

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Інформаційна система управління валютними операціями
Назва теми

КВРІСТ. 101039.21.01.02 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконав: студент III курсу, група ІСТс-21-1

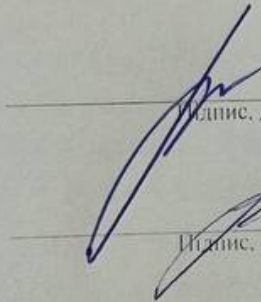


Підпис

В. В. Зозуля

Ініціали, прізвище

Керівник




Підпис, дата

Є. Г. Гнатчук

Ініціали, прізвище

Нормоконтролер



Підпис, дата

І.О. Засорнова

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри комп'ютерної
інженерії та інформаційних систем



Підпис

Т.О. Говоруценко

Ініціали, прізвище

« 6 » червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма ОСВІТНЯ ПРОГРАМА «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорущенко

“ 11 ” 01 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Зозулі Владиславу Валерійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління валютними операціями

Керівник проекту (роботи) Гнатчук Є. Г., к.т.н, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 05.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)





Дослідження предметної області та постановка задачі

Проектування інформаційної системи управління валютними операціями

Програмна реалізація інформаційної системи управління валютними операціями

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.С., доцент кафедри ІСТ		
Антиплагіат	Ніченорук А.О., доцент кафедри ІСТ		

7. Дата видачі завдання « 11 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – проєктування інформаційної системи управління валютними операціями	01.04.2024	виконано
5	Робота над розділом 3 – програмна реалізація інформаційної системи управління валютними операціями	30.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
8	Захист ВКР на засіданні ЕК	червень 2024 року	

Студент


Підпис

В.В. Зозуля
Ініціали, прізвище


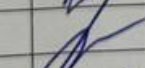


Керівник проекту (роботи)


Підпис

Є. Г. Гнатчук
Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРІСТ 101039.21.01.02 ПЗ	Пояснювальна записка	64		
			<u>Графічні матеріали</u>			
2		КвРІСТ 101039.21.01.02 Е8	Схема бази даних	1		
3		КвРІСТ 101039.21.01.02 Е8	Діаграма послідовності дій	1		
4		КвРІСТ 101039.21.01.02 Е8	Діаграма розгортання	1		

КвРІСТ 101039.21.01.02 ВП

Зм	Арк	№ докум	Підпис	Дата	Інформаційна система управління валютними операціями. Відомість проекту	Літера	Аркуш	Аркуші
Розробив		Зозуля		06.06		У	1	1
Перевір.		Гнатчук		06.06	ХНУ, ІСТс-21-1			
Н. контр.		Засорнова		06.06				
Затв.		Говорущенко		06.06				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління валютними операціями».

Автор роботи: Зозуля Владислав Валерійович .

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 64 с., 55 рис., 1 табл., , 49 джерела.

Графічна частина: 3 креслення.

ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, АВТОМАТИЗАЦІЯ,
ВЕБДОДАТОК, ВАЛЮТА, ОБМІН ВАЛЮТИ.

Метою роботи є надання користувачам зручного інструменту для ефективного управління валютними операціями, шляхом розроблення інформаційної системи управління валютними операціями.

Об'єктом дослідження є принципи побудови інформаційних систем управління валютними операціями.

Предметом дослідження є вебдодаток для управління валютними операціями.

Практичне значення має спроектована та реалізована інформаційна система управління валютними операціями у вигляді зручного для користувача вебдодатку.


Підпис студента

4.06.2024
Дата

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП	5
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз вхідних та вихідних даних інформаційної системи управління валютними операціями	6
1.2 Огляд існуючих рішень, методів та засобів розробки інформаційних систем управління валютними операціями	9
1.3 Формалізована модель інформаційної системи управління валютними операціями.....	16
1.4 Висновки. Постановка задачі	17
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВАЛЮТНИМИ ОПЕРАЦІЯМИ	19
2.1 Проєктування архітектури інформаційної системи управління валютними операціями.....	19
2.2 Проєктування бази даних інформаційної системи управління валютними операціями.....	26
2.3 Проєктування інтерфейсу інформаційної системи управління валютними операціями.....	31
2.4 Висновки	36
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВАЛЮТНИМИ ОПЕРАЦІЯМИ	37
3.1 Програмна реалізація інформаційної системи управління валютними операціями.....	37
3.2 Тестування інформаційної системи управління валютними операціями .	58
3.3 Інструкція з інсталяції та експлуатації інформаційної системи управління валютними операціями	61
3.4 Висновки	67

КВРІСТ 190362.21.01.09 ПЗ

№	Арк.	№ док.ум.	Підпис	Дата	Літера	Арк.вш	Арк.вшіє
ІКОНАВ		Зозуля В. В.		06.06			
РЕДІВІР.		Гнатюк С. Г.		06.06			
КОНТР.		Засорнова І. С.		06.06			
ТВЕР.		Говорунченко Т. О.		06.06			

Інформаційна система
управління валютними
операціями.
Пояснювальна записка.

ХНУ ІСТс-21-1

ВИСНОВКИ	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	69
ДОДАТОК А	74
Копія креслення «Схема бази даних».....	74
ДОДАТОК Б	75
Копія креслення «Діаграма послідовності дій».....	75
ДОДАТОК В	76
Копія креслення «Діаграма розгортання».....	76

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

Handwritten mark

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

АІС – автоматизована інформаційна система

БД – база даних

СУБД – система управління базами даних

НФ – нормальна форма

ІТ – інформаційні технології

UML – Unified Modelling Language або універсальна мова моделювання

SQL – Structured Query Language або структурована мова запитів

HTML – мова розмітки гіпертексту (HyperText Markup Language)

CSS – спеціальна мова стилю сторінок (Cascading Style Sheets)

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Валютний ринок є одним із найбільших і найдинамічніших у світовій економіці, де щодня відбуваються мільярди транзакцій. Фінансові установи, такі як банки, інвестиційні компанії та брокерські контори, беруть активну участь у цьому ринку шляхом здійснення валютних операцій від імені своїх клієнтів або для власних інвестиційних цілей. Ефективне управління цими операціями є критичним завданням для забезпечення прибутковості, мінімізації ризиків та дотримання нормативних вимог.

Традиційні методи управління валютними операціями, що базуються на паперових документах та розрізненних електронних таблицях, є недостатніми для сучасних вимог. Вони є трудомісткими, схильними до помилок та ускладнюють відстеження й аналіз операцій. Крім того, зростаючі регуляторні вимоги щодо прозорості та звітності вимагають більш надійного та централізованого підходу до управління даними.

Саме тому фінансові установи потребують комплексної інформаційної системи, яка б могла автоматизувати та оптимізувати процеси управління валютними операціями.

Створення комплексної інформаційної системи управління валютними операціями є важливим кроком для фінансових установ, що прагнуть підвищити ефективність своїх операцій, мінімізувати ризики та залишатися конкурентоспроможними на динамічному валютному ринку. Така система забезпечить централізоване управління валютними операціями, автоматизацію ключових процесів, покращену аналітику та відповідність нормативним вимогам.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз вхідних та вихідних даних інформаційної системи управління валютними операціями

Предметна область для розробки інформаційної системи управління валютними операціями є комплексною та багатогранною. Вона охоплює широкий спектр бізнес-процесів та функцій, пов'язаних із здійсненням валютних угод, управлінням валютними позиціями, хеджуванням валютних ризиків, а також забезпеченням відповідності нормативним вимогам.

Ключовим аспектом є торгівля валютами на міжбанківському ринку, що включає купівлю та продаж іноземних валют за різними інструментами, такими як спот, форвард та свопи. Система повинна дозволяти отримувати котирування валютних курсів, укладати та виконувати валютні угоди, як у готівковій, так і в безготівковій формі.

Предметна область охоплює також операції з платежами та розрахунками у різних валютах, включаючи здійснення міжнародних платежів, конвертацію валют, інтеграцію з банківською системою НБУ.

Система має вести повний облік усіх валютних операцій, угод та позицій, формувати звітність про валютні операції. Також вона має гарантувати безпеку та конфіденційність даних, контроль доступу користувачів, аудит їхніх дій та відстеження змін.

Виходячи з проведеного аналізу предметної області, слід визначити вхідні характеристики предметної області інформаційної системи управління валютними операціями.

Розглянемо основні вхідні об'єкти предметної області та їх характеристики.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Насамперед слід визначитись із, напевно, головним вхідним об'єктом предметної області – «Валюта». Його характеристиками є:

- назва валюти (гривня);
- код валюти символічний (UAH);
- графічний символ (₴);
- курс валюти відносно гривні (за даними НБУ);
- дата оновлення курсу валют;
- одиниця вимірювання для курсу валют.

Наступна сутність предметної області – «Користувачі» має такі характеристики:

- нікнейм користувача;
- електронна адреса користувача;
- пароль;
- аватар користувача;
- тип користувача (адміністратор, користувач).

Враховуючи те, що у кожного користувача може бути багато рахунків для здійснення валютних операцій, слід це врахувати за допомогою наступного об'єкта предметної області «Засоби оплати користувача»

- назва карткового рахунку;
- прізвище, ім'я, по батькові власника;
- телефон;
- адреса;
- додаткова інформація;
- валюта карткового рахунку;
- тип оплати (картка, віртуальна оплата).

Наступний важливий вхідний об'єкт предметної області – «Обмін валют». Його атрибути наступні:

- тип обміну (купівля, продаж, переказ);

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

- статус обміну (виконано, відхилено);
- користувач, який здійснює операцію;
- додаткова інформація про обмін;
- ознака подвійної конвертації;
- журнал операції;
- сума валюти, яку слід обмінити;
- її сума обміну;
- деталі обміну валюти, яку обмінюємо (інформація про рахунок, картку);
- сума валюти, яка отримана після обміну;
- її сума обміну;
- деталі обміну валюти, на яку обмінюємо (інформація про рахунок, картку).

Розглянувши основну вхідну інформацію, визначимо які ж будуть основні вихідні (результуючі) дані та способи їх формування.

Зрозуміло, що вхідна інформація, що описана вище має бути й вихідною інформацією. Вона буде виводитись, наприклад, у вигляді таблиці з курсом валюти чи звіт про виконану операцію обміну.

Також при видачі інформації відбувається її фільтрування, а саме користувач отримує лише ту інформацію, яка потрібна.

Для адміністративної частини буде можливість виведення всієї інформації, що міститься в інформаційній системі.

Частота виведення результуючої інформації не визначена. Виведення даних відбувається за вимогою.

Ці вхідні та результуючі характеристики допоможуть визначити обсяг та межі предметної області, а також забезпечити розробку ефективної інформаційної системи управління валютними операціями, яка задовольнить усі потреби користувачів.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Огляд існуючих рішень, методів та засобів розробки інформаційних систем управління валютними операціями

Перш ніж приступити до проектування та програмування додатку доцільно розглянути та проаналізувати існуючі аналоги, методи та засоби розробки.

Проведемо огляд аналогів та існуючих рішень для розробки інформаційної системи управління валютними операціями

На жаль, в Україні немає відомих широкомасштабних комплексних інформаційних систем управління валютними операціями, які б використовувалися великими фінансовими установами чи корпораціями. Однак є кілька локальних рішень, які пропонуються невеликими компаніями або були розроблені банками для внутрішнього використання:

Багато українських комерційних банків розробляли власні індивідуальні системи для автоматизації валютних операцій, але ці рішення, як правило, є закритими та використовуються тільки внутрішньо. Їх функціональність може варіюватись від простих систем торгівлі валютами до більш складних рішень з управління ризиками та звітністю.

Як приклад розглянемо систему «Валютні операції (ПУМБ)» [16].

Це внутрішня система українського ПУМБ, яка була розроблена для автоматизації валютних операцій банку. Вона включає можливості торгівлі валютами, управління позиціями, здійснення розрахунків та формування звітності. Однак система є закритою та використовується виключно в межах банку.

Для звичайних клієнтів доступний сайт <https://persona.pumb.ua/operatsii-s-valyutoy>, який рекомендує здійснювати керування валютними операціями за допомогою мобільних додатків або в касах банку (рисунок 1).

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

реалізуються як вебсторінки, розглянемо деякі такі додатки, що мають схоже призначення.

1. MetaTrader Web Terminal [17] (<https://metatraderweb.app/trade>).

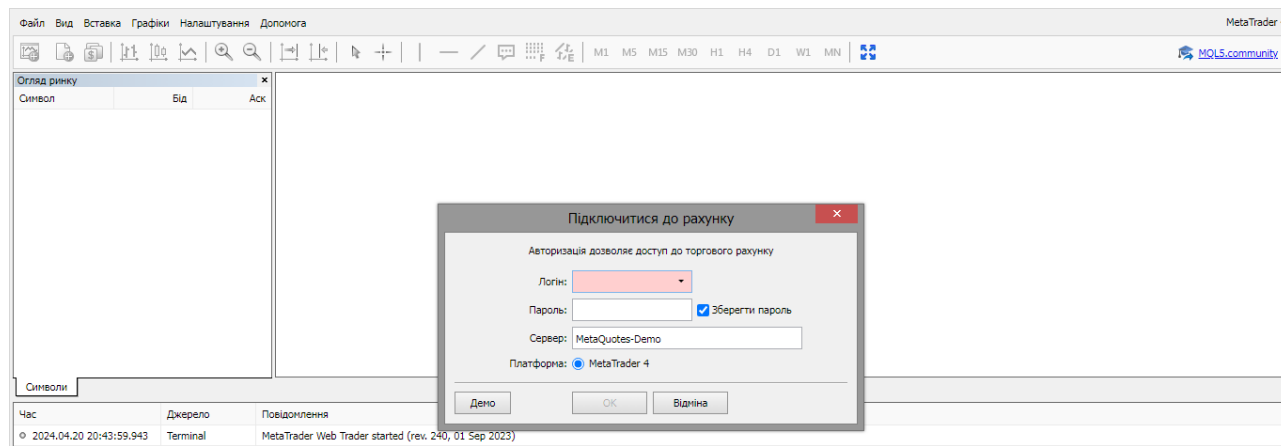


Рисунок 2 – Вебдодаток <https://metatraderweb.app/trade>

Головними можливостями додатку MetaTrader Web Terminal є: торгівля валютами, технічний аналіз, управління позиціями.

Проведемо його короткий аналіз:

- переваги: широко використовується, інтуїтивний інтерфейс, мобільна версія;
- недоліки: відсутність функцій управління ризиками та розрахунків.

2. TradingView [18] (<https://www.tradingview.com/>)

Здійснивши аналіз вебдодатку TradingView, можна виділити такі ключові особливості та характеристики:

1. Функціональність:

- торгівля валютами, акціями, криптовалютами, фьючерсами та іншими інструментами;
- потужні інструменти технічного аналізу (понад 100 індикаторів та інструментів малювання);

Зм.	Арк.	№ докум.	Підпис	Дата

- візуалізація даних за допомогою професійних графіків і діаграм;
- доступ до фундаментальних даних, новин та аналітики;
- можливість підключення зовнішніх скриптів та індикаторів.

2. Аналітичні можливості:

- величезна бібліотека готових індикаторів та стратегій;
- можливість створення власних скриптів та індикаторів за допомогою мови програмування Pine;
- обмін стратегіями з іншими користувачами через спільноту TradingView;
- відстеження ідей трейдерів та аналітиків у реальному часі.

3. Інтеграція та зв'язок:

- підключення до брокерських рахунків для торгівлі безпосередньо з платформи;
- підтримка популярних брокерів, таких як Interactive Brokers, TD Ameritrade тощо;
- можливість публікації аналітичних матеріалів та ідей на веб-сайті TradingView;
- відкритий API для інтеграції з іншими системами та додатками.

4. Спільнота та соціальні функції:

- величезна спільнота трейдерів та аналітиків з усього світу;
- можливість слідкувати за ідеями та публікаціями інших користувачів;
- система репутації та відгуків для оцінки якості контенту.

5. Зручність використання:

- інтуїтивний та зрозумілий інтерфейс;
- підтримка різних пристроїв (веб, настільні додатки, мобільні додатки);

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

- можливість налаштування робочого простору та персоналізації інтерфейсу.

6. Ціноутворення:

- безкоштовний базовий рівень доступу;
- преміум-підписки з додатковими функціями та можливостями;
- гнучкі тарифні плани залежно від потреб користувача.

Сильними сторонами TradingView є потужні аналітичні інструменти, величезна спільнота користувачів, підтримка інтеграції з брокерами та відкритий API. Однак, додаток зосереджений переважно на торгівлі та технічному аналізі, і може не задовольняти всі потреби управління валютними операціями в організаціях, такі як управління ризиками, платежі, звітність тощо.

Загалом, TradingView є чудовим рішенням для технічного аналізу ринків, торгівлі та обміну ідеями з іншими трейдерами, але може не підходити як комплексна система управління валютними операціями для великих фінансових установ або корпорацій.

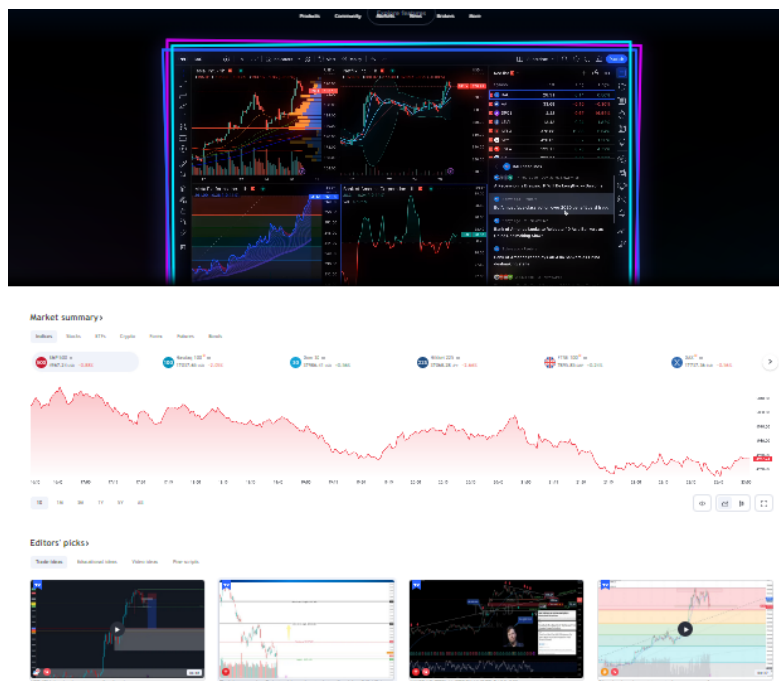


Рисунок 3 – Вебдодаток www.tradingview.com

Зм.	Арк.	№ докум.	Підпис	Дата

Під час аналізу було виявлено, що більшість існуючих веб-додатків зосереджені на торгівлі валютними парами та технічному аналізі. Вони пропонують функції для укладання угод, відстеження позицій та візуалізації даних, проте мають обмежені можливості для управління валютними ризиками, здійснення розрахунків, формування звітності та інтеграції з іншими системами.

Крім того, більшість цих веб-додатків розроблені для окремих трейдерів та індивідуальних інвесторів, і можуть не відповідати вимогам великих фінансових установ чи корпорацій, яким потрібні більш комплексні рішення.

Важливим недоліком більшості аналізованих веб-додатків є відсутність або обмежені можливості управління валютними ризиками, такі як хеджування, моделювання сценаріїв ризиків та розрахунок показників ризику.

Загалом, аналіз показує, що на даний момент відсутні повноцінні веб-додатки, які б задовольняли усі потреби комплексного управління валютними операціями для великих організацій. Це може бути можливістю для розробки спеціалізованого українського веб-рішення управління валютними операціями, що скероване на індивідуальне використання.

Огляд та аналіз додатків, що розглянуті вище вказує на необхідність проведення додатково аналізу та розгляду сучасних методів та засобів для реалізації поставленого завдання.

Розглянемо методи розробки вебдодатку для управління валютними операціями.

1. Вибір технологічного стеку:

- для розробки веб-інтерфейсу можна використати сучасні JavaScript-фреймворки, такі як React, Angular або Vue.js. Вони забезпечують високу продуктивність, гнучкість та модульність, що важливо для складних вебдодатків;

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

- для серверної частини доцільно використати Node.js, який дозволяє застосовувати JavaScript [29], [30] як на клієнтській, так і на серверній стороні, спрощуючи розробку та підтримку;
- для роботи з базами даних можна обрати MongoDB (документоорієнтована NoSQL) або PostgreSQL (реляційна), в залежності від специфіки зберігання даних.

2. Архітектура та структура додатку:

- для забезпечення масштабованості, гнучкості та модульності системи можна застосувати архітектурний шаблон Micro-Services;
- розділення додатку на окремі мікросервіси дозволить розробляти, тестувати та розгортати їх незалежно;
- для комунікації між мікросервісами можна використовувати REST API або event-driven архітектуру на основі брокера повідомлень (RabbitMQ, Apache Kafka);
- застосування парадигми "Програмування реактивних систем" (Reactive Programming) дозволить додатку ефективно реагувати на зміни даних та оновлювати інтерфейс в режимі реального часу.

3. Розробка клієнтської частини:

- для створення інтуїтивно зрозумілого та адаптивного веб-інтерфейсу можна використати компонентно-орієнтований підхід, властивий сучасним фреймворкам;
- розробка мобільно-орієнтованого дизайну з використанням адаптивної верстки, медіазапитів та гнучких компонентів є важливою вимогою, адже буде зручно користуватимуться додатком за допомогою мобільних пристроїв.

4. Розробка серверної частини:

- реалізація REST API для взаємодії клієнтського додатку з серверними сервісами;

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

- створення сховищ даних (MongoDB, PostgreSQL, MySQL) для зберігання інформації про користувачів, валюти, обміни.

Альтернативним технологічним стеком для розробки вебдодатку є використання PHP-фреймворків [14], таких як Yii, Laravel або Symfony. Такий стек більш простий, вивчений інструментально для мене, водночас є досить потужним для реалізації поставленого завдання. Тому зосередимось саме на PHP-фреймворку Laravel [1] виходячи із наступних міркувань:

Особливості PHP-фреймворку Laravel:

- Laravel – популярний PHP-фреймворк, який надає всі необхідні інструменти для швидкої та ефективної розробки веб-додатків [2];
- для клієнтської частини можна використати класичний стек технологій HTML, CSS, JavaScript [12], [27], [28];
- для роботи з базами даних Laravel [3] пропонує вбудовану підтримку таких СУБД, як MySQL, PostgreSQL, SQLite та SQL Server. Використаємо саме MySQL, керуючись принципом, простота/якість;
- використання парадигми "Модель-Представлення-Контролер" (MVC)[19], властивої Laravel, для розділення логіки додатку на окремі шари.

Використовуючи потужний фреймворк Laravel у поєднанні з описаними методами розробки, можна створити високоякісний, масштабований та функціональний вебдодаток для управління валютними операціями.

1.3 Формалізована модель інформаційної системи управління валютними операціями

Спроекуємо та розглянемо формалізовану модель вебдодатку, інформаційної системи управління валютними операціями. Для цього спроекуємо концептуальну модель.

Концептуальна модель – це змістовний опис модельованої системи на неформальній мові. Неформалізований опис розроблюваної імітаційної моделі включає визначення основних елементів модельованої системи, їх характеристики і взаємодія між елементами власною мовою. При цьому можуть використовуватися таблиці, графіки, діаграми і т.д. Такий опис моделі потрібний як розробникам, так і фахівцями інших профілів.

Концептуальна модель містить вихідну інформацію для системного аналітика, що виконує формалізацію системи і використовує для цього певну методологію та технологію. В подальшому формалізований опис перетвориться в програму - імітатор відповідно до деякої технології програмування.

Отже, для створення концептуальної моделі використаємо уніфіковану мову моделювання UML [15]. UML – це мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення. Ця мова була створена для візуалізації, проектування та документування, в основному, програмних систем. Вона не відноситься до мов програмування, але на базі моделей є можливість генерування програмного коду.

1.4 Висновки. Постановка задачі

Грунтуючись на проведеному аналізі предметної області та визначенні вхідної/вихідної інформації, огляді існуючих аналогів, засобів та методів розробки, а також створеної формальної моделі задачі можемо сформулювати постановку задачі для розробки вебдодатку, інформаційної системи управління валютними операціями.

Отже, в ході виконання дипломного проєкту слід розробити вебдодаток, що має на меті допомогу з управління валютними операціями.

При цьому слід реалізувати наступний функціонал: CRUD про користувачів, карткові та інші рахунки цих користувачів, курси валют, транзакції з купівлі чи продажу валют, переказу коштів, стан виконання цих транзакцій.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

Тобто необхідно розробити комплексний вебдодаток, інформаційну систему, який надаватиме користувачам зручні інструменти для ефективного управління валютними операціями. Основними функціональними вимогами до системи є:

- показувати актуальні курси валют. Додати мінімум вісім валют;
- реалізувати можливість вибору валюти для купівлі та валюти для продажу та інтерактивної конвертації валют за актуальним курсом;
- ведення статистики кількості операцій по обміну валюти за день/місяць;
- ведення статистики скільки і якої валюти було продано за день/місяць та куплено за день/місяць;
- ведення журналу історії операцій з обміну валюти, що має містити число, час, валюта для купівлі, валюта для продажу;
- реалізувати можливість здійснення переказів коштів з можливою конвертацією валют, як готівкою на різні рахунки/карти, так і можливість вибрати безготівковий варіант (тобто форма карти: номер, строк дії, CVV). При цьому передбачити можливість обміну валюти за крос курсом. Реалізувати форму для вводу реквізитів карти або банківського рахунку, на яку буде здійснено переказ.

Ці аспекти охоплюють основні вимоги до інформаційної системи управління валютними операціями та визначають ключові функціональні можливості, які повинні бути реалізовані під час її розробки.

Аналітичний огляд існуючих методів та засобів дозволяє здійснити обґрунтований вибір технологій для реалізації завдання. Отже, для реалізації вебдодатку, інформаційної системи управління валютними операціями слід використати платформу Laravel, мову програмування PHP, СУБД MySQL [32].

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВАЛЮТНИМИ ОПЕРАЦІЯМИ

2.1 Проєктування архітектури інформаційної системи управління валютними операціями

Виходячи з того, що обраний для реалізації поставленого завдання фреймворк Laravel передбачає використання архітектурного шаблону MVC[20], тому спроектуємо архітектурну модель вебдодатку, інформаційної системи управління валютними операціями саме на основі такого патерну (рисунок 4).

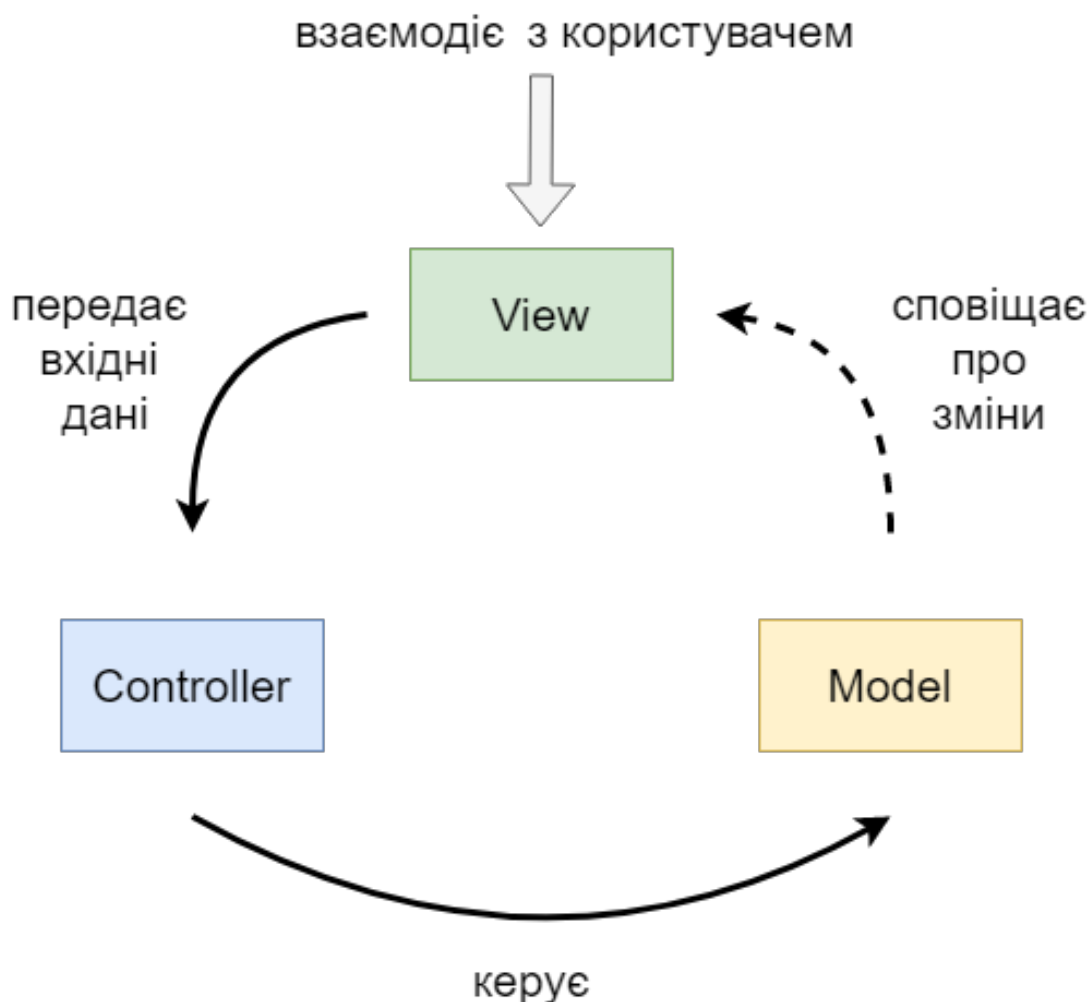


Рисунок 4 – Діаграма взаємодії між компонентами шаблону MVC

Для цього використаємо проведений у попередньому розділі аналіз предметної області, огляд існуючих методів та засобів, формалізовані моделі задачі.

Спершу слід спроектувати модель даних (Data Model).

Модель у патерні MVC відповідає за базу даних та основну логіку програми для роботи з нею, обробки запитів, проведення обчислень тощо;

На цьому етапі виконаємо загальне проектування моделі даних на основі аналізу предметної області, проведеного раніше.

Отже, головними об'єктами моделі даних вебдодатку велосипедний туризм будуть наступні:

- користувачі;
- рахунки користувачів;
- типи рахунків;
- валюта;
- обмін валют;
- тип операції обміну;
- стан операції обміну.

Детальне проектування моделі даних буде виконано у наступному розділі 2.2 «Проектування бази даних інформаційної системи управління валютними операціями».

Далі спроектуємо основні модулі додатку:

1. UsersModule:
 - управління профілями користувачів, інформацією про їх рахунки;
 - аутентифікація, авторизація.
2. CurrencyModule:
 - пошук та оновлення даних про валюти, їх актуальний курс.
3. TransfersModule:

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

- управління валютними операціями: купівлею, продажем, переказами тощо;
- визначення типу та статусу валютних операцій.

Наступний блок патерну, який слід спроектувати – це View. View відповідає за інтерфейс користувача, усі візуальні елементи, з якими буде взаємодіяти користувач;

Основними представленнями (Views) додатку є:

1. UserView:

- реєстрація, авторизація користувачів, їх права, рахунки.

2. CurrencyView:

- перегляд даних про валюти, їх актуальний курс.

3. TransfersView:

- перегляд, додавання та управління валютними операціями: купівлею, продажем, переказами. Перегляд типу та статусу валютних операцій.

Далі перейдемо до проектування контролерів. Контролери у патерні MVC – це прошарок між View і Model, що реагує на зміни.

Основними контролерами (Controllers) додатку є

1. UsersController:

- реєстрація, авторизація, оновлення профілю, у тому числі рахунків користувача.

2. CurrencyController:

- пошук, перегляд, збереження валют та їх актуальних курсів на сайті НБУ;
- додавання, оновлення, видалення валют та їх актуальних курсів.

3. TransfersController:

- пошук, перегляд, додавання валютних операцій;
- управління інформацією про валютні операції, їх тип та статус.

Як підсумок проектування архітектури вебдодатку, інформаційної системи управління валютними операціями покажемо його основні спроектовані структурно-функціональні блоки, використовуючи UML діаграму компонентів (рисунок 5).

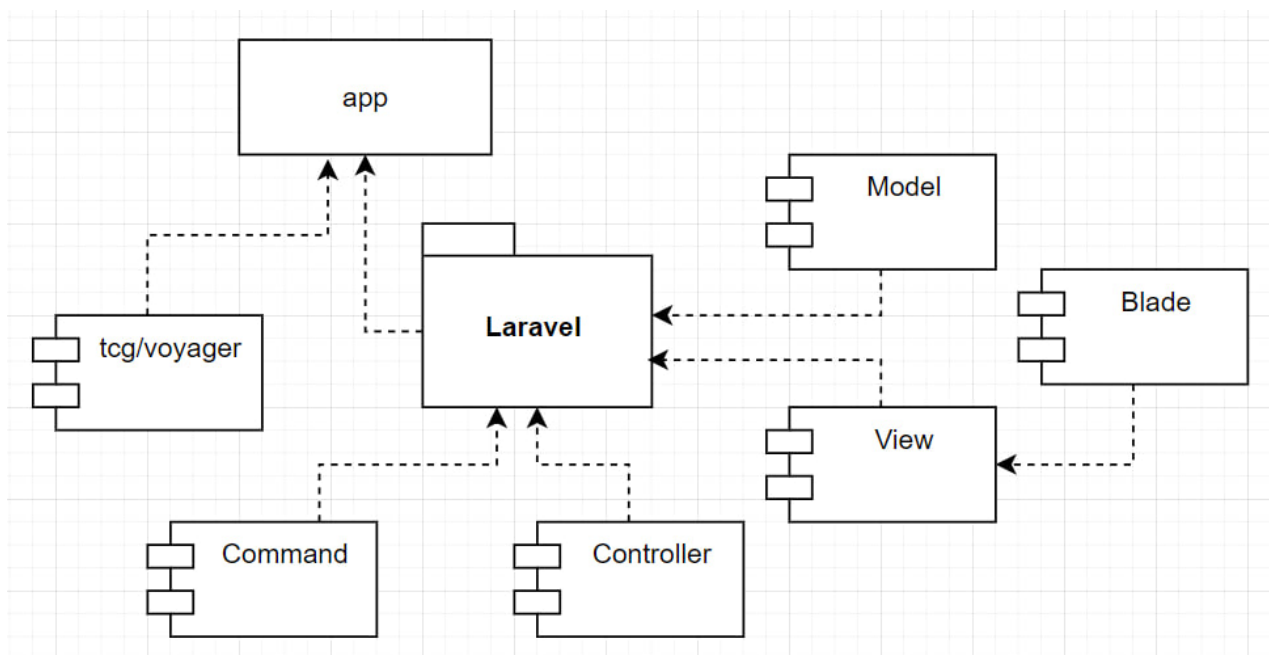


Рисунок 5 – Діаграма компонентів інформаційної системи управління валютними операціями

Приведена на рисунку 8 архітектурна модель вебдодатку включає в себе основні компоненти системи. Вона забезпечує структурований підхід до розробки функціонального та масштабованого вебдодатку, інформаційної системи управління валютними операціями.

Для визначення основних варіантів використання системи, що розробляється, використаємо діаграму прецедентів (Use Case).

Отже, в програмній системі буде визначено два типи користувачів: *звичайний користувач* (рисунок 6) – матиме права після реєстрації та авторизації на перегляд інформації про:

- курси валют;
- власні засоби оплати. Також можна додати новий засіб оплати;
- виконані транзакції, а саме купівлі/продажу валюти, переказ коштів.

При роботі з такими операціями доцільно користуватись засобами фільтрації.

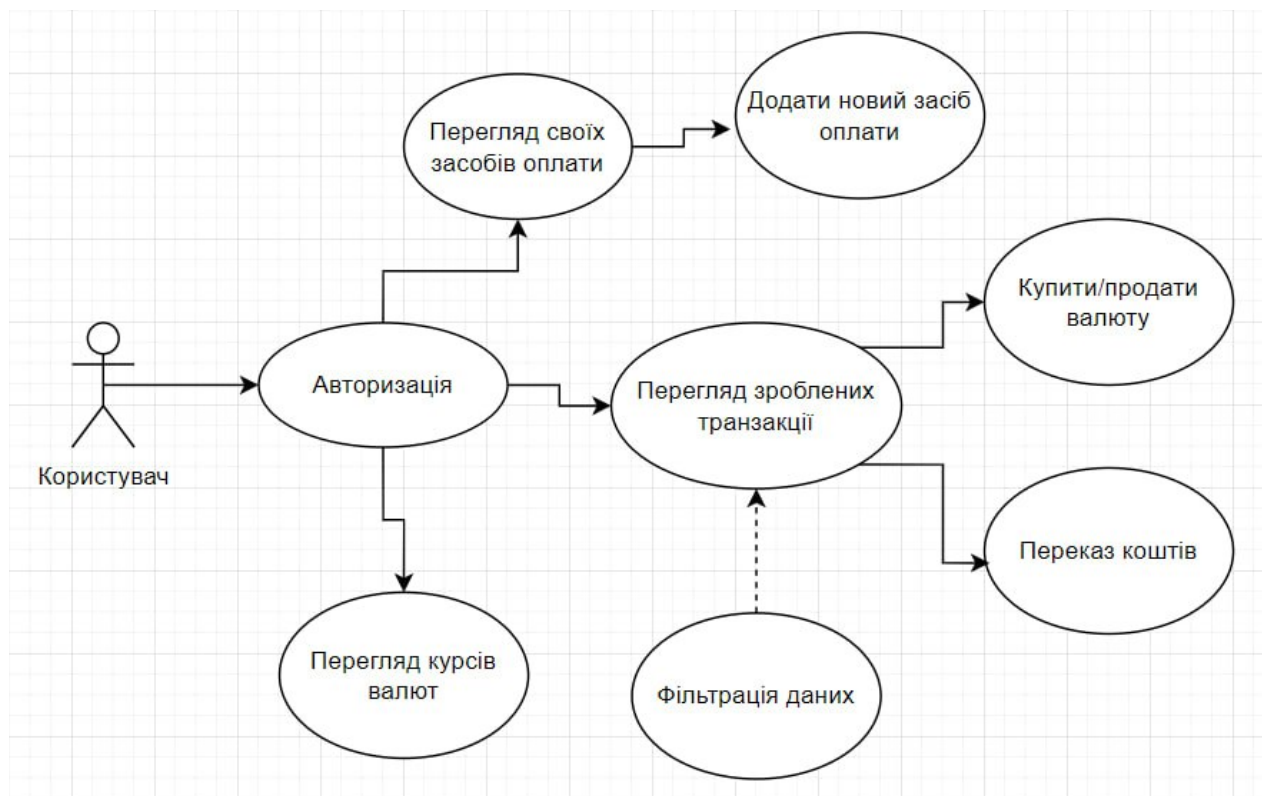


Рисунок 6 – Діаграма прецедентів для користувача

адміністратор (рисунок 7) – матиме наступні можливості після реєстрації та авторизації:

- перегляд курсів валют;
- облік користувачів;
- перегляд статистичних даних;
- конвертація валюти;

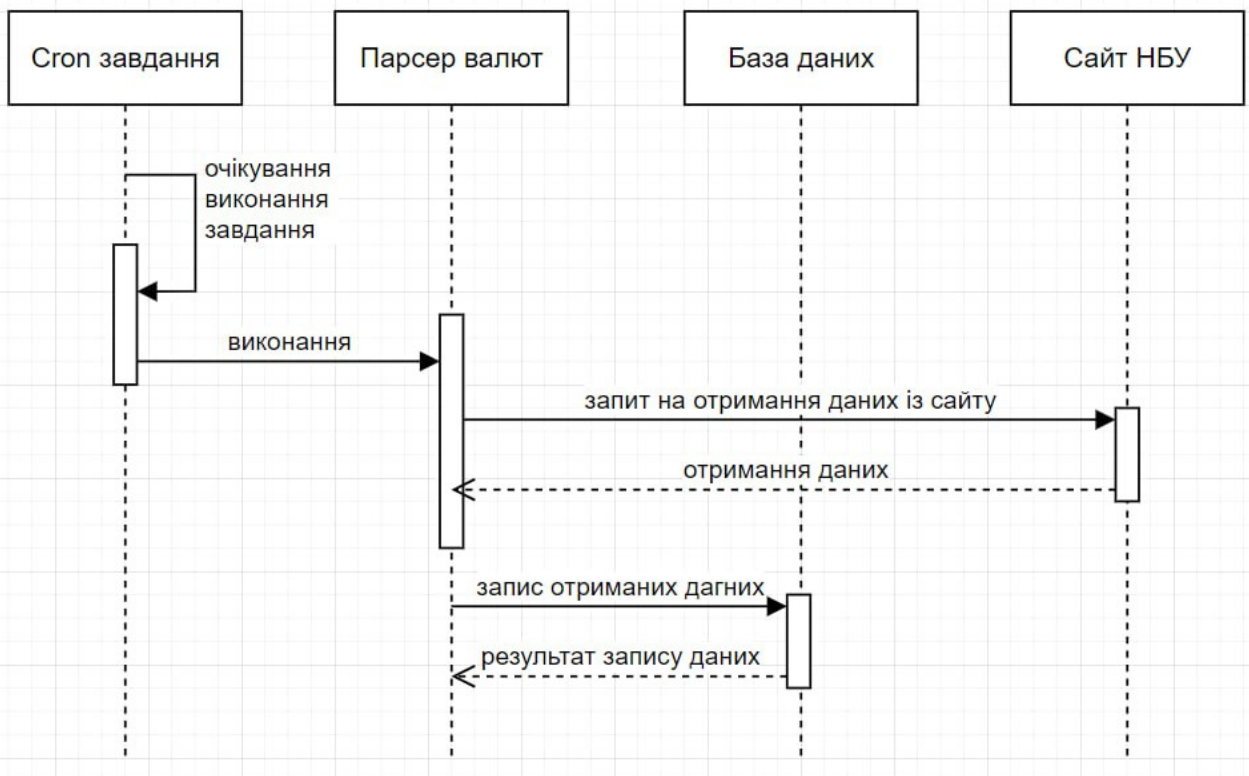


Рисунок 8 – Діаграма послідовності дій інформаційної системи управління валютними операціями

Як видно з рисунку 8 послідовні роботи користувача з інформаційною системою наступна:

- система виконує так зване стоп завдання, тобто завдання, що періодично повторюється (наприклад, купівля/продаж валюти);
- після очікування на виконання стоп завдання, система виконує парсер валют із сайту НБУ [22]. Звідти парсяться, або ж отримуються дані про поточні курси валют, крос курси тощо. Після отримання таких даних, вони записуються до бази даних додатку.

2.2 Проектування бази даних інформаційної системи управління валютними операціями

Детальне проектування моделі даних, як частини архітектури MVC, передбачає проектування бази даних, що буде основною структурою, що забезпечуватиме збереження та опрацювання інформації.

Детальне проектування бази даних полягає у процесі нормалізації на основі загальної моделі даних, що описана у попередньому розділі [31].

Для приведення спроектованих об'єктів/таблиць моделі даних до першої нормальної форми слід визначити усі їх характеристики/атрибути. Далі усунути дублювання полів та перевірити чи усі поля є атомарними (неподільними).

Для приведення до другої нормальної форми слід до таблиць додати унікальні ідентифікатори – первинні ключі, від яких мають безпосередньо залежати усі інші поля таблиці.

На цьому ж етапі проведемо формалізацію зв'язків між таблицями.

Опишемо процес приведення до першої та другої нормальної форми нижче.

Нижче приведемо структуру таблиць бази даних. Призначення основних полів таблиць було описане у розділі 1.1 «Аналіз вхідних та вихідних даних інформаційної системи управління валютними операціями». Спершу спроектуємо простіші таблиці-довідники.

Структура таблиці «statuses» (статуси операцій обміну) приведена на рисунку 9.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

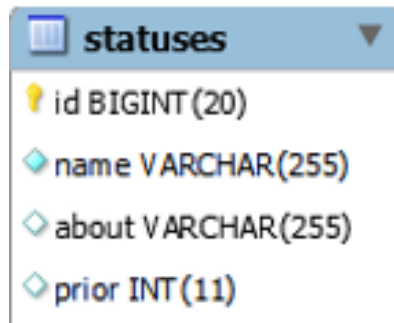


Рисунок 9 – Таблиця «statuses»

Структура таблиці «transfer_types» (типи обмінних операцій) приведена на рисунку 10.

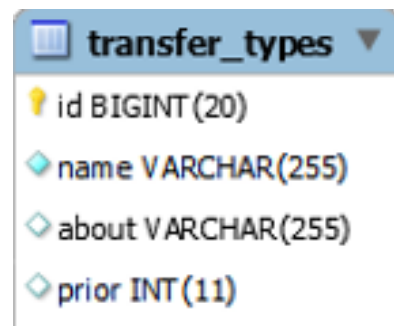


Рисунок 10 – Таблиця «transfer_types»

Структура таблиці «users» (користувачі) приведена на рисунку 11.

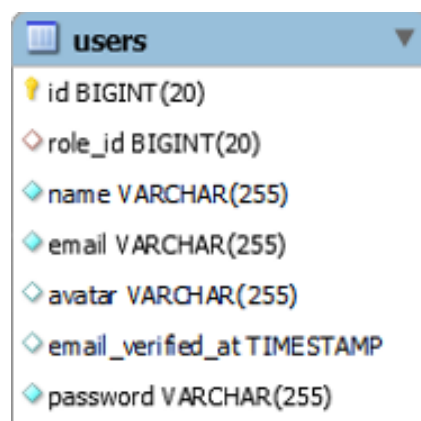


Рисунок 11 – Таблиця «users»

Ці три таблиці безпосередньо пов'язані із таблицею «transfers» (операції обміну) відношенням один до багатьох. Для реалізації таких зв'язків слід додати вторинні ключі.

Структура таблиці «transfers» (операції обміну) приведена на рисунку 12.

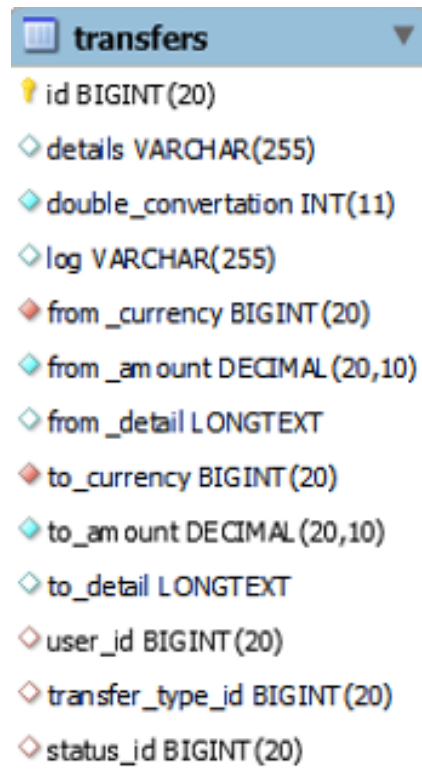


Рисунок 12 – Таблиця «transfers»

Для виконання операцій обміну слід ще створити таблицю «currencies» (валюти), що також буде пов'язана із таблицею «transfers» (операції обміну) відношенням один до багатьох (рисунок 13)

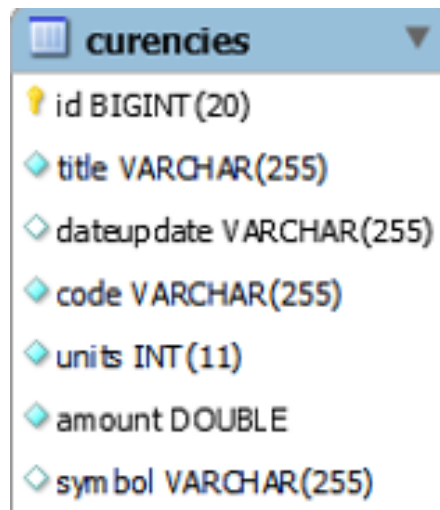


Рисунок 13 – Таблиця «transfers»

У користувачів може бути багато засобів оплати (наприклад, карткові рахунки). З іншого боку ці картки можуть мати різні основні валюти. Тобто слід реалізувати між таблицями «users» та «curencies» зв'язок багато до багатьох. Для цього слід спроектувати додаткову таблицю «user_payments» (засоби оплати), структуру якої приведено на рисунку 14.

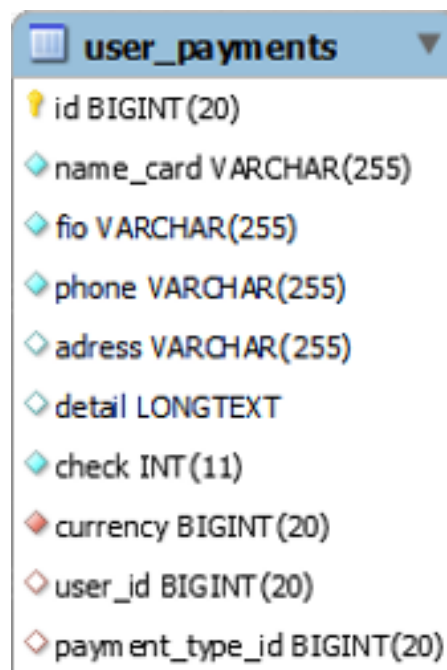


Рисунок 14 – Таблиця «user_payments»

Ці засоби оплати можуть бути різних типів. Тому створимо ще одну таблицю «payment_types» (типи оплати), що буде зв'язана із таблицею «user_payments» відношенням один до багатьох (рисунок 15).

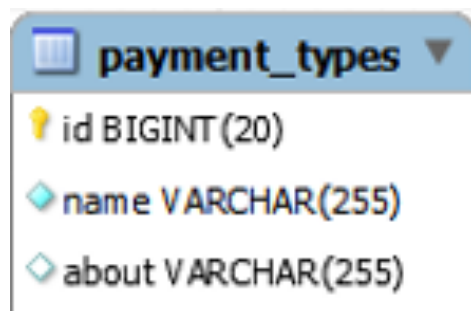


Рисунок 15 – Таблиця «payment_types»

Наступним кроком нормалізації є приведення спроектованих вище таблиць до третьої нормальної форми. Для цього в таблицях слід усунути транзитивні залежності. Аналіз структури спроектованих таблиць показав, що таких транзитивних залежностей немає, а, отже, база даних відповідає вимогам третьої нормальної форми.

Спроектвана й нормалізована база даних приведена на рисунку 16.

На цьому проєктування моделі даних закінчимо й перейдемо до проєктування третьої компоненти патерну MVC видів(Views) або інтерфейсу.

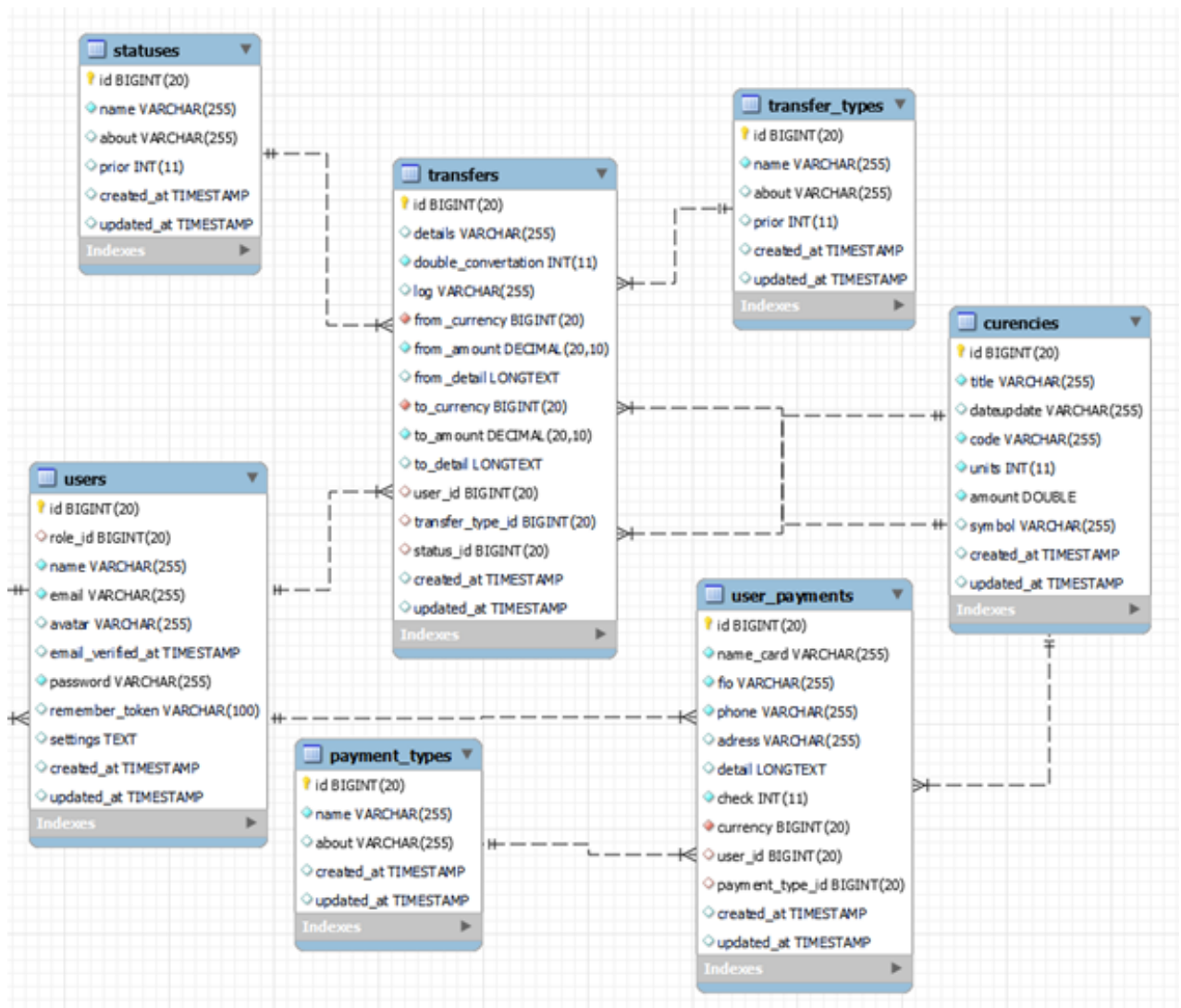


Рисунок 16 – Структура бази даних інформаційної системи управління валютними операціями

2.3 Проектування інтерфейсу інформаційної системи управління валютними операціями

Згідно архітектурного шаблону MVC, за взаємодію з користувачем, тобто за інтерфейсну частину інформаційної системи відповідає компонент Вид (View). Тому, власне, завершальним етапом реалізації архітектурного шаблону MVC й буде проектування інтерфейсу вебдодатку, інформаційної системи управління валютними операціями.

Основи загального проєктування інтерфейсу, зокрема визначення основних Views, було виконано у попередньому розділі. Нижче ж проведемо більш детальне проєктування інтерфейсу, макетування сторінок вебдодатку.

Проєктування інтерфейсу для інформаційної системи управління валютними операціями вимагає врахування призначення різних груп користувачів системи (адміністраторів, звичайного користувача тощо) та їхніх конкретних потреб і обов'язків. Інтерфейс має бути адаптованим до цих вимог. Необхідно розробити логічну та зрозумілу структуру інтерфейсу з інтуїтивною навігацією, меню, панелями інструментів та макетами для полегшення доступу до різних функцій і розділів системи. Оскільки валютні операції часто пов'язані з великими обсягами числових даних, важливо використовувати наочні візуалізації для кращого відображення інформації.

Безпека та контроль доступу відіграють ключову роль, тому необхідно передбачити автентифікацію користувачів, обмеження доступу, захисту конфіденційної інформації та операцій. Також слід забезпечити зручний спосіб отримання зворотного зв'язку та повідомлень від системи через вікна з підказками, повідомлення про помилки та підтвердження дій. Треба намагатись створити інтуїтивний та легкий у використанні інтерфейс з стандартними елементами управління та візуальними метафорами, знайомими користувачам, а також забезпечити належну підтримку та документацію.

Інтерфейс повинен бути адаптивним до різних пристроїв та розширень екрану, а також масштабованим для можливих майбутніх змін у системі без значних змін в інтерфейсі.

Виходячи із вище сказаного, спершу спроектуємо інтерфейс адміністративної частини вебдодатку.

Зрозуміло, що доступ для роботи з цією адміністративною частиною інформаційної системи буде мати користувач «адміністратор».

Макет та попередній інтерфейс адміністративної панелі додатку приведемо на рисунку 17.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

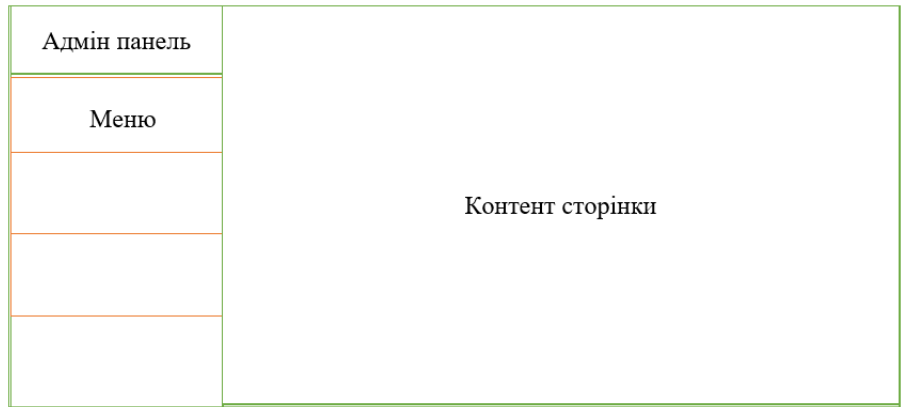


Рисунок 17 – Макет та інтерфейс адміністративної панелі додатку.

Контент сторінки адміністративної панелі в залежності від вибраного пункту меню чи кнопки буде змінюватись (рисунок 18).

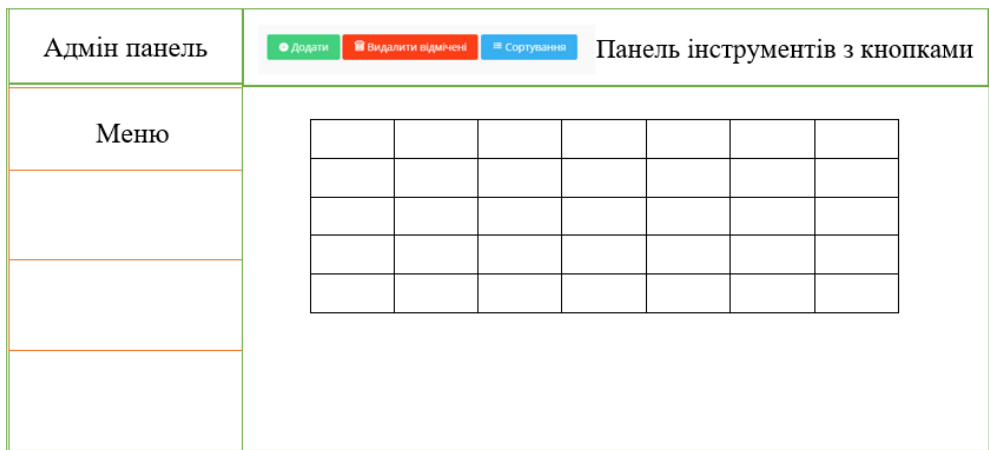


Рисунок 18 – Макет та інтерфейс адміністративної панелі додатку в залежності від вибраного пункту меню чи кнопки.

Для реалізації можливостей додання, оновлення, видалення інформації усіх таблиць спроектованої раніше бази даних слід забезпечити налаштування адміністративної панелі за допомогою наступного інтерфейсу (рисунок 19)

Адмін панель		
Мен	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>
		<input type="text"/>

Рисунок 19 – Налаштування адміністративної панелі.

Проектування інтерфейсу адміністративної частини інформаційної системи виконано.

Далі виконаємо проектування інтерфейсу безпосередньо вебдодатку користувача.

Для незареєстрованого користувача слід спроектувати інтерфейс вебдодатку для перегляду інформації предметної області про валюти, їх курс, послуги, що можуть надаватись тощо.

Для цього доцільно спроектувати інтерфейс такої частини вебдодатку у вигляді односторінкового сайту (Landing Page).

Основні частини спроектованого макету такої сторінки показано на рисунку 20.

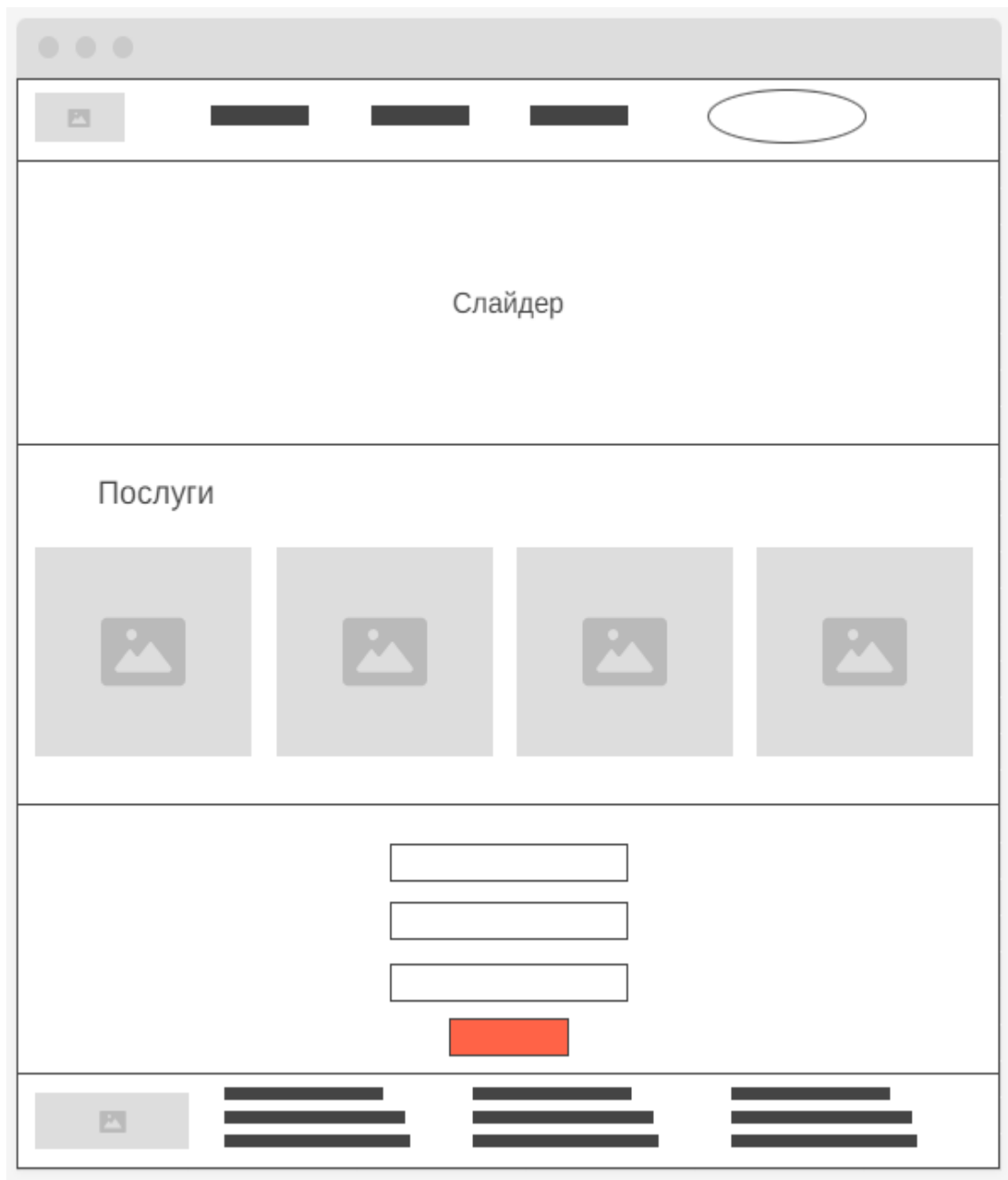


Рисунок 20 – Макет інтерфейсу односторінкового сайту для незареєстрованого користувача.

Зареєстрований користувач має мати ширші можливості інтерфейсу для виконання операцій обміну, безготівкових операцій, тощо.

Для нього слід спроектувати окремий макет інтерфейсу сторінки вебдодатку. Основні частини спроектованого макету такої сторінки для зареєстрованого користувача показано на рисунку 21.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ВАЛЮТНИМИ ОПЕРАЦІЯМИ

3.1 Програмна реалізація інформаційної системи управління валютними операціями

Після завершення проектування системи необхідно перейти до реалізації додатку. На цьому етапі потрібно за допомогою середовища розробки програмного забезпечення та мови програмування по результатам проектних рішень здійснити виконання поставленого завдання.

Проект розроблятиметься у середовищі розробки програмного забезпечення PhpShtorm, який є одним із найпопулярніших, серед розробки веб-додатків з використанням мови програмування php.

Використовуючи командний рядок локального сервера OpenServer та Composer створимо новий проект.

Composer – це пакетний менеджер рівня додатків для мови програмування РНР, який надає кошти з управління залежностями в РНР-додатку. Ідея створення пакетних менеджерів рівня додатків не нова і його автори надихалися вже існуючими на той момент часу npm для Node.js.

Для створення проекту виконаємо команди:

```
$ composer create-project laravel/laravel:^8.0 MoneyExchanger
```

Як результат буде інстальовано фреймворк laravel та створено проект:

```
$ composer create-project laravel/laravel:^8.0 MoneyExchanger
Creating a "laravel/laravel:^8.0" project at "./MoneyExchanger"
Installing laravel/laravel (v8.6.12)
 - Installing laravel/laravel (v8.6.12): Extracting archive
Created project in E:\Project\lara\MoneyExchanger
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
```

Рисунок 21 – Створення проекту laravel

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі перейдемо до програмування спроектованої бази даних за допомогою утиліти phpMyAdmin (рисунок 22).

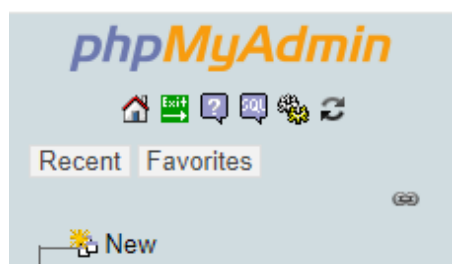


Рисунок 22 – Програмування бази даних в утиліті phpMyAdmin

Далі налаштуємо зв'язок з бд у файлі .env (рисунок 23)

```
.env
APP_ENV=local
APP_DEBUG=true
APP_KEY=base64:FUUzqezbPUoY6H5ggy9opkKM6XRspSfKXu+HJQVIRLg=
APP_URL=http://localhost:8000

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=change_db
DB_USERNAME=root
DB_PASSWORD=root
```

Рисунок 23 – Налаштування зв'язку з бд у файлі .env

Щоб забезпечити можливість додання, видалення, оновлення інформації, адміністрування вебдодатку слід реалізувати адміністративну частину. Для цього використаємо готове стандартне рішення. Для цього слід під'єднати адмінчастину за посиланням

<https://voyager-docs.devdojo.com/getting-started/installation> [21]

Далі встановлюємо цю адмінчастину через композер, виконуючи команди *composer require tcg/voyager*

Інсталуємо скачане, виконуючи команди

```
php artisan voyager:install
```

Результат встановлення адмінчастини приведено на рисунку 24

```
$ php artisan voyager:install --with-dummy
Publishing the Voyager assets, database, and config files
Copied Directory [\vendor\tcg\voyager\publishable\database\seeders] To [\
e\seeders]
Publishing complete.
Migrating the database tables into your application
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (299.19ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (267.04ms)
Migrating: 2016_01_01_000000_add_voyager_user_fields
Migrated: 2016_01_01_000000_add_voyager_user_fields (423.42ms)
Migrating: 2016_01_01_000000_create_data_types_table
Migrated: 2016_01_01_000000_create_data_types_table (1,111.28ms)
Migrating: 2016_05_19_173453_create_menu_table
Migrated: 2016_05_19_173453_create_menu_table (1,233.16ms)
Migrating: 2016_10_21_190000_create_roles_table
```

Рисунок 24 – Результат встановлення адмінчастини

При встановленні, розгортаються міграції таблиць адмін панелі. Як результат міграцій, у phpMyAdmin з'являться наступні таблиці (рисунок 25).

	Table ▲	Action
<input type="checkbox"/>	data_rows	★ [іконки]
<input type="checkbox"/>	data_types	★ [іконки]
<input type="checkbox"/>	menus	★ [іконки]
<input type="checkbox"/>	menu_items	★ [іконки]
<input type="checkbox"/>	migrations	★ [іконки]
<input type="checkbox"/>	password_resets	★ [іконки]
<input type="checkbox"/>	permissions	★ [іконки]
<input type="checkbox"/>	permission_role	★ [іконки]
<input type="checkbox"/>	roles	★ [іконки]
<input type="checkbox"/>	settings	★ [іконки]
<input type="checkbox"/>	translations	★ [іконки]
<input type="checkbox"/>	users	★ [іконки]

Рисунок 25 – Результат міграцій таблиць у phpMyAdmin.

Далі можемо запустити на виконання створений проект, виконавши команди

```
php artisan serve
```

Після цього, перейшовши за посиланням <http://localhost:8000/admin> можемо зареєструватися адміністратором (рисунок 26).

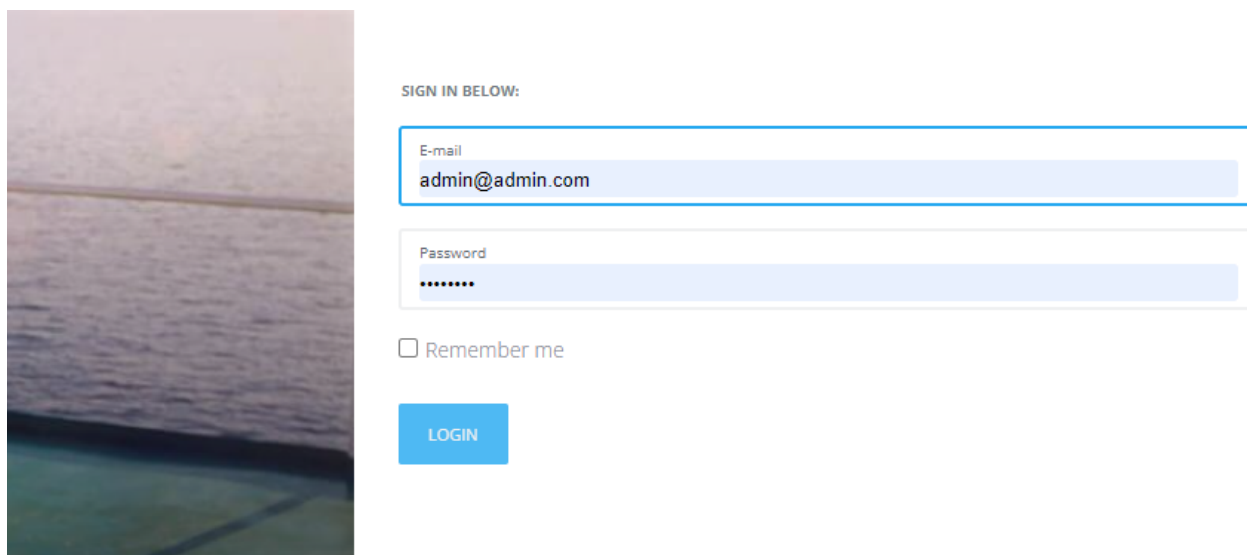


Рисунок 26 – Реєстрація адміністратора

Далі можемо авторизуватися адміністратором, використовуючи наступні логін та пароль:

- email: admin@admin.com
- password: password

Після успішної авторизації побачимо наступну адміністративну панель (рисунок 27).

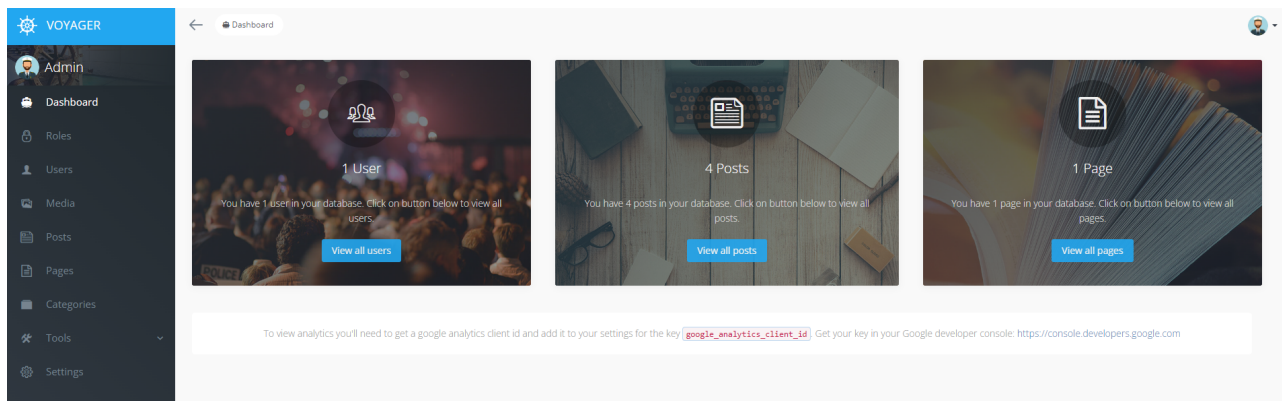


Рисунок 27 – Адміністративна панель додатку.

Далі додаємо можливість реєстрації користувачам, виконавши команди:

```
composer require laravel/ui
php artisan ui:auth --views
```

Для визначення актуальних курсів валют, їх оновлення в додатку, будемо використовувати офіційний сайт НБУ. Для цього слід спарсити дані із офіційного сайту НБУ [22]

```
Route::get('/json', function () {
    $url = 'https://bank.gov.ua/NBU_Exchange/exchange?json';
    $response = file_get_contents($url);
    $newsData = json_decode($response);
    $data = array_filter($newsData);
    if(isset($_GET['v']))
    $data = collect($data)->where("CurrencyCodeL", $_GET['v'])->all();
    return response()->json($data); });
```

Результатом такого парсингу (оновлення) буде json масив, вміст якого можна перевірити за наступним посиланням

<http://localhost:8000/json?v=USD>

Власне, один з наборів даних оновлення, буде мати такий вигляд

```
{"7":{"StartDate":"15.04.2024","TimeSign":"0000","CurrencyCode":"840","CurrencyCodeL":"USD","Units":1,"Amount":39.399}}
```

Далі виконаємо генерації міграцій, моделей та контролерів. Розглянемо приклад таких операцій для моделі «Curency». [25] Для цього слід виконати наступний код

```
php artisan make:model Curency -a
```

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

Результат виконання такої команди успішний та приведений на рисунку 28

```
Model created successfully.  
Factory created successfully.  
Created Migration: 2024_04_14_083721_create_curencies_table  
Seeder created successfully.  
Request created successfully.  
Request created successfully.  
Controller created successfully.  
Policy created successfully.
```

Рисунок 28 – Результат міграції моделі «Curency».

Далі створимо структуру таблиць у міграції/

```
class CreateCurenciesTable extends Migration  
{  
    public function up()  
    {  
        Schema::create('curencies', function (Blueprint $table) {  
            $table->id();  
            $table->string('title');  
            $table->string('dateupdate')->nullable();  
            $table->string('code');  
            $table->integer('units')->default(1);  
            $table->double('amount')->default(1);  
            $table->string('symbol')->nullable();  
            $table->timestamps();  
        });  
    }  
    public function down()  
    {  
        Schema::dropIfExists('curencies');  
    }  
}
```

Та внесемо зміни в модель, щоб можна було оновлювати дані із сайту

```
class Curency extends Model  
{  
    use HasFactory;  
    protected $fillable = [  
        'title', 'dateupdate', 'code',  
        'units', 'amount', 'symbol'  
    ];  
}
```

Розгорнемо міграцію, виконавши команду

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

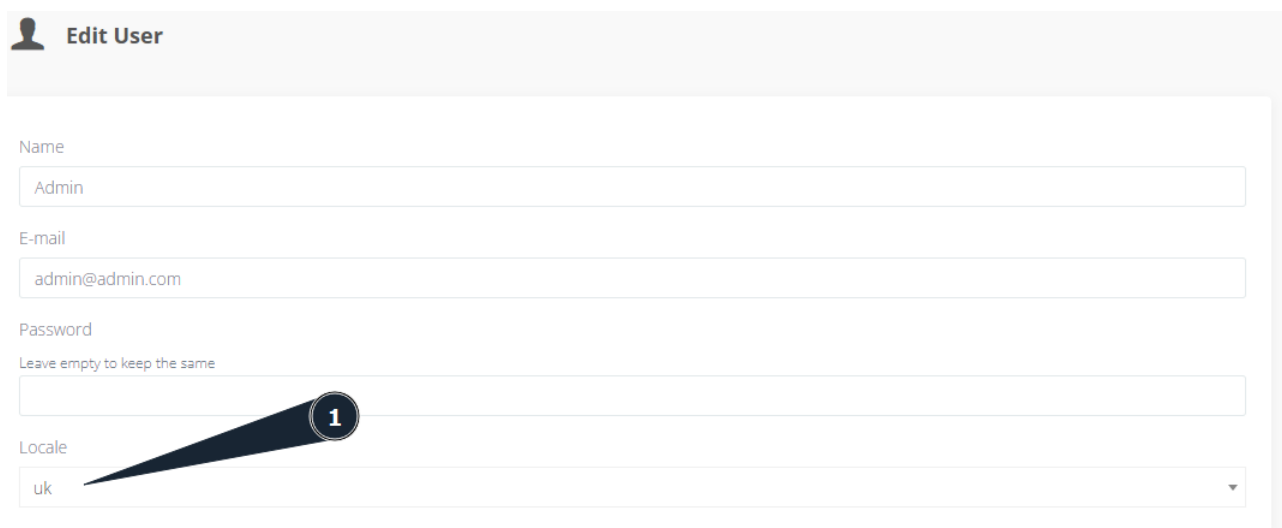
php artisan migrate

Результат виконання такої команди успішний та приведений на рисунку 29.

```
Migrating: 2024_04_14_083721_create_curencies_table  
Migrated: 2024_04_14_083721_create_curencies_table (672.59ms)
```

Рисунок 29 – Розгортання міграції моделі «Curency».

Далі реалізуємо у адміністративній панелі можливість роботи з таблицею, але спершу здійснимо налаштування для підтримки адміністративної панелі української мови, зазначивши це в профілі адміністратора (рисунок 30).



Edit User

Name
Admin

E-mail
admin@admin.com

Password
Leave empty to keep the same

Locale
uk

Рисунок 30 – Налаштування для підтримки адмінпанелі української мови

Використаємо інструмент BREAD, щоб створити вигляд (View) для роботи із таблицею Curency в адміністративній панелі (рисунок 31).

Create BREAD for currencies table

Currencies BREAD info

Table Name
currencies

Display Name (Singular)
Курс валют

Display Name (Plural)
Курси валют

URL Slug (must be unique)
currencies

Icon (optional) Use a Voyager Font Class
Icon to use for this Table

Model Name
App\Models\Currency

Controller Name
Controller Name

Policy Name
Policy Class Name

Generate Permissions
 Yes No

Server-side Pagination
 No

Order column
id

Order display column
id

Order direction
Ascending

Default server-side search field
-- None --

Description
Description

Рисунок 31 – Використання інструменту BREAD, щоб створити вигляд (View) для роботи із таблицею Currencies в адміністративній панелі

Далі додамо новий запис у таблицю Currencies бази даних (рисунок 32).

Курси валют Додати Видалити відмічені Сортування

Показати 10 записів Пошук:

<input type="checkbox"/>	Title	Dateupdate	Code	Units	Amount	Symbol	Created At	Дії
<input type="checkbox"/>	Американський долар		USD	1	1	\$	2024-04-14 08:58:32	Переглянути Редагувати Видалити

Показано від 1 до 1 з 1 записів < Попередня 1 Наступна >

Рисунок 32 – Додання нового запису у таблицю Currencies бази даних

Тепер створимо в Laravel команду, яку можна буде виконати з консолі для того, щоб відбувся парсинг валют:

```
php artisan make:command UpdCurrencies --command=currency:cron
```

, де currency:cron – власне команда для виконання.

Після цього структура файлів проекту буде змінена наступним чином (рисунок 33).

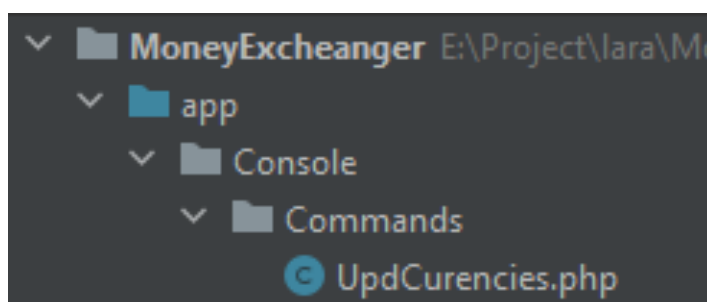


Рисунок 33 – Змінена структура файлів проекту

Далі запрограмуємо наступну можливість, щоб команда передивлялася які є валюти в бд і відповідно до існуючих оновлювала дані з сайту НБУ (https://bank.gov.ua/NBU_Exchange/exchange?json) [22]

```
public function handle()
{
    $all = \App\Models\Currency::get();
    foreach ($all as $cur) {
        $value = $this->getCurrency($cur->code);
        $cur->amount = $value->Amount;
        $cur->units = $value->Units;
        $cur->dateupdate = $value->StartDate;
        $cur->save();
    }
}

private function getCurrency($cur){
    $url = 'https://bank.gov.ua/NBU_Exchange/exchange?json';
    $response = file_get_contents($url);
    $newsData = json_decode($response);
    $data = array_filter($newsData);
    $data = collect($data)->where("CurrencyCodeL",$cur)->first();
    return $data;
}
```

Тепер зробимо так, щоб ця команда автоматично повторювалась через певний час, тобто щоб кожного дня оновлювались дані із валютами. [23]

Для цього слід створити cron job, який створюється досить просто, необхідно у файлі app/Console/Kernel.php прописати виконання команди у цьому розділі:

```
protected function schedule(Schedule $schedule)
{
    $schedule->command('inspire')->hourly();
}
```

В цей метод пропишемо виклик нашої команди, і зробимо, щоб оновлялись дані кожні 6 годин.

```
$schedule->command('currency:cron')
->everySixHours()
->appendOutputTo('cron.txt');
```

Для запуску cron job, виконаємо наступну команду, яка запустить відповідний процес.[24]

```
php artisan schedule:work
```

Результат виконання такої команди успішний та приведений на рисунку 34.

Рисунок 34 – Результат виконання cron job

За такою ж схемою далі створимо міграції, моделі для всіх необхідних таблиць, що спроектовані у попередньому розділі та зображені на рисунку 16 – Структура бази даних інформаційної системи управління валютними операціями.

Таким чином описано програмну реалізацію адміністративної частини вебдодатку.

Для програмної реалізації спроектованого макету сторінки незареєстрованого користувача, що приведено на рисунку 20 у попередньому розділі використаємо набір технологій html + css + js [29].

Зверстаний односторінковий сайт для незареєстрованого користувача слід його адаптувати у laravel'і. Тобто слід налаштувати підвантаження ресурсів, налаштувати посилання, щоб вони працювали вірно. Тому всі файли стилів,

шрифтів, зображень та скриптів помістимо в папку public\landing (рисунок 35).

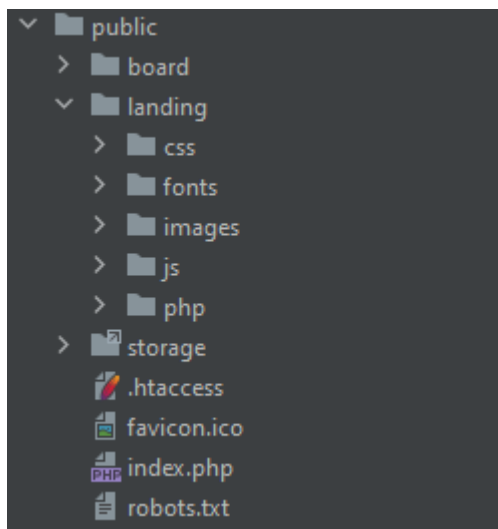


Рисунок 35 – Налаштування односторінкового сайту

Саму розмітку помістимо в папку resource\view\landing\ . Для файлу розмітки сторінки змінимо розширення з .html до .blade.php для того, щоб можна було використовувати можливості шаблонізатора Blade у файлі розмітки (рисунок 36).

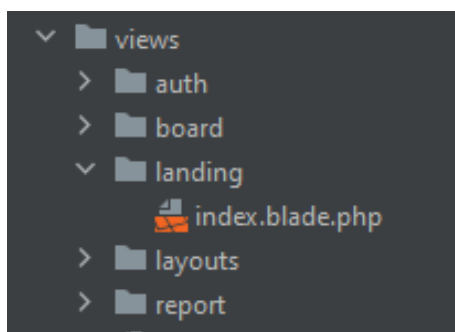


Рисунок 36 – Налаштування розмітки односторінкового сайту

Далі внесемо деякі зміни в код для того, щоб підтягувати ресурси із відповідних шляхів у проекті.

```
<header class="top-header">
  <div class="header_top">
    <div class="container">
      <div class="row">
```

```

        <div class="logo_section">
            <a class="navbar-brand" href="/">

            </a></div>
            <div class="site_information">
                <ul>
                    <li>
                        <a href="mailto:exchang@gmail.com">

                        exchang@gmail.com</a>
                    </li>
                    <li>
                        <a href="tel:exchang@gmail.com">

                        +380980000098</a>
                    </li>
                    <li>
<a class="join_bt" href="{{ route('Login') }}">Приєднатись</a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</header>

```

, де `asset('landing/images/Logo.png')` – доступ до ресурсів в папці `public\landing`.

`route('Login')` – використання посилання засобами `blade` та `Laravel`.

Таким самим способом підключаємо і скрипти:

```

<script src="{{asset('landing/js/jquery.min.js')}}"></script>
<script src="{{asset('landing/js/popper.min.js')}}"></script>
<script src="{{asset('landing/js/bootstrap.min.js')}}"></script>

```

Після адаптації, слід зробити так, щоб при переході на головну адресу сайту відкривався цей односторінковий сайт. Для цього потрібно в файлі роутів (`route\web.php`) прописати необхідний код:

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

```
Route::get('/', function () {  
    return view('landing.index');  
});
```

Далі перейдемо до програмної реалізації сторінки вебдодатку для зареєстрованого користувача.

Спочатку встановимо пакунки для роботи з користувачами (реєстрація/авторизація)

```
composer require laravel/ui  
php artisan ui:auth --views
```

Після цього, при переході за адресою /login отримаємо форму для авторизації. Таким чином реалізується модуль роботи з користувачами, що автоматично включає реєстрацію та авторизацію користувачів (рисунок 37).

ExchangeCurrence Авторизація Реєстрація

Рисунок 37 – Налаштування розмітки односторінкового сайту

За допомогою міграцій створимо інші моделі та внесемо необхідні зміни для реалізації інформаційних сутностей додатку (рисунок 38).

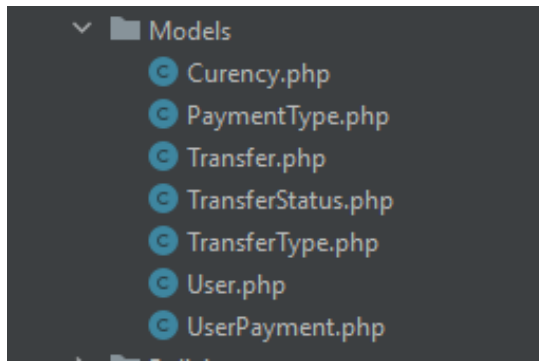


Рисунок 38 – Моделі вебдодатку

Далі в моделях опишемо доступні поля для роботи з додатком.

```
protected $table = 'user_payments';
protected $primaryKey='id';
protected $casts = [
    'detail' => 'array',
];
protected $fillable = [
    'name_card', 'fio', 'phone', 'adress', 'detail',
    'check', 'currency', 'user_id', 'payment_type_id'
];
```

\$table – назва таблиці, з якою зв'язана модель

\$primaryKey – ключове поле

\$fillable – доступні поля для редагування

\$casts – перетворення поля типу *json* в даному випадку масиву даних

Пропишемо код для зв'язування даних з різних таблиць, для зручного доступу із зв'язаних моделей:

```
public function user()
{
    return $this->belongsTo(User::class);
}
public function type()
{
    return $this->belongsTo(PaymentType::class, 'payment_type_id');
}
```

Аналогічно зробимо зі всіма іншими моделями.

Далі виконуємо верстання макету, що приведений на рисунку 21 та проводимо адаптацію до laravel. Це виконаємо аналогічно до описаного вище при верстанні односторінкового сайту.

Для цього всі файли стилів, шрифтів, зображень та скриптів помістимо в папку public\board (рисунок 39).

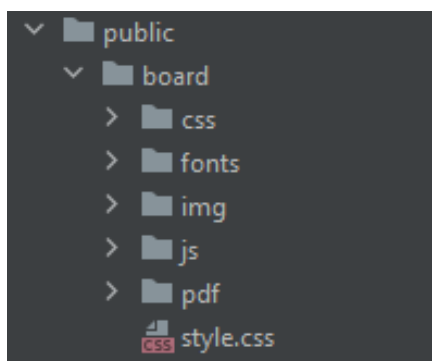


Рисунок 39 – Адаптація до Laravel сайту зареєстрованого користувача

Саму розмітку помістимо в папці resource\view\board\, а для файлу розмітки змінимо розширення з .html до .blade.php для використання можливостей шаблонізатора Blade (рисунок 40).

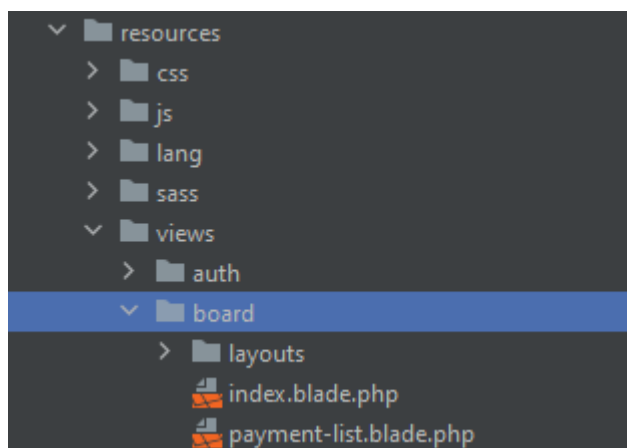


Рисунок 40 – Налаштування розмітки сайту зареєстрованого користувача

Для того, щоб в усіх сторінках не дублювати код заголовка сторінки (хедера) та підвалу сторінки (футера) винесемо його в окремий файл і потім просто підключатимемо необхідний контент, коли це буде потрібно. Таку можливість легко надає шаблонізатор Blade (рисунок 41).

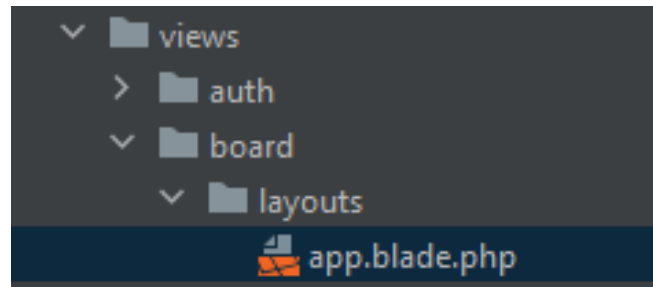


Рисунок 41 – Шаблон заголовка та підвалу сторінки

Вигляд файлу розмітки заголовку сайту та підвалу сторінки приведемо на рисунку 42.

У цьому файлі `@yield('content')` вказує місце, куди вставлятиметься необхідний контент.

```
<!doctype html>
<html class="no-js" lang="en">

<head...>

<body>
  <!-- ... -->

  <div class="left-sidebar-pro">
    <nav id="sidebar" class="">
      <div class="sidebar-header" ...>
        <div class="nalika-profile" ...>
          <div class="left-custom-menu-adp-wrap comment-scrollbar">
            <nav class="sidebar-nav left-sidebar-menu-pro">
              <ul class="metismenu" id="menu1">
                <li class="active">
                  <a class="has-arrow" href="index.html">
                    <i class="icon nalika-home icon-wrap"></i>
                    <span class="mini-click-non">Меню</span>
                  </a>
                  <ul class="submenu-angle" aria-expanded="true">
                    <li><a title="Головна" href="{{ route('home') }}"><span class="mini-sub-pro">Головна</span></a>
                    <li><a title="Product List" href="{{ route('transfers') }}"><span class="mini-sub-pro">Тр...</a>
                    <li><a title="Product List" href="{{ route('payments') }}"><span class="mini-sub-pro">Спо...</a>
                  </ul>
                </li>
              </ul>
            </nav>
          </div>
        </div>
      </nav>
    </div>

    <!-- Start Welcome area -->
    <div class="all-content-wrapper">
      <div class="container-fluid" ...>
        <div class="header-advance-area" ...>

          @yield('content')

        <div class="footer-copyright-area" ...>
      </div>
    </div>
  </body>
</html>
```

Рисунок 42 – Верстання заголовка та підвалу сторінки.

Таким чином, реалізуємо головну сторінку для авторизованого користувача (рисунок 43).

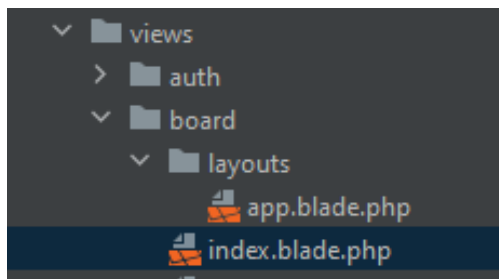


Рисунок 43 – Верстання заголовка та підвалу сторінки

Весь необхідний вміст сторінки помістимо в наступну конструкцію:

```
@extends('board.Layouts.app')
@section('content')
... код
@endsection
```

, де `@extends('board.layouts.app')` – вказує, в який вигляд вставлятимемо код сторінки для відображення;

`@section('content')` – тут має бути код, який необхідний для виводу.

Оскільки на цю сторінку потрібно виводити певні дані, то їх слід витягнути із БД. Оскільки Laravel використовує архітектуру MVC, всю логіку має виконувати контролер. Тому слід створити необхідний контролер, де і будуть опрацьовуватись запити від користувача

```
class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }
}
```

Оскільки головний роут за правилом позначається словом `index`, тому створимо відповідний метод у контролері. Так як хочемо відображати актуальну інформацію, то в методі викличемо раніше створену команду для отримання

В кодї використано метод `getCurrencyAWeek`, який відповідає за отримання даних за останній тиждень для певної валюти

```
private function getCurrencyAWeek($cur){
    $day1 = Carbon::now();
    $day2 = Carbon::now()->subDays(7);
    $url=
    'https://bank.gov.ua/NBU_Exchange/exchange_site?start='.$day2.'&end='.$day1.'&valcode='.$cur.'&sort=exchangedate&order=desc&json';
    $response = file_get_contents($url);
    $newsData = json_decode($response);
    $data = array_filter($newsData);
    $res = [];
    foreach ($data as $item){
        array_push($res, $item->rate);
    }
    return $res; }

```

Далі слід зареєструвати маршрут для відкриття відповідної сторінки:

```
Route::get('/home',
    [App\Http\Controllers\HomeController::class, 'index'])
->name('home');

```

Для створення графіку зміни курсу валют за тиждень використаємо бібліотеку Chart. Щоб використовувати її можливості спершу слід її підключити до сторінки:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js">
</script>

```

Для використання графіку пропишемо наступний код створюючи об'єкт класу Chart, в який потрібно передати параметри вісів:

```
<script>
    new Chart(document.getElementById("line-chart"), {
        type: 'line',
        data: {
            labels: {{ json_encode($days) }},
            datasets: [{
                data: {{ json_encode($usd) }},
                label: "Долар США",
                borderColor: "#3cba9f",
                fill: false
            }, {
                data: {{ json_encode($eur) }},
                label: "Євро",
            }
        }
    });

```

```

        borderColor: "#8e5ea2",
        fill: false
    }
]
},
options: {
    title: {
        display: true,
        text: 'Курс валют за останій тиждень'
    }
}
});
</script>

```

Далі створимо сторінку для відображення способів оплати метод в контролері для вибірки даних:

```

public function payments()
{
    $payments= UserPayment::where('user_id', user()->id)->get();
    $currency = Currency::get();
    $user_count = User::get()->count();
    $transfers_count = Transfer::get()->count();
    $user_payment_count = UserPayment::get()->count();
    $currency_count = $currency->count();
    return view('board.payment-list', compact('payments',...));
}

```

Створимо маршрут опрацювання запиту користувача:

```

Route::get('/payments',
    [App\Http\Controllers\HomeController::class, 'payments'])
->name('payments');

```

Далі створимо розмітку з використанням можливостей фреймворка для побудови таблиці із значеннями:

```

@extends('board.layouts.app')
@section('content')

<table>
    <tr>
        <th>Користувач</th>
        <th>Тун</th>
        <th>Назва</th>
        <th>Валюта</th>
        <th>Кількість коштів</th>
        <th>Активна</th>
    </tr>

```

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		


```
[App\Http\Controllers\HomeController::class, 'payments_store'])
->name('payments_store');
```

Код опрацювання в контролері:

```
public function payments_store(Request $request)
{
    $requestData = $request->all();
    $requestData['payment_type_id'] =
        PaymentType::where('name', 'Картка')->firstOrFail();
    $requestData['user_id'] = Auth::user()->id;
    $temp = UserPayment::create($requestData);
    $arra = array(
        'bank' => $request->bank,
        'number' => $request->number,
        'name' => $request->name,
        'expiry' => $request->expiry,
        'cvc' => $request->cvc
    );
    $str = json_encode($arra, JSON_FORCE_OBJECT);
    $temp->update([
        'detail' => $str
    ]);
    $temp->save();

    return redirect('payments')
        ->with('flash_message', 'Картку додано!');
}
```

Таким чином реалізовано основні кроки програмної реалізації поставленого завдання. Далі слід перейти до тестування програмного продукту.

3.2 Тестування інформаційної системи управління валютними операціями

Тестування – це процес технічного дослідження, який виконується на вимогу замовників і призначений для визначення інформації про якість продукту відносно контексту, в якому він має використовуватись. До цього процесу входить виконання програми з метою знаходження помилок.

Тестування не може повністю забезпечити коректність програмного забезпечення, воно тільки порівнює стан і поведінку продукту зі вказаною специфікацією.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

Для проведення етапу тестування було розроблено тестові випадки для тестування користувацького інтерфейсу. Кожен тестовий випадок містить детальні кроки, тестові дані і очікувані результати.

Метою тестування користувацького інтерфейсу є перевірка правильності навігації по об'єкту тестування та відповідність опису елементів до дій, які вони виконують.

Результати виконаних тестових випадків користувацького інтерфейсу наведені у таблиці 1.

Таблиця 1 – Випадки тестування користувацького інтерфейсу

Кроки/дії	Очікувані результати	Стан
1. Завантаження початкової сторінки	Завантаження Landing Page вебдодатку.	Пройдено
1	2	3
2. Реєстрація та авторизація адміністратора	Завантаження форми реєстрації/авторизації і виконання таких операцій.	Пройдено
3. Завантаження адміністративної панелі для керування вмістом	З'явилась панель керування адміністратора для додання, редагування, видалення даних.	Пройдено
4. Виконання операцій додання, оновлення, видалення даних.	Успішне здійснення таких операцій	Пройдено
5. Виведення курсів валют	Виведення таблиці із курсами валют	Пройдено

Кроки/дії	Очікувані результати	Стан
6. Виведення графіку зміни курсу валюти	Має завантажитись графік зміни курсу для певної валюти	Пройдено
7. Додання нової карти користувача	Додання усіх необхідних даних нової картки на відповідній формі і створення картки	Пройдено
8. Вибір способу оплати банківською карткою, вибір однієї з карт користувача при обміні валют	Вибір із доступних карток користувача потрібної для здійснення операцій	Пройдено
9. Виведення таблиці із трансферами (обмінами) та їх статусом	Виводяться виконані та очікуючі трансфери у таблиці	Пройдено
10. Виконання трансферу	Вказання параметрів обміну у формі трансферу та виконання операції обміну	Пройдено

Отже, було проведено 10 тестів «Випадки тестування користувацького інтерфейсу», із яких усі пройшли успішно. Тому тестування користувацького інтерфейсу розглядається як успішне.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

3.3 Інструкція з інсталяції та експлуатації інформаційної системи управління валютними операціями

Для встановлення серверної частини необхідно розгорнути сервер бази даних на основі СУБД MySQL, веб-сервер, що включатиме розроблений веб-проект та середовище PHP. Розгортання серверної частини можна реалізувати за допомогою будь-якого локального вебсервера, наприклад OpenServer.

Для роботи кінцевих користувачів із вебдодатку необхідним є діюче під'єднання до мережі Інтернет, як сервера так і користувача. Якщо таке з'єднання активне, то для роботи користувачів достатнім є наявність на його робочій станції будь-якої програми веббраузера.

Проте для остаточної коректної роботи ресурсу його слід викласти на будь-якому безкоштовному чи платному хостингу.

Основні компоненти та вимоги для успішної інсталяції вебдодатку приведені на рисунку 45.

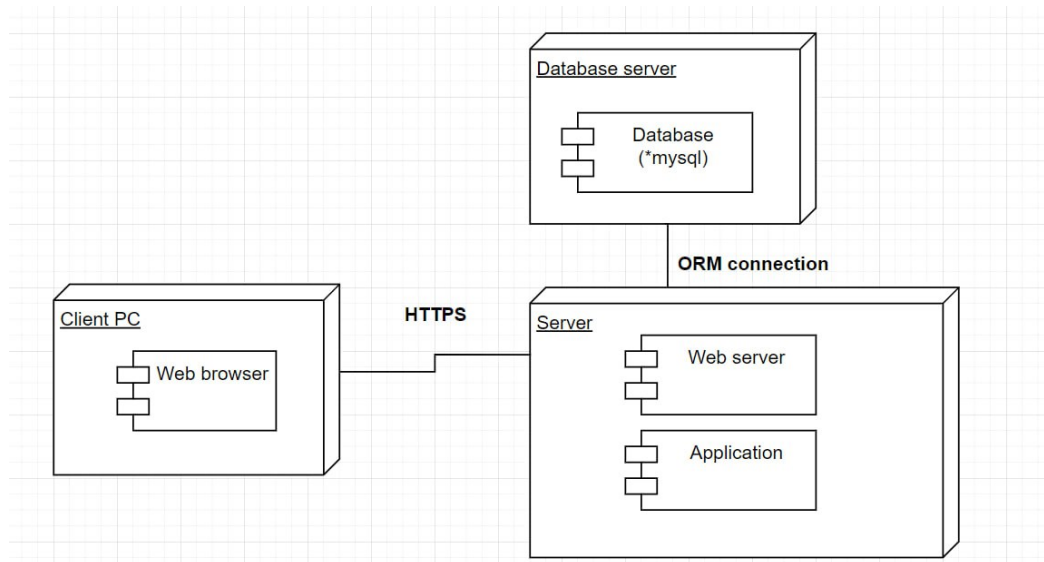


Рисунок 45 – Діаграма розгортання додатку

Після інсталяції можна почати експлуатацію інформаційної системи управління валютними операціями.

доступні у вебдодатку, для реєстрації чи авторизації, щоб отримати більше можливостей (рисунок48).

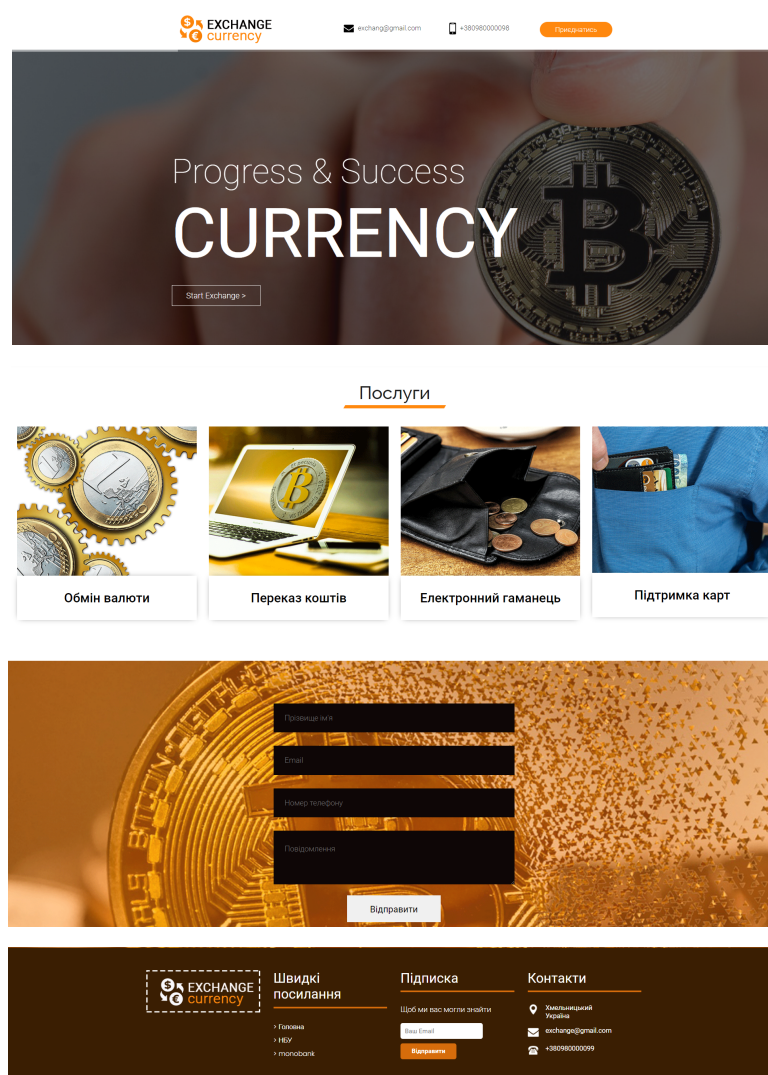


Рисунок 48 – Вигляд одно сторінкового сайту вебдодатку

Для отримання доступу до отримання послуг трансферу, обміну валют та інших можливостей користувач має зареєструватися. Для цього слід внести інформацію у наступну форму (рисунок 49).

										Арк.
										63
Зм.	Арк.	№ докум.	Підпис	Дата						

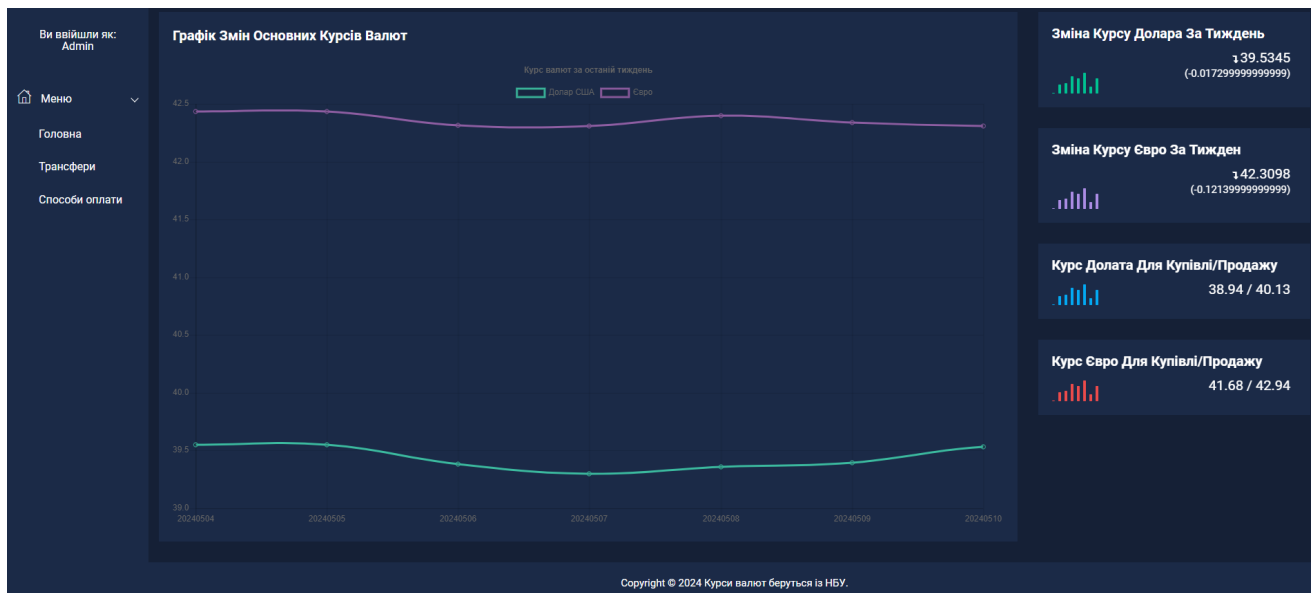


Рисунок 51 – Графік зміни курсу валют.

Для виконання операцій обміну валют слід спершу обрати спосіб оплати (рисунок 52).

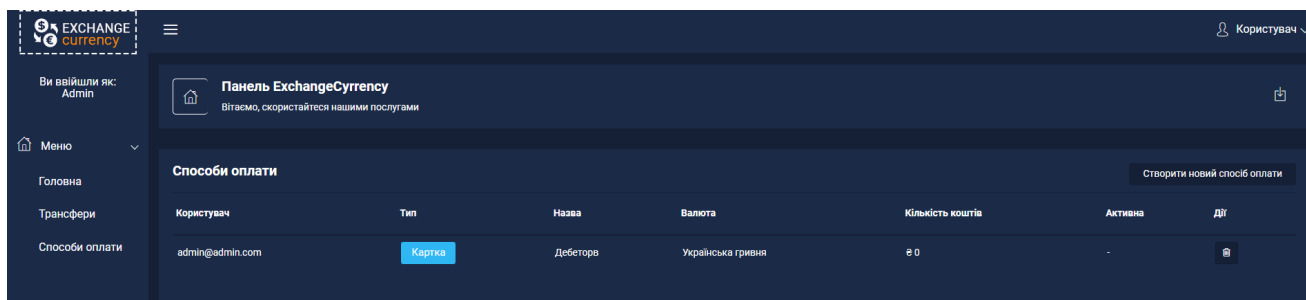


Рисунок 52 – Вибір способу оплати.

Для цього може бути необхідно додати банківську карту для здійснення безготівкової оплати. Для цього слід внести дані у таку форму (рисунок 53).

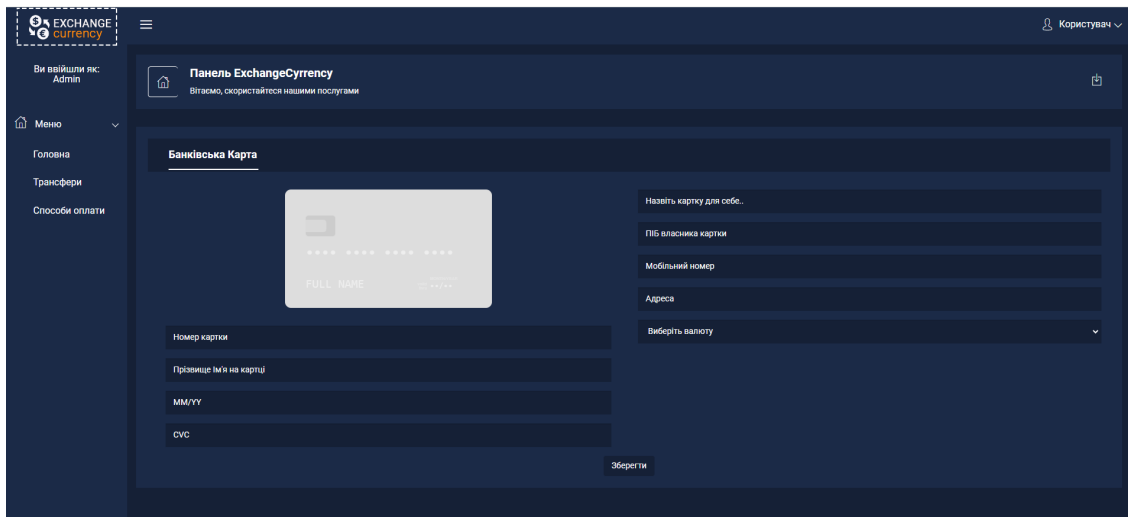


Рисунок 53 – Додавання нової банківської картки.

Власне після цього можна перейти на вкладку «Трансфери», де користувач може побачити виконані та створені трансфери (рисунок 54).

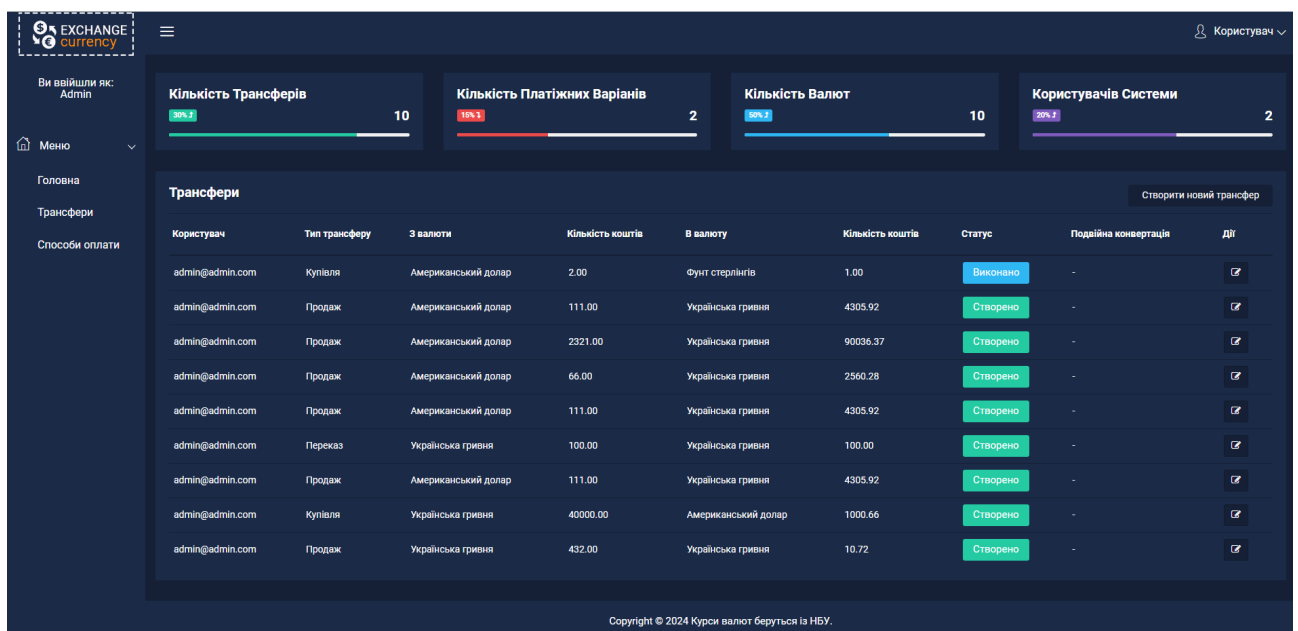


Рисунок 54 – Виконані та створені трансфери.

У цьому вікні можна створити новий трансфер, натиснувши відповідну кнопку. З'явиться нове вікно, де слід вибрати суму, валюти для обміну, способи оплати та отримання коштів (рисунок 55).

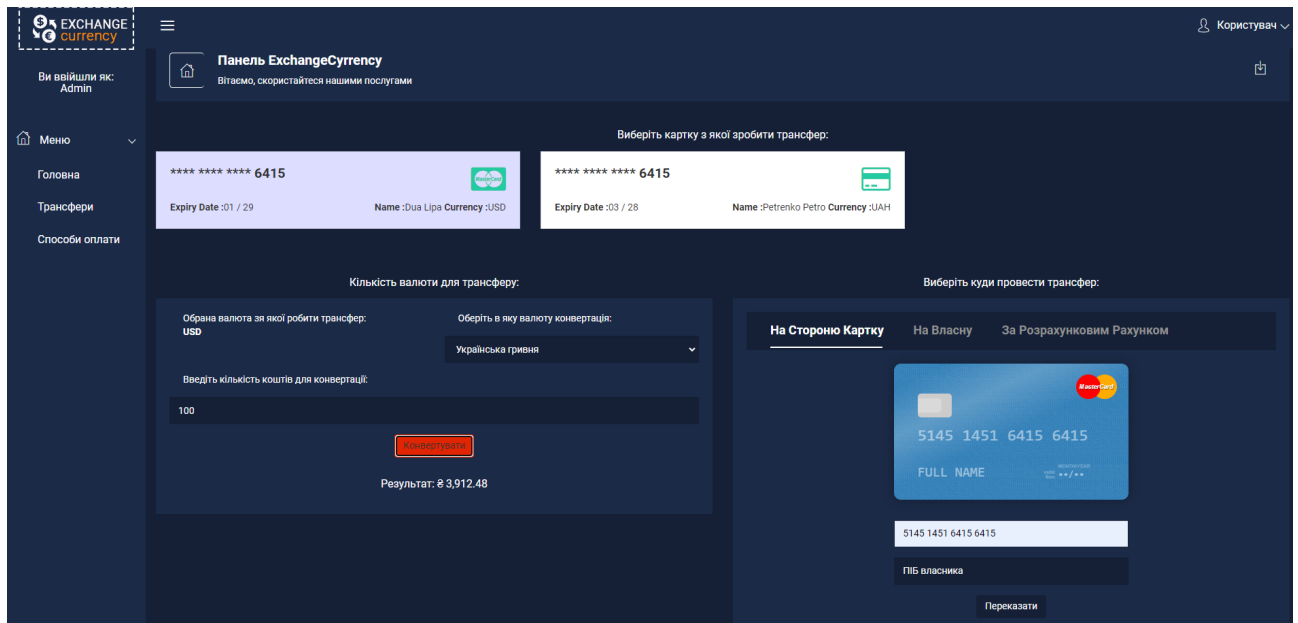


Рисунок 55 – Виконання трансферу (обміну валюти).

На цьому опис основних етапів інсталяції та експлуатації вебдодатку закінчено.

3.4 Висновки

У цьому розділі описаний хід програмної реалізації вебдодатку починаючи від встановлення платформи, розгортання міграцій, створення моделей, контролерів, представлень, та завершуючи проведенням тестування для визначення ступеня готовності програмного забезпечення для подальшої експлуатації.

Також була складена інструкція з інсталяції та експлуатації програмного продукту, що має полегшити відповідні дії користувачів.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі було розроблено вебдодаток управління валютними операціями.

Метою кваліфікаційної роботи була розробка комплексного вебдодатку, який надає користувачам зручні інструменти для ефективного управління валютними операціями.

Проведено огляд та аналіз предметної області, на основі якого було поставлено задачі до кваліфікаційної роботи. В результаті аналітичного огляду було визначено недоліки та переваги подібних програм.

В розділах роботи проаналізовано основні функціональні та не функціональні вимоги до програмного продукту, спроектовано структуру кваліфікаційного проекту, яка супроводжується UML діаграмою варіантів використання програмного засобу. Програмне забезпечення розроблене з використанням вебтехнологій, за основу взято фреймворк Laravel, програмна реалізація здійснювалась засобами мови програмування PHP, СУБД MySQL.

Розробка програмної системи перебувала у всіх стадіях життєвого циклу створення програмного забезпечення. Спроектовано інтерфейс сайту, який забезпечує максимальну зручність та інтуїтивну зрозумілість у користуванні ним.

Здійснено тестування програмної системи, за допомогою тестових випадків, що дали позитивні результати та підтвердити працеспроможність програмного рішення.

Зазначені аспекти роблять інформаційну систему управління валютними операціями комплексним, функціональним та, потенційно, актуальним на ринку IT програмним продуктом.

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Офіційна документація Laravel. URL: <https://laravel.com/docs> (дата звернення: 11.04.2024).
2. Блог по Laravel. URL: <https://laracasts.com/> (дата звернення: 11.04.2024).
3. Офіційний репозиторій Laravel. URL: <https://github.com/laravel/laravel> (дата звернення: 11.04.2024).
4. Колекція корисних ресурсів. URL: <https://github.com/alrik11es/awesome-laravel> (дата звернення: 13.04.2024).
5. Форум та спільнота по Laravel. URL: <https://laravel.io/forum> (дата звернення: 15.04.2024).
6. Форум та спільнота по Laravel. URL: <https://stackoverflow.com/questions/tagged/laravel> (дата звернення: 15.04.2024).
7. Документація пакетів та бібліотек. URL: <https://github.com/barryvdh/laravel-debugbar> (дата звернення: 15.05.2024).
8. Документація пакетів та бібліотек. URL: <https://github.com/laravel/sanctum> (дата звернення: 15.04.2024).
9. Довідник з HTML та CSS. URL: <https://css.in.ua> (дата звернення: 15.05.2024).
10. Bootstrap. URL: <https://getbootstrap.com> (дата звернення: 5.05.2024).
11. PHP Documentation. URL: <https://www.php.net> (дата звернення: 5.05.2024).
12. Довідкова інформація по UML (Unified Modelling Language). URL: https://uk.wikipedia.org/wiki/Unified_Modeling_Language (дата звернення: 5.05.2024).
13. Додаток банку ПУМБ для здійснення керування валютними операціями. URL: <https://persona.pumb.ua/operatsii-s-valyutoy> (дата звернення: 5.05.2024).

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

14. Додаток MetaTrader Web Terminal. URL: <https://metatraderweb.app/trade> (дата звернення: 15.05.2024).

15. Додаток TradingView. URL: <https://www.tradingview.com/> (дата звернення: 15.05.2024).

16. Коротка довідка по mvc. URL: <https://brander.ua/technologies/mvc> (дата звернення: 15.05.2024).

17. Довідка по патерну Модель-вид-контролер. URL: <https://uk.wikipedia.org/wiki> (дата звернення: 15.05.2024).

18. Адміністративна панель додатку. URL: <https://voyager-docs.devdojo.com/getting-started/installation> (дата звернення: 15.05.2024).

19. Офіційний сайт НБУ для оновлення курсів валют. URL: https://bank.gov.ua/NBU_Exchange/exchange (дата звернення: 15.05.2024).

20. Автоматизуйте завдання за допомогою Laravel Cron: повний посібник. URL: <https://arjunamrutiya.medium.com/automate-tasks-with-laravel-cron-a-complete-guide-989e742a1d71> (дата звернення: 15.05.2024).

21. Як налаштувати cronjobs для проекту Laravel. URL: <https://dev.to/dreywandowski/how-to-setup-cronjobs-for-a-laravel-project-2li2> (дата звернення: 15.05.2024).

22. Як створити файли моделі/контролера/міграції/ресурсу/засівача за допомогою однієї команди в Laravel за допомогою php artisan. URL: https://medium.com/@chirag.viradiya_30404/how-to-generate-model-controller-migration-resource-seeder-files-with-single-command-in-laravel-0e3d13fab415 (дата звернення: 15.05.2024).

23. Що таке подвійна конвертація та як її уникнути. URL: <https://rates.fm/ua-uk/banks/sho-take-podvijna-konvertaciya-ta-yak-yiyi-uniknuti/> (дата звернення: 15.05.2024).

24. Фрімен Е., Робсон Е. Head First. Програмування на JavaScript. Легкий для сприйняття довідник: Харків, 2022. С. 672.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

25. Фленаган Д. JavaScript повне керівництво; довідник з найпопулярнішої мови програмування. Київ, 2021. С. 720.
26. Шеляг Г. Прикладна веб-розробка. HTML, CSS, JavaScript. Київ, 2020. С. 620.
27. Лобачевський В. PHP та MySQL. Розробка веб-застосунків: Для початківців. Харків, 2019. С. 432.
28. Петренко О., Сидоренко І. Laravel. Основи розробки веб-застосунків: Посібник для початківців. Київ, 2020. С. 368.
29. Ковальчук М., Іванов В. Архітектура веб-додатків на прикладі фреймворку Laravel. Вінниця, 2018. С. 256.
30. Чорна О., Іванов А. Моделювання програмного забезпечення за допомогою UML. Київ, 2021. С. 312.
31. Мельник П., Сидоров Д. Бази даних: Основи та практичні аспекти. Київ, 2017. С. 480.
32. Литвиненко В., Петров О. SQL для початківців: Повний курс вивчення. Київ, 2016. С. 368.
33. Кузнецов О., Шевченко С. Практичні аспекти роботи з PHP: Від початку до майстерності. Київ, 2015. С. 416.
34. Попов Д., Ігнатенко О. Розробка веб-додатків на фреймворку Laravel: Посібник з прикладами. Львів, 2014. С. 384.
35. Степаненко М., Коваленко С. Управління базами даних: Теорія та практика. Львів, 2013. С. 432.
36. Іваненко А., Григоров Д. SQL для професіоналів: Ефективні стратегії та практичні поради. Львів, 2012. С. 320.
37. Smith J., Brown R. Exploring the Power of PHP: A Comprehensive Overview. *Journal of Web Development*. 2023. Vol. 5. P. 112-128.
38. Patel A., Gupta S. Optimizing Laravel Performance: Strategies and Techniques. *International Journal of Software Engineering*. 2022. Vol. 12. P. 220-235.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

39. Wang X., Li Y. Understanding MVC Architecture in Web Development. *Software Design and Development Journal*. 2021. Vol. 8. P. 75-90.
40. Lee H., Kim S. Practical Applications of UML in Software Engineering. *Journal of Systems Architecture*. 2020. Vol. 15. P. 180-195.
41. Yang H., Chen G. Effective Use of PHP Frameworks: A Comparative Study. *Journal of Web Application Development*. 2017. Vol. 4. P. 90-105.
42. Laudon K. C., & Laudon, J. P. Management information systems: managing the digital firm. Pearson Education. 2018. Vol. 8. P. 2-10.
43. Kim Y., Park H. Exploring Laravel Eloquent ORM: Features and Performance. *Journal of Web Engineering*. 2016. Vol. 3. P. 110-125.
44. Adams R., Brown S. Model-View-Controller (MVC) Design Pattern: Implementation Strategies and Challenges. *Software Architecture Journal*. 2015. Vol. 6. P. 100-115.
45. Gretzel U., Sigala M., Xiang Z., Koo C. Smart tourism: foundations and developments (2nd edition). 2019. Vol. 25, No 3. P. 179-188.
46. Johnson D., Lewis M. Practical Applications of Laravel: Case Studies and Best Practices. *Journal of Web Development Practices*. 2020. Vol. 9. P. 200-215.
47. Kim J., Lee S. Optimizing UML Diagrams for Large-Scale Software Projects. *Journal of Software Design and Evolution*. 2019. Vol. 8. P. 170-185.
48. O'Connor P., Frew A. J., Morrison A. M. Revenue management and pricing analytics in the sharing economy: Research themes and future directions. *Journal of Revenue and Pricing Management*. 2018. Vol. 15, No 1. P. 270-283.
49. Li Q., Liu L. Effective Use of Design Patterns in PHP Development: Case Studies and Examples. *Journal of PHP Development*. 2015. Vol. 4. P. 230-245.
50. Rodriguez C., Martinez E. Practical Guide to Database Transactions: Techniques and Best Practices. *Transactions Processing Journal*. 2014. Vol. 3. P. 250-265.

					КВРІСТ 101039.21.01.02 ПЗ	Арк. 72
Зм.	Арк.	№ докум.	Підпис	Дата		

51. Mahdavi-Hezavehi M., Abrishami, A. та Jalali, M. (2016). Інженерія нефункціональних вимог: систематичний огляд. Журнал систем і програмного забезпечення. 2018. С. 273-293.

52. Мішкевич І.В. Інформаційні системи та технології. *Центр учбової літератури*. 2018. С. 34-42.

53. Evans S., Collins P., Murphy L. "A Framework for Evaluating Product Search Algorithms in E-commerce." *Journal of Computer Science and Technology*. 2018. Vol. 11, No. 2. P. 78-91.

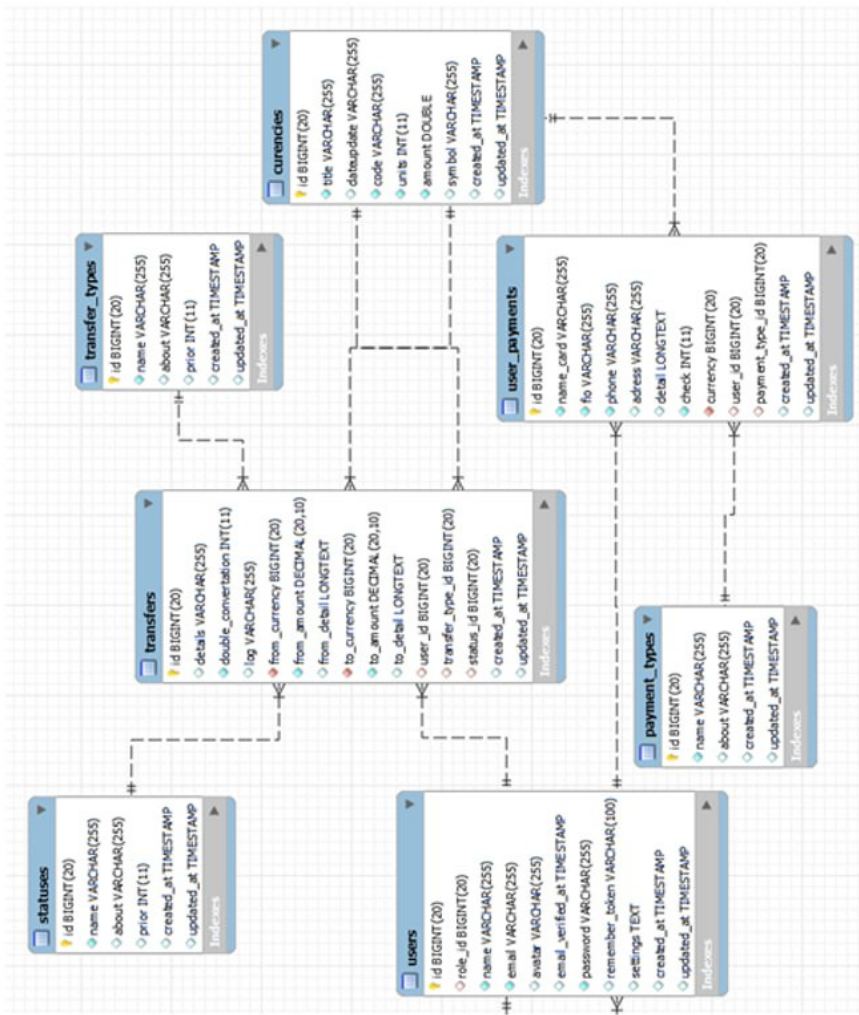
54. Бутенко С.В., Верес Ю.В., Гаврилова Т.В. Моделювання та проектування інформаційних систем. Видавничий дім "Ліра". 2016. С. 48-53.

55. Wallace L. G. та Hendricks, L. G. Каркас нефункціональних вимог до інформаційних систем. Журнал управління інформаційними технологіями. 2019. С. 16-30

					КВРІСТ 101039.21.01.02 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

ДОДАТОК А (обов'язковий)

Копія креслення «Схема бази даних»



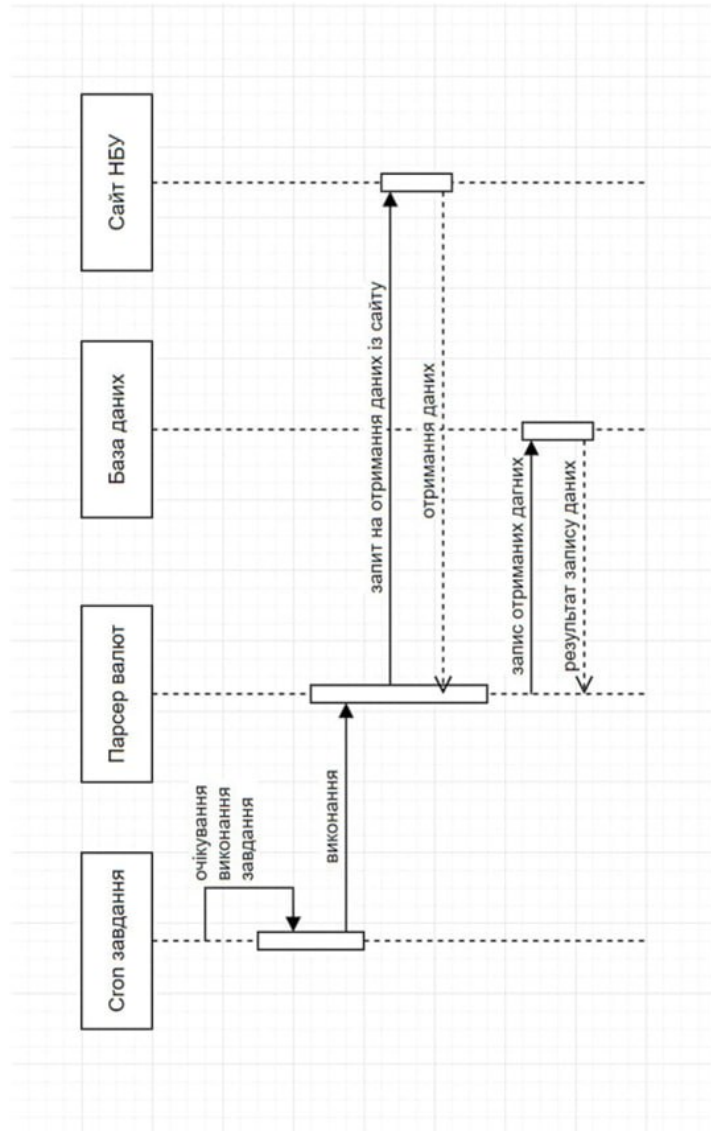
КВРІСТ 101039.21.01.02 ІЗ

КВРІСТ 101039.21.01.02 ІЗ	
Інформація стосовно управління валютними операціями	Литера
Структура бази даних	Маса
Зм. Акт. № докум. Підпис Дата	Масштаб
Розроб. Іванів В.	
Перевір. Іванів С.Г.	
Н. копії	Архив 1
Т. копії	Архив 3
Зам.	ХНУ, ПР.ІСТ-21-1

ДОДАТОК Б (обов'язковий)

Копія креслення «Діаграма послідовності дій»

Діаграма послідовності дій



КвРІСТ 101039.21.01.02 ПЗ

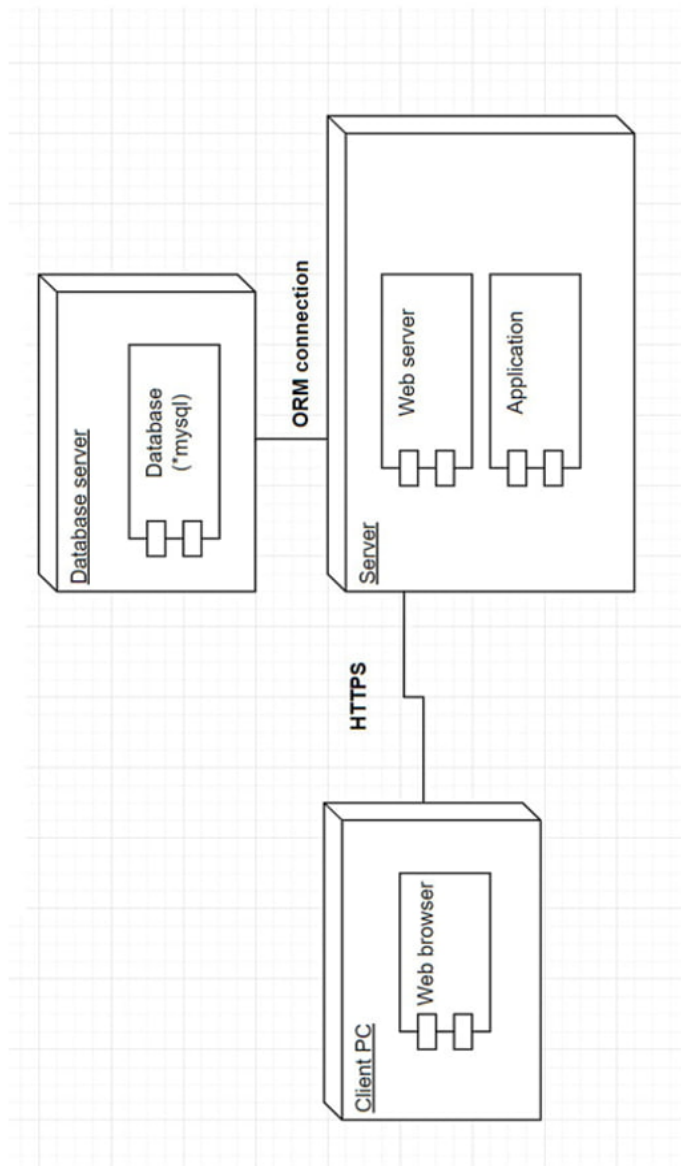
КвРІСТ 101039.21.01.02 ПЗ			
Літера	Місяц	Квартал	
Інформаційна система управління валютним операціям.			
Зам. Док.	№ докум.	Підпис	Дата
Сторг	Виконав		
Н. КОЛІС	В. КОЛІС		
Т. КОЛІС		В. КОЛІС	
Зам.		Виконав	
ХНУ, ГР ІСТ-21-1			

ДОДАТОК В

(обов'язковий)

Копія креслення «Діаграма розгортання»

Діаграма розгортання



КВРІСТ 101039.21.01.02 ІІЗ

КВРІСТ 101039.21.01.02 ІІЗ		Листопад	Місяць	Масштаб
Інформаційна система управління кваліфікацією персоналу				
Зм. Акт	№ докум.	Підпис	Дата	
Розроб.	Вправа В.В.			
Перевір.	Гавриш С.Г.			
Т. конгр.				
Т. конгр.	Закорюк І.О.			
Затв.	Петрушак Т.О.			
			Аркуш 1	Аркуш 3
			ХНУ, ГР.ІСТс-21-1	

Ім'я користувача:
Кафедра КІ
Дата перевірки:
04.06.2024 14:06:30 EEST
Дата звіту:
04.06.2024 17:17:57 EEST

ID перевірки:
1016319140
Тип перевірки:
Doc vs Internet + Library
ID користувача:
100005591

Назва документа: Зозуля_Інформаційна система управління валютними операціями

Кількість сторінок: 72 Кількість слів: 8541 Кількість символів: 70405 Розмір файлу: 8.37 MB ID файлу: 1016117190

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

8.34%
Схожість

Найбільша схожість: 1.84% з Інтернет-джерелом (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/13769/1/%d0%94>)

7.96% Джерела з Інтернету 521 Сторінка 74

3.17% Джерела з Бібліотеки 154 Сторінка 76

0.53% Цитат

Цитати 2 Сторінка 77

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 4

Підозріле форматування 22 сторінки

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 14%

ID: 128292 Назва: БКР Інформаційна система управління валютними операціями Додано в БД: 2024-06-04 Автора: В. В. Зогуля Керівники: С. Г. Гнатчук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	55913	592	1669 (3%)	28 (5%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Зозуля Владислав Валерійович

Тема: Інформаційна система управління валютними операціями

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: у роботі запропоновано комплексний вебдодаток, який надає користувачам зручні інструменти для ефективного управління валютними операціями. Проведено огляд та аналіз предметної області, на основі якого було поставлено задачі до кваліфікаційної роботи. В результаті аналітичного огляду було визначено недоліки та переваги подібних програм.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В розділах роботи проаналізовано основні функціональні та нефункціональні вимоги до програмного продукту, спроектовано структуру проекту, яка супроводжується UML діаграмою варіантів використання програмного засобу. Програмне забезпечення розроблене з використанням вебтехнологій, за основу взято фреймворк Laravel, програмна реалізація здійснювалась засобами мови програмування PHP, СУБД MySQL. Розробка програмної системи перебувала у всіх стадіях життєвого циклу створення програмного забезпечення. Спроектowano інтерфейс сайту, який забезпечує максимальну зручність та інтуїтивну зрозумілість у користуванні ним. Здійснено тестування програмної системи, за допомогою тестових випадків, що дали позитивні результати та підтвердити працеспроможність програмного рішення. Зазначені аспекти роблять інформаційну систему управління валютними операціями комплексним, функціональним та, потенційно, актуальним на ринку IT програмним продуктом.

4. Позитивні сторони роботи: Запропонований комплексний вебдодаток, який надає користувачам зручні інструменти для ефективного управління валютними операціями.

5. Негативні сторони роботи: В роботі не приділено достатньої уваги аналізу ефективності спроектованої системи, щоб дозволило порівняти результати роботи спроектованої системи з існуючими.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому професійному рівні

8. Інші зауваження: -

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)
доцент кафедри АКТ та Р Кофецька Л.О.

" 05 " 06 2024р.



Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорушенко Т. О.

Зозулі Владислава Валерійовича
ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи ІСТс-21-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5 червня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованою системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система управління валютними операціями
 Автор: Зозуля Владислав Валерійович
 Спеціальність: 126 – Інформаційні системи та технології
 Освітня програма: освітньо-професійна
 Науковий керівник: Гнатчук Є.Г., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та дорацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення, розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості Unicheck, складає 8.34% і адресується до 521 першоджерела; та системою Anti-Plagiarism складає 1%.

Керівник роботи
 Гарант ОП
 Завідувач кафедри КІС



Є. Г. Гнатчук
 Є. Г. Гнатчук
 Т. О. Говорушенко