

**Державний торговельно – економічний університет
Кафедра комп'ютерних наук та інформаційних систем**

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка Web-застосунку «Облік власних фінансів»

Студента 4 курсу, 8 групи
спеціальності
122 «Комп'ютерні науки»

Чеховського
Данила
Сергійовича

підпис студента

Доктор педагогічних наук,
доцент

Підгорна
Тетяна
Володимирівна

підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2023

Київський національний торговельно-економічний університет

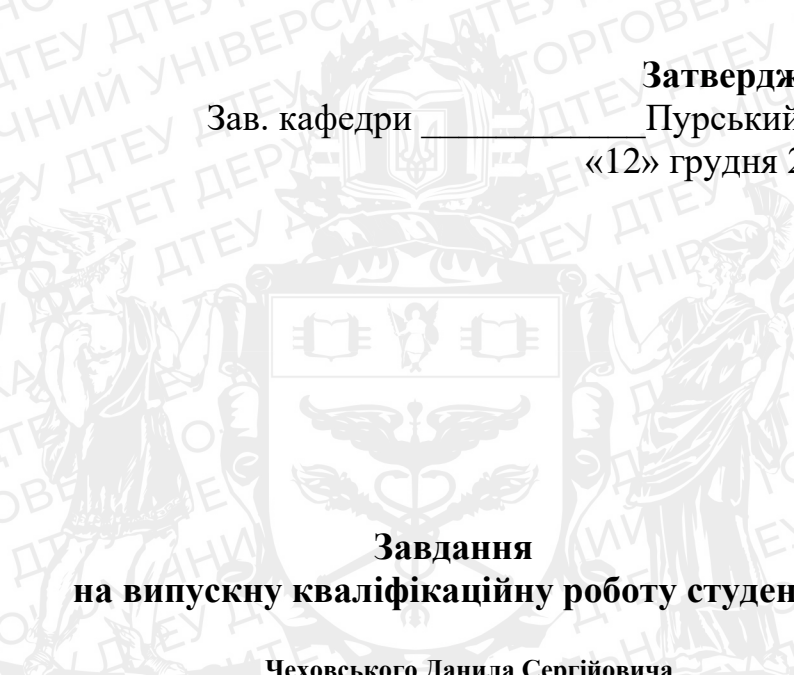
Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____

Затверджую

Пурський О.І.

«12» грудня 2022р.



**Завдання
на випускн кваліфікаційну роботу студенту**

Чеховського Данила Сергійовича

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи

«Розробка Web-застосунку «Облік власних фінансів»»

Затверджена наказом ректора від «09» грудня 2022 р. № 3332

2. Строк здачі студентом закінченої роботи 30 травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: розробка і практична реалізація застосунку для виконання обліку фінансів.

Об'єкт дослідження: процес розробки web-застосунку з обліку фінансів.

Предмет дослідження: web-застосунок з обліку власних фінансів

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Підгорна Т.В.	15.12.2022 р.	15.12.2022 р.
2	Підгорна Т.В.	15.12.2022 р.	15.12.2022 р.
3	Підгорна Т.В.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Теоретико-методичні основи обліку власних фінансів у Web-застосунку.

1.1. Теоретичний аналіз сучасних підходів до обліку фінансів

1.2. Проектування й розробка Web-сервісів

1.3. Опис функціональної моделі

1.4. Огляд існуючих сервісів для обліку фінансів. Доцільність та актуальність розробки застосунку обліку власних фінансів

1.5. Висновки до розділу

РОЗДІЛ 2. Розробка моделі Web-застосунку

2.1. Обґрунтування використаних мов програмування

2.2. Особливості програмування на базі мови PHP

2.3. Особливості програмування на базі мови SQL

2.4. Огляд засобів створення web-застосунку обліку власних фінансів

2.5. Висновки до розділу

РОЗДІЛ 3. Практична реалізація застосунку обліку власних фінансів

3.1. Програмна реалізація застосунку та його особливості

3.2. Інструкція щодо використання

3.3. Висновки до розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи
-------	---	-------------------------------

№	Назва завдання	За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.10.2022	01.10.2022
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2022	15.12.2022
3	Вступ	03.02.2023	
4	РОЗДІЛ 1. Теоретико-методичні основи обліку власних фінансів у Web-застосунку.	28.02.2023	
5	РОЗДІЛ 2. Розробка моделі Web-застосунку	06.04.2023	
6	РОЗДІЛ 3. Практична реалізація застосунку обліку власних фінансів	12.05.2023	
7	Висновки	15.05.2023	
8	Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику	20.05.2023	
9	Попередній захист випускної кваліфікаційної роботи	26.05.2023	
11	Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи	27.05.2023	
12	Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі	30.05.2023	
13	Публічний захист випускної кваліфікаційної роботи	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

9. Керівник випускної кваліфікаційної роботи

Підгорна Т.В.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи

Керівник випускної кваліфікаційної роботи

30.05.2023 р.

(підпис, дата)

13. Висновок про випускню кваліфікаційну роботу

Випускна кваліфікаційна робота студента Чеховський Д.С.

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми

Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри

Пурський О.І.

(підпис, прізвище, ініціали)

« » 2023 р.

ЗМІСТ

АНОТАЦІЯ	7
ВСТУП.....	8
РОЗДІЛ 1 Теоретико-методичні основи обліку власних фінансів у Web-застосунку.	10
1.1. Теоретичний аналіз сучасних підходів до обліку фінансів.....	10
1.2. Проектування й розробка Web-сервісів	13
1.3. Опис функціональної моделі.....	14
1.4. Огляд існуючих сервісів для обліку фінансів. Доцільність та актуальність розробки застосунку обліку власних фінансів.....	16
1.5. Висновки до розділу.....	19
РОЗДІЛ 2 Розробка моделі Web-застосунку	20
2.1. Обґрунтування використаних мов програмування	20
2.2. Особливості програмування на базі мови PHP	20
2.3. Особливості програмування на базі мови SQL	23
2.4. Огляд засобів створення web-застосунку обліку власних фінансів.....	25
2.5. Висновки до розділу.....	29
РОЗДІЛ 3 Практична реалізація застосунку обліку власних фінансів	31
3.1. Програмна реалізація застосунку та його особливості.....	31
3.2. Інструкція щодо використання	33
3.3. Висновки до розділу.....	49
РЕЗУЛЬТАТИ І ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

АНОТАЦІЯ

Розробник: Чеховський Д. С..

Тема роботи: «Розробка Web-застосунку «Облік власних фінансів»».

У випускній кваліфікаційній роботі здійснено комплексну розробку проекту ведення обліку власних фінансів. Проведено теоретичний аналіз предметної області сфери обліку фінансів та сучасних методів ведення обліку фінансів. Теоретично обґрунтовано основні положення формування і проведення обліку фінансів. Розроблено вимоги та завдання до створення проекту. Використано найбільш оптимальні засоби розробки web-застосунку. Розроблено захищену реляційну базу даних для зберігання користувацької інформації. Створено web-застосунок для ведення обліку власних фінансів.

Ключові слова: облік фінансів, web-застосунок, web-розробка, php, sql, база даних.

ANOTATION

Developer: Chekhovskyi D. S.

Work topic: «Development of a Web application «Accounting of one's own finance»».

In the graduation qualification work, a comprehensive development of the financial accounting project was carried out. A theoretical analysis of the subject area of financial accounting and modern methods of financial accounting was conducted. The main provisions of forming and conducting financial accounting are theoretically substantiated. The requirements and tasks for the creation of the project have been developed. The most optimal web application development tools are used. A secure relational database for storing user information has been developed. A web application for keeping records of one's own finances has been created.

Keywords: financial accounting, web application, web development, php, sql, database.

ВСТУП

Як стверджує стара істина, гроші люблять, коли їх рахують. Використання сучасних гаджетів дозволяє автоматизувати процес обліку фінансів, позбавляючи необхідності здійснення рутинних обчислень. Збереження особистих даних в електронному вигляді є зручним способом. Їх можна зберігати на особистому комп'ютері, телефоні, смартфоні тощо, також можна зберігати дані у хмарних сховищах. Для опрацювання різноманітних даних, зокрема здійснення різних розрахунків, можна застосовувати веб-орієнтовані технології, що дозволяє не встановлюючи відповідне програмне забезпечення використовувати його.

Усі компанії, навіть невеликі фірми, проводять облік фінансів. Це робиться не тільки через вимоги контролюючих органів, але й тому, що це є необхідною умовою їх життєдіяльності. Компанії створюються з метою отримання прибутку. Крім того, будь-хто може вести власний облік фінансів для контролю та планування фінансових операцій. Контроль над персональними фінансами допоможе краще розуміти обсяги доходів та шляхи їх витрачання. Ведення фінансів дає можливість зрозуміти, як і на що ми витрачаємо, а правильні висновки з нього допоможуть нам жити в межах власних фінансових можливостей. Кожен має свій власний підхід та знаходить свою користь у веденні персональних фінансів.

Дослідженню питань обліку фінансів присвячені праці таких дослідників, як Іщенко Я.П., Подолянчук О.А., Коваль Н.І. Атамас П. Й., Атамас О. П. [1-2]. Однак, питання, що стосуються використання веб-орієнтованих технологій для ведення обліку власних фінансів не достатньо розкриті в технічній літературі. Зважаючи на це було обрано тему дослідження «Розробка web-застосунку «Облік власних фінансів»».

Об'єкт дослідження: процес розробки реалізації web-застосунку з обліку фінансів.

Предмет дослідження: web-застосунок з обліку власних фінансів

Мета роботи: розробка і практична реалізація застосунку для виконання обліку власних фінансів.

Завдання проекту:

1. Проаналізувати сферу обліку фінансів, розробити вимоги до архітектури проекту з обліку власних фінансів;
2. Провести дослідження та аналіз доступних методів та засобів виконання поставлених задач розробки web-застосунку з обліку власних фінансів, з яких обрати найбільш оптимальні;
3. Розробити веб-застосунок, що відповідає поставленим вимогам;
4. Розробити для проекту супроводжуючу документацію.

Методи дослідження. Теоретичною основою дослідження є науково-технічні положення, що подані в працях провідних вчених.

Для практичного вирішення поставлених задач використовувалися такі методи:

- загальнонауковий аналітичний метод;
- методи теорії БД для формування інформаційно-логічної моделі предметної області та БД;
- методи Web-розробки для створення функціонального застосунку ведення обліку фінансів.

Практичне значення. Отримані результати дослідження в сфері обліку фінансів матимуть практичне значення, допомагаючи у розробці ефективного фінансового планування, аналізу фінансових даних, управлінні ризиками та розвитку інноваційних рішень у фінансовій сфері. Вони сприятимуть зростанню бізнесу та досягненню фінансової стабільності.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 15 найменувань і містить 44 сторінки основного тексту, 37 рисунків і 1 таблицю.

РОЗДІЛ 1 Теоретико-методичні основи обліку власних фінансів у Web-застосунку.

1.1. Теоретичний аналіз сучасних підходів до обліку фінансів.

Фінансовий облік ведеться згідно Закону України «Про бухгалтерський облік та фінансову звітність в Україні»[3].

Фінансовий облік визначається як система збору та узагальнення облікової інформації, яка служить для бухгалтерського фіксування та реєстрації господарських операцій, а також складання фінансової звітності. Цей вид обліку охоплює широкий спектр бухгалтерських аспектів, включаючи збереження даних про активи організації, такі як основні засоби, нематеріальні активи, орендоване майно, фінансові вкладення, кошти та їх еквіваленти, інші оборотні активи, а також про зобов'язання, капітал та інші джерела формування майна та господарські процеси.

Основним завданням обліку фінансів є забезпечення користувачів фінансової звітності зрозумілою, достовірною та доречною інформацією про діяльність підприємства для прийняття рішень.

Деякі з переваг ведення обліку власних доходів й витрат:

- Ведення обліку власних коштів сприяє стабілізації і покращенню фінансової ситуації.
- Надається можливість дізнатися багато нового про рух коштів.
- Можна дізнатися кількість витрачених грошей протягом місяця.
- Також можна дізнатися, на що насправді витрачається не така сума коштів як здається.
- Перед очима буде наглядна картина того, куди йдуть гроші.
- Можливо дізнатися, які статті витрат з'їдають більшу частину грошей, а які – меншу.
- Можливість тримати фінанси під контролем.

Існують різні способи ведення обліку фінансів, які використовуються підприємствами, такі як:

1. Традиційний паперовий облік: це старовинний метод, при якому фінансова інформація реєструється на паперових документах, таких як рахунки, журнали та книги обліку. Цей підхід вимагає фізичного зберігання та обробки документів і може бути трудомістким.

2. Електронний облік: це сучасний метод, при якому фінансові дані записуються та зберігаються за допомогою комп'ютерних програм і електронних систем. Цей підхід дозволяє автоматизувати процеси обліку, зберігати дані в цифровому форматі і швидко отримувати необхідну інформацію.

3. Хмарний облік: це форма обліку, при якій фінансові дані зберігаються та обробляються в хмарних сервісах. Це дозволяє підприємствам отримувати доступ до своїх фінансових даних з будь-якого місця та пристрою з підключенням до Інтернету.

4. Офлайн-облік: це метод, при якому фінансові дані зберігаються та обробляються без підключення до Інтернету. Це може бути використано у випадку обмеженого доступу до Інтернету або для збереження конфіденційної інформації.

5. Інтегрована система управління: це комплексна система, яка поєднує облік фінансів з іншими аспектами управління підприємством, такими як логістика, продажі, виробництво та інші. Цей підхід дозволяє забезпечити взаємозв'язок та автоматичний обмін інформацією між різними функціональними областями підприємства.

Вибір форми ведення обліку фінансів залежить від розміру підприємства, його потреб у доступності, швидкості та надійності обробки фінансових даних, а також від наявних технічних ресурсів та фінансових можливостей.

Веб-застосунки, іноді також називаються веб-системами, представляють собою різноманітні програмні продукти, до яких можна отримати доступ через веб-інтерфейс. Розробка веб-застосунків є необхідною для різних комерційних

організацій, тому їх функціонал часто включає потужні інструменти, спрямовані на задоволення бізнес-потреб.

Створення веб-застосунків дозволяє вирішувати найрізноманітніші завдання комерційних компаній. Так, наприклад, за допомогою веб-застосунків можна вести облік часу роботи всіх співробітників, навіть якщо компанія має мережу віддалених філій, здійснювати облік вантажних або пасажирських перевезень, здійснювати моніторинг діяльності компанії, керувати роботою персоналу, а також нараховувати заробітну плату.

Створення веб-сайту для обліку фінансів має кілька важливих переваг:

1. **Централізована доступність:** Веб-сайт дає можливість отримувати доступ до фінансових даних з будь-якого пристрою з підключенням до Інтернету. Це дозволяє працювати з фінансовою інформацією в режимі реального часу і спрощує обмін даними між різними користувачами.

2. **Зручність інформаційного збереження:** Веб-сайт обліку фінансів дозволяє зберігати всю фінансову інформацію в електронному форматі. Це робить його легко доступним для пошуку, аналізу і звітності, а також захищає дані від втрати або пошкодження.

3. **Автоматизація інтеграції даних:** Веб-сайти обліку фінансів можуть бути підключені до інших фінансових систем і програм, що дозволяє автоматизувати процес обміну даними. Це спрощує введення інформації, уникнення помилок і забезпечує швидку і точну обробку фінансових даних.

4. **Аналітика та звітність:** Веб-сайт обліку фінансів надає можливість проводити аналіз фінансової інформації і створювати звіти з використанням різноманітних параметрів і показників. Це допомагає виявляти тенденції, виявляти проблемні області і приймати обґрунтовані фінансові рішення.

5. **Безпека та конфіденційність:** Завдяки веб-сайту обліку фінансів можна встановлювати рівні доступу для різних користувачів, обмежуючи їх права доступу до конфіденційної фінансової інформації. Це дозволяє забезпечити безпеку і збереження даних від несанкціонованого доступу.

Веб-застосунки можуть бути корисними не тільки для бізнесу. Некомерційні організації теж часто потребують веб-застосунків. Так, популярні останнім часом електронні щоденники — яскравий приклад веб-застосунків некомерційного характеру. За допомогою електронних щоденників взаємодія між школою, кожним окремо взятим учнем і його батьками стає більш прозорою.

На основі виконаних досліджень можна зробити висновок, що саме використання веб-застосунку для ведення обліку фінансів, тому що даний спосіб ведення обліку має череду значних переваг, таких як централізований доступ до даних з будь-якого пристрою з Інтернет-підключенням, зручне зберігання і аналіз інформації, автоматизація обміну даними, можливість аналізу та створення звітів, а також забезпечення безпеки та контролю доступу до фінансової інформації.

1.2. Проектування й розробка Web-сервісів

Проектування веб-сайту обліку власних фінансів включає кілька етапів:

Визначення вимог: На цьому етапі встановлюються основні вимоги до веб-сайту. Це включає визначення функціональності, яку повинен надавати сайт (наприклад, можливість ведення бюджету, відстеження витрат, генерація звітів тощо), а також визначення потрібних ролей та рівня доступу для користувачів.

Проектування інтерфейсу: На цьому етапі розробляється дизайн веб-сайту та його інтерфейс. Важливо врахувати зручність використання, логіку навігації, зручність розміщення інформації та елементів управління.

Розробка функціональності: За допомогою програмування та розробки веб-додатків реалізується необхідна функціональність веб-сайту. Це можуть бути модулі для ведення бухгалтерського обліку, реєстрації транзакцій, генерації звітів тощо.

Тестування та відладка: Після розробки веб-сайту проводиться його тестування з метою виявлення та виправлення можливих помилок, недоліків чи проблем з безпекою. Важливо перевірити, чи працює функціональність правильно і чи задовольняє веб-сайт потребам користувачів.

Розгортання та підтримка: Після успішного тестування веб-сайт готовий до розгортання на сервері та доступу для користувачів. Після запуску важливо забезпечити постійну підтримку, включаючи виправлення помилок, оновлення функціоналу та забезпечення безпеки даних.

Розробка веб-застосунків забезпечує повну сумісність при роботі на різних платформах. Якщо звичайний оффлайн-застосунок доводиться перероблювати під певні системні вимоги конкретної платформи, то веб-застосунок завжди і скрізь може бути доступний на будь-якій локальній машині. Все, що потрібно, — це браузер і включений доступ до інтернету.

Усі сучасні веб-застосунки володіють важливими властивостями, серед яких особливо велике значення мають:

- масштабованість веб-системи;
- інтеграція з іншими системами;
- розмежування прав доступу до різного функціоналу;
- зручне розгортання та обслуговування системи.

Наявність подібних властивостей дозволяє використовувати веб-застосунки та веб-системи максимально ефективно і зручно. Наприклад, завдяки властивості масштабованості можна без внесення кардинальних змін розширювати веб-систему для роботи з постійно зростаючим числом користувачів, додавати до неї нові функції. У свою чергу, завдяки зручному розгортанню процес переходу компанії на роботу з тим чи іншим веб-застосунком займає мінімум часу, а в умовах великого завантаження це може бути важливо.

Таким чином, розробка веб-застосунків надає широкі можливості зі створення для компаній багатофункціональних онлайн-інструментів для оптимізації або вирішення різних бізнес-задач.

1.3. Опис функціональної моделі

Основна мета створення будь-якого веб-застосунку — створення такого веб-застосунку, який допомагає користувачеві виконувати свої повсякденні завдання. Для створення таких програм визначаються вимоги, якими повинен задовольняти

застосунок. Для того, щоб більш точно зрозуміти як повинен працювати застосунок, все частіше використовується опис функціональності застосунку через варіанти використання (Use Case або прецеденти).

Діаграма прецедентів є візуальним зображенням взаємозв'язків між акторами та прецедентами у системі. Головна мета цієї діаграми полягає у відображенні варіантів використання, за якими актори взаємодіють з проектованою системою. Це зображення демонструє безліч сутностей або акторів, які взаємодіють з системою шляхом використання різних сценаріїв. Варіант використання використовують для описання послуг, які система надає актору.

Сайтом можуть користуватися тільки зареєстровані користувачі.

Вони можуть виконувати такі дії:

- користувач має можливість зареєструватися та авторизуватися на сайті;
- зареєстрований користувач має можливість переглядати та редагувати свій особистий кабінет;
- є можливість вести записи витрат та доходів;
- переглядати статистику доходів та витрат;
- створювати нові категорії.



Рис. 1.1. Діаграма прецедентів

Таким чином, даний програмний продукт є цілісною системою з багатьма складовими елементами, кожен з яких має свої функції та призначення для забезпечення ефективної та якісної взаємодії з користувачем.

1.4. Огляд існуючих сервісів для обліку фінансів. Доцільність та актуальність розробки застосунку обліку власних фінансів.

Під час розгляду аналогів веб-сайтів було відібрано три вебзастосунки (вигляд сайтів зображено на рисунках 1.2 – 1.4):

- вебзастосунок «remonline.ua»: <https://remonline.ua/>[4];
- вебзастосунок «mybudget.ws»: <https://mybudget.ws/>[5];
- вебзастосунок «finmap.online»: <https://finmap.online/>[6].

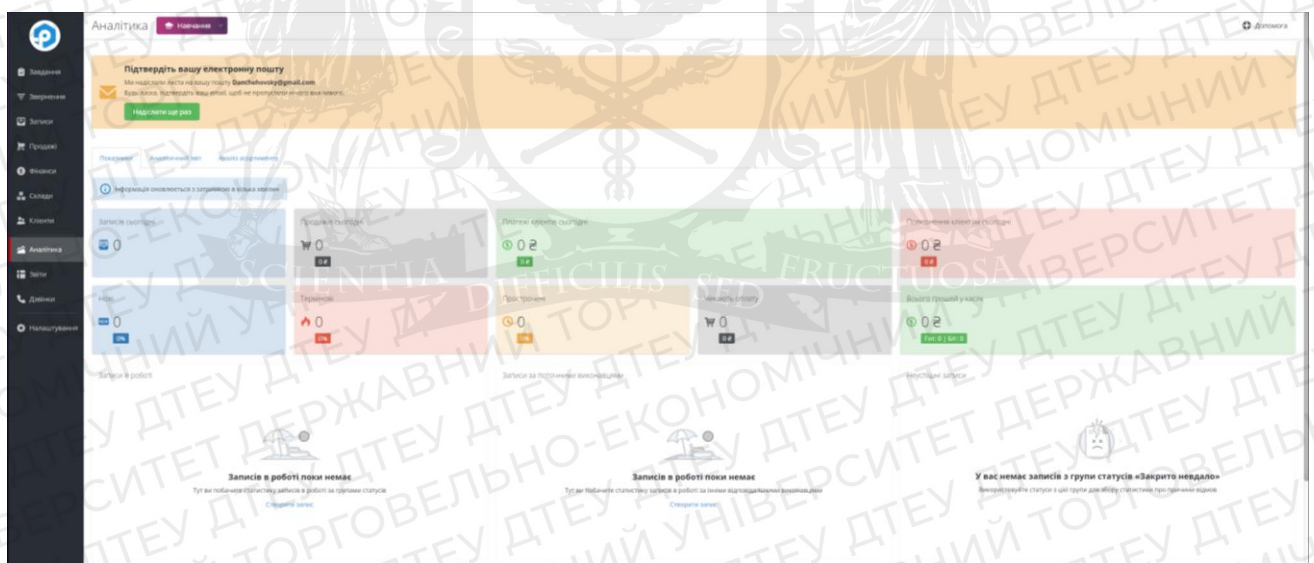


Рис. 1.2. Вебзастосунок обліку фінансів «remonline»

Вебзастосунок «remonline.ua» має в наявності багатокористувацький режим. Також вебзастосунок має можливість режиму планувальника, звітів, відстеження динаміки залишків, обробки програмою банківських смс для автоматичного введення транзакцій. Резервне копіювання даних шляхом відправки бекапов на пошту та приємний дизайн з присвоєнням картинок категоріям. Безкоштовна версія застосунку має доступ до майже всіх функцій сервісу.

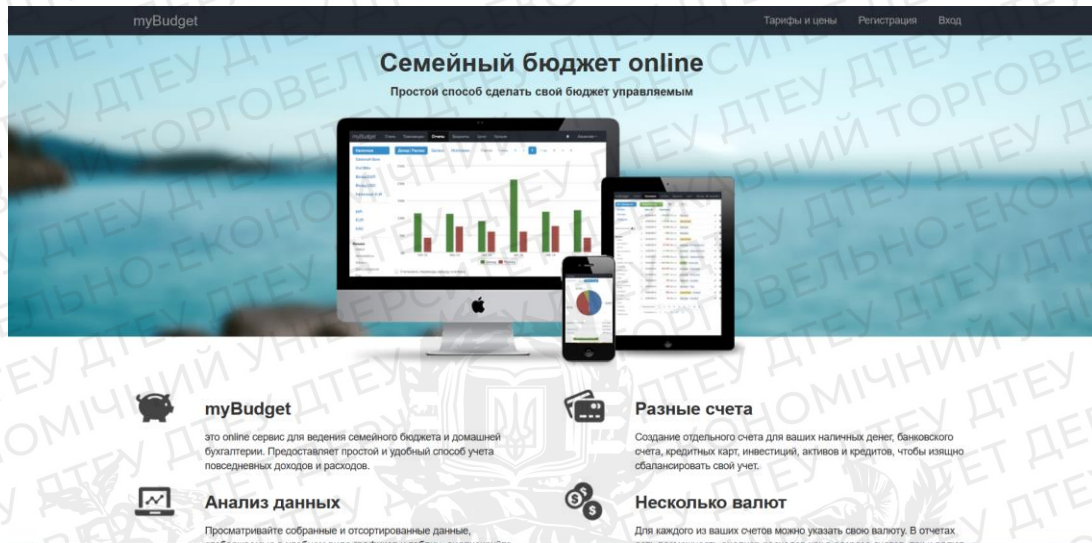


Рис. 1.3. Вебзастосунок обліку фінансів «Mybudget»

Вебзастосунок «Mybudget» має ведення обліку по декількох рахунках: готівкові, карти, заощадження, в наявності синхронізація комп'ютерної та мобільної версії, зручна побудова звітів у вигляді графіків і таблиць, можливість присвоєння ярликів категоріям. Фінансова мета. Програма допомагає ставити цілі і рухатися до їх реалізації. Відсутність багатокористувацького інтерфейсу та недостатньо можливостей безкоштовної версії.

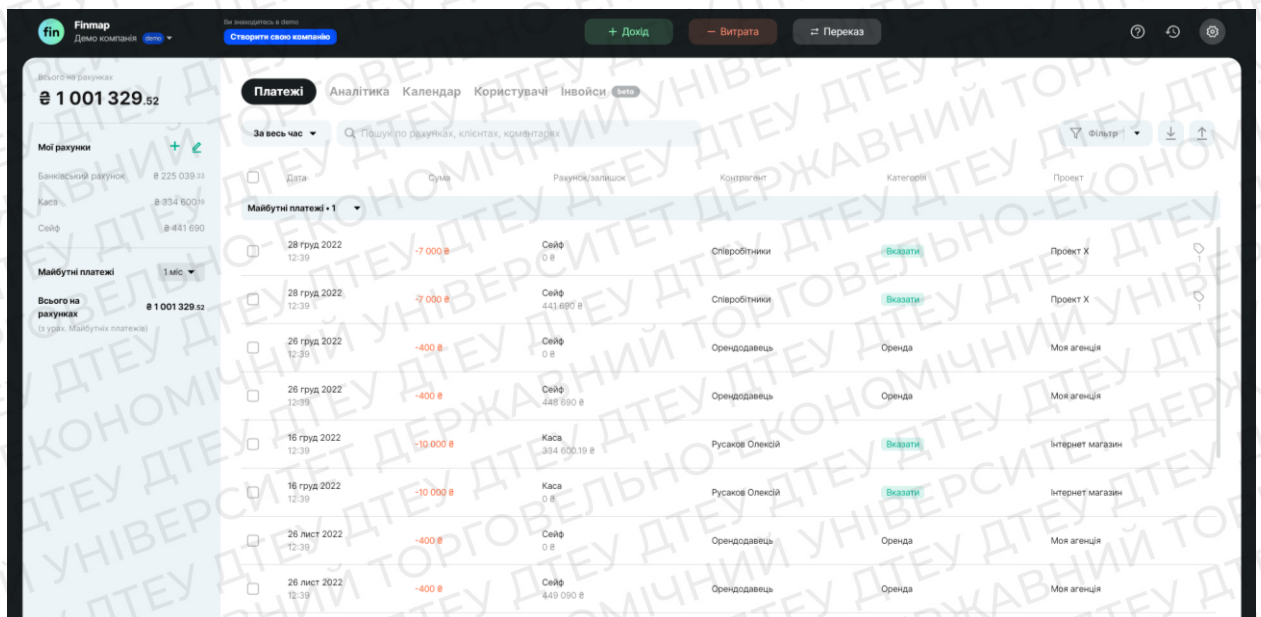


Рис. 1.4. Вебзастосунок обліку фінансів «Finmap»

Вебзастосунок «Finmap» розрахований на багатокористувацький режим. В наявності є багато можливостей для аналізу фінансів, є різні таблиці і графіки, порівняння доходів і витрат за періодами. Можливість аналізу дебіторської та кредиторської заборгованості. Також присутня функція створення псевдо-компанії для простішого ознайомлення з функціями сервісу. Але бувають помилки з синхронізацією даних.

Загальне порівняння функціональності оглянутих веб-застосунків зображено на таблиці 1.1.

Табл.1.1.

“Порівняння сайтів”

Назва	Багатокористувацький режим	Синхронізація між платформами	Багатофункціональна безкоштовна версія
Remonline	+	+	+
Mybudget	-	+	-
Finmap	+	+	+

Призначення створення веб-застосунку обліку власних фінансів – спростити ведення обліку доходів і витрат.

Даний веб-застосунок створюється для спрощення ведення обліку власних фінансів. Для виконання поставлених цілей необхідно стало вирішення кількох важливих технічних завдань:

- Забезпечення простоти і інтуїтивної зрозумілості інтерфейсу застосунку;

- забезпечення відображення необхідної для користувача інформації;
- забезпечення анонімності всіх користувачів;
- розробка простої та привабливої версії сайту для мобільних телефонів та планшетів.

1.5. Висновки до розділу

В даному розділі було проведено дослідження сфери обліку фінансів, в результаті чого зроблено висновок, що система обліку фінансів є важливим інструментом для підприємств та організацій, оскільки дозволяє зрозуміти фінансовий стан, ефективність та стійкість діяльності. Вона використовується для прийняття рішень, планування, оцінки ризиків та звітування перед зацікавленими сторонами, такими як власники, керівництво, інвестори та фінансові установи.

Було зроблено огляд та порівняння існуючих сервісів ведення обліку фінансів, в результаті чого для спрощення ведення обліку було вирішено створити даний веб-застосунок.

Обрана тема для розробки веб-застосунку є дуже актуальною та перспективною. В даному розділі була поставлена мета, цілі та задачі створення веб-застосунку. Був проаналізований функціонал проекту та, на основі проведеного аналізу, була побудована діаграма варіантів використання. На даному етапі розробки поставлено чіткі питання, які має вирішувати готовий проект.

РОЗДІЛ 2 Розробка моделі Web-застосунку

2.1. Обґрунтування використаних мов програмування

Для розробки функціональної частини веб-сторінки було обрано мову програмування PHP.

PHP[7] є універсальним інструментом, який використовується переважно для розробки динамічних веб-сторінок. Завдяки широкому спектру контент-менеджерів, таких як Moodle і WordPress, які базуються на PHP, ця мова стає найбільш оптимальним вибором для створення блогів, систем управління навчанням і електронно-комерційних веб-сайтів.

Дана мова програмування має вбудовані модулі підключення до баз даних, що значно спрощує та пришвидшує створення веб-застосунків зі значною залежністю від серверної частини та бази даних.

Для розробки функціональної частини веб-сторінки було обрано мову програмування SQL.

Мова SQL є стандартною мовою для взаємодії з реляційними базами даних і користується широкою популярністю в бізнес-середовищі. Вона забезпечує простий доступ та можливість зміни даних, що робить її привабливою для багатьох компаній.

Особливо велику популярність мова SQL здобула в фінансовій галузі, де бази даних використовуються широко. Банківські додатки та платіжні процесори є найпоширенішими прикладами застосування SQL. Оскільки безпека онлайн-транзакцій є пріоритетом для банків, особливу увагу приділяють заходам безпеки, і тут SQL забезпечує потрібні функції разом із надійною підтримкою баз даних.

2.2. Особливості програмування на базі мови PHP

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим кодом. PHP спеціально сконструйований для веб-розробок, і його код може впроваджуватися безпосередньо в HTML.

Замість рутинного виведення HTML-коду командами мови (як це відбувається, наприклад, Perl або C), скрипт PHP містить HTML з вбудованим кодом. Код PHP відокремлюється спеціальними початковими та кінцевими тегами `<?php і ?>`, які дозволяють "перемикатися" в "PHP-режим" і виходити з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері та генерують HTML, який надсилається клієнту. Якби на сервері був розміщений скрипт, подібний до вищенаведеного, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код його зробив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, тому клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипту.

PHP вкрай простий для освоєння, але водночас здатний задовольнити запити професійних програмістів. Не потрібно лякатися довгого списку можливостей PHP. З даною мовою можна швидко розпочати, і вже протягом кількох годин вміти створювати прості PHP-скрипти.

Хоча PHP, головним чином, призначений для роботи в середовищі веб-серверів, область застосування не обмежується тільки цим. Читайте далі і не пропустіть розділ Можливості PHP або, почніть безпосередньо з вступного посібника, якщо вас цікавить виключно веб-програмування.[8]

PHP - скриптова мова загального призначення, що інтенсивно застосовується для розробки веб-застосунків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

В області веб-програмування, зокрема серверної частини, PHP - одна з популярних сценарних мов (разом з JSP, Perl і мовами, використовуваними в ASP.NET).

Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів і додаткових модулів для розробки веб-застосунків. Основні з них:

- автоматичне вилучення POST- і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle Database (OCI8), Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird і InterBase, Paradox File Access, MaxDB, інтерфейс PDO, Redis);
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з cookies і сесіями;
- робота з локальними і віддаленими файлами, сокетамі;
- обробка файлів, що завантажуються на сервер;
- робота з XForms.

При дослідженні функціональності мови було виявлено, що використання PHP має кілька важливих переваг, які призвели до вибору цієї мови програмування:

- **Веб-орієнтованість:** PHP спеціально розроблена для розробки веб-додатків. Вона має велику кількість вбудованих функцій і бібліотек, що полегшують роботу з веб-технологіями, такими як HTTP-запити, обробка форм, робота з базами даних і створення динамічного контенту.
- **Велика спільнота і підтримка:** PHP є однією з найпопулярніших мов програмування для веб-розробки, що означає наявність великої активної спільноти розробників. Це забезпечує доступ до безлічі ресурсів, документації, форумів та плагінів, а також регулярні оновлення і підтримку.
- **Простота використання:** PHP має простий синтаксис, що дозволяє швидко освоювати мову навіть початківцям. Вона також підтримує велику кількість фреймворків, які пропонують готові рішення і шаблони для швидкої розробки веб-додатків.

- **Переносимість:** PHP підтримується на більшості веб-серверів і операційних систем, включаючи Windows, Linux і macOS. Це дозволяє розгорнути PHP-додатки на різних середовищах без зайвих зусиль.
- **Інтеграція з базами даних:** PHP має потужну підтримку для роботи з реляційними базами даних, такими як MySQL, PostgreSQL і Oracle. Вона надає зручні інструменти для з'єднання, запитів і обробки даних з баз даних.
- **Широкий спектр застосувань:** PHP може використовуватися для розробки різноманітних веб-додатків, від простих статичних сайтів до складних веб-порталів і електронних комерційних систем.

2.3. Особливості програмування на базі мови SQL

SQL (мова структурованих запитів) - це мова програмування, призначена для управління і маніпулювання базами даних, він використовується для пошуку та вилучення інформації з величезних баз даних.

SQL відображається в реляційних базах даних, формі бази даних, отриманої з реляційної моделі Едгара Ф. Кодда для зберігання інформації таким чином, щоб вона була доступною та ідентифікованою по відношенню до іншої інформації. Мова спочатку називалася SEQUEL (структурована англійська мова запитів) і була розроблена Дональдом Чемберліном і Реймондом Бойсом в IBM (1974) SQL містить чотири основні категорії з двома додатковими командами, які в основному використовуються для комп'ютерних баз даних. SQL – це мова четвертого покоління, його команди близькі до людської мови. А ось мови програмування Java та C++ – мови третього покоління, які абстрактні для недосвідченого ока.

SQL є широко використовуваною мовою для маніпулювання та запиту даних у реляційних базах даних. Її можна розбити на різні компоненти залежно від їх призначення:

1. **Мова маніпулювання даними (DML)** використовується для керування вмістом бази даних, включаючи операції вилучення, оновлення, видалення та вставки. DML включає команди "SELECT", "INSERT", "UPDATE" та "DELETE", які дозволяють управляти інформацією.

2. Мова визначення даних (DDL) використовується для створення та зміни структури таблиць та індексів. Це включає команди для створення, зміни та видалення таблиць та інших об'єктів бази даних.

3. Мова керування даними (DCL) використовується для управління доступом до бази даних та надання прав користувачам. DCL включає команди "GRANT" і "REVOKE", які використовуються для надання та відкликання прав доступу, забезпечуючи безпеку бази даних і аутентифікацію.

4. Мова запитів даних (DQL) використовується для вибору, вставки, оновлення та видалення даних з бази даних. DQL схожа на DML, оскільки вона дозволяє виконувати запити до бази даних і отримувати необхідну інформацію.

SQL - декларативна мова програмування, застосовувана для створення, модифікації та управління даними в реляційній базі даних, керованої відповідною системою управління базами даних.

Є перш за все інформаційно-логічною мовою, призначеною для опису, зміни і вилучення даних, що зберігаються в реляційних базах даних. SQL вважається мовою програмування, в загальному випадку (без ряду сучасних розширень) не є Тьюринг-повною, тобто в ній неможливо вирахувати результат будь-якої функції, але разом з тим стандарт мови специфікацією SQL / PSM передбачає можливість його процедурних розширень.

Спочатку SQL був основним способом роботи користувача з базою даних і дозволяв виконувати наступний набір операцій:

- створення в базі даних нової таблиці;
- додавання в таблицю нових записів;
- зміна записів;
- видалення записів;
- вибірка записів з однієї або декількох таблиць (відповідно до заданого умовою);
- зміна структур таблиць.

При дослідженні функціональності мови було виявлено, що використання мови програмування SQL має кілька важливих переваг, які призвели до вибору цієї мови програмування:

- **Маніпуляція даними:** SQL дозволяє виконувати різноманітні операції з даними в базі даних, такі як створення, зчитування, оновлення та видалення даних. Вона надає потужні інструменти для роботи зі структурованими даними і здійснення складних запитів до баз даних.
- **Універсальність:** SQL є стандартною мовою запитів для багатьох реляційних баз даних. Це означає, що знання SQL можна застосовувати до різних систем управління базами даних, таких як MySQL, Oracle, SQL Server, PostgreSQL та інші. Це забезпечує переносимість і зручність при роботі з даними.
- **Простота використання:** SQL має простий і зрозумілий синтаксис, що дозволяє легко вивчити основні конструкції мови. Вона використовує декларативний підхід, де розробник описує, що потрібно зробити з даними, а не як це зробити. Це полегшує створення і виконання запитів до баз даних.
- **Швидкість та ефективність:** SQL є оптимізованою для роботи з базами даних і дозволяє виконувати запити ефективно. Вона підтримує індекси, оптимізацію запитів і інші механізми, що сприяють швидкому доступу до даних і оптимальній роботі з базою даних.
- **Безпека:** SQL має вбудовані механізми для забезпечення безпеки даних, такі як контроль доступу і шифрування. Вона дозволяє управляти правами доступу до бази даних і забезпечувати конфіденційність та цілісність даних.
- **Аналітика та звітність:** SQL надає потужні можливості для аналізу даних та створення звітів. Запити можна виконувати для вибірки, групування, сортування та обчислення даних з різних таблиць, що дозволяє отримати цінну інформацію для прийняття рішень.

2.4. Огляд засобів створення web-застосунку обліку власних фінансів

Для розробки дипломного проекту використовувалися програмні середовища OSPanel, phpMyAdmin і Visual Studio Code, у яких використовувалися мови програмування SQL і PHP.

Open Server Panel [9] - це портативне програмне середовище, створене спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів.

Open Server Panel широко використовується з метою розробки, налагодження і тестування веб-проектів, а також для надання веб-сервісів в локальних мережах.

Проект існує вже більше 10 років, він успішно зарекомендував себе як першокласний, простий і надійний інструмент, необхідний кожному веб-майстру.

PhpMyAdmin - це веб-застосунок з відкритим вихідним кодом, розроблений на PHP, який надає веб-інтерфейс для управління Системою Управління Базами Даних (СУБД) MySQL. PhpMyAdmin дозволяє адмініструвати сервер MySQL, виконувати SQL-команди та переглядати дані таблиць і баз даних через веб-браузер. Цей засіб є дуже популярним серед веб-розробників, оскільки дозволяє управляти базами даних MySQL без необхідності безпосереднього введення SQL-запитів.

Проект на даний момент часу локалізований на більш ніж 62 мовах.

Починаючи з версії 3.0.0, phpMyAdmin приєднався до ініціативи GoPHP5 і відмовився від підтримки сумісності з застарілими версіями PHP і MySQL. Для роботи phpMyAdmin 3.0.0 і вище потрібна наявність PHP 5.2 і MySQL 5. Для використання старих версій PHP і MySQL продовжує розвиватися друга гілка скрипта, однак її підтримка обмежується закриттям знайдених вразливостей, нових функцій в неї не додається.

Visual Studio Code [10] — це редактор вихідного коду, який можна використовувати з різними мовами програмування, включаючи C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust. Він базується на структурі Electron, яка

використовується для розробки веб-застосунків Node.js, які працюють на механізмі компонування Blink.

До функціоналу VS Code належать такі функції, як заміна кодової сторінки при зберіганні документа, заміна мови програмування поточного документа і встановлення зовнішніх розширень.

Для роботи облікового функціоналу сервісу, такого як запис кількості витрачених або отриманих грошей або вибір категорії фінансової транзакції є необхідність створення взаємопов'язаної бази даних.

База даних (БД) представляє собою структуроване утворення, створене для зберігання, зміни та обробки взаємозв'язаної інформації, зокрема великих обсягів. Широко застосовується використання баз даних для динамічних веб-сайтів з великою кількістю даних, таких як інтернет-магазини, портали та корпоративні сайти. Зазвичай ці сайти розробляються з використанням серверних мов програмування (наприклад, PHP) або на основі систем управління вмістом (CMS), наприклад, WordPress, і не мають статичних сторінок, подібних до HTML-сайтів. Сторінки динамічних веб-сайтів формуються "на льоту" шляхом взаємодії скриптів і баз даних після відповідного запиту клієнта до веб-сервера.

Дослідження у сфері розробки баз даних, зокрема їх розробці на базі мови SQL описується в наукових роботах таких дослідників, як М. В. Добролюбова, М. В. Філіппова, О. М. Маркіна, та Alan Beaulieu.[11-12]

Для правильного функціонування бази даних вона має дозволяти виконувати такі завдання:

- записувати нових користувачів після реєстрації;
- записувати фінансові дані;
- створювати нові користувацькі категорії;
- виводити звіт про загальний фінансовий стан.

Опираючись на завдання бази даних створено список даних, які вона має зберігати, на основі якого створюється структура самої бази:

- таблиця users - список користувачів;

- таблиці categories_costs, categories_finance – категорії відповідно витрат та прибутку за замовчуванням;
 - таблиці new_categories, new_categories_finance – нові категорії, які створює для себе користувач;
 - таблиця payment_type – вид грошової транзакції: готівковий або безготівковий;
 - таблиця finance – звітність про фінансовий стан кожного користувача.
- На рис. 2.1. зображена структура бази даних «finance»

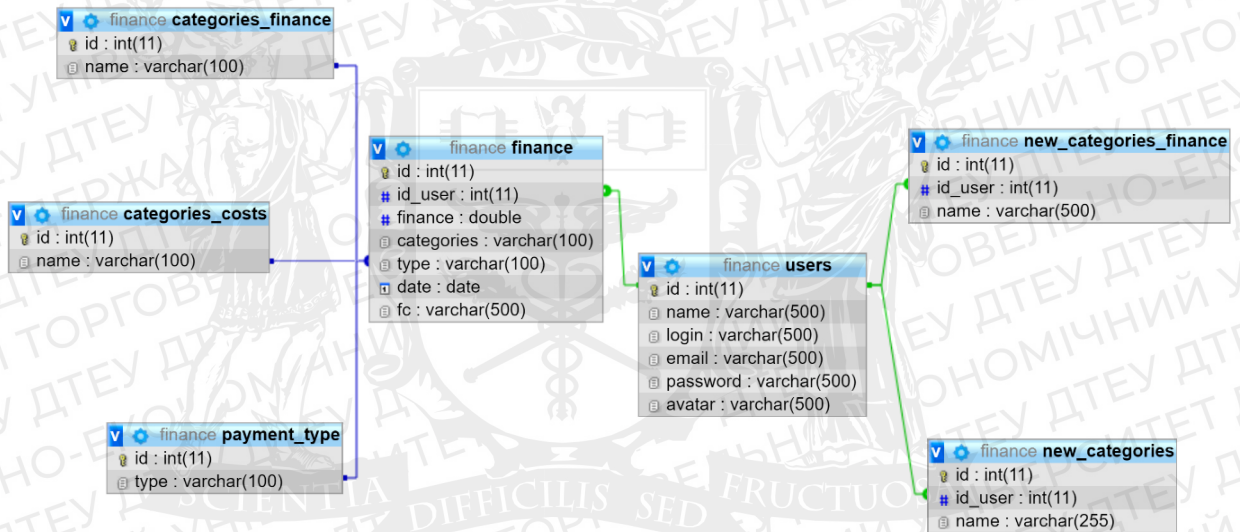


Рис. 2.2. База Даних

Головною таблицею бази даних, з якої буде формуватися підсумковий звіт про фінансовий стан користувача, є таблиця finance. На рис. 2.2 зображена структура таблиці

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
2	id_user	int(11)			Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
3	finance	int(11)			Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
4	categories	varchar(100)	utf8_general_ci		Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
5	type	varchar(100)	utf8_general_ci		Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
6	date	date			Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]
7	fc	varchar(500)	utf8_general_ci		Нет	Нет			[Edit] [Delete] [Add] [Refresh] [Undo] [Redo] [Zoom]

Рис. 2.3. Структура таблиці finance

- id – номер запису;
- id_user – номер користувача;
- finance – кількість витрат на категорії;
- categories – категорії;
- type – тип оплати ;
- date – дата ;
- fc – дохід/витрати.

Невід’ємною частиною будь-якої бази даних є захист інформації, тому заради протидії можливому злому бази особисті паролі користувачів мають бути зашифровані. Приклад зашифрованих даних зображено на рис. 3.3



id	name	login	email	password
1	yt	yt	yt@gmail.com	e10adc3949ba59abbe56e057f20f883e
2	Наталия	natap	natap2004@ukr.net	29e4b71616763eb28bfd6b1c8c72a134
3	Danya	Kek	Danya@gmail.com	09b4b7bf9c276cd9636cd679f595f744
4	Данило Чеховський	Данило	Daniel@gmail.com	d8578edf8458ce06fbc5bb76a58c5ca4

Рис. 3.3. Шифрування користувацьких паролів

База даних створювалась у середовищі PhpMyAdmin, а наповнюється за допомогою спеціальних команд, які приводить у дію скрипт написаний мовою PHP.

2.5. Висновки до розділу

В даному розділі було вирішено питання засобів виконання розробки, в результаті чого обрано середовище розробки програмного продукту, і зроблено наступні висновки:

- PhpMyAdmin є незамінним інструментом для розробників та адміністраторів баз даних, які працюють з MySQL.
- Visual Studio Code є потужним та гнучким інструментом розробки, який надає розробникам зручність, продуктивність та широкі можливості для розвитку їх проєктів.

Було розроблено структуру даних проекту. На даному етапі проект має чітку сформовану структуру.



РОЗДІЛ 3 Практична реалізація застосунку обліку власних фінансів

3.1. Програмна реалізація застосунку та його особливості

Теорія та принципи розробки верстки та дизайну веб-застосунків детально описується у роботі зарубіжного автора Jon Duckett[13], а дійсним методам впровадження мови програмування PHP в веб-застосунки присвячені книги авторів Josh Lockhart і Matt Zandstra [14-15].

Для веб-застосунку обліку власних фінансів «Finance.com» потрібно було виконати перелік таких дій:

1. реалізувати верстку – розробити весь зовнішній інтерфейс застосунку;
2. створити базу даних – сховище та джерело усіх користувацьких даних;
3. підключити базу даних до сайту – створити зв'язок між сервером і сайтом;
4. реалізувати функціональну частину – можливість створювати та передивлятися записи.

Для того щоб підключити базу даних до сайту потрібно в файлі `connected.php` заповнити змінні даними. Зміст файлу на рисунку 3.1.

avatar	Папка с файлами	
css	Папка с файлами	
js	Папка с файлами	
php	Папка с файлами	
add.php	Исходный файл PHP	5 КБ
authorization.php	Исходный файл PHP	3 КБ
categories.php	Исходный файл PHP	6 КБ
diagram.php	Исходный файл PHP	17 КБ
edit_profile.php	Исходный файл PHP	4 КБ
index.php	Исходный файл PHP	2 КБ
profile.php	Исходный файл PHP	5 КБ
registration.php	Исходный файл PHP	3 КБ

Рис. 3.1. Структура папки сайту

Файл `connected.php` підключається майже до всіх файлів з розширенням `.php`, без нього не можливо виконати операції, котрі виконуються с базою даних. Основні функціональні команди прописані на PHP. Інтерфейс сайту реалізований за допомогою HTML та CSS.

Структура сайту:

- `finance` — в даній паці знаходяться основні фаїли для роботи сайту та інші папки;
- `finance/avatar` — в даній паці знаходяться аватари користувачів;
- `finance/css` — в даній паці знаходяться усі стилі, які використовуються на сайті;
- `finance/js` — в даній паці знаходяться скрипти, які використовуються на сайті;
- `finance/php` — в даній паці знаходяться усі файли, котрі використовють функції `php`.

У програмному кодї використовуються такі функції MySQL:

- `mysql_connect` – головна функція для підключення до бази даних;
- `mysql_select_db` – функція вибору бази даних;
- `mysql_query` – функція відправки унікального запиту до бази даних;
- `mysql_fetch_assoc` – функція, що повертає рядок у вигляді асоціативного масиву;
- `mysql_fetch_array` - функція, що повертає рядок у вигляді числового масиву;
- `mysql_num_rows` – функція, що повертає кількість рядків.

Призначення створення веб-застосунку обліку власних фінансів – спростити ведення обліку доходів і витрат. Для відображення веб-застосунку на сторінці у браузері використовується локальний сервер Open Server.

На рисунку 3.2 зображена діаграма навігації по сайту.



Рис. 3.2 Навігація по сторінках сайту

3.2. Інструкція щодо використання

Для того, щоб відвідати сайт користувач повинен зареєструватися або авторизуватися, якщо у нього вже є обліковий запис. При відвідуванні сайту, авторизований користувач потрапляє на сторінку особистого кабінету, де в нього є можливість переглянути особисті дані, редагувати їх та побачити статистику доходів та витрат за весь час.

Вигляд форми початкової сторінки сайту зображений на рисунку 3.3

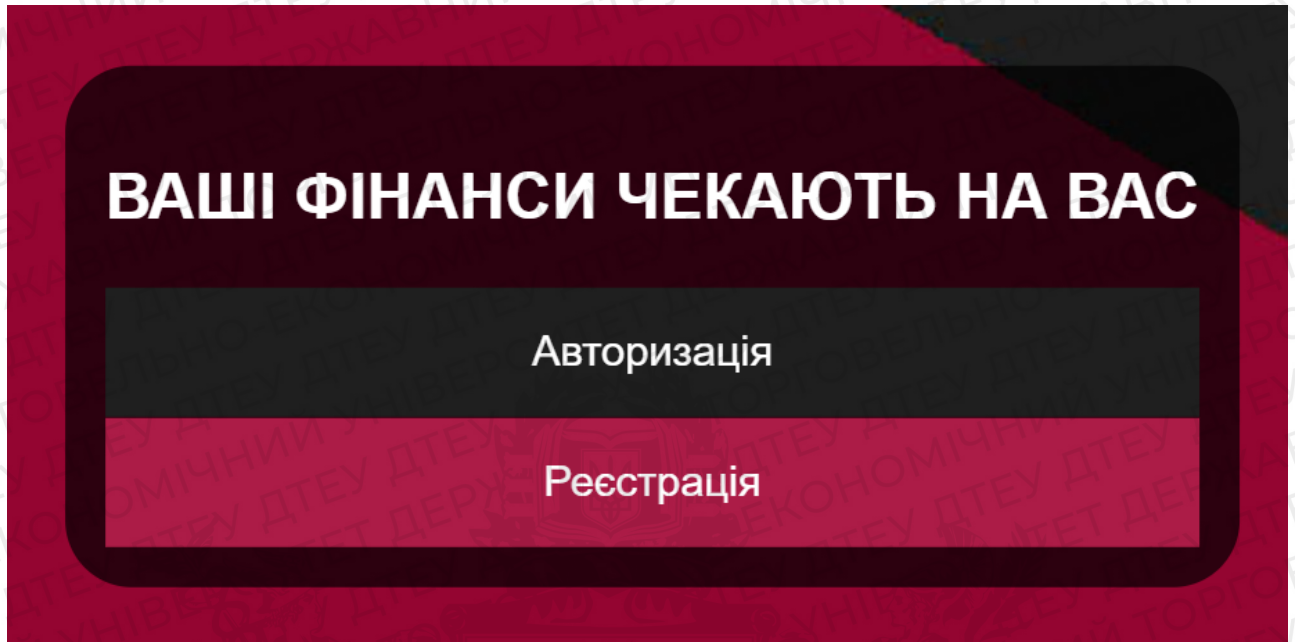


Рис. 3.3. Форма початкової сторінки сайту

При натисканні на кнопку «Реєстрація», користувач потрапляє на форму реєстрації, де потрібно заповнити такі поля:

- ім'я;
- логін;
- електронна пошта;
- пароль.

Вигляд форми реєстрації зображений на рисунку 3.4.

РЕЄСТРАЦІЯ

ІМ'Я

Введіть своє ім'я

ЛОГІН

Введіть свій логін

АДРЕС ЕЛЕКТРОНОЇ ПОШТИ

Введіть адрес електронної пошти

ПАРОЛЬ

Введіть пароль

ПІДТВЕРДЖЕННЯ ПАРОЛЮ

Підтвердіть пароль

Зареєструватися

У вас вже є акаунт? - [Авторизуватися](#)

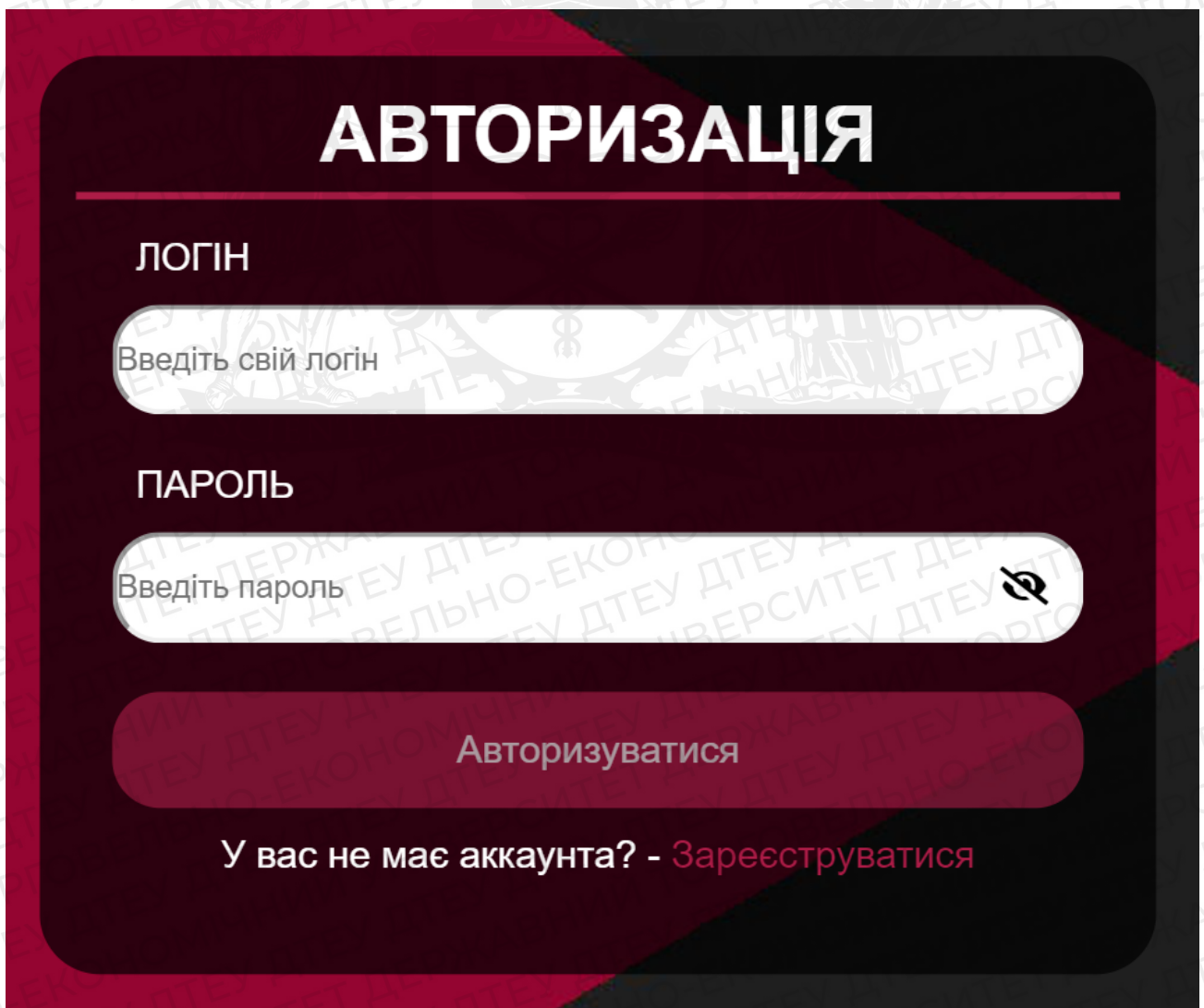
Рис. 3.4. Форма реєстрації

При коректному введенні даних буде створено обліковий запис в базі даних і користувача буде перенаправлено на початкову сторінку, де вже зареєстрований користувач зможе авторизуватися.

При натисканні на кнопку «Авторизація», користувача потрапляє на форму авторизації. Ця форма складається з таких полів:

- логін або електронна пошта;
- пароль.

Вигляд форми авторизації зображений на рисунку 3.5



The image shows a login form with a dark background and red accents. At the top, the word 'АВТОРИЗАЦІЯ' is written in large, white, bold letters. Below it, there are two input fields: one for the login (labeled 'ЛОГІН') and one for the password (labeled 'ПАРОЛЬ'). The password field has a small icon of a key and a lock. Below the input fields is a red button labeled 'Авторизуватися'. At the bottom, there is a link that says 'У вас не має аккаунта? - Зареєструватися'.

Рис. 3.5. Форма авторизації

При правильному введенні даних користувача буде перенаправлено на його особистий кабінет. Сторінка профілю відображена на рисунку 3.6

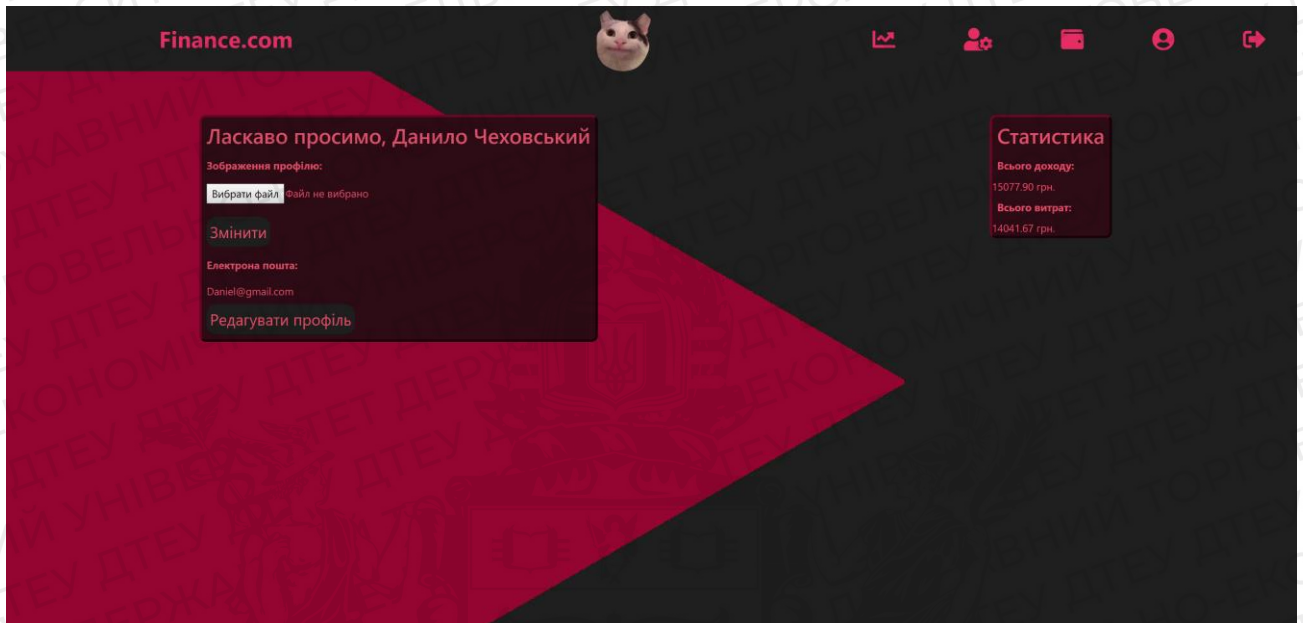


Рис. 3.6. Сторінка особистого кабінету користувача

Особистий кабінет зображує загальну інформацію про користувача, таку як його облікове ім'я, електронна пошта та сумарна статистика його доходів та витрат.

Реєстрація на сайті є обов'язковою. Якщо у користувача не має облікового запису, користувач не має можливості відвідати сайт. На початковій сторінці сайту, користувачеві надається можливість зареєструватися або авторизуватися, якщо вже є обліковий запис.

На рисунку 3.7 зображена сторінка профілю з точки зору програмного коду. Програма бере особисті дані з бази даних за допомогою SQL-запитів та зображує їх у вікні веб-сторінки.

```

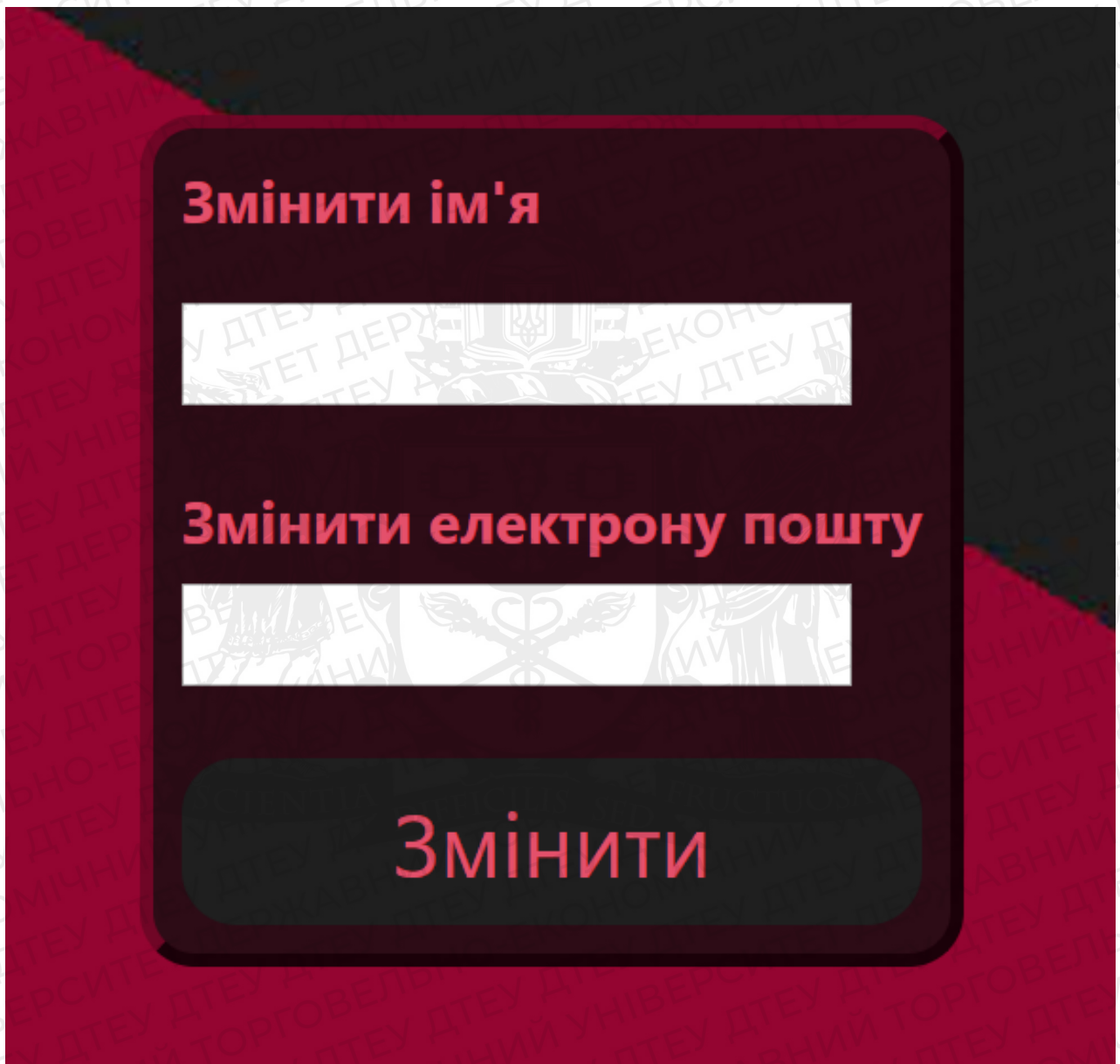
<div class='info'>
<div class="info-stat">
<div class="info-profile">
<h2 style="margin: 10px 0;"> Ласкаво просимо, <?
$user=$_SESSION['user']['id'];
$sql="SELECT name FROM users WHERE id=$user";
$result=mysql_query($sql);
while ($row = mysql_fetch_array($result))
{
    echo $row['name'];
}
?></h2>
<label><b>Зображення профілю:</b></label><br>
<form action="php/avatar.php" method="POST" enctype='multipart/form-data'>
<input type="file" name="new_avatar"><br>
<button type="submit" class='button' name="change_avatar">Змінити</button>
</form>
<label><b>Електронна пошта:</b></label>
<p><?
$user=$_SESSION['user']['id'];
$sql="SELECT email FROM users WHERE id=$user";
$result=mysql_query($sql);
while ($row = mysql_fetch_array($result))
{
    echo $row['email'];
}
?></p>
<div class="button-profile" style="margin-bottom:3px;">
<form action="edit_profile.php" method="POST">
<button class="button">Редагувати профіль</button>
</form>
</div>
</div>
<div class="stat">
<h2>Статистика</h2>
<?php
$user=$_SESSION['user']['id'];
$sql1="SELECT ROUND(SUM(`finance`),2)finance FROM finance WHERE fc='Дохід' and id_us
$result1= mysql_query($sql1);
while ($row1 = mysql_fetch_array($result1))
{
    echo "<b>Всього доходу:</b>".$row1['finance']." грн.<br>";
}
$sql2="SELECT ROUND(SUM(`finance`),2)finance FROM finance WHERE fc='Витрати' and id_us
$result2= mysql_query($sql2);
while ($row2 = mysql_fetch_array($result2))
{
    echo "<b>Всього витрат:</b>".$row2['finance']." грн.";
}
?>
</div>
</div>

```

Рис. 3.7. Код сторінки профілю

Якщо користувач бажає змінити записане в базі даних власне ім'я та прив'язану електронну пошту, він може це зробити, натиснувши на кнопку

«Редагувати профіль» та перейшовши на форму зміни. Форма зміни особистої інформації зображена на рис. 3.8.



The image shows a dark-themed user interface for editing personal information. It features three main sections, each with a red heading and a white input field:

- Змінити ім'я**: The first section has a red heading and a white rectangular input field.
- Змінити електрону пошту**: The second section has a red heading and a white rectangular input field.
- Змінити**: The third section is a large, rounded rectangular button with a red heading.

Рис. 3.8. Форма зміни особистої інформації

З особистого кабінету у користувача є можливість перейти на одну із сторінок за допомогою меню. На шапці сайту знаходиться назва сайту, аватар користувача та меню. Шапку сайту можна побачити на рисунку 3.9.

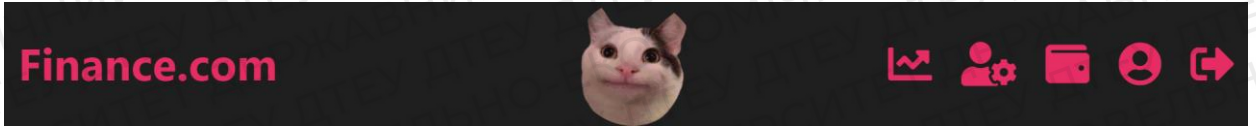


Рис. 3.9. Шапка сайту

Використовуючи меню, користувач має можливість переміщатися та використовувати різноманітні функції сайту. Меню має 6 кнопок, котрі перенаправляють користувача на різноманітні сторінки. Меню можна побачити на малюнку 3.10.

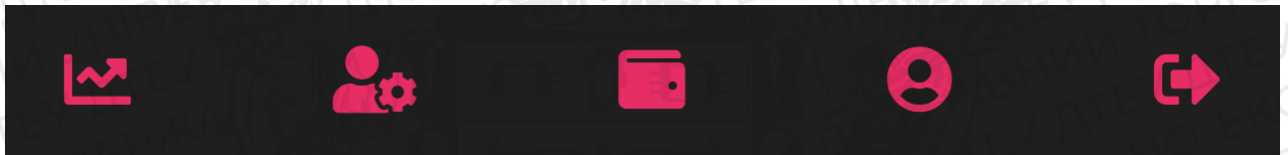


Рис. 3.10. Меню сайту

При натисканні на кнопку «Внесення записів», користувач потрапляю на сторінку, де може заповнити дані про дохід та витрати за певний період часу.

Усі вище перераховані функції можна побачити на рисунках 3.11-3.13.

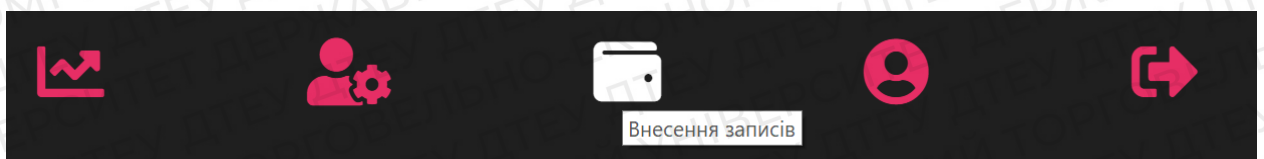


Рис. 3.11. Кнопка переходу на сторінку внесення записів

Витрати **Дохід**

Тип оплати

Безготівковий ▾

Категорії

Відпочинок ▾

Витрати

Дата

Зберегти

Рис. 3.12. Внесення витрат

Витрати **Дохід**

Готівка/Безготівкова

Безготівковий ▾

Тип заробітку

Зарплата ▾

Дохід

Дата

Зберегти

Рис. 3.13. Внесення доходів

При натисканні на кнопку «Створення нових категорій», користувач потрапляє на сторінку, де має можливість створити нові категорії для доходів та витрат. Якщо користувач зробив помилку або бажає видалити категорію, є можливість скористуватися списком, що випадає, де будуть відображені усі особисті створені категорії. Вибравши одну із створених категорій та натиснувши кнопку «Видалити» обрана категорія буде видалена. Подивитися на всі ці функції можна на рисунках 3.14-3.17.

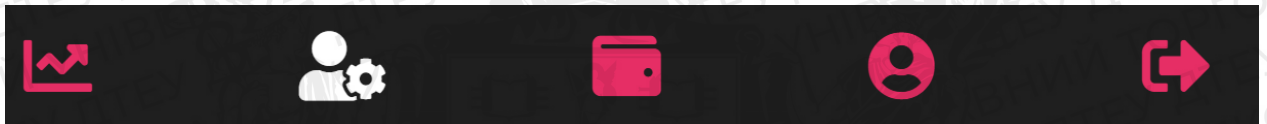


Рис. 3.14. Кнопка переходу на сторінку створення нових категорій

Витрати **Дохід** **Видалення категорій**

Створити нову категорію

Введіть назву категорії

Додати

Рис. 3.15. Створення нової категорії для витрат

The screenshot shows a web interface with three tabs: 'Витрати' (Expenses), 'Дохід' (Income), and 'Видалення категорій' (Delete Categories). The 'Дохід' tab is selected. Below the tabs, the text 'Створити нову категорію' (Create new category) is displayed. A text input field contains the placeholder 'Введіть назву категорії' (Enter category name). At the bottom, there is a large black button with the text 'Додати' (Add) in red.

Рис. 3.16. Створення нової категорії для доходів

The screenshot shows the same web interface as Figure 3.16, but with the 'Видалення категорій' (Delete Categories) tab selected. The text 'Видалити категорію' (Delete category) is displayed. Below it is a dropdown menu with the text 'Оберіть категорію, котру бажаєте видалити' (Select the category you want to delete) and a downward arrow. At the bottom, there is a large black button with the text 'Видалити' (Delete) in red.

Рис. 3.17. Видалення створених категорій

При натисканні на кнопку «Статистика», користувач потрапляю на сторінку, де має можливість переглянути статистику у вигляді діаграми та таблиці. Таблицю можна не тільки переглянути, а також видалити запис або роздрукувати, якщо користувачеві це потрібно. Усе це можна побачити на рисунках 3.18-3.21.



Рис. 3.18. Кнопка переходу на сторінку статистики

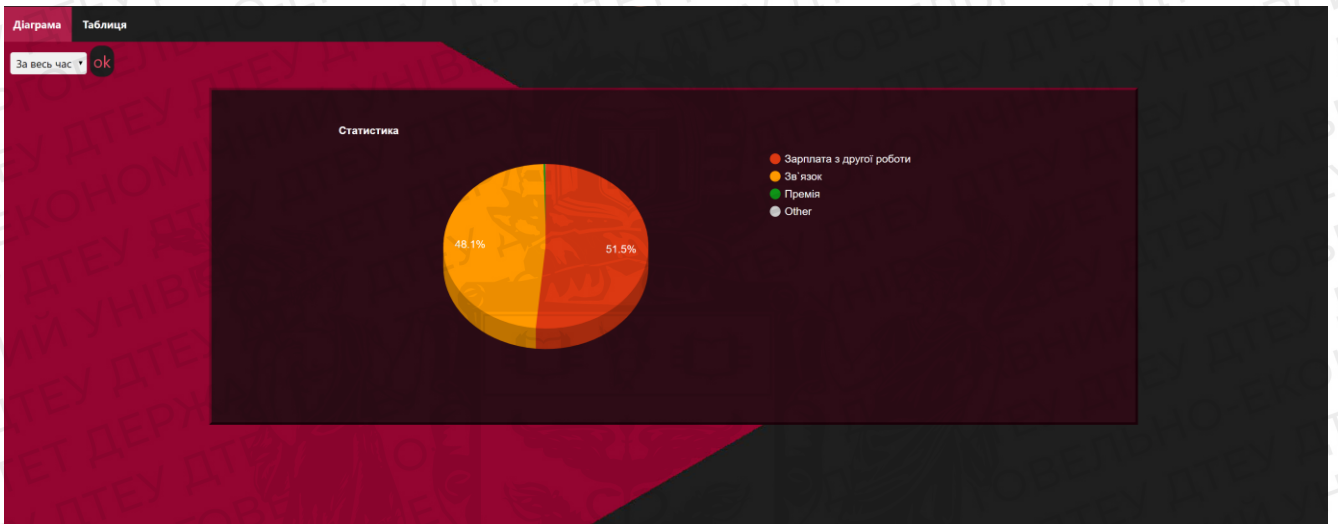


Рис. 3.19. Статистика в діаграмі


Категорії	Дата	Тип оплати	Докід грн.	Витрати грн.	X
Житло	2021-01-01	Безготівковий		1.67	x
Харчування	2023-05-19	Безготівковий		40	x
Зв'язок	2021-01-01	Готівковий		14000	x
Премія	2021-01-01	Безготівковий	77.9		x
Зарплата з другої роботи	2021-01-01	Безготівковий	15000		x
Всього			15077.90	14041.67	
Безготівковий			15036.23		
Залишок			1036.23		

Рис. 3.20. Статистика в таблиці

Друк

Усього: 1 аркуш паперу

Друк Скасувати

Місце призначення  Microsoft Print to PDF

Змінити...

Сторінки Все

напр., 1–5, 8, 11–13

Орієнтація Альбомна

Колір Колір

+ Більше налаштувань

Друкувати за допомогою діалогового вікна системи... (Ctrl+Shift+P)

21.05.2023

Категорії	Дата	Тип оплати	Дохід грн.	Витрати грн.
Житло	2021-01-01	Безготівковий		1.67
Харчування	2023-05-19	Безготівковий		40
Зв'язок	2021-01-01	Готівковий		14000
Премія	2021-01-01	Безготівковий	77.9	
Зарплата з другої роботи	2021-01-01	Безготівковий	15000	
Всього			15077.90	14041.67
Безготівковий			15036.23	
Залишок				1036.23

Рис. 3.21. Друк таблиці

На рисунку 3.22. зображено частину коду, необхідного для формування таблиці всіх фінансових записів користувача. Програма отримує інформацію з бази даних, після чого динамічно створює таблицю та записує дані в клітинки.

```

<table border="1">
  <thead>
    <tr>
      <th>Категорії</th>
      <th>Дата</th>
      <th>Тип оплати</th>
      <th>Дохід грн.</th>
      <th>Витрати грн.</th>
      <th><i class='fas fa-times'></i></th>
    </tr>
  </thead>
  <tbody>
    <?php
      $month=$_POST['month'];
      $user=$_SESSION['user']['id'];
      if($month==0)
      {
        $sql="SELECT id,categories,type,finance,date FROM finance WHERE fc='Витрати' and id_user=$user";
        $result = mysql_query($sql);
        while ($row = mysql_fetch_array($result)) {
          echo "<tr><td>".$row['categories'].</td><td>".$row['date'].</td><td>".$row['type'].</td><td></td><td>".$row['finance']
        }

        $sql="SELECT id,categories,type,finance,date FROM finance WHERE fc='Дохід' and id_user=$user";
        $result = mysql_query($sql);
        while ($row1 = mysql_fetch_array($result)) {
          echo "<tr><td>".$row1['categories'].</td><td>".$row1['date'].</td><td>".$row1['type'].</td><td></td><td>".$row1['finance']
        }
        /*Всього*/
        $sql3="SELECT ROUND(SUM(`finance`),2)finance FROM finance WHERE fc='Дохід' and id_user=$user GROUP BY fc='Витрати'";
        $result3= mysql_query($sql3);

        while ($row3 = mysql_fetch_array($result3)){

          $sql2="SELECT ROUND(SUM(`finance`),2)finance FROM finance WHERE fc='Витрати' and id_user=$user GROUP BY fc='Витрати'";
          $result2= mysql_query($sql2);

          while ($row2 = mysql_fetch_array($result2)){
            echo "<tr><td><b>Всього</b></td><td colspan='2'></td><td><b>".$row3['finance'].</b></td><td><b>".$row2['finance']
          }
        }
      }
    </?php
  </tbody>
</table>

```

Рис. 3.22. Код таблиці записів

На рисунку 3.23 зображено код для створення кругової діаграми записів. Програма виконує SQL-запити для отримання даних, після чого виконує скрипт для створення візуалізації отриманої інформації.

```

<!--Діаграма1-->
<div id="Diagram" class="tabcontent">
<form method='POST' enctype='multipart/form-data'>
<select name="month">
<option value="0">За весь час</option>
<option value="1">Січень</option>
<option value="2">Лютий</option>
<option value="3">Березень</option>
<option value="4">Квітень</option>
<option value="5">Травень</option>
<option value="6">Червень</option>
<option value="7">Липень</option>
<option value="8">Серпень</option>
<option value="9">Вересень</option>
<option value="10">Жовтень</option>
<option value="11">Листопад</option>
<option value="12">Грудень</option>
</select>
<button class='button'>ок</button>
</form>
<script type="text/javascript">
google.charts.load("current", {packages:["corechart"]});
google.charts.setOnLoadCallback(drawChart);
function drawChart()
{
var data = google.visualization.arrayToDataTable([
['categories', 'cost'],
<?php
$month=$_POST['month'];
$user=$_SESSION['user']['id'];
if($month==0)
{
$sql = "SELECT date, SUM(`finance`)finance, categories FROM finance WHERE id_user=$user GROUP BY categories ";
$result = mysql_query($sql);

while ($a=mysql_fetch_assoc($result)){
echo["'".$a['categories'].","."$a['finance'].",";
}
}
else
{
$sql = "SELECT date, SUM(`finance`)finance, categories FROM finance WHERE id_user=$user and MONTH(date) = '$month' GROUP BY ca";
$result = mysql_query($sql);

while ($a=mysql_fetch_assoc($result)){
echo["'".$a['categories'].","."$a['finance'].",";
}
}
?>
]);

var options = {
title: 'Статистика',
backgroundColor: 'transparent',
legendTextStyle: { color: '#FFF' },
titleTextStyle: { color: '#FFF' },
is3D: true,
};

var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));

chart.draw(data, options);
}
</script>
<div id="piechart_3d" style="margin-left:15%;margin-right:15%;width: 70%; height: 500px;background-color: #2d0c16;border:3.5px solid

```

Рис. 3.23. Код для створення кругової діаграми

Після створення застосунку наступним етапом розробки є його тестування.

Тестування – це перевірка сайту, що спрямована на виявлення відхилень, тобто процес пошуку помилок. Під час розробки було проведено тестування верстки і тестування кросбраузерності.

Верстка – це розміщення елементів веб-застосунку відповідно до макету та вимог. Спочатку проводиться перевірка правильності розташування елементів та їх відповідність макетам. Після цього проводиться перевірка коду на валідність – це відповідність вихідного коду правилам та нормам.

Кросбраузерність - це коректна, адаптивна верстка для правильного відображення сайту на різних пристроях (комп'ютер, планшет, телефон) і браузерах (Chrome, Firefox, і т.д.). Реалізується вона, за допомогою інструментів для верстка сторінок сайту, таких як HTML і CSS.

Цей етап перевірки є досить важливим, адже досить часто з'являються нові версії різноманітних браузерів, до яких необхідно пристосовувати застосунок, щоб кожен користувач зміг ознайомитись з сайтом.

Вигляд сайту в різноманітних браузерах можна побачити на рисунках 3.24-3.26.

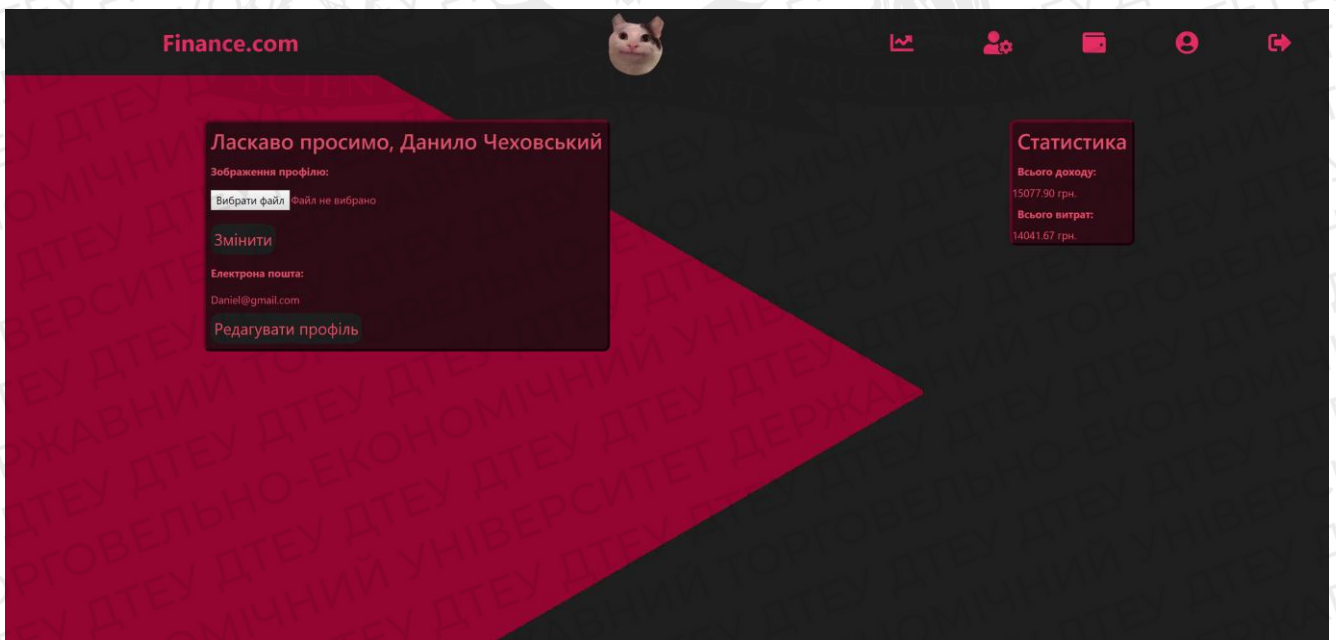


Рис. 3.24. Вигляд особистого кабінету в браузері Google Chrome

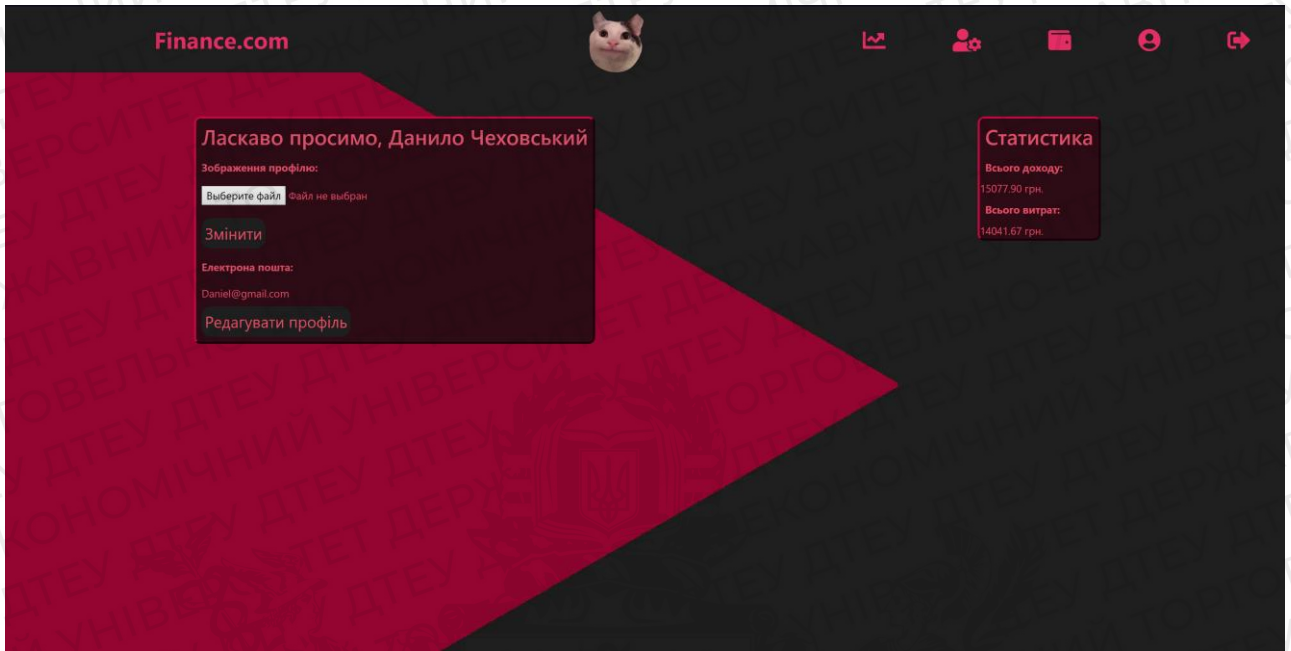


Рис. 3.25. Вигляд особистого кабінету в браузері Орега

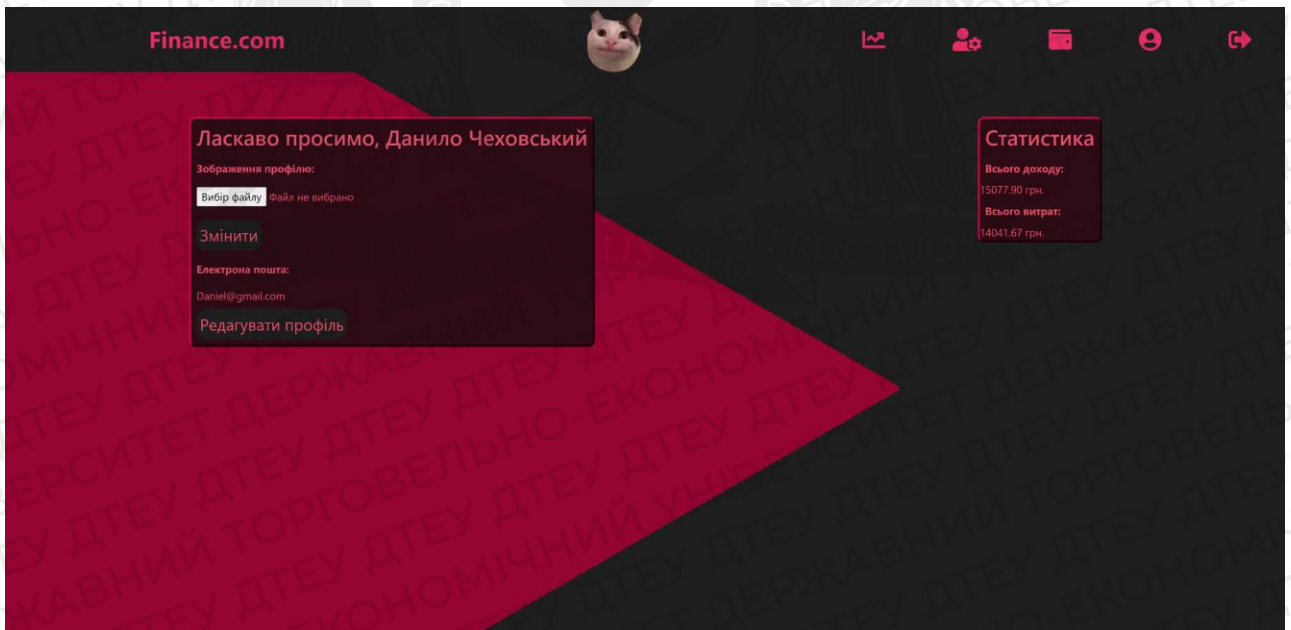


Рис. 3.26. Вигляд особистого кабінету в браузері Microsoft Edge

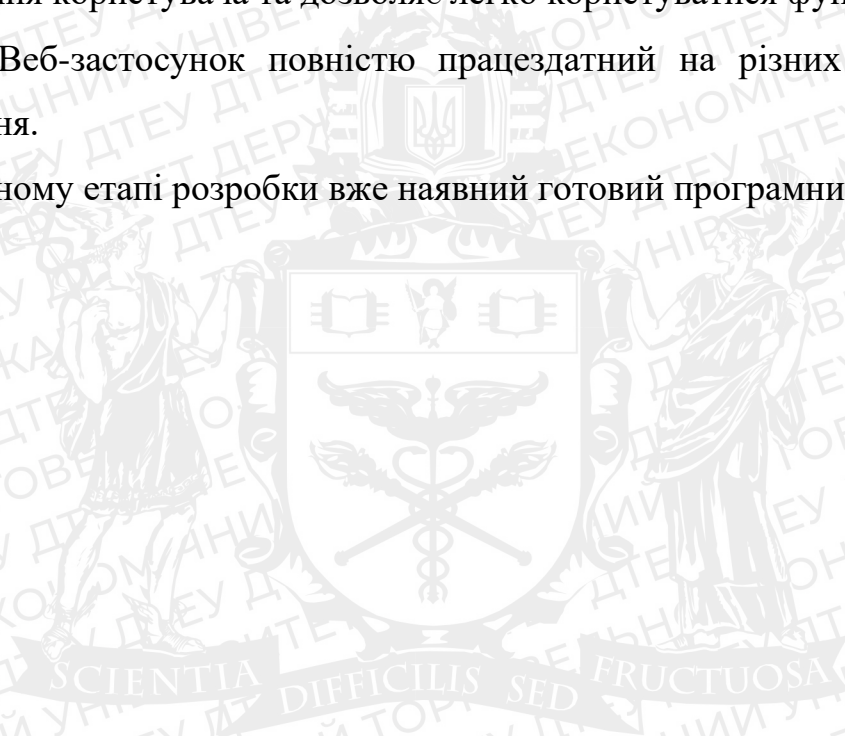
3.3. Висновки до розділу

В даному розділі було описано керівництво користувача та програміста. Розглянуто інтерфейс вебсайту. Проведено тестування верстки сайту. Визначено основне призначення та умови для запуску даного вебзастосунку.

Після проведення розгляду створеного веб-застосунку було зроблено такі висновки:

- Веб-застосунок виконує свою головну функцію: дозволяє створювати та ознайомлюватися з записами про власний фінансовий стан;
- Інтерфейс застосунку є простим та інтуїтивно-зрозумілим, що зменшує роздратування користувача та дозволяє легко користуватися функціями сервісу;
- Веб-застосунок повністю працездатний на різних платформах для використання.

На даному етапі розробки вже наявний готовий програмний продукт.



РЕЗУЛЬТАТИ І ВИСНОВКИ

Головною метою випускної кваліфікаційної роботи було створення такого застосунку, який дозволив би користувачу вести записи про власні витрати та прибуток, що створювало б чітку картину фінансового стану користувача.

У результаті розробки проекту було проведено виконано поставлені завдання, в результаті чого отримано наступні результати:

1. На основі виконаного дослідження сфери обліку фінансів було розроблено вимоги до розробки проекту;
2. Було проаналізовано та обрано найкращі засоби розробки для реалізації поставленої задачі;
3. Розроблений веб-застосунок повністю функціональний та відповідає поставленим вимогам;
4. Для супроводу проекту було розроблено відповідну документацію.

На основі отриманих результатів можна зробити висновки:

1. Впровадження сайту обліку фінансів виявилось важливим та корисним для організації/бізнесу. Він надає зручний спосіб зберігання, організації та аналізу фінансової інформації.
2. Сайт обліку фінансів сприяє поліпшенню процесів планування та контролю фінансової діяльності. Він дозволяє зручно створювати бюджети, відстежувати доходи та витрати, а також генерувати звіти та аналітичні дані.
3. Сайт обліку фінансів забезпечує доступність та зручність для користувачів. Він може бути доступним з будь-якого пристрою з Інтернет-підключенням, що дозволяє зручно виконувати фінансові операції та отримувати потрібну інформацію в режимі реального часу.
4. Використання бази даних для зберігання фінансової інформації дозволяє забезпечити безпеку та цілісність даних. Застосування мови програмування PHP та SQL дозволяє ефективно маніпулювати даними та забезпечувати потрібну функціональність сайту.

5. Результати впровадження сайту обліку фінансів показали його позитивний вплив на фінансовий контроль, планування та прийняття рішень. Він сприяє підвищенню ефективності фінансового управління, зменшенню ризиків та досягненню фінансової стабільності організації/бізнесу.

6. Рекомендації щодо подальшого вдосконалення сайту обліку фінансів включають розширення функціональності, покращення інтерфейсу користувача, врахування специфічних потреб організації/бізнесу та забезпечення регулярного оновлення та підтримки сайту.

Загалом, сайт обліку фінансів є важливим інструментом для забезпечення ефективного фінансового управління та досягнення фінансової стабільності. Впровадження такого сайту має позитивний вплив на організацію/бізнес та вимагає подальшого вдосконалення та підтримки для досягнення найкращих результатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Атамас П. Й., Атамас О. П. Фінансовий облік. Навчальний посібник / П. Й. Атамас, О. П. Атамас., 2020. – 356 с.
2. Іщенко Я.П., Подолянчук О.А., Коваль Н.І. Фінансовий облік 1: підручник / Я.П. Іщенко, О.А. Подолянчук, Н.І. Коваль. Вінниця: Видавництво ФОП Кушнір Ю. В. 2020. – 496 с.
3. ЗАКОН УКРАЇНИ Про бухгалтерський облік та фінансову звітність в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/996-14#Text>.
4. Програма для бізнесу RemOnline [Електронний ресурс] – Режим доступу до ресурсу: <https://remonline.ua/>.
5. Сервіс для ведення бюджету myBudget [Електронний ресурс] – Режим доступу до ресурсу: <https://mybudget.ws>.
6. Сервіс ведення обліку фінансів Finmap [Електронний ресурс] – Режим доступу до ресурсу: <https://finmap.online>.
7. Офіційний сайт PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net>.
8. Що таке PHP? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/intro-what-is.php>.
9. Офіційний сайт Open Server Panel [Електронний ресурс] – Режим доступу до ресурсу: <https://ospanel.io>.
10. Офіційний сайт Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com>.
11. Добролюбова, М. В. Програмування баз даних. Практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Інформаційні вимірювальні технології» спеціальності 152 «Метрологія та інформаційно-вимірювальна техніка» / М. В. Добролюбова, М. В. Філіппова, О. М. Маркіна ; КПІ ім. Ігоря Сікорського.– Київ : КПІ ім. Ігоря

Сікорського, 2021. – 164 с. – Режим доступу до ресурсу:

<https://ela.kpi.ua/bitstream/123456789/42414/1/PDB%20%28CP%29.pdf>.

12. Beaulieu A. Learning SQL: Generate, Manipulate, and Retrieve Data / Alan Beaulieu., 2020. – 377 с.
13. Duckett J. HTML and CSS: Design and Build Websites / Jon Duckett., 2011. – 490 с.
14. Lockhart J. Modern PHP / Josh Lockhart., 2015. – 371 с.
15. Zandstra M. Modern PHP / Matt Zandstra., 2016. – 603 с.

