

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри інформаційних
систем і технологій

_____ **Ігор БОЛБОТ**
(підпис)

_____ **Михайло ШВИДЕНКО**
(підпис)

«___» _____ 2026 р.

«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Веб-портал ведення особистого фінансового обліку»

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми

к.т.н., доцент

_____ **Роман РУДЕНСЬКИЙ**
(підпис)

**Керівник бакалаврської
кваліфікаційної роботи**

Доктор філософії, доцент

_____ **Валентина КОРОЛЬЧУК**
(підпис)

Виконав

_____ **Сергій ЖЕЛЄЗНЯКОВ**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і
технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО
(підпис)

«___» _____ 2026 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Желєзнякову Сергію Володимировичу
(прізвище, ім'я, по батькові)

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Тема бакалаврської кваліфікаційної роботи

«Веб-портал ведення особистого фінансового обліку»

затверджена наказом від «6» січня 2026 р. № 46 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):
методології аналізу та проєктування програмних систем, об'єктно-орієнтований підхід,
UML-моделювання, методи побудови баз даних, вимоги до веб-порталу особистого
фінансового обліку.

Перелік питань, що підлягають дослідженню:

аналіз предметної області та існуючих рішень; визначення вимог до системи;
моделювання системи за допомогою UML та ER-діаграм; проєктування бази даних;
вибір технологій; реалізація, тестування та впровадження веб-порталу.

Дата видачі завдання «6» січня 2026 р.

Керівник бакалаврської
кваліфікаційної роботи

_____ **Валентина КОРОЛЬЧУК**
(підпис)

Завдання взяв до
виконання

_____ **Сергій ЖЕЛЄЗНЯКОВ**
(підпис)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП.....	6
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Опис предметної області	9
1.2 Огляд інформаційних джерел та існуючих рішень	13
1.3 Аналіз вимог програмної системи	18
1.4 Моделювання предметної області	22
1.5 Постановка завдання	25
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-ПОРТАЛУ	28
2.1 Логічна модель даних у вигляді ER-діаграми	28
2.2 Фізична модель даних і структура бази даних	32
2.3 Діаграма компонентів і пакетна структура	35
2.4 Алгоритмічне забезпечення фінансової аналітики	37
2.5 Висновки до другого розділу	40
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-ПОРТАЛУ	42
3.1 Обґрунтування вибору технологій.....	42
3.2 Представлення архітектури системи	43
3.3 Висновки до третього розділу	45
4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ	47
4.1 План тестування програмних модулів.....	47
4.2 Тестування системи.....	48
4.3 Розгортання системи, склад інсталяційного пакету	53
4.4 Висновки до четвертого розділу	55
ВИСНОВКИ	56
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК А	64

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- API - програмний інтерфейс прикладного програмування
- CSS - каскадні таблиці стилів
- CSV - формат табличних даних із розділенням значень комами
- DB - база даних
- ER - модель «сутність-зв'язок»
- FK - зовнішній ключ
- HTML - мова розмітки гіпертексту
- HTTP - протокол передавання гіпертексту
- HTTPS - захищений протокол передавання гіпертексту
- ID - унікальний ідентифікатор запису
- JSON - текстовий формат обміну даними
- MySQL - система керування реляційними базами даних
- Node.js - серверне середовище виконання JavaScript
- PDF - формат електронного документа
- PK - первинний ключ
- REST - архітектурний стиль взаємодії вебсервісів
- SQL - мова структурованих запитів
- TCP/IP - набір мережевих протоколів передавання даних
- UI - користувацький інтерфейс
- UML - уніфікована мова моделювання
- URL - уніфікована адреса ресурсу
- UX - досвід користувача під час взаємодії із системою
- Web - вебсередовище доступу до інформаційних ресурсів через браузер
- БД - база даних
- ІС - інформаційна система
- ПЗ - програмне забезпечення

- СКБД - система керування базами даних
- ФО - фінансова операція

ВСТУП

Щоденна діяльність людини супроводжується виконанням великої кількості фінансових операцій, пов'язаних із отриманням доходів, здійсненням витрат, плануванням бюджету та аналізом фінансового стану. У зв'язку з цим виникає потреба у використанні сучасних інформаційних інструментів, які забезпечують систематизацію фінансових даних, їх облік, збереження та подальший аналіз. Застосування веборієнтованих інформаційних систем у даній сфері дозволяє організувати централізоване зберігання фінансової інформації, забезпечити зручний доступ до неї через мережу Інтернет та створити умови для ефективної обробки фінансових показників користувача.

Розвиток технологій веб-програмування та інтернет-сервісів відкриває нові можливості для створення інтерактивних програмних рішень, спрямованих на підтримку процесів ведення особистого фінансового обліку. Такі системи надають користувачам інструменти для фіксації фінансових операцій, класифікації доходів і витрат, формування статистичних показників та отримання аналітичної інформації щодо структури власних фінансів. Реалізація подібного функціоналу у вигляді веб-порталу дозволяє забезпечити постійний доступ до системи з різних пристроїв, підвищує зручність використання програмного забезпечення та спрощує процес його оновлення й подальшого розвитку. Завдяки цьому веб-портали фінансового обліку виступають ефективним інструментом для підтримки прийняття фінансових рішень та підвищення рівня фінансової дисципліни користувачів.

Актуальність теми зумовлена необхідністю створення зручних та функціональних програмних рішень для ведення особистого фінансового обліку. Незважаючи на наявність різноманітних фінансових сервісів, значна частина з них має обмежені можливості персоналізації, складний інтерфейс або орієнтована переважно на мобільні застосунки, що знижує ефективність їх використання у веб-середовищі. Крім того, не всі існуючі системи забезпечують

інтегровані механізми аналізу фінансових операцій, планування бюджету та формування аналітичних звітів у межах єдиної інформаційної платформи. У зв'язку з цим розробка веб-порталу ведення особистого фінансового обліку, що поєднує функції управління фінансовими операціями, аналітики та планування бюджету, є актуальним напрямом дослідження.

Підставами для розробки даної тематики є зростання потреби користувачів у контролі власних фінансових ресурсів, розвиток технологій веб-програмування та широке використання інформаційних систем для автоматизації фінансових процесів. Вихідними даними для реалізації проєкту стали сучасні методології аналізу та проєктування програмних систем, зокрема використання об'єктно-орієнтованого підходу та мови моделювання UML для опису архітектури програмного забезпечення, а також застосування методів моделювання даних для побудови структур баз даних інформаційної системи.

Завданнями кваліфікаційної роботи є:

1. виконати системний аналіз предметної області ведення особистого фінансового обліку та визначити основні функціональні характеристики програмної системи;
2. проаналізувати вимоги до інформаційної системи та сформувати їх специфікацію;
3. здійснити моделювання предметної області із використанням UML-діаграм та ER-моделей;
4. провести огляд існуючих програмних рішень у сфері систем управління особистими фінансами;
5. розробити логічну модель даних та побудувати основні діаграми проєктування системи (діаграму класів, пакетів, компонентів);
6. обґрунтувати вибір технологічного інструментарію та реалізувати інформаційне і програмне забезпечення веб-порталу, включаючи базу даних і програмні модулі;
7. надати рекомендації щодо впровадження, тестування та подальшої експлуатації розробленої інформаційної системи.

Метою дослідження є розробка концепції та реалізація програмного забезпечення веб-порталу ведення особистого фінансового обліку, що забезпечує зручне управління фінансовими операціями, планування бюджету, аналіз доходів і витрат та формування аналітичних звітів для підтримки прийняття фінансових рішень користувачем.

Об'єктом дослідження є процеси проєктування та розробки веб-орієнтованих інформаційних систем управління персональними фінансовими даними.

Предметом дослідження виступають методи та засоби створення веб-порталів для обліку та аналізу фінансових операцій, включаючи моделювання структури даних, розробку архітектури програмної системи та забезпечення ефективної взаємодії користувача із системою.

Наукова новизна роботи полягає у розробці структурованого підходу до створення веб-порталу персонального фінансового обліку, що поєднує механізми ведення фінансових операцій, категоризації доходів і витрат, планування бюджету та формування аналітичних звітів у межах єдиної інтегрованої інформаційної системи. Запропоновано архітектурні рішення, спрямовані на забезпечення масштабованості системи, централізованого зберігання фінансових даних та ефективної обробки інформації для підтримки фінансового аналізу користувача.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

У межах даної предметної області розглядаються дії, що забезпечують облік доходів і витрат, класифікацію фінансових операцій за категоріями, планування бюджету, перегляд балансу, формування історії операцій, а також побудову звітів і аналітичних відомостей. Практична значущість такої системи полягає в тому, що вона дозволяє перевести процес контролю особистих фінансів із фрагментарного або ручного ведення у впорядковане цифрове середовище, у якому фінансова інформація накопичується, структурується та використовується для подальшого аналізу.

Особливістю даної предметної області є те, що її основу становлять не лише окремі фінансові записи, а логічно пов'язані між собою інформаційні процеси. Користувач у межах системи не просто фіксує дохід або витрату, а формує цілісну модель власної фінансової поведінки, у якій кожна операція має суму, дату, категорію, опис, а також впливає на загальний баланс, бюджетні обмеження та аналітичні показники. За такого підходу портал виконує функцію не електронного блокнота, а інтегрованого інформаційного засобу підтримки фінансового самоконтролю, що забезпечує цілісне представлення фінансової активності користувача.

На рис. 1.1 подано контекстне уявлення предметної області у вигляді DFD 0-го рівня, яке відображає загальну взаємодію основних зовнішніх сутностей із веб-порталом ведення особистого фінансового обліку.

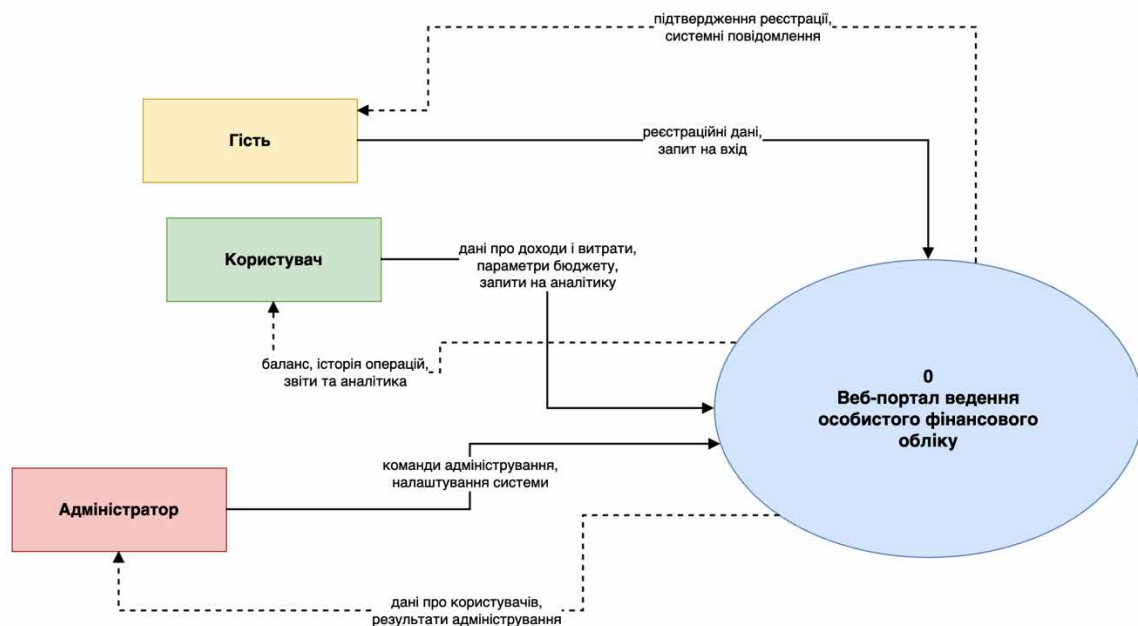


Рис. 1.1. Контекстна DFD 0-го рівня веб-порталу ведення особистого фінансового обліку

Як показано на рис. 1.1, у межах предметної області виділено три основні зовнішні ролі: гість, користувач і адміністратор. Гість взаємодіє із системою на початковому етапі, коли виконується реєстрація або вхід до облікового запису. Користувач є основним суб'єктом предметної області, оскільки саме він формує масив фінансових даних, вводить відомості про доходи та витрати, задає параметри бюджету, ініціює перегляд аналітики та використовує результати обробки даних для оцінювання власного фінансового стану. Адміністратор, у свою чергу, забезпечує підтримку нормативного й організаційного функціонування системи, виконуючи керування користувачами та системними параметрами. Наведена схема акцентує увагу не на внутрішній програмній реалізації, а на логіці інформаційного обміну в предметній області, що є доцільним для етапу системного аналізу.

З точки зору змісту предметної області ключовим об'єктом є фінансова операція, яка відображає факт надходження або витрачання коштів. Сукупність таких операцій формує основу для розрахунку поточного балансу, побудови історії змін фінансового стану та визначення структури витрат і доходів за обраними категоріями. Не менш важливим компонентом предметної області є

бюджетне планування, оскільки воно дає змогу встановлювати фінансові орієнтири, контролювати відхилення фактичних показників від запланованих та виявляти перевищення допустимого рівня витрат. Таким чином, предметна область поєднує як облікові, так і аналітичні процеси, а її цифровізація спрямована на підвищення керованості особистих фінансів.

Важливою характеристикою предметної області є наявність класифікаційних та довідникових сутностей, без яких неможливо забезпечити впорядковане ведення фінансової інформації. Зокрема, категорії доходів і витрат використовуються для групування операцій, підвищення змістовності аналітики та формування зведених звітів. Профіль користувача визначає персональний контекст функціонування системи, тоді як звіти та аналітичні представлення є похідними інформаційними об'єктами, що створюються на основі накопичених фінансових записів. Узагальнену характеристику основних об'єктів предметної області наведено в табл. 1.1.

Таблиця 1.1

Основні об'єкти предметної області веб-порталу ведення особистого фінансового обліку

Об'єкт предметної області	Призначення	Основний зміст
Користувач	Формує та використовує персональний фінансовий простір системи	Реєстраційні дані, обліковий запис, параметри профілю
Фінансова операція	Відображає факт надходження або витрачання коштів	Тип операції, сума, дата, категорія, опис
Категорія	Забезпечує структурування фінансових записів	Назва категорії, тип, ознака активності
Бюджет	Використовується для планування та контролю фінансових меж	Ліміт, період дії, пов'язані категорії або напрями витрат
Баланс	Характеризує поточний фінансовий стан користувача	Підсумок доходів, підсумок витрат, залишок коштів

Продовження таблиці 1.1

Історія операцій	Забезпечує хронологічне подання фінансової активності	Набір операцій за період із можливістю перегляду та аналізу
Звіт та аналітика	Підтримує оцінювання фінансової поведінки користувача	Зведення за категоріями, динаміка показників, аналітичні вибірки
Системні налаштування	Забезпечують адаптацію та адміністрування середовища	Параметри доступу, службові конфігурації, керування користувачами

Дані, наведені в табл. 1.1, свідчать про те, що предметна область має чітко виражену інформаційно-аналітичну природу. Її структура формується навколо персональних фінансових даних користувача, однак ефективність системи визначається не лише наявністю механізму введення операцій, а й можливістю подальшого узагальнення, контролю та інтерпретації накопиченої інформації. Саме тому для даної предметної області принципово важливими є цілісність даних, їх логічна структурованість, зручність введення та швидкість отримання аналітичного результату.

Отже, предметна область веб-порталу ведення особистого фінансового обліку характеризується наявністю взаємопов'язаних процесів реєстрації користувачів, ведення фінансових операцій, керування категоріями, бюджетного планування та аналітичної обробки даних. Її специфіка полягає в поєднанні облікових функцій із засобами аналізу та контролю, що обумовлює потребу у створенні інформаційної системи з упорядкованою структурою даних, зрозумілою логікою взаємодії користувачів і можливістю формування змістовних фінансових висновків на основі накопиченої інформації.

1.2 Огляд інформаційних джерел та існуючих рішень

У процесі дослідження предметної області важливим етапом є аналіз сучасних інформаційних джерел і програмних рішень, що реалізують функції управління особистими фінансами. Подібні системи спрямовані на автоматизацію обліку фінансових операцій, планування бюджету, аналіз структури витрат і доходів, а також формування аналітичних звітів. Вивчення існуючих програмних продуктів дозволяє визначити їх функціональні можливості, виявити переваги та обмеження, а також сформулювати вимоги до розроблюваної інформаційної системи.

Одним із найбільш відомих сервісів управління особистими фінансами є система Mint, яка призначена для автоматичного агрегування фінансових даних користувача та формування аналітичної інформації щодо структури його витрат. Система надає можливість синхронізації з банківськими рахунками, категоризації фінансових операцій та відображення фінансової статистики у вигляді графіків і таблиць. Загальний вигляд інтерфейсу системи Mint наведено на рис. 1.2.

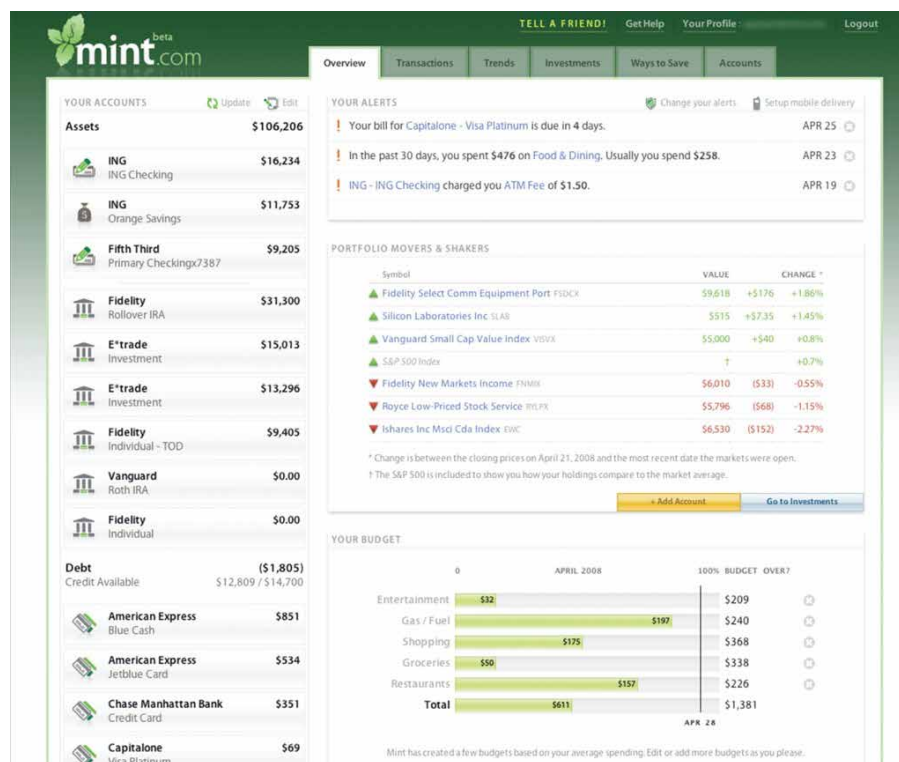


Рис. 1.2. Інтерфейс системи управління особистими фінансами Mint

Іншим популярним програмним рішенням є система YNAB (You Need a Budget), основною метою якої є допомога користувачам у плануванні особистого бюджету та контролі витрат відповідно до фінансових цілей. Дана система реалізує концепцію попереднього розподілу доходів за категоріями витрат і надає інструменти для формування детальних звітів щодо фінансової активності користувача. Приклад інтерфейсу системи YNAB наведено на рис. 1.3.



Рис. 1.3. Інтерфейс системи бюджетного планування YNAB

Серед сучасних мобільних сервісів для управління фінансами слід відзначити PocketGuard, який орієнтований на швидкий аналіз поточного фінансового стану користувача. Система забезпечує автоматичний облік транзакцій, формування категорій витрат, а також надання рекомендацій щодо оптимізації бюджету. Візуальні інструменти системи дозволяють відображати структуру витрат у зручному графічному вигляді. Приклад інтерфейсу системи PocketGuard наведено на рис. 1.4.

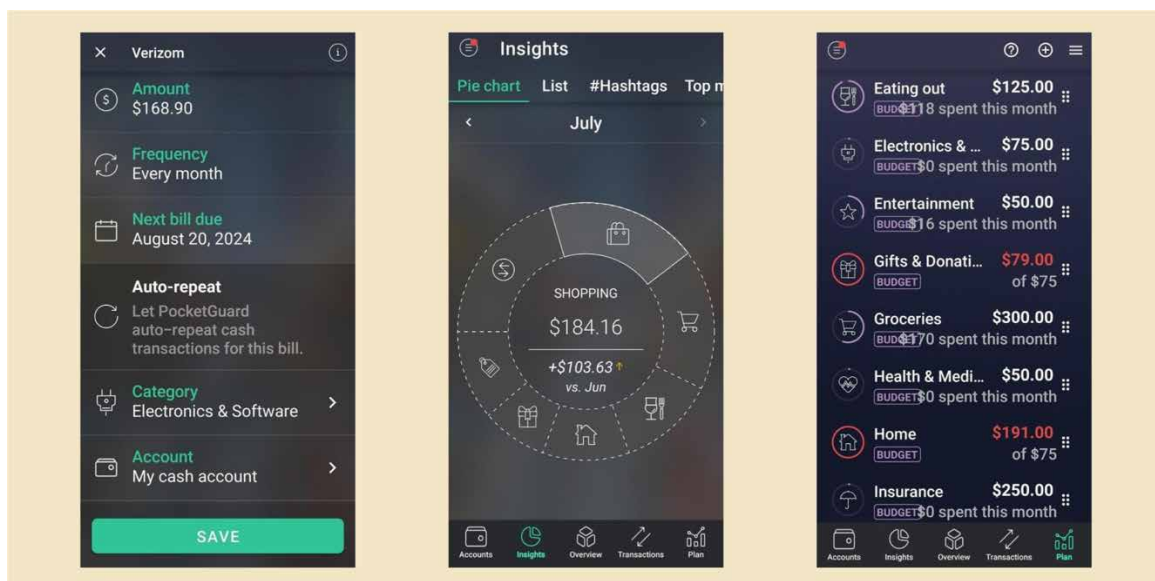


Рис. 1.4. Інтерфейс системи управління фінансами PocketGuard

Ще одним прикладом сучасного програмного рішення є Quicken Simplifi, що призначений для комплексного управління персональними фінансами та формування детальної аналітики фінансових показників. Система підтримує інтеграцію з банківськими рахунками, відстеження змін фінансового стану користувача та побудову статистичних звітів щодо динаміки доходів і витрат. Загальний вигляд інтерфейсу системи наведено на рис. 1.5.

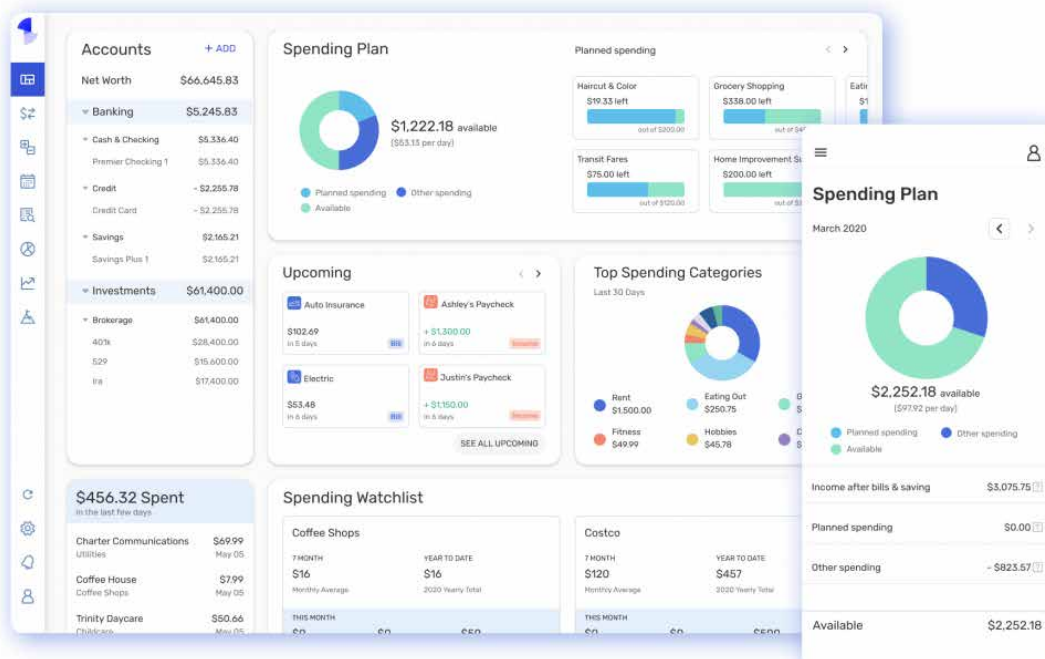


Рис. 1.5. Інтерфейс системи управління особистими фінансами Quicken Simplifi

До сучасних рішень у сфері управління персональними фінансами також належить Monarch Money, що поєднує функції фінансового планування, аналізу витрат та формування інвестиційної аналітики. Система надає користувачам інструменти для аналізу структури активів, планування фінансових цілей та формування довгострокових фінансових стратегій. Приклад інтерфейсу системи Monarch Money наведено на рис. 1.6.

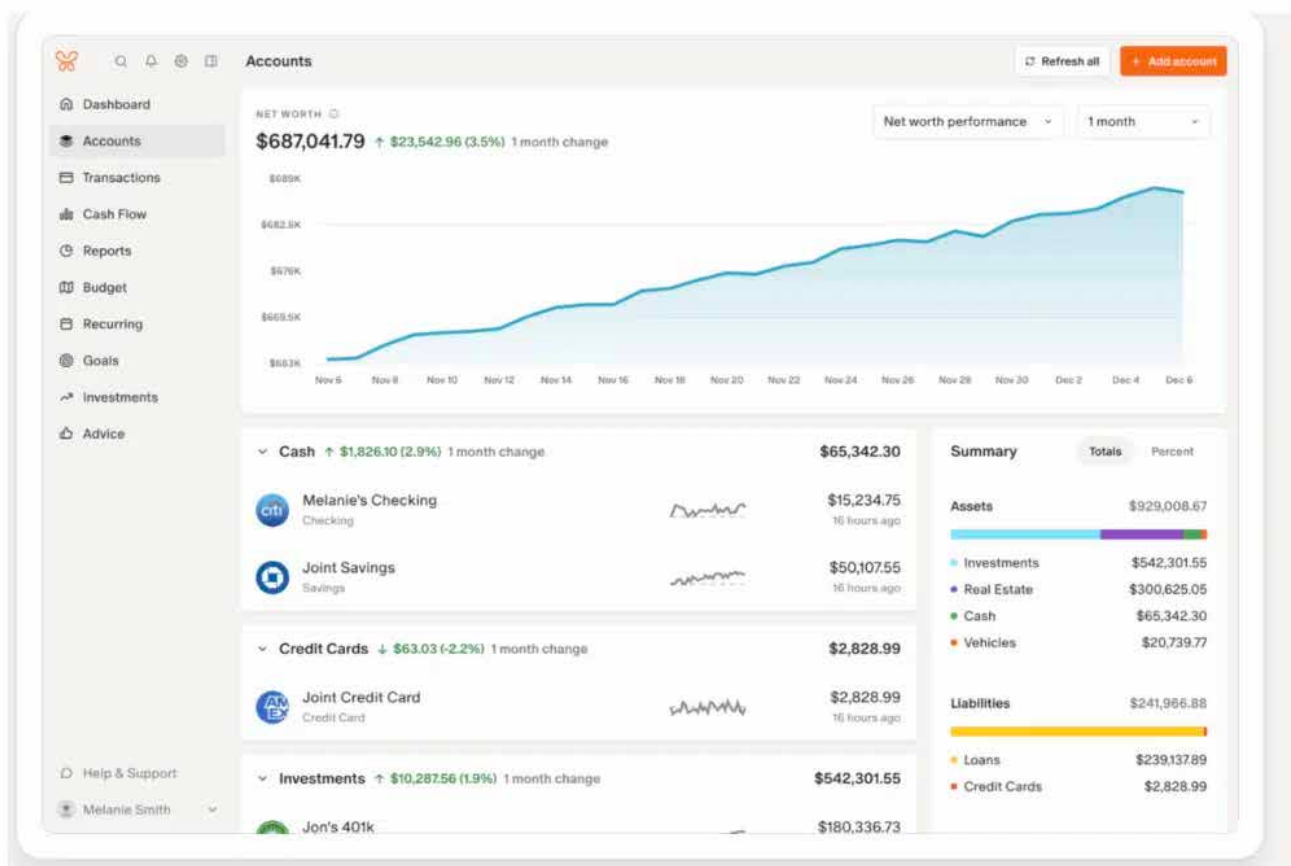


Рис. 1.6. Інтерфейс системи управління особистими фінансами Monarch Money

Для узагальнення результатів аналізу існуючих програмних рішень доцільно виконати їх порівняння за ключовими функціональними характеристиками. Основні можливості розглянутих систем наведено в табл. 1.2.

Таблиця 1.2

Порівняння існуючих систем управління особистими фінансами

Критерій	Mint	YNAB	PocketGuard	Quicken Simplifi	Monarch Money	Розроблювана система
Тип системи	Веб-сервіс	Веб + мобільна	Мобільна	Веб-сервіс	Веб-сервіс	Веб-портал
Облік доходів і витрат	+	+	+	+	+	+
Категоризація операцій	+	+	+	+	+	+
Планування бюджету	+	+	+	+	+	+
Перегляд балансу	+	+	+	+	+	+
Формування звітів	+	+	±	+	+	+
Аналітика фінансових даних	+	+	±	+	+	+
Керування категоріями	±	+	+	+	+	+
Адміністрування системи	–	–	–	–	–	+

Аналіз даних, наведених у табл. 1.2, показує, що більшість сучасних систем управління особистими фінансами реалізують базові функції обліку фінансових операцій, категоризації витрат і формування статистичних звітів. Водночас багато із зазначених сервісів орієнтовані переважно на мобільні застосунки або мають обмежені можливості налаштування системи та управління користувачами. Крім того, значна частина рішень орієнтована на інтеграцію із банківськими сервісами, що може обмежувати їх використання у середовищах, де користувач здійснює фінансовий облік самостійно.

Проведений аналіз показує, що існуючі програмні системи забезпечують базові інструменти управління персональними фінансами, проте не всі вони надають комплексне веб-орієнтоване середовище для централізованого обліку фінансових операцій, планування бюджету, формування аналітичних звітів та

адміністрування системи. Це обумовлює доцільність розроблення веб-порталу ведення особистого фінансового обліку, який поєднує функції реєстрації користувачів, управління фінансовими операціями, аналізу фінансових даних та формування звітної інформації в межах єдиної інформаційної системи.

1.3 Аналіз вимог програмної системи

Аналіз вимог є одним із ключових етапів розробки інформаційних систем, оскільки саме на цьому етапі формується структуроване уявлення про функціональні можливості програмного продукту, його обмеження, технічні параметри та очікування кінцевих користувачів. Чітко сформульовані вимоги дозволяють узгодити бачення системи між розробником і користувачем, визначити архітектурні принципи побудови програмного забезпечення, а також сформувати основу для подальшого тестування, перевірки та оцінювання якості створюваної системи.

Для веб-порталу ведення особистого фінансового обліку аналіз вимог є особливо важливим, оскільки система повинна забезпечувати не лише фіксацію фінансових операцій, але й підтримувати аналітичну обробку фінансових даних, планування бюджету, формування звітності та зручну взаємодію користувача із системою. У процесі аналізу було визначено основні категорії вимог, які традиційно поділяються на функціональні, нефункціональні, вимоги до безпеки та технічні вимоги.

Функціональні вимоги описують основні можливості програмної системи та визначають набір операцій, які вона повинна виконувати для задоволення потреб користувача. Вони відображають зовнішню поведінку системи, сервіси, що надаються користувачам, а також реакцію системи на певні дії або події.

Таблиця 1.3

Функціональні вимоги до веб-порталу ведення особистого фінансового обліку

№	Вимога	Опис функціональності
1	Реєстрація та авторизація	Створення облікового запису користувача та вхід до системи
2	Керування профілем	Можливість редагування персональних даних користувача
3	Додавання фінансових операцій	Введення інформації про доходи та витрати
4	Керування категоріями	Створення та редагування категорій доходів і витрат
5	Перегляд балансу та історії операцій	Відображення фінансових записів та поточного стану бюджету
6	Планування бюджету	Встановлення фінансових лімітів та контроль витрат
7	Формування звітів та аналітики	Побудова статистичних звітів щодо структури доходів і витрат
8	Адміністрування системи	Керування користувачами та системними параметрами

Представлений функціонал забезпечує реалізацію основних процесів предметної області, пов'язаних із обліком персональних фінансів, аналізом фінансової активності користувача та підтримкою прийняття фінансових рішень.

Нефункціональні вимоги визначають якісні характеристики програмної системи, які безпосередньо впливають на ефективність її використання. До них належать показники продуктивності, зручності використання, масштабованості, надійності та сумісності системи.

Таблиця 1.4

Нефункціональні вимоги до програмної системи

№	Категорія	Опис вимог
1	Продуктивність	Час відгуку системи не більше 1–2 секунд при виконанні основних операцій
2	Зручність використання	Наявність інтуїтивно зрозумілого веб-інтерфейсу

Продовження таблиці 1.4

3	Портативність	Доступ до системи через сучасні веб-браузери
4	Масштабованість	Можливість розширення функціоналу без зміни базової архітектури
5	Надійність	Стійкість системи до помилкових дій користувача
6	Сумісність	Коректна робота на різних пристроях та операційних системах

Дотримання зазначених нефункціональних вимог забезпечує стабільність роботи системи, комфортну взаємодію користувача із програмним забезпеченням та можливість подальшого розвитку програмного продукту.

Оскільки система працює з персональними фінансовими даними користувачів, питання захисту інформації є одним із пріоритетних. Наявність механізмів безпеки дозволяє запобігти несанкціонованому доступу до даних та гарантує їх цілісність і конфіденційність.

Таблиця 1.5

Вимоги до безпеки програмної системи

№	Вимога	Опис
1	Аутентифікація користувача	Доступ до системи лише після введення логіна та пароля
2	Захист персональних даних	Збереження конфіденційної інформації у захищеному вигляді
3	Розмежування прав доступу	Розподіл ролей користувача та адміністратора
4	Журналювання операцій	Фіксація дій користувачів у системі
5	Захист від несанкціонованого доступу	Обмеження доступу до системних функцій

Застосування зазначених механізмів дозволяє підвищити рівень захисту інформації та забезпечити безпечне використання системи користувачами.

Технічні вимоги визначають мінімальні параметри програмного та апаратного забезпечення, необхідні для функціонування веб-порталу. Вони

описують технологічне середовище реалізації системи та вимоги до ресурсів, необхідних для її стабільної роботи.

Таблиця 1.6

Технічні вимоги до програмної системи

№	Параметр	Значення
1	Операційна система	Windows, Linux, macOS
2	Браузер	Google Chrome, Mozilla Firefox, Microsoft Edge, Safari
3	Процесор	Мінімум 2 ядра
4	Оперативна пам'ять	Від 4 ГБ
5	Вільне місце на диску	Від 500 МБ
6	Програмне середовище	Веб-сервер, СУБД, сучасні веб-технології

Визначені технічні параметри забезпечують працездатність веб-порталу на більшості сучасних комп'ютерних систем і мобільних пристроїв без необхідності використання високопродуктивного обладнання.

Проведений аналіз вимог дозволив сформулювати структурований перелік характеристик, яким повинна відповідати розроблювана інформаційна система ведення особистого фінансового обліку. Чітке розмежування функціональних, нефункціональних, безпекових та технічних вимог забезпечує цілісне розуміння можливостей майбутньої системи, визначає основні напрями її проектування та створює основу для подальшої розробки архітектури програмного забезпечення. Особлива увага приділена захисту персональних фінансових даних користувачів, що є важливим фактором забезпечення надійності та довіри до програмного продукту. Врахування технічних вимог дозволяє забезпечити стабільну роботу системи у різних програмних середовищах та підвищує доступність веб-порталу для широкого кола користувачів.

1.4 Моделювання предметної області

Моделювання предметної області є важливим етапом проєктування програмної системи, який дозволяє формалізувати структуру, поведінку та взаємодію компонентів майбутнього програмного продукту. Використання методології UML (Unified Modeling Language) забезпечує можливість візуалізації основних процесів функціонування системи, ролей користувачів та взаємодії між ними. Завдяки використанню графічних моделей спрощується процес аналізу архітектури програмної системи, визначення її основних функцій та подальшого проєктування програмної реалізації.

У межах моделювання предметної області веб-порталу ведення особистого фінансового обліку було побудовано декілька типів UML-діаграм, які відображають різні аспекти роботи системи: діаграму варіантів використання, діаграму послідовності та діаграму діяльності.

На рисунку 1.4 представлено діаграму варіантів використання, яка демонструє основні функції системи та взаємодію користувачів із веб-порталом.

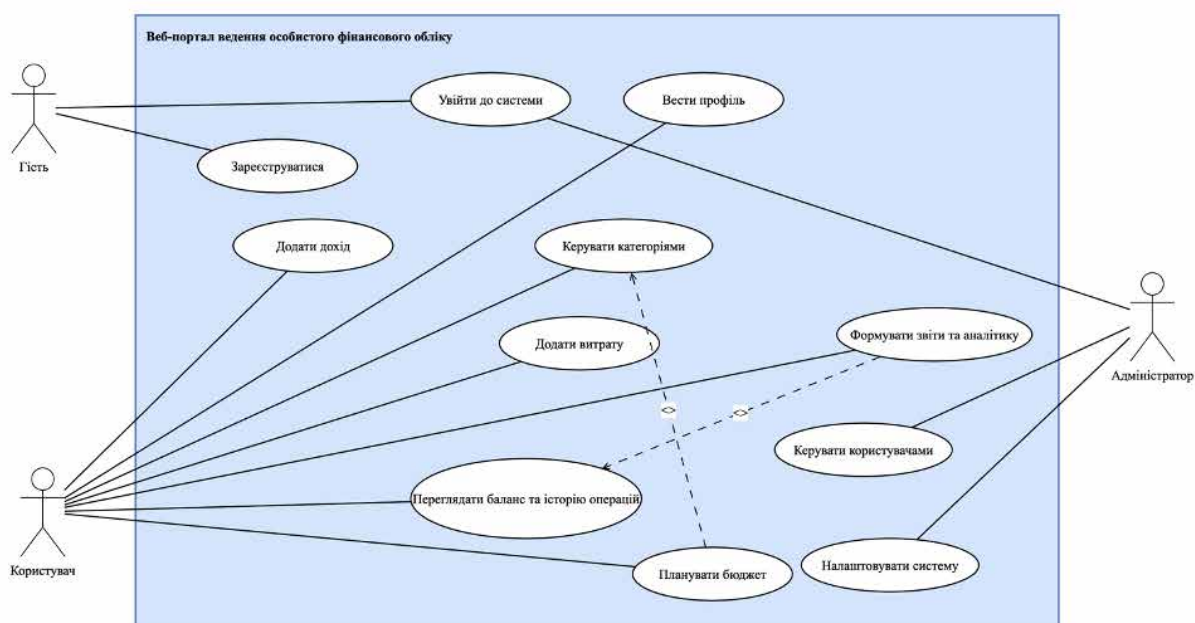


Рис. 1.4. Діаграма варіантів використання веб-порталу ведення особистого фінансового обліку

Як видно з рисунка 1.4, у системі визначено три основні актори: гість, користувач та адміністратор. Гість має можливість здійснювати базові операції

взаємодії із системою, зокрема переглядати веб-портал, реєструвати новий обліковий запис або виконувати вхід до системи. Після авторизації користувач отримує доступ до функціоналу ведення персонального фінансового обліку, включаючи додавання доходів і витрат, керування категоріями фінансових операцій, перегляд балансу та історії транзакцій, а також планування бюджету та формування аналітичних звітів. Адміністратор системи, у свою чергу, має розширені права доступу, що дозволяють здійснювати керування користувачами, формувати системні звіти та виконувати налаштування програмного середовища.

Для детальнішого відображення процесів взаємодії між користувачами, веб-порталом та базою даних було побудовано діаграму послідовності, яка представлена на рисунку 1.5.

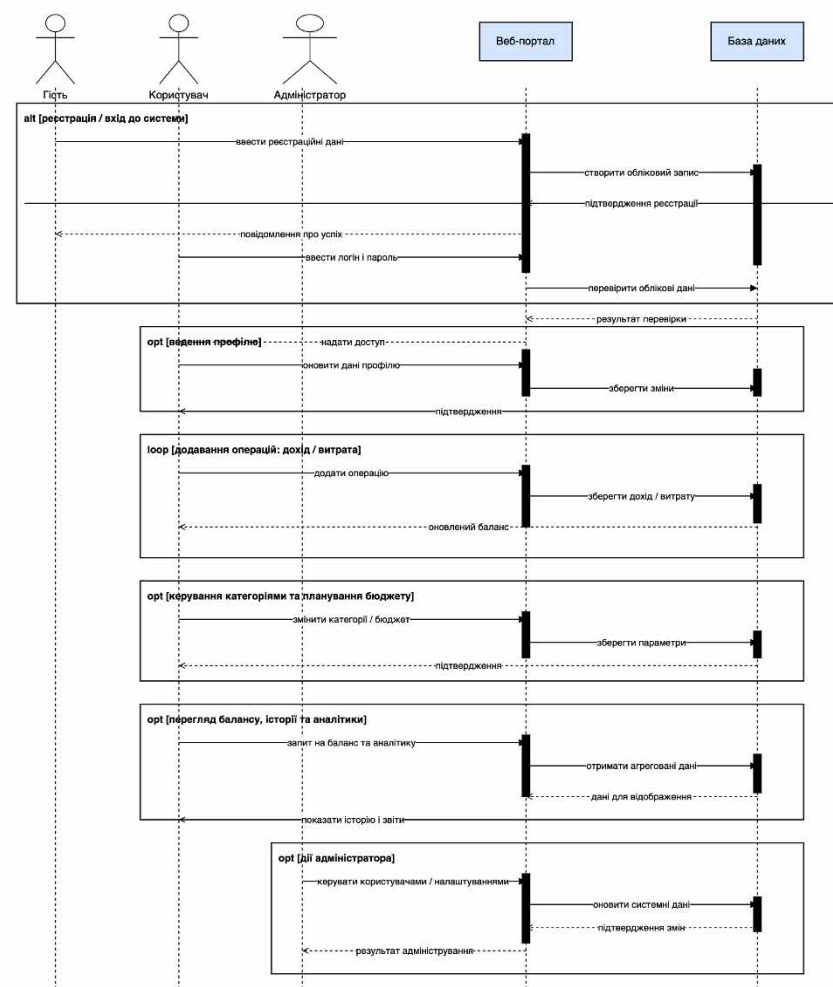


Рис. 1.5. Діаграма послідовності взаємодії користувачів із веб-порталом

Дана діаграма демонструє послідовність виконання основних операцій у системі. Взаємодія починається з процесу реєстрації або авторизації користувача, під час якого веб-портал перевіряє введені облікові дані та створює новий обліковий запис у базі даних або підтверджує доступ до існуючого. Після успішної автентифікації користувач отримує доступ до функцій системи, зокрема ведення профілю, додавання фінансових операцій, керування категоріями витрат та доходів, а також формування фінансових звітів і аналітики. Усі дії користувача обробляються веб-порталом та зберігаються у базі даних, що забезпечує цілісність та актуальність фінансової інформації.

Для відображення логіки виконання основних процесів системи було побудовано діаграму діяльності, представлену на рисунку 1.6.

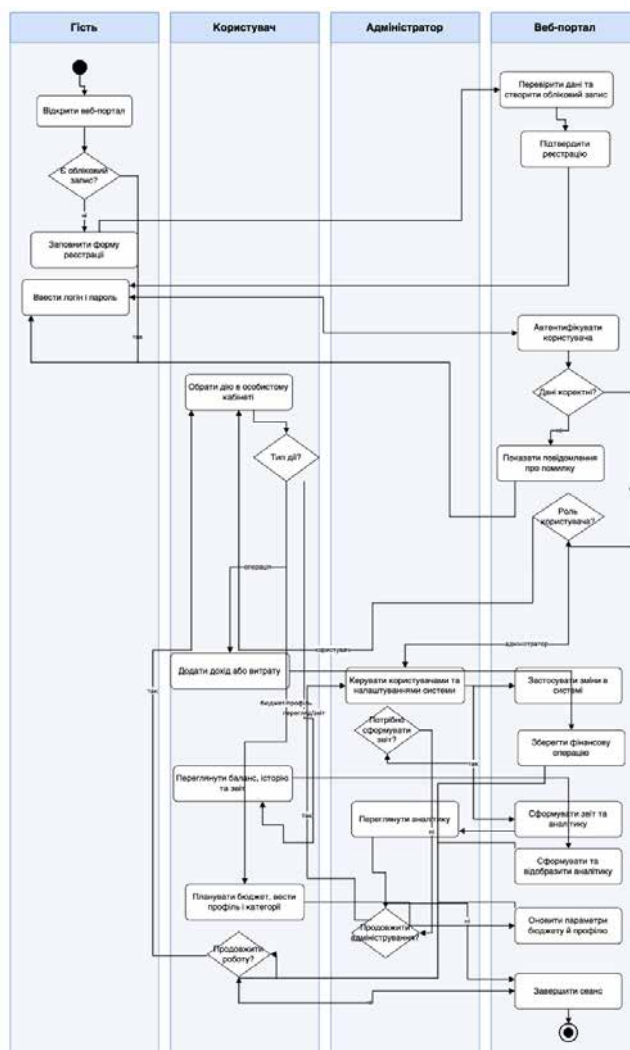


Рис. 1.6. Діаграма діяльності веб-порталу ведення особистого фінансового обліку

Діаграма діяльності відображає послідовність виконання операцій у системі та логіку переходів між різними етапами взаємодії користувача з веб-порталом. Процес починається з відкриття веб-порталу та перевірки наявності облікового запису користувача. У разі його відсутності здійснюється реєстрація нового користувача. Після успішної автентифікації користувач отримує доступ до особистого кабінету, де може виконувати різні фінансові операції, переглядати історію транзакцій, формувати звіти або здійснювати планування бюджету. Адміністратор системи має можливість виконувати операції адміністрування, включаючи керування користувачами та налаштування системних параметрів.

Таким чином, побудовані UML-діаграми дозволили сформулювати цілісне уявлення про структуру та функціонування веб-порталу ведення особистого фінансового обліку. Діаграма варіантів використання відображає основні функції системи та ролі користувачів, діаграма послідовності демонструє порядок взаємодії між компонентами системи, а діаграма діяльності описує логіку виконання основних процесів. Сукупність отриманих моделей створює концептуальну основу для подальшого проектування архітектури програмного забезпечення та реалізації функціональних модулів системи.

1.5 Постановка завдання

Постановка завдання є завершальним етапом системного аналізу, під час якого визначаються основні цілі розробки програмної системи, способи їх реалізації та очікувані результати функціонування програмного продукту. На основі проведеного аналізу предметної області, огляду існуючих інформаційних джерел та програмних рішень, а також сформованих функціональних і нефункціональних вимог було визначено необхідність створення веб-орієнтованої інформаційної системи, призначеної для ведення особистого фінансового обліку.

Основною метою розроблюваної системи є створення веб-порталу, який надаватиме користувачам можливість зручно та ефективно фіксувати фінансові операції, контролювати структуру доходів і витрат, планувати бюджет та отримувати аналітичну інформацію щодо власного фінансового стану. Використання веб-орієнтованої архітектури дозволяє забезпечити доступ до системи з різних пристроїв, спростити процес оновлення програмного забезпечення та створити умови для подальшого масштабування функціоналу.

Розроблювана система має враховувати сучасні тенденції у сфері веб-технологій та інформаційних систем управління персональними фінансами. Вона має забезпечувати зручну взаємодію користувача з фінансовими даними, їх структуроване збереження, а також формування звітів і аналітичних показників на основі накопиченої інформації. Особливу увагу при розробці необхідно приділити безпеці персональних фінансових даних, надійності збереження інформації та стабільності функціонування програмного забезпечення.

У межах реалізації поставленої задачі програмна система має забезпечувати виконання таких основних функцій:

1. Реєстрацію користувачів у системі та виконання процедури авторизації для доступу до персонального фінансового кабінету.
2. Можливість керування профілем користувача, включаючи редагування персональних даних та налаштувань облікового запису.
3. Додавання, редагування та збереження фінансових операцій, зокрема записів про доходи та витрати.
4. Керування категоріями фінансових операцій для структурування та систематизації фінансових даних.
5. Перегляд поточного балансу користувача та історії фінансових операцій.
6. Планування особистого бюджету з можливістю встановлення лімітів витрат для окремих категорій.
7. Формування статистичних звітів та аналітичної інформації щодо структури доходів і витрат.

8. Надання адміністративного функціоналу для керування користувачами та системними параметрами.

Крім основного функціоналу, система має забезпечувати додаткові можливості, що підвищують ефективність її використання:

1. Відображення фінансової статистики у вигляді графіків і діаграм.
2. Формування звітів за визначений період часу.
3. Автоматичне оновлення балансу користувача після внесення нових фінансових операцій.
4. Збереження історії фінансових змін для подальшого аналізу.
5. Реалізацію механізмів повідомлень або підказок для підвищення зручності користування системою.

У процесі реалізації веб-порталу передбачається використання сучасних веб-технологій, що дозволять створити масштабовану та зручну інформаційну систему. Дані користувачів будуть зберігатися у базі даних, що забезпечить їх структурованість, цілісність та можливість подальшої аналітичної обробки.

Розробка системи охоплюватиме основні етапи життєвого циклу програмного забезпечення, включаючи проєктування архітектури системи, моделювання предметної області, реалізацію програмних компонентів, тестування та підготовку до експлуатації. Особлива увага приділятиметься забезпеченню зручності користування системою, стабільності її роботи та можливості подальшого розвитку функціоналу.

Реалізація поставленого завдання дозволить створити сучасний веб-портал для ведення особистого фінансового обліку, який забезпечуватиме ефективне управління фінансовими даними користувачів, підвищуватиме рівень фінансової дисципліни та сприятиме прийняттю обґрунтованих фінансових рішень.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-ПОРТАЛУ

2.1 Логічна модель даних у вигляді ER-діаграми

Логічна модель даних веб-порталу ведення особистого фінансового обліку призначена для формалізованого опису основних інформаційних об'єктів системи та зв'язків між ними. На цьому етапі визначаються сутності, які відображають зміст предметної області, а також правила їх взаємодії в межах процесів обліку доходів, витрат, бюджетного планування та формування фінансової аналітики.

Основу логічної моделі становлять сутності User, Category, Financial_Transaction, Budget, Analytics_Report, Income та Expense. Центральним об'єктом предметної області є фінансова операція, оскільки саме вона фіксує факт надходження або витрачання коштів. Усі інші об'єкти забезпечують структурування, групування, контроль і подальший аналіз таких операцій.

Сутність User описує користувача веб-порталу. Вона зберігає облікову інформацію, яка необхідна для авторизації, персоналізації фінансового простору та прив'язки всіх фінансових записів до конкретної особи. Один користувач може створювати багато категорій, бюджетів, фінансових операцій і аналітичних звітів. Такий зв'язок дозволяє забезпечити ізольоване зберігання персональних фінансових даних кожного користувача.

Сутність Category використовується для класифікації доходів і витрат. Категорія належить конкретному користувачу та визначає змістове групування фінансових операцій. Наприклад, для витрат можуть використовуватися категорії «Харчування», «Транспорт», «Комунальні платежі», а для доходів - «Заробітна плата», «Стипендія», «Підробіток». Завдяки категоріям система може формувати структуровану аналітику та показувати, які напрями доходів або витрат є найбільш значущими.

Сутність Financial_Transaction відображає окрему фінансову операцію. Вона містить суму, дату, опис, тип операції та зв'язки з користувачем, категорією і, за потреби, бюджетом. Кожна операція має належати одному користувачу та одній категорії. Такий підхід забезпечує логічну цілісність даних і дозволяє виконувати пошук, фільтрацію та групування операцій за періодами, категоріями й типами.

Сутність Budget призначена для планування фінансових лімітів. Бюджет створюється користувачем для певної категорії або напряму витрат і має визначений період дії. У межах бюджету зберігається запланована сума та порогове значення попередження. Це дозволяє системі контролювати фактичні витрати, порівнювати їх із плановими значеннями та повідомляти користувача про наближення до встановленого ліміту.

Сутність Analytics_Report використовується для збереження результатів аналітичної обробки фінансових даних. Звіт формується за визначений період і містить підсумкові показники доходів, витрат і балансу. Завдяки цьому користувач може переглядати не лише окремі операції, а й узагальнені результати фінансової активності.

Сутності Income та Expense деталізують фінансову операцію залежно від її типу. Якщо операція є доходом, для неї може зберігатися джерело надходження коштів. Якщо операція є витратою, додатково фіксується спосіб оплати. Такий підхід дозволяє не перевантажувати основну таблицю фінансових операцій другорядними полями та водночас зберігати специфічні дані для різних типів операцій.

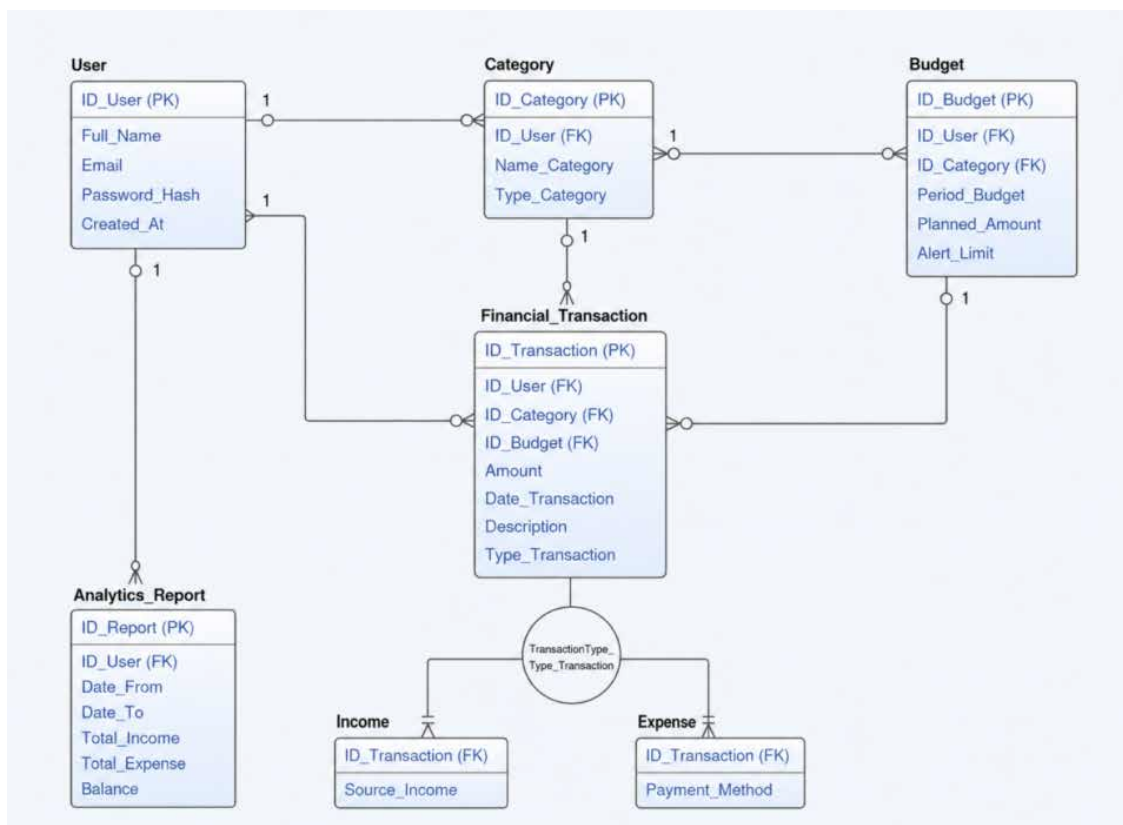


Рис. 2.1. Логічна ER-модель веб-порталу ведення особистого фінансового обліку

Характеристику основних сутностей логічної моделі наведено в таблиці 2.1.

Таблиця 2.1

Основні сутності логічної моделі даних

Сутність	Призначення	Основні дані
User	Зберігає дані користувача та забезпечує персоналізацію фінансового обліку	ПІБ, email, хеш пароля, дата створення
Category	Використовується для групування доходів і витрат	Назва категорії, тип категорії, ідентифікатор користувача
Financial_Transaction	Фіксує факт доходу або витрати	Сума, дата, опис, тип операції, користувач, категорія
Budget	Зберігає планові фінансові обмеження	Період бюджету, планова сума, ліміт попередження
Analytics_Report	Містить результати аналітичної обробки фінансових даних	Період, загальні доходи, витрати, баланс
Income	Деталізує операції доходу	Ідентифікатор операції, джерело доходу

Продовження таблиці 2.1

Expense	Деталізує операції витрати	Ідентифікатор операції, спосіб оплати
---------	----------------------------	---------------------------------------

Зв'язки між сутностями відображають логіку функціонування веб-порталу. Один користувач може мати багато категорій, бюджетів, фінансових операцій і звітів. Одна категорія може використовуватися в багатьох фінансових операціях і бюджетах. Один бюджет може бути пов'язаний з багатьма операціями, якщо вони належать до відповідної категорії та періоду. Фінансова операція може мати деталізацію або як дохід, або як витрата.

Таблиця 2.2

Основні зв'язки логічної моделі

Зв'язок	Тип зв'язку	Зміст
User – Category	1:N	Один користувач створює багато категорій
User – Financial Transaction	1:N	Один користувач має багато фінансових операцій
User – Budget	1:N	Один користувач може створити кілька бюджетів
User – Analytics_Report	1:N	Один користувач може формувати багато звітів
Category – Financial Transaction	1:N	Одна категорія використовується в багатьох операціях
Category – Budget	1:N	Для однієї категорії можуть створюватися бюджети за різні періоди
Budget – Financial Transaction	1:N	Один бюджет може охоплювати декілька фінансових операцій
Financial Transaction – Income	1:0..1	Дохід деталізує фінансову операцію відповідного типу
Financial Transaction – Expense	1:0..1	Витрата деталізує фінансову операцію відповідного типу

Запропонована логічна модель відповідає вимогам предметної області та забезпечує чітке розмежування облікових, довідкових і аналітичних даних. Вона створює основу для побудови фізичної структури бази даних, реалізації програмних модулів і формування звітності.

2.2 Фізична модель даних і структура бази даних

Фізична модель даних визначає практичну реалізацію логічної структури у вигляді таблиць реляційної бази даних. Для веб-порталу ведення особистого фінансового обліку база даних повинна забезпечувати збереження персональних облікових записів, категорій, бюджетів, фінансових операцій, аналітичних звітів і деталізованої інформації про доходи та витрати.

Фізична структура бази даних побудована з урахуванням принципів нормалізації. Дані про користувачів, категорії, операції, бюджети та звіти зберігаються в окремих таблицях. Це зменшує дублювання інформації, спрощує оновлення записів і забезпечує цілісність зв'язків між об'єктами. Для ідентифікації записів використовуються первинні ключі, а для зв'язування таблиць - зовнішні ключі.

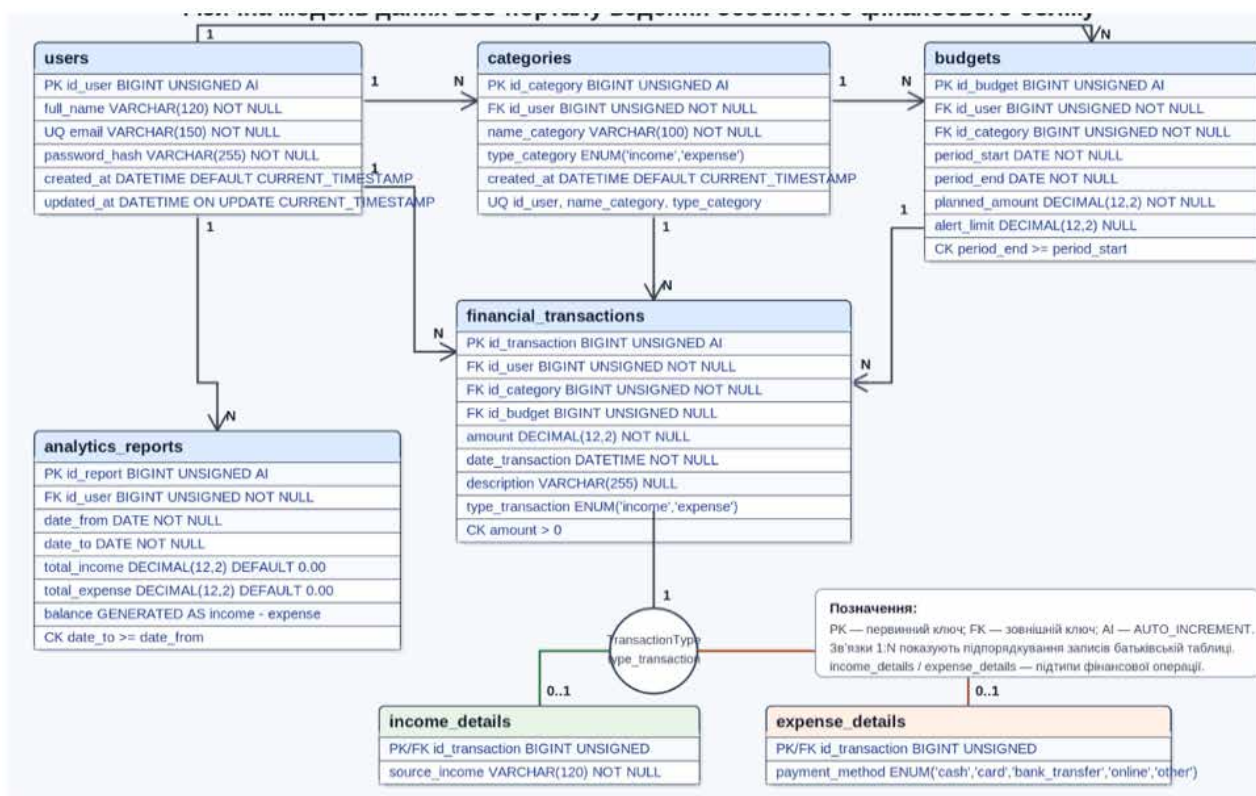


Рис. 2.2. Фізична модель бази даних веб-порталу ведення особистого фінансового обліку

Основні таблиці фізичної моделі наведено в таблиці 2.3.

Таблиця 2.3

Структура основних таблиць бази даних

Таблиця	Призначення	Ключові поля
users	Зберігає дані користувачів веб-порталу	id_user, email, password_hash
categories	Містить категорії доходів і витрат	id_category, id_user, name_category, type_category
financial_transactions	Зберігає фінансові операції користувача	id_transaction, id_user, id_category, amount, date_transaction, type_transaction
budgets	Зберігає планові бюджетні обмеження	id_budget, id_user, id_category, period_start, period_end, planned_amount
analytics_reports	Містить сформовані аналітичні звіти	id_report, id_user, date_from, date_to, total_income, total_expense, balance
income_details	Зберігає деталізацію доходів	id_transaction, source_income
expense_details	Зберігає деталізацію витрат	id_transaction, payment_method

Таблиця users є базовою для роботи системи, оскільки всі фінансові дані належать конкретному користувачу. У ній зберігаються ПІБ, електронна пошта, хеш пароля, дата створення та оновлення облікового запису. Використання поля password_hash дозволяє не зберігати пароль у відкритому вигляді.

Таблиця categories призначена для класифікації операцій. Кожна категорія належить певному користувачу та має тип income або expense. Це дає змогу відокремити категорії доходів від категорій витрат і використовувати їх у фільтрах, бюджетах та аналітичних звітах.

Таблиця financial_transactions зберігає основні записи про доходи й витрати. Для кожної операції зазначається користувач, категорія, бюджет, сума, дата, опис і тип операції. Саме ця таблиця є основним джерелом даних для розрахунку балансу, формування історії операцій і побудови фінансової аналітики.

Таблиця budgets використовується для планування витрат за категоріями та періодами. У ній зберігаються дата початку й завершення бюджету, планова

сума та значення ліміту попередження. Це дозволяє реалізувати контроль виконання бюджету й автоматично визначати ситуації, коли фактичні витрати наближаються до встановленої межі або перевищують її.

Таблиця `analytics_reports` містить підсумкові дані сформованих звітів. Вона може використовуватися для збереження результатів аналізу за період, щоб користувач мав доступ до попередньо сформованих фінансових підсумків без повторного виконання всіх розрахунків.

Окремі таблиці `income_details` та `expense_details` використовуються для деталізації фінансових операцій. Для доходів зберігається джерело надходження коштів, а для витрат - спосіб оплати. Така структура робить модель гнучкішою та дозволяє розширювати її без порушення основної таблиці операцій.

Таблиця 2.4

Основні обмеження цілісності фізичної моделі

Обмеження	Реалізація	Призначення
Первинні ключі	<code>id_user</code> , <code>id_category</code> , <code>id_transaction</code> , <code>id_budget</code> , <code>id_report</code>	Однозначна ідентифікація записів
Зовнішні ключі	<code>id_user</code> , <code>id_category</code> , <code>id_budget</code> , <code>id_transaction</code>	Підтримка зв'язків між таблицями
Унікальність <code>email</code>	UNIQUE для <code>email</code> у таблиці <code>users</code>	Запобігання дублюванню облікових записів
Перевірка типів	ENUM для <code>income</code> та <code>expense</code>	Обмеження допустимих типів операцій і категорій
Перевірка сум	CHECK для додатних значень <code>amount</code> і <code>planned_amount</code>	Захист від некоректних фінансових записів
Перевірка періодів	CHECK для <code>period_end</code> $\geq$ <code>period_start</code>	Контроль правильності бюджетного періоду

Для підвищення швидкодії системи доцільно створити індекси за полями, які часто використовуються у вибірках: `id_user`, `id_category`, `date_transaction`, `type_transaction`, `period_start`, `period_end`. Це прискорює побудову історії операцій, фільтрацію за періодами, формування звітів і перевірку виконання бюджету.

Отже, фізична модель даних забезпечує надійну основу для реалізації веб-порталу. Вона підтримує персоналізований облік фінансових операцій,

категоризацію доходів і витрат, бюджетне планування та формування аналітичних звітів.

2.3 Діаграма компонентів і пакетна структура

Для опису архітектурної організації програмного забезпечення веб-порталу ведення особистого фінансового обліку побудовано діаграму компонентів і діаграму пакетів. Ці моделі дозволяють визначити основні програмні частини системи, їхню відповідальність і напрями взаємодії між ними.

Діаграма компонентів відображає розподіл системи на клієнтський пристрій, веб-сервер, сервер застосунку, базу даних і зовнішній поштовий сервіс. Користувач працює з веб-порталом через браузер. Запити передаються на сервер через HTTPS. На сервері розміщується інтерфейс веб-порталу, REST API, модуль бізнес-логіки та модуль автентифікації. Дані зберігаються у MySQL Server у базі PersonalFinanceDB. Для надсилання повідомлень або підтверджень може використовуватися зовнішній поштовий сервіс.

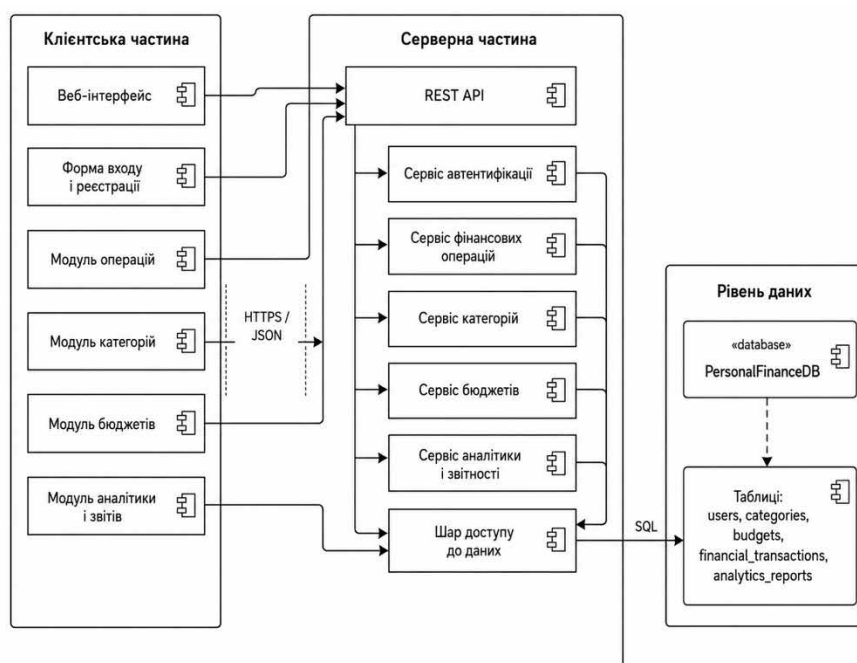


Рис. 2.3. Діаграма компонентів веб-порталу ведення особистого фінансового обліку

Основні компоненти програмної системи наведено в таблиці 2.5.

Таблиця 2.5

Призначення компонентів програмної системи

Компонент	Призначення
Веб-браузер	Забезпечує доступ користувача до інтерфейсу порталу
Інтерфейс веб-порталу	Відображає сторінки, форми, таблиці, графіки та повідомлення
REST API	Приймає запити клієнтської частини та передає їх до серверної логіки
Модуль автентифікації	Перевіряє облікові дані користувача та керує сеансом
Модуль бізнес-логіки	Обробляє операції, категорії, бюджети, аналітику та звіти
MySQL Server	Забезпечує роботу реляційної бази даних
PersonalFinanceDB	Зберігає користувачів, операції, категорії, бюджети та звіти
Email Service	Використовується для надсилання службових повідомлень

Пакетна структура деталізує внутрішню організацію програмного коду. У системі доцільно виділити презентаційний рівень, пакет безпеки та автентифікації, пакет прикладної логіки, пакет доступу до даних і пакет бази даних. Такий поділ спрощує розробку, супровід і подальше розширення системи.



Рис. 2.4. Пакетна структура веб-порталу ведення особистого фінансового обліку

Презентаційний рівень відповідає за взаємодію користувача з веб-порталом. Він містить сторінки входу, реєстрації, перегляду балансу, введення операцій, керування категоріями, планування бюджету та перегляду звітів.

Пакет безпеки та автентифікації забезпечує перевірку користувача, захист доступу до персональних фінансових даних і розмежування можливостей користувача та адміністратора. Це важливо, оскільки система працює з конфіденційними фінансовими записами.

Пакет прикладної логіки містить основні функціональні модулі: користувачі, категорії, фінансові операції, бюджети, аналітика і звіти. Саме ці модулі реалізують головні бізнес-процеси системи. Пакет доступу до даних відповідає за виконання запитів до бази, а пакет бази даних зберігає постійну інформацію.

Отже, компонентна та пакетна структура забезпечує логічний поділ системи на самостійні частини. Це підвищує керованість проєкту, спрощує тестування окремих модулів і створює основу для подальшого розвитку веб-порталу.

2.4 Алгоритмічне забезпечення фінансової аналітики

Алгоритмічне забезпечення веб-порталу призначене для автоматизованого оброблення фінансових операцій, контролю виконання бюджету та формування аналітичних звітів. Основна задача алгоритмів полягає в тому, щоб перетворити набір окремих фінансових записів на зрозумілу користувачу інформацію про доходи, витрати, баланс і фінансову дисципліну.

У системі передбачено алгоритм формування звіту доходів і витрат. Його виконання починається з вибору користувачем звітного періоду. Після цього система зчитує фінансові операції з бази даних, відбирає записи за датою, розділяє їх на доходи та витрати, обчислює підсумкові значення, групує дані за категоріями та формує звіт.

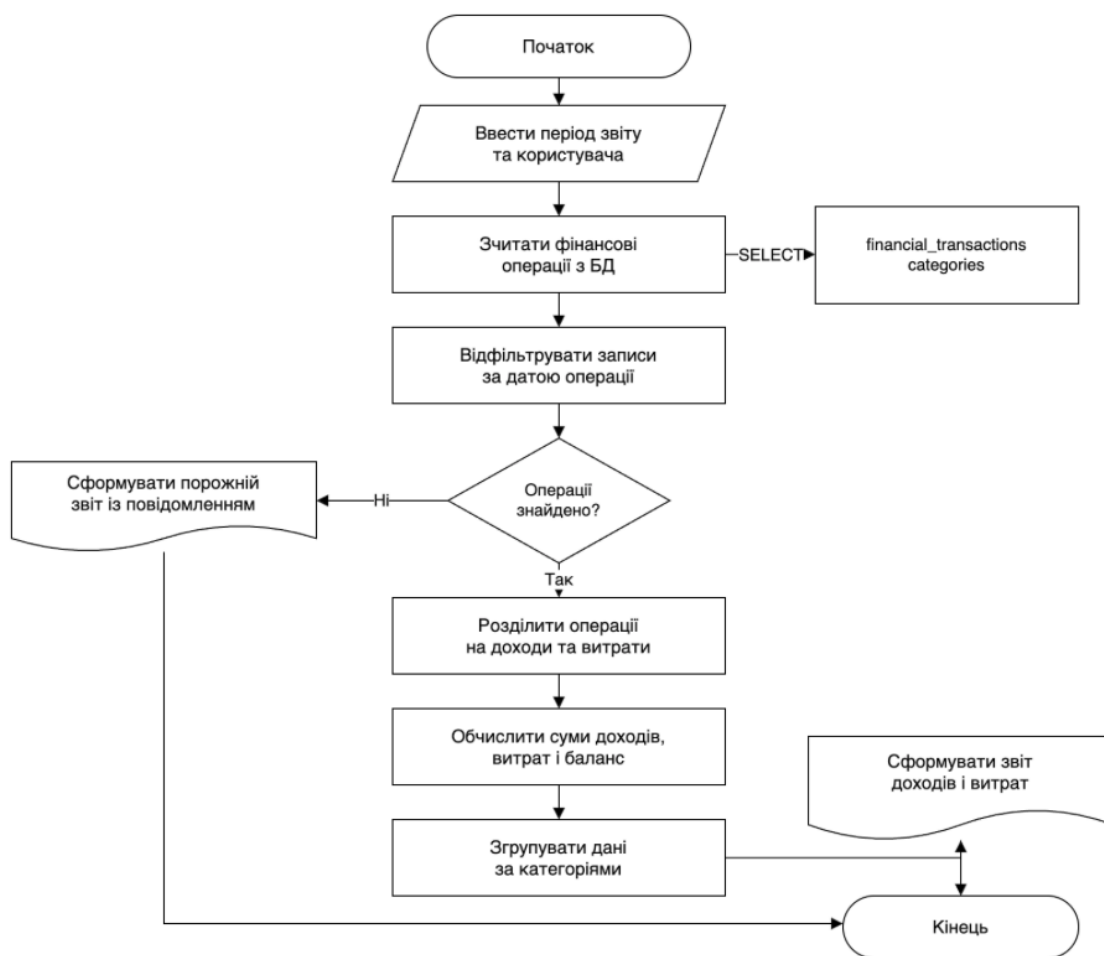


Рис. 2.5. Алгоритм формування звіту доходів і витрат

Якщо за вибраний період фінансові операції відсутні, система формує порожній звіт із відповідним повідомленням. Якщо операції знайдено, користувач отримує таблицю або графічне представлення з підсумками за доходами, витратами і категоріями. Такий алгоритм дозволяє швидко оцінити фінансову активність за тиждень, місяць або довільний період.

Другим важливим алгоритмом є перевірка виконання бюджету. Користувач обирає період і категорію бюджету, після чого система зчитує заплановані значення та фактичні витрати. Далі виконується порівняння фактичних витрат із плановим лімітом. Якщо витрати перевищують план, система позначає бюджет як перевищений. Якщо витрати наближаються до встановленого порогу, система фіксує статус наближення до ліміту. Якщо показники не перевищують установлених меж, бюджет має статус виконання в межах норми.

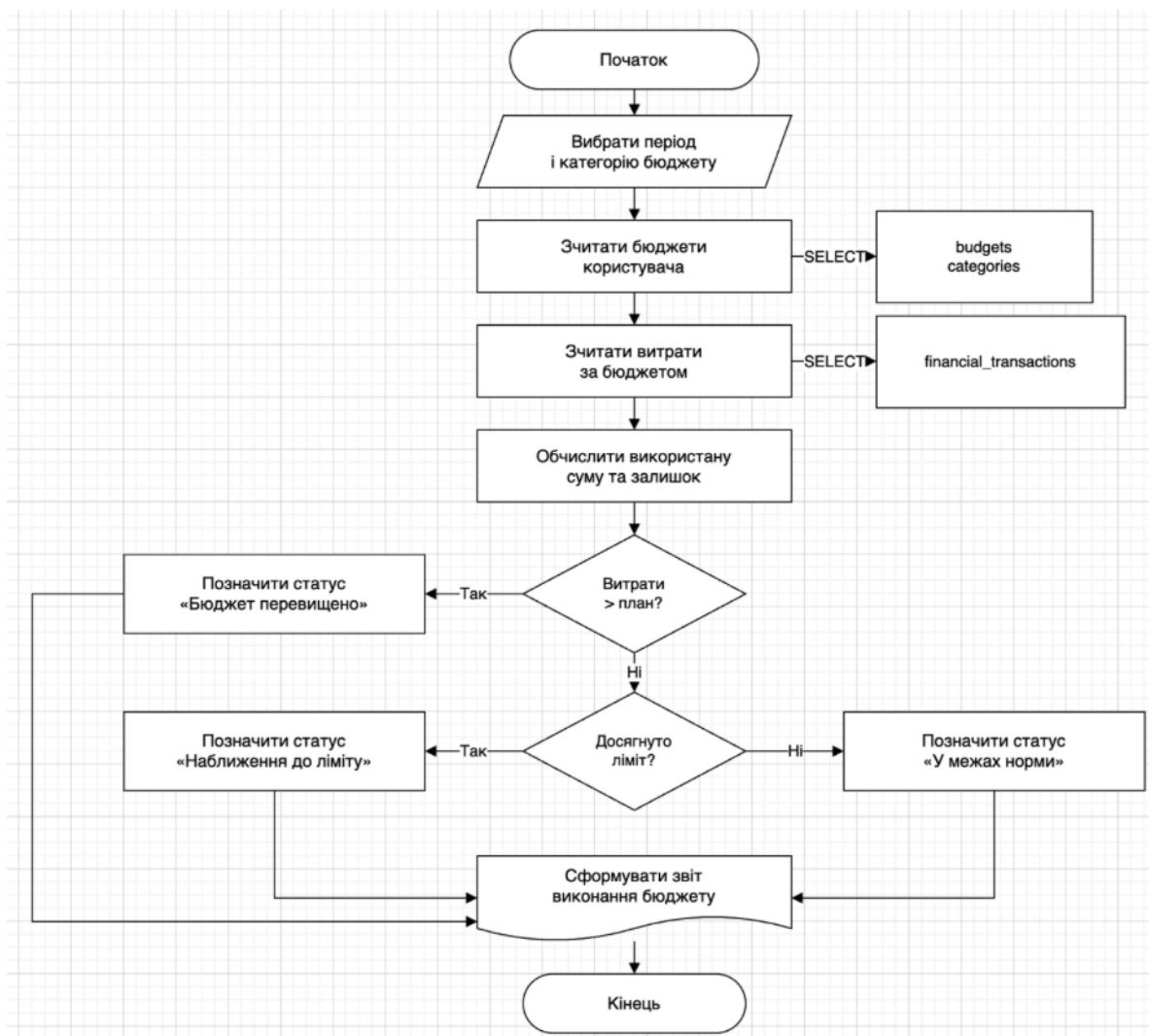


Рис. 2.6. Алгоритм контролю виконання бюджету

Окремо передбачено алгоритм оновлення балансу користувача. Після внесення нової операції система визначає її тип. Якщо це дохід, сума додається до загального балансу. Якщо це витрата, сума зменшує баланс. Після цього оновлене значення відображається в особистому кабінеті користувача та використовується під час формування звітів.

Таблиця 2.6

Основні алгоритми веб-порталу

Алгоритм	Вхідні дані	Основні операції	Результат
Формування звіту доходів і витрат	Користувач, період	Вибірка операцій, групування за типом і категоріями	Звіт про доходи, витрати і баланс
Контроль виконання бюджету	Користувач, категорія, період	Порівняння фактичних витрат із планом	Статус бюджету та попередження

Продовження таблиці 2.6

Оновлення балансу	Нова фінансова операція	Визначення типу операції та зміна балансу	Актуальний баланс користувача
Перегляд історії операцій	Користувач, фільтри	Вибірка записів за датою, типом і категорією	Таблиця фінансових операцій
Формування аналітики за категоріями	Операції за період	Групування витрат і доходів за категоріями	Структура фінансової активності

Алгоритми веб-порталу орієнтовані на щоденне використання. Вони не повинні ускладнювати роботу користувача, тому більшість розрахунків виконується автоматично після введення операцій або вибору параметрів звіту. Це дає змогу зменшити кількість ручних підрахунків і підвищити точність фінансового аналізу.

Отже, алгоритмічне забезпечення є важливою частиною веб-порталу, оскільки саме воно забезпечує перехід від простого введення операцій до повноцінного аналізу фінансової інформації. Запропоновані алгоритми підтримують формування звітів, контроль бюджетів, оновлення балансу та побудову аналітичних показників.

2.5 Висновки до другого розділу

У другому розділі виконано проектування інформаційного та програмного забезпечення веб-порталу ведення особистого фінансового обліку. Побудовано логічну модель даних, яка відображає основні сутності предметної області: користувача, категорію, фінансову операцію, бюджет, аналітичний звіт, дохід і витрату. Визначено зв'язки між цими сутностями та показано їх роль у процесах обліку, планування й аналізу фінансів.

Розроблено фізичну модель бази даних, яка забезпечує практичну реалізацію логічної структури. Визначено основні таблиці, ключі, обмеження цілісності та призначення кожної частини бази. Така структура дозволяє

зберігати персональні фінансові записи користувачів, уникати дублювання даних і формувати звіти за періодами, категоріями та типами операцій.

Також у розділі представлено компонентну і пакетну структуру програмної системи. Веб-портал поділено на клієнтську частину, серверну частину, модулі прикладної логіки, пакет доступу до даних і базу даних. Такий поділ забезпечує модульність програмного забезпечення, спрощує супровід і створює умови для подальшого розвитку системи.

Окремо розроблено алгоритмічне забезпечення фінансової аналітики. Описано алгоритми формування звіту доходів і витрат, контролю виконання бюджету, оновлення балансу та групування фінансових даних за категоріями. Отримані рішення створюють основу для подальшої реалізації веб-порталу та забезпечують його відповідність вимогам предметної області.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-ПОРТАЛУ

3.1 Обґрунтування вибору технологій

Для реалізації веб-порталу ведення особистого фінансового обліку обрано веборієнтований підхід. Таке рішення є доцільним, оскільки користувач повинен мати доступ до власних фінансових даних із різних пристроїв через браузер. Вебзастосунок не потребує встановлення окремої програми на комп'ютер користувача, спрощує оновлення програмного забезпечення та дозволяє централізовано зберігати інформацію.

Клієнтську частину доцільно реалізувати з використанням HTML, CSS і JavaScript. HTML визначає структуру сторінок веб-порталу, CSS відповідає за зовнішній вигляд інтерфейсу, а JavaScript забезпечує динамічну взаємодію користувача з формами, таблицями, фільтрами та повідомленнями. Такий набір технологій є достатнім для створення сторінок входу, реєстрації, особистого кабінету, історії операцій, бюджетів і звітів.

Для серверної частини обрано Node.js та Express.js. Node.js дозволяє реалізувати серверну логіку на JavaScript, а Express.js спрощує побудову маршрутів, обробку HTTP-запитів, створення API та взаємодію з базою даних. Такий підхід забезпечує єдине технологічне середовище для клієнтської і серверної частин системи.

Для зберігання даних обрано реляційну СУБД MySQL. Вона добре підходить для задачі персонального фінансового обліку, оскільки дані мають чітку табличну структуру та сталі зв'язки. Користувач пов'язаний із категоріями, операціями, бюджетами і звітами, тому реляційна модель дозволяє надійно підтримувати цілісність таких зв'язків.

Таблиця 3.1

Обґрунтування вибору технологій

Технологія	Призначення	Обґрунтування
HTML	Структура сторінок	Дозволяє створити форми, таблиці, меню та сторінки кабінету
CSS	Оформлення інтерфейсу	Забезпечує єдиний стиль і зручність сприйняття даних
JavaScript	Клієнтська логіка	Використовується для обробки дій користувача та оновлення сторінок
Node.js	Серверне середовище	Дозволяє реалізувати серверну частину веб-порталу
Express.js	Маршрутизація запитів	Спрощує створення API та обробку HTTP-запитів
MySQL	Зберігання даних	Підходить для реляційної структури фінансових записів
SQL	Робота з даними	Забезпечує вибірку, фільтрацію, групування та оновлення записів
CSV / PDF	Експорт звітів	Дозволяє зберігати результати аналізу у зручному форматі

Для захисту користувацьких даних у системі передбачено механізм автентифікації. Паролі мають зберігатися у вигляді хешів, а доступ до фінансових операцій повинен надаватися лише авторизованому користувачу. Адміністратор має окремі можливості для керування користувачами та системними параметрами, однак персональні фінансові дані повинні бути захищені від стороннього доступу.

Отже, обраний технологічний стек забезпечує можливість створення веб-порталу з повним набором функцій: реєстрацією, авторизацією, веденням фінансових операцій, керуванням категоріями, бюджетним плануванням, формуванням звітів та аналітикою.

3.2 Представлення архітектури системи

Архітектура веб-порталу ведення особистого фінансового обліку побудована за клієнт-серверним принципом. Система складається з клієнтського рівня, серверного рівня, рівня бази даних і зовнішнього поштового сервісу.

Такий поділ дозволяє відокремити інтерфейс користувача від прикладної логіки та зберігання даних.

Клієнтський рівень представлений пристроєм користувача з веб-браузером. Через браузер користувач відкриває веб-портал, вводить фінансові операції, переглядає баланс, налаштовує категорії, створює бюджети та формує звіти. Передавання даних між клієнтом і сервером виконується через HTTPS.

Серверний рівень містить веб-сервер і сервер застосунку. На цьому рівні розміщуються інтерфейс веб-порталу, REST API, модуль бізнес-логіки та модуль автентифікації. REST API приймає запити від клієнта та передає їх відповідним модулям. Модуль бізнес-логіки обробляє фінансові операції, бюджети, категорії та звіти. Модуль автентифікації перевіряє логін і пароль користувача, підтримує сеанс роботи та захищає персональні дані.

Рівень бази даних представлений MySQL Server і базою PersonalFinanceDB. У ній зберігаються користувачі, категорії, бюджети, фінансові операції та аналітичні звіти. З'єднання між сервером застосунку і базою даних виконується через TCP/IP із використанням SQL-запитів.

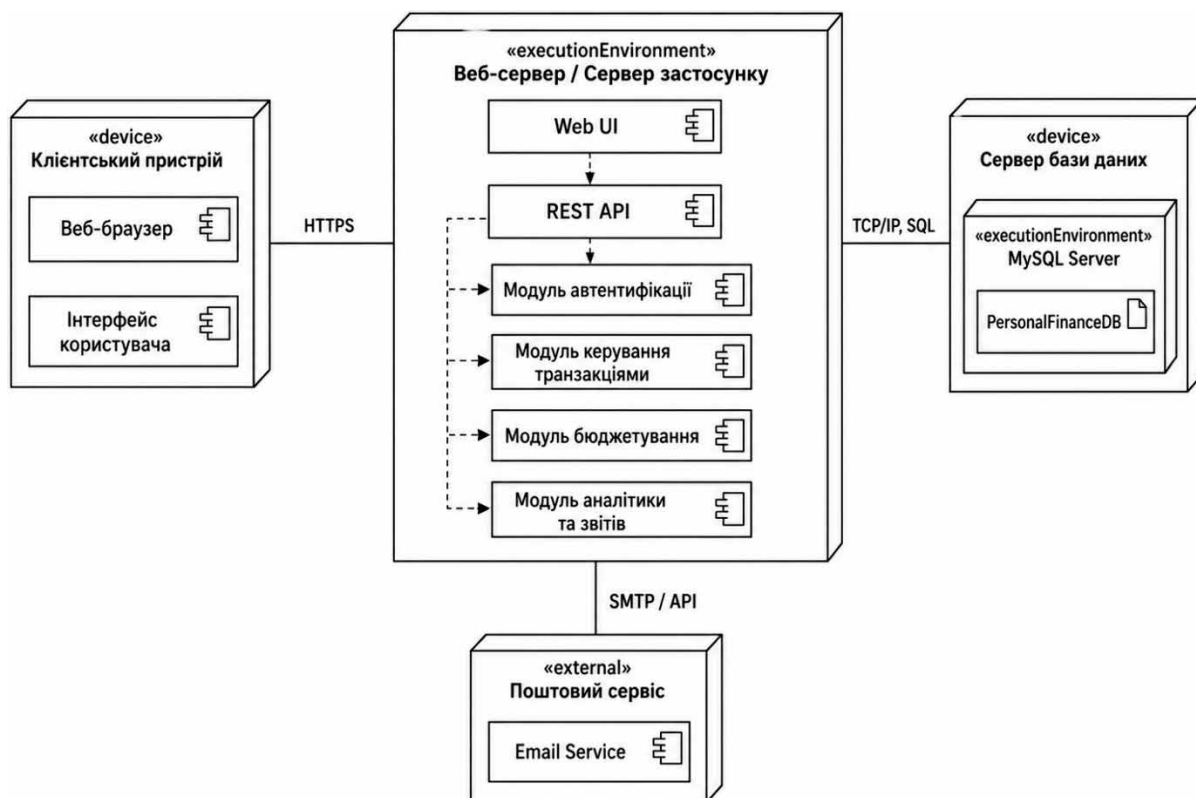


Рис. 3.1. Архітектура веб-порталу ведення особистого фінансового обліку

Основні рівні архітектури наведено в таблиці 3.2.

Таблиця 3.2

Рівні архітектури веб-порталу

Рівень	Компоненти	Призначення
Клієнтський рівень	Клієнтський пристрій, веб-браузер	Забезпечує взаємодію користувача із системою
Серверний рівень	Веб-сервер, REST API, бізнес-логіка, автентифікація	Обробляє запити, перевіряє доступ, виконує операції
Рівень даних	MySQL Server, PersonalFinanceDB	Зберігає користувачів, операції, категорії, бюджети і звіти
Зовнішній сервіс	Email Service	Забезпечує надсилання повідомлень або підтверджень

Запропонована архітектура є зручною для подальшого розвитку системи. У разі потреби можна додати модуль повідомлень, інтеграцію з банківськими API, розширені графіки або мобільну версію інтерфейсу. При цьому основна логіка системи залишається незмінною.

3.3 Висновки до третього розділу

У третьому розділі обґрунтовано вибір технологій і представлено архітектуру веб-порталу ведення особистого фінансового обліку. Для реалізації клієнтської частини обрано HTML, CSS і JavaScript, для серверної частини - Node.js та Express.js, а для зберігання даних - MySQL. Такий стек дозволяє створити веборієнтовану систему, доступну через браузер і придатну для роботи з персональними фінансовими даними.

Представлено клієнт-серверну архітектуру системи, яка включає клієнтський пристрій, веб-сервер, сервер застосунку, базу даних і зовнішній поштовий сервіс. Описано роль REST API, модуля автентифікації, бізнес-логіки та бази PersonalFinanceDB у загальній структурі порталу.

Також розглянуто реалізацію основних програмних модулів: автентифікації, профілю користувача, фінансових операцій, категорій, бюджетів,

аналітики та адміністрування. Запропонована структура забезпечує модульність, зручність супроводу та можливість подальшого розширення системи.

4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ

4.1 План тестування програмних модулів

Тестування веб-порталу ведення особистого фінансового обліку виконується з метою перевірки працездатності основних програмних модулів, коректності взаємодії з базою даних і відповідності реалізованого функціоналу вимогам. Особлива увага приділяється модулям, які працюють із персональними фінансовими даними: автентифікації, фінансовим операціям, бюджетам, звітам і аналітиці.

Об'єктом тестування є веб-портал, що включає клієнтський інтерфейс, серверну логіку, базу даних PersonalFinanceDB і засоби експорту звітності. Тестування проводиться за основними сценаріями використання: реєстрація, вхід, додавання операції, створення категорії, планування бюджету, перегляд балансу, формування звіту та адміністрування користувачів.

Таблиця 4.1

План тестування програмних модулів

Модуль	Мета тестування	Тестові дії	Очікуваний результат
Реєстрація	Перевірка створення облікового запису	Заповнити форму реєстрації	Користувач створюється в системі
Авторизація	Перевірка входу до порталу	Ввести email і пароль	Відкривається особистий кабінет
Профіль	Перевірка редагування даних	Змінити ПІБ або email	Дані зберігаються і відображаються
Категорії	Перевірка створення категорій	Додати категорію доходу або витрати	Категорія доступна для операцій
Фінансові операції	Перевірка обліку доходів і витрат	Додати операцію з сумою, датою і категорією	Операція зберігається в історії
Бюджети	Перевірка планування витрат	Створити бюджет за категорією	Бюджет відображається у списку

Продовження таблиці 4.1

Аналітика	Перевірка звітів	Сформувати звіт за період	Система показує доходи, витрати і баланс
Експорт	Перевірка збереження результатів	Вивантажити звіт	Файл створюється у вибраному форматі
Адміністрування	Перевірка керування користувачами	Переглянути список користувачів	Адміністратор бачить службові дані

У процесі тестування необхідно перевірити як правильні, так і помилкові сценарії. Наприклад, система повинна коректно реагувати на введення порожніх полів, неправильного пароля, від'ємної суми операції, некоректного періоду бюджету або спроби неавторизованого доступу до сторінок користувача.

Таблиця 4.2

Критерії успішності тестування

Критерій	Очікуване значення
Авторизація	Доступ надається лише зареєстрованим користувачам
Збереження операцій	Доходи та витрати коректно записуються до бази
Баланс	Після додавання операцій показник балансу оновлюється
Категорії	Користувач бачить власні категорії
Бюджет	Система визначає стан виконання бюджету
Звіти	Дані звітів відповідають фінансовим операціям
Безпека	Персональні фінансові дані недоступні стороннім користувачам

Отже, план тестування охоплює всі ключові модулі веб-порталу та дозволяє перевірити готовність системи до практичного використання.

4.2 Тестування системи

Після перевірки окремих модулів було виконано тестування системи в цілому. Основна мета цього етапу полягає в перевірці повного циклу роботи користувача: від входу до системи до створення фінансових операцій, перегляду балансу, налаштування бюджету й формування аналітичного звіту.

Першим етапом перевірено сторінку входу до веб-порталу. Користувач вводить email і пароль, після чого система перевіряє дані та відкриває особистий кабінет. Якщо облікові дані неправильні, система не надає доступ і виводить повідомлення про помилку.

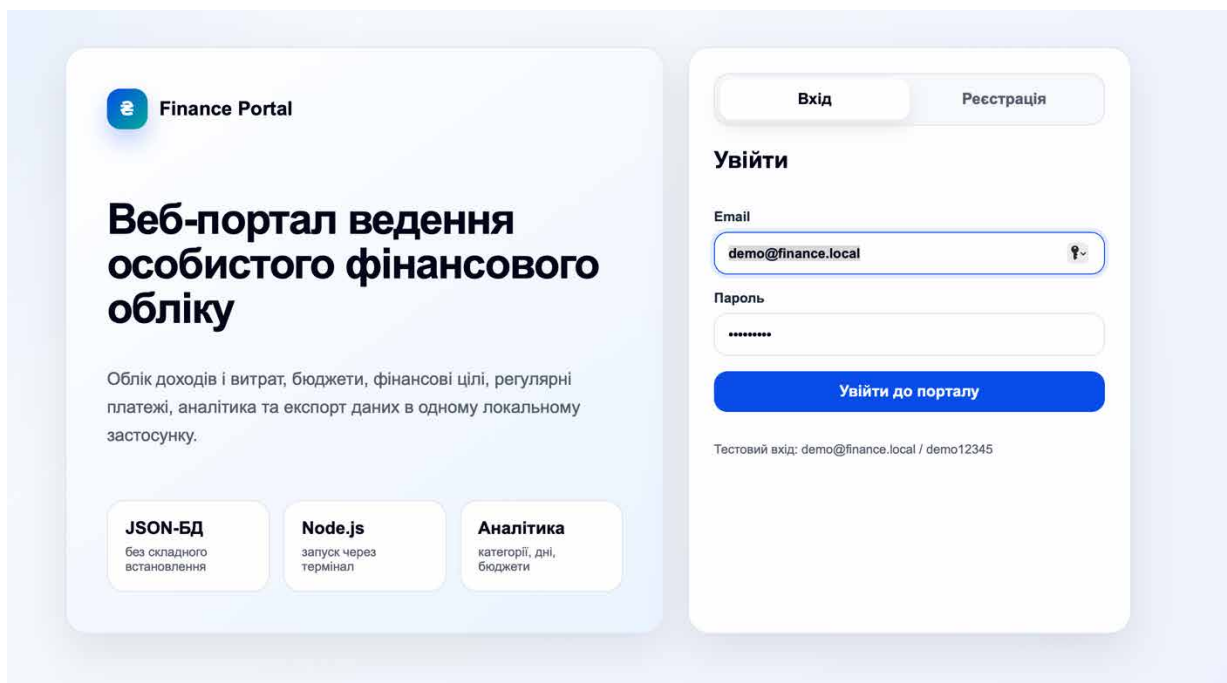


Рис. 4.1. Сторінка входу до веб-порталу ведення особистого фінансового обліку

Після успішної авторизації користувач переходить до головної сторінки системи. На ній відображаються поточний баланс, загальна сума доходів, загальна сума витрат, кількість операцій і короткі аналітичні показники. Така сторінка дозволяє швидко оцінити фінансовий стан користувача.

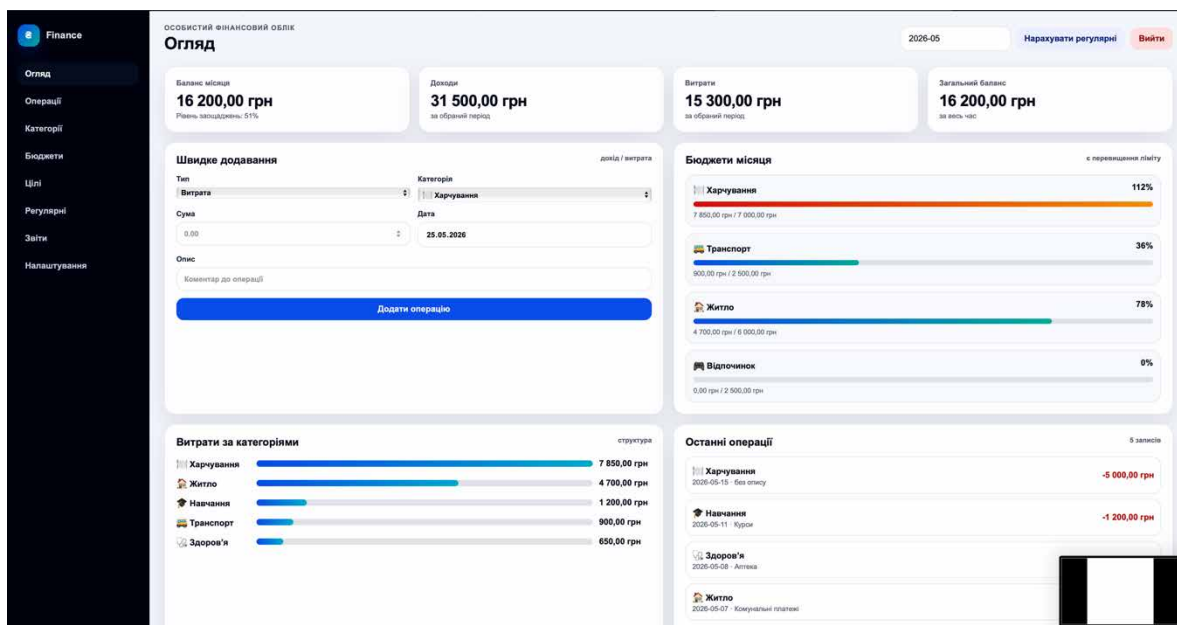


Рис. 4.2. Головна сторінка особистого фінансового кабінету

Далі було перевірено модуль фінансових операцій. Користувач додає новий дохід або витрату, вказує суму, дату, категорію та опис. Після збереження операція з'являється в історії, а баланс автоматично оновлюється. Під час тестування також перевірено фільтрацію операцій за типом, категорією та періодом.

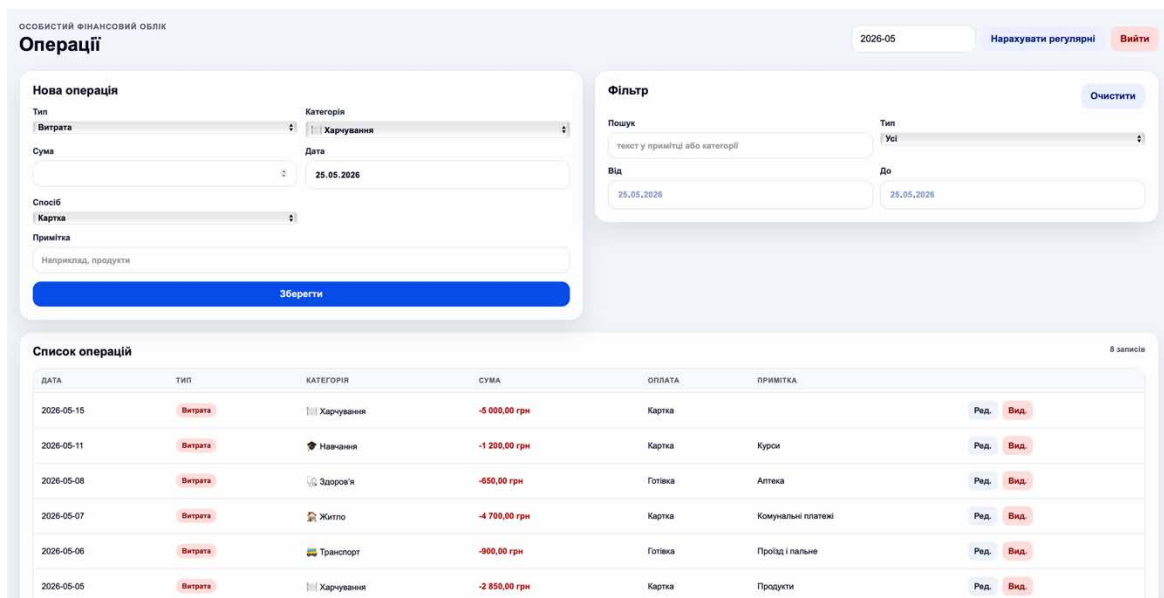


Рис. 4.3. Сторінка додавання та перегляду фінансових операцій

Окремо протестовано модуль категорій. Користувач може створювати категорії доходів і витрат, редагувати їх назви та використовувати під час

внесення операцій. Перевірка показала, що категорії коректно прив'язуються до користувача і не змішуються з даними інших облікових записів.

ОСОБИСТІЙ ФІНАНСОВИЙ ОБЛІК
Категорії

2026-05 [Нарахувати регулярно](#) [Вийти](#)

Додати категорію доходи / витрати

Назва: Тип:

Іконка:

[Додати](#)

Категорії 9 категорій

- Зарплата · дохід
- Підробіток · дохід
- Харчування · витрата
- Транспорт · витрата
- Житло · витрата
- Здоров'я · витрата
- Відпочинок · витрата
- Навчання · витрата
- ти · витрата

Рис. 4.4. Сторінка керування категоріями доходів і витрат

Наступним етапом перевірено модуль бюджетного планування. Користувач створює бюджет для вибраної категорії, задає період і планову суму. Система порівнює фактичні витрати з плановим значенням і визначає стан бюджету. Якщо витрати перевищують ліміт, користувач отримує відповідне попередження.

ОСОБИСТІЙ ФІНАНСОВИЙ ОБЛІК
Бюджети

2026-05 [Нарахувати регулярно](#) [Вийти](#)

Місячний бюджет ліміт за категорією

Місяць: Категорія:

Ліміт:

[Зберегти бюджет](#)

Контроль бюджетів факт / ліміт

Харчування	112%	Видалити
7 850,00 грн / 7 000,00 грн		
Транспорт	36%	Видалити
900,00 грн / 2 500,00 грн		
Житло	78%	Видалити
4 700,00 грн / 6 000,00 грн		
Відпочинок	0%	Видалити
0,00 грн / 2 500,00 грн		

Рис. 4.5. Сторінка планування та контролю бюджету

Модуль звітності перевірено шляхом формування звіту за вибраний період. Система зчитує фінансові операції, групує їх за доходами, витратами й категоріями та виводить підсумкову інформацію. Користувач може переглянути структуру витрат і визначити основні напрями використання коштів.

Рис. 4.6. Сторінка фінансової аналітики та звітності

Результати тестування основних сценаріїв наведено в таблиці 4.3.

Таблиця 4.3

Результати тестування основних сценаріїв

Сценарій	Дії користувача	Очікуваний результат	Результат
Вхід до системи	Введення email і пароля	Відкриття особистого кабінету	Виконано
Додавання доходу	Введення суми, дати, категорії	Дохід збережено в історії	Виконано
Додавання витрати	Введення суми, дати, категорії	Витрату збережено, баланс оновлено	Виконано
Створення категорії	Додавання назви та типу	Категорію створено	Виконано
Створення бюджету	Вибір категорії, періоду і суми	Бюджет збережено	Виконано
Формування звіту	Вибір періоду	Сформовано підсумкову аналітику	Виконано
Перевірка помилкових даних	Введення некоректних значень	Система повідомляє про помилку	Виконано

За результатами тестування встановлено, що система виконує основні функції відповідно до поставлених вимог. Веб-портал забезпечує реєстрацію та вхід користувача, облік фінансових операцій, роботу з категоріями, планування бюджету, перегляд балансу і формування звітності.

4.3 Розгортання системи, склад інсталяційного пакету

Розгортання веб-порталу ведення особистого фінансового обліку передбачає підготовку серверного середовища, встановлення необхідних програмних компонентів, створення бази даних і запуск вебзастосунку. Оскільки система реалізована як веб-портал, користувачі працюють із нею через браузер, а основні обчислення й збереження даних виконуються на сервері.

Схему розгортання системи наведено на рисунку 4.7.

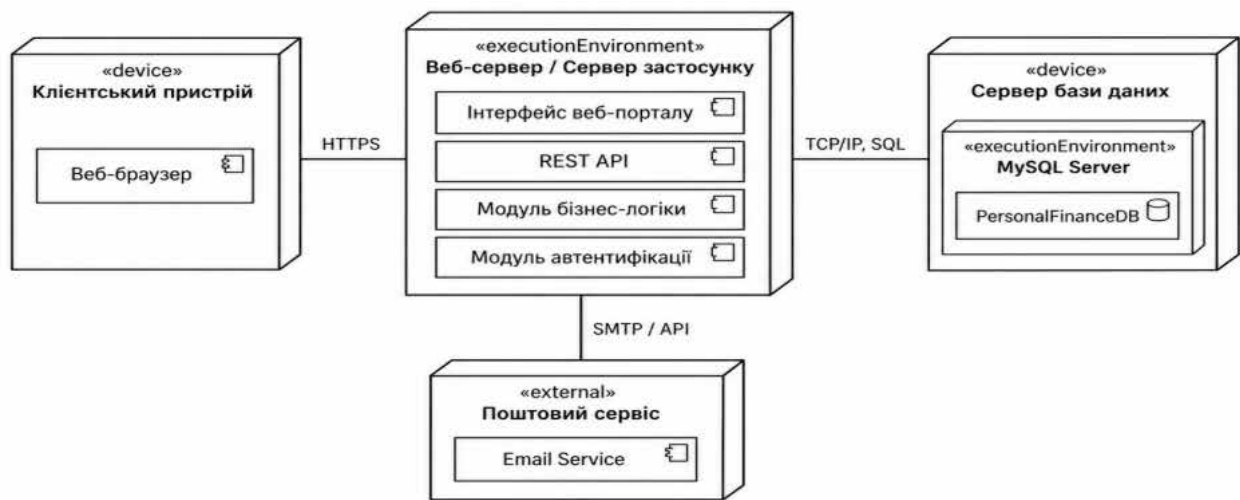


Рис. 4.7. Діаграма розгортання веб-порталу ведення особистого фінансового обліку

Клієнтський пристрій містить веб-браузер, через який користувач відкриває портал. Сервер застосунку містить веб-сервер, REST API, модуль бізнес-логіки та модуль автентифікації. Сервер бази даних містить MySQL Server і базу PersonalFinanceDB. Обмін між клієнтом і сервером виконується через HTTPS, а взаємодія серверної частини з базою даних - через TCP/IP і SQL.

Таблиця 4.4

Середовище розгортання системи

Компонент	Призначення
Клієнтський пристрій	Забезпечує доступ користувача до порталу
Веб-браузер	Відображає інтерфейс користувача
Веб-сервер	Приймає запити та передає їх застосунку
Сервер застосунку	Виконує бізнес-логіку системи
MySQL Server	Забезпечує роботу бази даних

Продовження таблиці 4.4

PersonalFinanceDB	Зберігає фінансові дані користувачів
HTTPS	Захищає обмін даними між клієнтом і сервером
SQL/TCP/IP	Забезпечує взаємодію застосунку з базою даних

Інсталяційний пакет системи повинен містити файли клієнтської частини, серверний код, SQL-скрипти, конфігураційні файли та інструкцію запуску. Склад пакету наведено в таблиці 4.5.

Таблиця 4.5

Склад інсталяційного пакету

Елемент	Призначення
app / server	Серверна логіка, маршрути й обробники запитів
public	HTML, CSS, JavaScript і статичні ресурси
views / templates	Шаблони сторінок веб-порталу
database / schema.sql	Скрипт створення структури бази даних
database / seed.sql	Початкові демонстраційні дані
config	Налаштування підключення до бази та сервера
package.json	Перелік залежностей і команд запуску
README.md	Інструкція з встановлення та експлуатації
exports	Каталог для збереження сформованих звітів

Порядок розгортання включає підготовку сервера, встановлення Node.js, MySQL і веб-сервера, створення бази PersonalFinanceDB, імпорт SQL-скриптів, встановлення залежностей і запуск вебзастосунку. Після запуску адміністратор перевіряє доступність сторінки входу, створює тестового користувача та перевіряє основні функції системи.

Отже, запропонована схема розгортання дозволяє використовувати веб-портал як у локальному середовищі, так і на віддаленому сервері. Склад інсталяційного пакету забезпечує повний цикл встановлення, запуску та подальшого супроводу системи.

4.4 Висновки до четвертого розділу

У четвертому розділі виконано тестування та описано порядок розгортання веб-порталу ведення особистого фінансового обліку. Сформовано план тестування основних програмних модулів, зокрема реєстрації, авторизації, профілю користувача, категорій, фінансових операцій, бюджетів, аналітики, експорту й адміністрування.

Під час тестування перевірено основні сценарії роботи системи: вхід користувача, додавання доходів і витрат, створення категорій, планування бюджету, перегляд балансу та формування фінансових звітів. Результати тестування підтвердили, що система коректно виконує основні функції та забезпечує збереження даних у базі.

Описано схему розгортання системи, яка передбачає роботу клієнтського пристрою, веб-сервера, сервера застосунку та сервера бази даних. Визначено склад інсталяційного пакету, необхідного для встановлення, запуску та супроводу веб-порталу.

Отже, результати четвертого розділу підтверджують готовність системи до практичного використання. Веб-портал забезпечує ведення персонального фінансового обліку, підтримує роботу з бюджетами, формує аналітичні звіти та може бути розгорнутий у локальному або серверному середовищі.

ВИСНОВКИ

У кваліфікаційній роботі виконано розроблення веб-порталу ведення особистого фінансового обліку, призначеного для зручного зберігання, оброблення та аналізу персональних фінансових даних користувача. Розроблена система поєднує функції реєстрації та авторизації, обліку доходів і витрат, керування категоріями, планування бюджету, перегляду балансу, формування історії операцій та створення аналітичних звітів. Такий підхід відповідає потребі у цифровому інструменті, який не просто накопичує фінансові записи, а допомагає користувачу оцінювати власну фінансову поведінку та приймати більш обґрунтовані рішення. У вихідній частині роботи зазначено, що актуальність теми пов'язана з потребою у зручних веборієнтованих рішеннях для фінансового обліку, аналізу операцій, планування бюджету та формування звітності.

У першому розділі проведено системний аналіз предметної області ведення особистого фінансового обліку. Визначено, що основними процесами системи є реєстрація користувача, ведення фінансових операцій, класифікація доходів і витрат за категоріями, планування бюджетів, перегляд балансу та формування аналітичних відомостей. Розглянуто основні ролі взаємодії з порталом: гість, користувач і адміністратор. Користувач є центральним учасником системи, оскільки саме він створює фінансові записи, задає категорії, формує бюджети та аналізує результати власної фінансової активності. У роботі також підкреслено, що предметна область має інформаційно-аналітичний характер, оскільки ефективність системи визначається не лише введенням операцій, а й можливістю їх подальшого узагальнення, контролю та інтерпретації.

У межах аналізу існуючих рішень розглянуто сучасні системи управління особистими фінансами, зокрема Mint, YNAB, PocketGuard, Quicken Simplifi та Monarch Money. Проведене порівняння показало, що більшість таких сервісів

підтримують базовий облік доходів і витрат, категоризацію операцій, планування бюджету та формування звітів. Водночас частина рішень орієнтована переважно на мобільне використання, банківські інтеграції або має обмежені можливості адміністрування й персонального налаштування. Це підтвердило доцільність створення окремого веб-порталу, який поєднає облік, бюджетування, аналітику та адміністрування в межах єдиного програмного середовища.

У другому розділі виконано проектування інформаційного та програмного забезпечення веб-порталу. Побудовано логічну модель даних, у якій визначено основні сутності: користувач, категорія, фінансова операція, бюджет, аналітичний звіт, дохід і витрата. На основі логічної моделі сформовано фізичну структуру бази даних PersonalFinanceDB, що забезпечує зберігання даних про користувачів, категорії, бюджети, фінансові операції та звіти. Структура бази даних передбачає використання первинних і зовнішніх ключів, обмежень цілісності та перевірок коректності даних, що дозволяє уникати дублювання і підтримувати узгодженість фінансової інформації.

Також у другому розділі розроблено компонентну та пакетну структуру програмної системи. Веб-портал поділено на клієнтську частину, серверну частину та рівень даних. Клієнтська частина забезпечує взаємодію користувача з інтерфейсом, серверна частина реалізує бізнес-логіку, автентифікацію, обробку фінансових операцій, роботу з категоріями, бюджетами та аналітикою, а рівень даних відповідає за збереження інформації. Окремо сформовано алгоритмічне забезпечення для формування звітів доходів і витрат, контролю виконання бюджету, оновлення балансу та групування фінансових операцій за категоріями.

У третьому розділі обґрунтовано вибір технологій реалізації веб-порталу. Для клієнтської частини обрано HTML, CSS і JavaScript, що дозволяє створити доступний і зрозумілий веб-інтерфейс із формами введення, таблицями, фільтрами та аналітичними блоками. Для серверної частини використано Node.js та Express.js, які забезпечують маршрутизацію запитів, реалізацію REST API, автентифікацію користувачів і взаємодію з базою даних. Для зберігання

інформації обрано MySQL, оскільки реляційна модель добре відповідає структурі персонального фінансового обліку, де користувач пов'язаний із категоріями, операціями, бюджетами та звітами.

Архітектуру системи побудовано за клієнт-серверним принципом. Користувач працює з порталом через веб-браузер, сервер застосунку обробляє запити та виконує прикладну логіку, а сервер бази даних забезпечує централізоване зберігання фінансових записів. Такий підхід спрощує супровід програмного забезпечення, дозволяє оновлювати систему централізовано та створює умови для подальшого розширення функціоналу. У реалізації передбачено основні програмні модулі: автентифікації, профілю користувача, фінансових операцій, категорій, бюджетів, аналітики, звітності та адміністрування.

У четвертому розділі виконано тестування та описано розгортання системи. Сформовано план тестування програмних модулів, який охоплює перевірку реєстрації, авторизації, роботи з профілем, категоріями, фінансовими операціями, бюджетами, звітами, експортом і адмініструванням. Під час тестування перевірено типові сценарії використання: створення облікового запису, вхід до системи, додавання доходів і витрат, створення категорій, формування бюджету, перегляд балансу та побудова аналітичного звіту. За результатами перевірки встановлено, що основні функції системи працюють коректно та відповідають поставленим вимогам.

Описано порядок розгортання веб-порталу, який передбачає підготовку серверного середовища, встановлення необхідних програмних компонентів, створення бази PersonalFinanceDB, імпорт SQL-структури, налаштування підключення та запуск вебзастосунку. Визначено склад інсталяційного пакету, до якого входять серверна частина, клієнтські файли, шаблони сторінок, SQL-скрипти, конфігураційні файли, документація та каталог для експортованих звітів. Це забезпечує можливість розгортання системи як у локальному середовищі, так і на віддаленому сервері.

Практичним результатом кваліфікаційної роботи є веб-портал, який дає змогу користувачу вести особистий фінансовий облік у структурованому цифровому середовищі. Система забезпечує фіксацію доходів і витрат, розподіл операцій за категоріями, контроль бюджетних лімітів, перегляд фінансової історії, аналіз структури витрат і формування звітів. Завдяки цьому користувач отримує інструмент для регулярного контролю фінансового стану, виявлення основних напрямів витрат і підвищення рівня фінансової дисципліни.

Отже, мету кваліфікаційної роботи досягнуто. Розроблено та обґрунтовано програмне забезпечення веб-порталу ведення особистого фінансового обліку, яке поєднує облікові, аналітичні та бюджетні функції в єдиній інформаційній системі. Подальший розвиток системи може бути пов'язаний з додаванням імпорту банківських виписок, інтеграцією з платіжними сервісами, розширенням графічної аналітики, впровадженням повідомлень про перевищення бюджету, резервним копіюванням даних і створенням адаптивної мобільної версії веб-порталу.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 25.05.2026).
2. Про інформацію : Закон України від 02.10.1992 № 2657-XII. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/2657-12> (дата звернення: 25.05.2026).
3. Про захист інформації в інформаційно-комунікаційних системах : Закон України від 05.07.1994 № 80/94-ВР. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80> (дата звернення: 25.05.2026).
4. Про електронні документи та електронний документообіг : Закон України від 22.05.2003 № 851-IV. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/851-15> (дата звернення: 25.05.2026).
5. Про електронну ідентифікацію та електронні довірчі послуги : Закон України від 05.10.2017 № 2155-VIII. Верховна Рада України. URL: <https://zakon.rada.gov.ua/laws/show/2155-19> (дата звернення: 25.05.2026).
6. ДСТУ ISO/IEC/IEEE 12207:2018. Інженерія систем і програмних засобів. Процеси життєвого циклу програмних засобів. Київ : ДП «УкрНДНЦ», 2018.
7. ДСТУ ISO/IEC 25010:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів. Київ : ДП «УкрНДНЦ», 2016.
8. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ : ДП «УкрНДНЦ», 2016.
9. ISO/IEC 9075-1:2023. Information technology. Database languages SQL. Part 1: Framework (SQL/Framework). Geneva : International Organization for

Standardization, 2023. URL: <https://www.iso.org/standard/76583.html>(дата звернення: 25.05.2026).

10. OMG Unified Modeling Language. Version 2.5.1. Object Management Group, 2017. URL: <https://www.omg.org/spec/UML/2.5.1> (дата звернення: 25.05.2026).

11. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2016. 816 p.

12. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2020. 736 p.

13. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Boston : Addison-Wesley, 2004. 208 p.

14. Elmasri R., Navathe S. B. Fundamentals of Database Systems. 7th ed. Boston : Pearson, 2016. 1272 p.

15. Connolly T., Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management. 6th ed. Harlow : Pearson, 2015. 1440 p.

16. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : doctoral dissertation. Irvine : University of California, 2000. URL: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (дата звернення: 25.05.2026).

17. MDN Web Docs. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 25.05.2026).

18. MDN Web Docs. CSS: Cascading Style Sheets. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 25.05.2026).

19. MDN Web Docs. JavaScript Guide. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 25.05.2026).

20. Node.js Documentation. Introduction to Node.js.
URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (дата звернення: 25.05.2026).
21. Express.js. Node.js web application framework.
URL: <https://expressjs.com> (дата звернення: 25.05.2026).
22. MySQL Documentation. MySQL Reference Manual. Oracle Corporation.
URL: <https://dev.mysql.com/doc/> (дата звернення: 25.05.2026).
23. MySQL 8.4 Reference Manual. Oracle Corporation.
URL: <https://dev.mysql.com/doc/refman/8.4/en/> (дата звернення: 25.05.2026).
24. NGINX Documentation. Admin Guide.
URL: <https://docs.nginx.com/nginx/admin-guide/> (дата звернення: 25.05.2026).
25. Apache HTTP Server Documentation. Version 2.4.
URL: <https://httpd.apache.org/docs/2.4/> (дата звернення: 25.05.2026).
26. OWASP Application Security Verification Standard. OWASP Foundation. URL: <https://owasp.org/www-project-application-security-verification-standard/> (дата звернення: 25.05.2026).
27. OWASP Cheat Sheet Series. Authentication Cheat Sheet. OWASP Foundation.
URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата звернення: 25.05.2026).
28. OWASP Cheat Sheet Series. Authorization Cheat Sheet. OWASP Foundation.
URL: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html (дата звернення: 25.05.2026).
29. YNAB. Features. You Need A Budget.
URL: <https://www.ynab.com/features> (дата звернення: 25.05.2026).
30. PocketGuard. Budgeting App & Finance Planner.
URL: <https://pocketguard.com> (дата звернення: 25.05.2026).
31. Quicken. Quicken Simplifi.
URL: <https://www.quicken.com/simplifi/> (дата звернення: 25.05.2026).

32. Monarch Money. Personal Finance App.
URL: <https://www.monarchmoney.com> (дата звернення: 25.05.2026).
33. Investopedia. Is Credit Karma a Good Replacement for Mint?
URL: <https://www.investopedia.com/is-credit-karma-a-good-replacement-for-mint-8718563> (дата звернення: 25.05.2026).
34. W3C. Web Content Accessibility Guidelines (WCAG) 2.2.
URL: <https://www.w3.org/TR/WCAG22/> (дата звернення: 25.05.2026).
35. Git Documentation. Reference Manual. URL: <https://git-scm.com/docs>

ДОДАТОК А

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

Я, Желєзняков Сергій Володимирович, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Веб-портал ведення особистого фінансового обліку» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):

- o ☐ інструменти генеративного ШІ не використовувалися.
- o ☐ інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL,

_____ інші (вказати назву)) були використані для:

- ☐ перекладу іншомовних джерел;
- ☐ стилістичного редагування та перевірки граматики;
- ☐ допомоги у написанні/відлагодженні програмного коду;
- ☐ інше: _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

« » _____ 2026 р. _____ **Сергій ЖЕЛЄЗНЯКОВ**