вуличних активностей;
- `festival_basename` – основна назва фестивалю вуличних активностей;
- `festival_addname` – додаткова назва фестивалю вуличних активностей: наприклад, якщо фестиваль вуличних активностей проходить щорічно або сезонно, то основна назва визначає постійну частину, а додаткова має відношення до року (сезону);

- `festival_organizer` – організатор вуличних активностей, який влаштовує проведення фестивалю;

- `main_stractivitytype` – основний тип вуличних активностей, що представлений на фестивалі;

- `citystreet` – вулиця міста, на якій відбувається фестиваль вуличних активностей;

- `number_start` – будинок на вулиці міста, біля якого відбувається фестиваль вуличних активностей або з якого розпочинається розташування фестивалю вуличних активностей, якщо фестиваль розташований вздовж частини вулиці;

- `number_end` – будинок на вулиці міста, до якого відбувається фестиваль вуличних активностей;

- `festival_start` – час початку фестивалю вуличних активностей;

- `festival_end` – час закінчення фестивалю вуличних активностей;

- `festival_about` – опис проведення фестивалю вуличних активностей;

– review_result – оцінка за результатами рецензування фестивалю вуличних активностей в цій версії (за цей рік, наприклад);

– festival_website – вебсайт фестивалю вуличних активностей (за наявності).

Фотографії фестивалю вуличних активностей виділені в таблицю Stractivityfestivalphoto з полями:

– id – ідентифікатор фотографії фестивалю вуличних активностей;

– stractivityfestival – фестиваль вуличних активностей, де було зроблено цю фотографію;

– photo – фотографія з фестивалю вуличних активностей.

Виконавці вуличних активностей є певною узагальнюючою сутністю. Виконавцем вуличної активності може бути музикант, який грає на гітарі, певний музичний колектив, колектив, який організовує вуличну виставу, продавець товарів власного виробу тощо. Оскільки типи вуличних активностей можуть бути дуже різними, то використано термін саме виконавців вуличних активностей.

Відповідно одна група виконавців вуличних активностей виділяється в таблицю Stractivitybusiness з полями:

– id – ідентифікатор групи виконавців вуличних активностей (група може складатися і з одного виконавця – наприклад, музикант, який грає на гітарі сам);

– business_name – назва групи виконавців вуличних активностей;

– business_about – представлення групи виконавців вуличних активностей;

– review_result – оцінка за результатами рецензування групи виконавців вуличних активностей;

– business_website – вебсайт групи виконавців вуличних активностей (за наявності);

– appuseradded – користувач, який вніс ці дані про групу виконавців вуличних активностей: необхідне поле, оскільки інформацію про себе може

внести не обов'язково сам виконавець, адже він може взагалі не використовувати програму, а це тоді буде заважати користувачам програми взагалі знати про нього, тому вносити інформацію може будь-який користувач, а далі вже виконавець може додатися як користувач і запитати можливість керування даними через програму.

Заявки на те, що певний користувач є виконавцем, який представляє в програмі групу виконавців вуличних активностей, виділені в таблицю `Stractivitybusinessapplication` з полями:

- `id` – ідентифікатор заявки на те, що певний користувач є виконавцем, який представляє в програмі групу виконавців вуличних активностей;
- `stractivitybusiness` – група виконавців вуличних активностей, яку за заявкою представляє в програмі цей користувач;
- `appuser` – користувач, який заявив про себе, що він є представником групи виконавців вуличних активностей, що вже була на цей момент внесена в програму.

Фотографії групи виконавців вуличних активностей виділені в таблицю `Stractivitybusinessphoto` з полями:

- `id` – ідентифікатор фотографії групи виконавців вуличних активностей;
- `stractivitybusiness` – група виконавців вуличних активностей, що зображені на фотографії;
- `photo` – фотографія групи виконавців вуличних активностей.

Типи вуличних активностей, які використовуються групою виділені в таблицю `Stractivitybusinesstype` з полями:

- `id` – ідентифікатор типу вуличних активностей, що використовується групою виконавців;
- `stractivitybusiness` – група виконавців вуличних активностей, що використовує і здійснює цей тип вуличних активностей;
- `stractivitytype` – тип вуличної активності, що використовується і здійснюється цією групою виконавців вуличних активностей.

Звичайно, що група може взагалі реалізовувати вуличні активності тільки одного типу (наприклад, грати джаз), але можливо їхня вулична активність підпадає під декілька типів: і музика, і вулична вистава. Тоді потрібно реалізувати зв'язок багато-до-багатьох, що і робиться цієї таблицею.

Вулиці міста, на якій виступає або реалізує вуличні активності група виконавців, виділена в таблицю `Stractivitybusinessstreet` з полями:

- `id` – ідентифікатор вулиці міста, на якій виступає або реалізує вуличні активності група виконавців;
- `stractivitybusiness` – група виконавців вуличних активностей, яка виступає або реалізує ці активності на цій вулиці;
- `citystreet` – вулиця міста, на якій виступає або реалізує вуличні активності ця група виконавців;
- `number` – номер будинку, біля якого на цій вулиці зазвичай виступає ця група виконавців вуличних активностей (якщо таке відомо).

Таблиця `Stractivitybusinessstreet` необхідна через те, що в різний час або в різних випадках одна й та сама група виконавців вуличних активностей може виступати в різних місцях.

Час стандартних виступів групи виконавців вуличних активностей виділено в таблицю `Stractivitybusinessstreettime` з полями:

- `id` – ідентифікатор часу стандартного виступу групи виконавців вуличних активностей;
- `stractivitybusinessstreet` – конкретна вулиця міста, на якій виступає або реалізує вуличні активності група виконавців у цей час;
- `regular` – стандартний час виступу групи виконавців вуличних активностей або ні: це означає, що група виконавців виступає завжди в цей день у цей час або що такий виступ може відбуватися, а може і ні, але загалом він відбувається у цей час в такий день;
- `week_day` – день тижня, в який у цей час виступає на цій вулиці ця група виконавців вуличних активностей;

– activity_start – час, коли в цей день тижня починається виступ або реалізація вуличних активностей цією групою на цій вулиці міста;

– activity_end – час, коли в цей день тижня закінчується виступ або реалізація вуличних активностей цією групою на цій вулиці міста.

Фестиваль вуличних активностей може чітко визначати ті групи виконавців вуличних активностей, які беруть участь у фестивалі, а така інформація виділяється в таблицю Stractivityfestivalbusiness з полями:

– id – ідентифікатор групи виконавців вуличних активностей, які беруть участь у фестивалі;

– stractivityfestival – фестиваль вуличних активностей, в якому бере участь ця група виконавців;

– stractivitybusiness – група виконавців, яка бере участь у цьому фестивалі вуличних активностей.

Виконавці, які є користувачами програми і входять до складу групи виконавців вуличних активностей, виділені в таблицю Stractivitybusinessman з полями:

– id – ідентифікатор виконавця, який є користувачами програми і входить до складу групи виконавців вуличних активностей;

– stractivitybusiness – група виконавців вуличних активностей, до якої входить цей користувач;

– appuser – користувач програми, який входить до складу групи виконавців вуличних активностей.

Рецензування фестивалів вуличних активностей виділяються в таблицю Stractivityfestivalreview з полями:

– id – ідентифікатор рецензування фестивалів вуличних активностей;

– stractivityfestival – фестиваль вуличних активностей, рецензування якого здійснено цим користувачем;

– appuser – користувач, який рецензував цей фестиваль вуличних активностей;

– review_time – час рецензування фестивалів вуличних активностей;

– review_scale – оцінка, яку фестивалю вуличних активностей виставив користувач;

– review_full – повний текст рецензування фестивалю вуличних активностей.

Рецензування виступів груп виконавців вуличних активностей виділяються в таблицю Stractivitybusinessreview з полями:

– id – ідентифікатор рецензування виступу групи виконавців вуличних активностей;

– stractivitybusiness – група виконавців вуличних активностей, рецензування якої здійснено користувачем;

– appuser – користувач, який рецензував цю групу виконавців вуличних активностей;

– review_time – час рецензування групи виконавців вуличних активностей;

– review_scale – оцінка, яку групі виконавців вуличних активностей виставив користувач;

– review_full – повний текст рецензування групи виконавців вуличних активностей.

Користувачі програми виділені в таблицю Appuser з полями:

– id – ідентифікатор користувача програми;

– django_account – обліковий запис, за яким працює цей користувач;

– displayed_name – ім'я користувача, що відображається в програмі;

– photo – фотографія користувача.

Міста, доступні для визначення вуличних активностей, виділені в таблицю City з полями:

– id – ідентифікатор міста, доступного для визначення вуличних активностей;

– city_name – назва міста, доступного для визначення вуличних активностей;

– country_name – назва країни, де знаходиться місто, доступне для визначення вуличних активностей.

Вулиці міст, доступні для визначення вуличних активностей, виділені в таблицю Citystreet з полями:

– id – ідентифікатор вулиці міста, доступної для визначення вуличних активностей;

– city – місто, в якому знаходиться вулиця, доступна для визначення вуличних активностей;

– street_name – назва вулиці міста, доступної для визначення вуличних активностей.

Збереження фестивалю вуличних активностей виділено в таблицю Stractivityfestivalmemo з полями:

– id – ідентифікатор збереження фестивалю вуличних активностей;

– stractivityfestival – фестиваль вуличних активностей, який збережено користувачем;

– appuser – користувач, який зберіг фестиваль вуличних активностей.

Збереження групи виконавців вуличних активностей виділено в таблицю Stractivitybusinessmemo з полями:

– id – ідентифікатор збереження групи виконавців вуличних активностей;

– stractivitybusiness – група виконавців вуличних активностей, яку збережено користувачем;

– appuser – користувач, який зберіг групу виконавців вуличних активностей.

Збереження організатора вуличних активностей виділено в таблицю Stractivityorganizermemo з полями:

– id – ідентифікатор збереження організатора вуличних активностей користувачем;

– stractivityorganizer – організатор фестивалю вуличних активностей, якого збережено користувачем;

– appuser – користувач, який зберіг організатора вуличних активностей.

Запитання щодо фестивалю вуличних активностей виділено в таблицю Stractivityfestivalque з полями:

- id – ідентифікатор запитання щодо фестивалю вуличних активностей;
- stractivityfestival – фестиваль вуличних активностей, за яким поставлено запитання;
- question_appuser – користувач, який поставив за фестивалем вуличних активностей це запитання;
- question_time – час, коли було поставлено запитання за фестивалем вуличних активностей;
- question_full – саме запитання, поставлене за фестивалем вуличних активностей користувачем.

Відповіді, надані організатором фестивалю на запитання щодо фестивалю вуличних активностей, виділені в таблицю Stractivityfestivalquecomment з полями:

- id – ідентифікатор відповіді, наданої організатором фестивалю на запитання щодо фестивалю вуличних активностей;
- stractivityfestivalque – запитання щодо фестивалю вуличних активностей, до якого ця відповідь адресується;
- comment_appuser – користувач, який створив цю відповідь на запитання щодо фестивалю вуличних активностей;
- comment_time – час, коли було надано цю відповідь на запитання щодо фестивалю вуличних активностей;
- comment_full – текст відповіді на запитання щодо фестивалю вуличних активностей.

Запитання щодо діяльності групи виконавців вуличних активностей виділені в таблицю Stractivitybusinessque з полями:

- id – ідентифікатор запитання щодо діяльності групи виконавців вуличних активностей;

- stractivitybusiness – група виконавців вуличних активностей, щодо якої адресовано запитання;
- question_appuser – користувач, який поставив за групою виконавців вуличних активностей це запитання;
- question_time – час, коли було поставлено запитання за групою виконавців вуличних активностей;
- question_full – саме запитання, поставлене за групою виконавців вуличних активностей користувачем (рис. 2.6).

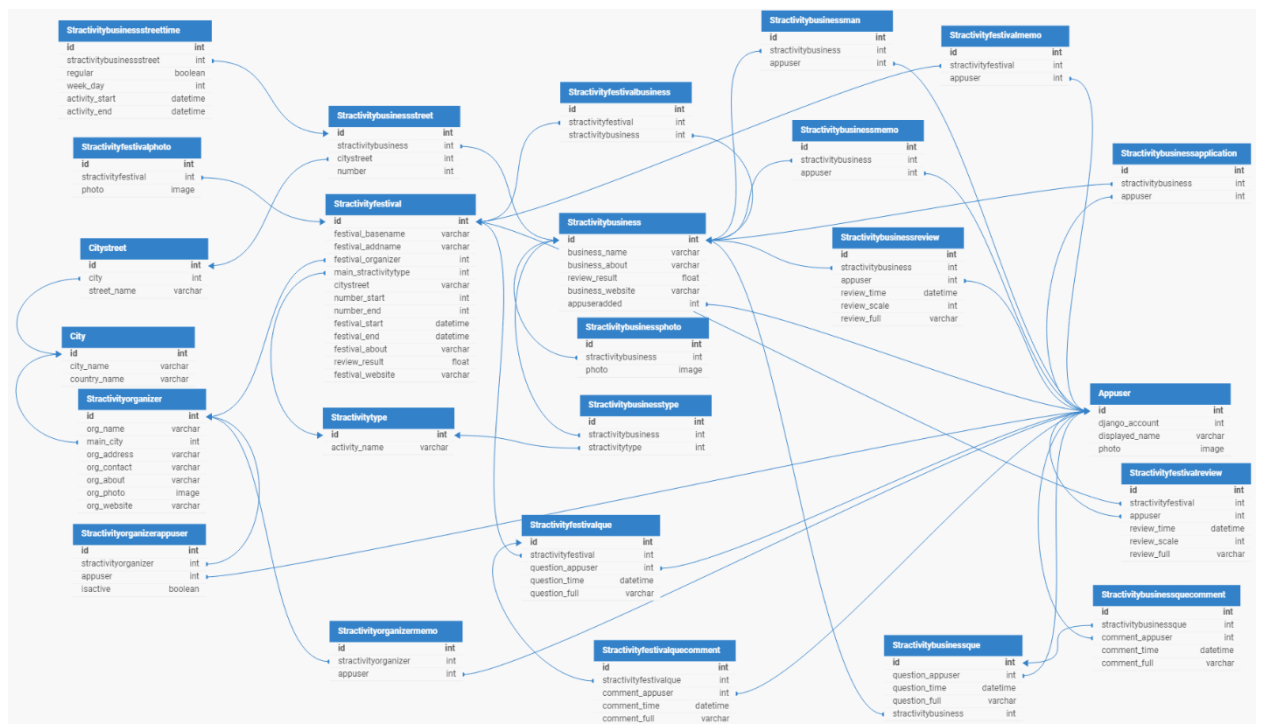


Рисунок 2.6 – Схема БД програми рецензування вуличних активностей

Відповіді, надані на запитання щодо групи виконавців вуличних активностей, виділені в таблицю Stractivitybusinessquecomment з полями:

- id – ідентифікатор відповіді, наданої на запитання щодо групи виконавців вуличних активностей;
- stractivitybusinessque – запитання щодо діяльності групи виконавців вуличних активностей, за яким було надано відповідь;

- comment_appuser – користувач, який створив цю відповідь на запитання щодо групи виконавців вуличних активностей;
- comment_time – час, коли було надано цю відповідь на запитання щодо групи виконавців вуличних активностей;
- comment_full – текст відповіді на запитання щодо групи виконавців вуличних активностей.

2.4 Висновки за розділом 2

Програма рецензування вуличних активностей за результатами виконання даного розділу має визначену структуру роботи з даними через встановлення таблиць і зв'язків у БД за допомогою системи MySQL, має обрані засоби розробки у вигляді Python-фреймворку для веброзробки Django, має встановлені сценарії використання.

3 РОЗРОБКА ПРОГРАМИ

3.1 Порядок реалізації функцій програми

Усі функції програми упорядковані під час роботи з користувачем, надаючи йому тільки певний базовий набір до авторизації (рис. 3.1).



Рисунок 3.1 – Схема функціонування програми рецензування вуличних активностей

Після авторизації можливо виконати або ті функції, які доступні авторизованому користувачу загалом, або додаткові в тому випадку, якщо цей авторизований користувач є модератором або він додатково має повноваження представляти організатора вуличних активностей.

3.2 Реалізація функцій програми

Функції програми забезпечено створенням програмних функцій та класів. Програмні функції реалізують представлення додатку, а класи – моделі.

Функція відтворення сторінки організатора вуличних активностей забезпечена реалізацією функції `street_activity_organizer_view` з параметром `id_stractivityorganizer`, де він визначає ідентифікатор організатора вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `saorganizer.html` даними про самого організатора, про 5 останніх фестивалів, які ним організовуються, про збереження організатора вуличних активностей користувачем, що авторизувався (якщо він авторизувався). У розробці шаблонів сторінок використано шаблон [7], що є вільним.

Функція відтворення переліку фестивалів вуличних активностей організатора забезпечена реалізацією функції `street_activity_organizer_festivals_view` з параметром `id_stractivityorganizer`, де він визначає ідентифікатор організатора вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `saorgfestivals.html` даними про організатора, всі його фестивалі (як заплановані в майбутньому, так і ті, що минули), про збереження організатора вуличних активностей та відповідних фестивалів користувачем, що авторизувався (якщо він авторизувався).

Функція відтворення сторінки фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_view` з параметром `id_stractivityfestival`, де він визначає ідентифікатор фестивалю вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `safestival.html` даними про фестиваль вуличних активностей, фотографіями

цього фестивалю, 5 останніми рецензуваннями фестивалю, 3 останніми запитаннями про фестиваль з відповідями на них, даними про збереження користувачем, який авторизувався (якщо він авторизувався), цього фестивалю вуличних активностей, про роль користувача як представника організатора цього фестивалю, про групи виконавців вуличних активностей, які беруть участь у фестивалі, про збереження користувачем цим груп.

Функція відтворення сторінки групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_view` з параметром `id_stractivitybusiness`, де він визначає ідентифікатор групи виконавців вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `sabusiness.html` даними про саму групу виконавців вуличних активностей, 5 останніх рецензувань цієї групи виконавців вуличних активностей, фотографії виступів, 3 останні запитання щодо групи виконавців вуличних активностей, дані про те, чи є авторизований користувач виконавцем групи, про фестивалі, в яких бере участь група, про адреси, за якими група виконавців вуличних активностей має представлення.

Функція відтворення переліку рецензій групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_reviews_view` з параметром `id_stractivitybusiness`, де він визначає ідентифікатор групи виконавців вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `sabusinessreviews.html` даними про цю групу виконавців вуличних активностей, 3 фотографіями цієї групи, всіма рецензуваннями цієї групи виконавців вуличних активностей, інформацією про те, чи виконував рецензування авторизований користувач (якщо він авторизувався), чи зберіг він цю групу собі, чи є він сам виконавцем цієї групи або чи подавав заявку щодо представництва.

Функція відтворення переліку рецензій фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_reviews_view` з параметром `id_stractivityfestival`, де він визначає ідентифікатор фестивалю вуличних активностей. Результатом є

вебсторінка, заповнена на основі шаблону `safestreviews.html` даними про цей фестиваль вуличних активностей, 3 фотографіями з цього фестивалю, всіма рецензуваннями цього фестивалю вуличних активностей, інформацією про те, чи виконував рецензування авторизований користувач (якщо він авторизувався), чи зберіг він цей фестиваль собі, чи є він представником організатора цього фестивалю.

Функція відтворення переліку рецензій користувача забезпечена реалізацією функції `street_activity_appuser_reviews_view`. Результатом є вебсторінка, заповнена на основі шаблону `sauserreviews.html` даними про всі рецензування фестивалів вуличних активностей, які виконав авторизований користувач, та рецензування груп виконавців вуличних активностей, які виконав цей авторизований користувач.

Функція відтворення запитань до фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_questions_view` з параметром `id_stractivityfestival`, де він визначає ідентифікатор фестивалю вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `safestquestions.html` даними про базову інформацію самого фестивалю вуличних активностей, запитання та відповіді щодо даного фестивалю

Функція відтворення запитань до групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_questions_view` з параметром `id_stractivitybusiness`, де він визначає ідентифікатор групи виконавців вуличних активностей. Результатом є вебсторінка, заповнена на основі шаблону `sabusquestions.html` даними про базову інформацію самої групи виконавців вуличних активностей та всі запитання з відповідями до них стосовно цієї групи виконавців вуличних активностей.

Функція відтворення власних запитань забезпечена реалізацією функції `street_activity_appuser_questions_view`. Результатом є вебсторінка, заповнена на основі шаблону `saappuserquestions.html` даними про базову

інформацію стосовно самого користувача, всі запитання, які були поставлені цим авторизованим користувачем стосовно всіх фестивалів вуличних активностей, груп виконавців вуличних активностей, надані відповіді на ці запитання.

Функція пошуку організаторів вуличних активностей забезпечена реалізацією функції `street_activity_organizer_search_view`. Значення пошукових параметрів передаються через GET-запит та використовуються для виділення тих організаторів вуличних активностей, що відповідають цим параметрам. До параметрів входять:

- назва організатора вуличних активностей;
- місто, до якого належить організатор вуличних активностей;
- країна, де є організатор вуличних активностей.

Функція пошуку фестивалів вуличних активностей забезпечена реалізацією функції `street_activity_festival_search_view`. Значення пошукових параметрів передаються через GET-запит та використовуються для виділення тих фестивалів вуличних активностей, що відповідають цим параметрам. До параметрів входять:

- назва фестивалю вуличних активностей;
- місто, в якому проводиться фестиваль вуличних активностей;
- країна, в якій проводиться фестиваль вуличних активностей;
- назва організатора фестивалю вуличних активностей;
- вулиця міста, на якій проводиться фестиваль вуличних активностей;
- період проведення фестивалю вуличних активностей;
- мінімальна прийнятна оцінка минулих проведення фестивалю вуличних активностей.

Функція пошук груп виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_search_view`. Значення пошукових параметрів передаються через GET-запит та використовуються для виділення тих груп виконавців вуличних активностей, що відповідають цим параметрам. До параметрів входять:

- назва групи виконавців вуличних активностей;
- мінімальний прийнятний бал рецензування;
- місто, в якому виступає група виконавців вуличних активностей;
- вулиця, на якій виступає група виконавців вуличних активностей;
- типи вуличних активностей, з якими виступають виконавці групи.

Функція відтворення графіку вуличних активностей у місті забезпечена реалізацією функції `street_activity_city_schedule_view` з параметрами `id_city`, який задає місто, в якому ці вуличні активності мають відбуватися, `period_start` та `period_end`, що визначають дати початку і закінчення інтервалу, за яким має надаватися графік. Відбираються всі вуличні активності, в тому числі фестивалі, заплановані стандартні виступи груп виконавців вуличних активностей, що відповідають цим параметрам, заповнюючи шаблон вебсторінки `sacityschedule.html`.

Функція відтворення графіку вуличних активностей за вулицею міста забезпечена реалізацією функції `street_activity_street_schedule_view` з параметрами `id_street`, який задає вулицю міста, на якій ці вуличні активності мають відбуватися, `period_start` та `period_end`, що визначають дати початку і закінчення інтервалу, за яким має надаватися графік. Відбираються всі вуличні активності, в тому числі фестивалі, заплановані стандартні виступи груп виконавців вуличних активностей, що відповідають цим параметрам, заповнюючи шаблон вебсторінки `sastreetschedule.html`.

Функція відтворення графіку вуличних активностей групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_schedule_view` з параметрами `id_business`, який задає групу виконавців вуличних активностей, за якою мають відбиратися вуличні активності, `period_start` та `period_end`, що визначають дати початку і закінчення інтервалу, за яким має надаватися графік. Відбираються всі вуличні активності, в тому числі фестивалі, заплановані стандартні виступи цієї групи виконавців вуличних активностей, що відповідають цим параметрам, заповнюючи шаблон вебсторінки `sabusinessschedule.html`.

Функція створення запитання до фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_question_view` з параметром `id_festival`, який задає номер фестивалю вуличних активностей, а саме запитання забезпечується формою `StractivityfestivalqueForm`, у випадку валідності даних в якій результати зберігаються.

Функція створення запитання до групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_question_view` з параметром `id_business`, який задає номер групи виконавців вуличних активностей, а саме запитання забезпечується формою `StractivitybusinessqueForm`, у випадку валідності даних в якій результати зберігаються.

Функція збереження фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_memo_view` з параметром `festival_num`, який задає номер фестивалю вуличних активностей, а авторство збереження визначається за авторизацією. Функція працює так, що якщо збереження цим же користувачем цього ж фестивалю вуличних активностей вже відбулось, тобто в БД ця інформація вже внесена, то реалізується видалення з БД, тобто видалення збереження.

Функція збереження групи виконавців вуличних активностей забезпечена реалізацією функції `street_activity_business_memo_view` з параметром `business_num`, який задає номер групи виконавців вуличних активностей, а авторство збереження визначається за авторизацією. Функція працює так, що якщо збереження цим же користувачем цієї ж групи виконавців вуличних активностей вже відбулось, тобто в БД ця інформація вже внесена, то реалізується видалення з БД, тобто видалення збереження.

Функція організатора вуличних активностей забезпечена реалізацією функції `street_activity_organizer_memo_view` з параметром `organizer_num`, який задає номер організатора вуличних активностей, а авторство збереження визначається за авторизацією. Функція працює так, що якщо збереження цим же користувачем цього ж організатора вуличних активностей вже відбулось,

тобто в БД ця інформація вже внесена, то реалізується видалення з БД, тобто видалення збереження.

Функція внесення даних фестивалю вуличних активностей забезпечена реалізацією функції `street_activity_festival_add_view`. Створена функція через підтримку на основі класу форми `StractivityfestivalForm`.

Функція редагування даних організатора вуличних активностей забезпечена реалізацією функції `street_activity_festival_edit_view` з параметром `festival_num`, який задає ідентифікатор організатора вуличних активностей, змінювання даних якого на основі форми `StractivityfestivalForm` має відбуватися. Збереження змін і надання даних у формі реалізується тоді, коли авторизований користувач є представником організатора вуличних активностей.

Аналогічно реалізовані всі функції за функціональними вимогами.

Моделі реалізовані для підтримки ОРВ і забезпечення роботи з даними.

Клас моделі організаторів вуличних активностей `Stractivityorganizer` має визначений набір полів.

Поле `org_name` забезпечує іменування організатора вуличних активностей типом `CharField`.

Поле `main_city` забезпечує місто, в якому відбувається основна діяльність організатора вуличних активностей, типом `ForeignKey` за моделлю `City`.

Поле `org_address` забезпечує адресу організатора вуличних активностей в цьому місті типом `CharField`.

Поле `org_contact` забезпечує контактний номер організатора вуличних активностей типом `CharField`.

Поле `org_about` забезпечує представлення організатора вуличних активностей текстом типом `CharField`.

Поле `org_photo` забезпечує ілюстративну фотографію організатора вуличних активностей типом `ImageField`.

Поле `org_website` забезпечує вебсайт організатора вуличних активностей типом `URLField`.

Клас моделі користувачів, які працюють над функціями організатора вуличних активностей в програмі, `Stractivityorganizerappuser` має визначений набір полів.

Поле `stractivityorganizer` забезпечує організатора вуличних активностей, з функціями якого може працювати цей користувач, типом `ForeignKey` за моделлю `Stractivityorganizer`.

Поле `appuser` забезпечує користувача, який може працювати над функціями цього організатора вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `isactive` забезпечує типом `BooleanField` позначення того, чи на даний момент цей користувач все ще може працювати над функціями цього організатора вуличних активностей, чи це вже застарілі дані.

Клас моделі типу вуличних активностей `Stractivitytype` має визначене поле `activity_name`, що забезпечує назва типу вуличної активності типом `CharField`.

Клас моделі фестивалю вуличних активностей `Stractivityfestival` має визначений набір полів.

Поле `festival_basename` забезпечує основну назву фестивалю вуличних активностей типом `CharField`.

Поле `festival_addname` забезпечує додаткову назву фестивалю вуличних активностей типом `CharField`.

Поле `festival_organizer` забезпечує організатора вуличних активностей, який влаштовує проведення фестивалю, типом `ForeignKey` за моделлю `Stractivityorganizer`.

Поле `main_stractivitytype` забезпечує основний тип вуличних активностей, що представлений на фестивалі, типом `ForeignKey` за моделлю `Stractivitytype`.

Поле `citystreet` забезпечує вулицю міста, на якій відбувається фестиваль вуличних активностей, типом `ForeignKey` за моделлю `Citystreet`.

Поле `number_start` забезпечує будинок на вулиці міста, біля якого відбувається фестиваль вуличних активностей або з якого розпочинається розташування фестивалю вуличних активностей, якщо фестиваль розташований вздовж частини вулиці, типом `PositiveIntegerField`.

Поле `number_end` забезпечує будинок на вулиці міста, до якого відбувається фестиваль вуличних активностей, типом `PositiveIntegerField`.

Поле `festival_start` забезпечує час початку фестивалю вуличних активностей типом `DateTimeField`.

Поле `festival_end` забезпечує час закінчення фестивалю вуличних активностей типом `DateTimeField`.

Поле `festival_about` забезпечує опис проведення фестивалю вуличних активностей типом `CharField`.

Поле `review_result` забезпечує оцінку за результатами рецензування фестивалю вуличних активностей в цій версії типом `FloatField`.

Поле `festival_website` забезпечує вебсайт фестивалю вуличних активностей типом `URLField`.

Поле `stractivitybusinesses` забезпечує групи виконавців вуличних активностей, які виступають на цьому фестивалі, типом `ManyToManyField` за моделлю `Stractivitybusiness`.

Клас моделі фотографії фестивалю вуличних активностей `Stractivityfestivalphoto` має визначений набір полів.

Поле `stractivityfestival` забезпечує фестиваль вуличних активностей, де було зроблено цю фотографію, типом `ForeignKey` за моделлю `Stractivityfestival`.

Поле `photo` забезпечує фотографію з фестивалю вуличних активностей за моделлю `ImageField`.

Клас моделі групи виконавців вуличних активностей `Stractivitybusiness` має визначений набір полів.

Поле `business_name` забезпечує назву групи виконавців вуличних активностей типом `CharField`.

Поле `business_about` забезпечує представлення групи виконавців вуличних активностей типом `CharField`.

Поле `review_result` забезпечує оцінку за результатами рецензування групи виконавців вуличних активностей типом `FloatField`.

Поле `business_website` забезпечує вебсайт групи виконавців вуличних активностей типом `URLField`.

Поле `appuseradded` забезпечує користувача, який вніс ці дані про групу виконавців вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `stractivitytypes` забезпечує типи вуличних активностей, які виконуються цією групою виконавців вуличних активностей, типом `ManyToManyField` за моделлю `Stractivitytype`.

Поле `appusers` забезпечує перелік користувачів, які представляють цю групу виконавців вуличних активностей, типом `ManyToManyField` за моделлю `Appuser`.

Клас моделі заявки на те, що певний користувач є виконавцем, який представляє в програмі групу виконавців вуличних активностей, `Stractivitybusinessapplication` має визначений набір полів.

Поле `stractivitybusiness` забезпечує групу виконавців вуличних активностей, яку за заявкою представляє в програмі цей користувач, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `appuser` забезпечує користувача, який заявив про себе, що він є представником групи виконавців вуличних активностей, що вже була на цей момент внесена в програму. Тип поля – `ForeignKey` за моделлю `Appuser`.

Клас моделі фотографії групи виконавців вуличних активностей `Stractivitybusinessphoto` має визначений набір полів.

Поле `stractivitybusiness` забезпечує групу виконавців вуличних активностей, що зображені на фотографії, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `photo` забезпечує фотографію групи виконавців вуличних активностей типом `CharField`.

Клас моделі вулиць міста, на яких виступає або реалізує вуличні активності група виконавців, `Stractivitybusinessstreet` має визначений набір полів.

Поле `stractivitybusiness` забезпечує групу виконавців вуличних активностей, яка виступає або реалізує ці активності на цій вулиці, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `citystreet` забезпечує вулицю міста, на якій виступає або реалізує вуличні активності ця група виконавців, типом `ForeignKey` за моделлю `Citystreet`.

Поле `number` забезпечує номер будинку, біля якого на цій вулиці зазвичай виступає ця група виконавців вуличних активностей, типом `SmallIntegerField`.

Клас моделі часу стандартних виступів групи виконавців вуличних активностей `Stractivitybusinessstreettime` має визначений набір полів.

Поле `stractivitybusinessstreet` забезпечує конкретну вулицю міста, на якій виступає або реалізує вуличні активності група виконавців у цей час, типом `ForeignKey` за моделлю `Stractivitybusinessstreet`.

Поле `regular` забезпечує визначення того, чи це стандартний час виступу групи виконавців вуличних активностей або ні, типом `BooleanField`.

Поле `week_day` забезпечує день тижня, в який у цей час виступає на цій вулиці ця група виконавців вуличних активностей, типом `PositiveSmallIntegerField`.

Поле `activity_start` забезпечує час, коли в цей день тижня починається виступ або реалізація вуличних активностей цією групою на цій вулиці міста, типом `DateTimeField`.

Поле `activity_end` забезпечує час, коли в цей день тижня закінчується виступ або реалізація вуличних активностей цією групою на цій вулиці міста, типом `DateTimeField`.

Клас моделі рецензування фестивалів вуличних активностей `Stractivityfestivalreview` має визначений набір полів.

Поле `stractivityfestival` забезпечує фестиваль вуличних активностей, рецензування якого здійснено цим користувачем, типом `ForeignKey` за моделлю `Stractivityfestival`.

Поле `appuser` забезпечує користувача, який рецензував цей фестиваль вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `review_time` забезпечує час рецензування фестивалів вуличних активностей типом `DateTimeField`.

Поле `review_scale` забезпечує оцінку, яку фестивалю вуличних активностей виставив користувач, типом `PositiveSmallIntegerField`.

Поле `review_full` забезпечує повний текст рецензування фестивалю вуличних активностей типом `CharField`.

Клас моделі рецензування виступів груп виконавців вуличних активностей `Stractivitybusinessreview` має визначений набір полів.

Поле `stractivitybusiness` забезпечує групу виконавців вуличних активностей, рецензування якої здійснено користувачем, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `appuser` забезпечує користувача, який рецензував цю групу виконавців вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `review_time` забезпечує час рецензування групи виконавців вуличних активностей типом `DateTimeField`.

Поле `review_scale` забезпечує оцінку, яку групі виконавців вуличних активностей виставив користувач, типом `PositiveSmallIntegerField`.

Поле `review_full` забезпечує повний текст рецензування групи виконавців вуличних активностей типом `CharField`.

Клас моделі користувачів програми `Appuser` має визначений набір полів.

Поле `django_account` забезпечує обліковий запис, за яким працює цей користувач, типом `ForeignKey` за моделлю `User`, базовою для Django.

Поле `displayed_name` забезпечує ім'я користувача, що відображається в програмі, типом `CharField`.

Поле `photo` забезпечує фотографію користувача типом `ImageField`.

Поле `stractivityfestivals` забезпечує перелік фестивалів вуличних активностей, які збережено користувачем, типом `ForeignKey` за моделлю `Stractivityfestival`.

Поле `stractivitybusinesses` забезпечує перелік груп виконавців вуличних активностей, які збережено користувачем, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `stractivityorganizers` забезпечує перелік Організаторів вуличних активностей, які збережено користувачем, типом `ForeignKey` за моделлю `Stractivityorganizer`.

Клас міст, доступних для визначення вуличних активностей, `City` має визначений набір полів.

Поле `city_name` забезпечує назву міста, доступного для визначення вуличних активностей, типом `CharField`.

Поле `country_name` забезпечує назву країни, де знаходиться місто, доступне для визначення вуличних активностей, типом `CharField`.

Клас моделі вулиць міст, доступних для визначення вуличних активностей, `Citystreet` має визначений набір полів.

Поле `city` забезпечує місто, в якому знаходиться вулиця, доступна для визначення вуличних активностей, типом `ForeignKey` за моделлю `City`.

Поле `street_name` забезпечує назву вулиці міста, доступної для визначення вуличних активностей, типом `CharField`.

Клас моделі запитання щодо фестивалю вуличних активностей `Stractivityfestivalque` має визначений набір полів.

Поле `stractivityfestival` забезпечує фестиваль вуличних активностей, за яким поставлено запитання, типом `ForeignKey` за моделлю `Stractivityfestival`.

Поле `question_appuser` забезпечує користувача, який поставив за фестивалем вуличних активностей це запитання, типом `ForeignKey` за моделлю `Appuser`.

Поле `question_time` забезпечує час, коли було поставлено запитання за фестивалем вуличних активностей, типом `DateTimeField`.

Поле `question_full` забезпечує саме запитання, поставлене за фестивалем вуличних активностей користувачем, типом `CharField`.

Клас моделі відповіді, наданої організатором фестивалю на запитання щодо фестивалю вуличних активностей, `Stractivityfestivalquecomment` має визначений набір полів.

Поле `stractivityfestivalque` забезпечує запитання щодо фестивалю вуличних активностей, до якого ця відповідь адресується, типом `ForeignKey` за моделлю `Stractivityfestivalque`.

Поле `comment_appuser` забезпечує користувача, який створив цю відповідь на запитання щодо фестивалю вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `comment_time` забезпечує час, коли було надано цю відповідь на запитання щодо фестивалю вуличних активностей, типом `DateTimeField`.

Поле `comment_full` забезпечує текст відповіді на запитання щодо фестивалю вуличних активностей типом `CharField`.

Клас моделі запитання щодо діяльності групи виконавців вуличних активностей `Stractivitybusinessque` має визначений набір полів.

Поле `stractivitybusiness` забезпечує групу виконавців вуличних активностей, щодо якої адресовано запитання, типом `ForeignKey` за моделлю `Stractivitybusiness`.

Поле `question_appuser` забезпечує користувача, який поставив за групою виконавців вуличних активностей це запитання, типом `ForeignKey` за моделлю `Appuser`.

Поле `question_time` забезпечує час, коли було поставлено запитання за групою виконавців вуличних активностей, типом `DateTimeField`.

Поле `question_full` забезпечує саме запитання, поставлене за групою виконавців вуличних активностей користувачем, типом `CharField`.

Клас моделі відповіді, наданої на запитання щодо групи виконавців вуличних активностей, `Stractivitybusinessquecomment` має визначений набір полів.

Поле `stractivitybusinessque` забезпечує запитання щодо діяльності групи виконавців вуличних активностей, за яким було надано відповідь, типом `ForeignKey` за моделлю `Stractivitybusinessque`.

Поле `comment_appuser` забезпечує користувача, який створив цю відповідь на запитання щодо групи виконавців вуличних активностей, типом `ForeignKey` за моделлю `Appuser`.

Поле `comment_time` забезпечує час, коли було надано цю відповідь на запитання щодо групи виконавців вуличних активностей, типом `DateTimeField`.

Поле `comment_full` забезпечує текст відповіді на запитання щодо групи виконавців вуличних активностей типом `CharField`.

3.3 Висновки за розділом 3

Усі заплановані для реалізації функціональні можливості програми рецензування вуличних активностей програмно реалізовано. Основні класи і функції, забезпечуючи реалізацію на основі мультипарадигмального програмування таким чином, описані.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Програма надає можливість отримувати інформацію про вуличні активності, які проводяться у місті, час таких виступів, рецензії від інших користувачів, які стали свідками таких активностей

4.2 Умови виконання програми

Експлуатація програми операторами в різних ролях не потребує особливих навичок окрім навичок роботи з вебдодатками, але налаштування серверу має враховувати навички у адміністратора для роботи з системою MySQL для забезпечення бази даних, керування Python-фреймворком веброзробки Django (зокрема і роботи з адміністративною панеллю), інтерпретатором Python.

Технічні засоби визначені на мінімальному рівні, охоплюючи наступні вимоги: процесор не менше чотирьохядерного з частотою 2,0 ГГц, оперативна пам'ять – 4 Гб, жорсткий диск – 20 Гб простору.

4.3 Виконання і тестування програми

У цьому підрозділі описано логіку виконання програми, що слідує для користувача, який звертається до вебдодатку з клієнтської сторони. На основі даної логіки було виконано звернення до сервера та протестовано роботу кожної функціональної можливості, що була визначена для виконання. Усі вони в підсумку були успішно протестовані.

Будь-який користувач при зверненні до сервера з клієнта має основні засоби керування, розташовані в меню (рис. 4.1). Для неавторизованого користувача меню складається з 4 розділів: виконавців, активностей, фестивалів та організаторів, а також кнопки для виконання авторизації.

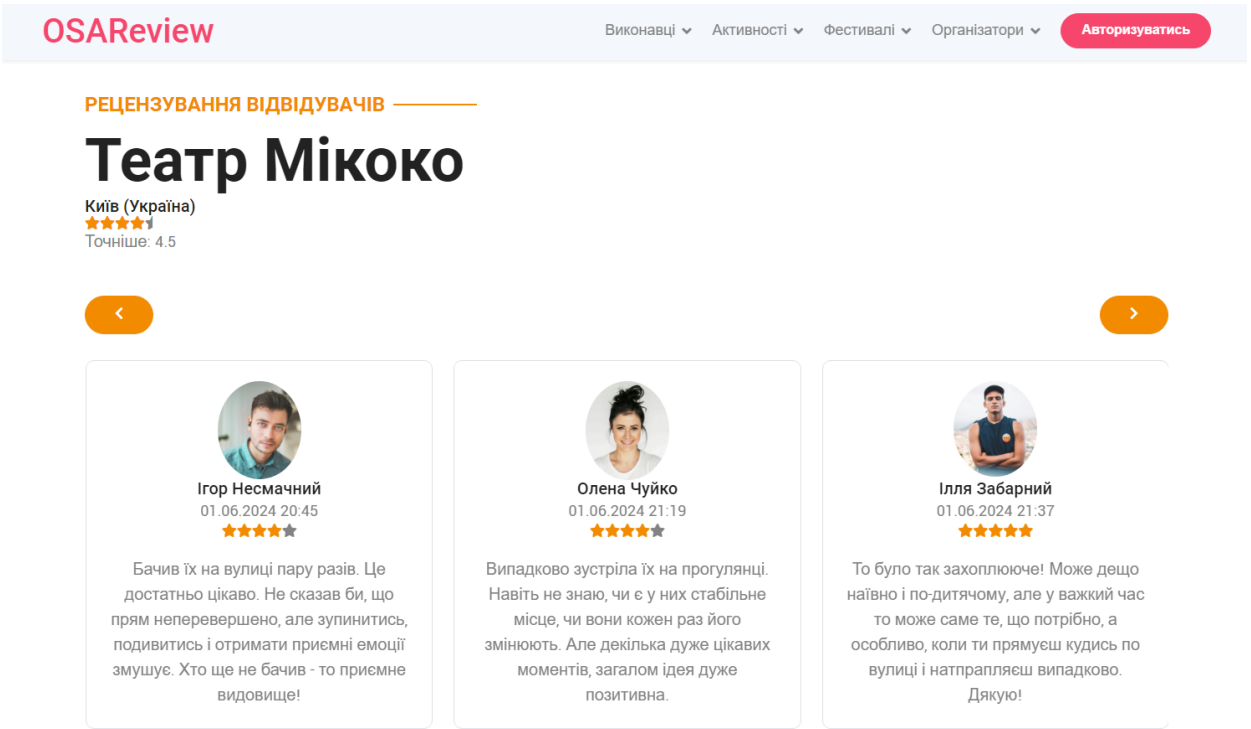


Рисунок 4.1 – Рецензування групи виконавців вуличних активностей для неавторизованого користувача

Розділ виконавців містить можливості доступу до груп виконавців вуличних активностей, включаючи пошук таких груп виконавців вуличних активностей, що є єдиною функцією доступною для неавторизованого користувача, а також внесення даних, відтворення збережених груп виконавців вуличних активностей, відтворення переліку тих виконавців, за яких є відповідальним користувач. Але всі 3 функції останні доступні тільки після авторизації. Для демонстрації відмінностей між авторизованим та неавторизованим користувачем було розглянуто сторінку рецензування групи виконавців вуличних активностей, до якої можна перейти зі сторінки самої групи виконавців вуличних активностей. У режимі неавторизованого користувача (рис. 4.1) ця сторінка містить тільки середній бал, самі рецензування з можливістю перемикатися між ними стрілками, а вище цього переліку 3 фотографії групи виконавців вуличних активностей. У режимі авторизованого користувача (рис. 4.2) сторінка містить у меню функцію переходу до даних користувачів, звідки можна ці дані не тільки переглядати,

але і редагувати (власні), функцію переходу до поставлених авторизованим користувачем запитань. Також тоді на сторінці може відображатися позначка про те, що користувач є учасником групи або посилання для подання заявки щодо представлення групи виконавців вуличних активностей. Але якщо ці дані вже були початково внесені виконавцем, який представляє групу, то така позначка не відображається. Окрім того авторизованому користувачу доступна кнопка для збереження групи та для створення нового відгуку.

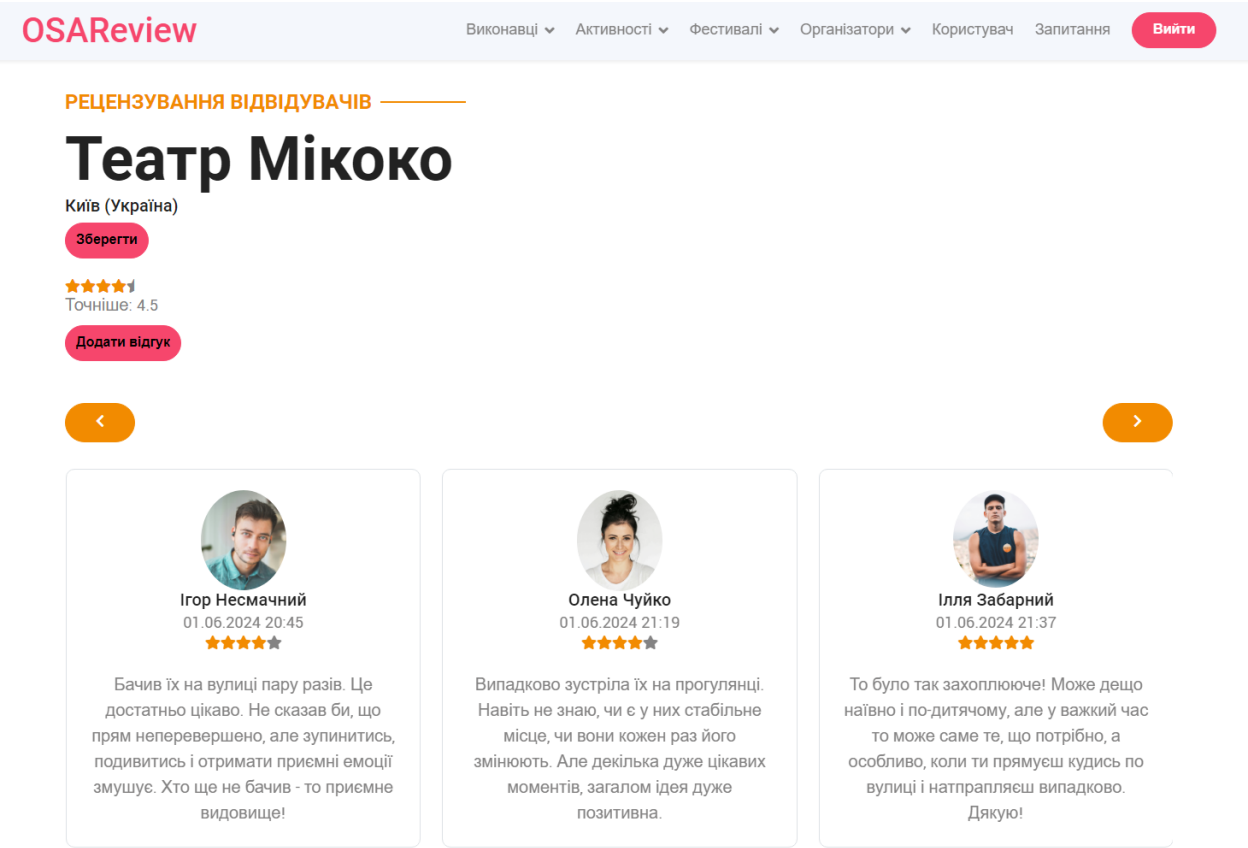


Рисунок 4.2 – Рецензування групи виконавців вуличних активностей для авторизованого користувача

Описана вище в меню функція відтворення переліку тих виконавців, за яких є відповідальним користувач, реалізує роботу через окрему сторінку (рис. 4.3). Для кожної такої групи доступні кнопки для збереження або відкидання зі збережених групи, що є стандартними, кнопки для переходу до

переліків поставлених за групою запитань, звідки можна давати відповідь за своєю групою виконавців вуличних активностей.

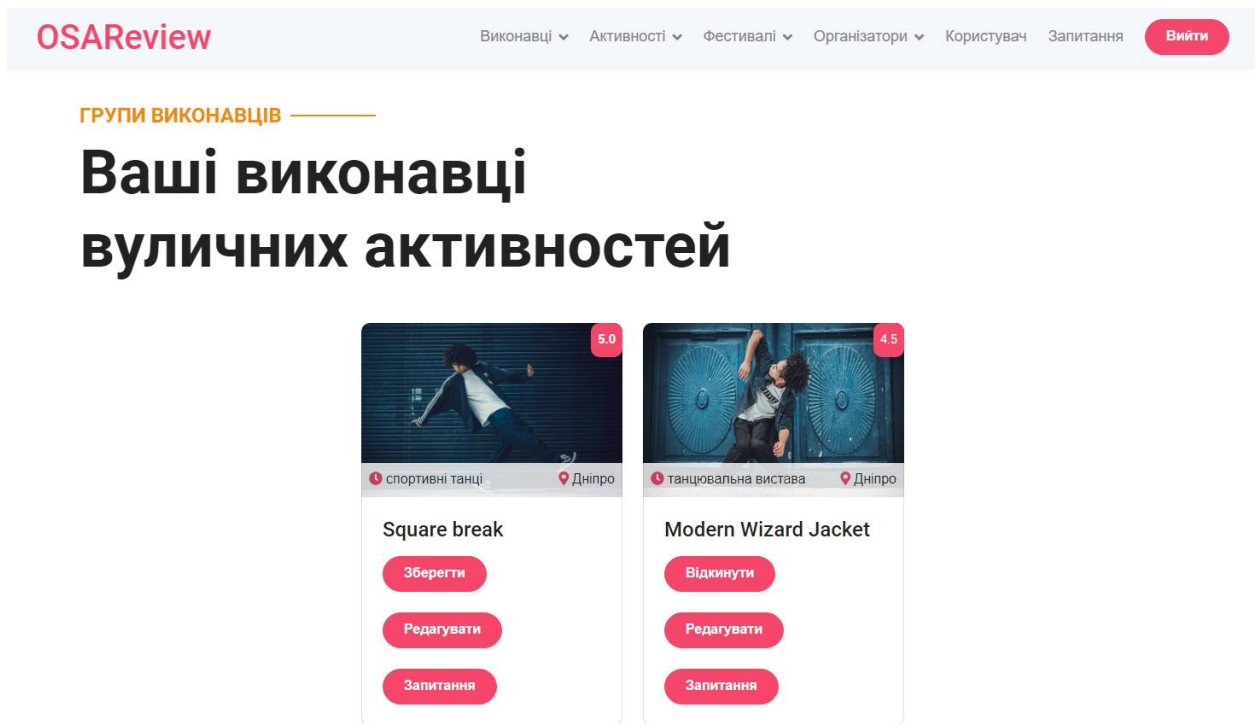


Рисунок 4.3 – Підтримувані авторизованим користувачем виконавці вуличних активностей

Розділ меню активностей містить функції, які пов'язані з демонстрацією графіків вуличних активностей. Усі ці засоби доступні в повній мірі навіть неавторизованому користувачу. Вони включають графіки вуличних активностей за містом, вулицею або групою і дозволяють спочатку через пошук (введенням назви та підбором з випадючого переліку) задати відповідно місто, вулицю або групу виконавців вуличних активностей, далі задати часовий проміжок, якщо це потрібно, і далі побачити, коли саме мають відбуватися виступи.

Розділ меню з фестивалями містить для неавторизованого користувача тільки пункт для пошуку фестивалів, а для авторизованого – додатково пункт для відтворення збережених фестивалів вуличних активностей. Якщо користувач авторизувався, але ще й представляє організатора вуличних

активностей, то він може за допомогою відповідного пункту меню створювати фестивалі, вносячи про них дані.

Зі сторінки пошуку фестивалів можна перейти до сторінки конкретного фестивалю (рис. 4.4). У верхній частині на сторінці містяться фотографії, далі базова інформація, а ще нижче знаходиться перелік груп виконавців вуличних активностей, які на фестивалі виступають (рис. 4.5). Цією інформацією все не обмежується, також відображаються 3 останні запитання за фестивалем з можливістю відкрити всі, відгуки про фестиваль.

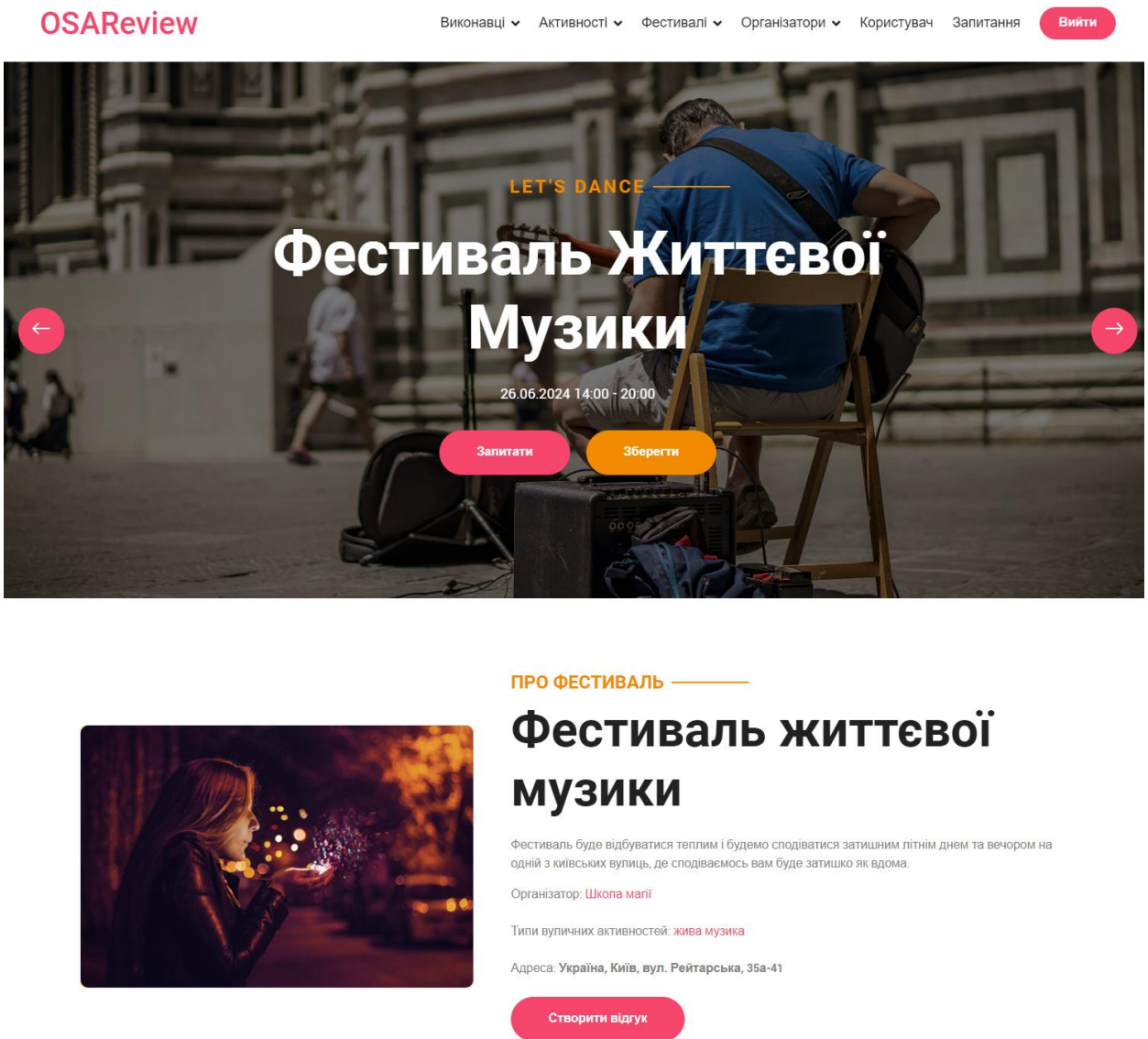


Рисунок 4.4 – Відтворення інформації про фестиваль вуличних активностей (верхня частина сторінки)

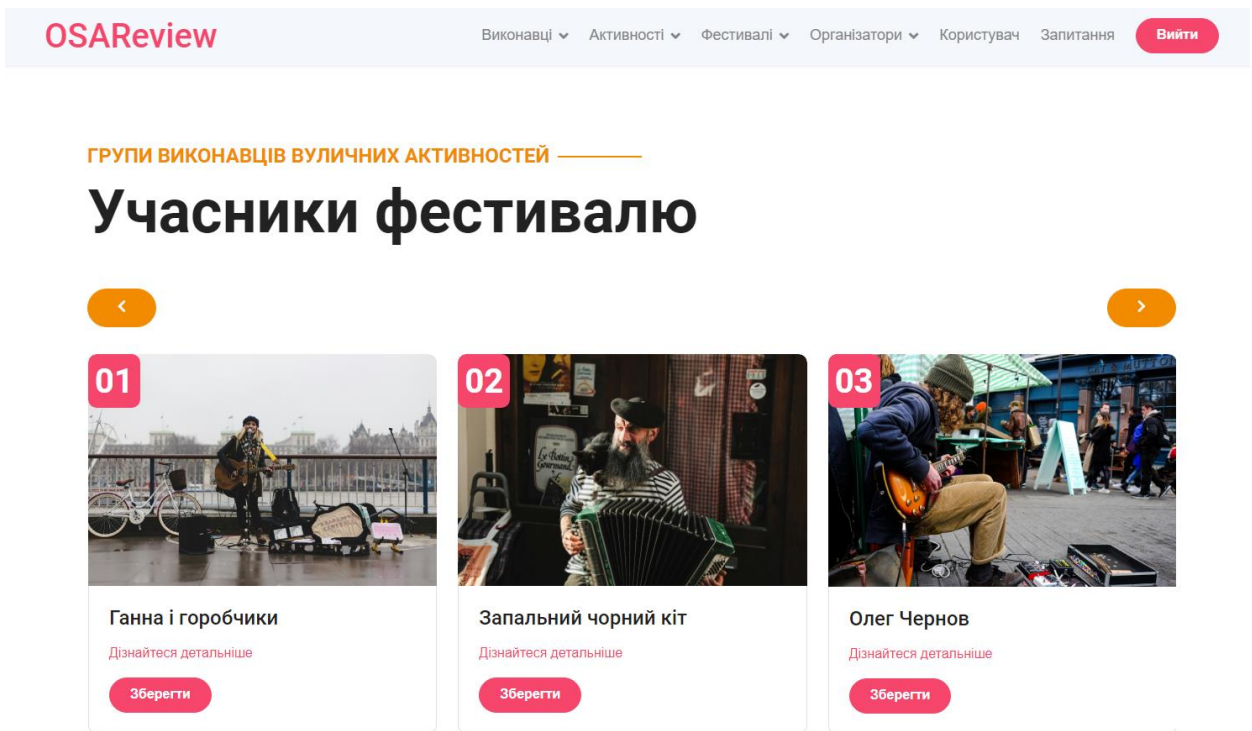


Рисунок 4.5 – Відтворення інформації про фестиваль вуличних активностей
(середня частина сторінки)

Розділ меню з організаторами містить засоби пошуку організаторів вуличних активностей для всіх, починаючи від неавторизованих користувачів, відтворення збережених організаторів та власних організаторів, що представляються користувачем, для авторизованих користувачів, а також функцію внесення організаторів до програми для модератора.

4.4 Висновки за розділом 4

Описано логіку виконання програми, що слідує для користувача, який звертається до вебдодатку з клієнтської сторони. На основі даної логіки було виконано звернення до сервера та протестовано роботу кожної функціональної можливості, що була визначена для виконання. Усі вони в підсумку були успішно протестовані.

ВИСНОВКИ

Роботу присвячено розгляду процесу виступу виконавців у формі вуличних активностей. Для підтримки даного процесу було запропоновано розробити програмне забезпечення рецензування вуличних активностей для підвищення інформованості користувачів, що було визначено в якості мети роботи, яку було в результаті досягнуто.

Вебдодаток розроблено мовою Python за допомогою Python-фреймворку Django та системи керування БД MySQL.

Розроблена програма надає можливість отримувати інформацію про вуличні активності, які проводяться у місті, час таких виступів, про фестивалі, в межах яких такі вуличні активності відбуваються, рецензії від інших користувачів, які стали свідками таких активностей, та створювати власні рецензії. Створення програми як вебдодатку сприяє широті доступу користувачів з різних пристроїв, з різних локацій, що дозволяє охоплювати ширшу аудиторію, яка може стати відвідувачами фестивалів вуличних активностей, глядачами виступів вуличних артистів, долучатися до розвитку вуличних активностей як бізнесу, докладатися до розвитку вулиць міста, на яких такі активності відбуваються.

Усі функціональні можливості успішно реалізовані, що було підтверджено тестуванням. У програмі реалізовано надання інформації про організаторів вуличних активностей, фестивалі, групи виконавців вуличних активностей, пошук таких даних, внесення відповідних даних та редагування як звичайними користувачами, так і самими виконавцями, які можуть в подальшому подавати заявки щодо представництва, рецензування фестивалів вуличних активностей та виступів груп виконавців вуличних активностей, визначення графіку вуличних активностей у місті загалом, за вулицею міста, за групою, особисте збереження фестивалю вуличних активностей, групи виконавців та організатора вуличних активностей, створення запитань і відповідей на запитання про групи, фестивалі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Open Streets Day [Electronic resource]. – Access mode : <https://open-streets-day.nowwemove.com/>.
2. What is Django? [Electronic resource]. – Access mode : <https://www.ibm.com/topics/django>.
3. Dinder, M. Becoming an Enterprise Django Developer [Text] : Discover best practices, tooling, and solutions for writing and organizing Django applications in production / M. Dinder. – Birmingham : Packt Publishing, 2022. – 526 p.
4. Technical guides – Documentation – WordPress.org [Electronic resource]. – Access mode : <https://wordpress.org/documentation/technical-guides/>.
5. What is WordPress? Explained for beginners [Electronic resource]. – Access mode : <https://kinsta.com/knowledgebase/what-is-wordpress/>.
6. Shah, A. Top Advantages of MySQL for Developers [Electronic resource] / A. Shah. – Access mode : <https://www.linkedin.com/pulse/top-advantages-mysql-developers-aakash-shah-3ihxc>.
7. JustDance – Dance Class Website Template [Electronic resource]. – Access mode : <https://htmlcodex.com/dance-class-website-template/>.

ДОДАТОК А
Технічне завдання

Вступ

Вуличні активності є засобом розваг, який тісно інтегрований з життям вулиці, допомагаючи в підсумку розвивати вулицю, привертати до неї увагу, а одночасно за рахунок розвитку, створення певної історії вулиці, привертати додаткову увагу до самих вуличних активностей, до виконавців цих активностей. Прикладами вуличних активностей є виступи музикантів, при чому і індивідуальні, і в групах, театралізовані виступи, інші типи виступів вуличних активностей, але окрім того це може бути і проведення фестивалів, у яких беруть участь продавців різних товарів, зокрема власноруч виготовлених, суміщення такого продажу з певними виступами. Розповсюдження інформації про такі події, які можуть відбуватися регулярно або одноразово, є важливим, адже це не концерти, на які продаються квитки, створюються афіші. Зазвичай інформація про такі вуличні активності розповсюджується від людей до людей. Тож створення вебдодатку має сприяти розвитку цієї сфери.

А.1 Підстави для розробки

Підстави розробки за дипломною кваліфікаційною роботою визначені та затверджені наказом № 168 від 24 квітня 2024 р. за Національним університетом «Запорізька політехніка», у якому зафіксовано тему «Програмне забезпечення рецензування вуличних активностей».

А.2 Призначення розробки

Програма повинна надати можливість отримувати інформацію про вуличні активності, які проводяться у місті, час таких виступів, рецензії від інших користувачів, які стали свідками таких активностей.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Підтримка роботи користувача з програмою має забезпечуватися наступними функціями:

- відтворення сторінки організатора вуличних активностей;
- внесення даних організатора вуличних активностей;
- редагування даних організатора вуличних активностей;
- пошук організаторів вуличних активностей;
- додавання користувача до підтримки організатора вуличних активностей;
- скасування участі користувача в підтримці організатора вуличних активностей;
- деактивація участі користувача в підтримці організатора вуличних активностей;
- відтворення переліку організаторів вуличних активностей, які підтримує користувач;
- внесення даних фестивалю вуличних активностей;
- редагування даних фестивалю вуличних активностей;
- відтворення переліку фестивалів вуличних активностей організатора;
- відтворення сторінки фестивалю вуличних активностей;
- внесення даних групи виконавців вуличних активностей, включаючи внесення типів вуличних активностей групи;
- пошук фестивалів вуличних активностей;
- відтворення сторінки групи виконавців вуличних активностей;
- редагування даних групи виконавців вуличних активностей;
- пошук груп виконавців вуличних активностей;
- подання заявки на відповідальність за групу виконавців вуличних активностей;

- підтримання або скасування заявки на відповідальність за групу виконавців вуличних активностей;
- додавання відповідального за групу виконавців вуличних активностей;
- вилучення відповідального за групу виконавців вуличних активностей;
- відтворення переліку груп виконавців вуличних активностей, за які відповідальний користувач;
- рецензування фестивалю вуличних активностей;
- відтворення переліку рецензій фестивалю вуличних активностей;
- відтворення переліку рецензій користувача;
- авторизація;
- реєстрація;
- відтворення сторінки користувача;
- рецензування групи виконавців вуличних активностей;
- відтворення переліку рецензій групи виконавців вуличних активностей;
- редагування даних користувача;
- відтворення графіку вуличних активностей у місті;
- відтворення графіку вуличних активностей за вулицею міста;
- відтворення графіку вуличних активностей групи;
- збереження фестивалю вуличних активностей або видалення збереження;
- збереження групи виконавців вуличних активностей або видалення збереження;
- збереження організатора вуличних активностей або видалення збереження;
- створення запитання до фестивалю вуличних активностей;
- відтворення запитань до фестивалю вуличних активностей;
- відтворення власних запитань;

- створення відповіді на запитання до фестивалю вуличних активностей;
- створення запитання до групи виконавців вуличних активностей;
- відтворення запитань до групи виконавців вуличних активностей;
- створення відповіді на запитання до групи виконавців вуличних активностей;
- відтворення переліку заявок на відповідальність за групу виконавців вуличних активностей.

A.3.2 Умови експлуатації

Експлуатація програми операторами в різних ролях не потребує особливих навичок окрім навичок роботи з вебдодатками, але налаштування серверу має враховувати навички у адміністратора для роботи з системою MySQL для забезпечення бази даних, керування Python-фреймворком веброзробки Django (зокрема і роботи з адміністративною панеллю), інтерпретатором Python.

A.3.3 Вимоги до складу та параметрів технічних засобів

Технічні засоби визначені на мінімальному рівні, охоплюючи наступні вимоги:

- процесор не менше чотирьохядерного з частотою 2,0 ГГц;
- оперативна пам'ять – 4 Гб;
- жорсткий диск – 20 Гб простору.

A.4 Порядок контролю та приймання

Строки виконання та результати виконання за кожним етапом дипломної кваліфікаційної роботи визначаються календарним планом у

завданні на роботу та контролюються керівником роботи, який закріплений наказом.

Захист роботи реалізується перед екзаменаційною комісією як відкрита процедура.

ДОДАТОК Б
Текст програми

```

from .models import Stractivityorganizer
from django.shortcuts import render
from .models import Appuser
from .models import Stractivityfestival
from .models import Stractivityfestivalphoto

def street_activity_organizer_view(request,
id_stractivityorganizer):
    street_activity_organizer_context = {}
    festivals = None
    stractivity_organizer =
Stractivityorganizer.objects.select_related("main_city").filter(
id = id_stractivityorganizer)
    if stractivity_organizer:
        stractivity_organizer = stractivity_organizer[0]
        street_activity_organizer_context["organizer"] =
stractivity_organizer
        festivals =
Stractivityfestival.objects.select_related("citystreet")
        festivals =
festivals.select_related("citystreet__city")
        festivals =
festivals.select_related("main_stractivitytype")
        festivals = festivals.filter(festival_organizer =
stractivity_organizer)
        festivals = festivals.order_by("-festival_start")
        if festivals:
            if festivals.count() > 5:
                festivals = festivals[:5]
            for stractivity_festival in festivals:
                stractivity_festival.photo =
Stractivityfestivalphoto.objects.filter(stractivityfestival =
stractivity_festival)
                stractivity_festival.photo =
stractivity_festival.photo[0]
            if request.user.is_active:
                if stractivity_organizer:
                    organizer_appusers =
Stractivityorganizerappuser.objects.filter(stractivityorganizer
= stractivity_organizer)
                    organizer_appusers_active =
organizer_appusers.filter(isactive = True)
                    appuser =
Appuser.objects.prefetch_related("stractivityfestivals")

```

```

        appuser =
appuser.prefetch_related("stractivityorganizers")
        appuser = appuser.filter(django_account =
request.user)
        if appuser:
            appuser = appuser[0]
            if appuser in organizer_appusers_active:

street_activity_organizer_context["appusers"] =
organizer_appusers
            else:

street_activity_organizer_context["appusers"] =
organizer_appusers_active
            if stractivity_organizer in
appuser.stractivityorganizers:
                stractivity_organizer.save = True
            else:
                stractivity_organizer.save = False

street_activity_organizer_context["appuser"] = appuser
            if festivals:
                for stractivity_festival in festivals:
                    if stractivity_festival in
appuser.stractivityfestivals:
                        stractivity_festival.save =
True
                    else:
                        stractivity_festival.save =
False
            street_activity_organizer_context["festival"] =
festivals
            return render(request,
"streetactivityprog/saorganizer.html",
street_activity_organizer_context)

def street_activity_organizer_festivals_view(request,
id_stractivityorganizer):
    street_activity_organizer_context = {}
    festivals = None
    stractivity_organizer =
Stractivityorganizer.objects.select_related("main_city").filter(
id = id_stractivityorganizer)
    if stractivity_organizer:

```

```

        stractivity_organizer = stractivity_organizer[0]
        street_activity_organizer_context["organizer"] =
stractivity_organizer
        festivals =
Stractivityfestival.objects.select_related("citystreet")
        festivals =
festivals.select_related("citystreet__city")
        festivals =
festivals.select_related("main_stractivitytype")
        festivals = festivals.filter(festival_organizer =
stractivity_organizer)
        festivals = festivals.order_by("-festival_start")
        if festivals:
            for stractivity_festival in festivals:
                stractivity_festival.photo =
Stractivityfestivalphoto.objects.filter(stractivityfestival =
stractivity_festival)
                stractivity_festival.photo =
stractivity_festival.photo[0]
            if request.user.is_active:
                if stractivity_organizer:
                    appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                    appuser =
appuser.prefetch_related("stractivityorganizers")
                    appuser = appuser.filter(django_account =
request.user)
                    if appuser:
                        appuser = appuser[0]
                        if stractivity_organizer in
appuser.stractivityorganizers:
                            stractivity_organizer.save = True
                        else:
                            stractivity_organizer.save = False

street_activity_organizer_context["appuser"] = appuser
        if festivals:
            for stractivity_festival in festivals:
                if stractivity_festival in
appuser.stractivityfestivals:
                    stractivity_festival.save =
True
                else:

```

```

stractivity_festival.save =
False
        street_activity_organizer_context["festival"] =
festivals
        return render(request,
"streetactivityprog/saorgfestivals.html",
street_activity_organizer_context)

        from .models import Stractivityfestivalreview
        from .models import Stractivityfestivalquecomment
        from .models import Stractivitybusinessphoto

        def street_activity_festival_view(request,
id_stractivityfestival):
            street_activity_festival_context = {}
            stractivity_festival =
Stractivityfestival.objects.select_related("festival_organizer")
            stractivity_festival =
stractivity_festival.select_related("festival_organizer__main_ci
ty")
            stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses")
            stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses__st
ractivitytypes")
            stractivity_festival = stractivity_festival.filter(id =
id_stractivityfestival)
            if stractivity_festival:
                stractivity_festival = stractivity_festival[0]
                street_activity_festival_context["festival"] =
stractivity_festival
                festival_photos =
Stractivityfestivalphoto.objects.filter(stractivityfestival =
stractivity_festival)
                street_activity_festival_context["photos"] =
festival_photos
                festival_reviews =
Stractivityfestivalreview.objects.select_related("appuser")
                festival_reviews = festival_reviews.order_by("-
review_time")
                street_activity_festival_context["reviews"] =
festival_reviews[:5]

```

```

        festival_questions =
Strativityfestivalque.objects.select_related("question_appuser"
)
        festival_questions =
festival_questions.filter(strativityfestival =
strativity_festival)
        festival_questions = festival_questions[:3]
        festival_questions = festival_questions.order_by("-
question_time")
        for festival_question in festival_questions:
            festival_comments =
Strativityfestivalquecomment.objects.select_related("comment_ap
puser")
            festival_comments =
festival_comments.filter(strativityfestivalque =
festival_question)
            festival_comments =
festival_comments.order_by("-comment_time")
            festival_question.comments = festival_comments
            street_activity_festival_context["questions"] =
festival_questions
            if request.user.is_active:
                if strativity_festival:
                    appuser =
Appuser.objects.prefetch_related("strativityfestivals")
                    appuser =
appuser.prefetch_related("strativityorganizers")
                    appuser =
appuser.prefetch_related("strativitybusinesses")
                    appuser = appuser.filter(django_account =
request.user)
                    if appuser:
                        appuser = appuser[0]
                        festival_organizer =
strativity_festival.festival_organizer
                        appusers =
Strativityorganizerappuser.objects.filter(strativityorganizer
= festival_organizer, appuser = appuser, isactive = True)
                        if appusers:

street_activity_festival_context["organizer"] = True
                        else:

street_activity_festival_context["organizer"] = False

```

```

        if stractivity_festival in
appuser.stractivityfestivals:

street_activity_festival_context["save"] = True
        else:

street_activity_festival_context["save"] = False
        festival_businesses =
stractivity_festival.stractivitybusinesses
        for stractivity_business in
festival_businesses:
            if stractivity_business in
appuser.stractivitybusinesses:
                stractivity_business.save = True
            else:
                stractivity_business.save = False
                stractivity_business.photo =
Stractivitybusinessphoto.objects.filter(stractivitybusiness =
stractivity_business)

street_activity_festival_context["businesses"] =
festival_businesses
        return render(request,
"streetactivityprog/safestival.html",
street_activity_festival_context)


from .models import Stractivitybusinessreview
from .models import Stractivitybusinessque
from .models import Stractivitybusinessquecomment
from .models import Stractivitybusinessstreet
from .models import Stractivitybusinessstreettime
from .models import Stractivitybusinessapplication


def street_activity_business_view(request,
id_stractivitybusiness):
    street_activity_business_context = {}
    stractivity_business =
Stractivitybusiness.objects.select_related("appuseradded")
    stractivity_business =
stractivity_business.prefetch_related("stractivitytypes")
    stractivity_business =
stractivity_business.prefetch_related("appusers")
    stractivity_business = stractivity_business.filter(id =
id_stractivitybusiness)

```

```

        if stractivity_business:
            stractivity_business = stractivity_business[0]
            street_activity_business_context["business"] =
stractivity_business
            business_photos =
Stractivitybusinessphoto.objects.filter(stractivitybusiness =
stractivity_business)
            street_activity_business_context["photos"] =
business_photos
            business_reviews =
Stractivitybusinessreview.objects.select_related("appuser")
            business_reviews = business_reviews.order_by("-
review_time")
            street_activity_business_context["reviews"] =
business_reviews[:5]
            business_questions =
Stractivitybusinessque.objects.select_related("question_appuser"
)
            business_questions =
business_questions.filter(stractivitybusiness =
stractivity_business)
            business_questions = business_questions.order_by("-
question_time")
            business_questions = business_questions[:3]
            for business_question in business_questions:
                business_comments =
Stractivitybusinessquecomment.objects.select_related("comment_ap
puser")
                business_comments =
business_comments.filter(stractivitybusinessque =
business_question)
                business_comments =
business_comments.order_by("-comment_time")
                business_question.comments = business_comments
                street_activity_business_context["questions"] =
business_questions
            if request.user.is_active:
                if stractivity_business:
                    appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                    appuser =
appuser.prefetch_related("stractivityorganizers")
                    appuser =
appuser.prefetch_related("stractivitybusinesses")

```

```

        appuser = appuser.filter(django_account =
request.user)
        if appuser:
            appuser = appuser[0]
            if appuser in
stractivity_business.appusers:

street_activity_business_context["member"] = True
            else:

street_activity_business_context["member"] = False
                applications =
Stractivitybusinessapplication.objects.select_related("appuser")
                applications =
applications.filter(stractivitybusiness = stractivity_business)
                if appuser in applications:

street_activity_business_context["pretender"] = True
                    else:

street_activity_business_context["pretender"] = False
                        festivals =
Stractivityfestival.objects.prefetch_related("stractivitybusines
ses")
                            festivals =
festivals.select_related("citystreet")
                            festivals =
festivals.select_related("citystreet__city")
                            festivals =
festivals.select_related("main_stractivitytype")
                            festivals =
festivals.select_related("festival_organizer")
                            festivals =
festivals.filter(stractivitybusinesses__in =
[stractivity_business])
                                for festival in festivals:
                                    if festival in
appuser.stractivityfestivals:
                                        festival.save = True
                                    else:
                                        festival.save = False

street_activity_business_context["festivals"] = festivals

```

```

        if stractivity_business in
appuser.stractivitybusinesses:

street_activity_business_context["save"] = True
        else:

street_activity_business_context["save"] = False
        businesses_addresses =
Stractivitybusinessstreet.objects.filter(stractivitybusiness =
stractivity_business)
        for businesses_address in
businesses_addresses:
            businesses_address.times =
Stractivitybusinessstreettime.objects.filter(stractivitybusiness
street = businesses_address)

street_activity_business_context["addresses"] =
businesses_addresses
        return render(request,
"streetactivityprog/sabusiness.html",
street_activity_business_context)

def street_activity_business_reviews_view(request,
id_stractivitybusiness):
    street_activity_business_context = {}
    stractivity_business =
Stractivitybusiness.objects.select_related("appuseradded")
    stractivity_business =
stractivity_business.prefetch_related("stractivitytypes")
    stractivity_business =
stractivity_business.prefetch_related("appusers")
    stractivity_business = stractivity_business.filter(id =
id_stractivitybusiness)
    if stractivity_business:
        stractivity_business = stractivity_business[0]
        street_activity_business_context["business"] =
stractivity_business
        business_photos =
Stractivitybusinessphoto.objects.filter(stractivitybusiness =
stractivity_business)
        street_activity_business_context["photos"] =
business_photos[:3]
        business_reviews =
Stractivitybusinessreview.objects.select_related("appuser")

```

```

        business_reviews = business_reviews.order_by("-
review_time")
        street_activity_business_context["reviews"] =
business_reviews
        if request.user.is_active:
            if stractivity_business:
                appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                appuser =
appuser.prefetch_related("stractivityorganizers")
                appuser =
appuser.prefetch_related("stractivitybusinesses")
                appuser = appuser.filter(django_account =
request.user)
                if appuser:
                    appuser = appuser[0]
                    if appuser in
stractivity_business.appusers:

street_activity_business_context["member"] = True
                    else:

street_activity_business_context["member"] = False
                        applications =
Stractivitybusinessapplication.objects.select_related("appuser")
                        applications =
applications.filter(stractivitybusiness = stractivity_business)
                        if appuser in applications:

street_activity_business_context["pretender"] = True
                            else:

street_activity_business_context["pretender"] = False
                                appuser_review =
Stractivitybusinessreview.objects.filter(stractivitybusiness =
stractivity_business, appuser = appuser)
                                if appuser_review:

street_activity_business_context["userreview"] = appuser_review
                                    if stractivity_business in
appuser.stractivitybusinesses:

street_activity_business_context["save"] = True
                                        else:

```

```

street_activity_business_context["save"] = False
    return render(request,
"streetactivityprog/sabusinessreviews.html",
street_activity_business_context)

    def street_activity_festival_reviews_view(request,
id_stractivityfestival):
        street_activity_festival_context = {}
        stractivity_festival =
Stractivityfestival.objects.select_related("festival_organizer")
        stractivity_festival =
stractivity_festival.select_related("festival_organizer__main_ci
ty")

        stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses")
        stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses__st
ractivitytypes")
        stractivity_festival = stractivity_festival.filter(id =
id_stractivityfestival)
        if stractivity_festival:
            stractivity_festival = stractivity_festival[0]
            street_activity_festival_context["festival"] =
stractivity_festival
            festival_photos =
Stractivityfestivalphoto.objects.filter(stractivityfestival =
stractivity_festival)
            street_activity_festival_context["photos"] =
festival_photos[:3]
            festival_reviews =
Stractivityfestivalreview.objects.select_related("appuser")
            festival_reviews = festival_reviews.order_by("-
review_time")
            street_activity_festival_context["reviews"] =
festival_reviews
            if request.user.is_active:
                if stractivity_festival:
                    appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                    appuser =
appuser.prefetch_related("stractivityorganizers")
                    appuser =
appuser.prefetch_related("stractivitybusinesses")

```

```

        appuser = appuser.filter(django_account =
request.user)
        if appuser:
            appuser = appuser[0]
            festival_organizer =
stractivity_festival.festival_organizer
            appusers =
Stractivityorganizerappuser.objects.filter(stractivityorganizer
= festival_organizer, appuser = appuser, isactive = True)
            if appusers:

street_activity_festival_context["organizer"] = True
                else:

street_activity_festival_context["organizer"] = False
                    if stractivity_festival in
appuser.stractivityfestivals:

street_activity_festival_context["save"] = True
                        else:

street_activity_festival_context["save"] = False
                            return render(request,
"streetactivityprog/safestreviews.html",
street_activity_festival_context)

    def street_activity_appuser_reviews_view(request):
        street_activity_reviews_context = {}
        if request.user.is_active:
            appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
            appuser =
appuser.prefetch_related("stractivityorganizers")
            appuser =
appuser.prefetch_related("stractivitybusinesses")
            appuser = appuser.filter(django_account =
request.user)
            if appuser:
                appuser = appuser[0]
                street_activity_reviews_context["appuser"] =
appuser

                festival_reviews =
Stractivityfestivalreview.objects.select_related("stractivityfes
tival")

```

```

        festival_reviews =
festival_reviews.select_related("stractivityfestival__festival_o
rganizer")
        festival_reviews =
festival_reviews.filter(appuser = appuser)
        festival_reviews = festival_reviews.order_by("-
review_time")
        for festival_review in festival_reviews:
            stractivity_festival =
festival_review.stractivityfestival
            if stractivity_festival in
appuser.stractivityfestivals:
                festival_review.save = True
            else:
                festival_review.save = False
            festival_organizer =
stractivity_festival.festival_organizer
            appusers =
Stractivityorganizerappuser.objects.filter(stractivityorganizer
= festival_organizer, appuser = appuser, isactive = True)
            if appusers:
                festival_review.org = True
            else:
                festival_review.org = False

street_activity_reviews_context["festival_reviews"] =
festival_reviews
        business_reviews =
Stractivitybusinessreview.objects.select_related("stractivitybus
iness")
        business_reviews =
business_reviews.prefetch_related("stractivitybusiness__appusers
")
        business_reviews =
business_reviews.filter(appuser = appuser)
        business_reviews = business_reviews.order_by("-
review_time")
        for business_review in business_reviews:
            if appuser in
business_review.stractivitybusiness.appusers:
                business_review.member = True
            else:
                business_review.member = False

```

```

        applications =
Stractivitybusinessapplication.objects.select_related("appuser")
        applications =
applications.filter(stractivitybusiness =
business_review.stractivitybusiness)
        if appuser in applications:
            business_review.pretender = True
        else:
            business_review.pretender = False

street_activity_reviews_context["business_reviews"] =
business_reviews
        return render(request,
"streetactivityprog/sauserreviews.html",
street_activity_reviews_context)

def street_activity_festival_questions_view(request,
id_stractivityfestival):
    street_activity_festival_context = {}
    stractivity_festival =
Stractivityfestival.objects.select_related("festival_organizer")
    stractivity_festival =
stractivity_festival.select_related("festival_organizer__main_ci
ty")
    stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses")
    stractivity_festival =
stractivity_festival.prefetch_related("stractivitybusinesses__st
ractivitytypes")
    stractivity_festival = stractivity_festival.filter(id =
id_stractivityfestival)
    if stractivity_festival:
        stractivity_festival = stractivity_festival[0]
        street_activity_festival_context["festival"] =
stractivity_festival
        festival_photos =
Stractivityfestivalphoto.objects.filter(stractivityfestival =
stractivity_festival)
        street_activity_festival_context["photos"] =
festival_photos[:3]
        festival_questions =
Stractivityfestivalque.objects.select_related("question_appuser"
)

```

```

        festival_questions =
festival_questions.filter(stractivityfestival =
stractivity_festival)
        festival_questions = festival_questions
        festival_questions = festival_questions.order_by("-
question_time")
        for festival_question in festival_questions:
            festival_comments =
Stractivityfestivalquecomment.objects.select_related("comment_ap
puser")
            festival_comments =
festival_comments.filter(stractivityfestivalque =
festival_question)
            festival_comments =
festival_comments.order_by("-comment_time")
            festival_question.comments = festival_comments
            street_activity_festival_context["questions"] =
festival_questions
            if request.user.is_active:
                if stractivity_festival:
                    appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                    appuser =
appuser.prefetch_related("stractivityorganizers")
                    appuser =
appuser.prefetch_related("stractivitybusinesses")
                    appuser = appuser.filter(django_account =
request.user)
                    if appuser:
                        appuser = appuser[0]
                        festival_organizer =
stractivity_festival.festival_organizer
                        appusers =
Stractivityorganizerappuser.objects.filter(stractivityorganizer
= festival_organizer, appuser = appuser, isactive = True)
                        if appusers:

street_activity_festival_context["organizer"] = True
                            else:

street_activity_festival_context["organizer"] = False
                                if stractivity_festival in
appuser.stractivityfestivals:

```

```

street_activity_festival_context["save"] = True
    else:

street_activity_festival_context["save"] = False
    return render(request,
"streetactivityprog/safestquestions.html",
street_activity_festival_context)

    def street_activity_business_questions_view(request,
id_stractivitybusiness):
        street_activity_business_context = {}
        stractivity_business =
Stractivitybusiness.objects.select_related("appuseradded")
        stractivity_business =
stractivity_business.prefetch_related("stractivitytypes")
        stractivity_business =
stractivity_business.prefetch_related("appusers")
        stractivity_business = stractivity_business.filter(id =
id_stractivitybusiness)
        if stractivity_business:
            stractivity_business = stractivity_business[0]
            street_activity_business_context["business"] =
stractivity_business
            business_photos =
Stractivitybusinessphoto.objects.filter(stractivitybusiness =
stractivity_business)
            street_activity_business_context["photos"] =
business_photos[:3]
            business_questions =
Stractivitybusinessque.objects.select_related("question_appuser"
)
            business_questions =
business_questions.filter(stractivitybusiness =
stractivity_business)
            business_questions = business_questions.order_by("-
question_time")
            business_questions = business_questions
            for business_question in business_questions:
                business_comments =
Stractivitybusinessquecomment.objects.select_related("comment_ap
puser")

```

```

        business_comments =
business_comments.filter(stractivitybusinessque =
business_question)
        business_comments =
business_comments.order_by("-comment_time")
        business_question.comments = business_comments
        street_activity_business_context["questions"] =
business_questions
        if request.user.is_active:
            if stractivity_business:
                appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
                appuser =
appuser.prefetch_related("stractivityorganizers")
                appuser =
appuser.prefetch_related("stractivitybusinesses")
                appuser = appuser.filter(django_account =
request.user)
                if appuser:
                    appuser = appuser[0]
                    if appuser in
street_activity_business.appusers:

street_activity_business_context["member"] = True
                    else:

street_activity_business_context["member"] = False
                    applications =
Stractivitybusinessapplication.objects.select_related("appuser")
                    applications =
applications.filter(stractivitybusiness = street_activity_business)
                    if appuser in applications:

street_activity_business_context["pretender"] = True
                    else:

street_activity_business_context["pretender"] = False
                    return render(request,
"streetactivityprog/sabusquestions.html",
street_activity_business_context)

def street_activity_appuser_questions_view(request):
    street_activity_business_context = {}
    if request.user.is_active:

```

```

        appuser =
Appuser.objects.prefetch_related("stractivityfestivals")
        appuser =
appuser.prefetch_related("stractivityorganizers")
        appuser =
appuser.prefetch_related("stractivitybusinesses")
        appuser = appuser.filter(django_account =
request.user)
        if appuser:
            appuser = appuser[0]
            appuser_festival_questions =
Stractivityfestivalque.objects.select_related("stractivityfestiv
al")
            appuser_festival_questions =
appuser_festival_questions.filter(question_appuser = appuser)
            appuser_festival_questions =
appuser_festival_questions.order_by("-question_time")
            for festival_question in
appuser_festival_questions:
                festival_comments =
Stractivityfestivalquecomment.objects.select_related("comment_ap
puser")
                festival_comments =
festival_comments.filter(stractivityfestivalque =
festival_question)
                festival_comments =
festival_comments.order_by("-comment_time")
                festival_question.comments =
festival_comments
                street_activity_business_context
["festival_questions"] = appuser_festival_questions
                appuser_business_questions =
Stractivitybusinessque.objects.select_related("stractivitybusine
ss")
                appuser_business_questions =
appuser_business_questions.filter(question_appuser = appuser)
                appuser_business_questions =
appuser_business_questions.order_by("-question_time")
                for business_question in
appuser_business_questions:
                    business_comments =
Stractivitybusinessquecomment.objects.select_related("comment_ap
puser")

```

```

        business_comments =
business_comments.filter(stractivitybusinessque =
business_question)
        business_comments =
business_comments.order_by("-comment_time")
        business_question.comments =
business_comments
        street_activity_business_context
["business_questions"] = appuser_business_questions
        return render(request,
"streetactivityprog/saappuserquestions.html",
street_activity_business_context)

```

ДОДАТОК В
Слайди презентації

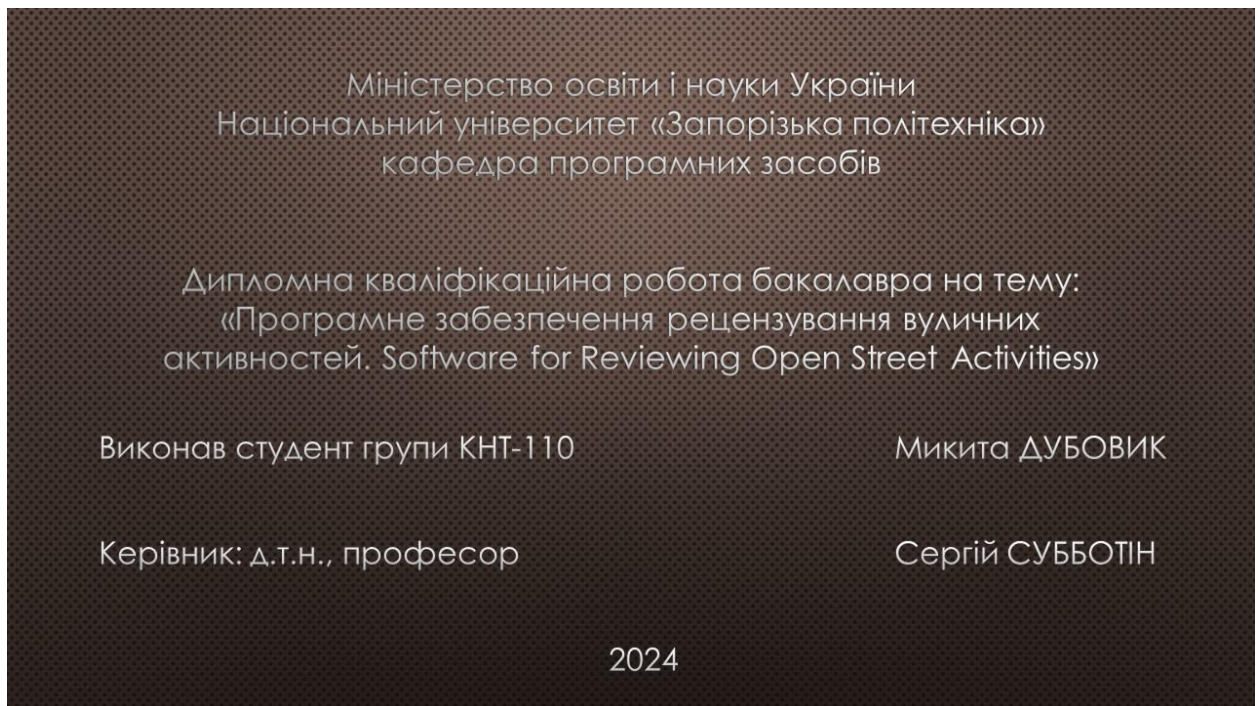


Рисунок В.1 – Початковий слайд

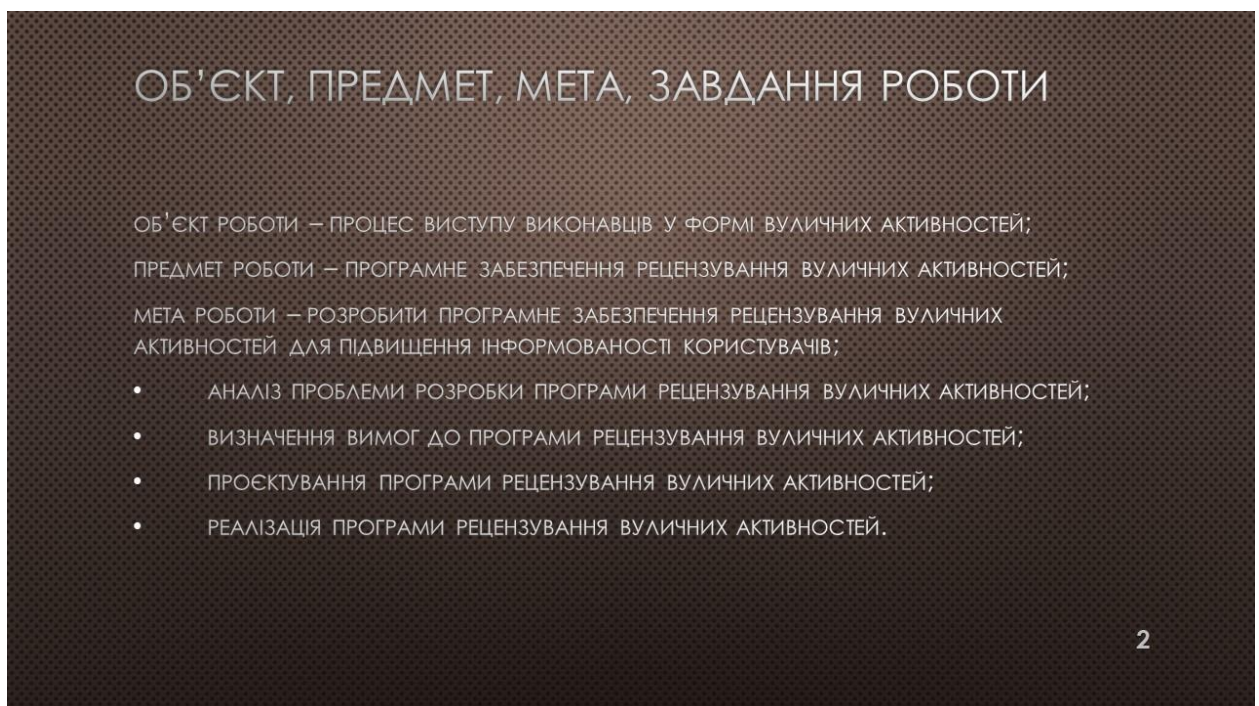


Рисунок В.2 – Об'єкт, предмет, мета, завдання роботи

ПОРІВНЯННЯ РОЗРОБКИ З ПРОТОТИПОМ		
Характеристика	Open Streets Day	Розробка
Пошук вуличних активностей	+	+
Представлення інформації про фестивалі вуличних активностей	+	+
Реєстрація власного фестивалю вуличних активностей	+	+
Реєстрація окремих виконавців вуличних активностей або їх групи	—	+
Представлення інформації про окремих виконавців вуличних активностей або їх групи	—	+
Створення запитань за фестивалями або окремими виконавцями	—	+
Рецензування фестивалів вуличних активностей, окремих виступів	—	+

3

Рисунок В.3 – Порівняння розробки з прототипом

ПОРІВНЯННЯ РІШЕНЬ ДЛЯ СТВОРЕННЯ ПРОГРАМИ РЕЦЕНЗУВАННЯ ВУЛИЧНИХ АКТИВНОСТЕЙ		
Показник	Django	WordPress
Тип інструменту	Фреймворк для веброзробки	Система керування вмістом
Доступні інструменти для організації баз даних	PostgreSQL, MariaDB, MySQL, Oracle, SQLite (офіційно) з підтримкою OPB	MySQL, MariaDB
Мова	Python	PHP
Безпека роботи	Відповідає дуже високому рівню, забезпечуючи обробку багатьох ситуацій	Може мати проблеми зокрема за рахунок необхідності встановлення рішень третіх сторін для реалізації
Ліцензія	Відкрита	Відкрита

4

Рисунок В.4 – Порівняння рішень для створення програми рецензування вуличних активностей

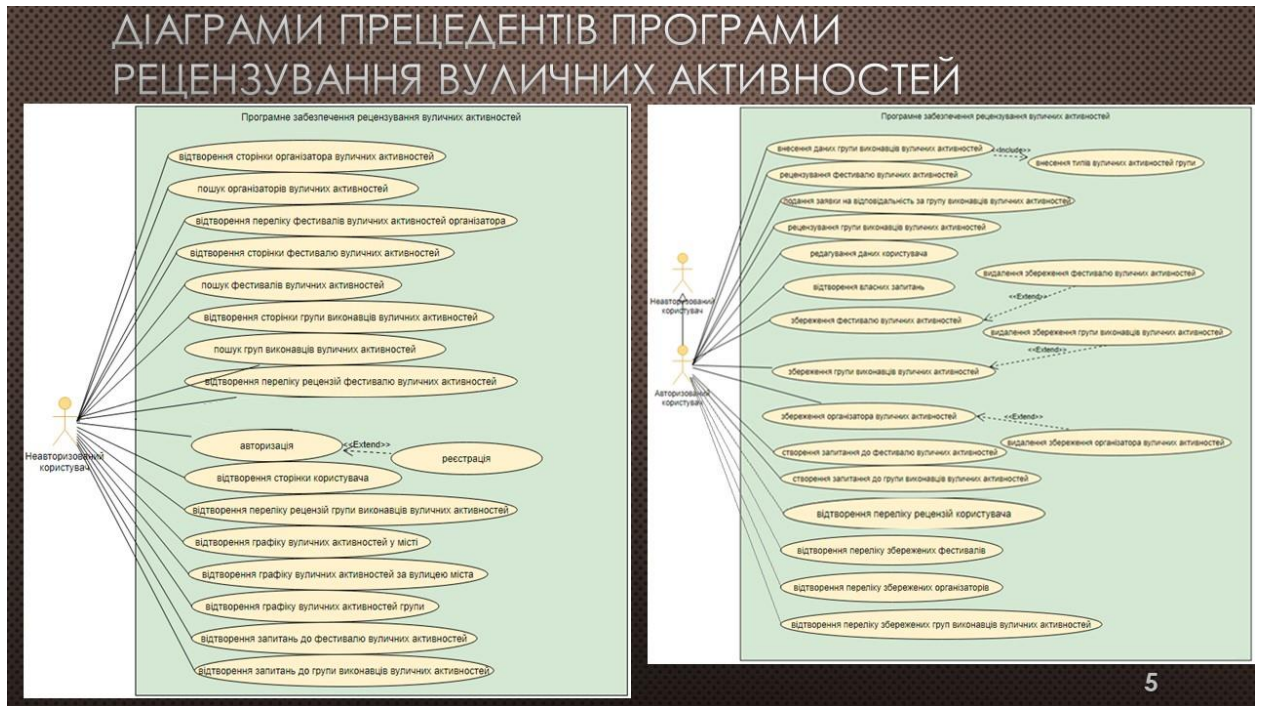


Рисунок В.5 – Діаграми прецедентів програми рецензування вуличних активностей

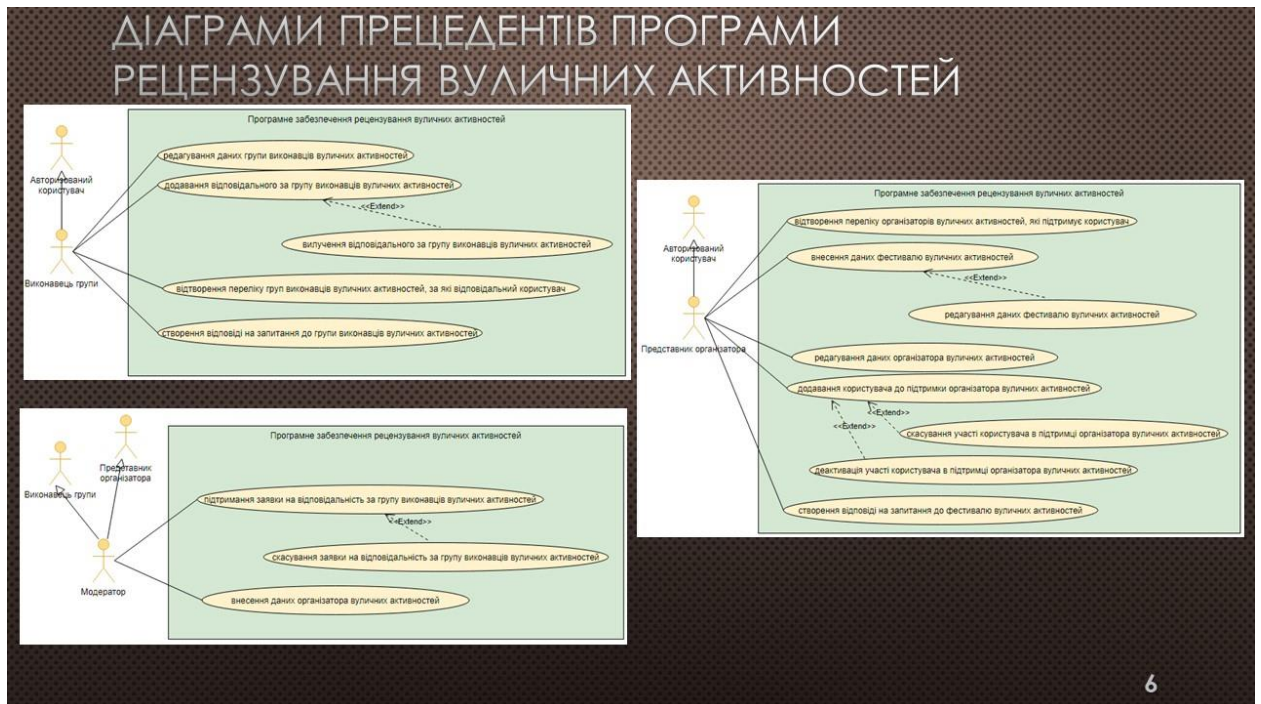


Рисунок В.6 – Діаграми прецедентів програми рецензування вуличних активностей

Рисунок В.7 – Схема бази даних програми рецензування вуличних активностей



Рисунок В.8 – Схема функціонування програми рецензування вуличних активностей

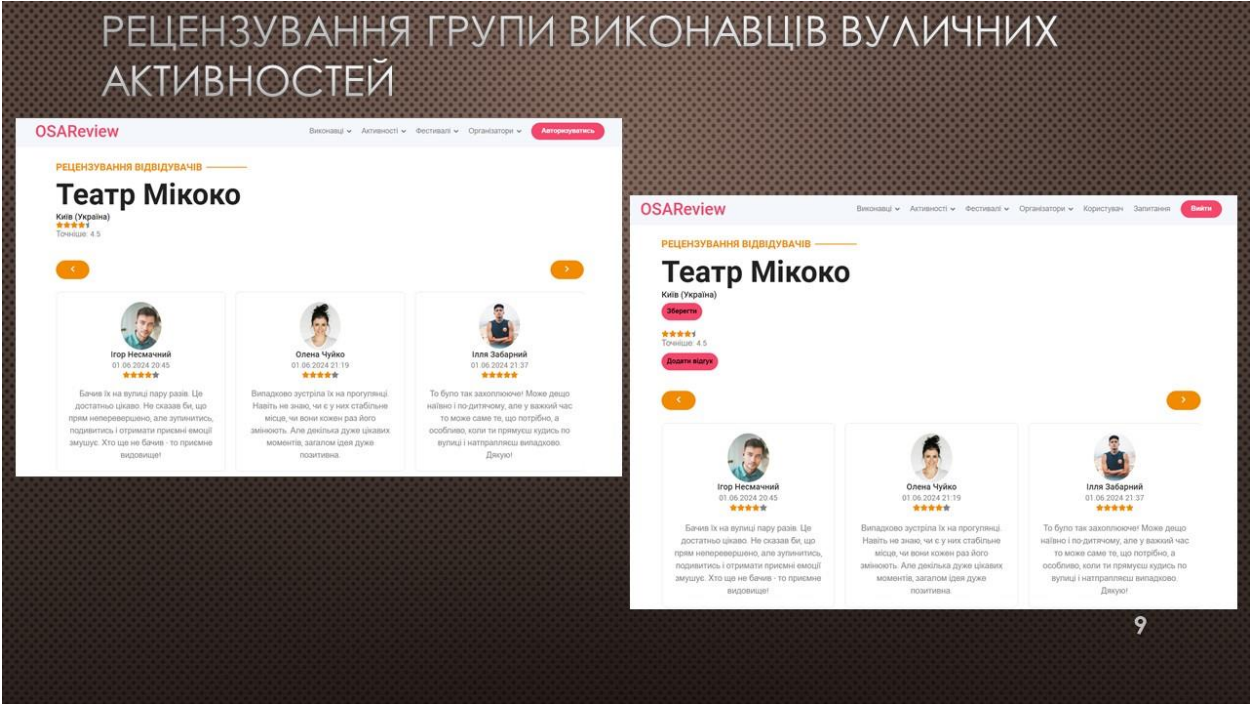


Рисунок В.9 – Рецензування групи виконавців вуличних активностей

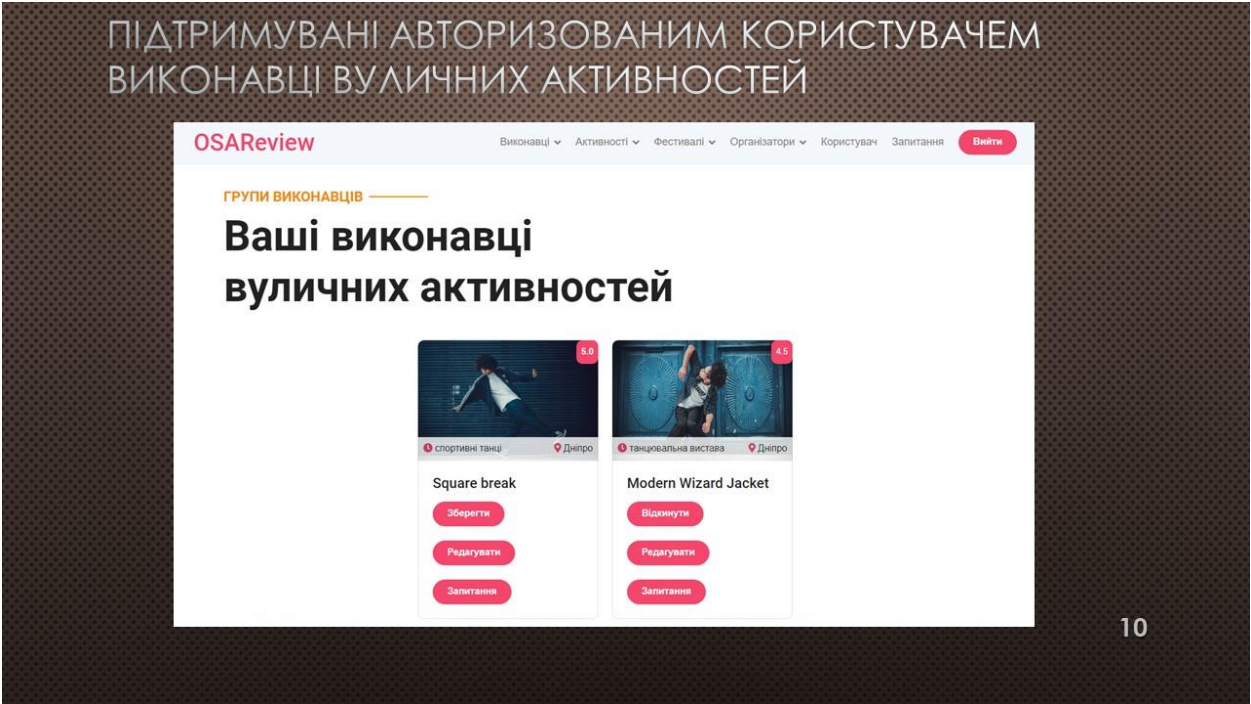


Рисунок В.10 – Підтримувані авторизованим користувачем виконавці вуличних активностей

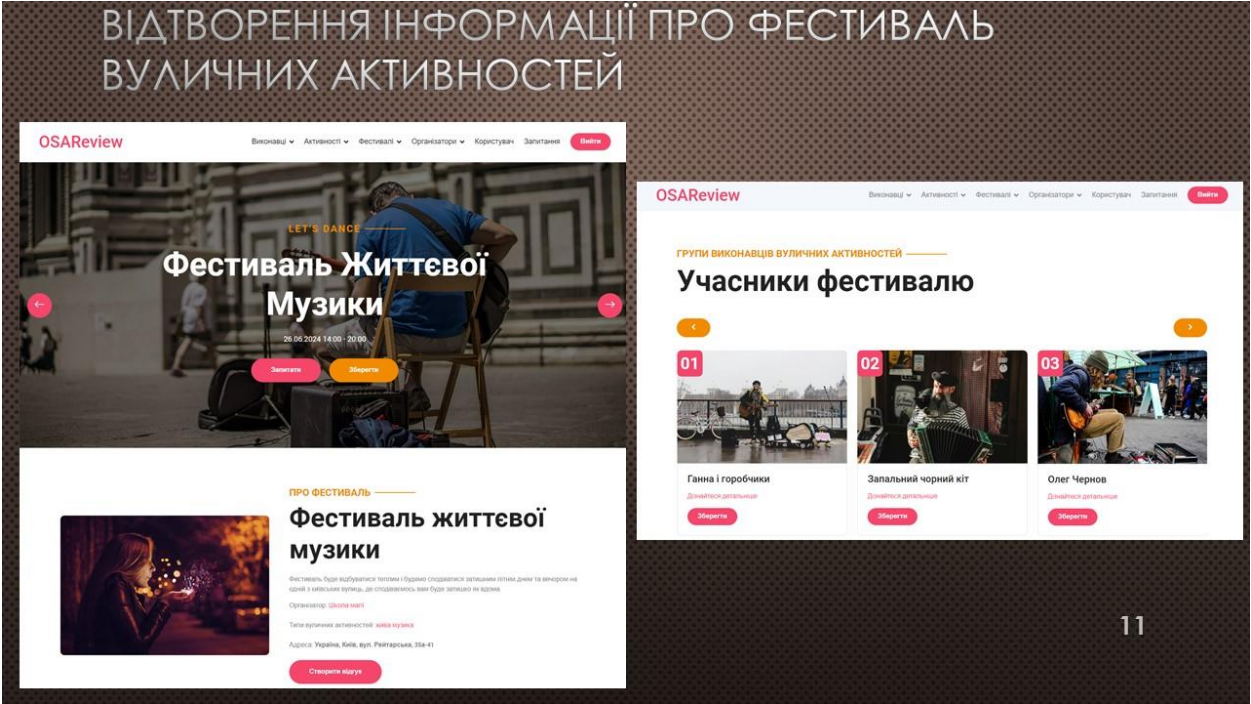


Рисунок В.11 – Відтворення інформації про фестиваль вуличних активностей

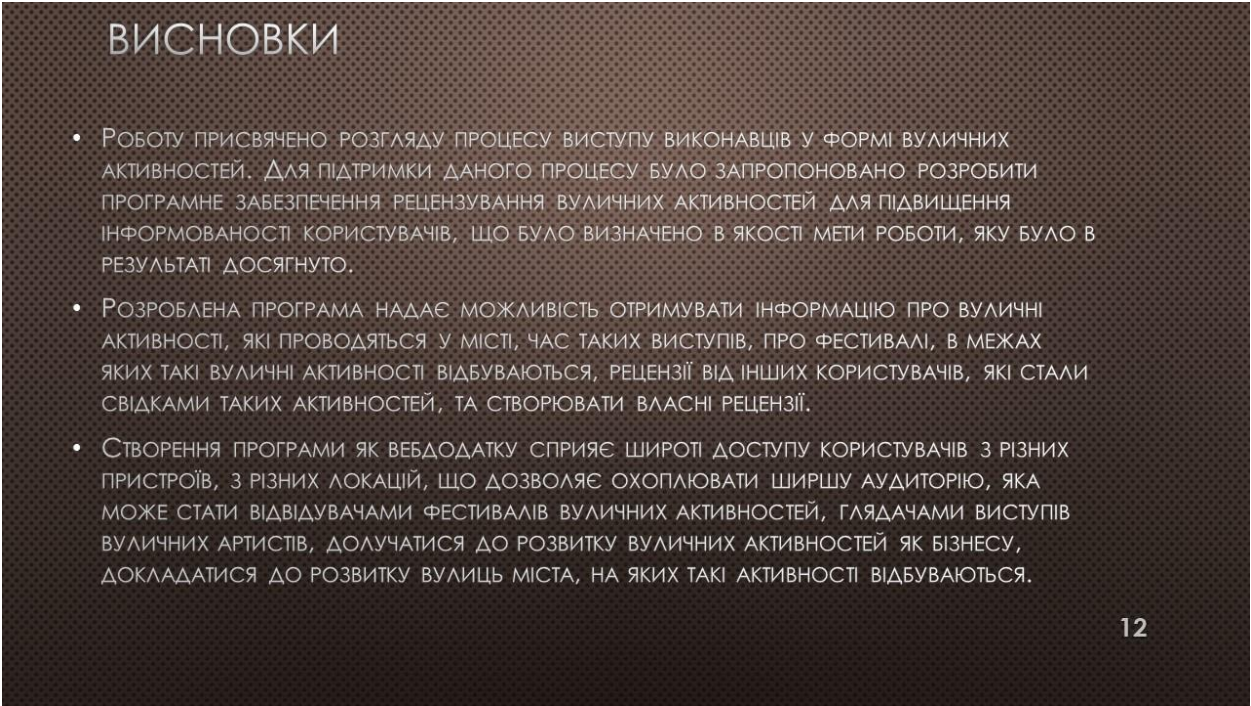


Рисунок В.12 – Висновки