

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Проектування та розробка мобільного застосунку “Exratedes” для прогнозування динаміки зміни курсу валют (комплексна робота)

Виконали: студенти IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Шимків В. С.

(прізвище та ініціали)

(підпис)

Ципняк Д. О.

(прізвище та ініціали)

Керівник

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Микитишин А.Г.

(прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«19» червня 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Шимківу Владиславу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та проектування мобільного застосунку "Exratedes" для прогнозування динаміки зміни курсу валют (комплексна робота)

Керівник роботи Никитюк Вячеслав Вячеславович, канд. техн. наук, доцент, доцент каф КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 20 червня 2023р.

3. Вихідні дані до роботи код для мобільного застосунку для динаміки іноземних валют. Розроблена база даних. Прототип дизайну мобільного застосунку.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Постановка завдання та формулювання вимог до мобільного застосунку "Exratedes". Аналіз предметної області. Формування переліку вимог до мобільного застосунку

Пошук актантів та опис варіантів використання. Середовище та засоби розробки мобільного застосунку. Формування архітектури мобільного застосунку "Exratedes". Розробка структури Мобільного застосунку. Висновок до першого розділу. 2 Проектування та реалізація мобільного застосунку "Exratedes". Опис основних структур для розробки мобільного застосунку. Аналіз, проектування та реалізація бази даних мобільного застосунку "Exratedes".

Створення макету дизайну мобільного застосунку "Exratedes". Реалізація мобільного застосунку.

3. Безпека життєдіяльності, основи охорони праці. Долікарська допомога при ураженні електричним струмом. Загальні вимоги безпеки з охорони праці для користувачів ПК. Висновок до третього розділу. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

№1 Аналіз предметної області. №2 Пошук актантів та опис варіантів використання.

№4 Основні структури для розробки мобільного застосунку. №5 Проектування та реалізація бази даних. №6 Створення прототипу макету дизайну мобільного застосунку. №7 Прототип дизайну мобільного застосунку "Exratedes". №7 Реалізація мобільного застосунку Backend.

№9 Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ	05.06.2023	08.06.2023

7. Дата видачі завдання 23 січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	23.01.2023	<i>Виконано</i>
2.	Підбір джерел про розробку програмних застосунків	24.01.2023-26.01.2023	<i>Виконано</i>
3.	Опрацювання джерел по темі кваліфікаційної роботи	27.01.2023-31.01.2023	<i>Виконано</i>
4.	Виконання дослідження щодо динаміки курсу іноземних валют	01.02.2023-07.02.2023	<i>Виконано</i>
	Розроблення мобільного застосунку “Exratedes”		
5.	Оформлення розділу «Постановка завдання та формування вимог до мобільного застосунку»	08.02.2023-09.02.2023	<i>Виконано</i>
6.	Оформлення розділу «Проектування та реалізація мобільного застосунку “Exratedes”»	10.02.2023-12.02.2023	<i>Виконано</i>
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	05.06.2023-06.06.2023	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Основи охорони праці»	07.06.2023-08.06.2023	<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи	09.06.2023-11.06.2023	<i>Виконано</i>
10.	Нормоконтроль	12.06.2023-13.06.2023	<i>Виконано</i>
11.	Перевірка на плагіат	14.06.2023	<i>Виконано</i>
12.	Попередній захист кваліфікаційної роботи	15.06.2023	<i>Виконано</i>
13.	Захист кваліфікаційної роботи	20.06.2023	

Студент

(підпис)

Шимків В.С.

(прізвище та ініціали)

Керівник роботи

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«19» червня 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ципняку Денису Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та проектування мобільного застосунку "Exratedes" для прогнозування динаміки зміни курсу валют (комплексна робота)

Керівник роботи Никитюк Вячеслав Вячеславович, канд. техн. наук, доцент, доцент каф КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 20 червня 2023р.

3. Вихідні дані до роботи код для мобільного застосунку для динаміки іноземних валют. Реалізація мобільного застосунку.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Постановка завдання та формулювання вимог до мобільного застосунку "Exratedes". Аналіз предметної області. Формування переліку вимог до мобільного застосунку

Пошук актантів та опис варіантів використання. Середовище та засоби розробки мобільного застосунку. Формування архітектури мобільного застосунку "Exratedes". Розробка структури Мобільного застосунку. Висновок до першого розділу. 2 Проектування та реалізація мобільного застосунку "Exratedes". Опис основних структур для розробки мобільного застосунку. Аналіз, проектування та реалізація бази даних мобільного застосунку "Exratedes".

Створення макету дизайну мобільного застосунку "Exratedes". Реалізація мобільного застосунку.

3. Безпека життєдіяльності, основи хорони праці. Вплив виробничого середовища на працездатність та здоров'я користувачів комп'ютерів. Методи боротьби з монотонністю праці на виробництві. Висновок до третього розділу. Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

№1 Аналіз предметної області. №2 Пошук актантів та опис варіантів використання.

№3 Проектування та реалізація мобільного застосунку. №8 Реалізація мобільного застосунку Frontend. №9 Висновки.

АНОТАЦІЯ

Розробка та проектування мобільного застосунку “Exratedes” для прогнозування динаміки зміни курсу валют (комплексна робота) // Кваліфікаційна робота освітнього рівня «Бакалавр» // Шимків Владислав Сергійович, Ципняк Денис Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СН-41 // Тернопіль, 2023 // С. – 65, рис. – 24, табл. – 2, кресл. – 11, додат. – 2, бібліогр. – 30.

Ключові слова: курс, дизайн, валюта, програмування, база даних, безпека, застосунок, реалізація.

Кваліфікаційна робота присв’ячена розробці мобільного застосунку для динаміки іноземних валют. В першому розділі кваліфікаційної роботи описано постановку завдання та формулювання вимог до мобільного застосунку. Висвітлено аналіз предметної області. Розглянуто формування переліку вимог до мобільного застосунку. Проаналізовано застосунок. Формування архітектури мобільного застосунку “Exratedes” та розробка структури мобільного застосунку.

В другому розділі кваліфікаційної роботи Проектування та реалізація мобільного застосунку “Exratedes” Досліджено Опис основних структур для розробки мобільного застосунку Подано Аналіз, проектування та реалізація бази даних мобільного застосунку “Exratedes”. Реалізація мобільного застосунку Створення макету дизайну мобільного застосунку “Exratedes”.

В третьому розділі кваліфікаційної роботи описано безпека життєдіяльності, основи хорони праці.

ANNOTATION

Design and Development of Mobile Application "Exredades" for Dynamics Prediction of the Currency Exchange Rate Alteration // Qualification work of the educational level "Bachelor" // Shymkiv Vladyslav, Tsypniak Denis // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-41 // Ternopil, 2023 // P. - 65 , fig. - 24 , tabl. - 2, chair. - 11, annexes. – 2, references - 30.

Keywords: course, design, currency, programming, database, security, application, implementation.

The qualification work is devoted to the development of a mobile application for the dynamics of foreign currencies. The first section of the qualification work describes the setting of the task and the formulation of requirements for the mobile application. The analysis of the subject area is covered. The formation of the list of requirements for the mobile application is considered. Formation of the architecture of the mobile application "Exratedes" and development of the structure of the mobile application.

In the second section of the qualification work Design and implementation of the mobile application "Exratedes" The description of the main structures for the development of the mobile application was studied. The analysis, design and implementation of the database of the mobile application "Exratedes" was presented. Implementation of the mobile application Creation of the design layout of the mobile application "Exratedes".

In the third section of the qualification work, the safety of life activities, the basics of labor protection are described.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ

MIT (англ. Massachusetts Institute of Technology) – Массачусетський технологічний інститут.

БД – база даних

ОС – операційна система

ВТ – валюти

СКБД – система керування базами даних

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ ТА ФОРМУЛЮВАННЯ ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ	11
1.1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.2 ФОРМУВАННЯ ПЕРЕЛІКУ ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ	13
1.3 ПОШУК АКТАНТІВ ТА ОПИС ВАРІАНТІВ ВИКОРИСТАННЯ	14
1.4 ФОРМУВАННЯ АРХІТЕКТУРИ ЗАСТОСУНКУ “EXRATEDES”	16
1.5 РОЗРОБКА СТРУКТУРИ ЗАСТОСУНКУ “EXRATEDES”.....	17
1.6 ВИСНОВОК ДО ПЕРШОГО РОЗДІЛУ.....	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ “EXRATEDES”	20
2.1 ОПИС ОСНОВНИХ СТРУКТУР ДЛЯ РОЗРОБКИ МОБІЛЬНОГО ЗАСТОСУНКУ.	20
2.2 АНАЛІЗ, ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ МОБІЛЬНОГО ЗАСТОСУНОК “EXRATEDES”.....	22
2.3 СТВОРЕННЯ ПРОТОТИПУ МАКЕТУ ДИЗАЙНУ МОБІЛЬНОГО ЗАСТОСУНОК “EXRATEDES”	30
2.4 РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ	35
2.5 ВИСНОВОК ДО ДРУГОГО РОЗДІЛУ	46
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	47
3.1 ДОЛІКАРСЬКА ДОПОМОГА ПРИ УРАЖЕННІ ЕЛЕКТРИЧНИМ СТРУМОМ	47
3.2 ЗАГАЛЬНІ ВИМОГИ БЕЗПЕКИ З ОХОРОНИ ПРАЦІ ДЛЯ КОРИСТУВАЧІВ ПК.....	50

3.3 МЕТОДИ БОРОТЬБИ З МОНОТОННІСТЮ ПРАЦІ НА ВИРОБНИЦТВІ	52
3.4 ВПЛИВ ВИРОБНИЧОГО СЕРЕДОВИЩА НА ПРАЦЕЗДАТНІСТЬ ТА ЗДОРОВ'Я КОРИСТУВАЧІВ КОМП'ЮТЕРІВ ...	55
3.5 ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ	58
ВИСНОВОК	60
ПЕРЕЛІК ДЖЕРЕЛ	61
ДОДАТОК А	65
ДОДАТОК Б	81

ВСТУП

У сучасному світі, де глобалізація є неодмінною частиною нашого повсякденного життя, фінансові ринки стають все більш доступними та складними. Зміна валютних курсів може має значний вплив на економіку країни, бізнес-операції, подорожі та індивідуальний фінансовий стан людей. Тому маючи актуальну та точну інформацію про динаміку валют, користувачі здатні приймати обґрунтовані рішення та виконувати фінансові операції з найбільшою вигодою.

Метою цієї дипломної роботи є розробка та проектування мобільного застосунку, який забезпечує користувачам легкий доступ до актуальних даних про курси іноземних валют. Наш додаток буде забезпечувати широкий спектр функціональних можливостей, таких як відображення актуальних курсів в режимі реального часу, аналітика динаміки валют, сповіщення про зміни курсів, конвертація валют та інше.

Для досягнення поставленої мети, ми будемо проводити аналіз сучасних мобільних додатків фінансової сфери та валютних сервісів, вивчати ринкові тенденції, а також залучати методи та інструменти розробки мобільних додатків. Ми розглянемо різні аспекти проектування та розробки, такі як інтерфейс користувача, база даних, алгоритми оновлення та аналізу даних, забезпечення безпеки та захисту інформації.

В результаті нашої роботи, ми очікуємо отримати функціональний та ефективний мобільний застосунок, який дозволить користувачам з легкістю отримувати та аналізувати актуальну інформацію про динаміку іноземних валют. Наша розробка може бути корисною для широкого спектру користувачів, від індивідуальних фінансових учасників до бізнес-сектору та подорожуючих.

В цій дипломній роботі ми плануємо представити детальний опис розробки застосунку, описати використані технології та методи, а також провести оцінку та аналіз отриманих результатів. Ми впевнені, що наша робота

стане важливим внеском у сферу фінансових технологій та сприятиме поліпшенню фінансової грамотності та прийняттю обґрунтованих фінансових рішень.

З нетерпінням чекаємо на результати нашої роботи та сподіваємося, що наш мобільний застосунок знайде практичне застосування та буде корисним для багатьох користувачів.

Авторський внесок Шимківа Владислава: Вступ, Розділ 1 Постановка та формулювання вимог до мобільного застосунку “Exratedes”, п. 1.1 Аналіз предметної області, п. 1.2 Формування переліку вимог до мобільного застосунку, п. 1.3 Пошук актантів та опис варіантів використання, 1.4 Формування архітектури мобільного застосунку, п. 1.6 Висновок до першого розділу, Розділ 2 Проектування та реалізація мобільного застосунку “Exratedes”, п. 2.1 Опис основних структур для розробки мобільного застосунку, п. 2.2 Аналіз проектування та реалізація бази даних мобільного застосунку “Exratedes”, п. 2.3 Створення макету прототипу дизайну мобільного застосунку “Exratedes”, Розділ 3 Безпека життєдіяльності та охорони праці, п. 3.1 Долікарська допомога при ураженні електричним струмом, п. 3.2 Загальні вимоги безпеки охорони праці для користувачів ПК, п. 3.5 Висновок до третього розділу, Перелік лжерел.

Авторський внесок Ципняка Дениса: Розділ 1 Постановка та формулювання вимог до мобільного застосунку “Exratedes”, п. 1.5 Розробка структури мобільного застосунку, Розділ 2 Проектування та реалізація мобільного застосунку “Exratedes”, п. 2.4 Реалізація мобільного застосунку. Розділ 3 Безпека життєдіяльності та охорони праці, п. 3.3 Вплив виробничого середовища на працевдатність та здоров'я користувачів комп'ютерів, п. 3.4 Методи боротьби з монотонністю на виробництві.

РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ ТА ФОРМУЛЮВАННЯ ВИМОГ ДО МОБІЛЬНОГО ЗАСТОСУНКУ

1.1 Аналіз предметної області

Метою кваліфікаційної роботи є розроблення рекомендаційного мобільного застосунок з інтерактивними можливостями для вибору іноземних валют. Який на основі аналізу відомостей про стан ВТ, зможе формувати графіки динаміки курсу валют та надаватиме базові відомості про ВТ. Мобільний застосунок “Exratedes” займається збором інформації, аналізом, та візуалізацією валют. Тому було прийняте рішення про розробку мобільну програму на операційній системі Android, який на основі аналізу інформації з відкритих ресурсів, привертатиме увагу користувача, та надаватиме відповідні рекомендації та пропозиції щодо вибору валют[1].

Сформулюємо перелік функціональних вимог для коректної роботи застосун-ку. Прицьому визначимо список функцій, що повинні бути реалізовані:

- Надання онлайн-доступу до мобільного застосунок з використання АРК формату програми;
- Надання користувачам мобільного застосунок “Exratedes” можливості інтерактивного перегляду ;
- Надання бібліотеки валют;
- Оперативне зберігання відомостей для БД;
- Інтеркативне формування системи побудови графіків;
- Адаптивний список рекомендацій із актуальними парами валют;
- Адаптивний список із нещодавно доданими валютами;
- Формування графічного профілю для користувача на основі “Головна”, “Бібліотека валют”, “Дізнатись курс”, “Налаштування”, “Інформація про розробників”[1].

Після аналізу та формулювання вимог для роботи програми, переходиться до визначення основних сутностей. До основних сутностей можна віднести користувач, валюта, головга, бібліотека валют, курс, про розробників, налаштування.

Сутність “Користувач” може:

- Переглядати бібліотеку валют;
- Здійснювати переходи по сторінкам;
- Переглядати адаптивний список рекомендацій з актуальними валютами;
- Переглядати адаптивний список нещодавно доданих валют;
- Проводити дію із взиванням курсу валют та її динаміки відносно іншої;
- Переглядати відомості про ВТ;

Сутність “Валюта” містить основні відомості щодо іноземних валют:

- Назва;
- Символ;
- Код;
- Короткий опис;

Сутності “Головна”, “Бібліотека валют”, “Дізнатись курс”, “Налаштування”, “Інформація про розробників” пов'язані між собою, адже дають безпосередню можливість навігації в застосунок. Знайдені в сутності “Головна” нещодавно додані ВТ та популярні пари зв'язані з “Бібліотека валют” та “Дізнатись курс”. Сутність “Налаштування” та “Інформація про розробників” стоять убособлено оскільки надають окрему доцільну інформацію.

Сутність “Дізнатись курс” працює в парі з популярні пари. Принцип роботи простий, а саме взята інформація з популярних пар автоматично передається в “Дізнатись” та проходить боробка запиту з подальши наданням усієї потрібної інформації.

1.2 Формування переліку вимог до мобільного застосунку

Після аналізу предметної області та визначення ключових сутностей форму-ється перелік вимог до мобільного застосунок[2]:

- Функціональні вимоги;
- Вимоги щодо користувачів;
- Вимоги до графічного інтерфейсу;
- Вимоги до структури мобільної програми “Exratedes”;

Перелік функціональних вимог:

- Фільтрація списку валют відповідно обраних критеріїв;
- Перегляд окремих сторінок валют та ознайомлення з детальною інформацією про них;

- Перегляд вкладки налаштування, головга, ВТ, курс, розробники;

Перелік вимог для користувача:

- Зручність використання;
- Надання рекомендацій та пропозицій на основі новітніх тенденцій;

Перелік вимог до інтерфейсу застосунку:

- Адаптивність до зміни розмірів та роздільної здатності екрану;
- Динамічне завантаження даних для структурних елементів за допомогою AJAX/JSON;

- Кольорова схема інтерфейсу повинна бути достатньо контрасною;
- Зручність та простота користування;

Перелік вимог щодо структури мобільного застосунок:

- Компоненти мобільного застосунок повинні бути реалізовані у вигляді окремих програмних файлів;

- Програмні файли для виконання реалізації адміністративних функцій, повинні міститися в окремому каталозі сервера, не пов'язаному з звичайного користувача;

- Файли із зображеннями повинні бути розміщені в окремому каталозі серверу;

- Службові файли програмних компонентів мають бути розміщені в окремому каталозі серверу;

1.3 Пошук актантів та опис варіантів використання

Для спроектованого мобільного застосунку визначимо наступних акторів[1] дивитись рисунок 1.1):

- Користувач;
- Незареєстрований користувач;
- Адміністратор;
- Застосунок;

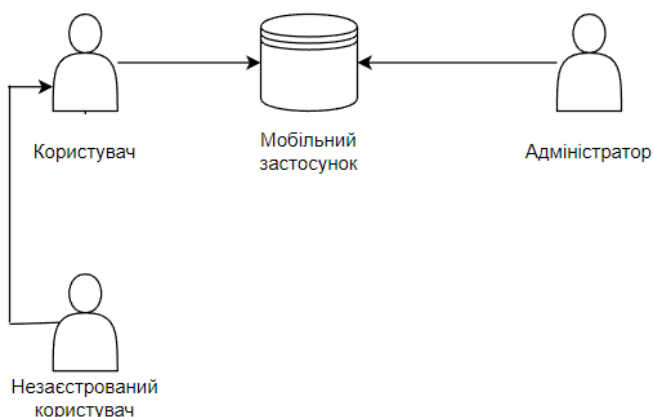


Рисунок 1.1 — Актори застосунку

Після аналізу списку функціональних вимог, виділимо ключові варіанти використання мобільного застосунку “Exratedes”. Їхній перелік та короткі описи подаються в таблицях- 1.3.1 – 1.3.2. До них можна віднести незареєстрованого користувача та користувача. Функції які може виконувати незареєстрований користувач: перегляд бібліотек, візуалізація курсу, перегляд сторінки ВТ. Функції які може виконувати користувач все що і незареєстрований користувач і навіть більше.

Таблиця 1.3.1 Реєстр варіантів використання для актора
“Незареєстрований користувач”

Актор	Варіанти використання	опис
Незареєстрований користувач	Перегляд бібліотеки валют	Надає акторові можливість переглядати бібліотеку валют
	Візуалізація курсу	Дозволяє дослідити динаміку іноземної ВТ
	Відображення сторінки ВТ	Надає акторові можливість переглядати розширені відомості окремої ВТ

Варіанти використання для актора “Користувач” сформульовано на основі варіантів використання для актора “Незареєстрований користувач” з додаванням нових, які подані для ознайомлення в таблиці 1.3.2.

Таблиця 1.3.2 Реєстр варіантів використання для актора “Користувач”

Актор	Варіанти Використання	Опис
Користувач	Перегляд головної сторінки	Надає можливість для ознайомлення з нещодавно доданими ВТми та популярними парами
	Перегляд бібліотеки валют	Надає акторові можливість переглядати бібліотеку валют

Продовження Таблиця 1.3.2 Реєстр варіантів використанні для актора
“Користувач”

	Візуалізація курсу	Дозволяє дослідити динаміку іноземної ВТ
	Відображення сторінки ВТ	Надає акторові можливість переглядати розширені відомості окремої ВТ
	Прегляд сторінки налаштування	Надає можливість акторові детально налаштовувати додаток
	Перегляд сторінки про розробників	Надає можливість користувачу ознайомитись з розробниками та зв'язатись з ними

На основі поданих таблиць, було наведено коротка та в одночас вичерпна інформація про розподілення прав акторів та їх різницю. Для актора “Незарєєстрований користувач” доступні варіанти використання перегляд бібліотеки валют, візкалізація курсу, відображення сторінки ВТ.

Для актора “Користквач” доступні перегляд головної сторінки, перегляд бібліотеки валют, візуалізація курсу, відображення сторінки ВТ, перегляд сторінки налаштування, перегляд сторінки про розробників.

1.4 Формування архітектури застосунку “Exratedes”

Відповідно до поданих вище результатів аналізу предметної обслатсті та описів варіантів використання, найбільш ефективною для проектуваного

застосунку “Exratedes” буде тривимірною архітектурою, що містить рівні актора, структурно логічний та даних із яким можна ознайомитись на рисунку 1.2 [4].

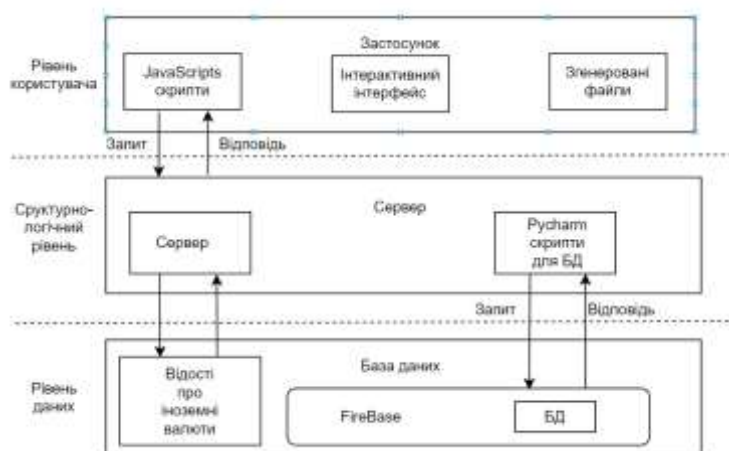


Рисунок 1.2 — Архітектура застосунку “Exratedes”

На рівні актора розміщено графічний інтерфейс, що відображається для кінцевого користувача. На структурно логічному рівні розташовано сервер застосунків для забезпечення функціонування програмних компонентів. Додатково на цьому рівні зосереджено ряд сервісних функціональних засобів, зокрема службові модулі, бібліотеки.

На структурно-логічному рівні знаходяться програмно-алгоритмічні Python та Java-скриптів для взаємодії з користувачем, формуються обробки отриманих від користувача даних.

На рівні даних представлено сервер БД, який забезпечує виконання процедур пошуку, отримання та зберігання даних. На цьому рівні розміщена СКБД Firebase, яка виконує функції зберігання даних та операції щодо їх змісту.

1.5 Розробка структури застосунку “Exratedes”

Структура застосунку “Exratedes” для ознайомлення надано рисунок 1.3 умовно поділена на три частини “Front-end”, “Back-end”, “База даних”[5].

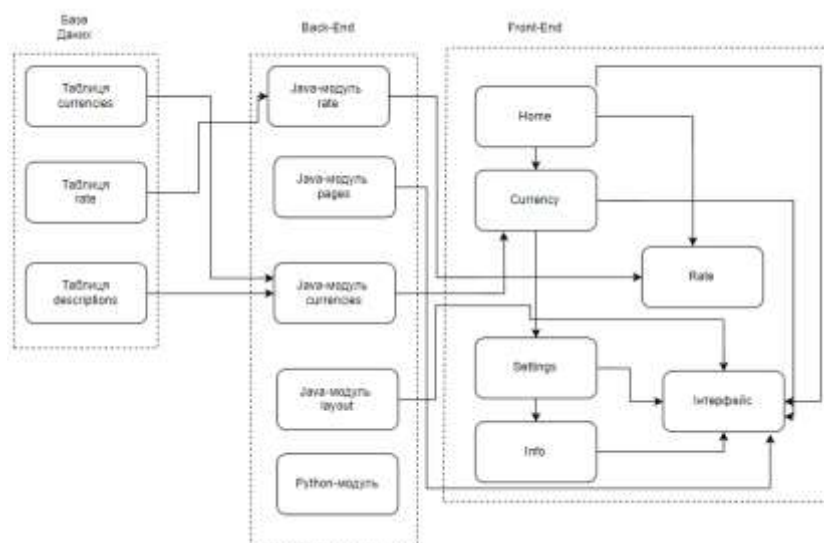


Рисунок 1.3 — Структура застосунку “Exratedes”

Група “Front-end” відповідає за відображення всіх функціональних елементів з якими взаємодіє користувач при роботі з застосунком.

Файли з частини “Back-end” відповідають за генерацію структурного зв'язку з базою даних та виконання обчислень, необхідних для роботи застосунку.

1.6 Висновок до першого розділу

В першому розділі кваліфікаційної роботи проведено аналіз предметної області мобільного застосунку. Розробляється для операційної системи Android та використовує аналіз інформації з відкритих ресурсів для застосунків, таких як надання онлайн-доступу, інтерактивний перегляд валют, зберігання відомостей у базі даних, адаптивний список рекомендацій та інші. Окремі сутності, які використовуються в застосунк, включають користувача, валюту, головну сторінку, бібліотеку валют, курси, налаштування та інформацію про розробників.

Після аналізу предметної області і визначення ключових сутностей формується перелік вимог до мобільного застосунків, включаючи функціональні

вимоги, вимоги щодо користувачів, вимоги до графічного інтерфейсу та вимоги до структури програми.

Також враховувались актори, такі як користувач, незареєстрований користувач, адміністратор та сам додаток, які взаємодіють у мобільному застосунку.

Отже, на основі першого розділу можна зробити висновок, що метою кваліфікаційної роботи є розробка рекомендаційного мобільного застосунок для вибору іноземних валют, який надає користувачам інтерактивні можливості та інформацію про ВТ. Додаток має задовольняти перелік функціональних вимог та вимог до інтерфейсу, а взаємодіють у ньому різні актори.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ “EXRATEDES”

2.1 Опис основних структур для розробки мобільного застосунку.

Для створення основних функцій мобільного застосунку використовувались такі компоненти:

- Google Firebase;
- IDE PyCharm Community;
- Android Studio;
- Редактор коду Visual Studio Code;
- Ехро GO
- Бібліотека React-Native;
- Створення макету інтерфейсу для мобільного застосунку Adobe

Photoshop 2018;

- Gitlab

Google Firebase

При розробці використовувалась Firebase 9.22.1 версія для мови JavaScript, AdMod 22.1.0 для андроїдної системи, AdminPython 6.1.0, AdminJava 9.1.1 . Більш детальніша інформація викладена в розділі про 2.2 Аналіз, проектування та реалізація БД мобільного застосунку “Exratedes”[6].

PyCharm Community

Для виконання кваліфікаційної роботи самим застосунком PyCharm 2022.3.3 (Community Edition) створено файл requirements.txt де містяться версії додаткових модулів anyio 3.6.2, certifi 2022.12.7, charsetnormalizer 3.1.0, click 8.1.3, colorama 0.4.6, fastapi 0.95.0, firebase 4.0.1[7].

Наступна група модулів h11 0.14.0, idna 3.4, pydantic 1.10.6, requests 2.29., sniffio 1.3.0, starlette 0.26.1, typing_ex, tensions 4.5.0, urllib3 1.26.15, uvicorn 0.21.1.

Visual Studio Code

Visual Studio Code (VS Code) – безкоштовний редактор коду, створений компанією Microsoft. VS Code підтримує різні мови програмування, такі як C++, Python, Java, JavaScript, TypeScript та багато інших, та дозволяє розробникам працювати з різними типами проектів[8]. При розробці прототипу використовувались дані характеристики програми:

- Version: 1.78.2 (user setup)
- Commit: b3e4e68a0bc097f0ae7907b217c1119af9e03435
- Date: 2023-05-10T14:39:26.248Z
- Electron: 22.5.2
- Chromium: 108.0.5359.215
- Node.js: 16.17.1
- V8: 10.8.168.25-electron.0
- OS: Windows_NT x64 10.0.19045
- Sandboxed: Yes

Бібліотека React-Native

Відомості про бібліотеки React-Native для мобільного застосунку[9]:

- react-native: 0.71.7;
- react-native-chart-kit: 6.12.0;
- react-native-select-dropdown: 3.3.2;
- react-native-svg: 13.4.0;
- react-native-web: 0.18.11;
- react-router-native: 6.10.0;

Android Studio

При розробці вибрано Android Studio версією Android Studio Electric Eel | 2022.1.1 Patch 2. [10]. Основною перевагою є детальне налаштування пристрою від розмірів екрану до версії операційної системи Android. ОП Android використовувалась від Android 9 Pie до Android 12. З основних фірм постачальників відібрано Samsung, Google, Xiaomi.

2.2 Аналіз, проектування та реалізація бази даних мобільного застосунок “Exratedes”.

Проектування та реалізація БД була проведена у редакторі FireBase.

Перед початком створення БД, було проведено аналіз основних вимог до її структури і параметрів, а саме:

Код ВТ – це трьома літерами (наприклад, USD для долара США, EUR для євро, GBP для фунта стерлінгів тощо).

Повна назва ВТ – наприклад, “Долар США”, “Євро”, “Фунт стерлінгів” тощо.

Символ ВТ – символ, що використовується для позначення ВТ (наприклад, \$ для долара США, € для євро, £ для фунта стерлінгів тощо).

Країна або регіон, де використовується ця валюта.

Курс валют – ставка обміну однієї ВТ на іншу.

Також була проведена вибірка з основних валют світу, які використовуються безпосередньо для міжнародних транзакцій та торгівлі на світових ринках. Нище наведено список із двадцяти однієї однієї ВТ, які були вибрані та використані у даній базі даних мобільного застосунок.

Список валют для БД:Українська Гривня, Долар США ,Євро ,Японська ена, Фунт стерлінгів Великої Британії, Швейцарський франк, Канадський долар, Австралійський долар, Новозеландський долар, Китайський юань, Гонконгський долар, Сінгапурський долар, Шведська крона, Норвезька крона, Південнокорейський вон, Болгарський лев, Хорватська куна, Чеська крона, Данська крона, Венгерський форинт, Польський злотий, Румунський лей. Поданий список валют є актуальним на період дві тисячі двадцять третього року.

Безпосередньо перед початком створення БД передувало створення відповідного серверу Exratedes. У даному сервері проводилась основна робота над створенням БД “Exratedes”. Дивитись рисунок 2.1 створення проекту БД.

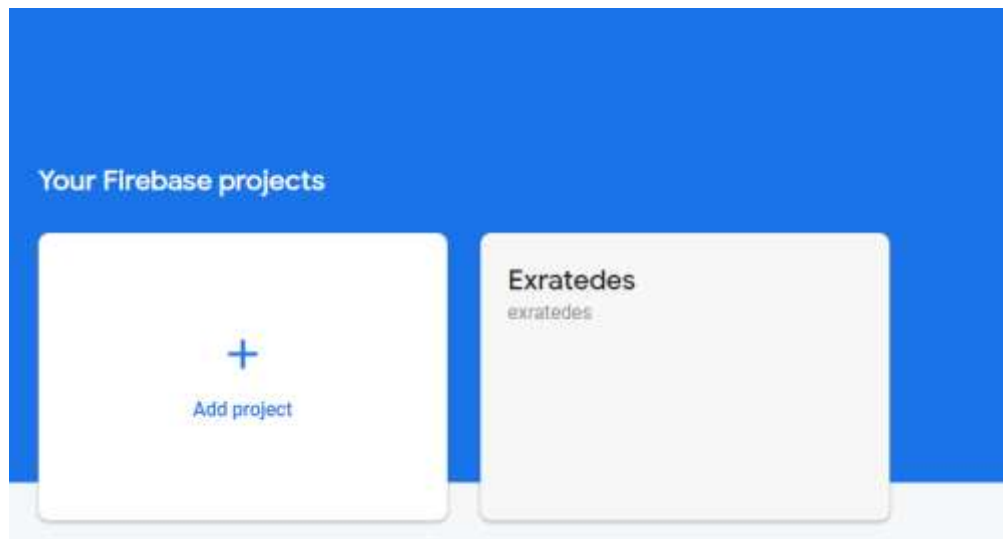


Рисунок 2.1 — Створення проекту БД

Увійшовши в проект “Exratededes” firebase надає поточну інформацію про базу даних. Проект інформує про загруженість БД, її вагу загрузки та вагу зберігання. Дивитись рисунок 2.2 — основні відомості про проект.

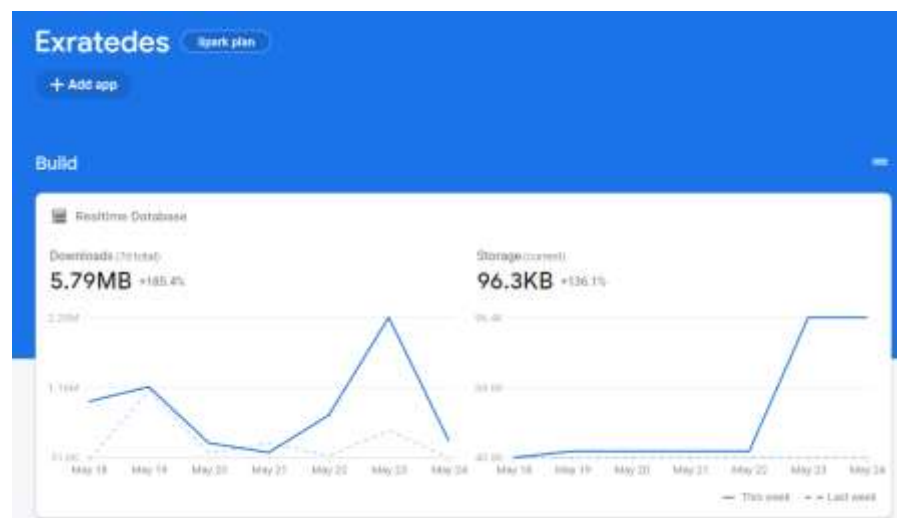


Рисунок 2.2 — Основні відомості про проект.

Для подальшої роботи з базою потрібно перейти на меню проекту, яке знаходиться у лівій частині екрану. Для безпосередньої роботи з базою даних потрібно вибрати меню “Realtime Database”. Дивитись на рисунок 2.3 — Realtime Database.

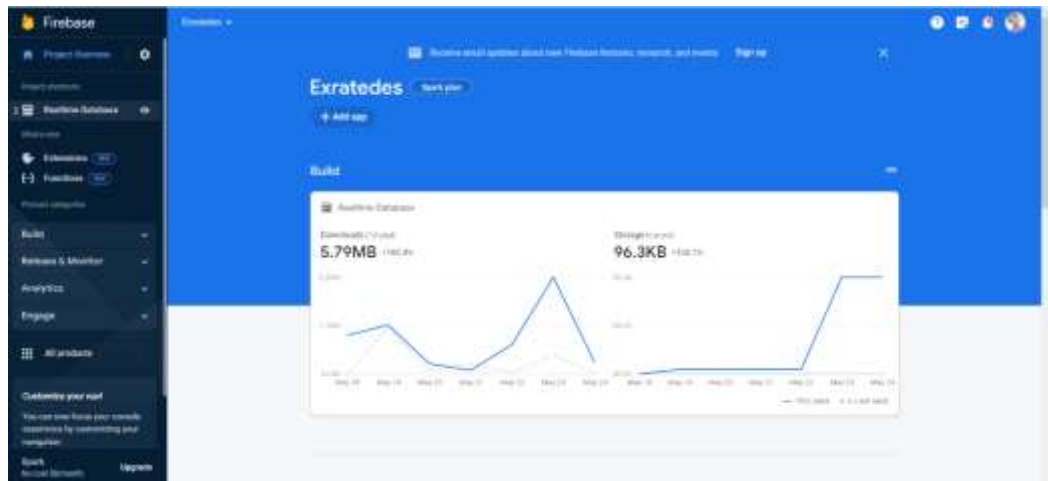


Рисунок 2.3 — Realtime Database.

В меню “Realtime Database” надає інформацію про базу даних проекту, надає можливість редагувати її безпосередньо в графічному меню, або у форматі файлів json. База даних складається з трьох основних груп даних: currencies (ВТТ), descriptions (опис), rate (курс). Дивитись рисунок 2.4 — база даних.

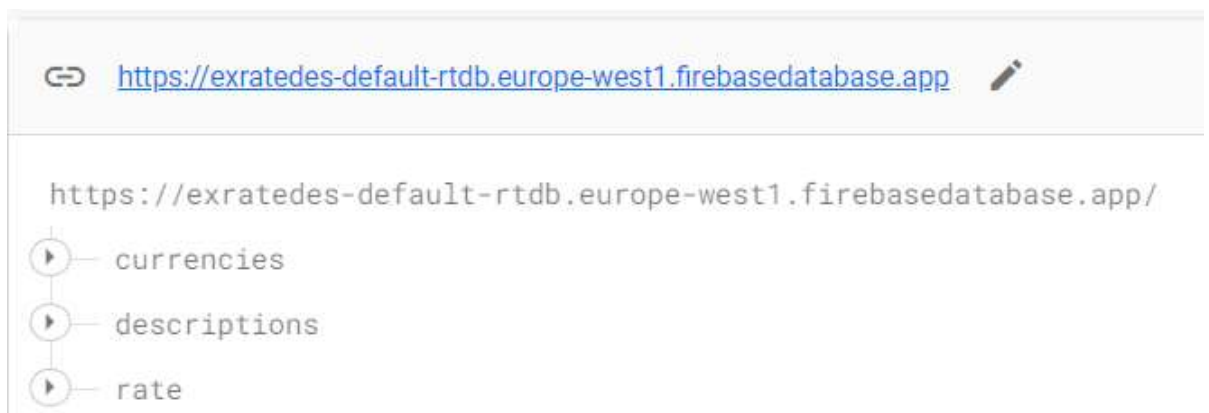


Рисунок 2.4 — База даних.

До розгляду надано перша група даних currencies (ВТТ). Дана група вміщає основні відомості валют: назва, регіон, символ. Для прикладу надається до ознайомлення рисунок 2.5 — відомості currencies для ВТ Українська Гривня. Аналогічні дані надаються і на інші ВТ.



Рисунок 2.5 — Відомості currencies для VT Українська Гривня.

Друга на розгляді група descriptions (опис). Дана група володіє описом кожної окремої VT. Дані були взяті з відкритих джерел, тому інформація по суті відривна та легка для сприйняття. Для прикладу надається до ознайомлення рисунок 2.6 — опис VT Українська Гривня.

twb: TWD (Новий тайванський долар) є офіційною валютою Тайваню. Долар використовується як засіб платежу в країні. Валюта поділяється на 100 центів.
 uan: Гривня (₴) є офіційною валютою України. Вона введена в обіг у 1996 році після отримання незалежності країною. Гривня поділяється на 100 копійок. К.
 usd: USD (Долар США) є офіційною валютою Сполучених Штатів Америки. Долар є однією з найбільш використовуваних валют у світі. Валюта поділяється +

Рисунок 2.6 — Опис VT Українська Гривня

Третя на розгляді група rate (курс). Дана група володіє описом курсу кожної окремої VT. Дані були взяті з відкритого джерела, а саме сайту Minfin[11]. Додатково вказується що дані курсу взяті від двітися двадцятого року. Для прикладу надається до ознайомлення рисунок 2.7 — курс VT Українська Гривня 2020-2021, рисунок 2.8 — курс VT Українська Гривня 2022-2023. Для кращої роботи програми по відображенню даних, було прийняте рішення про вибір значень із першої дати кожного місяця. Цим кроком було проведено оптимізацію даних, яке в свою чергу посприяло зниженню ваги кількості даних яких пропускає БД.



Рисоунок 2.7 — курс ВТ Українська Гривня 2020-2021

Дана база даних була створена за допомогою окремого програмного коду парсеру. Основною функцією парсингово коду є його безпосереднє зчитування з сайту даних та раписування їх у файл під розширенням json. Заповнені файли пересилаються в базу даних.

2022-01-01: 1
 2022-02-01: 1
 2022-03-01: 1
 2022-04-01: 1
 2022-05-01: 1
 2022-06-01: 1
 2022-07-01: 1
 2022-08-01: 1 2023-01-01: 1
 2022-09-01: 1 2023-02-01: 1
 2022-10-01: 1 2023-03-01: 1
 2022-11-01: 1 2023-04-01: 1
 2022-12-01: 1 2023-05-01: 1

Рисоунок 2.8 — курс ВТ Українська Гривня 2022-2023

Структура міні-програми складається з тьох файлів main.py, test.py, vercel.json.

Перший файл `main.py` цей код є прикладом веб-застосунок, який використовує фреймворк FastAPI для створення API. Він дозволяє отримувати дані про валютні курси та BT з Firebase Realtime Database[12].

Детальний розглянемо його поетапно:

Імпорт необхідних модулів:

UUID з модуля `uuid` - це клас, який дозволяє створювати та маніпулювати значеннями унікальних ідентифікаторів.

CORSMiddleware з модуля `fastapi.middleware.cors` - це проміжний програмний засіб, який додає підтримку міжсайтових запитів (CORS) до застосунків FastAPI.

FastAPI з модуля `fastapi` - це основний клас застосунків FastAPI, який використовується для створення API.

RedirectResponse з модуля `starlette.responses` - це клас, який дозволяє перенаправляти відповідь на іншу URL-адресу.

`firebase` - це модуль, який надає доступ до Firebase Realtime Database.

Ініціалізація об'єкту `data` з використанням `firebase.FirebaseApplication`, який підключається до Firebase Realtime Database за допомогою URL-адреси.

Створення екземпляру класу FastAPI з ім'ям `app`.

Додавання проміжного програмного засобу CORSMiddleware до застосунків `app`. Цей проміжний програмний засіб дозволяє перехоплювати міжсайтові запити та встановлювати правила CORS.

Визначення маршруту за допомогою декоратора `@app.get("/")`. Цей маршрут вказує на URL-адресу кореневої сторінки. При отриманні запиту на цей маршрут відбувається перенаправлення на URL-адресу `"/docs"`.

Визначення маршрутів для отримання даних про валютні курси та BT з Firebase Realtime Database. Кожен маршрут має свій унікальний шлях (URL) та параметри:

`/rate/{currency_code}` - повертає дані про валютний курс для заданого `currency_code`.

`/currency/{currency_code}` - повертає дані про валюту для заданого `currency_code`.

`/description/{currency_code}` - повертає опис ВТ для заданого `currency_code`.

`/currencies` - повертає всі дані про доступні ВТ.

Усі ці маршрути використовують метод GET для отримання даних.

Кожен маршрут має відповідну функцію-обробник, яка виконується при отриманні запиту на відповідний маршрут. Функції-обробники повертають дані, отримані з Firebase Realtime Database за допомогою об'єкту `data`.

Цей код реалізує простий веб-сервер, який надає API для отримання даних про валютні курси та ВТ з Firebase Realtime Database.

Другий файл `vercel.json` цей код є конфігураційним файлом для розгортання веб-застосунк на платформі Vercel. Давайте розглянемо його детальніше:

`"devCommand": "uvicorn main:app --host 0.0.0.0 --port 3000"` - Ця команда визначає команду розробника, яка виконується для локального режиму розробки. У цьому випадку, використовується сервер Uvicorn для запуску веб-застосунк, вказаного у файлі `main.py`, на локальному хості (0.0.0.0) та порту 3000[13].

`"builds": [...]` - Цей розділ визначає налаштування для збірки проекту на платформі Vercel.

`"src": "main.py"` - Вказує шлях до файлу, який є основним файлом застосунк.

`"use": "@vercel/python"` - Вказує, що для збірки проекту використовується середовище Python на платформі Vercel.

`"routes": [...]` - Цей розділ визначає маршрутизацію запитів для веб-застосунк на платформі Vercel.

`"src": "/(.*)"` - Вказує шаблон URL-адреси, до якого відповідає цей маршрут. У цьому випадку, `/(.*)` відповідає будь-якому URL-адресу.

`"dest": "main.py"` - Вказує файл, який обробляє запити для відповідного маршруту. У цьому випадку, всі запити, що відповідають `/(.*)`, будуть направлені до файлу `main.py`.

Цей код визначає налаштування для розгортання веб-застосунок на платформі Vercel, включаючи команду розробника, налаштування збірки та маршрутизацію запитів. Він вказує платформі, як обробляти запити та як запустити веб-додаток для розгортання.

Третій файл `test.py` даний код виконує веб-скрапінг для отримання валютних курсів з веб-сайту <https://index.minfin.com.ua>. Детальніше розглянемо його поетапно:

Імпорт необхідних модулів:

`json` - модуль для роботи з форматом JSON.

`defaultdict` з модуля `collections` - це спеціальний вид словника, який автоматично створює значення за замовчуванням для нових ключів.

`datetime` з модуля `datetime` - це модуль для роботи з датою та часом.

`requests` - модуль для здійснення HTTP-запитів.

`BeautifulSoup` з модуля `bs4` - це бібліотека для парсингу HTML-коду.

`relativedelta` з модуля `dateutil.relativedelta` - це клас, який дозволяє виконувати арифметичні операції з датами.

Визначення початкової та кінцевої дат для скрапінгу валютних курсів.

`start_date` - початкова дата (2020-01-01).

`end_date` - кінцева дата (2023-05-01).

Ініціалізація порожніх словників для зберігання валютних курсів та даних про ВТ.

`rates` - словник, де ключами є валютні коди, а значеннями - словники, де ключами є дати, а значеннями - валютні курси.

`currencies` - словник, де ключами є валютні коди, а значеннями - словники, які містять інформацію про ВТ (назви).

Виконання циклу, який проходить через кожен місяць від початкової до кінцевої дати.

За допомогою `strftime` отримується форматована дата у вигляді рядка.

Друкуються дати для перевірки.

Формування URL-адреси для скрапінгу валютних курсів на певну дату.

Використовується отримана дата для побудови URL-адреси.

Виконання HTTP-запиту до веб-сторінки з валютними курсами.

Використовується `requests.get` для отримання веб-сторінки за вказаною URL-адресою.

Отриманий відповідний текст сторінки.

Парсинг HTML-коду з використанням `BeautifulSoup`.

Створюється об'єкт `BeautifulSoup` з отриманого тексту сторінки.

Пошук блоків з валютними курсами за вказаними умовами.

Перевірка умов для блоків з валютними курсами.

Цикл обробки кожного рядка в блоку з валютними курсами.

Отримання необхідних даних (валютний код, кількість одиниць, назва ВТ, вартість).

Обчислення валютного курсу.

Збереження валютного курсу у словнику `rates`.

Збереження інформації про валюту у словнику `currencies`.

Запис результатів у два JSON-файли:

1.json - валютні курси (`rates`).

2.json - інформація про ВТ (`currencies`).

Цей код виконує веб-скрапінг, отримує валютні курси та інформацію про ВТ з веб-сайту та зберігає їх у вигляді JSON-файлів.

Даними кроками було описані всі складові проекту БД мобільного застосунок для проектування динаміки іноземних валют.

2.3 Створення прототипу макету дизайну мобільного застосунок “Exratedes”

Створення прототипу макету мобільного застосунок є важливим етапом в розробці програмного продукту. Макет дозволяє побачити вигляд і функціонал застосунок на початковій стадії та допомагає зосередитися на ключових елементах і функціональних можливостях.

Першим кроком при створенні прототипу макету є збір вимог і потреб користувачів, які додаток має задовольняти. Наступним кроком є створення скетчів - простих нарисів ілюструючих основні елементи і структуру застосунок. Скетчі можна створювати на папері або використовувати програмне забезпечення для створення макетів, таке як Sketch або Figma.

Створення якісного макету мобільного застосунок потребує дотримання ряду вимог, щоб дизайн був ефективним, зручним у використанні та забезпечував користувачам максимальний комфорт під час взаємодії з додатком. Ось деякі вимоги, які слід враховувати при створенні макету мобільного застосунок:

Простота та зрозумілість: дизайн макету має бути простим та лаконічним, щоб користувачам було легко зрозуміти, як користуватися додатком.

Консистентність: дизайн макету повинен бути консистентним по всьому застосунок, тобто має використовувати однакові стилі та елементи, щоб користувачам було легко зорієнтуватися та знаходити необхідні функції.

Наявність важливих функцій на головному екрані: на головному екрані застосунок мають бути відображені найважливіші функції, які користувачі використовують найчастіше. Це допомагає забезпечити зручний та швидкий доступ до необхідної інформації.

Грамотне використання кольорів: використання кольорів має бути обґрунтоване та логічним, щоб не викликати замішань чи незручностей під час користування додатком.

Адаптивність: макет має бути адаптивним до різних розмірів екранів, які використовуються на різних мобільних пристроях, щоб додаток виглядав добре на будь-якому пристрої.

Легкість у навігації: дизайн макету має бути зручним у використанні та легким у навігації.

Застосування відповідної типографії: типографія впливає на сприйняття текстового контенту та на загальний вигляд макету. Важливо використовувати

читабельний та зрозумілий шрифт, щоб забезпечити зручне сприйняття текстової інформації.

Відповідна величина елементів: елементи у макеті повинні мати відповідну величину та розміщуватися на екрані з розрахунком на зручність їх використання пальцями на сенсорному екрані.

Правильне використання іконок та графічних елементів: іконки та графічні елементи мають допомагати користувачам зорієнтуватися в застосунку та знайти потрібні функції. Важливо використовувати зрозумілі та логічні іконки та графічні елементи, щоб уникнути замішань.

Після створення скетчів, слід створити високорівневий макет, який відображає загальну структуру застосунку і його основні функціональні можливості. У цьому етапі можна використовувати зображення, текст та інші елементи для демонстрації взаємодії користувача з додатком.

Наступним кроком є створення детального макету, в якому відображаються всі сторінки та функції застосунку. У цьому етапі детально пророблюються всі елементи дизайну, включаючи кольори, шрифти, іконки та інші деталі. Важливо, щоб детальний макет був максимально точним і відображав реалістичний вигляд застосунку. На даному етапі всі дії проводились в застосунку Adobe Photoshop 2018. Даний макет складається з п'яти складових: головна, ВТ, візуалізація курсу, інше[14].

Перша складова – головна сторінка. Головна сторінка мобільного застосунку - це перший екран, який користувач побачить після запуску застосунку. Основною метою головної сторінки є привернення уваги користувача та надання швидкого доступу до основних функцій застосунку.

Головна сторінка має зрозумілий та лаконічний дизайн, щоб користувачі могли легко зорієнтуватися та швидко знайти потрібну інформацію. На цій сторінці було розміщено логотип застосунку, його назва, основна задача застосунку, банери з актуальними пропозиціями, іконки, які відображають основні функції застосунку. Дизайн головної сторінки є інтуїтивно зрозумілим та привабливим для користувача, щоб збільшити його зацікавленість та

мотивацію використовувати додаток. При створенні використовувались яскраві та насичені кольори, щоб привернути увагу користувача та зробити дизайн запам'ятовуванішим.

Головна сторінка також має розділи для різних категорій функцій застосунку, щоб допомогти користувачам знайти потрібну інформацію швидше. Наприклад даний додаток спеціалізується на візуалізації курсу валют, тому був доданий банер новини, який повідомляє про нещодавно додані ВТ. Також були створені три міні банера, які повідомлять користувача про популярні ВТ з якими проводиться візуалізація, та додані їх поточні курси один до одного. З нижнього меню навігації можна переміститись на інші сторінки.

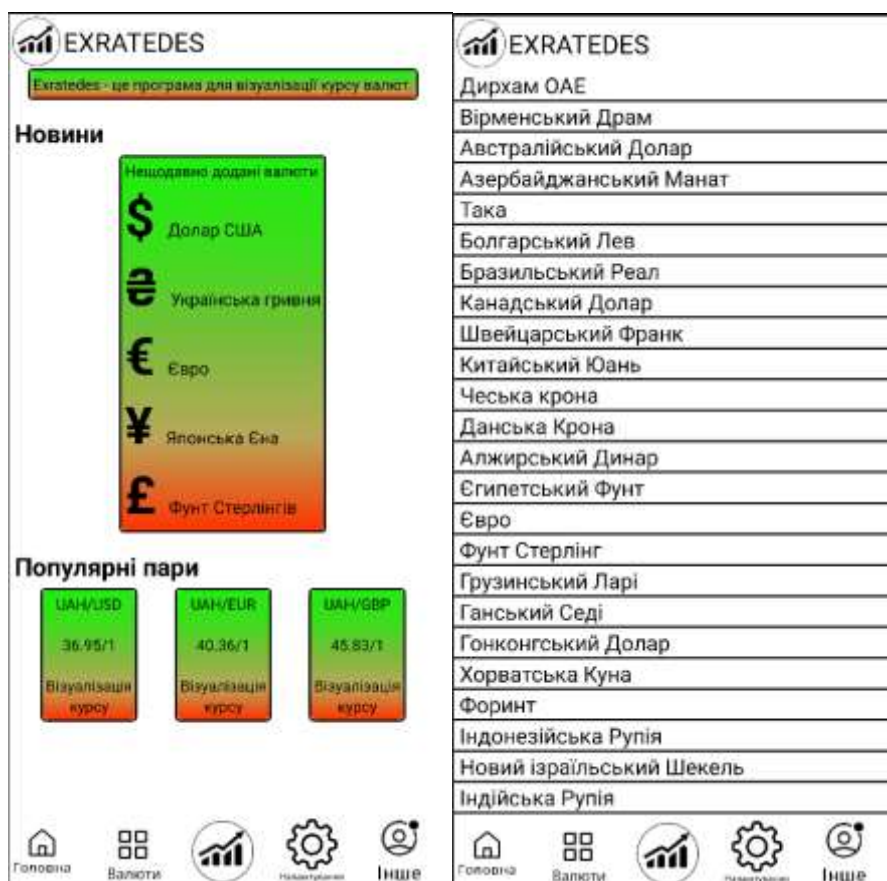


Рисунок 2.9 — макету мобільного застосунку “Exratedes”

Друга складова – сторінка ВТ. Сторінка ВТ зберігає в собі дані про всі ВТ які використовує додаток напряду з БД. Друга сторінка виконує ознайомчу роль. Роль допомагає користувачу дізнатися основну інфо-рмацію про валюту.

Третя складова – візуалізує курс валют. Основна сторінка застосунок. Так як вона призначена для виконання основної задачі застосунок. На даній сторінці розташовані два рядки в яких можна вибрати дві будь-яких ВТ та прости з ними візуалізацію. Візуалізація базується на двох рівнях. Перший рівень – це точний курс однієї ВТ відповідно до іншої. Другий рівень – це побудова графіку курсу ВТ відносно однієї до іншої.

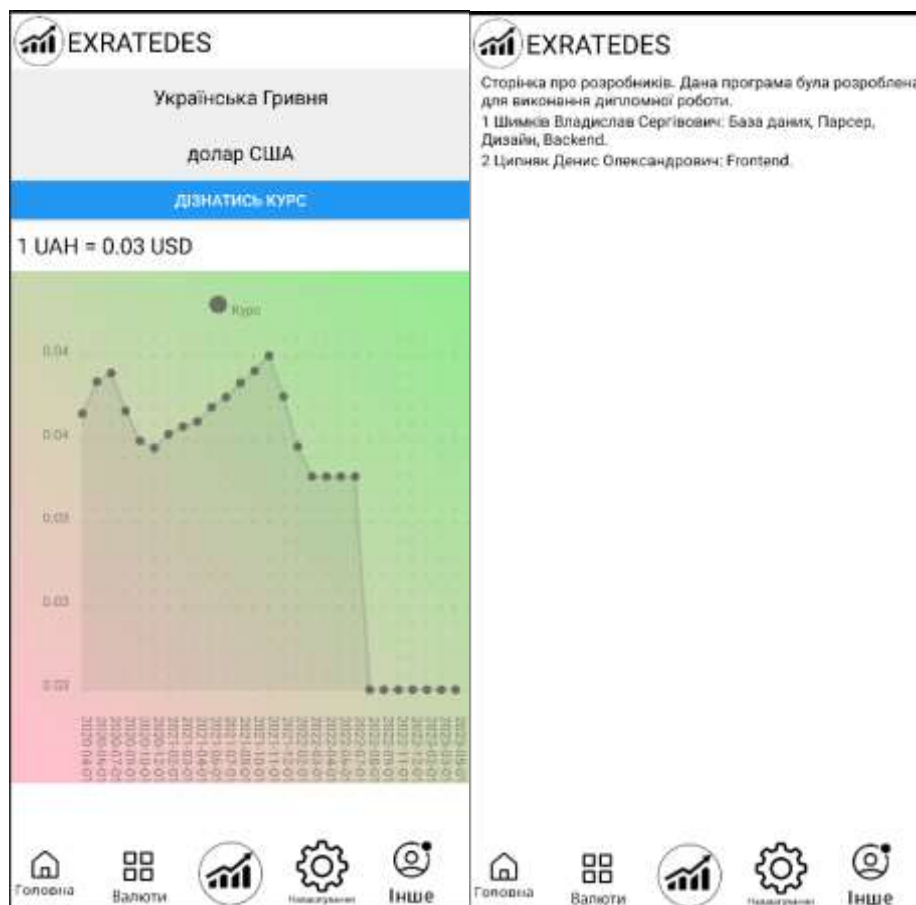


Рисунок 2.10 — макету мобільного застосунок “Exratedes”

П'ята складова – інше, остання сторінка яка містить у собі про розробника. Додатково про вмість додано основну інформацію про роботу, яка була зроблена розробниками.

Завершальним етапом є перевірка макету на практиці, щоб впевнитися, що всі елементи взаємодії користувача працюють правильно та дизайн відповідає потребам користувачів. У цьому етапі можна проводити тестування з фокус-групами та отримувати відгуки користувачів. Після завершення

тестування і внесення необхідних змін у макет, можна переходити до наступного етапу - розробки застосунків. Макет служить основою для програмістів, які розробляють додаток, тому важливо, щоб макет був детальним і максимально точно відображав функціонал застосунку.

Крім того, макет може бути корисним інструментом для комунікації з командою розробників, дизайнерами та замовника. Він дозволяє показати, як має виглядати додаток, що допомагає уникнути непорозумінь і помилок на етапі розробки.

У процесі створення макету важливо пам'ятати про потреби користувачів і створювати дизайн, який буде зручним і легким у використанні. Також важливо дотримуватися сучасних трендів у дизайні мобільних додатків і використовувати найкращі практики для створення макету.

Отже, створення макету мобільного застосунку є важливим етапом в розробці програмного продукту. Цей процес допомагає проробити всі деталі дизайну та функціоналу застосунку на початковій стадії та підвищити якість та ефективність розробки.

2.4 Реалізація мобільного застосунку

Реалізація мобільного застосунку проводиться в програмі для редагування та створення коду Visual Studio Code. Даний проект створюється під назвою "Exratedes". В міст проекту входять такі папки як assets (зображення), components (компоненти), layout (макет), pages (сторінки), utils (утиліти). А також два додаткових файлів App.js та app.json.

Перша група Assets вміщає в себе усі зображення які використовуються при створенні мобільного застосунку. До таких відносяться картинки для меню навігації, початкового загрузочного екрану, іконка застосунку, головний логотип застосунку.

Друга група Components вміщає три файли header.jsx, logo.jsx, rate_dropdown.jsx та папку layout.

Файл `header.jsx` код є частиною додатка на основі `React Native` і виконує налаштування інтерфейсу для верхньої частини екрана (заголовка).

При імпорті бібліотеки `React-Native`, використовуються імпорти `StyleSheet`, `View` і `Text` для налаштування стилів та відображення елементів на екрані. Також використовується компонент `Logo`, який імпортується з файлу `'./logo'`.

`Header` - компонент, який представляє заголовок. Він повертає `JSX`-код, що відображається на екрані. У компоненті `Header` знаходиться `View`, який містить два дочірні елементи: `Logo` і `Text`.

`Logo` - компонент, який відображається в заголовку.

`Text` - компонент, який відображає текст `"EXRATEDES"` на екрані.

Компонент `Header` має власні стилі, які визначаються об'єктом `styles`.

Об'єкт `styles` створюється за допомогою метода `StyleSheet.create()`, і включає два ключі: `header` і `text`.

`header` задає розташування заголовка. Він використовує `flexDirection: 'row'`, щоб елементи `Logo` і `Text` розміщувалися в одному рядку, `position: 'absolute'` розташовує заголовок в абсолютній позиції, `top: 0` встановлює заголовок вгорі екрана, а `alignItems: 'center'` вирівнює елементи по центру по вертикалі.

`text` задає стилі для тексту. В даному випадку, `fontSize` встановлено на 24.

На останньому рядку експортується компонент `Header` для використання в інших частинах застосунків.

Файл `logo.jsx` У даному фрагменті коду використовується імпорт `Image` з бібліотеки `react-native`.

`Image` - компонент, який дозволяє відображати зображення на екрані.

Функція `Logo` є компонентом, який відображає зображення логотипу.

У компоненті `Logo` використовується `JSX`-код, що включає компонент `Image`.

У властивості `source` компонента `Image` використовується шлях до зображення `../assets/rate_LOGO.png`. Цей шлях вказує на місце розташування

файлу зображення у проєкті. Зображення завантажується за допомогою функції `require()`, яка отримує шлях до зображення.

У властивості `style` встановлюються стилі для зображення. В даному випадку, зображення має ширину 50 і висоту 50, оскільки `style={{width: 50, height: 50}}`.

На останньому рядку експортується компонент `Logo` за допомогою `export default`, що дозволяє використовувати його в інших частинах застосунок.

Файл `rate_dropdown.jsx` Даний код є компонентом `RateDropDown`, який використовується для відображення та вибору ВТ зі списку.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів.

Компонент `RateDropDown` приймає два пропси: `value` і `first`.

У середині компоненту визначаються стани `options` і `selected` за допомогою функції `useState()`. Змінна `options` зберігається у стані для збереження варіантів валют, а `selected` використовується для збереження індексу вибраної ВТ.

Також використовуються хуки `useParams()` та `useNavigate()` з бібліотеки `react-router-native`. `params` зберігає параметри URL, а `navigate` - функція для переходу на іншу сторінку[15].

У компоненті використовується ефект `useEffect()`, який виконується після рендерингу компоненту. У функції ефекту виконується асинхронний запит до зовнішнього API, щоб отримати список валют. Після успішного отримання даних, вони зберігаються у стан `options`. Також обчислюється індекс вибраної ВТ за допомогою `Object.keys(currencies).indexOf(value)`, де `value` - це переданий пропс[16].

У поверненому JSX-кодi компонента використовується `SelectDropdown`, який є компонентом для відображення та вибору елементів зі списку. Властивість `data` встановлює значення списку валют зі стану `options`. `onSelect` - це функція, яка виконується при виборі елемента зі списку. Вона обробляє вибраний елемент та індекс і генерує шлях URL для переходу на іншу сторінку. Також використовується функція `navigate()` для переходу на нову сторінку.

Деякі інші властивості `SelectDropDown` встановлюють вигляд кнопки та тексту вибраного елемента.

На останньому рядку екпортується компонент `RateDropDown`. Також визначаються стилі для кнопки за допомогою `StyleSheet.create()`, де `button` задає ширину кнопки на всю ширину екрану.

Наступною до розгляду надано папку `layout` із її двома файлами `layout.jsx` та `layout.button.jsx`.

Файл `layout.jsx` даний код є компонентом `Layout`, який відображає розміщені горизонтально кнопки для навігації на різні сторінки додатка.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів.

Компонент `Layout` використовує хук `useNavigate()` з бібліотеки `react-router-native` для отримання функції навігації.

У поверненому JSX-кодї компонента використовується `View` для створення контейнера, в якому будуть розміщені кнопки.

Компонент `LayoutButton` використовується для кожної кнопки навігації. Кожна кнопка має своє зображення, яке передається через пропс `image`, та функцію `onPress`, яка виконується при натисканні кнопки та викликає відповідний роутер для переходу на відповідну сторінку.

В `View` розміщені п'ять `LayoutButton`, які відповідають за навігацію на головну сторінку, сторінку зі списком валют, сторінку з курсами обміну валют, сторінку налаштувань та сторінку інформації.

У стилі `layout` визначаються властивості контейнера `View`, такі як вирівнювання елементів по центру по горизонталі (`alignItems: 'center'`), розташування елементів в одному рядку (`flexDirection: 'row'`), абсолютна позиція контейнера (`position: 'absolute'`) внизу екрану (`bottom: 0`), `zIndex` встановлює значення шару над іншими елементами і фон контейнера встановлено на білий колір (`backgroundColor: 'white'`)[17].

На останньому рядку екпортується компонент `Layout`.

Файл `layoutbutton.jsx` Даний код є компонентом `LayoutButton`, який відображає кнопку з зображенням для використання в компоненті `Layout`.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів.

Компонент `LayoutButton` отримує два пропси: `onPress` (функція, яка виконується при натисканні кнопки) та `image` (шлях до зображення кнопки).

У поверненому JSX-кодi компонента використовується `View` для створення контейнера, в якому буде розміщено кнопку.

`TouchableOpacity` використовується для створення натисканої області, на яку можна натискати[18].

`Image` відображає зображення, яке передається через пропс `image`.

`TouchableOpacity` обгортає компонент `Image`, і коли користувач натискає на кнопку, виконується функція `onPress`.

У стилі `image` задаються розміри зображення (`height: 64, width: 64`).

У стилі `container` задається розтягнення контейнера на всю доступну ширину і висоту (`flex: 1`).

На останньому рядку експортується компонент `LayoutButton`.

Третя група `Pages` налічує шість файлів які відповідають своїм першо задуманим концептам сторінок для навігації користувача, та надання поточного користування.

Файл `home.jsx` даний код представляє компонент `Home`, який відображає головну сторінку додатка з інформацією про курс валют, новини та популярні пари валют. Ця сторінка виступає як стартова лінія в застосунок.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів.

Змінна `colors` містить масив кольорів, які використовуються для градієнтного фону деяких контейнерів.

Компонент `Home` використовує хук `useNavigate()` з бібліотеки `react-router-native` для отримання функції навігації.

У поверненому JSX-кодi компонента представлено розмiтку сторiнки з використанням View, Text та TouchableOpacity для створення контейнерiв, тексту та кнопок вiдповiдно.

LinearGradient з бiблiотеки expo-linear-gradient використовується для створення градiєнтного фону для деяких контейнерiв.

У першому View мiститься LinearGradient з текстом, який вiдображає опис додатка.

Далi йдуть роздiли "Новини" та "Популярнi пари" з вiдповiдними заголовками (Text) i горизонтальним розташуванням (View з flexDirection: 'row').

В роздiлi "Нещодавно доданi BT" є кiлька TouchableOpacity, якi мiстять текст про валюту. При натисканнi на кожену кнопку викликається функцiя onPress, яка вiдправляє користувача на вiдповiдну сторiнку за допомогою функцiї навігацiї.

Аналогiчно для роздiлу "Популярнi пари" створюються TouchableOpacity з iнформацiєю про валютнi пари та вiдповiдними функцiями навігацiї.

На останнiх рядках визначаються стилi за допомогою StyleSheet.create(), в яких задаються розмiри, вирiвнювання, колiр та iншi властивостi елементiв на сторiнцi.

Компонент Home експортується за допомогою export default.

Файл currency.jsx даний код представляє компонент Currency, який вiдображає iнформацiю про певну валюту на сторiнцi.

У перших рядках коду виконується iмпорт необхідних бiблiотек та компонентiв.

Компонент Currency використовує хук useState для збереження стану BT i хук useParams з бiблiотеки react-router-native для отримання параметрiв URL.

У хуку useEffect виконується асинхронна функцiя fetchData, яка отримує данi про валюту та опис BT зi специфiчного URL за допомогою axios.get(). Отриманi данi зберiгаються в станi currency за допомогою setCurrency.[19]

У поверненому JSX-кодi компонента вiдображаються рiзні атрибути `BT`, такі як назва, код, символ, рeгiон та опис. Умовні оператори `if-else` використовуються для перевiрки наявності атрибутiв та їх вiдображення на сторiнці. Якщо атрибут вiдсутній, вiдповiдний блок тексту не вiдображається.

На останньому рядку компонент `Currency` експортується за допомогою `export default`.

Файл `currencies.jsx` даний код представляє компонент `Currencies`, який вiдображає список валют на сторiнці.

У перших рядках коду виконується iмпорт необхідних бiблiотек та компонентiв, включаючи `react-native`, `axios` та `react-router-native`.

Компонент `Currencies` використовує хук `useState` для збереження стану валют та хук `useEffect` для асинхронного отримання даних про `BT` зi специфiчного URL за допомогою `axios.get()`. Отриманi данi зберiгаються в стані `currencies` за допомогою `setCurrencies`.

Змiнні `screenWidth` та `screenHeight` використовуються для отримання ширини та висоти екрана пристрою.

У поверненому JSX-кодi компонента використовується `View` та `ScrollView` для створення прокручуваного списку валют. Кожна валюта вiдображається як `TouchableOpacity`, де назва `BT` вiдображається у вигляді тексту. При натисканні на валюту виконується перехiд до сторiнки з детальною iнформацією про валюту за допомогою `navigate`.

Компонент `Currencies` експортується за допомогою `export default`.

Файл `rate.jsx` даний код представляє компонент `Rate`, який вiдображає валютний курс та графiк для вибраних валют.

У перших рядках коду виконується iмпорт необхідних бiблiотек та компонентiв, включаючи `react-native`, `axios` та додатковi утилити.

Компонент `Rate` використовує хук `useState` для збереження стану валютного курсу та графiку. Також використовується хук `useParams` для отримання параметрiв з URL.

У коді є дві асинхронні функції `get_rate_to_UAH` та `get_rate`, які виконують запити до API для отримання валютного курсу та обробки даних для відображення. Параметр `rate` розбивається на дві ВТ за допомогою деструктуризації.

У функції `get_rate` виконується запит до API за допомогою функції `get_rate_to_UAH`, отримані дані обробляються та зберігаються у стані `rateText` та `chart`.

У поверненому JSX-кодi компонента використовуються `View`, `Button` та `Text` для створення інтерфейсу користувача. Для вибору валют використовуються компоненти `RateDropDown`. При натисканні на кнопку "Дізнатись курс" виконується функція `get_rate`, яка оновлює стан валютного курсу та графіку.

Графік відображається за допомогою `LineChart` з бібліотеки `react-native-chart-kit`. Налаштування графіка задаються у `chartConfig`.

Компонент `Rate` експортується за допомогою `export default`.

Файл `settings.jsx` даний код представляє компонент `Settings`, який відображає налаштування програми, зокрема вибір мови.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів, включаючи `react-native` та `react-native-select-dropdown`.

Компонент `Settings` містить у собі вибірку для вибору мови. Вибір мови представлений за допомогою компонента `SelectDropdown`, який отримує дані `languages` та деякі опції налаштування. Масив `languages` містить доступні мови для вибору.

У поверненому JSX-кодi компонента використовуються `View`, `Text` та `SelectDropdown` для створення інтерфейсу користувача. Заголовок та підзаголовок стилізуються з використанням `styles`.

Компонент `Settings` експортується за допомогою `export default`.

Файл `Info.jsx` даний код представляє компонент `Info`, який відображає сторінку з інформацією про розробників програми.

У перших рядках коду виконується імпорт необхідних бібліотек та компонентів, включаючи react-native.

Компонент Info містить у собі розмітку View та текстові елементи Text, які відображають інформацію про розробників та опис програми. Текстові рядки відображаються у форматі списку, де кожен рядок містить інформацію про розробника та його внесок у проект.

У поверненому JSX-коді компонента використовуються View та Text для створення інтерфейсу користувача та відображення текстової інформації.

Компонент Info експортується за допомогою export default.

Четверта група Utils відповідає за утилітні настройки зв'язку програми. Дана група налічує два файли divide_objects.jsx та rate_chart_data.jsx.

Файл divide_objects.jsx Цей код представляє функцію divideObjects, яка приймає два об'єкти obj1 і obj2 і повертає новий об'єкт, в якому кожне значення властивості prop об'єкта obj1 ділиться на відповідне значення властивості prop об'єкта obj2.

У функції створюється порожній об'єкт result, який буде містити результат ділення. Потім за допомогою циклу for...in проходиться по властивостях об'єкта obj1. У кожній ітерації перевіряється, чи має об'єкт obj2 властивість з таким самим ім'ям (hasOwnProperty(prop)). Якщо така властивість існує, то значення властивості prop об'єкта obj1 ділиться на значення властивості prop об'єкта obj2, і результат присвоюється властивості prop об'єкта result.

На кожній ітерації циклу об'єкт result розширюється новою властивістю, яка представляє результат ділення об'єктів obj1 і obj2. Після проходження всіх властивостей об'єкта obj1, функція повертає об'єкт result, який містить результати ділення.

Код дозволяє розділити властивості двох об'єктів за певною логікою і створити новий об'єкт з результатами.

Файл `rate_chart_data.jsx` це функція `getRateChartData` приймає об'єкт `rate`, який містить дані курсу, і повертає об'єкт з даними, які можна використовувати для візуалізації графіка.

Властивість `labels` містить масив ключів об'єкта `rate`, які використовуються для позначення осі X на графіку.

Властивість `datasets` містить масив об'єктів, в яких властивість `data` містить масив значень об'єкта `rate`. Ці значення використовуються для побудови графіку.

Властивість `legend` містить масив з одним елементом "Курс", який використовується для позначення легенди графіку.

Отже, ця функція допомагає структурувати дані курсу в форматі, який можна використовувати для відображення графіка.

П'ята група `Default` налічує файли низку файлів, які відповідають за базові настройки програми. Іншими словами група `Default` є основою даного застосунок. З перелічених файлів можна виділити два файли `App.js` та `App.json`. Дані файли є структурною [20].

Файл `App.js` - код є головним файлом додатка `React Native` і відповідає за налаштування маршрутизації та рендеринг сторінок додатка.

Основний компонент `App` містить такі дії:

Імпортує необхідні компоненти з інших файлів, такі як `Home`, `Currencies`, `Rate`, `Info`, `Settings`, `Layout`, `Header`, `Currency`.

Використовує компоненти `SafeAreaView` та `StatusBar` для налаштування безпечного виду та статус-бару додатка.

Використовує `NativeRouter` для налаштування маршрутизації в застосунок.

Розміщує заголовок `Header` над вмістом додатка.

Використовує компонент `Routes` для визначення шляхів та компонентів, які пов'язані з кожним шляхом.

Компоненти `Route` визначають шляхи та пов'язані з ними компоненти, які мають бути відображені на відповідній сторінці.

Компонент `Layout` розміщується під `Routes` і відображає фіксовані елементи, що з'являються на кожній сторінці.

Код забезпечує основну структуру та функціональність додатка, розміщуючи відповідні компоненти на відповідних сторінках за допомогою маршрутизації.

Файл `App.json` - код є конфігураційним файлом для додатка `Expo`. Він містить різні параметри налаштувань для додатка, такі як назва, версія, орієнтація, іконка, розмір сплеш-екрану, налаштування для платформ `iOS` та `Android`, налаштування для веб-версії та додаткові параметри.

Основні параметри включають:

- `name`: Назва додатка.
- `slug`: Слаг (ідентифікатор) додатка.
- `version`: Версія додатка.
- `orientation`: Орієнтація додатка (`portrait` або `landscape`).
- `icon`: Шлях до іконки додатка.
- `splash`: Налаштування сплеш-екрану, включаючи зображення, режим зміни розміру та колір фону.
- `assetBundlePatterns`: Шаблони для пакетів активів, які повинні бути включені в додаток.
- `ios`: Налаштування для платформи `iOS`, такі як підтримка планшетів.
- `android`: Налаштування для платформи `Android`, такі як адаптивна іконка та ідентифікатор пакету.
- `web`: Налаштування для веб-версії, такі як фавіконка.
- `extra`: Додаткові параметри, які можуть бути використані для розширеного налаштування, наприклад, проекту `EAS`.

Код допомагає визначити налаштування додатка `Expo` та забезпечує відповідність платформам `iOS`, `Android` та веб.

2.5 Висновок до другого розділу

Таким чином був наведений опис коду для декількох компонентів, функцій та конфігураційних файлів в застосунок React Native, які містяться в вашому проекті. Були наведені коментарі та пояснення для кожної частини коду, пояснюючи, що вона робить і як вона працює.

Додаток включає різні сторінки, такі як головна сторінка, сторінка зі списком валют, сторінка з курсом валют, сторінка з інформацією про розробників та сторінка налаштувань. Використовується роутер для навігації між сторінками.

Крім того є компоненти для відображення даних, такі як графік курсу валют та випадаючий список для вибору мови.

Також було надано приклади додаткових функцій, таких як функція для розділення об'єктів, функція для отримання даних для графіка курсу валют і функція для форматування даних графіка.

У загальному, розділі описаний додаток, який використовує бібліотеки React Native, axios та react-router-native для створення мобільного застосунок для відстеження курсу валют. Розроблені компоненти для відображення даних, навігації між сторінками та взаємодії з API для отримання актуальних даних про курси валют, створення та використання БД і розробка дизайну мобільного застосунку.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Долікарська допомога при ураженні електричним струмом

Перш за все варто розуміти, що удар струмом – це критична загроза здоров'ю та навіть життю людини, тому виклик лікаря в таких випадках є обов'язковим. Окрім того, будь-який прояв зволікання при допомозі чи не якісне її надання можуть заподіяти більших травм чи навіть викликати смерть потерпілого.

Спочатку необхідно проаналізувати поточну ситуацію та зрозуміти, чи потерпілий все ще перебуває під дією електричного струму. Якщо це так, то потрібно, при можливості, якомога швидше спробувати ізолювати джерело струму, вимкнути його, перерізати провідник струму за допомогою інструмента з ізольованими або електро-непровідними ручками, відштовхнути його від потерпілого за допомогою електро-непровідного підручного засобу.

Окрім того, варто спробувати відтягнути потерпілого від джерела електричного струму, уникаючи при цьому контакту із оголеними частинами тіла або навколишніми, особливо металевими предметами. При цьому необхідно ізолювати себе – за наявності одягнути діелектричні рукавиці, обмотати долоні сухою тканиною. Вищеописані дії будуть справедливими у випадках, коли на потерпілого діє струм напругою до 1000 В [21].

Якщо ж відомо, що потерпілий уражений струмом напругою від 1000 В і більше, то безпечно допомогти йому можна тільки при використанні спеціального діелектричного взуття і рукавиць, при цьому, за можливості, відкидаючи джерело струму спеціальною ізольованою штангою. Також можна спробувати заземлити провідники, дія яких спричинила до електричного ураження.

В обидвох випадках, якщо джерелом є провід, що торкається землі чи звисає з ЛЕП, не варто забувати про значну небезпеку так званої крокової

напруги [21]. Ось чому, після звільнення потерпілого від дії джерела електричного струму, без наявних засобів захисту на кшталт вищезгаданого 45 діелектричного взуття, життєво необхідно в зоні розтікання струму (вона становить приблизно 8 метрів) пересуватись таким чином, щоб не відривати однієї ноги від іншої (див. рис. 3.1).

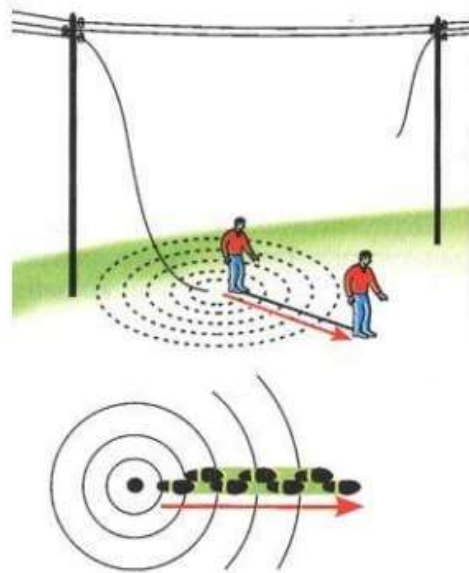


Рисунок 3.1 – Пересування в 8-ми метровій зоні дії крокової напруги

Після того, як дію провідника електричного струму на потерпілого було припинено (або вона була відсутня з самого початку), необхідно провести діагностування стану людини. Загалом, в наслідок дії електричного струму на організм, прийнято виділяти три стани [22]:

– перший стан: потерпілий знаходиться в свідомості. В цьому випадку необхідно надати йому зручного положення та забезпечувати спокій до прибуття медичного працівника [22]. Окрім того, не можна дозволяти людині рухатись чи продовжувати будь-яку фізичну діяльність, адже це може погіршити його стан;

– другий стан: потерпілий знаходиться без свідомості, але продовжує дихати. Аналогічно до першого, необхідно покласти людину в стабільне положення, розстебнути його одяг, що заважає диханню та, по можливості, забезпечити надходження свіжого повітря [22]. Окрім того, можна спробувати

привести потерпілого до тями, давши йому понюхати нашатирний спирт або бризнувши на його обличчя водою;

– третій стан: потерпілий не дихає зовсім, або його дихання переривчасте, судомне, поверхневе. Це найважчий стан із всіх трьох. При ньому необхідно невідкладно почати проводити заходи серцево-легеневої реанімації у вигляді зовнішнього масажу серця та штучного дихання (рис. 3.2). Якщо цього не зробити, то до приїзду бригади швидкої допомоги неодмінно настане смерть потерпілого [23]. Перед проведенням штучного дихання необхідно надати потерпілому правильного положення, перевірити наявність сторонніх тіл чи рвотних або кров'яних мас в ротовій порожнині та, за необхідності, очистити її.



Рисунок 3.2 – Схематичне зображення виконання серцево-легеневої реанімації однією людиною

Справедливим до усіх трьох станів буде наступне: переміщати потерпілого можна тільки тоді, коли існує загроза його життю чи життю людини, яка надає першу допомогу. Також, при роботі з застосунками користувач повинен правильно поводитися з ПК, адже це потенційне джерело напруги та небезпеки. Окрім того, удар електричним струмом здатен призвести до смерті. Ось чому надання негайної долікарської допомоги є важливим фактором задля збереження здоров'я та життя. Для превентивного запобігання

уражень електричним струмом при роботі з ПК слід встановити додаткові захисні пристрої, які гарантуватимуть ізолюваність струмо-провідних частин.

3.2 Загальні вимоги безпеки з охорони праці для користувачів ПК

Перелік загальних вимог для користувачів ПК є досить об'ємними. Так, перш за все, до самостійної роботи із ЕОМ чи ПК повинні допускатись тільки ті особи, які ознайомлені із правилами безпеки, пройшли відповідний навчальний інструктаж з питань охорони праці, а також пожежної безпеки при роботі з ПК. Окрім того, за необхідності дані персони повинні мати довідку про медичний огляд та дозвіл бути допущеними до комп'ютера за станом здоров'я [24].

Сам користувач повинен дотримуватися ряду заздалегідь визначених та затверджених документально правил: не можна допускати до наданого ПК сторонніх осіб; забороняється встановлювати на обладнання неліцензійне ПЗ; категорично забороняється виконувати дії, що суперечать загальним вимогам з охорони праці.

Якщо ж говорити про вимоги безпеки з охорони праці для користувачів ПК безпосередньо перед початком роботи, то останні повинні переконатись у відсутності пошкоджень обладнання та електро мережі, а у разі їх виявлення повідомити відповідальну особу і ні в якому разі не приступати до роботи. Окрім того, робоче місце повинно розміщуватись на відстані не менше 1 метра від стіни. Кут нахилу монітора ПК при цьому повинен становити від 10° до 25° , а відстань від очей користувача до монітору – в діапазоні від 500 мм до 600 мм. Окрім того, до поля зору користувача не повинні попадати прямі чи віддзеркалені сонячні промені [25].

Відповідно до цього робоче місце варто розмістити таким чином, щоб світло від вікна падало з лівого чи правого боків. Правильне облаштування робочого місця подано на рисунку 3.3. Якщо поруч наявне будь-яке джерело тепла, то, відповідно до вимог безпеки з охорони праці, ПК повинен розташовуватись від нього на відстані не менше, ніж 1 метр.

При цьому освітленість робочого місця користувача ПК має становити не менше 400 лк на висоті 80 см від підлоги та не більше 200 лк у площині екрану [25]. Приклад коректної посадки користувача ПК та його робочого місця відповідно до загальних вимог безпеки з охорони праці подано на рисунку 3.3.

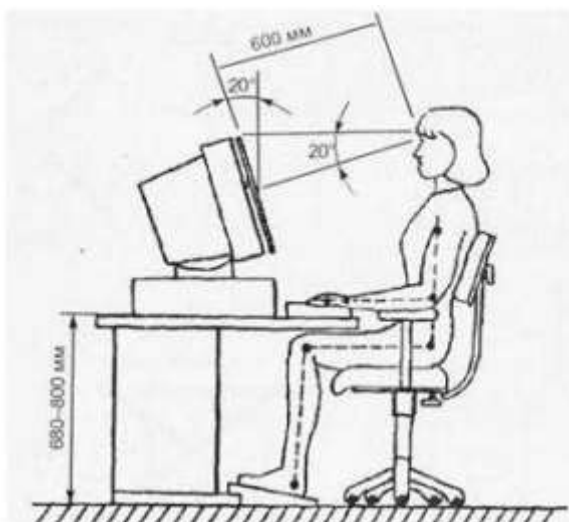


Рисунок 3.3 – Зображені схематично вимоги до облаштування та організації робочого місця користувача ЕОМ чи ПК

Окрім того, відповідно до вимог з безпеки охорони праці для користувачів ПК, останні мають знати алгоритм дій в критичних чи аварійних ситуаціях [35]. Так, при виявленні стороннього звуку чи гореного запаху, необхідно негайно припинити роботу із ПК, вимкнути все обладнання, знеструмити його та повідомити про це відповідальну особу.

При роботі із застосунками та ПК загалом користувач повинен дотримуватись усіх вищезазначених правил та рекомендацій, що сприятиме комфортній та достатньо продуктивній роботі.

3.3 Методи боротьби з монотонністю праці на виробництві

Реалізація творчих здібностей особистості, підвищення мотивації до праці за рахунок так званого «збагачення» праці набувають дедалі більшого значення в розвитку виробництва на сучасному етапі.

Обґрунтування системи заходів щодо запобігання монотонності і її негативних наслідків базується на вченні І. П. Павлова і І. М. Сеченова про необхідність розширення поля коркової активності в процесі праці і виключення «довбання» в одну клітину.

Найрадикальнішим заходом є проектування раціональних трудових процесів і операцій на основі оптимального поділу праці. Завдання полягає в тому, щоб кожную операцію зробити змістовною, яка сприяла б розвитку у працівника творчого мислення. Основним принципом проектування раціонального трудового процесу (операції) є принцип збереження певної логічної завершеності і структурної цілісності виконуваної операції. Навіть в умовах глибокої диференціації технологічного процесу необхідно встановлювати таку кількість елементів операції і послідовність їх виконання, яка сприймалася б працівником як логічно завершена одиниця.

Другим важливим принципом проектування трудового процесу є забезпечення достатнього енергетичного рівня операції. Спеціальними дослідженнями встановлено, що негативні психічні стани більшою мірою виявляються при виконанні тих робіт, які через незначну енергетичну вартість не стимулюють функціональної активності організму. Якщо монотонна робота досить інтенсивна за затратами енергії, то нудьга, сонливість, психічне перенасичення можуть не виникати. Доведено, що при фізичній роботі для підтримання активного тону кори затрати енергії не повинні бути меншими за 2,5 ккал/хв (150 ккал/год).

Запобігання монотонності і підвищенню змістовності праці сприяє укрупнення трудових операцій. Завдяки укрупненню операцій у працівника формується більш складний стереотип трудових дій, що позитивно позначається на стані психофізіологічних функцій. Досвід показує, що операція

повинна складатися не менш як з 5 — 6 елементів за умови збереження цільового змісту.

Важливим засобом боротьби з монотонністю є чергування операцій, кожна з яких є монотонною. Науковою основою чергування операцій є ефект Сеченова, суть якого в тому, що при зміні діяльності активізується інша група нервових центрів, а в раніше працюючих ефективно відбувається «заправка» енергією. Отже, принцип чергування операцій полягає в заміщенні і компенсації психофізіологічних функцій, активізації інших м'язових груп, нервових центрів, зменшенні надмірного напруження працюючих м'язів. Значення чергування операцій, таким чином, полягає в ліквідації негативного впливу однобоких навантажень. На практиці застосовується декілька варіантів чергування операцій: через кожен годину, через 2,5 год, один раз протягом зміни, через день. Відносно зняття фактора монотонності найбільш ефективно чергування операцій один раз протягом зміни, хоча в конкретних виробничих умовах це питання вирішується по-різному. Враховуються умови праці, структура операцій, майстерність працівників.

Чергування операцій пов'язане із суміщенням професій і трудових функцій. Зазначимо, що оволодіння працівником другими і суміжними професіями, крім подолання монотонності і підвищення привабливості праці, підвищує конкурентоспроможність працівника на ринку праці і мобільність на самому підприємстві.

Для зняття монотонності необхідно, щоб операції відрізнялися за характером навантажень, але в той же час були позбавлені інтерферентних елементів.

Основні умови суміщення професій і трудових функцій, які забезпечують зменшення монотонності:

- суміщувані професії повинні змінювати рівень завантаженості різних органів і систем;
- суміщувана операція повинна бути легшою, ніж основна. При легкій монотонній роботі ефективна зміна на більш важку;

– більш монотонну роботу необхідно суміщувати з менш монотонною; – суміщувані трудові комплекси повинні забезпечувати роботу за участю м'язів-антагоністів, а також зміну робочих поз;

– статичні навантаження повинні компенсуватися помірними динамічними навантаженнями.

При організації монотонних робіт важливе значення має вибір темпу роботи. Темп може бути вільним або примусовим. Кожний з них має переваги і недоліки. Тому при виборі темпу роботи слід виходити зі специфіки конкретного виробництва. В одних випадках доцільним є оптимальний заданий темп з регулюванням швидкості конвеєра відповідно до кривої працездатності. Варіація швидкості не повинна перевищувати 10—15 %. В інших випадках ефективне самостійне регулювання робочого темпу. Останнє застосовується на автономних конвеєрах, що забезпечує не лише свободу ритму, а й регулювання змісту роботи.

Ефективним засобом боротьби з монотонністю є бригадно-групова форма організації потоку. Суть її в тому, що бригада виконує операції всього циклу по виготовленню більш-менш закінченого продукту (вузла). Процеси виготовлення кожного вузла виділяються в самостійні виробничі секції. Робітники працюють у вільному ритмі, а вузли з'єднуються в монтажній секції. В цьому випадку трудовий процес менше розчленований і тісніше кооперований.

Зменшенню негативного впливу монотонних робіт на психічний стан працівників і показники їхньої праці сприяють такі заходи:

- раціоналізація режимів праці і відпочинку;
- естетизація виробничого середовища;
- застосування функціональної музики.

До факторів зменшення монотонності відносяться також психологічні заходи, покликані посилити внутрішні мотиви діяльності. Це, зокрема, психологічна стимуляція трудової діяльності за рахунок постановки проміжних виробничих цілей, забезпечення працівників поточною інформацією щодо

виконання роботи. Особливе значення мають залучення робітників до управління і розв'язання виробничих проблем, а також сприятливий соціальнопсихологічний клімат, створення умов для спілкування в процесі праці, якщо це можливо. Усе це формує позитивні емоційні стани у працівників, посилює їх монотоностійкість[26].

3.4 Вплив виробничого середовища на працездатність та здоров'я користувачів комп'ютерів

До небезпечних психофізіологічних та шкідливих виробничих чинників належать фізичні (статичні, динамічні та гіподинамічні) і нервовопсихічні перевантаження (розумове, зорове, емоційне). Праця користувачів комп'ютерів характеризується тривалою багатогодинною (8 год і більше) працею в одноманітному напруженому положенні, малою руховою активністю при значних локальних динамічних навантаженнях. Робоче положення “сидячи” супроводжується статичним навантаженням значної кількості м'язів ніг, плечей, шиї та рук, що дуже втомлює. М'язи перебувають довгий час у скороченому стані і не розслабляються, що погіршує кровообіг. В результаті, виникають больові відчуття в руках, шиї, верхній частині ніг, спині та плечових суглобах. Внаслідок динамічного навантаження на кістково-м'язовий апарат кистей рук виникають больові відчуття різної сили в суглобах та м'язах кистей рук; оніміння та уповільнена рухливість пальців; судоми м'язів кисті; ниючий біль в ділянці зап'ястя. Як результат, виникають локальні м'язові перенапруження хронічні розтягнення м'язів травматичного характеру, що можуть викликати професійні захворювання: дисоціативні моторні розлади, захворювання периферійної нервової та кістково-м'язової систем. Крім того, робота “сидячи” призводить до зниження м'язової активності - гіподинамії. За браком рухів відбувається зниження споживання кисню тканинами організму, сповільнюється обмін речовин. Це сприяє розвитку атеросклерозу, ожиріння,

може стати причиною дистрофії міокарда, хронічного головного болю, запаморочення, безсоння, роздратування[27].

Трудова діяльність користувачів комп'ютерів належить до категорії робіт, які пов'язані з використанням великих обсягів Інформації, із застосуванням комп'ютеризованих робочих місць, із частим прийняттям відповідальних рішень в умовах дефіциту часу, безпосереднім контактом із людьми різних типів темпераменту тощо. Це зумовлює високий рівень нервово-психічного перевантаження, знижує функціональну активність центральної нервової системи, призводить до розладів в її діяльності, розвитку втоми, перевтоми, стресу[28].

Будь-яка діяльність, якщо вона оптимальна для організму по інтенсивності і тривалості та проходить у сприятливих виробничих умовах, позитивно впливає на організм і сприяє його удосконалюванню. Ефективність діяльності людини базується на рівні психічної напруги, яка прямо пропорційна складності завдання.

Психічна напруга – це фізіологічна реакція організму, яка мобілізує його ресурси (біологічно і соціально корисна реакція). Під впливом психічної напруги змінюються життєво важливі функції організму: обмін речовин, кровообіг, дихання. В поведінці людини спостерігається загальна зібраність, дії стають більш чіткими, підвищується швидкість рухових реакцій, зростає фізична працездатність. При цьому загострюється сприйняття, прискорюється процес мислення, поліпшується пам'ять, підвищується концентрація уваги. Пристосувальні можливості психічної напруги тим більші, чим вище психічний потенціал особистості. Механізм емоційної стимуляції має фізіологічний бар'єр, за яким настає негативний ефект (поза межна форма психічної напруги)[29].

При надмірній інтенсивності чи тривалості робота приводить до розвитку вираженого стомлення, зниження продуктивності, неповного відновлення за період відпочинку. Стомлення – загальний фізіологічний процес, яким супроводжуються усі види активної діяльності людини. З біологічної точки

зору стомлення – це тимчасове погіршення функціонального стану організму людини, що виявляється в змінах фізіологічних функцій і є захисною реакцією організму. Воно спрямоване проти виснаження функціонального потенціалу центральної нервової системи і характеризується розвитком гальмівних процесів у корі головного мозку.

Втома – це сукупність тимчасових змін у фізіологічному та психологічному стані людини, які з'являються внаслідок напруженої чи тривалої праці і призводять до погіршення її кількісних і якісних показників, нещасних випадків. Втома буває загальною, локальною, розумовою, зоровою, м'язовою та ін. Оскільки організм – єдине ціле, то межа між цими видами втоми умовна і нечітка. Хід збільшення втоми та її кінцева величина залежать від індивідуальних особливостей працюючого, трудового режиму, умов виробничого середовища тощо. Залежно від характеру вихідного функціонального стану працівника втома може досягати різної глибини, переходити у хронічну втому або перевтому.

Перевтома – це сукупність стійких несприятливих для здоров'я працівників функціональних порушень в організмі, які виникають внаслідок накопичення втоми. Основною відмінністю втоми від перевтоми є зворотність зрушень при втомі і неповна зворотність їх при перевтомі. Поза межні форми психічної напруги викликають дезінтеграцію психічної діяльності різної виразності. При цьому втрачається жвавість і координація рухів, знижується швидкість відповідних реакцій (гальмівний тип), з'являються непродуктивні форми поведінки – гіперактивність, тремтіння рук, запальність, невластива різкість і ін. (збудливий тип). Обидва типи поза межної напруги супроводжуються вираженими вегетативно-судинними змінами (блідість обличчя, краплі поту, прискорений пульс). До суб'єктивних ознак перевтоми відноситься почуття втоми, бажання знизити ритм роботи чи припинити її, почуття слабості в кінцівках. Важке стомлення – крайній варіант фізіологічного стану, що граничить з патологічними формами реакції. При перевтомі порушуються відновні процеси в організмі. Прикмети втоми не зникають до початку роботи

наступного дня. При наявності хронічної перевтоми часто зменшується маса тіла, змінюються показники серцево-судинної системи, знижується опір організму до інфекції. Відомо, що розвиток втоми та перевтоми веде до порушення координації рухів, зорових розладів, неуважності, втрати пильності та контролю реальної ситуації. При цьому працівник порушує вимоги технологічних інструкцій, припускається помилок та неузгодженості в роботі; у нього знижується відчуття небезпеки. Крім того, перевтома супроводжується хронічною гіпоксією (кисневою недостатністю), порушенням нервової діяльності. Проявами перевтоми є головний біль, підвищена стомлюваність, дратівливість, нервозність, порушення сну, а також такі захворювання як вегето-судинна дистонія, артеріальна гіпертонія, виразкова хвороба, ішемічна хвороба серця, інші професійні захворювання. Втома характеризується фізіологічними та психічними показниками її розвитку. Фізіологічними показниками розвитку втоми є артеріальний кров'яний тиск, частота пульсу, систолічний і хвилинний об'єм крові, зміни у складі крові. Психічними показниками розвитку втоми є: погіршення сприйняття подразників, внаслідок чого працівник окремі подразники зовсім не сприймає, а інші сприймає із запізненням; зменшення здатності концентрувати увагу, свідомо її регулювати; посилення мимовільної уваги до побічних подразників, які відволікають працівника від трудового процесу; погіршення запам'ятовування та труднощі пригадування інформації, що знижує ефективність професійних знань; сповільнення процесів мислення, втрата їх гнучкості, широти, глибини і критичності; підвищення дратівливості, поява депресивних станів; порушення сенсомоторної координації, збільшення часу реакцій на подразники; зміни частоти слуху, зору[30].

3.5 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описані чотири питання, по два із безпеки життєдіяльності та охорони праці.

Перше питання алгоритм надання першої долікарської допомоги постраждалій від електричного струму людині.

Друге питання загальні вимоги безпеки з охорони праці для користувачів ПК. Якщо ж говорити про вимоги безпеки з охорони праці для користувачі ПК безпосередньо перед початком роботи, то останні повинні переконатись у відсутності пошкоджень обладнання та електро мережі, а у разі їх виявлення повідомити відповідальну особу і ні в якому разі не приступати до роботи.

Третє питання методи боротьби з монотонністю праці на виробництві. Важливим засобом боротьби з монотонністю є чергування операцій, кожна з яких є монотонною.

Четверте питання вплив виробничого середовища на працездатність та здоров'я користувачів комп'ютерів. Так як мобільний застосунок розробляється розробниками, тому важливо взяти до уваги методи оцінки ефективності заходів як ефективність діяльності людини базується на рівні психічної напруги, яка прямо пропорційна складності завдання, що дозволить надати метрику щодо успішності використання застосунку.

ВИСНОВОК

У цій дипломній роботі успішно розроблено та проектувано мобільний застосунок для динаміки іноземних валют. Мета полягала у створенні інноваційного застосунок, який забезпечуватиме користувачам легкий доступ до актуальних даних про курси іноземних валют, а також надаватиме широкий спектр функціональних можливостей.

Під час розробки застосунок, був проведений аналіз сучасних мобільних додатків фінансової сфери та валютних сервісів, вивчено ринкові тенденції та використано передові методи та інструменти розробки мобільних додатків. Розроблено зручний інтерфейс користувача, побудована ефективна база даних, реалізовані алгоритми оновлення та аналізу даних, а також впроваджено механізми безпеки та захисту інформації.

Мобільний застосунок дозволяє користувачам отримувати актуальну інформацію про динаміку іноземних валют у режимі реального часу. Він також надає можливість аналізувати цю динаміку, отримувати сповіщення про зміни курсів, а також виконувати конвертацію валют за актуальними курсами. Додаток може бути корисним для широкого спектру користувачів, від індивідуальних фінансових учасників до бізнес-сектору та подорожуючих.

В результаті проведених досліджень та розробки дійшовши висновку, що мобільний застосунок є функціональним, ефективним та зручним інструментом для отримання та аналізу актуальних даних про динаміку іноземних валют. Він відповідає потребам сучасного фінансового ринку та сприятиме поліпшенню фінансової грамотності та прийняттю обґрунтованих фінансових рішень.

Завдяки дослідженням та розробці мобільного застосунку, внесено свій внесок у сферу фінансових технологій та забезпечили зручний інструмент для користувачів, що працюють з іноземними валютами. Виникли сподівання, що пророблена робота знайде практичне застосування та буде корисною для багатьох користувачів, допомагаючи їм приймати обґрунтовані фінансові рішення та досягати своїх цілей у глобальному фінансовому середовищі.

ПЕРЕЛІК ДЖЕРЕЛ

1. Theresa N. Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps Second Edition / Neil Theresa. – New York: O'Reilly Media, Inc., 2014. – 408с.
2. Adrian M. Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps / Mendoza Adrian. – San Francisco: Morgan Kaufmann, 2013. – 248 с. – (Morgan Kaufmann).
3. Mike C. User Stories Applied: For Agile Software Development March 2004 / Cohn Mike. – Redwood: Addison Wesley Longman, 2004. – 350 с. – (Addison Wesley Longman)
4. Rick K. Software Architecture in Practice October / K. Rick, C. Paul, B. Len. – San Francisco: Addison Wesley Longman, 2012. – 640 с. – (Addison-Wesley Professional).
5. Martin K. Designing Data-Intensive Applications / Kleppmann Martin. – San Francisco: O'Reilly Media, Inc., 2017. – 491 с. – (O'Reilly Media, Inc.).
6. Learn Firebase fundamentals [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://firebase.google.com/docs/guides?hl=en>.
7. Quick start guide [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html#summary>.
8. Getting Started [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://code.visualstudio.com/docs>.
9. Meta O. React-Native Introduction [Електронний ресурс] / Open Meta // Meta Platforms. – 2023. Режим доступу до ресурсу: <https://reactnative.dev/docs/getting-started>.
10. Android Studio Developer Guides [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://developer.android.com/guide>.
11. MinFin [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://minfin.com.ua>.

12. FastAPI [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://fastapi.tiangolo.com>.
13. Introduction Uvicorn [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.uvicorn.org>.
14. Adobe Photoshop Help [Електронний ресурс] // adobe. – 2019. – Режим доступу до ресурсу: https://helpx.adobe.com/pdf/photoshop_reference.pdf.
15. JavaScript-бібліотека React [Електронний ресурс]. Режим доступу до ресурсу: <https://uk.reactjs.org>.
16. Посібник: знайомство з React [Електронний ресурс]. Режим доступу до ресурсу: <https://uk.reactjs.org/tutorial/tutorial.html>.
17. Expo Go [Електронний ресурс] // Expo. – 2021. – Режим доступу до ресурсу: <https://docs.expo.dev/get-started/expo-go/>.
18. TouchableOpacity [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://reactnative.dev/docs/touchableopacity>.
19. Getting Started [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://axios-http.com/docs/intro>.
20. Документація бібліотеки React.js [Електронний ресурс] – Режим доступу: <https://reactjs.org/>.
21. Миченко І. М. Забезпечення життєдіяльності людини в навколишньому середовищі / І. М. Миченко. – Кіровоград, 1998. – 292 с.
22. Джигирей В. С. Безпека життєдіяльності : навч. посіб. – 3-тє вид. – Львів: Афіша, 2000. – 256 с.
23. Кобилянський О. В. Охорона праці в робітничій професії: навч. посіб. / О. В. Кобилянський, В. В. Присяжнюк, В. В. Богачук. – Вінниця: ВНТУ, 2009. – 144 с.
24. Бедрій Я. І. Основи охорони праці користувачів персональних комп'ютерів / Я. І. Бедрій. – Тернопіль: Навчальна книга – Богдан, 2014. – 144 с.
25. Гандзюк М. П. Основи охорони праці : навч. посіб. / М. П. Гандзюк, Є. П. Желібо, О. М. Халімовський. – Київ: Каравелла, 2004. – 407 с.

26. Основні заходи, спрямовані на запобігання монотонності і підвищення змістовності праці [Електронний ресурс]. – Режим доступу до ресурсу: <https://studentbooks.com.ua/content/view/951/76/1/4>
27. Stellman J. M. Encyclopedia of Occupational Health and Safety (4th ed.). / J. M. Stellman. – New York: International Labour Organization., 2015. – 1000с.
28. Gavriel S. Human Factors and Ergonomics: A Briefing for Senior Managers. / S. Gavriel, K. Waldemar. – New York: O'Reilly Media, Inc., 2021. – 1527 с.
29. Computer Workstations eTool [Електронний ресурс] // Occupational Safety and Health Administration (OSHA). – 2011. – Режим доступу до ресурсу: <https://www.osha.gov/>.
30. Ergonomics Society [Електронний ресурс] // Ergonomics Society. – 2019. – Режим доступу до ресурсу: <https://www.ergonomics.org.uk/>.

ДОДАТКИ

Лістинги до кваліфікаційної роботи

Лістинг 1 — layout.jsx

```
import { StyleSheet, View } from 'react-native';
import { useNavigate } from 'react-router-native';
import LayoutButton from './layoutbutton'

const Layout = () => {
  const navigate = useNavigate()

  return (
    <View style={styles.layout}>
      <LayoutButton
image={require("../..../assets/home1.png")} onPress={() =>
{navigate('/')}}/>
      <LayoutButton
image={require("../..../assets/currencies1.png")} onPress={() =>
{navigate('/currencies')}}/>
      <LayoutButton image={require("../..../assets/rate.png")}
onPress={() => {navigate('/rate/-')}}/>
      <LayoutButton
image={require("../..../assets/settings.png")} onPress={() =>
{navigate('/settings')}}/>
      <LayoutButton image={require("../..../assets/info.png")}
onPress={() => {navigate('/info')}}/>
    </View>

  )

};

const styles = StyleSheet.create({
  layout: {
    alignItems: 'center',
    flexDirection: 'row',
    position: 'absolute',
```

```

    bottom: 0,
    zIndex: 1,
    backgroundColor:'white'

  }
});
export default Layout;

```

ЛІСТИНГ 2 — layoutbutton.jsx

```

import { TouchableOpacity, StyleSheet, Image, View } from 'react-
native';

```

```

const LayoutButton = ({onPress, image}) => {
  return (
    <View style={styles.container}>
      <TouchableOpacity onPress={onPress}>
        <Image source={image} style={styles.image} />
      </TouchableOpacity>
    </View>
  )
};

const styles = StyleSheet.create({
  image: {
    height: 64,
    width: 64
  },
  container: {
    flex: 1
  }
});

```

```

export default LayoutButton;

```

ЛІСТИНГ 3 — header.jsx

```

import { StyleSheet, View, Text } from 'react-native';
import Logo from './logo';

```

```

const Header = () => {
  return (
    <View style={styles.header}>
      <Logo />
      <Text style={styles.text}>EXRATEDES</Text>
    </View>
  )
};

const styles = StyleSheet.create({
  header: {

    flexDirection: 'row',
    position: 'absolute',
    top: 0,
    alignItems: 'center'
  },
  text: {
    fontSize: 24
  }
});
export default Header;

```

Лістинг 4 — logo.jsx

```

import { Image } from 'react-native';
export default function Logo() {
  return <Image
    source={require('../assets/rate_LOGO.png')}
    style={{width: 50, height: 50}}
  />;
}

```

Лістинг 5 — rate_dropdown.jsx

```

import React, { useEffect, useState } from "react";
import axios from "axios";
import SelectDropdown from "react-native-select-dropdown";

```

```

import { StyleSheet } from "react-native";
import { useNavigate, useParams } from "react-router-native";

function RateDropDown({value, first}) {
  const [options, setOptions] = useState();
  const [selected, setSelected] = useState();
  const params = useParams()
  const navigate = useNavigate()

  useEffect(() => {
    async function fetchData() {
      const response = await axios.get("https://dip-exratedes-
vladik02schymkiv.vercel.app/currencies");
      const currencies = response.data;
      setOptions( Object.entries(currencies));
      setSelected(Object.keys(currencies).indexOf(value))
    }
    fetchData();
  }, []);

  return (
    <SelectDropdown
      data={options}
      onSelect={(selectedItem, index) => {
        const rate=first?`${selectedItem[0]}-
${params.rate.slice(4,7)}`:`${params.rate.slice(0,3).split('-
').join('')}-${selectedItem[0]}`
        navigate(`/rate/${rate}`)
      }}
      defaultValueByIndex={selected}
      defaultButtonText = 'Виберіть валюту'

      buttonStyle={styles.button}

      buttonTextAfterSelection={(selectedItem, index) => {
        return selectedItem[1].name
      }}
      rowTextForSelection={(item, index) => {
        return item[1].name

```



```

        }}

    />
  );
}

export default RateDropDown;

const styles = StyleSheet.create({
  button: {
    width: '100%'
  }
})

```

Лістинг 6 — currencies.jsx

```

import React, { useState, useEffect } from 'react';
import { Text, View, StyleSheet, Dimensions, ScrollView,
TouchableOpacity } from 'react-native';
import axios from "axios";
import { useNavigate } from 'react-router-native';

const screenWidth = Dimensions.get("window").width;

export default function Currencies() {
  const navigate = useNavigate()
  const [currencies, setCurrencies] = useState([]);
  useEffect(() => {
    async function fetchData() {
      const response = await axios.get("https://dip-exratedes-
vladik02schymkiv.vercel.app/currencies");
      const currencies = response.data;
      setCurrencies(currencies);
    }
    fetchData();
  }, []);
  const screenHeight = Dimensions.get('window').height

  return (

```

```

<View style={{height: screenHeight}}>
  <ScrollView >
    {Object.entries(currencies).map(item => (
      <TouchableOpacity key={item[0]} style={styles.container}
onPress={() => {navigate(`/currency/${item[0]}`)}} >
        <Text style={styles.text}>{item[1].name}</Text>
      </TouchableOpacity >
    ))}
  </ScrollView>
</View>
)
}

```

```

const styles = StyleSheet.create({
  container: {
    width: screenWidth,
    paddingHorizontal: 5,
    borderBottomWidth: 2
  },
  text: {
    fontSize: 20,
  }
});

```

Лістинг 7 — currency.jsx

```

import React, { useState, useEffect } from 'react';
import { Text, View } from 'react-native';
import { useParams } from 'react-router-native';
import axios from "axios";

export default function Currency() {
  const [currency, setCurrency] = useState([]);
  const params = useParams()
  useEffect(() => {
    async function fetchData() {
      const response1 = await axios.get(`https://dip-exratedes-
vladik02schymkiv.vercel.app/currency/${params.code}`);

```

```

        const response2 = await axios.get(`https://dip-exratedes-
vladik02schymkiv.vercel.app/description/${params.code}`);
        const data = response1.data
        data.description = response2.data
        setCurrency(data);

    }
    fetchData();
}, []);

return (
  <View>
    <Text>Назва: {currency.name}</Text>
    <Text>Код: {params.code}</Text>
    {currency.symbol ? (
      <Text>Символ: {currency.symbol}</Text>
    ) : ( null )}
    {currency.region ? (
      <Text>Регіон: {currency.region}</Text>
    ) : ( null )}
    {currency.description ? (
      <Text>Опис: {currency.description}</Text>
    ) : ( null )}
  </View>
);
}

```

ЛІСТИНГ 8 — home.jsx

```

import { View, Text, StyleSheet, TouchableOpacity } from 'react-
native';

import { LinearGradient } from 'expo-linear-gradient';
import { useNavigate } from 'react-router-native';
const colors = ['#1FEE0F', '#3EDD1E', '#7CBB3C', '#BAA85A', '#FF3300']
export default function Home() {
  const navigate = useNavigate()
  return (
    <View>

```

```

        <LinearGradient style={styles.verticalContainer}
colors={colors}>
        <Text>Exratedes - це програма для візуалізації курсу
валют.</Text>
        </LinearGradient>
        <Text style={styles.header}>Новини</Text>
        <LinearGradient style={styles.verticalContainer}
colors={colors}>
        <Text>Нещодавно додані ВТ</Text>
        <TouchableOpacity onPress={()=>{navigate('/currency/USD')}}>
        <Text style={styles.text}>
        <Text style={styles.firstLetter}>$ </Text>
        Долар США
        </Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={()=>{navigate('/currency/UAH')}}>
        <Text style={styles.text}>
        <Text style={styles.firstLetter}>>₴ </Text>
        Українська гривня
        </Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={()=>{navigate('/currency/EUR')}}>
        <Text style={styles.text}>
        <Text style={styles.firstLetter}>>€ </Text>
        Євро
        </Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={()=>{navigate('/currency/JPY')}}>
        <Text style={styles.text}>
        <Text style={styles.firstLetter}>>¥ </Text>
        Японська Єна
        </Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={()=>{navigate('/currency/CHF')}}>
        </TouchableOpacity>
        <TouchableOpacity onPress={()=>{navigate('/currency/GBP')}}>
        <Text style={styles.text}>
        <Text style={styles.firstLetter}>>£ </Text>

```

```

        Фунт Стерлінгів
    </Text>
</TouchableOpacity>

</LinearGradient>
<Text style={styles.header}>Популярні пари</Text>
<View style={styles.horizontalContainer}>
    <LinearGradient style={styles.verticalContainer}
colors={colors}>
        <TouchableOpacity onPress={()=>{navigate('/rate/UAH-USD')}}
style={styles.view}>
            <Text>UAH/USD</Text>
            <Text></Text>
            <Text>36.95/1</Text>
            <Text></Text>
            <Text>Візуалізація</Text>
            <Text>курсу</Text>
        </TouchableOpacity>
    </LinearGradient>
    <LinearGradient style={styles.verticalContainer}
colors={colors}>
        <TouchableOpacity onPress={()=>{navigate('/rate/UAH-EUR')}}
style={styles.view}>
            <Text>UAH/EUR</Text>
            <Text></Text>
            <Text>40.36/1</Text>
            <Text></Text>
            <Text>Візуалізація</Text>
            <Text>курсу</Text>
        </TouchableOpacity>
    </LinearGradient>
    <LinearGradient style={styles.verticalContainer}
colors={colors}>
        <TouchableOpacity onPress={()=>{navigate('/rate/UAH-GBP')}}
style={styles.view}>
            <Text>UAH/GBP</Text>
            <Text></Text>
            <Text>45.83/1</Text>
            <Text></Text>

```

```
        <Text>Візуалізація</Text>
        <Text>курс</Text>
      </TouchableOpacity>
    </LinearGradient>
  </View>

</View>;

}
```

```
const styles = StyleSheet.create({
  verticalContainer: {
    marginHorizontal: 17,
    alignSelf: "center",
    borderWidth: 2,
    borderColor: 'black',
    borderRadius: 5,
    padding: 3,
  },
  horizontalContainer: {
    alignSelf: "center",
    justifyContent: 'center',
    alignItems: 'center',
    flexDirection: 'row'
  },
  header: {
    fontWeight: 'bold',
    marginTop: 15,
    marginBottom: 5,
    marginLeft: 5,
    fontSize: 23
  },
  text: {
    fontSize: 16,
    alignSelf: "flex-start"
  },
  firstLetter: {
    fontSize: 48,
```

```

    fontWeight: 'bold',

  },
  view: {
    justifyContent: 'center',
    alignItems: 'center',
  }
});

```

Лістинг 9 — info.jsx

```

import { Text, View, StyleSheet } from 'react-native';
export default function Info() {

  return (<View style={styles.view}>
    <Text>Сторінка про розробників. Дана програма була розроблена
для виконання дипломної роботи.</Text>
    <Text>1 Шимків Владислав Сергійович: База даних, Парсер,
Дизайн, Backend. </Text>
    <Text>2 Ципняк Денис Олександрович: Frontend.</Text>
  </View>

  )
}

const styles = StyleSheet.create({
  view: {
    marginLeft: 10,
  }
});

```

Лістинг 10 — rate.jsx

```

import { Text, Button, View, StyleSheet, Dimensions } from 'react-
native';
import React, { useState } from "react";
import RateDropDown from '../components/rate_dropdown';
import { LineChart } from "react-native-chart-kit";
import divideObjects from '../utils/divide_objects';
import getRateChartData from '../utils/rate_chart_data';

```

```

import axios from "axios";
import { useParams } from 'react-router-native';

const screenWidth = Dimensions.get("window").width;

export default function Rate() {
  const [rateText, setRateText] = useState();
  const [chart, setChart] = useState();

  const params = useParams()
  const [first, second]=params.rate.split('-')

  async function get_rate_to_UAH(currency_code) {
    const response = await axios.get(`https://dip-exratedes-
vladik02schymkiv.vercel.app/rate/${currency_code}`);
    return response.data
  }

  async function get_rate(){
    if (first&&second){
      const rate= divideObjects(await get_rate_to_UAH(first),
await get_rate_to_UAH(second));
      setRateText(`1 ${first} =
${Object.values(rate)[Object.values(rate).length-1].toFixed(2)}
${second}`);
      setChart(
        <LineChart
          data={getRateChartData(rate)}
          width={screenWidth}
          //withVerticalLabels={false}
          verticalLabelRotation={90}
          height={400}
          chartConfig={{color: (opacity = 1) => `rgba(100, 100,
100, ${opacity})`, backgroundGradientFrom: 'pink',
          backgroundGradientTo: 'lightgreen',
          }}
        />
      )
    }
  }
}

```



```

    }
  }

  return (
    <View style={styles.container}>
      <RateDropDown value={first} first={true}/>
      <RateDropDown value={second} first={false}/>
      <Button title='Дізнатись курс' onPress={get_rate}/>
      <Text style={styles.rateText}> {rateText} </Text>
      {chart}
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    width: screenWidth,

  },
  body: {
    position: 'absolute',
    top: 50
  },
  button: {
    width: '100%'
  },
  rateText:{
    marginVertical: 10,
    fontSize: 20
  }
});

```

Лістинг 11 — settings.jsx

```

import { StyleSheet, Text, View } from 'react-native';
import SelectDropDown from "react-native-select-dropdown";
const languages=['Українська', 'Англійська']
export default function Settings() {

```

```

return (
<View style={styles.view}>
  <Text style={styles.title}>Налаштування</Text>
  <Text style={styles.subtitle}>Мова</Text>
  <SelectDropdown
    data={languages}
    onSelect={(selectedItem, index) => {}}
    buttonTextAfterSelection={(selectedItem, index) => {
      return selectedItem
    }}
    rowTextForSelection={(item, index) => {
      return item
    }}
    defaultValueByIndex={0}
  />
</View>
);
}

```

```

const styles = StyleSheet.create({
  title: {
    fontSize: 20,
    fontWeight: 'bold',
  },
  subtitle: {
    fontSize: 20,
  },
  view: {
    marginLeft: 10,
  }
})

```

Лістинг 12 — divide_objects.jsx

```

export default function divideObjects(obj1, obj2) {
  var result = {};
  for (var prop in obj1) {
    if (obj2.hasOwnProperty(prop)) {
      result[prop] = obj1[prop] / obj2[prop];
    }
  }
}

```

```

    }
    return result;
}

```

Лістинг 13 — rate_chart_data.jsx

```

export default function getRateChartData(rate) {
  return {
    labels: Object.keys(rate),
    datasets: [
      {
        data: Object.values(rate)
      }
    ],
    legend: ["Курс"]
  }
}

```

Лістинг 14 — App.js

```

import { StyleSheet, View, StatusBar, SafeAreaView } from 'react-
native';
import Home from './pages/home';
import Currencies from './pages/currencies';
import Rate from './pages/rate';
import Info from './pages/info';
import Settings from './pages/settings';
import { NativeRouter, Route, Routes } from 'react-router-native';
import Layout from './components/layout/layout';
import Header from './components/header';
import Currency from './pages/currency';

export default function App() {

  return (
    <View style={styles.container}>
      <SafeAreaView style={styles.container}>
        <NativeRouter>
          <Header/>
          <View style={styles.body}>

```

```

    <Routes>
      <Route path="/" element={<Home/>} />
      <Route path="/currencies" element={<Currencies/>} />
      <Route path="/currency/:code" element={<Currency/>}
/>

      <Route path="/rate/:rate" element={<Rate/>} />
      <Route path="/settings" element={<Settings/>} />
      <Route path="/info" element={<Info/>} />
    </Routes>
  </View>
  <Layout/>
  </NativeRouter>
  </SafeAreaView>
  <StatusBar style="auto" />
</View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
  },
  body: {
    position: 'absolute',
    top: 50
  }
});

```



Логотип мобільного застосунку “Exratedes”



Головна

Іконка “Головна”



Валюти

Іконка “ВТ”



Інше

Іконка “Інше”



Налаштування

Іконка “Налаштування”



Іконка “Прапор/тризуб”