

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ	Економіки та бізнес-освіти
Кафедра	Економіки та цифрового бізнесу
Спеціальність	Комп'ютерні науки
Форма навчання	Денна

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Горбенко Анастасії Олександрівни	
(прізвище, ім'я, по батькові здобувача)	
на тему	Розробка вебсистеми управління коштами благодійного фонду
(повна назва теми)	
за матеріалами	
(повна назва бази дослідження)	
науковий керівник	Шокотько Л.М.
(наук. ступінь, вчене звання)	(підпис) (прізвище, ініціали)

Робота допущена до захисту в ЕК

Протокол засідання кафедри

від 12.06.2026р. № 15

Завідувач кафедри

(підпис)

к.е.н., доцент
наук. ступень, вчене звання

Радько В.М.
прізвище, ініціали

ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки, молоді та спорту України

29 березня 2012 року № 384

Форма № Н-9.01

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ
(повне найменування вищого навчального закладу)

Кафедра економіки та цифрового бізнесуОсвітній ступінь бакалаврСпеціальність 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ **В.М. Радько**“30” березня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА ЗДОБУВАЧУ
Горбенко Анастасії Олександрівні

1. Тема роботи Розробка вебсистеми управління коштами благодійного фондунауковий керівник роботи Шокотько Людмила Миколаївна,затверджені наказом вищого навчального закладу від «23» березня 2026 р. № 193-ст2. Строк подання здобувачем роботи 29.05.2026р.

3. Зміст кваліфікаційної бакалаврської роботи, об'єкт, предмет та мета дослідження:

Розділ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУРозділ 2 ПРОЄКТУВАННЯ СТРУКТУРИ ТА БАЗИ ДАНИХ ВЕБСИСТЕМИРозділ 3 РОЗРОБКА ВЕБСИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУ

Об'єкт дослідження – процеси цифровізації обліку, контролю, розподілу та звітування щодо руху коштів у діяльності благодійного фонду.

Предмет дослідження – моделі, методи та програмні засоби проєктування і реалізації вебсистеми управління коштами благодійного фонду, зокрема засоби обліку пожертв, витрат, бюджетів, фінансових звітів і аналітичних показників.

Мета кваліфікаційної бакалаврської роботи – проєктування та розроблення вебсистеми для управління коштами благодійного фонду, що забезпечує облік пожертв, контроль витрат, бюджетування, формування фінансової звітності, розрахунок аналітичних показників і розмежування прав доступу користувачів.

4. Дата видачі завдання 03.04.2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 17.04.2026р.	17.04.2026р.
2	Підготовка розділу 2	до 08.05.2026р.	08.05.2026р.
3	Підготовка розділу 3	до 25.05.2026р.	25.05.2026р.
4	Реєстрація завершеної кваліфікаційної роботи	до 29.05.2026р.	29.05.2026р.
5	Отримання відгуку від наукового керівника	04.06.2026р.	04.06.2026р.
6	Отримання зовнішньої рецензії	05.06.2026р.	05.06.2026р.
7	Попередній захист кваліфікаційної роботи на кафедрі	08.06.2026р.	08.06.2026р.
8	Перевірка кваліфікаційної роботи на плагіат	12.06.2026р.	12.06.2026р.
9	Допуск кафедрою кваліфікаційної роботи до захисту	12.06.2026р.	12.06.2026р.
10	Підготовка студента до захисту в ЕК	до 19.06.2026р.	19.06.2026р.

Завдання підготував науковий керівник

Шокотько Л. М.

Завдання одержав здобувач

Горбенко А.О.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 82 сторінки, 19 рисунків, 8 таблиць, 27 джерел, 8 додатків.

Об'єкт дослідження: процес управління коштами благодійного фонду в умовах використання вебсистеми.

Мета: проєктування та розроблення вебсистеми для управління коштами благодійного фонду з підтримкою обліку фінансових операцій, формування звітності, відображення аналітичних показників і розмежування прав доступу користувачів.

У роботі проаналізовано особливості діяльності благодійних фондів та сучасні вебрішення для фінансового обліку. Визначено функціональні вимоги до системи, спроектовано її архітектуру та структуру бази даних. Реалізовано веб-систему на основі Python, Django та SQLite.

Система FondUA забезпечує реєстрацію та автентифікацію користувачів, рольовий доступ, перегляд фондів і благодійних проєктів, облік пожертв і витрат, бюджетування, формування фінансових звітів та відображення аналітичних показників. Передбачено ролі донора, представника фонду та суперадміністратора. Особливу увагу приділено фінансовій логіці системи: розрахунку надходжень, витрат, залишку коштів і показників виконання зборів. Проведено тестування основних сценаріїв роботи, що підтвердило коректність функціонування розробленого програмного забезпечення.

Практична цінність роботи полягає у можливості використання системи для автоматизації фінансових процесів благодійних організацій, підвищення прозорості використання коштів та покращення звітності.

Ключові слова: автентифікація, аналітика, база даних, благодійний фонд, бюджетування, вебсистема, витрата, пожертва, рольовий доступ, управління коштами, фінансова звітність, цифровізація, Django, Python, SQLite.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУ	10
1.1. Особливості фінансового управління у діяльності благодійних фондів.....	10
1.2. Огляд сучасних веб-рішень для благодійних зборів і фінансової звітності.....	13
1.3. Аналіз інформаційних потоків та постановка задачі	18
Висновки до розділу 1	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ СТРУКТУРИ ТА БАЗИ ДАНИХ ВЕБСИСТЕМИ FondUA	23
2.1. Проєктування загальної архітектури системи	23
2.2. Моделювання бази даних для обліку фондів, проєктів і фінансових операцій.....	27
2.3. Розроблення фінансової логіки та аналітичних показників системи	37
Висновки до розділу 2	44
РОЗДІЛ 3. РОЗРОБКА ВЕБСИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУ FondUA	45
3.1. Обґрунтування вибору інструментів для створення веб-застосунку	50
3.2. Розробка серверної частини та механізмів керування даними	61
3.3. Реалізація клієнтської частини та аналітичної панелі системи	61
3.4. Перевірка працездатності вебсистеми та аналіз результатів реалізації	71
Висновки до розділу 3	76
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	83

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СКБД – Система керування базами даних.

API (Application Programming Interface) – інтерфейс програмування застосунків.

CSS (Cascading Style Sheets) – каскадні таблиці стилів.

CSV – Comma-Separated Values.

Django ORM – механізм об'єктно-реляційного відображення Django.

ER-діаграма – Entity Relationship Diagram.

HTML (HyperText Markup Language) – мова розмітки гіпертексту.

JS – JavaScript.

JSON (JavaScript Object Notation) – формат обміну даними.

ORM – Object-Relational Mapping.

RDBMS – Relational Database Management System.

SQL (Structured Query Language) – структурована мова запитів.

UI (User Interface) – інтерфейс користувача.

URL (Uniform Resource Locator) – уніфікований покажчик ресурсу.

UX (User Experience) – користувацький досвід.

XLSX – Microsoft Excel Open XML Spreadsheet.

ВСТУП

Благодійні фонди дедалі активніше використовують цифрові інструменти для публікації зборів, взаємодії з донорами, фіксації пожертв і подання фінансової звітності. Для таких організацій важлива не лише наявність інформаційного сайту, а повноцінна вебсистема, у якій публічне представлення благодійних проєктів поєднується з внутрішнім обліком коштів. Якщо пожертви, витрати, бюджетні плани, підтвердні документи й фінансові звіти зберігаються окремо, ускладнюється контроль фінансових потоків і знижується прозорість роботи фонду.

Актуальність теми зумовлена потребою у створенні вебсистеми, що забезпечує єдиний інформаційний контур для управління коштами благодійного фонду. У такій системі мають бути пов'язані публічні сторінки фондів і проєктів, облік пожертв, контроль витрат, бюджетування, фінансова звітність, аналітичне подання показників і розмежування доступу між різними категоріями користувачів. Для благодійної сфери це має практичне значення, оскільки донори очікують відкритої інформації про використання коштів, а представники фонду потребують зручних інструментів для роботи з фінансовими записами.

Ступінь опрацювання проблеми є достатнім для формування теоретичної та технічної основи роботи, але нерівномірним за змістом. У наукових працях і технічних матеріалах розглянуто цифровізацію благодійної діяльності, вимоги до платформ пожертв, принципи проєктування інформаційних систем, баз даних, вебархітектури й захисту даних. Водночас поєднання публічного контуру благодійної платформи з внутрішнім управлінням пожертвами, витратами, бюджетами, звітністю та аналітичними показниками розкрито недостатньо. Це визначає потребу у проєктуванні вебсистеми FondUA.

Проблема роботи полягає в тому, що багато цифрових рішень для благодійності зосереджені лише на публічному зборі коштів або на окремому поданні звітності. Для повноцінної роботи фонду цього недостатньо, оскільки фінансові дані мають проходити повний маршрут: від створення фонду й благодійного проєкту до фіксації пожертви, внесення витрати, планування

бюджету, формування звіту та аналізу результатів. У FondUA цей маршрут покладено в основу архітектури, структури бази даних і функціональних модулів системи.

Джерельну базу роботи становлять нормативно-правові акти України щодо благодійної діяльності та захисту персональних даних, наукові й навчальні матеріали з проєктування інформаційних систем, баз даних, вебзастосунків і кібербезпеки, офіційна документація Django, Python, SQLite, HTML, CSS, JavaScript і Chart.js, а також матеріали програмної реалізації FondUA.

Об'єктом роботи є процеси цифровізації обліку, контролю, розподілу та звітування щодо руху коштів у діяльності благодійного фонду.

Предметом роботи є моделі, методи та програмні засоби проєктування і реалізації клієнт-серверної вебсистеми FondUA для обліку фондів, благодійних проєктів, пожертв, витрат, бюджетів, фінансових звітів та аналітичних показників.

Метою роботи є проєктування та розроблення вебсистеми FondUA для управління коштами благодійного фонду, що забезпечує облік фондів і благодійних проєктів, фіксацію пожертв, контроль витрат, бюджетування, формування фінансової звітності, розрахунок аналітичних показників і розмежування доступу між відвідувачем, донором, представником фонду та суперадміністратором.

Для досягнення поставленої мети визначено такі завдання:

- проаналізувати особливості фінансового управління в діяльності благодійних фондів;
- розглянути сучасні вебрішення для благодійних зборів і визначити їхні функціональні обмеження;
- сформулювати вимоги до вебсистеми FondUA з урахуванням ролей користувачів;
- спроектувати архітектуру, інформаційну модель і структуру бази даних;
- розробити фінансову логіку для розрахунку пожертв, витрат, залишку коштів, план-фактного порівняння та аналітичних показників;

- реалізувати основні модулі вебсистеми та перевірити їхню працездатність за ключовими користувацькими сценаріями.

У роботі використано методи системного й порівняльного аналізу, інформаційного моделювання, реляційного проєктування бази даних та функціонального тестування.

Елемент новизни роботи полягає у поєднанні публічного представлення благодійних фондів і проєктів із внутрішнім контуром фінансового управління в межах єдиної клієнт-серверної вебсистеми. FondUA об'єднує рольовий доступ, облік пожертв, контроль витрат, бюджетування, фінансову звітність та аналітичну обробку даних.

Практична цінність одержаних результатів полягає у можливості використання FondUA як автономної вебсистеми для автоматизації роботи благодійного фонду, упорядкування фінансових операцій, контролю бюджетів, підготовки звітності, формування аналітичних показників і подальшого розширення функціоналу.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУ

1.1 Особливості фінансового управління у діяльності благодійних фондів

Благодійний фонд доцільно розглядати не лише як організацію, що залучає пожертви, а як учасника фінансових процесів, у яких поєднуються цільове надходження коштів, їх розподіл, контроль витрат, бюджетування та підготовка фінансової звітності. Відповідно до Закону України «Про благодійну діяльність та благодійні організації», благодійний фонд є одним із видів благодійних організацій, а благодійники мають право контролювати цільове використання благодійної пожертви [3]. Це означає, що для фонду важливо не тільки прийняти кошти, а й забезпечити простежуваність їх подальшого використання.

Фінансове управління у діяльності благодійного фонду охоплює кілька взаємопов'язаних етапів. Першим етапом є отримання пожертв від донорів, партнерів або організацій, які підтримують конкретний благодійний проєкт. Для коректного обліку недостатньо фіксувати лише суму надходження: важливими є дата операції, її статус, цільове призначення, зв'язок із фондом або проєктом і можливість подальшого включення цієї інформації до фінансового звіту. Наступним етапом є планування та облік витрат, оскільки фонд повинен контролювати, які потреби фінансуються, які суми вже використано, які витрати підтверджено і чи не перевищують вони запланований бюджет. Завершальним елементом цього циклу є звітність, яка узагальнює дані про надходження, витрати та залишки коштів [1].

У праці О. Корнак і Ю. Грицюка підкреслено, що система управління коштами благодійного фонду має забезпечувати збирання, реєстрацію, накопичення, зберігання, аналіз, захист і надання достовірної інформації учасникам благодійного процесу [1]. Такий підхід безпосередньо пов'язаний із сутністю інформаційної системи, яка впорядковує дані, процеси та зв'язки між

ними для підтримки управлінських рішень [15]. Для благодійного фонду це означає потребу в єдиному середовищі, де пожертви, витрати, категорії витрат, бюджети й фінансові звіти не існують окремо, а пов'язані з конкретними фондами та благодійними проектами.

Окремою особливістю фінансового управління благодійного фонду є контроль статусів фінансових записів. Пожертва, витрата або фінансовий звіт не повинні автоматично впливати на підсумкові показники лише через сам факт створення запису. Для коректного обліку потрібно розрізняти створені, підтверджені, відхилені та опубліковані записи. Наприклад, витрата зі статусом очікування не повинна зменшувати залишок коштів проекту, доки її не буде погоджено відповідальною роллю. Такий принцип дає змогу відокремити первинне внесення інформації від її включення до фінансової звітності та аналітики.

Фрагментований або ручний облік ускладнює фінансове управління. Якщо дані про пожертви зберігаються в одних таблицях, витрати - в інших документах, а звіти формуються вручну, виникає ризик дублювання записів, втрати підтвердних відомостей і помилок у визначенні залишку коштів. У такій ситуації складно швидко встановити, яка частина коштів уже зібрана, які витрати підтверджені, які суми заплановані в бюджеті та які дані мають бути відображені у звіті. Саме тому цифрова система управління коштами повинна забезпечувати не тільки збереження фінансових записів, а й логічний зв'язок між етапами руху коштів.

Для FondUA ці проблеми визначають базову логіку системи: кожна жертва, витрата, бюджетний запис і фінансовий звіт мають бути пов'язані з конкретним благодійним проектом. Це дає змогу уникати розрізненого обліку та формувати фінансові підсумки не вручну, а на основі пов'язаних даних системи. У межах такої логіки пожертви використовуються для визначення зібраної суми, підтверджені витрати - для розрахунку фактичного використання коштів, бюджетні плани - для план-фактного порівняння, а фінансові звіти й аналітичні показники - для узагальнення результатів.

Основні проблеми фінансового обліку благодійних фондів та їх наслідки узагальнено в табл. 1.1.

Таблиця 1.1.

Основні проблеми фінансового обліку благодійних фондів та їх наслідки

Проблема	Причина виникнення	Потенційні наслідки
Розрізнений облік пожертв	Дані надходять з різних каналів і фіксуються окремо	Помилки в балансі та складність перевірки
Ручне ведення обліку витрат	Використання таблиць, чатів, паперових документів	Затримки, дублювання записів, втрата доказів
Відсутність контролю статусів фінансових записів	Пожертви, витрати або звіти враховуються без належної перевірки	Викривлення фінансових показників і ризик включення неперевірених даних до звітності
Нечітке розмежування доступу	Різні ролі працюють без єдиних правил доступу	Несанкціонована зміна або перегляд даних
Розрив між бюджетом і фактичними витратами	Планові суми та реальні витрати ведуться окремо	Неможливість швидко оцінити перевищення бюджету або невикористані кошти
Неповна звітність	Відсутність єдиного реєстру та історії операцій	Зниження довіри донорів і партнерів

(Розроблено автором)

Окреме значення має прозорість взаємодії з донорами. У вимогах до інтернет-платформ для благодійних пожертв наголошено на потребі зрозумілого подання інформації, безпеки, перевірності даних і довіри до цифрового благодійного сервісу [2]. Для благодійного фонду прозорість означає можливість показати мету збору, поточний стан фінансування, підтверджені витрати та підсумкові фінансові звіти. Якщо донор бачить не лише сам факт збору, а й результат використання коштів, це посилює довіру до фонду та зменшує інформаційну невизначеність.

У межах вебсистеми FondUA зазначені особливості предметної області формують основні вимоги до програмного рішення. Система має підтримувати роботу з фондами, благодійними проєктами, жертвами, витратами, бюджетами, фінансовими звітами й аналітичними показниками. Доступ до дій має залежати від ролі користувача: неавторизований відвідувач переглядає відкриті сторінки фондів, проєктів і звітів; донор взаємодіє з жертвами; представник фонду працює зі

службовими даними власного фонду, проєктами, витратами, бюджетами та звітами; суперадміністратор контролює статуси, перевірку записів і коректність даних. Така рольова модель потрібна для впорядкування фінансових процесів і зменшення ризику неконтрольованих змін.

Під час проєктування FondUA важливо врахувати поділ відкритих і службових даних. Відвідувачеві та донору доцільно показувати мету збору, фінансовий прогрес, підтверджені витрати й опубліковані звіти, тоді як службові дані про перевірку витрат, повні персональні відомості та внутрішні фінансові записи мають бути доступні лише користувачам із відповідними правами. Це узгоджується з потребою захисту персональних даних і керованого доступу до фінансової інформації [4].

Отже, фінансове управління благодійного фонду потребує не окремого сайту для збору коштів, а цілісної вебсистеми, що поєднує облік пожертв, контроль витрат, бюджетування, звітність, аналітичні показники та розмежування доступу. Ці передумови обґрунтовують необхідність подальшого аналізу наявних вебрішень для благодійних зборів і фінансової звітності, а також створюють основу для проєктування вебсистеми FondUA.

1.2 Огляд сучасних веб-рішень для благодійних зборів і фінансової звітності

Сучасні вебрішення для благодійних зборів і фінансової звітності доцільно розглядати не як один універсальний тип системи, а як сукупність цифрових інструментів, кожен із яких підтримує окремий етап роботи благодійної організації. Для благодійного фонду важливо не лише публікувати збори та приймати пожертви, а й забезпечувати облік надходжень, контроль витрат, бюджетування, формування фінансових звітів, аналітичне подання показників і розмежування доступу між різними категоріями користувачів. Закон України «Про благодійну діяльність та благодійні організації» закріплює право благодійника контролювати цільове використання пожертви та визначає обов'язок благодійних організацій

щодо фінансової та іншої передбаченої законом звітності [3]. Через це цифрова система для благодійного фонду повинна підтримувати не тільки публічний збір коштів, а й внутрішній фінансовий контур.

У практиці цифрової благодійності можна виділити кілька основних груп вебрішень: публічні платформи благодійних зборів, сайти благодійних організацій із відкритою звітністю, внутрішні CRM або облікові системи, платіжні шлюзи, а також сервіси аналітики та звітності. У працях, присвячених вимогам до інтернет-платформ благодійних пожертв, наголошено на важливості прозорості, зручності подання інформації та довіри до сервісу, оскільки донор приймає рішення про підтримку на основі зрозумілості потреби, умов збору й доступності звітних даних [2]. Водночас проєктування інформаційної системи передбачає не лише створення зручного інтерфейсу, а й упорядкування даних, зв'язків і процесів, які забезпечують керованість усієї системи [15].

Публічні платформи благодійних зборів найкраще працюють на етапі залучення донорів. Прикладами таких рішень є dobro.ua та UNITED24, які орієнтовані на публічне представлення благодійних ініціатив, пояснення потреби, відображення результатів збору та спрощення процесу підтримки. Їхня сильна сторона полягає у зрозумілому сценарії для донора: користувач бачить мету збору, короткий опис, напрям підтримки, суму збору та може швидко виконати благодійну дію. Для публічної комунікації це зручно, оскільки платформа одразу показує потребу й формує довіру через відкритість інформації.

Однак для внутрішнього управління коштами можливостей публічних платформ недостатньо. Вони насамперед орієнтовані на збір коштів і комунікацію з аудиторією, тоді як службовий облік витрат, погодження фінансових записів, планування бюджету, контроль залишку коштів і підготовка внутрішніх звітів можуть залишатися поза межами основного сценарію. Для FondUA це є важливим обмеженням, оскільки система має підтримувати не тільки показ благодійного проєкту, а й повний маршрут фінансових даних: від створення фонду й проєкту до фіксації пожертви, внесення витрати, планування бюджету, формування звіту та розрахунку аналітичних показників.

Сайти благодійних організацій із відкритою звітністю виконують іншу функцію. Вони можуть містити опис фонду, історію діяльності, документи, новини, звіти, інформацію про завершені проєкти та результати допомоги. Такий сайт є важливим каналом публічної довіри, оскільки показує не лише факт збору коштів, а й діяльність організації загалом. Для донорів це має значення, адже текстове й візуальне подання благодійної діяльності впливає на сприйняття організації та зрозумілість її потреб [18].

Водночас сайт із відкритою звітністю не завжди є повноцінною інформаційною системою для фінансового управління. Якщо звітність готується в окремих таблицях або документах, а сайт лише публікує готові матеріали, між пожертвами, витратами, бюджетами та фінансовими звітами може не бути автоматизованого зв'язку. У такому випадку публічна прозорість залежить від ручного перенесення даних, що підвищує ризик помилок, дублювання записів і втрати підтвердної інформації. Для FondUA важливо усунути саме цей розрив між публічним представленням результатів і первинним фінансовим обліком.

Внутрішні CRM або donor management systems орієнтовані переважно на адміністративну роботу з донорами, контактами, історією взаємодії та службовими записами. Їхньою перевагою є краща організація внутрішніх процесів: фонд може бачити базу донорів, історію внесків, комунікації та службові примітки. Такі системи корисні для підтримки довгострокових відносин із благодійниками, однак вони не завжди поєднуються з публічним каталогом благодійних проєктів, відкритою звітністю, бюджетуванням і аналітикою за окремими проєктами. Для задачі FondUA цього недостатньо, оскільки система повинна зв'язувати внутрішній облік із відкритою частиною, де донор бачить фонд, проєкт, фінансовий стан збору та опубліковані результати. (табл. 1.2.)

Таблиця 1.2

Порівняння сучасних веб-рішень із потребами вебсистеми FondUA

Рішення або клас рішення	Основне призначення	Сильні сторони	Обмеження для задачі FondUA
Публічні платформи благодійних зборів, зокрема dobro.ua та UNITED24	Розміщення зборів, опис потреб, приймання пожертв, публічне інформування донорів	Зручний сценарій підтримки, швидке залучення донорів, зрозуміле подання мети збору	Обмежена підтримка внутрішнього фінансового контуру, витрат, бюджетів, службових статусів і детальної звітності
Сайти благодійних організацій із відкритою звітністю	Представлення фонду, новин, документів, звітів і результатів допомоги	Підвищення довіри, публічне пояснення діяльності фонду, відкритість результатів	Звітність може публікуватися окремо від первинного обліку пожертв, витрат і бюджетів
Внутрішні CRM або облікові системи фонду	Облік донорів, контактів, службових записів, історії взаємодії та операцій	Керованість внутрішніх процесів, збереження історії роботи з донорами	Можуть не мати відкритої частини для донорів, публічного каталогу проєктів і прозорої фінансової звітності
Платіжні шлюзи	Приймання онлайн-платежів, фіксація статусів транзакцій, формування платіжних підтверджень	Технічна підтримка переказу коштів, автоматизація платіжного етапу	Не забезпечують бюджетування, облік витрат, погодження фінансових записів, фінансову звітність і аналітику
Сервіси аналітики та звітності	Побудова діаграм, зведень, таблиць і панелей показників	Зручна візуалізація підготовлених даних, підтримка управлінського аналізу	Часто потребують імпорту даних з інших систем і не є первинним середовищем фінансового обліку

(Розроблено автором)

Платіжні шлюзи забезпечують приймання онлайн-платежів, оброблення транзакцій, фіксацію статусів операцій і формування платіжних підтверджень. Для благодійного фонду це важливий інфраструктурний компонент, але сам по собі він не вирішує задачу управління коштами. Платіжний сервіс не веде облік витрат, не формує бюджетні плани, не розподіляє службові права представників фонду й не створює цілісної фінансової звітності щодо використання пожертв. Тому в межах FondUA пожертва розглядається не як завершення процесу, а як початок подальшого фінансового ланцюга: надходження має бути пов'язане з благодійним проектом, витратами, бюджетом, звітністю та аналітичними показниками.

Групу становлять сервіси аналітики та звітності. Вони можуть формувати панелі показників, діаграми, зведені таблиці й візуалізацію фінансових даних. Такі інструменти корисні для аналізу надходжень, структури витрат, динаміки зборів і виконання бюджету. Проте зазвичай вони працюють із даними, підготовленими в іншій системі, тобто не є первинним середовищем обліку. Для FondUA важливо, щоб аналітичні показники формувалися на основі власних даних про фонди, благодійні проекти, пожертви, витрати, бюджетні плани та фінансові звіти, а не тільки після ручного імпорту або окремого опрацювання.

Окремо потрібно враховувати вимоги до звітності та роботи з даними користувачів. Законодавство у сфері благодійної діяльності передбачає фінансову й іншу обов'язкову звітність благодійних організацій [3]. Якщо система працює з даними донорів або представників фонду, вона також має враховувати вимоги до обробки персональних даних, зокрема визначеність мети обробки, прозорість роботи з такими даними та обмеження доступу до службової інформації [4]. Тому для FondUA важливими є не лише функції збору пожертв, а й рольове розмежування доступу, контроль службових дій, коректне подання звітів і зрозумілий зв'язок між публічною та внутрішньою частинами системи.

FondUA проєктується як навчально-практичне рішення, орієнтоване на поєднання обліку пожертв, витрат, бюджетування, фінансової звітності та аналітики в межах однієї вебсистеми. На відміну від рішень, що виконують лише окрему функцію, у FondUA передбачається зв'язок між фондом, благодійним

проектом, пожертвою, витратою, бюджетним планом, фінансовим звітом і аналітичними показниками. Це створює основу для подальшого визначення ролей користувачів, інформаційних потоків, вхідних і вихідних даних, а також постановки задачі проектування.

Отже, аналіз сучасних вебрішень показує, що наявні інструменти ефективно підтримують окремі задачі цифрової благодійності, але не забезпечують повного циклу управління коштами благодійного фонду в межах єдиної системи. Саме тому подальше проектування FondUA має бути спрямоване на створення цілісного інформаційного середовища, у якому публічна взаємодія з донорами поєднується з внутрішнім обліком пожертв, витрат, бюджетів, звітності, аналітики та рольового доступу.

1.3 Аналіз інформаційних потоків та постановка задачі

Після розгляду особливостей фінансового управління благодійного фонду та огляду сучасних вебрішень необхідно визначити інформаційні потоки, на яких ґрунтується подальше проектування FondUA. У межах цієї роботи інформаційний потік розглядається як послідовність введення, перевірки, збереження, оброблення, узагальнення та подання даних між користувачами, внутрішніми модулями системи й вихідними фінансовими результатами. Такий підхід дає змогу пов'язати діяльність благодійного фонду з конкретними функціональними вимогами до вебсистеми [15].

Для FondUA доцільно виокремити три взаємопов'язані групи даних. До першої належать довідкові та організаційні дані: облікові записи користувачів, профілі, благодійні фонди, благодійні проекти, категорії витрат, ролі та статуси. До другої групи належать операційні фінансові дані: записи пожертв, витрат, бюджетних планів, бюджетних рядків і фінансових звітів із прив'язкою до дати, періоду, категорії, проекту та фонду. Третю групу становлять вихідні дані, сформовані на основі попередніх двох груп: зібрані суми, підтверджені витрати,

залишки коштів, план-фактні показники, фінансові звіти, аналітичні зведення та публічні сторінки з результатами діяльності [1; 15].

Центральним елементом інформаційних потоків FondUA є благодійний проєкт. Саме він поєднує публічну та службову частини системи: на публічному рівні проєкт подається як мета збору, опис потреби та фінансовий прогрес, а на службовому рівні з ним пов'язуються пожертви, витрати, бюджетні записи, звіти й аналітичні показники. Така організація дає змогу простежити рух даних від моменту створення фонду і проєкту до формування фінансового результату й публічного подання інформації.

Основні інформаційні потоки вебсистеми FondUA наведено в табл. 1.3.

Таблиця 1.3

Основні інформаційні потоки вебсистеми FondUA

Джерело	Вхідні дані	Оброблення в системі	Вихідний результат
1	2	3	4
Неавторизований відвідувач	Параметри перегляду, навігаційні запити, вибір фонду або проєкту	Відбір відкритих фондів, активних або завершених проєктів, опублікованих звітів	Публічні сторінки фондів, проєктів, звітів і фінансових підсумків, доступних без авторизації
Донор	Дані пожертви, сума, спосіб оплати, контактні дані за потреби, ознака анонімності	Перевірка доступності проєкту, валідація форми, створення запису пожертви, фіксація статусу операції	Запис пожертви, повідомлення про результат дії, сторінка підтвердження або оновлений фінансовий стан проєкту
Представник фонду	Дані фонду, проєктів, витрат, бюджетів і фінансових звітів	Перевірка прав доступу, збереження службових записів, зв'язування даних із фондом і проєктом, встановлення службових статусів	Актуалізовані дані фонду, проєктів, витрат, бюджетів і звітів у службовому контурі

Продовження табл. 1.3

1	2	3	4
Суперадмін істратор	Дані для перевірки статусів, службові рішення щодо витрат, звітів і фондів	Контроль коректності даних, погодження або відхилення записів, перевірка доступу до службових дій	Підтверджені або відхилені записи, оновлені статуси фондів, витрат і звітів
Фінансові модулі системи	Пожертви, витрати, бюджетні рядки, періоди, категорії, статуси	Групування даних за фондом, проєктом, категорією, статусом і періодом; обчислення фінансових показників	Зібрана сума, підтверджені витрати, залишок коштів, прогрес збору, план-фактне порівняння
Звітність та аналітика	Підтверджені фінансові записи за вибраний період	Узагальнення даних, формування звітів, аналітичних зведень, таблиць і графічних показників	Фінансові звіти, аналітичні панелі, експортовані зведення, публічне або службове подання результатів

(Розроблено автором)

Участь користувачів в інформаційних потоках залежить від їхньої ролі. Неавторизований відвідувач працює лише з відкритою частиною системи: переглядає фонди, благодійні проєкти, описи зборів і опубліковані звіти. Донор взаємодіє з проєктом як із об'єктом підтримки: переглядає його, ініціює пожертву та отримує інформацію про результат виконаної дії. Представник фонду працює зі службовою частиною власної організації: керує даними фонду, створює й редагує проєкти, вносить витрати, формує бюджетні плани та готує фінансові звіти. Суперадміністратор виконує контрольну функцію: перевіряє службові записи, змінює статуси, погоджує або відхиляє витрати та звіти, а також контролює коректність даних у межах системи. Такий розподіл дає змогу поєднати відкриту взаємодію з донорами та внутрішній фінансовий контроль [1; 2].

Наведені потоки показують, що FondUA має працювати не з ізольованими записами, а з пов'язаною системою даних. Пожертва повинна бути пов'язана з конкретним благодійним проєктом, витрата - з проєктом, категорією та службовим

статусом, бюджетний план - із плановим періодом і бюджетними рядками, а фінансовий звіт - з перевіреними даними про надходження та витрати. Саме наявність таких зв'язків дає змогу формувати коректні вихідні дані: публічні сторінки фондів і проєктів, інформацію про суму збору, підтверджені витрати, залишок коштів, бюджетні показники, фінансові звіти та аналітичні показники [1].

Ключове значення в інформаційних потоках мають контрольні точки, у яких система перевіряє статус і допустимість використання даних. Не кожен створений запис повинен автоматично впливати на підсумкові фінансові показники. Для FondUA це означає, що до узагальнених результатів мають включатися лише ті записи, які відповідають визначеним правилам обліку: пожертви враховуються згідно з логікою їх статусу, витрати - після перевірки й підтвердження, а звітність формується на основі погоджених даних за відповідний період. Такий підхід забезпечує коректність фінансових підсумків і зменшує ризик викривлення інформації.

Окремо потрібно врахувати обмеження, пов'язані з цільовим використанням коштів, розмежуванням доступу та роботою з персональними даними. Благодійники мають право контролювати цільове використання пожертв, а благодійні організації повинні забезпечувати належну звітність щодо своєї діяльності [3]. Якщо в системі обробляються дані донорів, представників фонду або інших користувачів, така обробка має виконуватися з урахуванням визначеної мети, прозорості та обмеження доступу до службової інформації [4]. Для FondUA це означає потребу у відокремленні публічних і службових даних, фіксації статусів фінансових записів і перевірці прав користувача на кожному етапі роботи з даними.

Постановка задачі кваліфікаційної роботи полягає у проєктуванні вебсистеми FondUA, яка забезпечує зв'язаний маршрут руху даних від фонду й благодійного проєкту до пожертви, витрати, бюджету, фінансового звіту та аналітичного показника. Для розв'язання цієї задачі необхідно визначити основні сутності предметної області та зв'язки між ними, сформувати рольову модель для неавторизованого відвідувача, донора, представника фонду й суперадміністратора, описати правила перевірки та оброблення фінансових записів, передбачити бюджетування, формування звітності та аналітичне подання результатів.

Сформульовані інформаційні потоки створюють основу для переходу до проектування вебсистеми. У розділі 2 на їх основі буде розглянуто архітектуру FondUA, структуру бази даних, зв'язки між основними сутностями та фінансову логіку, необхідну для подальшої програмної реалізації.

Висновки до розділу 1

У першому розділі розглянуто предметну область управління коштами благодійного фонду та визначено, що для такої організації недостатньо мати лише публічний сайт для збору пожертв. Повноцінна цифрова підтримка діяльності фонду потребує цілісної вебсистеми, у якій надходження, витрати, бюджетування, фінансові звіти й аналітичні показники пов'язані з конкретними фондами та благодійними проєктами. Основними вимогами до такого середовища є прозорість фінансових операцій, перевірність даних, контроль статусів, розмежування доступу й зрозуміле подання результатів для різних категорій користувачів.

Огляд сучасних вебрішень показав, що наявні цифрові інструменти здебільшого підтримують лише окремі частини процесу: публічний збір пожертв, подання інформації для донорів, внутрішній облік, приймання платежів або формування аналітичних звітів. Для задачі FondUA принциповим є поєднання цих функцій у межах єдиної системи, де публічна частина забезпечує перегляд фондів, благодійних проєктів і звітів, а службова частина підтримує роботу з пожертвами, витратами, бюджетами, статусами, фінансовими підсумками та аналітичними показниками.

На основі аналізу інформаційних потоків сформульовано постановку задачі проектування вебсистеми FondUA. Система має забезпечувати взаємозв'язок між фондом, благодійним проєктом, жертвою, витратою, бюджетом, фінансовим звітом і аналітичними показниками, а також підтримувати рольову модель для неавторизованого відвідувача, донора, представника фонду та суперадміністратора. Отримані результати створюють основу для розділу 2, у якому буде розроблено архітектуру вебсистеми, структуру бази даних, зв'язки між основними сутностями та фінансову логіку FondUA.

РОЗДІЛ 2

ПРОЄКТУВАННЯ СТРУКТУРИ ТА БАЗИ ДАНИХ ВЕБСИСТЕМИ

2.1 Проєктування загальної архітектури системи

Проєктування загальної архітектури вебсистеми FondUA виконано з урахуванням специфіки благодійного фонду, для якого важливими є не лише публічне представлення благодійних проєктів, а й контроль руху коштів, облік пожертв, фіксація витрат, бюджетування, формування фінансових звітів і подання аналітичних показників. Для такої системи недостатньо створити набір сторінок із переліком проєктів, оскільки кожна фінансова дія має бути пов'язана з конкретним фондом, благодійним проєктом, користувачем, статусом запису та відповідним службовим процесом. Тому FondUA проєктується як цілісна клієнт-серверна вебсистема, у якій клієнтська частина, серверна логіка, функціональні модулі та база даних працюють в одному інформаційному контурі [1; 15].

Архітектура FondUA розглядається на концептуальному рівні як основа для подальшого проєктування структури даних, фінансової логіки та програмної реалізації. У межах цього підпункту визначаються основні рівні системи, ролі користувачів, функціональні зони, загальний напрям руху даних і взаємодія між публічним та службовим контурами. Такий підхід відповідає завданням інформаційного проєктування, оскільки дає змогу встановити зв'язок між предметною областю, інформаційними потоками та майбутньою структурою системи [15].

FondUA побудовано як клієнт-серверну вебсистему. На клієнтському рівні користувач взаємодіє зі сторінками, формами, картками фондів і благодійних проєктів, таблицями, повідомленнями, фінансовими звітами та аналітичними представленнями. Клієнтська частина приймає введені дані, передає їх на сервер і відображає результат оброблення. Основні рішення щодо перевірки прав доступу, валідації форм, збереження записів, формування звітів і підготовки аналітичних показників виконуються на серверному рівні. Така організація є типовою для

вебсистем, у яких браузер виступає засобом взаємодії користувача з прикладною логікою та даними, що обробляються на сервері [13; 14].

Серверна частина FondUA організована навколо Django як вебфреймворку, що підтримує маршрутизацію запитів, оброблення представлень, використання шаблонів, роботу з моделями, формами та механізмами автентифікації користувачів [5]. На концептуальному рівні це дає змогу поділити систему на окремі функціональні модулі, кожен із яких відповідає за свою частину предметної області. Такий поділ зменшує змішування відповідальностей між компонентами, спрощує підтримку системи й створює основу для подальшого розширення функціоналу.

У FondUA доцільно виділити дві взаємопов'язані площини роботи: публічну та службову. Публічна площина призначена для відкритого перегляду фондів, благодійних проєктів, основних фінансових підсумків, опублікованих звітів і аналітичних показників. Вона забезпечує прозоре подання інформації для відвідувачів і донорів. Службова площина призначена для авторизованих користувачів із розширеними правами: представника фонду та суперадміністратора. У ній виконуються дії з керування фондами й проєктами, внесення витрат, бюджетування, підготовки фінансових звітів, перевірки статусів і контролю службових записів.

Рольова модель є одним із ключових елементів архітектури FondUA. Неавторизований відвідувач має доступ лише до відкритих сторінок системи. Донор після автентифікації може працювати зі сценаріями, пов'язаними з пожертвами та переглядом власної взаємодії із системою. Представник фонду працює зі службовою частиною власної організації: створює та редагує інформацію про фонд, додає благодійні проєкти, вносить витрати, формує бюджетні плани й готує фінансові звіти. Суперадміністратор контролює користувачів, статуси фондів, службові записи, витрати, звіти та коректність даних. Такий поділ дозволяє узгодити користувацькі сценарії з функціональними зонами системи та обмежити доступ до операцій, які впливають на фінансові показники.

Архітектурно важливо, що відвідувач не є окремою роллю в моделі профілю користувача, а описує неавторизований стан доступу до відкритої частини системи. Ролі авторизованих користувачів використовуються для керування доступом до службових дій. Такий підхід дає змогу відокремити публічний перегляд інформації від операцій, що змінюють фінансові або службові записи.

Функціональна структура FondUA відповідає основним процесам управління коштами благодійного фонду. Модуль accounts забезпечує роботу з користувачами, профілями, автентифікацією та ролями. Модуль funds відповідає за дані благодійних фондів, статус фонду та зв'язок фонду з його представником. Модуль projects охоплює благодійні проєкти, їхній опис, статус, цільову суму та публічне подання. Модуль finance пов'язаний із пожертвами, витратами, категоріями витрат, бюджетними планами та бюджетними рядками. Модуль reports відповідає за створення, перевірку, публікацію та перегляд фінансових звітів. Модуль analytics забезпечує підготовку аналітичних показників на основі фінансових даних. Модуль core використовується для базових сторінок і загальної логіки вебсистеми.

У межах архітектури системи фінансові записи не розглядаються як ізольовані сутності. Пожертва має бути пов'язана з конкретним благодійним проєктом; витрата - з проєктом, категорією та статусом перевірки; бюджетний план - з проєктом і бюджетними рядками; фінансовий звіт - з фондом або проєктом і визначеним звітним періодом. Завдяки цьому система може формувати зібрану суму, підтверджені витрати, залишок коштів, план-фактне порівняння та аналітичні показники на основі структурованих даних, а не через ручне зіставлення окремих таблиць або документів.

Окреме місце в архітектурі займає контур статусів. Статус фонду визначає, чи може він бути доступним у публічній частині системи. Статус пожертви впливає на те, чи буде вона врахована у фінансових підсумках. Статус витрати визначає, чи зменшує вона залишок коштів проєкту. Статус фінансового звіту визначає, чи є звіт чернеткою, поданим на перевірку, погодженим, відхиленням або опублікованим. Така логіка потрібна для того, щоб створення запису не ототожнювалося з його фінансовим визнанням або публічним відображенням.

Загальний потік даних у FondUA будується за послідовною схемою. Користувач відкриває сторінку або надсилає форму через клієнтський інтерфейс. Серверна частина приймає запит, визначає відповідний маршрут і функціональний модуль, перевіряє автентифікацію користувача, його роль і допустимість дії. Після цього дані читаються з бази або змінюються в ній через Django ORM, а користувачу повертається HTML-сторінка, повідомлення про результат, оновлений список записів, фінансовий звіт або аналітичне представлення. Така схема дає змогу відокремити інтерфейс від прикладної логіки та рівня даних.

Загальну архітектуру вебсистеми FondUA, побудовану за клієнт-серверним принципом із використанням Django, Django ORM і SQLite, подано на рис. 2.1.

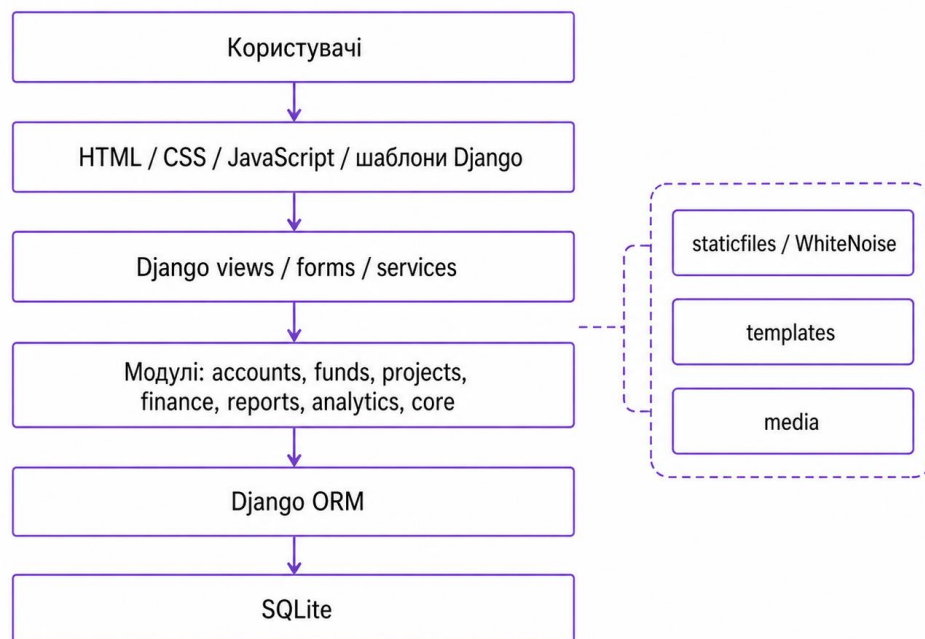


Рис. 2.1. Загальна архітектура вебсистеми FondUA

(Розроблено автором)

База даних у цій архітектурі виконує роль центрального сховища пов'язаних записів. У ній зберігаються користувачі, профілі, фонди, благодійні проекти, пожертви, витрати, категорії витрат, бюджетні плани, бюджетні рядки та фінансові звіти. Серверна частина не працює з цими даними як із довільним набором записів,

а використовує зв'язки між моделями. Саме ці зв'язки забезпечують коректне формування фінансових показників, звітів і аналітики.

На рис. 2.1 показано взаємодію користувача з клієнтським рівнем, серверною частиною Django, функціональними модулями та базою даних. Клієнтська частина відповідає за роботу зі сторінками, формами й таблицями, а серверна частина обробляє запити, перевіряє права доступу, звертається до моделей і формує відповідь. Функціональні модулі accounts, funds, projects, finance, reports, analytics і core відокремлюють різні частини предметної області, що спрощує підтримку та розширення системи. За рахунок такого поділу FondUA підтримує публічний перегляд благодійної інформації та службове управління фінансовими процесами в межах одного інформаційного середовища.

Отже, загальна архітектура FondUA поєднує клієнтський рівень, серверну частину Django, функціональні модулі, базу даних, контур рольового доступу, механізм статусів, фінансову звітність і аналітику. У цьому підпункті визначено концептуальну побудову системи та її основні функціональні зони. У підпункті 2.2 цю архітектуру деталізовано через структуру даних, основні сутності та зв'язки між ними, а в підпункті 2.3 розглянуто фінансову логіку й аналітичні показники вебсистеми FondUA.

2.2 Моделювання бази даних для обліку фондів, проєктів і фінансових операцій

Структура бази даних вебсистеми FondUA є основою для збереження, перевірки та подальшого опрацювання даних про користувачів, благодійні фонди, проєкти, пожертви, витрати, бюджетні плани, бюджетні рядки та фінансові звіти. Для системи управління коштами благодійного фонду база даних не може складатися з ізольованих таблиць, оскільки кожна фінансова операція має бути пов'язана з конкретним проєктом, фондом, користувачем, категорією витрат або звітним періодом. Через це під час проєктування структури даних FondUA використано реляційну модель, у якій сутності предметної області поєднуються

через первинні та зовнішні ключі [1; 15]. Такий підхід дає змогу уникнути дублювання інформації, зберігати логічні зв'язки між записами та забезпечувати коректність фінансових розрахунків. Крім цього, реляційна структура спрощує формування звітів, фільтрацію даних за фондами, проектами й періодами, а також подальше розширення системи без повної зміни її базової логіки.

Інформаційна модель FondUA охоплює чотири взаємопов'язані блоки: користувацький, організаційний, фінансовий та звітно-аналітичний. Користувацький блок містить обліковий запис користувача та його профіль, у якому зберігаються основні дані, роль у системі та інформація, потрібна для взаємодії з благодійними фондами й проектами. Організаційний блок описує благодійний фонд і благодійні проекти, що дає змогу відокремити загальні дані про фонд від конкретних ініціатив, на які спрямовуються кошти. Фінансовий блок включає пожертви, витрати, категорії витрат, бюджетні плани та бюджетні рядки, тому саме він відповідає за фіксацію руху коштів і контроль їх використання. Звітно-аналітичний блок пов'язаний із фінансовими звітами й аналітичними показниками, що формуються на основі підтверджених пожертв, погоджених витрат і бюджетних даних. Завдяки такому поділу модель залишається зрозумілою, логічною та придатною для подальшого розширення функціоналу системи.

Базовою користувацькою сутністю є User, що реалізується стандартною моделлю Django та зберігає дані, потрібні для автентифікації. Розширені відомості винесено в модель Profile, яка пов'язана з User зв'язком «один до одного». У профілі зберігаються роль користувача, контактний номер, дата створення та дата оновлення запису. У моделі Profile передбачено три ролі: donor, fund_manager і super_admin. Відвідувач не є роллю моделі Profile, оскільки це не окремий запис у базі даних, а неавторизований стан доступу до публічних сторінок системи. Представник фонду також не виділяється в окрему таблицю: його статус визначається поєднанням ролі fund_manager у профілі та зв'язку Fund.owner з обліковим записом користувача [5]. Основні ролі та обмеження доступу наведено в табл. 2.1.

Таблиця 2.1

Ролі користувачів і права доступу у вебсистемі FondUA

Роль / статус доступу	Представлення у системі	Права доступу	Модулі	Обмеження
1	2	3	4	5
Відвідувач	Неавторизований користувач; запис у Profile не створюється	Перегляд публічних фондів, публічних проєктів, відкритих фінансових звітів і публічної аналітики; створення пожертви через публічну форму	core, funds, projects, finance, reports, analytics	Не має особистого кабінету; не може керувати фондом, проєктами, витратами, бюджетами чи звітами; не має доступу до службових фінансових даних
Донор	Авторизований користувач із Profile.role = donor	Перегляд особистого кабінету; створення пожертв; перегляд власних пожертв; перегляд публічних фондів, проєктів, звітів і відкритої аналітики	accounts, finance, funds, projects, reports, analytics	Не може створювати або редагувати фонд, проєкт, витрату, бюджетний план чи фінансовий звіт; не має доступу до аналітики фонду або системної аналітики
Представник фонду	Авторизований користувач із Profile.role = fund_manager, пов'язаний із фондом через Fund.owner	Створення та редагування власного фонду; перегляд кабінету фонду; створення, редагування, архівування та відновлення власних проєктів; перегляд пожертв свого фонду; створення витрат; створення бюджетів; підготовка та подання фінансових звітів; перегляд аналітики власного фонду	accounts, funds, projects, finance, reports, analytics	Працює лише в межах власного фонду; не підтверджує і не відхиляє витрати; не підтверджує, не відхиляє і не публікує звіти; не має системного огляду всіх фондів

Продовження табл. 2.1

	2	3	4	5
Суперадміністратор	Авторизований користувач із Profile.role = super_admin або користувач із прапорами Django is_staff / is_superuser	Перегляд системного огляду; керування фондами, проектами, пожертвами, витратами, бюджетами та звітами; підтвердження і відхилення витрат; підтвердження, відхилення, публікація та зняття з публікації фінансових звітів; перегляд системної аналітики	accounts, funds, projects, finance, reports, analytics, адміністративний контур Django	Не обмежується одним фондом, але всі дії виконуються відповідно до правил валідації та бізнес-логіки системи

(Розроблено автором)

Для розмежування доступу в системі використано рольову модель, у якій права користувача залежать від стану авторизації, значення поля Profile.role та зв'язку користувача з фондом.

Сутність Fund описує благодійний фонд як організаційну одиницю системи. Для фонду зберігаються назва, slug, опис, електронна адреса, телефон, вебсайт, логотип, ознака розміщення на головній сторінці, дата створення та дата оновлення. Ключовим атрибутом є поле status, яке визначає стан фонду в системі. Значення pending означає, що фонд створено і він очікує адміністративної перевірки. Значення approved означає, що фонд підтверджено і він може відображатися у відкритих модулях системи. Значення blocked використовується для фонду, доступність якого обмежено. Зв'язок Fund.owner реалізовано як зв'язок «один до одного», тому один користувач може представляти не більше одного фонду.

Сутність Project описує благодійний проєкт, що належить конкретному фонду. Для проєкту зберігаються назва, slug, категорія, короткий опис, повний опис, цільова сума збору, зображення, статус, дати початку й завершення, ознака відображення на головній сторінці, дата створення та дата оновлення. Проєкт

пов'язаний із фондом зв'язком «багато до одного». У моделі передбачено статуси draft, active, completed і archived, що дає змогу розділяти чернетки, активні збори, завершені проєкти та архівні записи. Публічне відображення проєкту залежить не лише від його власного статусу, а й від статусу фонду, до якого він належить.

Центральною фінансовою сутністю є Donation, яка фіксує факт пожертви на конкретний благодійний проєкт. Модель пожертви містить посилання на проєкт, необов'язкове посилання на користувача-донора, ім'я донора, електронну адресу донора, суму, валюту, спосіб оплати, короткий опис способу оплати, статус пожертви, унікальний ідентифікатор транзакції, коментар, ознаку анонімності, дату створення та дату оновлення. Поле payment_method фіксує обраний спосіб внесення коштів і підтримує значення card, paypal, google_pay, bank_transfer, cash і test. Поле is_anonymous визначає, чи повинна пожертва відображатися як анонімна. Поле transaction_id зберігає унікальний ідентифікатор транзакції, який генерується автоматично, якщо його не передано під час створення запису. Поле payment_details_summary містить короткий текстовий опис платіжних реквізитів або службову позначку демонстраційного платежу. Якщо пожертву створено без авторизації, зв'язок із User може бути відсутній, але зв'язок із проєктом є обов'язковим.

Для обліку витрат використано сутності ExpenseCategory і Expense. Модель ExpenseCategory є довідником категорій витрат і містить назву, опис та дату створення. Модель Expense пов'язана з конкретним проєктом і категорією витрат. У ній зберігаються назва витрати, сума, опис, документальне підтвердження, статус перевірки, дата витрати, користувач, який створив запис, користувач, який виконав перевірку, дата перевірки, причина відхилення, дата створення та дата оновлення. Статуси витрати реалізовано значеннями pending, approved і rejected. На фінансові показники проєкту впливають лише витрати зі статусом approved, оскільки непогоджені або відхилені записи не повинні зменшувати баланс проєкту.

Бюджетування у FondUA представлено сутностями BudgetPlan і BudgetLine. BudgetPlan описує бюджет проєкту в межах визначеного періоду та містить посилання на проєкт, назву бюджету, дату початку й дату завершення періоду,

опис, користувача-автора, дату створення та дату оновлення. BudgetLine належить конкретному бюджетному плану й пов'язується з категорією витрат. Для бюджетного рядка зберігаються категорія, планова сума та опис. Обмеження унікальності для пари budget і category унеможливило дублювання одного напрямку витрат у межах одного бюджетного плану. Фактичне виконання бюджету не зберігається як окреме фізичне поле в таблиці BudgetLine, а обчислюється на основі погоджених витрат відповідної категорії за період бюджету. Показники planned_amount, actual_amount, difference_amount і usage_percent є похідними значеннями, що використовуються для план-фактного порівняння.

Сутність FinancialReport реалізує фінансовий звіт фонду або конкретного благодійного проєкту. Вона містить посилання на фонд, необов'язкове посилання на проєкт, назву звіту, межі звітного періоду, загальні надходження, загальні витрати, залишок коштів, кількість пожертв, кількість витрат, короткий опис, файл документа, статус, ознаку публічності, автора звіту, користувача, який погодив звіт, дату погодження, дату публікації, причину відхилення, дату створення та дату оновлення. Статуси звіту реалізовано значеннями draft, submitted, approved, rejected і published. Наявність необов'язкового зв'язку з проєктом дає змогу формувати як загальний звіт по фонду, так і звіт по окремому благодійному проєкту.

Аналітичні показники у FondUA не виділено в окрему фізичну таблицю бази даних. Вони формуються на сервісному рівні модуля analytics на основі даних моделей Project, Donation, Expense, BudgetPlan і FinancialReport. Такий підхід дає змогу уникнути дублювання агрегованих значень у базі даних і підтримувати узгодженість аналітики з первинними фінансовими записами.

У програмній реалізації FondUA описані сутності відповідають моделям Django, розміщеним у функціональних додатках системи. Модель Profile реалізовано в додатку accounts, Fund - у додатку funds, Project - у додатку projects, Donation, ExpenseCategory, Expense, BudgetPlan і BudgetLine - у додатку finance, а FinancialReport - у додатку reports. Такий розподіл забезпечує відокремлення користувацької, організаційної, фінансової та звітної логіки. Основні сутності

інформаційної моделі FondUA, їх призначення, зв'язки та програмну реалізацію наведено в табл. 2.2.

Таблиця 2.2.

Перелік основних сутностей та зв'язків FondUA

Сутність	Призначення	Основні зв'язки	Реалізація у FondUA
1	2	3	4
User	Базовий обліковий запис користувача	один Profile може бути власником одного Fund; може бути донором, автором витрати, автором бюджету, автором звіту	Стандартна модель django.contrib.auth.models.User; для автентифікації та як базова сутність доступу
Profile	Розширені дані користувача та його роль у системі	Пов'язаний з User зв'язком «один до одного»	Модель Profile містить role, phone, created_at, updated_at; підтримує ролі donor, fund_manager, super_admin
Fund	Опис благодійного фонду	Пов'язаний з User через owner; має багато Project; має багато FinancialReport	Модель Fund містить name, slug, description, email, phone, website, logo, status, is_featured, created_at, updated_at; підтримує стани pending, approved, blocked
Project	Опис благодійного проєкту та його фінансового контексту	Належить Fund; має багато Donation, Expense, BudgetPlan; може мати FinancialReport	Модель Project у додатку projects; містить title, slug, category, short_description, description, goal_amount, image, status, start_date, end_date, is_featured, created_at, updated_at; має обчислювані показники collected_amount, approved_expenses_amount, balance, progress_percent
Donation	Фіксація пожертви на проєкт	Пов'язана з Project; може бути пов'язана з User як донором	Модель Donation у додатку finance; містить donor_name, donor_email, amount, currency, payment_method, payment_details_summary, status, transaction_id, comment, is_anonymous, created_at, updated_at; підтримує стани pending, completed, cancelled, refunded

Продовження табл. 2.2

1	2	3	4
Donation	Фіксація пожертви на проєкт	Пов'язана з Project; може бути пов'язана з User як донором	Модель Donation у додатку finance; містить donor_name, donor_email, amount, currency, payment_method, payment_details_summary, status, transaction_id, comment, is_anonymous, created_at, updated_at; підтримує стани pending, completed, cancelled, refunded
ExpenseCategory	Довідник категорій витрат	Має багато Expense; використовується в BudgetLine	Модель ExpenseCategory у додатку finance; містить name, description, created_at
Expense	Фіксація витрати за проєктом	Пов'язана з Project і ExpenseCategory; містить відомості про автора й перевірку	Модель Expense у додатку finance; містить title, amount, description, document, status, spent_at, created_by, verified_by, verified_at, rejection_reason, created_at, updated_at; підтримує стани pending, approved, rejected
BudgetPlan	Бюджет проєкту на визначений період	Пов'язаний з Project; має багато BudgetLine	Модель BudgetPlan у додатку finance; містить title, period_start, period_end, description, created_by, created_at, updated_at; має обчислювані показники planned_amount, actual_amount, difference_amount, usage_percent
BudgetLine	Окремий напрям витрат у межах бюджету	Пов'язаний з BudgetPlan і ExpenseCategory	Модель BudgetLine у додатку finance; містить budget, category, planned_amount, description; має обмеження унікальності для пари budget і category; показники actual_amount, difference_amount, usage_percent обчислюються на основі погоджених витрат
FinancialReport	Узагальнений фінансовий звіт фонду або проєкту	Пов'язаний з Fund; може бути пов'язаний з Project	Модель FinancialReport у додатку reports; містить title, period_start, period_end, total_income, total_expenses, balance, donations_count, expenses_count, summary, document, status, is_public, created_by, approved_by, approved_at, published_at, rejection_reason, created_at, updated_at; підтримує стани draft, submitted, approved, rejected, published

(Розроблено автором)

Для пояснення одного з ключових фінансових сценаріїв використано UML-діаграму послідовності внесення пожертви. Вона показує взаємодію між донором, формою пожертви, Django-представленням, ORM-рівнем і базою даних SQLite під час створення запису Donation. Діаграму послідовності внесення пожертви у веб-системі FondUA наведено на рис. 2.2.

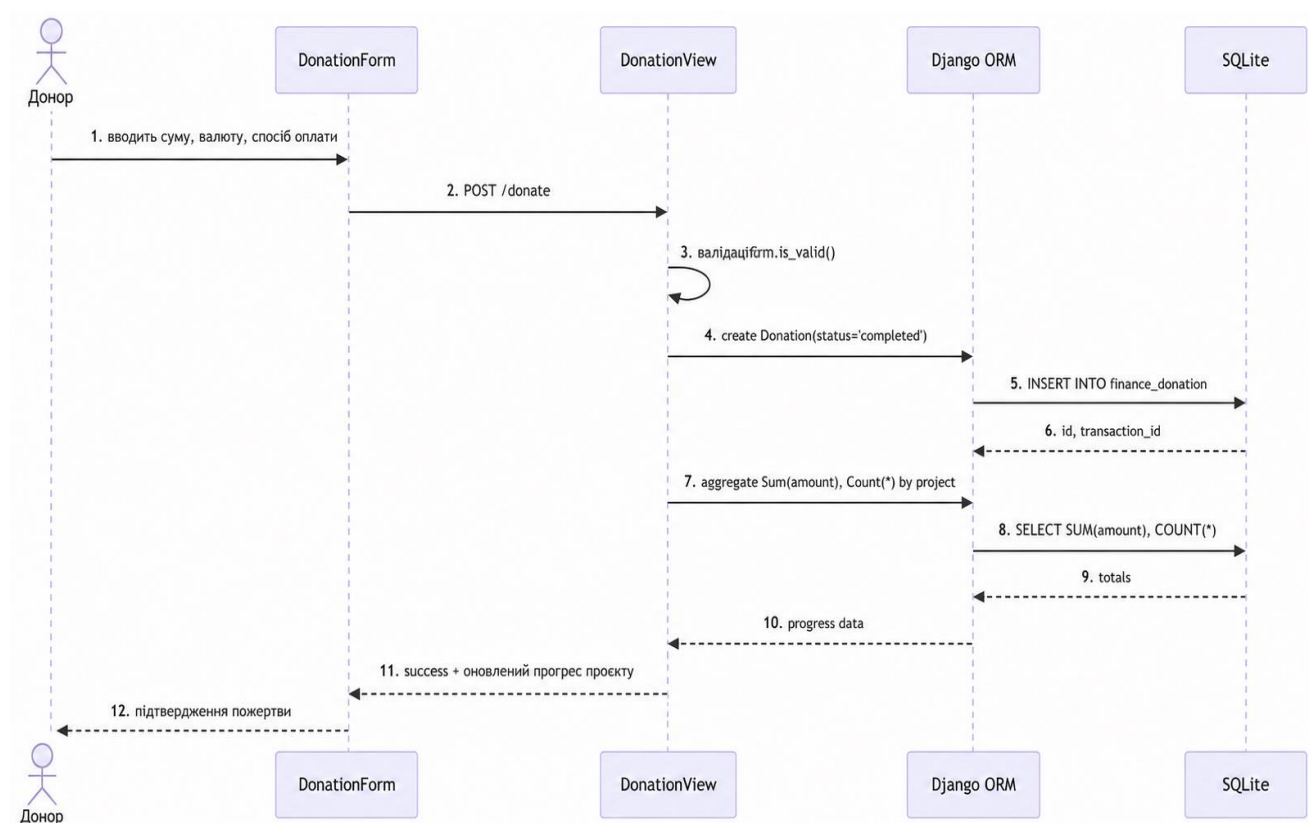


Рис. 2.2. UML-діаграма послідовності внесення пожертви у веб-системі

(Розроблено автором)

На рис. 2.2 показано, що користувач вводить суму, валюту та спосіб оплати у форму пожертви, після чого дані передаються до Django-представлення. Серверна частина виконує перевірку введених значень, створює запис Donation через Django ORM, зберігає його в базі даних SQLite і повертає користувачу повідомлення про результат операції. Фінансові показники проекту після цього визначаються не через дублювання окремих постійних полів, а через обчислювані властивості моделі Project, зокрема `collected_amount`, `approved_expenses_amount`, `balance` і `progress_percent`.

Структуру бази даних FondUA та зв'язки між основними сутностями подано на рис. 2.3.

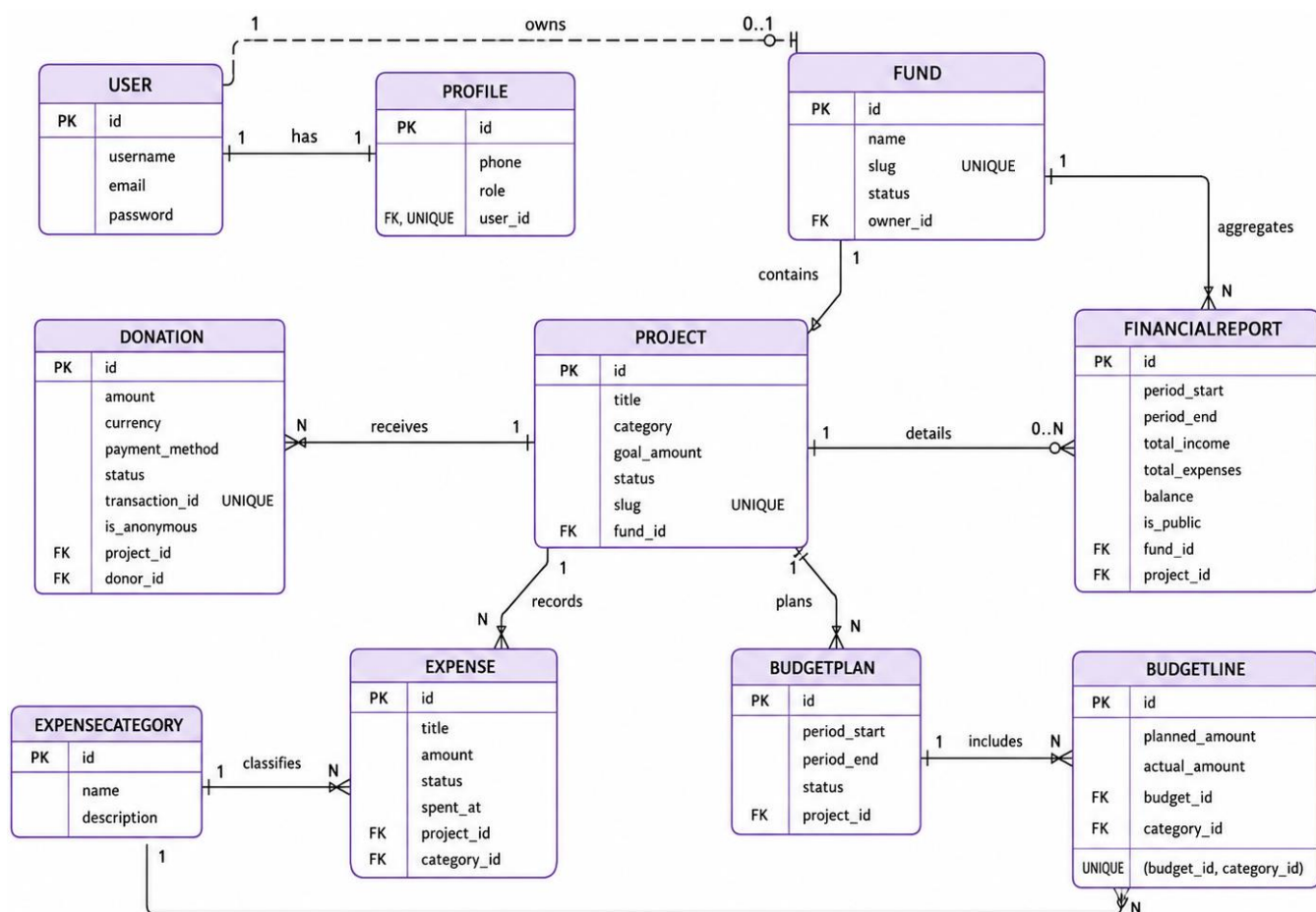


Рис. 2.3. ER-діаграма бази даних вебсистеми FondUA

(Розроблено автором)

ER-діаграма бази даних показує основні сутності, їх атрибути, первинні та зовнішні ключі, а також кардинальність зв'язків між таблицями. У моделі FondUA користувач має один профіль; фонд пов'язується з користувачем-представником; фонд містить багато благодійних проєктів; проєкт отримує багато пожертв, має витрати, бюджетні плани та може бути пов'язаний із фінансовими звітами. Витрата належить конкретному проєкту та класифікується через категорію витрат. Бюджетний рядок належить бюджетному плану й пов'язується з категорією витрат. Фінансовий звіт пов'язується з фондом або проєктом залежно від рівня узагальнення звітних даних.

Для забезпечення цілісності даних у структурі бази передбачено первинні ключі, зовнішні ключі, обов'язкові зв'язки, унікальні службові поля та допустимі статуси записів. Проєкт повинен бути пов'язаний із фондом, пожертва - з конкретним проєктом, витрата - з проєктом і категорією витрат, бюджетний рядок - із бюджетним планом і категорією, а фінансовий звіт - із фондом або фондом і конкретним проєктом. Для уникнення дублювання контролюється унікальність окремих службових значень, зокрема slug, transaction_id і пари budget / category у бюджетних рядках. Реляційна модель збереження даних і механізми обмежень цілісності підтримуються SQLite, що використовується в навчально-практичній версії FondUA як база даних [6].

Окрему увагу в структурі бази даних приділено даним користувачів і донорів. Профілі, контактні відомості та записи, пов'язані з пожертвами, можуть містити персональну інформацію, тому доступ до них має обмежуватися відповідно до ролі користувача. Відкриті сторінки системи не повинні розкривати службові дані донорів, а режим анонімної пожертви дає змогу приховати ім'я донора під час відображення запису. Такий поділ відкритих і службових даних узгоджується з вимогами до обробки персональної інформації [4].

Отже, структура бази даних FondUA відображає предметну область управління коштами благодійного фонду через систему пов'язаних сутностей. Вона забезпечує основу для обліку пожертв, витрат, бюджетних планів і фінансових звітів, а також для формування аналітичних показників без дублювання агрегованих даних у фізичних таблицях. У підпункті 2.3 на основі цієї структури розглянуто фінансову логіку системи та правила розрахунку показників, що характеризують надходження, витрати, залишок коштів і виконання бюджету.

2.3 Розроблення фінансової логіки та аналітичних показників системи

Фінансова логіка вебсистеми FondUA побудована навколо повного циклу управління коштами благодійного фонду: створення благодійного проєкту, надходження пожертв, фіксації витрат, планування бюджету, розрахунку залишку

коштів, підготовки фінансового звіту та формування аналітичних показників. Для такої системи недостатньо лише зберігати окремі записи в базі даних, оскільки важливо визначити, які кошти реально зібрано, які витрати погоджено, який залишок доступний і наскільки фактичне використання коштів відповідає запланованому бюджету [1; 15].

Основними вхідними даними для фінансових розрахунків є сума пожертви, статус пожертви, дата пожертви, благодійний проєкт, до якого вона належить, сума витрати, категорія витрати, статус витрати, дата витрати, бюджетний період, планова сума бюджетного рядка та звітний період. Вихідними даними є зібрана сума, підтверджені витрати, залишок коштів, відсоток виконання збору, залишок до цільової суми, планова сума бюджету, фактична сума витрат за бюджетом, відхилення плану від факту, підсумки фінансового звіту та дані для аналітичного подання.

Фінансові розрахунки FondUA виконуються з урахуванням статусів записів. До показника зібраних коштів не повинні входити всі створені пожертви, оскільки частина з них може перебувати в очікуванні, бути скасованою або поверненою. До фінансового результату включаються лише пожертви зі статусом *completed*, який підтверджує завершення операції. Аналогічно не всі витрати мають зменшувати залишок коштів. Витрати зі статусом *pending* або *rejected* не повинні впливати на баланс проєкту; для розрахунку використовується лише витрата зі статусом *approved*. Схему формування фінансових показників вебсистеми FondUA наведено на рис. 2.4.

Схема показує, що фінансові показники формуються не безпосередньо з усіх записів, а через відбір коректних операцій за статусами та зв'язками з конкретним благодійним проєктом. Пожертви формують обсяг зібраних коштів, погоджені витрати формують обсяг використаних коштів, різниця між ними визначає залишок, а бюджетні рядки дають змогу зіставити планові та фактичні витрати. У підсумку фінансовий звіт отримує агреговані значення, підготовлені за єдиними правилами.

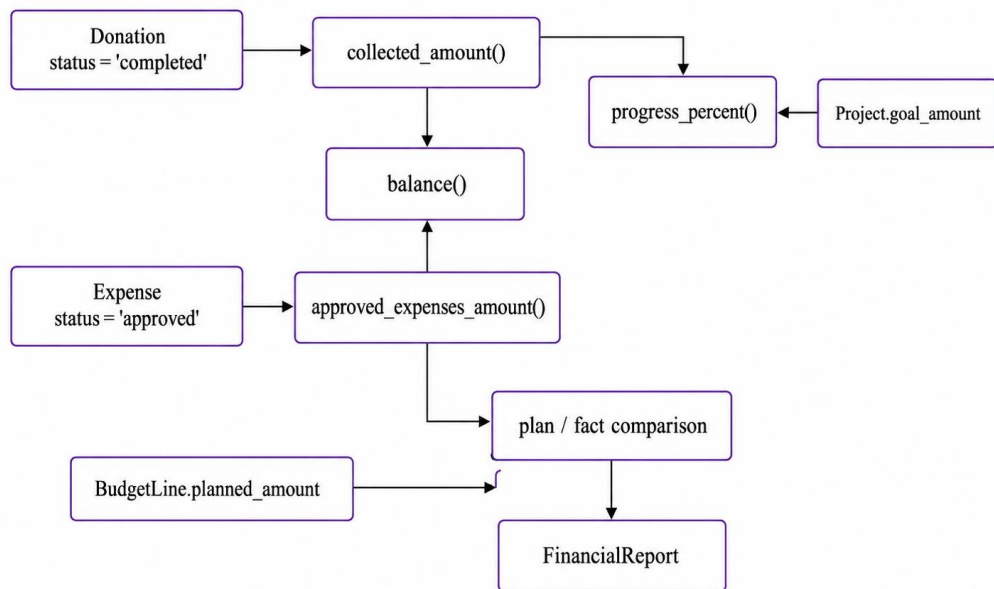


Рис. 2.4. Схема формування фінансових показників вебсистеми FondUA

(Розроблено автором)

Основні фінансові та аналітичні показники вебсистеми FondUA наведено в табл. 2.3.

Таблиця 2.3

Основні фінансові та аналітичні показники вебсистеми FondUA

Показник	Призначення	Джерело даних	Формула
1	2	3	4
collected_amount	сума зібраних коштів	finance_donation	SUM(amount), де status = completed
approved_expenses_amount	підтверджені витрати	finance_expense	SUM(amount), де status = approved
balance	залишок коштів проєкту	пожертви та витрати	collected_amount - approved_expenses_amount
progress_percent	відсоток виконання фінансової цілі проєкту	проєкт і пожертви	progress_percent = min(100, (collected_amount / goal_amount) · 100), якщо goal_amount > 0
planned_amount	планова сума бюджету	finance_budgetline	сума планових рядків

Продовження табл. 2.3.

1	2	3	4
BudgetPlan.actual_amount	фактична сума витрат за бюджетним планом	finance_expense , finance_budgetplan	SUM(amount), де expense.project = budget.project, expense.status = approved та expense.spent_at належить до періоду бюджету
BudgetLine.actual_amount	фактична сума витрат за окремим бюджетним рядком	finance_expense , finance_budgetline	SUM(amount), де expense.project = budget.project, expense.category = budgetline.category, expense.status = approved та expense.spent_at належить до періоду бюджету
difference_amount	відхилення плану від факту	бюджет і витрати	planned_amount - actual_amount
report_balance	підсумковий залишок у звіті	reports_financial_report	total_income - total_expenses

(Розроблено автором)

Для формалізації фінансової логіки FondUA використано такі розрахункові залежності:

$\text{CollectedAmount}(p) = \sum d.\text{amount}$, де $d.\text{project} = p$ та $d.\text{status} = \text{'completed'}$.

$\text{ApprovedExpenses}(p) = \sum e.\text{amount}$, де $e.\text{project} = p$ та $e.\text{status} = \text{'approved'}$.

$\text{Balance}(p) = \text{CollectedAmount}(p) - \text{ApprovedExpenses}(p)$.

$\text{ProgressPercent}(p) = \min(100, \text{CollectedAmount}(p) / \text{GoalAmount}(p) \times 100)$, якщо $\text{GoalAmount}(p) > 0$.

$\text{PlannedAmount}(b) = \sum bl.\text{planned_amount}$, де $bl.\text{budget} = b$.

$\text{ActualAmount}(b, c) = \sum e.\text{amount}$, де $e.\text{project} = b.\text{project}$, $e.\text{category} = c$, $e.\text{status} = \text{'approved'}$, $e.\text{spent_at}$ належить до періоду бюджету b .

$\text{DifferenceAmount}(bl) = bl.\text{planned_amount} - \text{ActualAmount}(bl.\text{budget}, bl.\text{category})$.

$\text{DifferenceAmount}(\text{bl}) = \text{bl.planned_amount} - \text{ActualAmount}(\text{bl.budget}, \text{bl.category}).$

$\text{ReportBalance}(\text{r}) = \text{TotalIncome}(\text{r}) - \text{TotalExpenses}(\text{r}).$

У наведених формулах *p* позначає благодійний проєкт, *d* - запис пожертви, *e* - запис витрати, *b* - бюджетний план, *bl* - бюджетний рядок, *r* - фінансовий звіт. Показник *BudgetPlanActualAmount(b)* використовується для оцінки фактичних витрат за всім бюджетним планом, а *BudgetLineActualAmount(bl)* - для розрахунку фактичної суми за окремою категорією витрат у межах бюджетного рядка. Використання статусів *completed* та *approved* потрібне для того, щоб до підсумкових фінансових показників не потрапляли непідтверджені пожертви або витрати, які ще не пройшли перевірку.

Показник *collected_amount* характеризує суму завершених пожертв за конкретним благодійним проєктом. Його призначення полягає у визначенні реального обсягу коштів, які можуть бути враховані у прогресі збору та подальших фінансових розрахунках. Показник *approved_expenses_amount* відображає суму витрат, що пройшли погодження та можуть бути включені до фінансового результату. Витрати, які ще очікують перевірки або були відхилені, не повинні зменшувати залишок коштів, оскільки вони не мають статусу підтвердженої фінансової операції.

Показник *balance* визначає залишок коштів проєкту як різницю між підтвердженими надходженнями та погодженими витратами. Якщо значення є додатним, кошти залишаються доступними для подальшого використання. Якщо баланс наближається до нуля, це свідчить про майже повне використання зібраних коштів. Від'ємний баланс потребує перевірки коректності фінансових записів, оскільки він може вказувати на витрати, що перевищують підтверджені надходження.

Окремим правилом *FondUA* є перевірка доступного балансу під час погодження витрати. Створення витрати ще не означає її включення до фінансового результату. Коли витрата переходить у статус *approved*, система має перевірити, чи не перевищує її сума доступний залишок коштів проєкту. Якщо сума

витрати більша за доступний баланс, така витрата не повинна бути погоджена без додаткового коригування фінансових даних. Це правило захищає систему від ситуації, у якій підтверджені витрати перевищують суму завершених пожертв.

Показник `progress_percent` використовується для оцінки виконання цільової суми благодійного проєкту. Він розраховується як відношення зібраної суми до цільової суми проєкту, помножене на 100. Якщо цільова сума дорівнює нулю або не задана, стандартне ділення застосовувати не можна, оскільки це призведе до некоректного розрахунку. Для публічного відображення значення `progress_percent` обмежується верхньою межею 100 %, щоб перевиконання збору не спотворювало інтерпретацію прогресу.

Бюджетна логіка `FondUA` побудована за принципом план-фактного порівняння. `BudgetPlan` визначає бюджетний період або загальний план використання коштів за проєктом, а `BudgetLine` деталізує план за категоріями витрат. Через це потрібно розрізняти два рівні фактичної суми. Для `BudgetPlan` показник `actual_amount` формується як сума всіх погоджених витрат проєкту в межах періоду бюджету. Для `BudgetLine` показник `actual_amount` формується лише за тією категорією витрат, яка відповідає конкретному бюджетному рядку. Таке розмежування потрібне для коректного аналізу як усього бюджету проєкту, так і окремих напрямів витрат.

Показник `difference_amount` визначається як різниця між плановою та фактичною сумою. Якщо значення додатне, це означає, що за відповідним бюджетом або категорією залишаються невикористані планові кошти. Якщо значення від'ємне, це свідчить про перевищення фактичних витрат над планом. Показник `usage_percent` показує відсоток використання бюджету і дає змогу швидко оцінити, наскільки фактичні витрати наближені до запланованих значень [1; 15].

Окремий блок фінансової логіки пов'язаний із формуванням фінансових звітів. Фінансовий звіт має містити фонд, необов'язковий зв'язок із конкретним благодійним проєктом, межі звітного періоду, статус, підсумкові надходження, підсумкові витрати, залишок коштів, кількість пожертв і кількість витрат. Дані

звіту мають формуватися на основі тих самих правил, що й аналітичні показники: враховуються лише завершені пожертви, погоджені витрати та записи, які належать до відповідного звітного періоду. Для пожертв період визначається за датою створення запису, а для витрат - за датою витрати.

Статус фінансового звіту визначає, чи може він бути використаний для публічного подання. Чернетка або звіт, поданий на перевірку, є службовим записом. Публічне відображення звіту доцільне лише після його погодження або публікації, оскільки донор має бачити не проміжні службові дані, а узгоджений фінансовий підсумок [2; 3].

Аналітичні показники у FondUA формуються на основі пов'язаних фінансових сутностей: Fund, Project, Donation, Expense, BudgetPlan, BudgetLine і FinancialReport. Вони не потребують окремої фізичної таблиці бази даних, оскільки можуть обчислюватися через агрегування вже наявних записів. Для узгодженості системи аналітичні показники мають використовувати ті самі правила відбору даних, що й фінансові звіти. Це дає змогу уникнути ситуації, коли на аналітичній панелі відображається один фінансовий результат, а у звіті - інший.

Під час розрахунків потрібно враховувати кілька обмежень. Розрахунки мають виконуватися в межах конкретного благодійного проєкту або фонду. Фінансові операції без коректного зв'язку з проєктом не повинні включатися до проєктних показників. Непідтверджені пожертви не мають збільшувати публічну суму збору, а непогоджені витрати не повинні зменшувати залишок. Бюджетний факт має формуватися з урахуванням періоду, категорії та статусу витрати. Дані фінансового звіту мають відповідати обраному звітному періоду. Виконання цих правил забезпечує узгодженість між обліком пожертв, контролем витрат, бюджетуванням, звітністю та аналітикою.

Фінансова логіка FondUA спирається на пов'язану модель сутностей, статусів і правил агрегування. Пожертви формують дохідну частину, витрати - видаткову, бюджетні рядки задають план, а фінансовий звіт фіксує підсумок за період. Завдяки цьому FondUA може використовуватися як інструмент контролю руху коштів, оцінки виконання зборів, перевірки план-фактних відхилень і

підготовки прозорої фінансової звітності. Отримані правила фінансової логіки завершують інформаційне та математичне проектування системи й створюють основу для переходу до опису програмної реалізації.

Висновки до розділу 2

У другому розділі сформовано інформаційну та математичну основу вебсистеми FondUA. На концептуальному рівні визначено загальну архітектуру системи, її публічну й службову площини, клієнт-серверну організацію, функціональні модулі та рольовий доступ для відвідувача, донора, представника фонду й суперадміністратора. Це дало змогу пов'язати користувацькі сценарії з основними функціональними зонами системи.

У межах моделювання бази даних визначено основні сутності FondUA: користувач, профіль користувача, фонд, благодійний проєкт, пожертва, витрата, категорія витрат, бюджетний план, бюджетний рядок і фінансовий звіт. Для цих сутностей описано призначення, основні поля, зв'язки та статуси, що впливають на доступність записів і фінансові розрахунки. Окремо уточнено, що відвідувач є неавторизованим станом доступу, а не окремою роллю в моделі профілю.

Розроблено фінансову логіку системи, яка визначає правила розрахунку зібраних коштів, погоджених витрат, залишку коштів, прогресу збору, доступного балансу, план-фактного виконання бюджету та підсумків фінансового звіту. У розрахунках враховано статуси пожертв, витрат і звітів, щоб до публічних і звітних показників потрапляли лише коректні та підтверджені фінансові записи.

Отримані результати створюють основу для програмної реалізації FondUA. Спроектowana архітектура, структура даних і фінансова логіка дають змогу перейти до опису серверної частини, клієнтського інтерфейсу, аналітичної панелі, тестування та результатів роботи вебсистеми в розділі 3.

РОЗДІЛ 3

РОЗРОБКА ВЕБСИСТЕМИ УПРАВЛІННЯ КОШТАМИ БЛАГОДІЙНОГО ФОНДУ

3.1 Обґрунтування вибору інструментів для створення вебзастосунку

Вибір інструментів програмної реалізації FondUA визначається призначенням системи як вебсередовища для управління коштами благодійного фонду. Система має підтримувати роботу з користувачами, фондами, благодійними проєктами, пожертвами, витратами, бюджетами, фінансовими звітами, аналітичними показниками та рольовим доступом. Через це технологічний стек повинен забезпечувати серверне оброблення запитів, збереження пов'язаних даних, формування сторінок, роботу з формами, обмеження доступу за ролями, підготовку даних для звітності та візуалізацію фінансової аналітики.

Основою серверної частини FondUA обрано Python і Django. Python використовується як основна мова програмування для реалізації бізнес-логіки, оброблення даних, фінансових розрахунків і взаємодії між компонентами вебзастосунку. Django застосовується як основний вебфреймворк, оскільки підтримує маршрутизацію запитів, представлення, шаблони, роботу з моделями, форми, автентифікацію користувачів та адміністративні можливості. Для FondUA це важливо, оскільки система потребує структурованої роботи з ролями, фондами, проєктами, пожертвами, витратами, бюджетами й фінансовими звітами [5].

Окреме значення має Django ORM, який використовується для взаємодії з базою даних через моделі. За допомогою ORM сутності предметної області FondUA подаються як програмні моделі: користувач, профіль користувача, фонд, благодійний проєкт, пожертва, витрата, категорія витрат, бюджетний план, бюджетний рядок і фінансовий звіт. Це дає змогу виконувати операції створення, читання, оновлення та видалення записів без ручного написання типових SQL-

запитів для кожної дії, а також підтримувати зв'язки між сутностями на рівні моделей [5].

Для збереження даних у поточній версії системи використовується SQLite. Ця база даних є вбудованою, не потребує окремого серверного розгортання та підходить для навчально-практичної, локальної або демонстраційної реалізації вебсистеми. Для FondUA важливо, що SQLite підтримує реляційну структуру даних, у якій можна пов'язувати фонди, благодійні проєкти, пожертви, витрати, бюджетні записи та фінансові звіти [6]. У разі промислового розгортання системи доцільним напрямом розвитку є перехід на серверну систему керування базами даних, наприклад PostgreSQL, однак для поточної навчальної реалізації SQLite є достатнім рішенням.

Клієнтська частина FondUA реалізується за допомогою HTML, CSS і JavaScript. HTML використовується для структурування сторінок, форм, таблиць, карток фондів і благодійних проєктів. CSS відповідає за візуальне оформлення сторінок, елементів навігації, службових таблиць, форм і адаптацію інтерфейсу. JavaScript застосовується для інтерактивних елементів, які покращують взаємодію користувача з вебсистемою, а також для відображення аналітичних графіків на клієнтському рівні. Такий набір інструментів відповідає базовій логіці вебпрограмування, де клієнтська частина забезпечує відображення даних і взаємодію користувача із серверною частиною [13; 14; 16].

Для подання аналітичних показників використовується Chart.js. Цей інструмент дає змогу відображати підготовлені серверною частиною фінансові дані у вигляді графіків і діаграм. У межах FondUA Chart.js не виконує фінансові розрахунки, а відповідає за візуальне подання вже сформованих показників, наприклад динаміки пожертв, структури витрат, співвідношення планових і фактичних бюджетних значень або загального стану фінансових показників [17].

Для підготовки та експорту табличних даних у FondUA використовуються pandas і openpyxl. Бібліотека pandas застосовується для формування структурованих табличних наборів даних, а openpyxl використовується під час створення XLSX-файлів. Це дає змогу представнику фонду або

суперадміністратору отримувати фінансові та аналітичні підсумки не лише у вебінтерфейсі, а й у вигляді файлу для подальшого збереження, перевірки або передавання.

Для роботи із зображеннями використовується Pillow. У FondUA зображення можуть застосовуватися для логотипів фондів і візуального подання благодійних проєктів. Оскільки Django використовує Pillow для коректної роботи з полями зображень, ця бібліотека є необхідною для підтримки завантаження й оброблення графічних файлів у межах моделей і форм.

Для покращення роботи з формами використовується django-widget-tweaks. Цей інструмент дає змогу зручніше налаштовувати відображення полів форм без надмірного ускладнення серверної логіки. Для FondUA це важливо, оскільки система містить багато форм: реєстрацію, профіль користувача, створення фонду, створення проєкту, внесення пожертви, додавання витрати, формування бюджету, підготовку фінансового звіту та фільтрацію аналітики.

Для фільтрації даних використовується django-filter. У FondUA фільтрація потрібна для роботи зі списками проєктів, фінансових записів, звітів або аналітичних даних за параметрами проєкту, фонду, статусу, категорії чи періоду. Використання django-filter спрощує побудову керованих фільтрів і дає змогу не дублювати однотипну логіку оброблення параметрів запиту.

Для роботи зі статичними файлами використовується стандартний механізм Django staticfiles і WhiteNoise. Статичні файли охоплюють CSS, JavaScript, зображення, іконки та інші ресурси інтерфейсу. WhiteNoise дає змогу обслуговувати статичні файли без окремого вебсервера у простій конфігурації розгортання. Це зручно для навчально-практичного проєкту, де важливо забезпечити працездатність інтерфейсу без складної інфраструктури.

Основні інструменти програмної реалізації FondUA та їх призначення наведено в табл. 3.1.

Таблиця 3.1

Інструменти програмної реалізації вебсистеми FondUA

Інструмент	Призначення у FondUA	Обґрунтування вибору
1	2	3
Python	Серверна логіка, оброблення даних, фінансові розрахунки	Придатний для реалізації бізнес-логіки, роботи з даними, формування звітів і взаємодії з Django
Django	Основний вебфреймворк	Підтримує маршрутизацію, шаблони, моделі, форми, автентифікацію, адміністративні можливості та структуровану побудову вебзастосунку
Django ORM	Робота з базою даних через моделі	Дає змогу описувати сутності FondUA та їхні зв'язки без ручного написання типових SQL-запитів
SQLite	Локальне реляційне збереження даних	Підходить для навчально-практичної та демонстраційної версії системи; підтримує пов'язані таблиці й базові транзакційні операції
HTML	Структура сторінок і форм	Забезпечує побудову публічних сторінок, службових форм, таблиць, карток фондів і благодійних проєктів
CSS	Візуальне оформлення інтерфейсу	Використовується для оформлення сторінок, навігації, форм, таблиць, карток і адаптації інтерфейсу
JavaScript	Інтерактивність клієнтської частини	Використовується для динамічних елементів інтерфейсу та взаємодії з підготовленими сервером даними
Chart.js	Візуалізація аналітичних показників	Дає змогу подавати фінансові показники у вигляді графіків і діаграм без перенесення розрахунків на клієнтську частину
pandas	Підготовка структурованих табличних даних	Використовується для формування наборів даних, які можуть експортуватися або подаватися в аналітичному контурі
openpyxl	Формування XLSX-файлів	Забезпечує створення Excel-файлів із фінансовими та аналітичними даними
Pillow	Робота із зображеннями	Потрібний для підтримки завантаження й оброблення зображень фондів і благодійних проєктів

Продовження табл. 3.1.

	2	3
django-widget-tweaks	Налаштування відображення полів форм	Спрощує оформлення форм у шаблонах без дублювання серверної логіки
django-filter	Фільтрація списків і аналітичних даних	Полегшує фільтрацію записів за статусами, проектами, фондами, категоріями та періодами
Django staticfiles	Підключення статичних ресурсів	Забезпечує організацію CSS, JavaScript, зображень та інших ресурсів інтерфейсу
WhiteNoise	Обслуговування статичних файлів	Дає змогу обслуговувати статичні файли у простій конфігурації розгортання без окремого вебсервера
python-dotenv	Робота зі змінними середовища	Дає змогу винести службові налаштування з коду в конфігураційне середовище

(Розроблено автором)

Для конфігураційних параметрів використовується python-dotenv. Він дає змогу винести окремі службові налаштування, зокрема секретний ключ, режим налагодження, дозволені хости, URL системи або інші параметри середовища, у змінні середовища. Такий підхід не замінює повноцінних механізмів безпеки, але допомагає відокремити конфігурацію від основного коду застосунку та зменшує ризик випадкового збереження службових параметрів у програмному коді.

Узагальнений стек інструментів відповідає задачам FondUA. Python і Django забезпечують серверну логіку, Django ORM і SQLite - роботу з пов'язаними даними, HTML, CSS і JavaScript - клієнтську частину, Chart.js - візуальне подання аналітики, pandas і openpyxl - підтримку табличної звітності, Pillow - роботу із зображеннями, а django-widget-tweaks і django-filter - зручніше оформлення форм та фільтрацію даних. Django staticfiles, WhiteNoise і python-dotenv забезпечують допоміжні механізми організації статичних ресурсів, конфігурації та простого розгортання.

Для запуску системи потрібне програмне середовище з установленою сумісною версією Python, залежностями з файлу requirements.txt, базою даних

SQLite, браузером для роботи з інтерфейсом і локальним сервером Django. У поточній навчально-практичній версії такий набір є достатнім для демонстрації основних функцій FondUA: реєстрації користувачів, роботи з фондами та проектами, фіксації пожертв, обліку витрат, бюджетування, формування фінансових звітів, аналітики й експорту табличних даних.

Обраний технологічний стек створює основу для реалізації серверної частини, моделей, форм, маршрутів, шаблонів, фінансових розрахунків, звітності та аналітичної панелі FondUA. У наступному підпункті розглянуто розробку серверної частини та механізмів керування даними.

3.2 Розробка серверної частини та механізмів керування даними

Серверна частина вебсистеми FondUA виконує роль центрального рівня, на якому обробляються запити користувачів, перевіряються права доступу, створюються та змінюються записи бази даних, формуються фінансові показники, готуються звіти й передаються дані для аналітичного подання. У межах Django ця логіка реалізується через моделі, представлення, форми, маршрути, шаблони, сервісні функції та ORM-рівень, що забезпечує зв'язок між діями користувача, серверною обробкою і збереженням даних у базі [5].

FondUA має модульну структуру та поділяється на кілька Django-додатків, кожен із яких відповідає за окрему частину предметної області. Додаток accounts забезпечує роботу з користувачами, профілями, автентифікацією та ролями. Додаток funds відповідає за дані благодійних фондів і зв'язок фонду з його представником. Додаток projects охоплює створення, редагування, перегляд, архівування та відновлення благодійних проєктів. Додаток finance реалізує роботу з пожертвами, витратами, категоріями витрат, бюджетними планами та бюджетними рядками. Додаток reports відповідає за фінансові звіти, analytics - за агрегування фінансових показників, а core - за загальні сторінки та базову навігаційну логіку системи.

Загальний серверний workflow побудовано за послідовною схемою. Користувач відкриває сторінку або надсилає форму, після чого запит передається відповідному маршруту з файлу `urls.py`. Представлення з `views.py` визначає потрібний сценарій, перевіряє автентифікацію, роль користувача, належність об'єкта конкретному фонду та коректність введених даних. Форми з `forms.py` виконують первинну валідацію даних, а сервісні функції використовуються для складніших операцій, пов'язаних із фінансовими розрахунками, перевіркою витрат, формуванням звітів або підготовкою аналітичних показників. Після цього через Django ORM система читає або змінює записи в базі даних SQLite, а результат повертається у шаблон у вигляді сторінки, повідомлення, таблиці, фінансового звіту або набору аналітичних даних [5; 6].

Першим важливим серверним сценарієм є реєстрація та автентифікація користувача. Під час реєстрації створюється обліковий запис `User`, після чого формується пов'язаний профіль `Profile` із додатковими даними та роллю. Роль користувача визначає набір доступних дій: неавторизований відвідувач працює лише з відкритою частиною системи; донор може створювати пожертви та переглядати власні записи; представник фонду керує фондом, проєктами, витратами, бюджетами й звітами в межах власної організації; суперадміністратор виконує контрольні дії та має доступ до системного рівня. Автентифікація є обов'язковою для службових операцій, пов'язаних зі створенням фонду, керуванням проєктами, витратами, бюджетними планами, фінансовими звітами та адміністративною перевіркою.

Приклад сторінки реєстрації або входу користувача у вебсистемі `FondUA` подано на рис. 3.1.

Другий серверний сценарій пов'язаний зі створенням і редагуванням благодійного фонду. Фонд є організаційним рівнем системи, до якого надалі прив'язуються благодійні проєкти, фінансові операції, бюджетні плани, звіти та аналітичні показники. Представник фонду заповнює основні дані організації, а серверна частина перевіряє, чи має користувач право створювати або змінювати відповідний фонд. Це потрібно для того, щоб представник одного фонду не міг

редагувати дані іншої організації. Статус фонду також має значення для відкритої частини системи: фонд зі статусом pending очікує перевірки, approved може відображатися у публічному контурі, а blocked обмежується в доступі.

Авторизація для донорів, фондів і адміністраторів

Увійдіть до FondUA, щоб переглядати власні пожертви, керувати благодійними проектами, додавати витрати або перевіряти фінансову звітність.
 Донор бачить історію власних пожертв і підтримані проекти.
 Фонд отримує доступ до проектів, витрат, документів і звітів.
 Супер-адмін перевіряє фонди, користувачів і фінансові записи.

Увійти Зареєструватися

FondUA

Вихід із системи виконано.

Регістрація

Тип акаунта

Донор / користувач

Ім'я або назва представника

Ваше ім'я

Email

example@email.com

Контактний номер

+380 XX XXX XX XX

Пароль

Мінімум 8 символів

☐ Погоджуюсь з обробкою даних

Створити акаунт

Вже зареєстровані? Увійти

Рис. 3.1. Сторінка реєстрації або входу користувача у веб-системі FondUA

(Розроблено автором)

Третій сценарій охоплює створення та керування благодійними проектами. Представник фонду створює проект, зазначає його назву, опис, категорію, цільову суму, статус, зображення та інші дані, потрібні для публічного й фінансового подання. Серверна частина перевіряє, чи пов'язаний проект із фондом користувача, чи коректно заповнено обов'язкові поля та чи має користувач право виконувати цю дію. Проект є центральною точкою фінансового контуру FondUA, оскільки з ним пов'язуються пожертви, витрати,

бюджетні плани, фінансові звіти й аналітичні показники. Повний вигляд сторінки благодійного проєкту наведено в додатку А.

Для керування життєвим циклом благодійного проєкту у FondUA передбачено не лише створення та редагування записів, а й зміну їхнього стану. Проєкт може перебувати в стані чернетки, бути активним, завершеним або архівованим. Архівування дає змогу прибрати проєкт з основного робочого переліку без видалення пов'язаних фінансових операцій. За потреби архівований проєкт може бути відновлений, якщо він знову має відображатися в робочому переліку фонду.

Модуль finance реалізує серверну логіку фінансових операцій. Сценарій внесення пожертви передбачає вибір благодійного проєкту, введення суми, валюти, способу оплати та додаткових даних донора, після чого система створює запис Donation і пов'язує його з відповідним проєктом. Якщо донор авторизований, запис може бути пов'язаний із його обліковим записом. Для пожертви зберігаються спосіб оплати, унікальний ідентифікатор транзакції, ознака анонімності та короткий опис платіжних реквізитів або демонстраційного платежу. До фінансових показників проєкту включаються лише ті пожертви, статус яких підтверджує завершення операції.

У поточній версії FondUA форма пожертви використовується для демонстраційної фіксації внеску в межах внутрішньої логіки системи. Реальну інтеграцію з платіжним сервісом не реалізовано, тому платіжні дані використовуються для моделювання сценарію внесення пожертви, а не для фактичного списання коштів.

Форму внесення пожертви до благодійного проєкту наведено на рис. 3.2.

Після створення запису пожертви користувач переходить на сторінку підтвердження, де відображається результат виконаної дії. Така сторінка завершує донорський сценарій: користувач бачить, що внесок зафіксовано системою, а запис пожертви пов'язано з відповідним благодійним проєктом. На рівні серверної частини після збереження пожертви можуть бути оновлені або

повторно обчислені фінансові показники проєкту, зокрема зібрана сума та відсоток виконання збору.

ДЕМОНСТРАЦІЙНА ОПЛАТА

Підтримати благодійний проєкт

Заповніть форму пожертвування. У межах демонстраційного проєкту ця операція використовується для демонстрації фінансового обліку, оновлення суми збору, аналітики та звітності.

Це демонстраційна форма оплати

Система не підключена до реального платіжного шлюзу та не списує кошти. Після відправлення створюється демонстраційний запис пожертви в базі даних.
номер банківської картки не вводиться;
CVV та строк дії картки не вводяться;
чутливі платіжні дані не зберігаються;
операція використовується тільки для демонстрації бізнес-логіки системи.

1. Облік пожертви

Система створює запис про надходження коштів і пошукує його з проєктом.

2. Оновлення проєкту

Після створення запису змінюється зібрана сума, залишок і прогрес збору.

3. Аналітика і звітність

Пожертва потрапляє в КРІ, графіки, таблиці та фінансові звіти.

Дані пожертви

Основні реалізати демонстраційного внеску: проєкт, сума, валюта та спосіб класифікації платежу.

Сума пожертви *

Валюта

500

UAH

Сума буде вписана в фінансовий аналітичний проєкт.

Для демонстрації основних валютних є гривня.

2250

2500

21 000

22 500

25 000

Спосіб демонстраційної оплати

Банківська картка

PayPal

Google Pay

Банківський переказ

Тестовий платіж

Ця форма класифікації демонстраційного платежу. Реальні банківські транзакції не виконуються.

ОБРАНИЙ ПРОЄКТ

Психологічна підтримка дітей

БФ Майбутнє дітей

Активний

Мета збору

2160000

Вже зібрано

248750

Підтверджені витрати

263782

Залишок

2-18042

Прогрес збору

30%

Фінансування консультацій психологів і груп підтримки дітей дітей.

Що відбудеться після відправлення

1. Створення запису

У базі даних буде створено запис пожертви з сумою, проєктом і способом демонстраційної оплати.

2. Оновлення фінансових показників

Сума збору, залишок і прогрес проєкту оновляться на сторінках системи.

3. Відображення в аналітиці

Пожертва потрапить у графіки, КРІ та фінансову звітність.

4. Використання у звітах

Після перерахунку фінансового звіту сума буде вказана у надходженнях за відповідний період.

Попередній вигляд квитанції

Після створення демонстраційної пожертви система може показати сторінку подяки або квитанцію з основними даними операції.

Тип операції

Демонстраційна пожертва

Рис. 3.2. Форма внесення пожертви до благодійного проєкту у вебсистемі (Розроблено автором)

Окремий workflow пов’язаний із додаванням і перевіркою витрат. Представник фонду створює запис витрати, зазначає проєкт, суму, категорію, дату, опис і документальне підтвердження. Після створення витрата отримує статус очікування перевірки та не впливає на фінансові показники проєкту до моменту погодження. Суперадміністратор переглядає суму, категорію, опис, дату, підтвердні матеріали та приймає рішення про погодження або відхилення

запису. Лише погоджені витрати враховуються під час обчислення витраченої суми, залишку коштів і звітних показників. Форму додавання витрати до проекту наведено на рис. 3.3.

ВИТРАТА ПРОЕКТУ

Додати витрату проекту

Заповніть дані про використання коштів: проект, категорію, суму, дату, опис і підтверджувальний документ. Після створення витрата потрапить в систему контролю та може бути підтверджена адміністратором.

Проект
 ▼
Оберіть проект, до якого належить ця витрата. Список обмежений доступними для вашої ролі проектами.

Категорія
 ▼
Категорія потрібна для аналізу витрат.

Назва витрати

Наприклад: закупівля медикаментів або доставка документів.

Сума

Сума витрати у гривнях. Сума не може перевищувати доступний залишок проекту.

Дата витрати
 📅
Дата фактичного використання коштів.

Статус перевірки
 ▼
Статус може змінювати лише адміністратор системи.

Документ / чек
 Файл не вибран
Можна перевірити PDF, JPG, PNG, WEBP, DOC або DOCX до 10 MB.

Опис

Опишіть, на що саме були використані кошти.

Як це працює
 Витрата створюється як фінансова операція, прив'язана до конкретного бюджетного проекту. Після підтвердження вона впливає на показники "витрачено" та "залишок" у проекті.

Правила перевірки
 Очікує перевірки
Витрата додана фондом, але ще не підтверджена.
 Підтверджено
Витрата враховується у фінансовій звітності.
 Відхилено
Витрата не враховується у підтверджених витратах.

Документи
 Для прозорості банкіно додавати чек, акт, рахунок або інший документ, який підтверджує використання коштів.
 Документ не є обов'язковим технічно, але для логіки дипломного проекту краще додавати його майже завжди. Так система покаже не лише факт витрати, а й підтвердження використання коштів.

Зберегти витрату **До опису витрат** **Фінансовий кабінет**

Рис. 3.3. Форма додавання витрати у веб-системі FondUA

(Розроблено автором)

Перевірка витрат у FondUA реалізується через службовий перегляд списку витрат із можливістю зміни їхнього статусу. Представник фонду створює запис витрати, після чого вона залишається у стані pending. Суперадміністратор може змінити статус витрати на approved або rejected. Якщо витрату відхилено, у системі може зберігатися причина відхилення. Такий підхід відокремлює

первинне внесення фінансової інформації від її включення до офіційних показників системи. Список витрат із діями перевірки та зміни статусу подано на рис. 3.4.

Фонд	Категорія витрат	Сума	Статус перевірки	Дата витрати	Хто додав	Хто перевіряв	Дата перевірки
БФ Турбота поруч	Інше	4 652.00 грн	Очікує перевірки	15 травня 2026 р.	admin_demo	-	-
БФ Майбутнє дітей	Логістика	31 847.00 грн	Підтверджено	12 травня 2026 р.	admin_demo	admin_demo	13 травня 2026 р. 23:06
БФ Майбутнє дітей	Інше	4 234.00 грн	Очікує перевірки	09 травня 2026 р.	admin_demo	-	-
БФ Турбота поруч	Обладнання	18 450.00 грн	Підтверджено	08 травня 2026 р.	admin_demo	admin_demo	06 травня 2026 р. 23:06
БФ Майбутнє дітей	Інше	2 345.00 грн	Очікує перевірки	07 травня 2026 р.	admin_demo	-	-
БФ Надія поруч	Інше	3 425.00 грн	Очікує перевірки	07 травня 2026 р.	admin_demo	-	-
БФ Турбота поруч	Інше	3 359.00 грн	Очікує перевірки	06 травня 2026 р.	admin_demo	-	-
БФ Турбота поруч	Продукти	11 178.00 грн	Підтверджено	06 травня 2026 р.	admin_demo	admin_demo	10 травня 2026 р. 23:06
БФ Надія поруч	Медикаменти	22 275.00 грн	Підтверджено	06 травня 2026 р.	admin_demo	admin_demo	14 травня 2026 р. 23:06
БФ Надія поруч	Інше	3 354.00 грн	Очікує перевірки	06 травня 2026 р.	admin_demo	-	-

Рис. 3.4. Сторінка перевірки та погодження витрат у веб-системі FondUA
(Розроблено автором)

Бюджетування реалізується через бюджетні плани та бюджетні рядки. Бюджетний план пов'язується з конкретним благодійним проектом і визначає період планування. Бюджетні рядки деталізують планові суми за категоріями витрат. На основі цих записів система зіставляє планові показники з фактичними погодженими витратами відповідних категорій. Це дає представнику фонду змогу контролювати план-фактне відхилення та виявляти напрями, у яких витрати перевищують заплановані значення.

Приклад деталізації бюджетного плану благодійного проекту наведено на рис. 3.5.

Модуль reports відповідає за формування фінансових звітів. Звіт створюється за фондом або окремим благодійним проектом у межах визначеного періоду та містить підсумок пожертв, витрат, залишку коштів, кількість

фінансових операцій і коротке пояснення. Серверна частина формує підсумкові значення на основі фінансових записів, які відповідають обраному періоду та мають допустимі статуси.

План/факт за категоріями				
Рядки бюджету показують, скільки планувалось витратити і скільки реально витрачено у відповідній категорії.				
Категорія	План	Факт	Різниця	Виконання
Адміністративні витрати	₴44418	₴0	₴44418	Виконано
Логістика	₴37637	₴0	₴37637	Виконано
Медикаменти	₴11974	₴0	₴11974	Виконано
Обладнання	₴47935	₴0	₴47935	Виконано
Продукти	₴10520	₴0	₴10520	Виконано

Рис. 3.5. Сторінка бюджетного плану благодійного проєкту у веб-системі FondUA

(Розроблено автором)

Фінансовий звіт у FondUA проходить кілька службових станів. На початковому етапі представник фонду створює чернетку звіту зі статусом draft. Після заповнення основних даних звіт може бути переданий на перевірку зі статусом submitted. Суперадміністратор перевіряє коректність підсумкових сум, періоду, зв'язку з фондом або проєктом і може погодити, відхилити або опублікувати звіт. Публічне відображення звіту виконується лише після

переходу до статусу published, щоб відкриті сторінки системи не показували службові або неперевірені записи.

Форму створення фінансового звіту наведено на рис. 3.6.

The screenshot shows a web form titled "Створити фінансовий звіт" (Create Financial Report) with a sub-header "НОВИЙ ФІНАНСОВИЙ ЗВІТ" (New Financial Report). The form includes the following sections:

- Фонд *** (Fund): A dropdown menu.
- Проект** (Project): A dropdown menu with a note: "Залиште поле порожнім, якщо звіт формується по всіх проектах обраного фонду." (Leave the field empty if the report is generated for all projects of the selected fund).
- Назва звіту *** (Report Name): A text input field with an example: "Наприклад: Звіт за травень 2026" (Example: Report for May 2026). Below it, a note says: "Наприклад: 'Звіт фонду за травень 2026' або 'Звіт по проекту закупівлі обладнання'." (Example: 'Report of the fund for May 2026' or 'Report by project procurement of equipment').
- Початок періоду *** (Start of period): A date picker with the format "ДД.ММ.РРРР".
- Кінець періоду *** (End of period): A date picker with the format "ДД.ММ.РРРР".
- Короткий опис звіту** (Short description of the report): A text area with a placeholder: "Коротко опишіть результати використання коштів за період." (Briefly describe the results of the use of funds for the period). Below it, a note says: "Опишіть основні результати здійснення коштів надійшло, на що вони були використані, які нагромади профінансовано та який залишок сформовано." (Describe the main results of the receipt of funds, for what they were used, what has been accumulated and financed, and what balance has been formed).
- Документ звіту** (Report document): A file upload section with a button "Виберіть файл" (Select file) and the text "Файл не вибран" (File not selected). Below it, a note says: "Можна додати PDF, скан, таблицю або інший підтвердний файл." (You can add PDF, scan, table or other confirming file).

At the bottom, there are two buttons: "Зберегти звіт" (Save report) and "Скасувати" (Cancel).

Рис. 3.6. Форма створення фінансового звіту у веб-системі FondUA
(Розроблено автором)

Повний вигляд публічного відображення фінансового звіту у FondUA наведено в додатку Б.

Модуль analytics агрегує фінансові дані та готує їх для подання в аналітичній частині системи. На серверному рівні формуються показники щодо зібраних коштів, підтверджених витрат, залишку, структури витрат за категоріями, виконання бюджету, стану окремих проєктів і прозорості фінансових даних. Аналітика у FondUA має кілька рівнів доступу: публічний контур, аналітику фонду та системну аналітику. Публічний контур містить узагальнені відкриті показники, представник фонду працює з даними власної організації, а суперадміністратор має доступ до системного огляду. Клієнтська частина отримує вже підготовлені дані, які можуть відображатися у вигляді таблиць, показників або діаграм. Детальне візуальне подання аналітичної панелі розглянуто в підпункті 3.3, оскільки воно пов'язане з клієнтським інтерфейсом.

Серверна частина FondUA забезпечує зв'язок між модулями. Наприклад, під час перегляду благодійного проєкту система отримує основні дані з projects, інформацію про фонд із funds, фінансові записи з finance, а підсумкові показники формуються через властивості моделей або сервісні функції. У результаті користувач бачить не набір окремих таблиць, а узгоджену інформацію про проєкт, його організатора, суму збору, підтверджені витрати, залишок коштів і звітні дані.

Управління даними здійснюється через Django ORM і базу даних SQLite. ORM забезпечує зв'язок між моделями системи та таблицями бази даних, а SQLite використовується для збереження даних у навчально-практичній версії проєкту [5; 6]. Під час оброблення фінансових записів серверна частина застосовує фільтрацію за проєктом, фондом, категорією, періодом і статусом. Це потрібно для того, щоб до фінансових показників потрапляли лише коректні записи, а службові, неперевірені або відхилені дані не спотворювали публічний результат.

Для узагальнення структури серверної частини основні Django-додатки FondUA та їхні функції наведено в табл. 3.2.

Таблиця 3.2

Основні модулі вебсистеми FondUA та їх функції

Модуль	Основне призначення	Моделі / дані	Представлення та маршрути	Форми, сервіси й шаблони
accounts	Керування користувачами, профілями та ролями	Profile, зв'язок із User	Реєстрація, вхід, вихід, профіль користувача, перевірка ролі	Форми реєстрації та профілю; шаблони особистого кабінету й автентифікації
funds	Робота з благодійними фондами	Fund	Перегляд фондів, створення та редагування фонду, кабінет представника фонду	Форми фонду; шаблони списку фондів, сторінки фонду та кабінету фонду
projects	Керування благодійними проектами	Project	Список проектів, детальна сторінка проекту, створення, редагування, архівування та відновлення	Форми проектів; шаблони списку, детальної сторінки та сторінок керування
finance	Фінансові операції та бюджетування	Donation, ExpenseCategory, Expense, BudgetPlan, BudgetLine	Створення пожертви, перегляд реєстру пожертв, додавання витрати, перевірка витрат, перегляд бюджетних планів	Форми пожертв, витрат і бюджетів; сервісна логіка перевірки витрат, бюджетування та розрахунку фінансових показників
reports	Фінансова звітність	FinancialReport	Створення, редагування, подання, перевірка, публікація та перегляд фінансових звітів	Форми звітів; сервісна логіка заповнення підсумків; шаблони звітів
analytics	Агрегування та підготовка аналітичних показників	Дані з Project, Donation, Expense, BudgetPlan, FinancialReport	Публічна аналітика, аналітика фонду, системна аналітика, експорт даних	Сервіси агрегування, підготовка таблиць, діаграм і файлів експорту
core	Загальні сторінки та навігація	Загальні дані для публічного інтерфейсу	Головна сторінка, інформаційні сторінки, базова навігація	Базові шаблони, блоки інтерфейсу, спільні елементи сторінок

(Розроблено автором)

Окрему увагу приділено контролю доступу. FondUA працює з персональними даними користувачів, контактними відомостями, пожертвами, витратами та службовими фінансовими записами, тому система відокремлює публічні дії від службових. Закон України «Про захист персональних даних» вимагає визначеності мети обробки та прозорості роботи з персональними даними [4]. На рівні серверної частини це підтримується автентифікацією, авторизацією, рольовим доступом і перевіркою належності об'єктів конкретному фонду. Загальні ризики порушення контролю доступу та помилок автентифікації також належать до ключових ризиків вебзастосунків за OWASP [7; 8; 9].

У результаті серверна частина FondUA виконує роль керуючого рівня вебсистеми: вона приймає запити, визначає сценарій роботи, перевіряє користувача, застосовує рольові обмеження, звертається до моделей, змінює або читає дані, формує фінансові показники та передає підготовлену інформацію до шаблонів. У наступному підпункті розглянуто клієнтську частину системи та способи подання результатів користувачу.

3.3 Реалізація клієнтської частини та аналітичної панелі системи

Клієнтська частина вебсистеми FondUA забезпечує взаємодію користувачів із системою через сторінки, форми, таблиці, картки, повідомлення, фінансові звіти та аналітичні панелі. На відміну від серверної частини, яка відповідає за оброблення запитів, перевірку прав доступу та роботу з даними, клієнтська частина подає користувачу вже підготовлену інформацію у зрозумілому вигляді. Інтерфейс FondUA побудовано з урахуванням ролей користувачів: неавторизований відвідувач переглядає відкриту інформацію, донор підтримує благодійні проєкти, представник фонду керує службовими даними власної організації, а суперадміністратор контролює статуси, фінансові записи та коректність роботи системи.

Для побудови клієнтської частини використано HTML-шаблони Django, CSS-стилі та JavaScript. HTML формує структуру сторінок, форм і таблиць, CSS відповідає за візуальне оформлення та розміщення елементів, а JavaScript використовується для окремих інтерактивних компонентів і подання графічних елементів. Дані для сторінок готуються на серверному рівні та передаються до шаблонів, тому клієнтська частина не дублює фінансову логіку, а відображає результати її виконання [5; 16]. Загальна логіка клієнт-серверної взаємодії у вебзастосунках передбачає, що браузер забезпечує подання даних і приймання користувацьких дій, а серверна частина виконує оброблення запиту та повертає підготовлений результат [13; 14].

Публічний інтерфейс FondUA орієнтований на відвідувача та донора. До нього належать головна сторінка, список благодійних фондів, сторінка окремого фонду, список благодійних проєктів, сторінка окремого проєкту, форма внесення пожертви та публічні фінансові звіти. Головна сторінка виконує роль початкової точки входу: вона пояснює призначення системи, містить навігацію до фондів і проєктів та спрямовує користувача до основних сценаріїв роботи. Головну сторінку вебсистеми FondUA наведено на рис. 3.7.

На головній сторінці відвідувач отримує загальне уявлення про систему, переходить до списку фондів або благодійних проєктів і може обрати подальшу дію. Ця сторінка демонструє публічний контур FondUA, тобто частину системи, доступну без авторизації.

Список благодійних фондів і список благодійних проєктів формують публічний каталог FondUA. На таких сторінках відвідувач або донор може переглянути доступні фонди, назви проєктів, короткі описи, цільові суми та стан збору. Для благодійної організації важливо, щоб публічні сторінки не обмежувалися загальними повідомленнями, а містили конкретні дані про мету проєкту, фінансовий прогрес і можливість переходу до звітності. Текстове наповнення сторінок благодійних організацій має безпосереднє значення для пояснення потреби та формування довіри до збору [18].

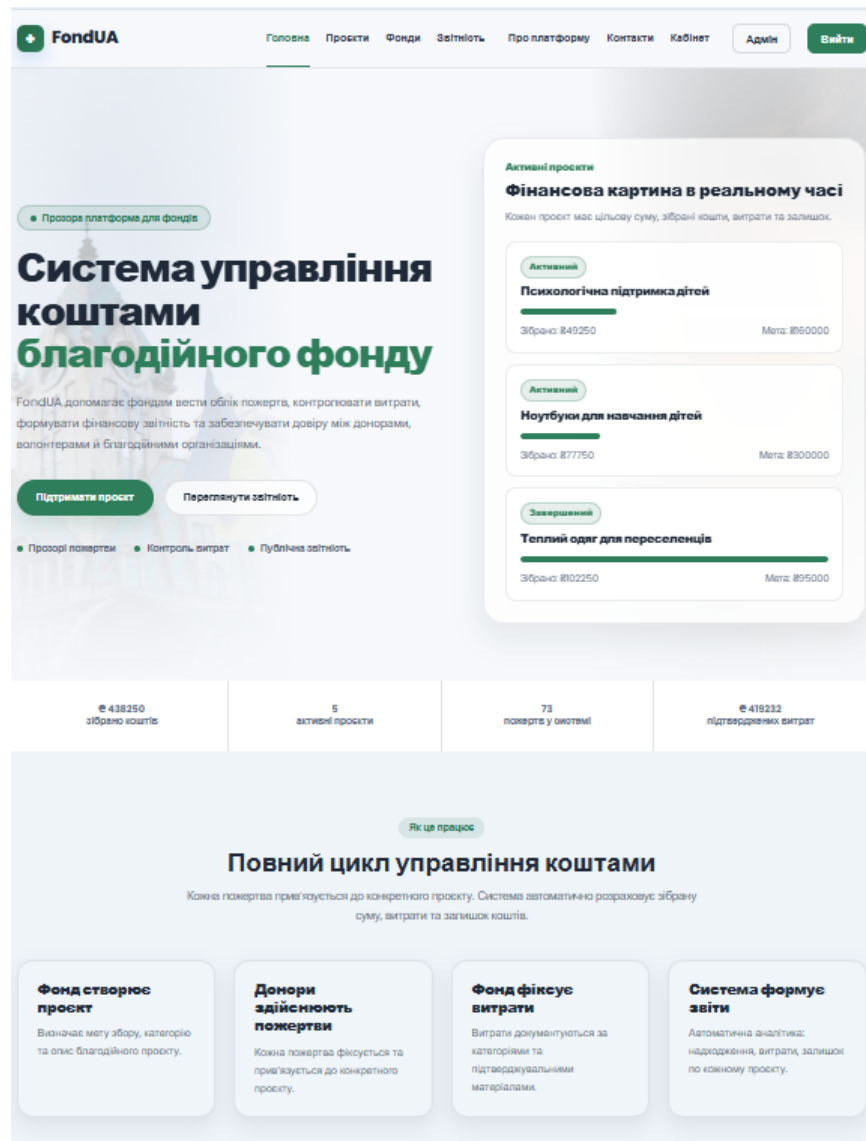


Рис. 3.7. Головна сторінка вебсистеми FondUA

(Розроблено автором)

Список благодійних проєктів у публічному інтерфейсі системи подано на рис. 3.8.

На сторінці списку проєктів відвідувач або донор бачить картки проєктів, короткий опис, цільову суму та поточний стан збору. Цей інтерфейс підтверджує, що система має відкритий каталог благодійних проєктів, а не лише службову частину для представників фондів.

Детальна сторінка благодійного проєкту є центральною сторінкою публічного контуру. На ній подається назва проєкту, опис потреби, інформація про фонд, цільова сума, сума підтверджених пожертв, підтверджені витрати,

залишок коштів і відсоток виконання збору. Для донора ця сторінка є основою прийняття рішення про підтримку, оскільки саме тут поєднуються змістова інформація про проєкт і фінансові показники, сформовані на основі серверних даних. Повний вигляд сторінки благодійного проєкту з фінансовими показниками наведено в додатку Б.

КАТАЛОГ ПРОЄКТІВ

Благодійні проєкти

Оберіть ініціативу для підтримки. Для кожного проєкту відображається цілі збору, фактично зібрані кошти, підтверджені витрати та залишок. Інформація про проєкти формується на основі даних, збережених у базі системи.

6 проєктів у системі
€ 438250 зібрані кошти
€ 419232 підтверджені витрати
€ 19018 поточний залишок

Створити проєкт

Пошук за назвою або описом | Усі категорії | Усі статуси | Спочатку нові | **Фільтрувати** | Скинути

Показано 6 з 0 проєктів | Сторінка 1 з 1

Назва проєкту	Категорія	Зібрано	Прогрес	Мета	Витрачено	Залишок
Психологічна підтримка дітей	Допомога дітям	€ 48250	37%	€ 160000	€ 63792	€ -14542
Ноутбуки для навчання дітей	Освіта	€ 77750	26%	€ 300000	€ 95284	€ -17534
Теплий одяг для переселенців	Гуманітарна допомога	€ 102250	100%	€ 95000	€ 53597	€ 48553
Допомога літнім людям	Допомога літнім людям	-	-	-	-	-
Гуманітарні набори для родин	Гуманітарна допомога	-	-	-	-	-
Медикаменти для дитячої лікарні	Медична допомога	-	-	-	-	-

Рис. 3.8. Список благодійних проєктів у веб-системі FondUA
(Розроблено автором)

Донорський сценарій у клієнтській частині пов'язаний із переглядом проєкту, внесенням пожертви та отриманням результату дії. Форма пожертви містить суму внеску, валюту, спосіб оплати, контактні дані донора, коментар та ознаку анонімності. У поточній версії FondUA не реалізовано інтеграцію з реальним платіжним сервісом; форма пожертви використовується для демонстраційної фіксації внеску в межах внутрішньої логіки обліку. Через це в роботі не стверджується, що система проводить фактичний онлайн-платіж. Форму внесення пожертви до благодійного проєкту наведено на рис. 3.2.

Після створення запису пожертви система надає користувачу повідомлення про результат дії або переводить його на сторінку підтвердження. Такий підхід завершує донорський сценарій і показує, що внесок зафіксовано як фінансовий запис, пов'язаний із конкретним благодійним проєктом. Якщо пожертва має статус завершеної, вона враховується у фінансових показниках проєкту, зокрема у зібраній сумі та прогресі збору.

Службовий інтерфейс представника фонду відрізняється від публічного контуру. Його призначення полягає не в перегляді відкритої інформації, а в керуванні фондом, благодійними проєктами, витратами, бюджетами та фінансовими звітами. У кабінеті представника фонду доступні переходи до сторінок керування фондом, списку проєктів, фінансових записів, бюджетного плану та звітності. Структура таких сторінок підтримує робочий маршрут: фонд - проєкти - фінансові записи - бюджет - звіт. Кабінет фонду у вебсистемі FondUA наведено на рис. 3.9.

На цій сторінці представник фонду отримує доступ до службових функцій, які недоступні звичайному відвідувачу або донору. Кабінет фонду виконує роль робочої зони користувача з відповідною роллю та забезпечує перехід до основних модулів керування даними.

Сторінки витрат і бюджетування є важливою частиною службового інтерфейсу. Таблиця витрат відображає назву витрати, суму, дату, категорію, статус і зв'язок із проєктом.

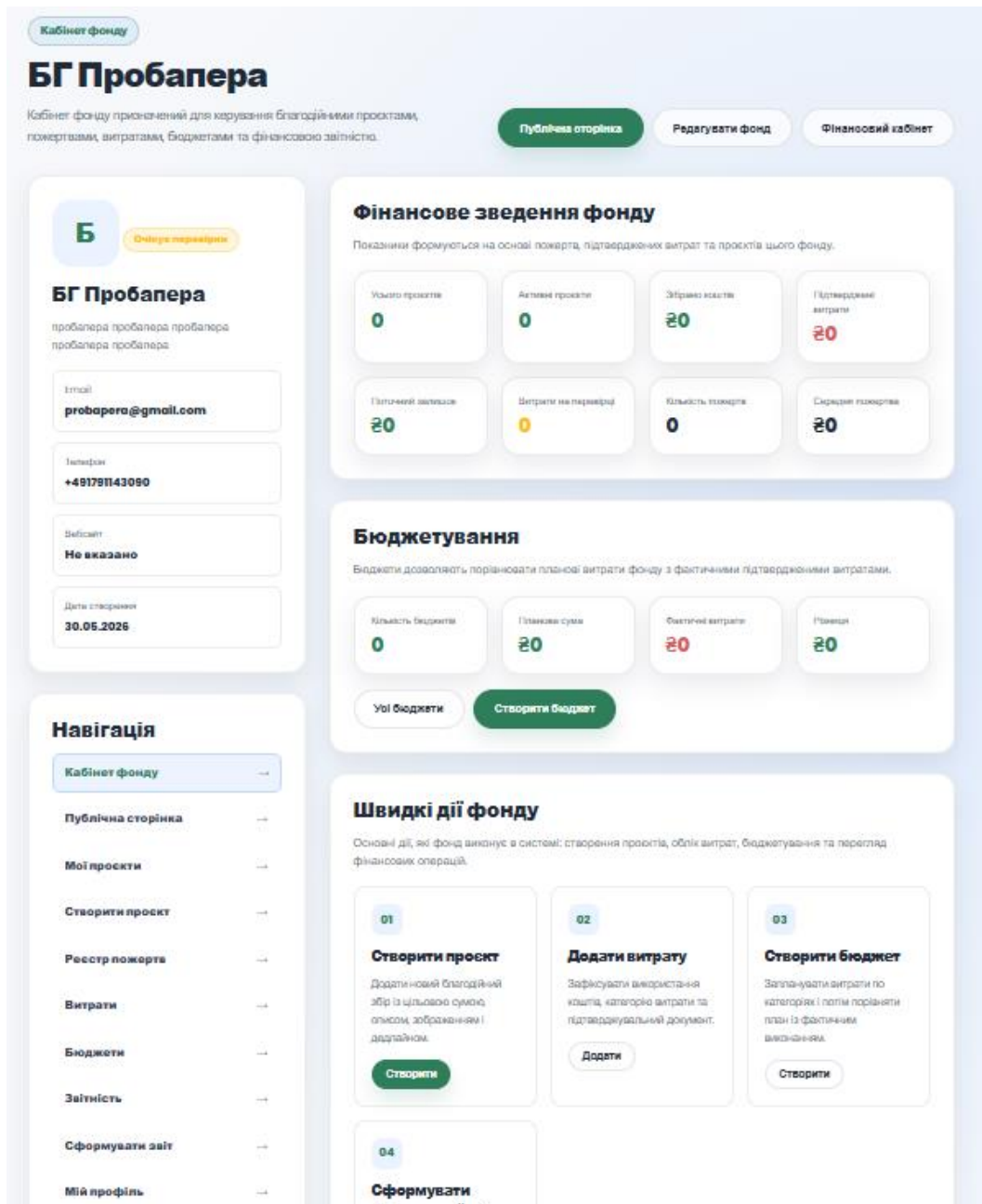


Рис. 3.9. Кабінет фонду у веб-системі FondUA

(Розроблено автором)

Статуси витрат мають бути видимими для представника фонду й суперадміністратора, оскільки лише погоджені витрати враховуються у фінансових показниках. Бюджетний план подає планові суми за категоріями, фактичні витрати та відхилення план/факт. Такий формат допомагає контролювати використання коштів без переходу до внутрішніх структур бази даних. Таблицю витрат благодійного проєкту наведено на рис. 3.10.

Витрати за проектом

Витрати показують, як саме фонд використовує отримані кошти. У фінансових підсумках враховуються тільки підтверджені витрати.

Дата	Категорія	Опис	Сума	Статус	Док.
09.05.2026	Інше	Витрата на перевірку Демонстраційна витрата, яка очікує перевірки адміністратором.	₴4234	Очікує перевірки	Немає
04.05.2026	Логістика	Доставка допомоги Оплата транспортування та логістики.	₴10394	Підтверджено	Немає
02.05.2026	Обладнання	Закупівля обладнання Придбання необхідного обладнання для реалізації проекту.	₴22712	Підтверджено	Немає
08.04.2026	Продукти	Формування продуктивних наборів Закупівля продуктів тривалого зберігання.	₴30686	Підтверджено	Немає

Рис. 3.10. Таблиця витрат благодійного проекту у веб-системі FondUA
(Розроблено автором)

На цій сторінці представник фонду або суперадміністратор може переглядати витрати, їхні категорії, суми та статуси. Скриншот демонструє, що витрати в системі подаються не як неструктурований текст, а як керовані фінансові записи з визначеним станом перевірки.

Сторінки фінансової звітності призначені для подання підсумків за фондом або благодійним проектом. У фінансовому звіті відображаються період, надходження, витрати, залишок, кількість фінансових операцій, статус звіту та короткий опис результатів. У публічній частині мають відображатися лише опубліковані звіти, тоді як службова частина використовується для створення, редагування, подання на перевірку, погодження або відхилення звітів. Фінансовий звіт благодійного проекту наведено на рис. 3.11.

На цій сторінці фінансові дані за визначений період подаються у зведеному вигляді. Рисунок демонструє, як дані про пожертви й витрати переходять у зрозумілий звітний формат для службового або публічного перегляду.

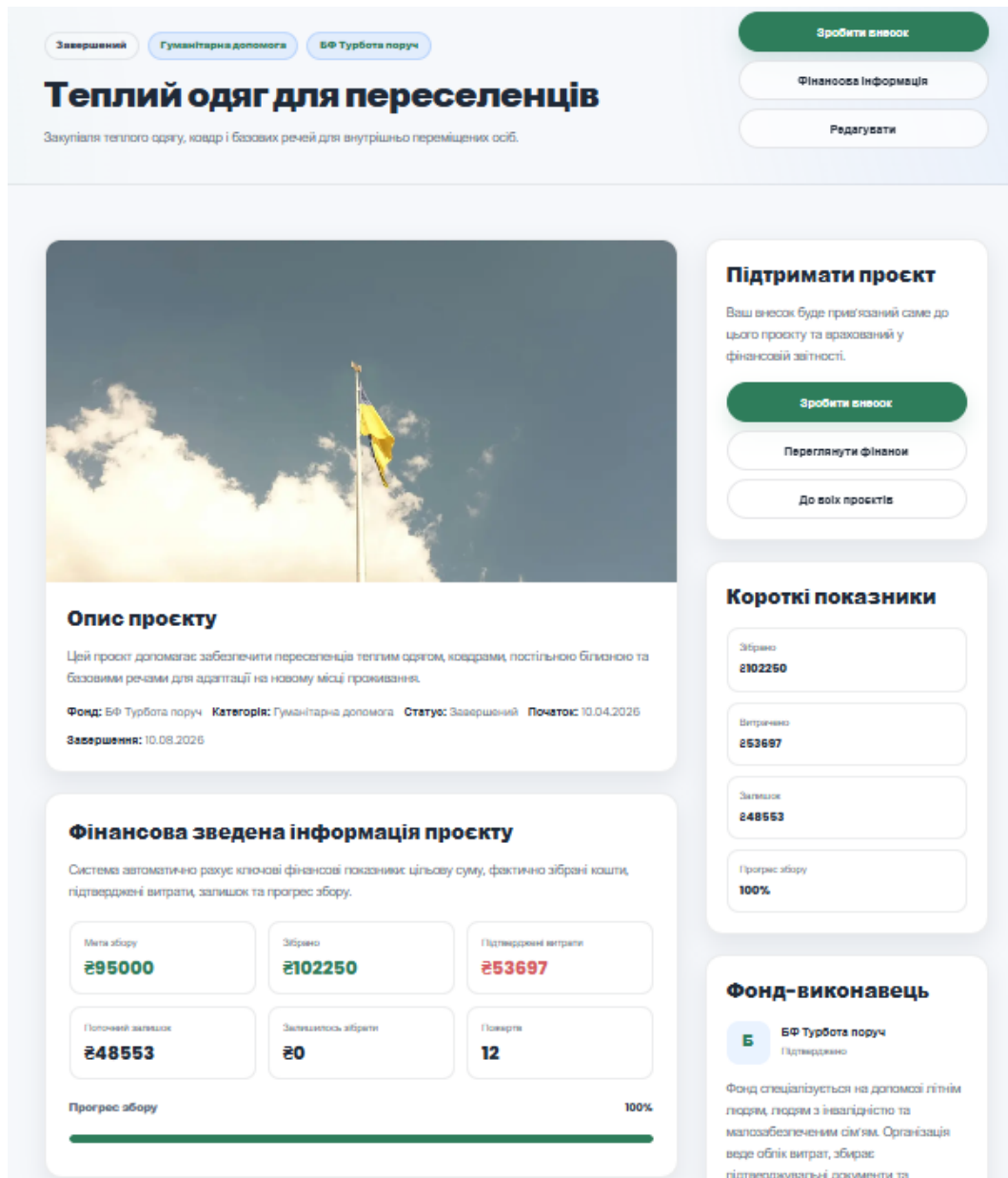


Рис. 3.11. Фінансовий звіт благодійного проєкту у веб-системі FondUA

(Розроблено автором)

Інтерфейс суперадміністратора відображає контрольний контур системи. На таких сторінках важливими є таблиці записів, статуси, фільтри, кнопки погодження або відхилення, а також доступ до узагальнених фінансових даних. Суперадміністратор не дублює функції представника фонду, оскільки його головне завдання полягає в перевірці коректності даних, зміні службових статусів і контролі фінансових записів. Приклад службового перегляду фінансових записів суперадміністратором подано на рис. 3.12.

Django адміністрування

ВІТАЄМО СУПЕР. ДІВІТИСЯ САЙТ / ЗМІНИТИ ПАРОЛЬ / ВІЙТИ

Домівка » Reports » Фінансові звіти

Почніть писати для фільтрації...

Виберіть Фінансовий звіт щоб змінити

ACCOUNTS

Профілі користувачів + Додати

FINANCE

Бюджети проєктів + Додати

Витрати + Додати

Категорії витрат + Додати

Пожертви + Додати

Різниця бюджету + Додати

FUNDS

Благодійні фонди + Додати

PROJECTS

Благодійні проєкти + Додати

REPORTS

Фінансові звіти + Додати

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Групи + Додати

Додати фінансовий звіт +

ВІДФІЛЬТРУВАТИ

Показати кількість

За Статус

Всі

Чернетка

Подано на перевірку

Підтверджено

Відхилено

Опубліковано

За Публічний звіт

Всі

Так

Ні

За Фонд

Всі

БФ Проабітера

БФ Майбутнє дітей

БФ Турбота поруч

БФ Надія поруч

За Проєкт

Всі

Положительна підтримка дітей

Неулюблені для навчання дітей

2025	2026	Дік	НАЗВА ЗВІТУ	ФОНД	ПРОЄКТ	ПЕРІОД	ЗАГАЛЬНИ НАДХОДЖЕННЯ	ЗАГАЛЬНИ ВИТРАТИ	ЗАЛИШОК	СТАТУС	ПУБЛІЧНИЙ ЗВІТ	СТВОРИВ	ДАТА СТВОРЕННЯ
			Публічний звіт проєкту «Теплий одяг для переселенців» за 05.2026	БФ Турбота поруч	Теплий одяг для переселенців	01.05.2026 — 26.05.2026	9000.00	32777.00	-23777.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний звіт проєкту «Допомога літнім людям» за 05.2026	БФ Турбота поруч	Допомога літнім людям	01.05.2026 — 26.05.2026	17000.00	18450.00	-1450.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний фінансовий звіт фонду «БФ Турбота поруч» за 05.2026	БФ Турбота поруч	Усі проєкти фонду	01.05.2026 — 26.05.2026	26000.00	51227.00	-25227.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний звіт проєкту «Медикаменти для дитячої лікарні» за 05.2026	БФ Надія поруч	Медикаменти для дитячої лікарні	01.05.2026 — 26.05.2026	6500.00	0.00	6500.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний звіт проєкту «Гуманітарні набори для родин» за 05.2026	БФ Надія поруч	Гуманітарні набори для родин	01.05.2026 — 26.05.2026	4000.00	22275.00	-18275.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний фінансовий звіт фонду «БФ Надія поруч» за 05.2026	БФ Надія поруч	Усі проєкти фонду	01.05.2026 — 26.05.2026	10500.00	22275.00	-11775.00	Опубліковано	✓	газета	26 травня 2026 р. 2029
			Публічний звіт проєкту «Психологічна підтримка дітей» за 05.2026	БФ Майбутнє дітей	Психологічна підтримка дітей	01.05.2026 — 26.05.2026	17750.00	33106.00	-15356.00	Опубліковано	✓	газета	26 травня 2026 р. 2029

Рис. 3.12. Перевірка фінансових записів суперадміністратором у вебсистемі
(Розроблено автором)

Аналітична панель є окремою частиною клієнтського подання FondUA. Вона не призначена для створення нових фінансових записів, а використовується для відображення підсумкових показників: зібраної суми, підтверджених витрат, залишку, прогресу збору, планових і фактичних бюджетних значень, структури витрат за категоріями та динаміки операцій. Аналітичний контур системи має три рівні доступу: публічний, фондовий і системний. Публічний рівень подає відкриті агреговані показники, фондовий рівень доступний представнику фонду для аналізу власних проєктів, а системний рівень використовується суперадміністратором для загального контролю стану системи.

Аналітична панель FondUA використовує агреговані дані з моделей пожертв, витрат, бюджетних планів і фінансових звітів. На ній відображаються загальні суми зібраних коштів, підтверджених витрат, залишку, кількість активних проєктів, співвідношення планових і фактичних витрат, структура витрат за категоріями та інші показники, потрібні для оцінювання фінансового стану фонду або всієї системи. Окремо передбачено підготовку даних для експорту у форматах CSV або XLSX, що дає змогу використовувати результати аналітики поза межами вебінтерфейсу.

Повний вигляд аналітичної панелі вебсистеми FondUA наведено в додатку Г. У цьому додатку доцільно подати розгорнутий скриншот панелі, щоб не перевантажувати основний текст розділу великим зображенням.

На аналітичній панелі фінансові дані подаються у вигляді узагальнених показників, таблиць і діаграм. Це дає змогу швидко оцінити стан зборів, структуру витрат, залишок коштів і відповідність фактичних витрат запланованому бюджету. Для візуалізації аналітичних показників використовується Chart.js. За допомогою графіків і діаграм можна подати динаміку пожертв, структуру витрат, співвідношення планових і фактичних показників або розподіл фінансових операцій за періодами. Chart.js у цьому випадку не виконує розрахунки, а відображає підготовлені сервером дані у вигляді графічних елементів [17].

У клієнтській частині FondUA важливими є не лише окремі сторінки, а й узгоджена структура навігації. Користувач повинен розуміти, у якій частині системи він перебуває, яку роль виконує і які дії доступні на поточній сторінці. Для цього використовуються меню, кнопки переходу, картки, таблиці, статуси записів і повідомлення про результат дії. Такі елементи мають функціональне значення: вони допомагають відокремити публічний перегляд від службового керування. Загальна структура сторінок, розміщення інформаційних блоків і зрозумілість взаємодії користувача з інтерфейсом є важливими складовими якості вебзастосунку [19; 20].

Клієнтська частина FondUA забезпечує наочне подання результатів серверної обробки та робить фінансову інформацію доступною для різних ролей користувачів. Відвідувач і донор працюють із публічними сторінками, представник фонду - зі службовими формами й таблицями, а суперадміністратор - із контрольними сторінками та статусами. Аналітична панель завершує цей контур, оскільки подає пожертви, витрати, бюджети, звіти й фінансові показники у зведеному вигляді. У наступному підпункті розглянуто перевірку працездатності вебсистеми та аналіз результатів реалізації.

3.4 Перевірка працездатності веб-системи та аналіз результатів реалізації

Після завершення реалізації основних модулів FondUA було виконано перевірку працездатності вебсистеми за ключовими функціональними сценаріями. Перевірка охопила публічні сторінки, реєстрацію та автентифікацію користувачів, рольовий доступ, створення фондів і благодійних проєктів, фіксацію пожертв, облік витрат, бюджетування, формування фінансових звітів та відображення аналітичних показників.

Метою перевірки було встановлення коректності роботи системи не лише на рівні відкриття сторінок, а й на рівні повного користувацького сценарію: введення даних, перевірки форми, збереження запису в базі даних, зміни статусу, перерахунку фінансових показників і відображення результату користувачу. Для негативних сценаріїв успішним результатом вважалося коректне блокування дії, перенаправлення на сторінку входу або показ повідомлення про помилку введення даних.

Перевірка виконувалася у двох напрямках. Перший напрям охоплював ручне функціональне тестування інтерфейсу, під час якого перевірялися сторінки, форми, повідомлення, таблиці та рольові обмеження. Другий напрям був пов'язаний з автоматизованими тестами Django, які перевіряли форми автентифікації, логіку ролей, фінансові операції, бюджетування, workflow фінансових звітів і фільтрацію аналітичних даних. Такий підхід дає змогу оцінити FondUA і як вебінтерфейс, і як програмну систему з окремими правилами обробки фінансових даних [5; 19].

Під час перевірки було використано такі групи тестових даних: користувачі з ролями донора, представника фонду та суперадміністратора; тестовий благодійний фонд; благодійний проєкт із цільовою сумою; записи пожертв із різними статусами; записи витрат із категоріями та статусами; бюджетний план із бюджетними рядками; фінансовий звіт за визначений період.

Це дало змогу перевірити зв'язок між користувачами, фондами, проєктами, пожертвами, витратами, бюджетами, звітами та аналітичними показниками.

Результати перевірки основних сценаріїв роботи вебсистеми FondUA подано в додатку Г. У додатку наведено перелік тестових сценаріїв, ролі користувачів, початкові умови, дії, очікувані результати, фактичні результати та статус виконання. У межах основного тексту доцільно подати узагальнений аналіз перевірки, оскільки детальна таблиця тест-кейсів має службовий характер і займає значний обсяг.

Наведені тест-кейси охоплюють основні функціональні блоки FondUA: accounts, funds, projects, finance, reports, analytics, рольовий доступ і публічний інтерфейс. Окрему увагу приділено сценаріям, які безпосередньо впливають на фінансову достовірність системи: створення пожертви, перевірка суми, створення витрати, зміна статусу витрати, формування бюджету, підготовка звіту та відображення аналітичних показників.

Перевірка рольового доступу є одним із ключових елементів оцінювання працездатності FondUA. Неавторизований користувач не повинен отримувати доступ до службових сторінок. Донор не повинен мати можливості змінювати фонд, проєкти, витрати, бюджетні плани або звіти. Представник фонду повинен працювати лише з власним фондом і пов'язаними з ним проєктами. Суперадміністратор виконує контрольні дії відповідно до реалізованої моделі доступу. Така перевірка пов'язана з ризиками порушення контролю доступу й автентифікації, які належать до важливих ризиків вебзастосунків за OWASP [7; 8; 9].

Фінансова логіка перевірялася через контроль статусів пожертв, витрат і фінансових звітів. До публічних фінансових показників мають потрапляти лише завершені пожертви, а витрати зі статусом очікування або відхилення не повинні зменшувати залишок коштів проєкту. Для бюджетування перевіряється відповідність фактичної суми підтвердженням витратам за категорією та періодом. Для фінансового звіту перевіряється відповідність підсумків обраному звітному періоду та коректність переходів між службовими статусами. Такі

сценарії підтверджують, що FondUA працює не лише як набір сторінок, а як система управління фінансовими даними.

Перевірка інтерфейсу охоплювала заповнення форм, повідомлення про помилки, повідомлення про успішні дії, відображення таблиць і доступність основних сценаріїв. Для форм пожертви та витрати важливо, щоб система не приймала порожні обов'язкові поля, нульові або від'ємні суми. Для сторінок звітності та аналітики важливо, щоб дані подавалися послідовно й не суперечили фінансовим записам. Перевірка інтерфейсу поєднує функціональне тестування і базову оцінку зручності роботи з формами та сторінками [19].

Повідомлення про помилку введення даних наведено на рис. 3.13.

Рис. 3.13. Повідомлення про помилку введення даних у веб-системі FondUA
(Розроблено автором)

Повідомлення про помилку підтверджує, що система реагує на некоректні вхідні дані й не дозволяє зберегти запис без виконання умов валідації. Це особливо важливо для фінансових форм, де нульові, від'ємні або порожні значення можуть призвести до некоректних розрахунків.

Зміну статусу фінансового запису після перевірки подано на рис. 3.14.

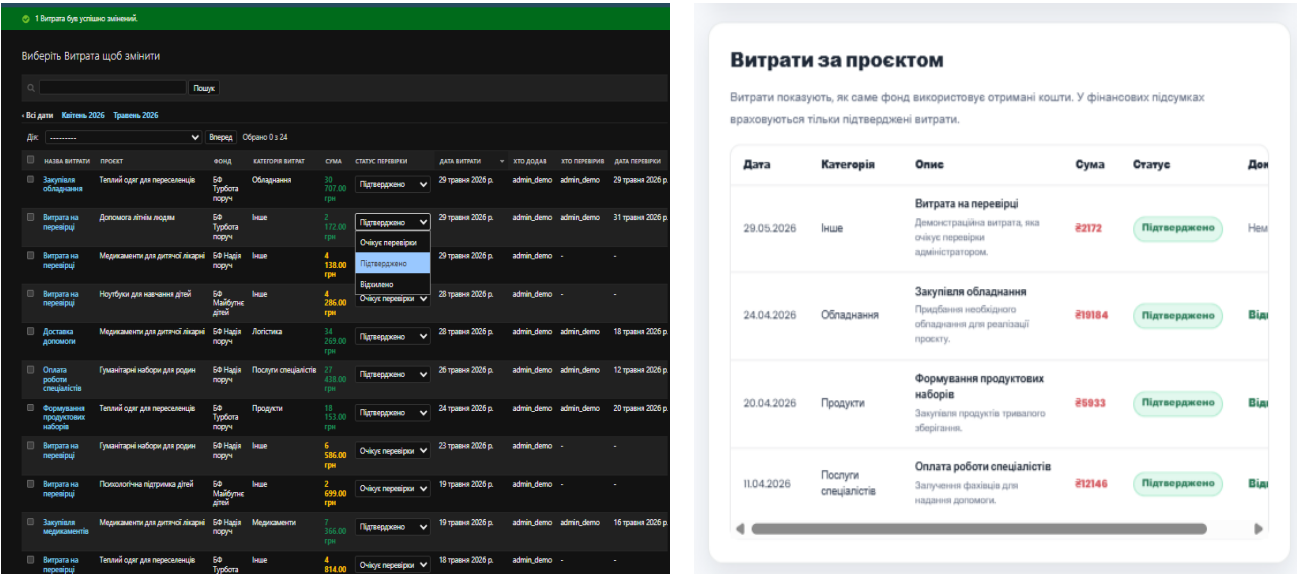
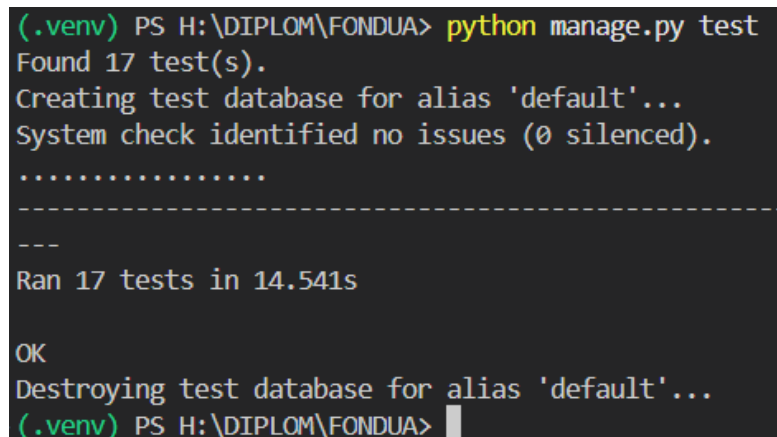


Рис. 3.14. Зміна статусу фінансового запису після перевірки
(Розроблено автором)

Зміна статусу фінансового запису демонструє контрольний механізм FondUA. Такий сценарій потрібний для того, щоб до фінансових показників, звітів та аналітики потрапляли лише перевірені записи, а службові або відхилені дані не спотворювали підсумкові значення.

Автоматизована перевірка охопила тестові модулі, пов’язані з користувачами, проєктами, фінансовими операціями, звітами та аналітикою. У модулі accounts перевірено створення профілю донора, створення представника фонду з фондом у статусі pending, відхилення дублювання електронної адреси та автентифікацію за електронною поштою. У модулі projects перевірено різницю між формою проєкту для суперадміністратора і представника фонду. У модулі finance перевірено маршрути фінансового блоку, створення витрати представником фонду зі статусом pending, можливість редагування статусу витрати суперадміністратором і розрахунок бюджету лише за підтвердженими витратами. У модулі reports перевірено розрахунок балансу звіту, перехід між статусами звіту, публікацію, зняття з публікації, відхилення звіту та заборону некоректного звітного періоду. У модулі analytics перевірено прийняття коректного фільтра та відхилення періоду, у якому кінцева дата є ранішою за початкову.

Результат запуску автоматизованих тестів наведено на рис. 3.15.



```
(.venv) PS H:\DIPLOM\FONDUA> python manage.py test
Found 17 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
---
Ran 17 tests in 14.541s

OK
Destroying test database for alias 'default'...
(.venv) PS H:\DIPLOM\FONDUA>
```

Рис. 3.15. Результат запуску автоматизованих тестів вебсистеми FondUA

(Розроблено автором)

Результат запуску тестів підтверджує, що перевірка охоплює не лише інтерфейсні сценарії, а й програмну логіку окремих модулів. Це важливо для FondUA, оскільки фінансові показники залежать від правильного зв'язку між моделями, статусами, формами, правами доступу та правилами розрахунку.

Поточна версія FondUA є навчально-практичною реалізацією вебсистеми та має окремі обмеження. У системі не реалізовано повноцінну інтеграцію з банківським або платіжним сервісом, тому пожертви фіксуються як записи внутрішнього обліку. Також система використовує SQLite, що є достатнім для локальної демонстраційної версії, але для промислового використання доцільним було б перейти на PostgreSQL. Подальший розвиток може передбачати інтеграцію платіжного шлюзу, розширене журналювання дій користувачів, посилення аудиту фінансових операцій і покращення механізмів безпеки.

Після перевірки працездатності встановлено, що вебсистема FondUA виконує основні функції: підтримує реєстрацію й авторизацію, розмежовує ролі користувачів, відображає публічні фонди та проекти, фіксує пожертви, обробляє витрати, формує бюджетні плани, створює фінансові звіти й відображає аналітичні показники. Отримані результати підтверджують працездатність

реалізованого програмного забезпечення та відповідність основним функціональним вимогам системи.

Висновки до розділу 3

У третьому розділі розглянуто програмну реалізацію вебсистеми FondUA на основі Python і Django. Обґрунтовано вибір технологічного стеку, який забезпечує реалізацію серверної логіки, роботу з базою даних, побудову користувацького інтерфейсу, формування аналітики та експорт звітних даних.

Серверну частину системи реалізовано через взаємодію функціональних модулів для роботи з користувачами, фондами, благодійними проєктами, пожертвами, витратами, бюджетами, фінансовими звітами та аналітичними показниками. Реалізовано рольове розмежування доступу для відвідувачів, донорів, представників фондів і суперадміністраторів.

Клієнтська частина FondUA містить публічні сторінки фондів і проєктів, форми пожертв, службові сторінки управління фінансами, бюджетами та звітністю, а також аналітичну панель. Система забезпечує пов'язаний облік пожертв, витрат, бюджетних планів і фінансових звітів, що дає змогу контролювати використання коштів і формувати аналітичні показники

Перевірка працездатності підтвердила коректну роботу основних сценаріїв: автентифікації користувачів, роботи з проєктами, створення фінансових записів, формування бюджетів, звітів та відображення аналітики. Це свідчить про працездатність FondUA як цілісної вебсистеми управління коштами благодійного фонду.

Поточна реалізація має навчально-практичний характер і може бути розширена шляхом інтеграції платіжного шлюзу, переходу на серверну СКБД, удосконалення механізмів безпеки та розвитку аналітичних можливостей. Отримані результати підтверджують можливість використання FondUA для обліку пожертв, контролю витрат, бюджетування та підготовки фінансової звітності.

ВИСНОВКИ

У кваліфікаційній роботі спроектовано та реалізовано вебсистему FondUA для управління коштами благодійного фонду. Поставлену мету досягнуто, оскільки створена система забезпечує облік фондів, благодійних проєктів, пожертв, витрат, бюджетів, фінансових звітів та аналітичних показників, а також підтримує розмежування доступу між відвідувачем, донором, представником фонду та суперадміністратором.

У процесі виконання роботи проаналізовано проблемну область управління коштами благодійного фонду. Встановлено, що для благодійної організації недостатньо мати лише публічний сайт для розміщення зборів. Повноцінна цифрова підтримка має охоплювати не тільки залучення пожертв, а й контроль їх цільового використання, облік витрат, бюджетування, підготовку фінансової звітності, публічне подання результатів і захист службових даних. Аналіз сучасних вебрішень показав, що публічні платформи зборів, сайти фондів, CRM-системи, платіжні шлюзи та сервіси аналітики здебільшого закривають окремі частини цього процесу, але не формують єдиного фінансового контуру для благодійного фонду.

На основі аналізу предметної області визначено інформаційні потоки FondUA та сформульовано задачу проектування системи. У роботі обґрунтовано потребу в поєднанні публічної частини, доступної відвідувачам і донорам, зі службовою частиною, призначеною для представників фонду та суперадміністратора. Такий підхід дає змогу пов'язати фонд, благодійний проєкт, пожертву, витрату, бюджетний план, фінансовий звіт і аналітичний показник у межах єдиної моделі даних.

Сформовано інформаційну та математичну основу FondUA. Спроектовано загальну архітектуру системи, визначено її клієнт-серверну побудову, функціональні модулі, публічний і службовий контури, а також рольову модель доступу. Розроблено структуру бази даних, що охоплює користувачів, профілі, фонди, благодійні проєкти, пожертви, витрати, категорії витрат, бюджетні плани, бюджетні рядки та фінансові звіти. Для цих сутностей визначено призначення, основні поля, зв'язки та статуси, які впливають на доступність записів і фінансові розрахунки.

Окремо розроблено фінансову логіку системи. Вона охоплює розрахунок зібраної суми, підтверджених витрат, залишку коштів, прогресу збору, доступного балансу, план-фактного виконання бюджету та підсумкових показників фінансового звіту. У розрахунках враховано статуси пожертв, витрат і звітів, завдяки чому до фінансових підсумків потрапляють лише коректні та підтверджені записи. Це забезпечує узгодженість між обліком пожертв, контролем витрат, бюджетуванням, звітністю та аналітикою.

Програмну реалізацію FondUA виконано як Django-застосунок із модульною структурою. Реалізовано функціональні модулі accounts, funds, projects, finance, reports, analytics і core, які відповідають за користувачів, ролі, фонди, благодійні проекти, пожертви, витрати, бюджетування, фінансову звітність, аналітичні показники та загальні сторінки системи. Для розробки використано Python, Django, Django ORM, SQLite, HTML, CSS, JavaScript, Chart.js, pandas, openpyxl, Pillow, django-widget-tweaks, django-filter, Django staticfiles, WhiteNoise і python-dotenv.

Реалізована система підтримує основні сценарії роботи: перегляд фондів і благодійних проектів, реєстрацію й автентифікацію користувачів, фіксацію пожертв, створення та перевірку витрат, формування бюджетних планів, підготовку фінансових звітів, відображення аналітичних показників і експорт табличних даних. Для представника фонду передбачено службові сценарії роботи з власним фондом, проектами, витратами, бюджетами та звітами. Донор отримує доступ до публічних проектів і сценарію внесення пожертви. Суперадміністратор виконує контрольні дії, пов'язані зі статусами, фінансовими записами, звітами та коректністю даних.

Клієнтська частина FondUA забезпечує публічне подання інформації про фонди й благодійні проекти, доступ до форм пожертв, службових таблиць витрат, бюджетних сторінок, фінансових звітів та аналітичної панелі. Аналітичні показники подаються у вигляді числових значень, таблиць, графіків і діаграм. Це дає змогу оцінювати стан збору, суму підтверджених витрат, залишок коштів, структуру витрат і виконання бюджету без ручного зіставлення розрізнених таблиць або документів.

Працездатність FondUA перевірено за основними функціональними сценаріями. Перевірка охопила автентифікацію користувачів, рольовий доступ,

публічний перегляд фондів і проєктів, створення фінансових записів, валідацію форм, зміну статусів витрат і звітів, бюджетування, формування фінансової звітності, відображення аналітики та експорт даних. Результати перевірки підтвердили, що FondUA працює як цілісна вебсистема управління коштами благодійного фонду, а не як набір ізольованих сторінок.

Практична цінність отриманих результатів полягає в можливості використання FondUA як навчально-практичної основи для автоматизації фінансових процесів благодійного фонду. Система дає змогу впорядкувати облік пожертв і витрат, контролювати бюджети, готувати фінансові звіти, формувати аналітичні показники, підвищувати прозорість руху коштів і покращувати взаємодію з донорами. За рахунок зв'язку між проєктами, фінансовими операціями, бюджетами, звітами та аналітикою FondUA демонструє практичну можливість побудови цілісної системи управління коштами благодійної організації.

Поточна реалізація має окремі обмеження. У системі не реалізовано повноцінну інтеграцію з банківським або платіжним сервісом, тому пожертви фіксуються як записи внутрішнього обліку. Також у навчально-практичній версії використовується SQLite, що є достатнім для локальної демонстрації, але для промислового використання доцільним був би перехід на серверну систему керування базами даних. Подальший розвиток FondUA може передбачати інтеграцію з платіжним сервісом, розширене журналювання дій користувачів, посилення безпеки, удосконалення аналітичної панелі, покращення експорту звітних даних і гнучкіше налаштування прав доступу.

Отже, у межах кваліфікаційної роботи досягнуто поставленої мети: спроектовано та реалізовано вебсистему FondUA, яка поєднує публічне представлення благодійних проєктів із внутрішнім контуром фінансового обліку, бюджетування, звітності й аналітики. Отримані результати підтверджують доцільність використання такого підходу для підвищення прозорості та керованості фінансових процесів благодійного фонду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корнак О. О., Грицюк Ю. І. Проектування системи управління коштами благодійного фонду // Scientific Bulletin of UNFU. 2021. Vol. 31(2). Р. 121–132. URL: <https://doi.org/10.36930/40310320> (дата звернення: 23.05.2026).
2. Romaniv Y., Shvorb I. Формування вимог для розробки інтернет-платформи для здійснення благодійних пожертв // Trends and directions of development of scientific approaches and prospects of integration of internet technologies into society. 2021. Р. 569–574.
3. Про благодійну діяльність та благодійні організації : Закон України від 05.07.2012 № 5073-VI. URL: <https://zakon.rada.gov.ua/laws/show/5073-17> (дата звернення: 23.05.2026).
4. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 23.05.2026).
5. Python Software Foundation. Django documentation. Version 5.2. URL: <https://docs.djangoproject.com/en/5.2/> (дата звернення: 23.05.2026).
6. SQLite Documentation. URL: <https://www.sqlite.org/docs.html> (дата звернення: 08.06.2026).
7. OWASP Foundation. OWASP Top 10:2021. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 23.05.2026).
8. OWASP Foundation. A01:2021 – Broken Access Control. URL: https://owasp.org/Top10/2021/A01_2021-Broken_Access_Control/ (дата звернення: 23.05.2026).
9. OWASP Foundation. A07:2021 – Identification and Authentication Failures. URL: https://owasp.org/Top10/2021/A07_2021-Identification_and_Authentication_Failures/ (дата звернення: 23.05.2026).
10. Python 3.10.20 documentation. URL: <https://docs.python.org/3.10/> (дата звернення: 08.06.2026).

11. Models | Django documentation. Version 5.2. URL: <https://docs.djangoproject.com/en/5.2/topics/db/models/> (дата звернення: 08.06.2026).
12. Using the Django authentication system | Django documentation. Version 5.2. URL: <https://docs.djangoproject.com/en/5.2/topics/auth/default/> (дата звернення: 08.06.2026).
13. Григор'єв О. В. Основи веб-програмування. Київ : Каравела, 2021. 312 с.
14. Konoplytskyi I. O., Kravchenko K. O. Web-programuvannia: pidruchnyk. Kyiv : KNEU, 2021. 412 p.
15. Gordienko D. M., Zhukov S. V. Osnovy proektuvannia informatsiinykh system. Kyiv : Tsentr uchebnoi literatury, 2020. 264 p.
16. HTML Living Standard. URL: <https://html.spec.whatwg.org/> (дата звернення: 08.06.2026).
17. Chart.js Documentation. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 08.06.2026).
18. Білець Д. Ю., Мельникова Я. О. Текст як ключовий елемент для сайту благодійної організації // Поліграфічні, мультимедійні та web-технології. 2025. Т. 2. С. 253–254.
19. Дейнеко Ж. В., Менделєва М. В. Методика тестування інтерфейсів сайтів на основі функціонального та юзабіліті тестування // Поліграфічні, мультимедійні та web-технології. 2024. С. 187–188.
20. Бородаєв Д. Модульна сітка як композиційна основа веб-сайта // Вісник Харківської державної академії дизайну і мистецтв. 2004. № 5. С. 108–116.
21. pandas documentation. Version 2.2.3. URL: <https://pandas.pydata.org/pandas-docs/version/2.2.3/> (дата звернення: 08.06.2026).
22. openpyxl documentation. Version 3.1.3. URL: <https://openpyxl.readthedocs.io/en/stable/> (дата звернення: 08.06.2026).
23. Pillow documentation. URL: <https://pillow.readthedocs.io/en/stable/> (дата звернення: 08.06.2026).

- 24.WhiteNoise documentation. Version 6.12.0. URL: <https://whitenoise.readthedocs.io/en/stable/> (дата звернення: 08.06.2026).
- 25.python-dotenv documentation. URL: <https://bbc2.github.io/python-dotenv/> (дата звернення: 08.06.2026).
- 26.django-filter documentation. Version 25.2. URL: <https://django-filter.readthedocs.io/en/stable/> (дата звернення: 08.06.2026).
- 27.django-widget-tweaks. Version 1.5.1. URL: <https://pypi.org/project/django-widget-tweaks/> (дата звернення: 08.06.2026).

Рис. А.1 – Сторінка благодійного проєкту у вебсистемі FondUA

[illegible]

Рис. Б.1 – Сторінка бюджетного плану благодійного проєкту

Повний вигляд сторінки фінансової звітності FondUA

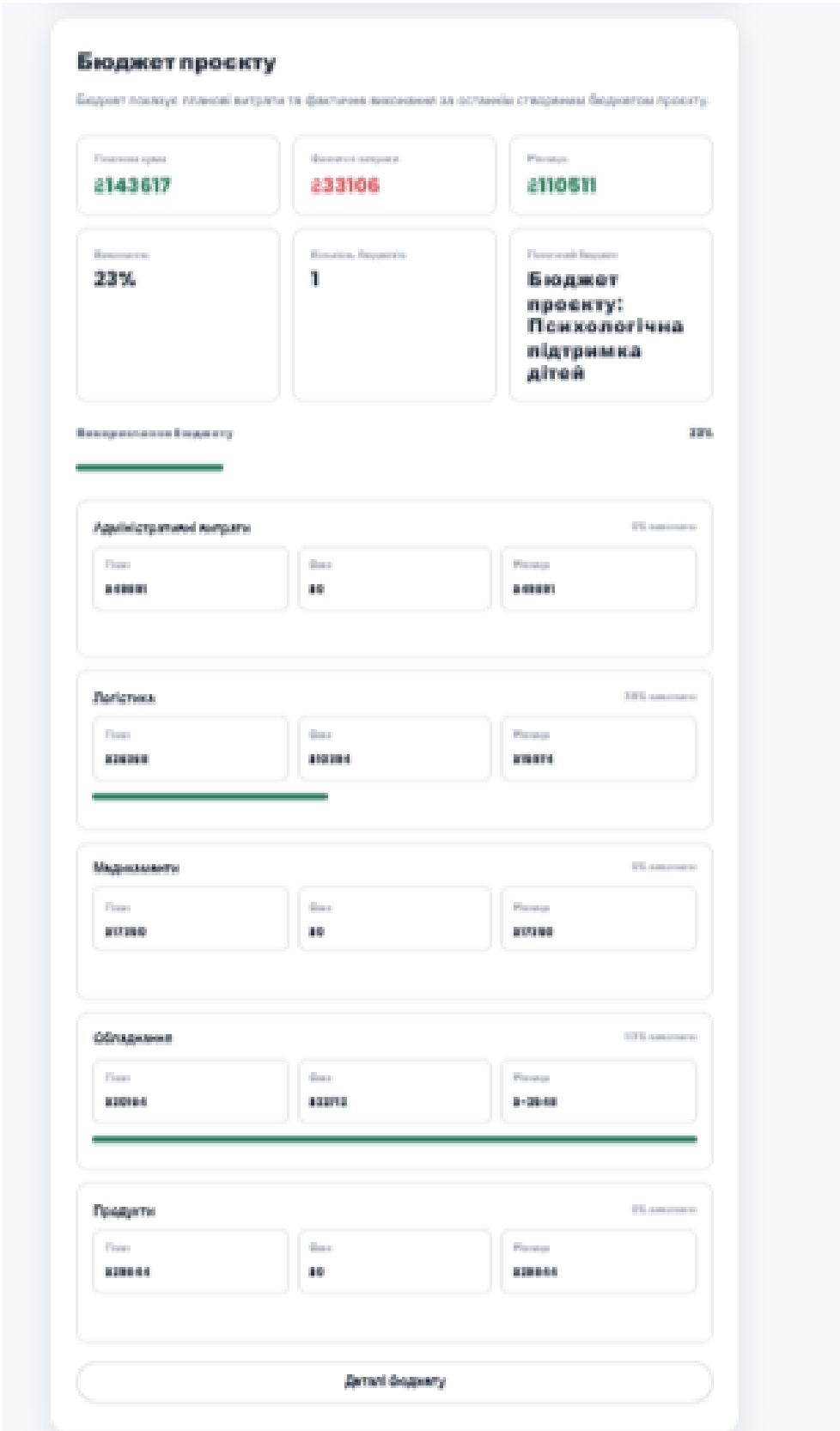


Рис. В.1 – Сторінка фінансової звітності у вебсистемі FondUA

Повний вигляд аналітичної панелі вебсистеми FondUA

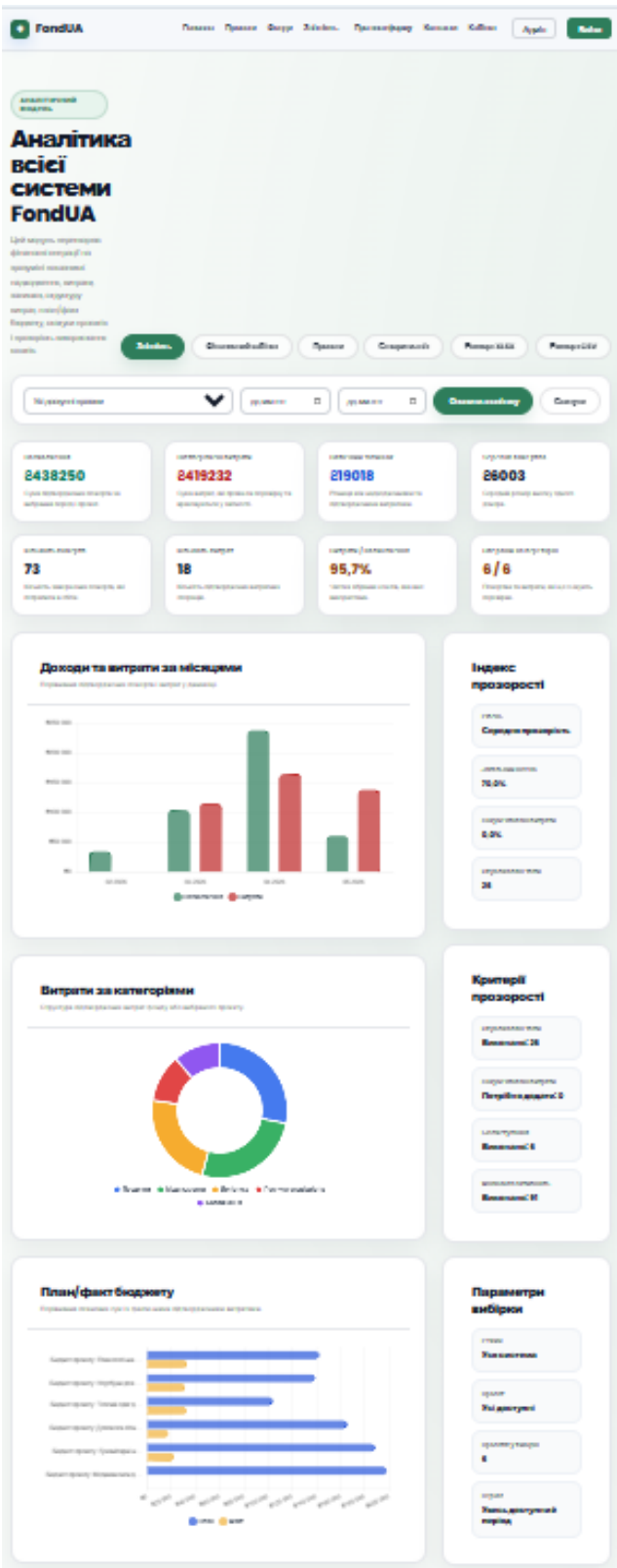


Рис. Г.1 – Аналітична панель вебсистеми FondUA

Додаток Д

Результати перевірки працездатності основних функцій FondUA

Сценарій перевірки	Роль користувача	Початкові умови	Дії	Очікуваний результат	Фактичний результат
1	2	3	4	5	6
Реєстрація донора	Донор	Користувач не має облікового запису	Заповнити форму реєстрації з типом облікового запису донора	Створюється обліковий запис і профіль із роллю donor	Обліковий запис створено, профіль користувача отримав роль донора, фонд для користувача не створюється
Реєстрація представника фонду	Представник фонду	Користувач не має облікового запису	Заповнити форму реєстрації з даними фонду	Створюється користувач із роллю fund_manager і фонд зі статусом очікування	Створено профіль представника фонду, пов'язаний фонд отримав статус pending
Перевірка дублювання електронної адреси	Донор / представник фонду	У системі вже є користувач із такою електронною адресою	Повторно ввести ту саму електронну адресу у формі реєстрації	Форма не проходить валідацію	Система відхиляє реєстрацію та показує помилку для поля електронної адреси
Вхід до системи з коректними даними	Авторизована роль	Обліковий запис існує	Ввести правильну електронну адресу та пароль	Користувач входить до системи	Форма входу приймає коректні дані, користувач автентифікується за електронною адресою
Вхід до системи з неправильними даними	Будь-який користувач	Обліковий запис існує	Ввести неправильний пароль	Система не виконує вхід і показує повідомлення про помилку	Вхід не виконується, користувач залишається неавторизованим

Продовження табл. Д.

1	2	3	4	5	6
Доступ до службової сторінки без авторизації	Відвідувач	Користувач не авторизований	Відкрити сторінку створення фонду, витрати, бюджету або звіту	Доступ блокується або виконується перенаправлення на сторінку входу	Службові сторінки недоступні неавторизованому користувачу
Перегляд публічних фондів і проєктів	Відвідувач	У системі є підтверджені фонди та активні проєкти	Відкрити список фондів і сторінку проєкту	Відображаються публічні дані без службових дій редагування	Користувач бачить відкрити інформацію про фонд, проєкт, цільову суму, зібрані кошти й прогрес збору
Створення благодійного фонду	Представник фонду	Користувач авторизований як представник фонду	Заповнити форму фонду	Створюється фонд, пов'язаний із користувачем	Фонд створюється та прив'язується до користувача через зв'язок owner
Спроба редагування чужого фонду	Представник фонду	У системі є фонд іншого представника	Спробувати відкрити або змінити чужий фонд	Система обмежує доступ	Представник фонду не отримує доступу до редагування чужої організації
Створення благодійного проєкту	Представник фонду	У користувача є власний фонд	Заповнити назву, опис, категорію, цільову суму та статус проєкту	Проєкт створюється та прив'язується до фонду користувача	Проєкт зберігається у базі даних і пов'язується з відповідним фондом

Продовження табл. Д.

1	2	3	4	5	6
Відображення поля фонду у формі проєкту	Суперадміністратор / представник фонду	У системі є фонд і проєкт	Відкрити форму створення або редагування проєкту	Суперадміністратор бачить поле вибору фонду, представник фонду працює лише зі своїм фондом	Для суперадміністратора поле фонду доступне, для представника фонду воно приховане
Створення пожертви з коректною сумою	Донор / відвідувач	У системі є активний благодійний проєкт	Ввести додатну суму, валюту, спосіб оплати та підтвердити дію	Створюється запис пожертви, пов'язаний із проєктом	Пожертва зберігається як запис Donation, після чого користувач переходить на сторінку підтвердження
Валідація некоректної суми пожертви	Донор / відвідувач	Відкрита форма пожертви	Ввести порожнє, нульове або від'ємне значення суми	Форма не повинна зберігати некоректний запис	Система показує повідомлення про помилку введення даних і не створює фінансовий запис
Створення витрати представником фонду	Представник фонду	У користувача є фонд і проєкт	Заповнити форму витрати з сумою, категорією, датою та описом	Створюється витрата зі статусом очікування	Витрата зберігається зі статусом pending, без автоматичного включення до підтверджених витрат
Зміна статусу витрати	Суперадміністратор	У системі є витрата зі статусом pending	Переглянути витрату та змінити її статус	Витрата переходить у стан approved або rejected	Суперадміністратор може погодити або відхилити витрату, а підтверджені витрати враховуються у фінансових показниках

Продовження табл. Д.

1	2	3	4	5	6
Розрахунок бюджету за підтвердженими витратами	Представник фонду / суперадміністратор	Є бюджетний план, бюджетний рядок і витрати з різними статусами	Порівняти планову суму з фактичними витратами	Фактична сума має враховувати лише погоджені витрати	План-фактне порівняння виконується на основі витрат зі статусом approved, непогоджені витрати не включаються
Створення фінансового звіту	Представник фонду	Є фонд, проєкт і фінансові записи за період	Створити звіт за визначений період	Формується звіт із надходженнями, витратами та залишком	Звіт створюється, баланс обчислюється як різниця між надходженнями та витратами
Workflow фінансового звіту	Представник фонду / суперадміністратор	Існує створений фінансовий звіт	Подати, погодити, опублікувати або зняти з публікації звіт	Статус звіту змінюється відповідно до виконаної дії	Звіт переходить між станами submitted, approved, published; після зняття з публікації перестає бути публічним
Відхилення фінансового звіту	Суперадміністратор	Існує фінансовий звіт, який потребує правок	Відхилити звіт із зазначенням причини	Звіт отримує статус rejected і не відображається як публічний	Причина відхилення зберігається, звіт залишається непублічним
Перевірка некоректного звітного періоду	Представник фонду	Відкрита форма фінансового звіту	Вказати дату завершення раніше за дату початку	Система не повинна приймати некоректний період	Запис із некоректним періодом не проходить перевірку

Продовження табл. Д.

1	2	3	4	5	6
Фільтрація аналітичних даних	Представник фонду / суперадміністратор	У системі є проекти та фінансові записи	Вибрати проєкт і період для аналітики	Система формує показники за обраними параметрами	Коректний фільтр приймається, некоректний період із датою завершення раніше за дату початку відхиляється
Запуск автоматизованих тестів	Системна перевірка	У проєкті наявні тестові модулі accounts, projects, finance, reports, analytics	Запустити автоматизовані тести Django	Тести мають завершитися без помилок	Виконано 17 автоматизованих тестів, помилок не виявлено

ЗГОДА здобувачки вищої освіти

Державного університету економіки і технологій про
перевірку кваліфікаційної роботи на прояви академічного
плагіату
та розміщення в Репозитарії Університету

Я, Горбенко Анастасія Олександрівна,

підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота

«Розробка вебсистеми управління коштами благодійного фонду»

виконана самостійно та не містить академічного плагіату. Я не надавала і не одержувала недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлена. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформована, що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомлена.

15.06.26 р

Горбенко А. О.