

Полтавський університет економіки і торгівлі
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА

«__» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ (Android, IOS)

РОЗКЛАДУ ПУЕТ»

зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня бакалавр

Виконавець роботи Кравченко Родіон Юрійович

_____ «__» _____ 202_ р.
(підпис)

Науковий керівник к.ф.-м.н., Ольховська Олена Володимирівна

_____ «__» _____ 202_ р.
(підпис)

Рецензент

ПОЛТАВА 2023

ЗАТВЕРДЖУЮ

Завідувач кафедри __ Олена ОЛЬХОВСЬКА
«__» вересня 202_р.

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ
КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Розробка мобільного додатку (Android, IOS) розкладу ПУЕТ»

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Кравченко Родіон Юрійович

Затверджена наказом ректора № _____-Н від «__» _____202_р.

Термін подання студентом роботи «__» _____202_р.

Вихідні дані до кваліфікаційно роботи: публікації з теми, навчальні тренажери в дистанційних курсах з комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовна постановка

1.2 Модель задачі та її характеристики

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

2.2 Позитивні аспекти оглянутих робіт

2.3 Вади розробок з оглянутих робіт

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття React Native

3.2 Алгоритм роботи розкладу

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки програмного забезпечення розкладу

4.2 Опис програми

4.3 Необхідна користувачу програми інструкція

ВИСНОВКИ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	ППП, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постанова задачі	Ольховська О.В		
Інформаційний огляд	Ольховська О.В		
Теоретична частина	Ольховська О.В		
Практична реалізація	Ольховська О.В		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» __ 202_ р.

Здобувач вищої освіти _____ Кравченко Родіон Юрійович

Науковий керівник _____ к. ф.-м. н., Ольховська О.В

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____

(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » 202_ р.

Секретар ЕК _____

(підпис)

(ініціал та прізвище)

Затверджую

Зав. кафедрою _____

к. ф.-м. н. Олена ОЛЬХОВСЬКА

«__» _____ 202_ р.

Погоджено

Науковий керівник _____

к. ф.-м. н. Олена ОЛЬХОВСЬКА

«__» _____ 202_ р.

План

кваліфікаційної роботи на тему «Розробка мобільного додатку (android, ios) розкладу пует»

зі спеціальності 122 Комп'ютерні науки

освітня програма 122 Комп'ютерні науки

ступеня бакалавр

Прізвище, ім'я, по батькові Кравченко Родіон Юрійович

ВСТУП

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовна постановка

1.2 Модель задачі та її характеристики

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

2.2 Позитивні аспекти оглянутих робіт

2.3 Вади розробок з оглянутих робіт

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття React Native

3.2 Алгоритм роботи розкладу

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки програмного забезпечення розкладу

4.2 Опис програми

4.3 Необхідна користувачу програми інструкція

ВИСНОВКИ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобувач вищої освіти _____ Родіон КРАВЧЕНКО

«__ » _____ 202_ р.

РЕФЕРАТ

Записка: 45 с., 6 рис., 5 таблиць, 1 додаток, 13 джерел.

РОЗКЛАД УНІВЕРСИТЕТУ, МОБІЛЬНИЙ ДОДАТОК, REACT NATIVE.

Об'єкт розробки - розробка мобільного додатку з використанням фреймворку React Native для операційних систем Android та iOS.

Мета роботи - проектування та програмна реалізація мобільного додатку на основі фреймворку React Native, який надає користувачам зручний та ефективний інструмент для вирішення їх потреб і завдань у сфері мобільних технологій.

Методи дослідження - в ході дослідження та розробки були використані методи аналізу вимог, проектування інтерфейсу, розробки мобільних додатків на основі фреймворку React Native, тестування та збір фідбеку від користувачів.

Даний розроблений мобільний додаток на основі фреймворку React Native призначений для пошуку розкладу занять в університеті. Додаток пропонує користувачеві зручний інтерфейс для введення параметрів пошуку, таких як початкова та кінцева дата, викладач, аудиторія та група. Крім того, в додатку реалізовано відображення календаря для вибору дати.

ЗМІСТ

РЕФЕРАТ	5
ВСТУП	8
1 ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 Змістовна постановка.....	10
1.2 Модель задачі та її характеристики.....	11
2 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	13
2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання.....	13
2.2 Позитивні аспекти оглянутих робіт	14
2.3 Вади розробок з оглянутих робіт.....	15
3 ТЕОРЕТИЧНА ЧАСТИНА	18
3.1 Поняття React Native	18
3.2 Алгоритм роботи розкладу	20
3.3 Обґрунтування вибору програмних засобів для реалізації завдання роботи	22
4 ПРАКТИЧНА ЧАСТИНА	24
4.1 Опис розробки програмного забезпечення розкладу.....	24
4.2 Опис програми	26
4.3 Необхідна користувачу програми інструкція.....	28
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34
ДОДАТОК А.....	36

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ**

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів, термінів
CLI	Command Line Interface
RAD	Rapid Application Development
QA	Quality Assurance (якість та забезпечення якості)
JavaScript	Високорівнева, інтерпретована мова програмування, яка використовується для створення динамічних інтерактивних веб-сторінок.
React	JavaScript-бібліотека для розробки інтерфейсів користувача.
React Native	Відкрита фреймворк-платформа для розробки мобільних додатків, яка дозволяє використовувати JavaScript та бібліотеку React для побудови нативних додатків для платформ Android та iOS.
Expo	Фреймворк та платформа для розробки мобільних додатків, яка спрощує процес створення, тестування та розгортання додатків для платформ Android, iOS та WEB.

ВСТУП

У сучасному світі розвиток мобільних технологій та програмного забезпечення набуває все більшої актуальності і значущості. Зростання популярності мобільних пристроїв серед студентів та викладачів університетів створює потребу у зручному та ефективному способі доступу до інформації про розклад занять. З метою вирішення цієї задачі, цей проект пропонує розробку мобільного додатку для розкладу університету на платформі React Native.

Мета цього проекту полягає в створенні зручного та функціонального мобільного додатку, який дозволить студентам та викладачам легко отримувати доступ до актуального розкладу занять безпосередньо зі своїх мобільних пристроїв. Це дозволить оптимізувати процес організації навчального процесу, покращити комунікацію між студентами та викладачами, а також забезпечити швидкий доступ до необхідної інформації щодо змін у розкладі або інших повідомлень.

Об'єктом дослідження є процес розробки мобільного додатку для розкладу університету на платформі RN. Предметом дослідження є функціональні можливості додатку, його інтерфейс, надійність та ефективність роботи.

Для досягнення поставленої мети використовуватимуться такі методи дослідження: аналіз вимог до розкладу університету, дослідження існуючих рішень та аналогів, проектування та реалізація мобільного додатку на платформі React Native, тестування та оцінка його продуктивності.

Цей проект вносить свій внесок у галузь розробки мобільних додатків та полегшує організацію навчального процесу в університетах. Новизна цього додатку полягає у використанні сучасних технологій розробки, таких як React

Native, для створення зручного та доступного інструменту для отримання розкладу занять.

Результати цього дослідження та розробки мають як теоретичне, так і прикладне значення. Теоретично, результати проекту сприятимуть покращенню розуміння процесу розробки мобільних додатків на платформі React Native. Прикладно, цей додаток забезпечить студентам та викладачам зручний та швидкий доступ до актуальної інформації про розклад занять, що полегшить їхню повсякденну діяльність.

Застосування результатів цього проекту вже може бути реалізоване в університетах, які бажають покращити організацію навчального процесу та забезпечити зручний доступ до розкладу занять для своїх студентів та викладачів. Проект може бути адаптований та розширений відповідно до конкретних потреб кожного навчального закладу.

1 ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовна постановка

Створити додаток, який надає користувачу можливість швидкого та зручного пошуку розкладу занять. Додаток повинен дозволяти вводити критерії пошуку, такі як дата, викладач, аудиторія та група, і відображати відповідний розклад занять, що відповідає цим критеріям.

Функціональні вимоги:

1. Вхід в додаток:

- Користувач повинен мати можливість ввести своє ім'я користувача та пароль для входу в додаток.

- Після успішного входу користувач повинен бути перенаправлений на сторінку розкладу занять.

2. Форма пошуку розкладу:

- Додаток повинен містити форму пошуку, де користувач може вказати критерії для пошуку розкладу.

- Критерії пошуку повинні включати початкову та кінцеву дату, викладача, аудиторію та групу.

- Користувач повинен мати можливість вибирати дати за допомогою календаря.

- Кнопка "Пошук" повинна запускати процес пошуку розкладу занять за введеними критеріями.

3. Очищення форми:

- Користувач повинен мати можливість очистити форму пошуку та скинути всі введені значення.

- Кнопка "Очистити" повинна скидати введені значення і повертати форму пошуку до початкового стану.

4. Навігація в додатку:

- Додаток повинен забезпечувати навігацію між сторінками, зокрема зі сторінки входу на сторінку розкладу занять.

1.2 Модель задачі та її характеристики

Модель задачі описує основні сутності та зв'язки між ними, що використовуються в програмі. Характеристики задачі визначають особливості її реалізації та функціональні можливості програми.

Модель задачі:

1. Користувач:

- Ім'я користувача (username): рядок, що ідентифікує користувача в системі.
- Пароль (password): рядок, який використовується для авторизації користувача в системі.

2. Форма пошуку розкладу:

- Початкова дата (startDate): дата, яка визначає початок періоду для пошуку розкладу занять.
- Кінцева дата (endDate): дата, яка визначає кінець періоду для пошуку розкладу занять.
- Викладач (teacher): рядок, який вказує на обраного викладача.
- Аудиторія (classroom): рядок, який вказує на обрану аудиторію.
- Група (group): рядок, який вказує на обрану групу.

3. Розклад занять:

- Дата (date): дата проведення занять.
- Час (time): час проведення занять.

- Викладач (teacher): ім'я викладача, який веде заняття.
- Аудиторія (classroom): номер аудиторії, де проводяться заняття.
- Група (group): назва групи, для якої проводяться заняття.

Характеристики задачі:

1. Авторизація користувача:

- Користувач вводить своє ім'я користувача та пароль для входу в додаток.
- Після успішної авторизації користувач отримує доступ до форми пошуку розкладу.

2. Пошук розкладу занять:

- Користувач може вказати початкову та кінцеву дати для визначення періоду пошуку.
- Користувач може обрати викладача, аудиторію та групу для фільтрації результатів пошуку.

3. Очищення форми пошуку:

- Користувач може натиснути кнопку "Очистити" для скидання всіх введених значень в формі пошуку.
- Після очищення форма повертається до початкового стану.

4. Навігація в додатку:

- Користувач може переходити між сторінками в додатку, зокрема зі сторінки входу на сторінку розкладу занять.

5. Вихід з додатку:

- Користувач може вийти з додатку, закривши його або натиснувши відповідну кнопку.

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

На сьогоднішній день існує велика кількість мобільних додатків для отримання розкладу університету, реалізованих на різних платформах. В цьому розділі розглянемо огляд деяких робіт з аналогічними завданнями та реалізацією, які вже існують на ринку.

Один з найпоширеніших підходів до розробки мобільних додатків для розкладу університету полягає використанні веб-технологій, таких як HTML, CSS та JavaScript. Цей підхід дозволяє створювати кросплатформенні додатки, які можуть працювати на різних пристроях та операційних системах. Проте, такі додатки можуть мати обмежені можливості та швидкість роботи через залежність від веб-браузера.

Також існують додатки, розроблені на платформі Android або iOS, використовуючи мови програмування, такі як Java або Swift відповідно. Ці додатки можуть мати високу продуктивність та доступ до функцій, що надаються операційними системами, однак потребують розробки окремо для кожної платформи, що може збільшити витрати на розробку та підтримку.

Останнім часом дедалі популярнішим стає використання фреймворків гібридної розробки, таких як React Native. Ці фреймворки дозволяють розробляти мобільні додатки за допомогою веб-технологій, але з використанням нативних компонентів та можливостей операційної системи. Вони поєднують в собі переваги кросплатформенності та продуктивності нативних додатків.

Наприклад, додаток "MySchedule" реалізований на платформі React Native і надає користувачам зручний інтерфейс для отримання розкладу університету. Він має функції, такі як перегляд розкладу занять, отримання сповіщень про

зміни у розкладі, можливість додавати заняття до календаря та інші корисні опції. Цей додаток отримав позитивні відгуки від користувачів за його зручність та функціональність.

На основі проведеного огляду можна зробити висновок, що використання фреймворку React Native для розробки мобільного додатку для розкладу університету є актуальним та перспективним підходом. Він поєднує переваги кросплатформенності та продуктивності нативних додатків, дозволяючи швидко розробляти та підтримувати додаток на різних платформах.

2.2 Позитивні аспекти оглянутих робіт

Під час огляду різних робіт з аналогічними завданнями та реалізацією мобільних додатків для розкладу університету, було виявлено кілька позитивних аспектів, які можна врахувати та використати при розробці нашого додатку на платформі React Native. Нижче наведено деякі з них:

1. Зручний та інтуїтивно зрозумілий інтерфейс: Багато розглянутих додатків мають добре організований та зрозумілий інтерфейс, що спрощує користування та навігацію для користувачів. Це дозволяє швидко зорієнтуватися у розкладі та знаходити необхідну інформацію без зайвих зусиль.

2. Сповіщення про зміни у розкладі: Багато додатків надають можливість отримувати сповіщення про зміни у розкладі занять, такі як перенесення або скасування занять. Це дозволяє користувачам завжди бути в курсі оновлень і уникати непередбачених ситуацій.

3. Інтеграція з календарем: Деякі додатки мають функціонал, що дозволяє додавати заняття до календаря на мобільному пристрої. Це забезпечує

зручний спосіб організації та планування часу для користувачів, оскільки вони можуть отримувати нагадування про наближення занять.

4. Підтримка різних типів розкладу: Деякі додатки надають можливість користувачам налаштувати свій розклад відповідно до індивідуальних потреб. Наприклад, можливість вибору конкретних груп або спеціальностей, розкладу викладачів тощо. Це робить додаток більш гнучким та придатним для різних категорій користувачів.

5. Підтримка офлайн-режиму: Деякі додатки мають функціонал, що дозволяє переглядати розклад занять у офлайн-режимі. Це особливо корисно, коли користувачі перебувають у місцях з обмеженим або відсутнім доступом до Інтернету.

Враховуючи ці позитивні аспекти з оглянутих робіт, ми можемо побудувати наш додаток на React Native з упевненістю, що він буде зручним, функціональним та задовольнить потреби користувачів у розкладі університету.

2.3 Вади розробок з оглянутих робіт

Під час огляду різних робіт з аналогічними завданнями та реалізацією мобільних додатків для розкладу університету було виявлено деякі вади, які варто врахувати та уникнути при розробці нашого додатку на платформі React Native. Нижче наведено деякі з них:

1. Відсутність стабільності: У деяких роботах виявлено проблеми зі стабільністю додатків, такі як затримки в роботі, некоректна відображення інформації, або відмови відповідати на певні дії користувачів. Це може створювати негативний вплив на користувачів і погіршувати їх досвід використання додатку.

2. Обмежені можливості налаштування: Деякі додатки мають обмежені можливості налаштування розкладу, що може бути незручним для користувачів з різними потребами. Наприклад, відсутність можливості вибору конкретних днів тижня або персоналізації сповіщень про зміни у розкладі.

3. Погана сумісність з різними пристроями: У деяких випадках, розроблені додатки не завжди належним чином пристосовані до різних типів мобільних пристроїв, що може призводити до проблем зі зображенням та функціональністю на певних пристроях.

4. Погана сумісність з різними пристроями: У деяких випадках, розроблені додатки не завжди належним чином пристосовані до різних типів мобільних пристроїв, що може призводити до проблем зі зображенням та функціональністю на певних пристроях.

5. Відсутність підтримки оновлень: Деякі додатки не отримують достатню підтримку і оновлення від розробників. Це може призвести до проблем із сумісністю з новими версіями операційних систем та відсутності виправлень помилок і оновлень функціональності.

6. Незадовільний дизайн інтерфейсу: У деяких роботах були виявлені проблеми з дизайном інтерфейсу, такі як незручне розташування елементів, неправильне використання кольорів або незбалансована композиція. Це може вплинути на зручність використання додатку та загальний візуальний досвід користувачів.

Враховуючи вади, виявлені під час огляду робіт з аналогічними завданнями, ми плануємо уникнути цих проблем у нашому додатку, надаючи особливу увагу стабільності, налаштуванням, сумісності з різними пристроями, регулярним оновленням та привабливому дизайну інтерфейсу. Наша мета - створити функціональний та зручний додаток, що задовольнить потреби

користувачів у розкладі університету, забезпечуючи позитивний досвід використання.

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Поняття React Native

React Native - це відкритий фреймворк, призначений для розробки мобільних додатків з використанням мови програмування JavaScript та бібліотеки React. QA на перший погляд, React Native може здатися простою інструментальною бібліотекою або розширеним фреймворком для реалізації Web-моделей на мобільних пристроях. Але це далеко не так - це повноцінний фреймворк, що дозволяє розробку стандартних мобільних додатків.

React Native дозволяє розробляти мобільні додатки для iOS та Android з використанням однієї і тієї ж кодової бази. Це означає, що розробники можуть писати код один раз і використовувати його для розробки додатків для обох платформ. Фреймворк має вбудовану підтримку живого оновлення, що дозволяє бачити зміни в додатку в режимі реального часу, що дуже зручно для розробки та тестування додатків.

React Native базується на ідеї компонентів архітектури, яка розділяє додаток на невеликі компоненти. Це дозволяє розробникам швидко створювати ефективні та повторно використовувані компоненти для своїх додатків. Крім того, React Native дозволяє розробникам використовувати ті ж JavaScript-файли, які використовуються для створення веб-додатків, що дозволяє більш ефективно використовувати ресурси.

Однією з основних переваг React Native є можливість безкоштовно завантажувати та використовувати сторонні бібліотеки, що спрощує розробку та підвищує продуктивність розробників. Фреймворк також має велику кількість вбудованих компонентів, таких як кнопки, форми та повідомлення, що дозволяє розробникам швидко створювати ефективні та естетичні рішення.

При розробці мобільних додатків на React Native, використовуються стандартні технології, такі як CSS, HTML та JavaScript, що дозволяє розробникам зрозуміти особливості роботи React Native, адаптуватись та швидко розробляти уявлення.

Одним з недоліків React Native є те, що додаток не буде написаний на "природніх" мовах, таких як Java або Swift/Objective-C, що дозволяє розробникам отримати максимальне нативне відтворення, однак, завдяки постійній розробці та оптимізації інструментів розробки, це знаходиться в процесі усунення.

React Native дозволяє створювати мобільні додатки з гарною анімацією та відмінною продуктивністю. React Native використовує компонентну архітектуру, що дозволяє створювати повторно використовувані компоненти та знижує час розробки додатків. Крім того, Rapid Application Development (RAD), яку надає React Native, дозволяє розробникам створювати ефективні та високоякісні мобільні додатки, зменшуючи час розробки.

React Native має гнучкішу структуру кодової бази, що дозволяє розробникам дуже швидко пристосовувати та модифікувати додатки. Він також дозволяє розробникам використовувати пакети-менеджери, такі як npm, для збірки та встановлення сторонніх пакетів, що ще більше прискорює розробку додатків.

React Native використовує технологію JavaScript для створення мобільних додатків, що дозволяє розробникам використовувати більш прості та інтуїтивні методи розробки. Це забезпечує більш організований та загальний підхід до розробки мобільних додатків.

React Native є відкритим фреймворком, що дозволяє розробникам вносити зміни та додавати функціональність до вихідного коду. Це означає, що React Native є гнучким та пристосовуючимся до потреб розробників.

Ще однією з переваг React Native є його здатність до відмінного відображення мобільних додатків на різних пристроях. React Native дозволяє більш точно пристосовувати мобільні додатки до конкретних пристроїв та платформ. Це дозволяє створювати мобільні додатки зі змінним вмістом та графікою, що не заповнюють всі екрани різних розмірів пристроїв.

React Native має велику інтернет-спільноту, що дозволяє розробникам отримати експертні поради, пропозиції та ділитися своїм досвідом. У цій общности також розвиваються стандартні та готові рішення для розробки додатків.

3.2 Алгоритм роботи розкладу

Алгоритм роботи додатку на React Native для розкладу університету може включати наступні основні компоненти:

1. Компонент авторизації:
 - Показати форму для введення облікових даних (логін та пароль).
 - Перевірити правильність введених даних.
 - У разі успішної авторизації перейти до головного екрану.
2. Головний екран:
 - Відобразити заголовок та меню навігації.
 - Запропонувати користувачеві вибрати день тижня або перейти до налаштувань.
3. Екран розкладу:
 - Завантажити розклад занять для вибраного дня тижня.
 - Відобразити список занять, включаючи назву предмета, час початку та закінчення, аудиторію та викладача.

- Додати можливість оновити розклад за необхідністю.
4. Екран налаштувань:
- Дати можливість користувачеві змінити налаштування, такі як вибір групи чи факультету, з яким пов'язаний розклад.
 - Зберегти зміни та оновити розклад відповідно до нових налаштувань.
5. Інші можливі компоненти:
- Повідомлення про помилки під час завантаження розкладу або неправильно введених даних.
 - Збереження облікових даних користувача для автоматичного входу.

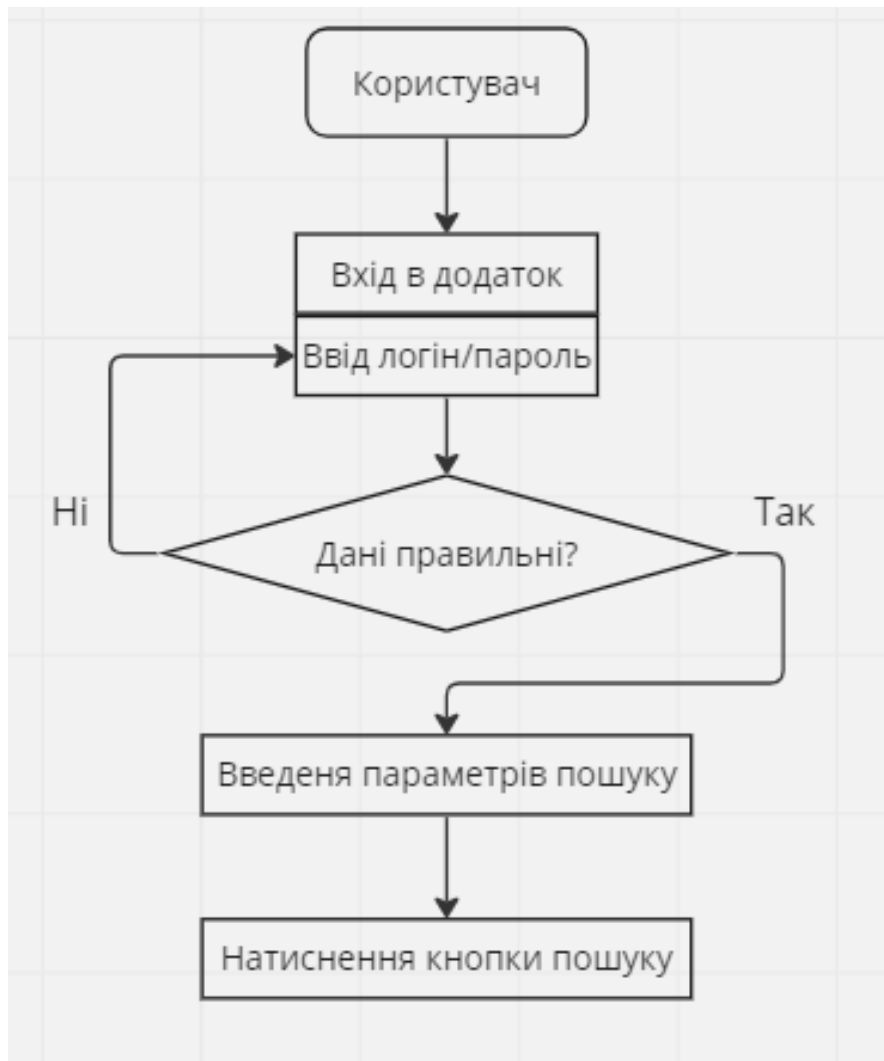


Рис. 3.1 Алгоритм роботи додатку

3.3 Обґрунтування вибору програмних засобів для реалізації завдання роботи

Для створення програмного продукту використані такі технології, як React Native, Expo та Visual Studio Code.

React Native є відкритою платформою для розробки мобільних додатків, яка базується на JavaScript і використовує популярну бібліотеку React для створення інтерфейсу користувача. Основна перевага React Native полягає в тому, що вона дозволяє розробникам створювати переносні мобільні додатки для iOS та Android з використанням одного кодової бази. Це значно зменшує затрати часу та ресурсів, порівняно з розробкою двох окремих додатків для кожної платформи. Враховуючи те, що нашим завданням є створення додатку для розкладу університету, який потребує підтримки обох платформ, вибір React Native є логічним рішенням.

Expo є інструментарієм для розробки мобільних додатків на базі React Native. Його основна перевага полягає в тому, що він надає широкий набір готових компонентів, API та сервісів, які спрощують розробку і розгортання мобільних додатків. Expo також надає можливість швидко переглядати результати розробки в реальному часі на фізичних пристроях або емуляторах. Враховуючи обмежений час реалізації проекту, Expo допоможе прискорити розробку та спростити процес тестування та розгортання додатку.

Visual Studio Code (VS Code) є популярним та потужним інтегрованим середовищем розробки, яке надає широкі можливості для роботи з різними мовами програмування, включаючи JavaScript і React Native. VS Code має багатий функціонал, такий як автодоповнення, підсвічування синтаксису, вбудовану підтримку Git та багато іншого. Він також має широкий вибір

розширень, що полегшують роботу з React Native. VS Code є популярним серед розробників

завдяки своїй продуктивності, зручному інтерфейсу та активній спільноті користувачів.

Узагальнюючи, вибір React Native, Expo та Visual Studio Code для реалізації нашого завдання зумовлений наступними факторами: можливість розробки переносних мобільних додатків для обох платформ за допомогою React Native, швидкість розробки та спрощення процесу розгортання завдяки Expo, а також потужне та зручне середовище розробки, яким є Visual Studio Code.

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки програмного забезпечення розкладу

Розробка програмного забезпечення розкладу університету включає в себе аналіз вимог, проектування, реалізацію, тестування та розгортання додатку.

Перший етап розробки програмного забезпечення - аналіз вимог. Це включає збір і аналіз вимог до функціональності додатку розкладу університету. Вимоги можуть включати такі пункти, як можливості перегляду розкладу, додавання/видалення предметів, повідомлення про зміни в розкладі тощо.

На етапі проектування розробляється архітектура додатку та його інтерфейс користувача. Визначаються компоненти, модулі та їх взаємодія для досягнення визначених вимог. Також вирішуються питання про збереження даних, взаємодію з сервером та інші аспекти розробки.

Після проектування переходимо до реалізації. Використовуючи вибрану технологію розробки, наприклад, React Native, починаємо писати код. Компоненти, модулі та логіка додатку реалізуються згідно з визначеною архітектурою. Звертають увагу на оптимізацію, читабельність та повторне використання коду.

Після реалізації додатку проводиться тестування, щоб переконатися у правильному функціонуванні програми. Тестування може включати модульні тести, інтеграційні тести, функціональні тести та тести користувальницького досвіду (user experience tests). Виявлені помилки виправляються та проводиться повторне тестування.

Розробка програмного забезпечення розкладу університету - це багатоетапний процес, який вимагає аналізу, проектування,

реалізації, тестування та розгортання додатку. Цей процес залежить від вимог проекту та обраної технології розробки.

Фронтенд додатку відповідає за візуальне представлення і взаємодію з користувачем.

Почнемо зі створення нового проекту React Native. Ми можемо використати інструмент командного рядка, такий як Expo або React Native CLI, для створення початкової структури проекту.

Розробка фронтенду включає створення різних компонентів, таких як екрани (screens), кнопки, списки, розкладки тощо. Кожен компонент має відповідний код на React Native, який визначає його зовнішній вигляд та поведінку.

Додаток розкладу університету має різні екрани, такі як головний екран, екран розкладу тощо. Навігація між цими екранами може бути реалізована за допомогою навігаційного стеку (stack navigation) або таб-навігації (tab navigation).

Після реалізації фронтенду додатку важливо провести тестування та налагодження. Це допоможе виявити та виправити помилки, а також перевірити коректність роботи додатку на різних пристроях та розмірах екрану.

Для забезпечення швидкодії та ефективності додатку можна використовувати різні методи оптимізації, такі як підвантаження компонентів по запиті, кешування даних та усунення зайвих рендерінгів.

Цей процес програмної реалізації фронтенду додатку розкладу університету на React Native надає загальний огляд кроків, які потрібно виконати.

4.2 Опис програми

1. Компонент getDatePicker:

- Цей компонент є функціональним компонентом, який відповідає за відображення вибору дати.

- Вхідні параметри: `str` (рядок) - текстовий опис, який буде відображатися поряд з кнопкою вибору дати.

- Стани: `open` (логічне значення) - відповідає за відкриття/закриття модального вікна з вибором дати. `date` (рядок) - обрана користувачем дата.

- Функції: `handleOnPress` - функція, яка викликається при натисканні на кнопку вибору дати. Вона змінює стан `open`, щоб відкрити або закрити модальне вікно з вибором дати. `handleChange` - функція, яка викликається при зміні вибраної дати. Вона оновлює стан `date` з вибраною користувачем датою.

- Відображення: компонент містить рядок тексту (`str`), кнопку з обраним значенням дати та модальне вікно, яке з'являється при натисканні на кнопку.

- Модальне вікно містить компонент `DatePicker`, який дозволяє користувачу вибрати дату.

- При виборі дати значення оновлюється на кнопці.

2. Компонент SearchSchedule:

- Цей компонент є функціональним компонентом, який відповідає за відображення форми пошуку розкладу занять.

- Стани: `teacher` (рядок) - обраний користувачем викладач. `classroom` (рядок) - обрана користувачем аудиторія. `group` (рядок) - обрана користувачем група.

- Функції: `handleSearch` - функція, яка викликається при натисканні на кнопку "Пошук". Вона виконує дії, пов'язані з пошуком розкладу занять.

- `handleClear` - функція, яка викликається при натисканні на кнопку "Очистити". Вона скидає значення полів форми на початкові.
 - Відображення: компонент містить два компоненти `getDatePicker` для вибору початкової та кінцевої дати. Компоненти `Picker` використовуються для вибору викладача, аудиторії та групи.
 - Кнопки "Пошук" та "Очистити" виконують відповідні функції.
 - Внизу сторінки є футер з авторською інформацією.
3. Компонент `LoginPage`:
- Цей компонент відповідає за відображення сторінки входу в додаток. Стани: `username` (рядок) - ім'я користувача, яке вводиться користувачем.
 - `password` (рядок) - пароль, який вводиться користувачем.
 - Функції: `handleLogin` - функція, яка викликається при натисканні на кнопку "Увійти". Вона перевіряє введені користувачем дані і переходить на сторінку "`SchedulePage`" при відповідності.
 - Відображення: компонент містить поле введення для імені користувача, поле введення для пароля та кнопку "Увійти".
 - При натисканні на кнопку виконується функція `handleLogin`.
4. Компонент `App`:
- Цей компонент відповідає за налаштування навігації між сторінками додатку.
 - Використовує пакети `@react-navigation/native` і `@react-navigation/stack` для створення стеку навігації.
 - Містить дві сторінки: "`LoginPage`" і "`SchedulePage`".
 - "`LoginPage`" відображається першою при запуску додатку.
 - "`SchedulePage`" відображається після успішного входу на сторінці "`LoginPage`".

4.3 Необхідна користувачу програми інструкція

Інструкція користувача допоможе вам ознайомитися з функціональністю програми та навчитися використовувати її елементи. Нижче наведено кроки, які вам слід виконати для успішного використання програми.

Крок 1: Вхід в додаток. Відкрийте додаток на своєму пристрої.

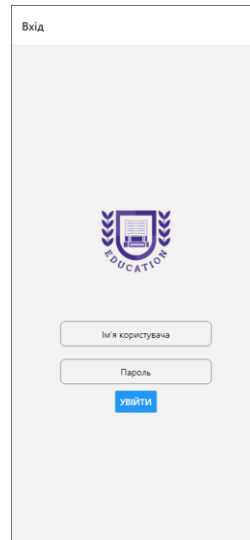


Рис. 4.1 – Головна сторінка

Введіть своє ім'я користувача та пароль у відповідні поля.

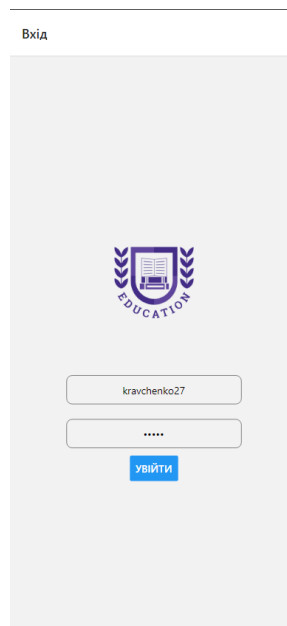


Рис. 4.2 – Введення пароля

Натисніть кнопку "Увійти".

Крок 2: Використання форми пошуку розкладу,

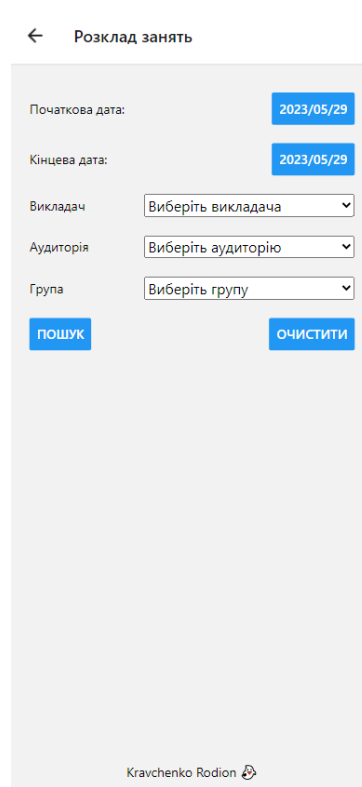


Рис. 4.3 - Форма пошуку розкладу

Після успішного входу ви будете перенаправлені на сторінку "Розклад занять".

На цій сторінці ви побачите форму пошуку розкладу занять.

Виберіть початкову та кінцеву дату за допомогою компонентів "Початкова дата" та "Кінцева дата".

Виберіть викладача, аудиторію та групу за допомогою відповідних підпунктів.

Після введення всіх параметрів натисніть кнопку "Пошук", щоб отримати розклад занять, що відповідає введеним критеріям.

Якщо ви хочете очистити форму та скинути всі введені значення, натисніть кнопку "Очистити".

Крок 3: Вибір дати,

Біля полів "Початкова дата" та "Кінцева дата" знаходяться кнопки з обраними датами.

Натисніть кнопку з датою, яку ви бажаєте змінити.

Відкриється модальне вікно з календарем.

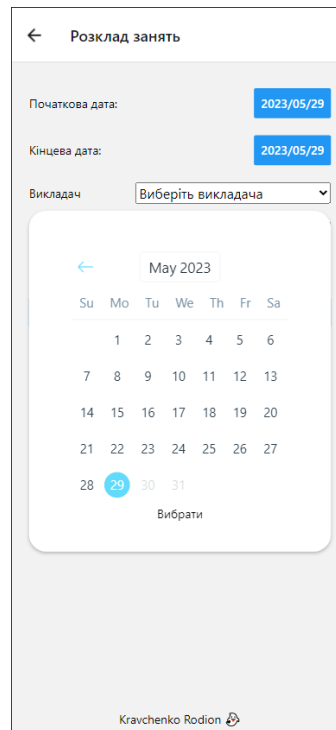


Рис. 4.4 - Модальне вікно з календарем

Виберіть потрібну дату за допомогою календаря.

Після вибору дати модальне вікно закриється, а обрана дата відобразиться на відповідній кнопці.

Крок 4: Використання випадаючих списків

У формі пошуку є випадаючі списки для вибору викладача, аудиторії та групи.

Натисніть на випадаючий список, щоб розгорнути його.

Оберіть потрібний варіант зі списку, натиснувши на нього.

← Розклад занять

Початкова дата: 2023/05/29

Кінцева дата: 2023/05/29

Викладач: Виберіть викладача

Аудиторія: Виберіть викладача

Група: Іван Іванович, Петро Петрович

ПОШУК ОЧИСТИТИ

Kravchenko Rodion

Рис. 4.5 – Вибір викладача

Вибраний варіант буде відображений у випадаючому списку.

Крок 5: Виконання пошуку та очищення форми

Після введення всіх необхідних параметрів пошуку натисніть кнопку "Пошук".

Програма виконає пошук за вказаними критеріями і відобразить розклад занять, що відповідає введеним параметрам.

Якщо ви бажаєте очистити форму пошуку та скинути всі введені значення, натисніть кнопку "Очистити".

Крок 6: Вихід з додатку

Щоб вийти з додатку, закрийте його або виконайте відповідні дії для виходу, які передбачені для вашого пристрою.

ВИСНОВКИ

У даній роботі був розроблений додаток для пошуку розкладу занять, який надає користувачам зручний спосіб отримання інформації про проведення занять. Результати дослідження мають наукову, практичну та соціальну цінність.

Наукова цінність полягає у вивченні та аналізі вимог та потреб користувачів щодо пошуку розкладу занять. Це дозволило розробити модель задачі, яка відповідає основним потребам користувачів. Також, в процесі розробки додатку були використані сучасні технології та інструменти, що сприяє вдосконаленню професійних навичок розробника.

Практична цінність полягає в тому, що розроблений додаток забезпечує зручний та ефективний спосіб пошуку розкладу занять для студентів, викладачів та інших зацікавлених осіб. Він дозволяє швидко та зручно знаходити розклад занять за певний період, з фільтрацією за викладачем, аудиторією та групою. Це зекономлює час та спрощує процес отримання актуальної інформації.

Соціальна цінність полягає в тому, що розроблений додаток сприяє полегшенню організації навчального процесу. Він допомагає студентам та викладачам ефективно планувати свій час, уникати зіткнень у розкладах та забезпечує швидкий доступ до необхідної інформації. Це покращує якість навчання та сприяє більш ефективному використанню ресурсів університету.

Отримані результати роботи демонструють успішну реалізацію поставленої задачі і задовольняючий функціонал додатку для пошуку розкладу занять. Рекомендації щодо використання розробленого додатку включають його впровадження в навчальні заклади та університети для полегшення організації навчального процесу та покращення доступу до розкладів занять. Також,

можливо подальше розширення функціоналу додатку додавання бекендної частини, нових можливостей та покращення інтерфейсу для ще більшої зручності користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розклад занять в ПУЕТ [Електронний ресурс] - Режим доступу: <https://schedule.puet.edu.ua>
2. React css - Компонентний підхід [Електронний ресурс] - Режим доступу: <https://monsterlessons.com/project/lessons/react-css-komponentyj-podhod>
3. React. Як підключити css [Електронний ресурс] - Режим доступу: <http://studyjavascript.blogspot.com/2019/07/reakt-css.html>
4. University wisdom : discover the secrets of getting the most from your experience at university and use them as stepping-stones to launch your life and career : a practical guide for all students / Vanessa McCallumn, Alexandra Brooke – 2016. – С. 232–235.
5. Найпопулярніші мобільні ОС за 2018 рік [Електронний ресурс] - Режим доступу: <https://cybercalm.org/novyny/75-smartfoniv-v-ukrayini-pratsyuye-naandroid-a-majzhe-polovyna-planshetiv-na-ios/>
6. Розробка мобільного додатку «РОЗКЛАД ЗАНЯТЬ» під платформу Android [Електронний ресурс] - Режим доступу: <https://jait.donnu.edu.ua/article/view/8932>
7. React Navigation. Routing and navigation for Expo and React Native apps. [Електронний ресурс] - Режим доступу: <https://reactnavigation.org>
8. Уроки React Native [Електронний ресурс] - Режим доступу: <https://itproger.com/course/react-native>
9. React Native [Електронний ресурс] - Режим доступу: <https://reactnative.dev>
10. Expo Native [Електронний ресурс] - Режим доступу: <https://expo.dev>
11. Mastering React Native: A Beginner's Guide. N.p., CRC Press.

12. Boduch, A., Derks, R. (2020). React and React Native: A Complete Hands-on Guide to Modern Web and Mobile Development with React.js. Індія: Packt Publishing. - Режим доступу: https://www.google.ro/books/edition/React_and_React_Native/XCLhDwAAQBAJ?hl=uk&gbpv=1&dq=react&printsec=frontcover
13. Mastering React: A Beginner's Guide. (n.d.). Сполучені Штати Америки: CRC Press. - Режим доступу: https://www.google.ro/books/edition/_j_-VEAAAQBAJ?hl=uk&gbpv=1

ДОДАТОК А

Вихідний код App.js:

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View, Button, SafeAreaView, Alert, TextInput } from
'react-native';
import React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import LoginPage from './LoginPage';
import SchedulePage from './SchedulePage';
const Stack = createStackNavigator();
function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen
          name="LoginPage"
          component={LoginPage}
          options={{ title: 'Вхід' }}
        />
        <Stack.Screen
          name="SchedulePage"
          component={SchedulePage}
          options={{ title: 'Розклад занять' }}
        />
      </Stack.Navigator>
    )
  }
}
```

```

    </NavigationContainer>
  );
}

export default App;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
    margin: 10
  },
  input: {
    backgroundColor: "white"
  },
});

```

Вихідний код LoginPage.js:

```

import React, { useState } from 'react';
import { TextInput, Button, View, StyleSheet, Image } from 'react-native';
import { useNavigation } from '@react-navigation/native';

function LoginPage() {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const navigation = useNavigation();

```

```
function handleLogin() {
  // Перевірка даних користувача та виконання входу в додаток
  if (username === 'admin' && password === 'admin') {
    // Навігація на іншу сторінку після успішного входу в додаток
    navigation.navigate('SchedulePage');
  }
}
```

```
return (
  <View style={styles.container}>
    <View style={styles.logoContainer}>
      <Image
        style={styles.logo}
        source={require('./logo.png')}
      />
    </View>
    <TextInput style={styles.input}
      placeholder="Ім'я користувача"
      onChangeText={text => setUsername(text)}
      value={username}
    />
    <TextInput style={styles.input}
      placeholder="Пароль"
      onChangeText={text => setPassword(text)}
      value={password}
      secureTextEntry={true}
    />
  </View>
)
```

```
<Button
  title="Увійти"
  onPress={handleLogin}
/>
</View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
  },
  input: {
    margin: 10,
    height: 40,
    width: 240,
    borderColor: 'gray',
    borderWidth: 1,
    borderRadius: 8,
    paddingLeft: 10,
    paddingRight: 10,
    marginBottom: 10,
    textAlign: 'center' },
  button: {
    margin: 10
```

```

    },
    logoContainer: {
      alignItems: 'center',
      marginBottom: 20,
    },
    logo: {
      width: 200,
      height: 200,
    },
  });
export default LoginPage;

```

Вихідний код SchedulePage.js:

```

import React, { useState } from 'react';
import { Footer, FooterTab, View, Text, Modal, Button, StyleSheet, Picker,
TouchableOpacity } from 'react-native';
import DateTimePicker from '@react-native-community/datetimepicker';
import { StatusBar } from "expo-status-bar";
import DatePicker from "react-native-modern-datepicker";
import { getToday, getFormattedDate } from "react-native-modern-datepicker";

function getDatePicker(str) {
  const today = new Date();
  const startDate = getFormattedDate(today.setDate(today.getDate()),
'YYYY/MM/DD')
  const minDate = getFormattedDate(today.setDate(today.getDate() - 1000),
'YYYY/MM/DD')

```



```
const maxDate = getFormattedDate(today.setDate(today.getDate() + 1000),  
'YYYY/MM/DD')
```

```
const [open, setOpen] = useState(false)
```

```
const [date, setDate] = useState(startDate)
```

```
function handleOnPress() {
```

```
  setOpen(!open)
```

```
}
```

```
function handleChange(propDate) {
```

```
  console.log(propDate)
```

```
  setDate(propDate)
```

```
}
```

```
return (
```

```
  <View style={styles.row}>
```

```
    <Text style={styles.label}>{str}</Text>
```

```
    <Button title={date} onPress={handleOnPress} />
```

```
    <Modal
```

```
      animationType='slide'
```

```
      transparent={true}
```

```
      visible={open}
```

```
>
```

```
  <View style={styles.centeredView}>
```

```
    <View style={styles.modalView}>
```

```
      <DatePicker
```

```
        mode='calendar'
```

```
        selected={date}
```

```

        minimumDate={minDate}
        maximumDate={maxDate}
        onChange={handleChange}
      />
      <TouchableOpacity onPress={handleOnPress}>
        <Text>Вибрати</Text>
      </TouchableOpacity>
    </View>
  </View>
</Modal>
<StatusBar style="auto" />
</View>

);
}

const SearchSchedule = () => {
  const [teacher, setTeacher] = useState("");
  const [classroom, setClassroom] = useState("");
  const [group, setGroup] = useState("");

  const handleSearch = () => {
    // дії, що потрібно виконати при пошуку розкладу
  };

  const handleClear = () => {
    setTeacher("");
  };
}

```

```

setClassroom("");
setGroup("");
};

return (
  <View style={styles.container}>
    {getDatePicker("Початкова дата:")}
    {getDatePicker("Кінцева дата:")}

    <View style={styles.row}>
      <Text style={styles.label}>Викладач</Text>
      <Picker style={styles.picker} selectedValue={teacher}
onValueChange={setTeacher}>
        <Picker.Item label="Виберіть викладача" value="" />
        <Picker.Item label="Іван Іванович" value="ivan_ivanovych" />
        <Picker.Item label="Петро Петрович" value="petro_petrovych" />
      </Picker>
    </View>
    <View style={styles.row}>
      <Text style={styles.label}>Аудиторія</Text>
      <Picker style={styles.picker} selectedValue={classroom}
onValueChange={setClassroom}>
        <Picker.Item label="Виберіть аудиторію" value="" />
        <Picker.Item label="101" value="101" />
        <Picker.Item label="102" value="102" />
      </Picker>
    </View>

```

```

<View style={styles.row}>
  <Text style={styles.label}>Група</Text>
  <Picker style={styles.picker} selectedValue={group}
onValueChange={setGroup}>
    <Picker.Item label="Виберіть групу" value="" />
    <Picker.Item label="Група 1" value="group_1" />
    <Picker.Item label="Група 2" value="group_2" />
  </Picker>
</View>
<View style={styles.row}>
  <Button style={styles.button} title="Пошук" onPress={handleSearch} />
  <Button style={styles.button} title="Очистити" onPress={handleClear} />
</View>
<View style={styles.footer}>
  <Text>Kravchenko Rodion ☹️</Text>
</View>
</View>
);
};

```

```

const styles = StyleSheet.create({
  footer: {
    position: 'absolute',
    bottom: 0,
    left: 0,
    right: 0,
    paddingVertical: 10,

```

```
paddingHorizontal: 20,  
textAlign: 'center'  
},  
container: {  
  flex: 1,  
  padding: 20,  
  justifyContent: 'flex-start',  
  alignItems: 'stretch',  
},  
row: {  
  marginVertical: 10,  
  flexDirection: 'row',  
  alignItems: 'center',  
  justifyContent: 'space-between',  
},  
label: {  
  marginRight: 10,  
  flex: 1,  
},  
picker: {  
  flex: 2,  
},  
button: {  
  margin: 10,  
},  
centeredView: {  
  flex: 1,
```

```
    justifyContent: "center",
    alignItems: "center",
    marginTop: 22,
  },
  modalView: {
    margin: 20,
    width: "90%",
    backgroundColor: "white",
    padding: 35,
    borderRadius: 20,
    alignItems: "center",
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 4,
    elevation: 5
  },
});
export default SearchSchedule;
```