МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота бакалавра

ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОБІЛЬНОГО ДОДАТКУ "МЕНЕДЖЕР ЗАПИСІВ" НА ОСНОВІ React Native

Здобувач освіти гр. ІН–82 Володимир КИРИЧЕНКО

Науковий керівник
кандидат фізико-математичних наук,
асистент кафедри комп'ютерних наук Сергій ШАПОВАЛОВ

Завідувач кафедри
доктор технічних наук, професор Анатолій ДОВБИШ

СУМИ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую_____
Зав. кафедрою Довбиш А.С.
"_____" _____2022 р.

ЗАВДАННЯ

до кваліфікаційної роботи

Здобувача вищої освіти четвертого курсу, групи ІН-82 спеціальності «122 – Комп'ютерні науки» денної форми навчання Кириченка Володимира Олексійовича

.**Тема: «ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОБІЛЬНОГО ДОДАТКУ "МЕНЕДЖЕР ЗАПИСІВ" НА ОСНОВІ React Native»**

Затверджена наказом по СумДУ

№_____ від _____2022 р.

Зміст пояснювальної записки: 1) Актуальність тематики роботи, огляд технологій та рішень 2) Основні положення та загальні теоретичні відомості, набір мов та інструментів для створення додатку 3) Вибір мови програмування, та технологій 4) Практична реалізація

Дата видачі завдання "_____"_____2022 р.

Керівник роботи _____ Сергій ШАПОВАЛОВ

Завдання прийняв до виконання _____ Володимир Кириченко

РЕФЕРАТ

Записка: 49 стор., 23рис., 1 таблиця, 3 додатки, 18 джерел.

Об’єкт дослідження – розробка мультиплатформенного мобільного додатку.

Мета роботи – розробка інформаційного та програмного забезпечення мобільного додатку.

Методи дослідження – методи збору та аналізу даних.

Результати – розроблено інформаційне та програмне забезпечення для мобільного мультиплатформенного додатку, с повною суміжністю с Android та IOS. **"менеджер записів".**
Створення фінкцій сапису та редагування записів, та фільмотеки.

Мобільний додаток, **"менеджер записів".**,

Технологія React native.

Зміст

ВСТУП

Метою дипломної роботи є створення програмного додатку для ОС Android та IOS тобто мобільного кроссплатформенного додатку. Додаток призначений для отримання та створення контенту, що дозволяє завантажити редагувати та видаляти необхідну інформацію користувачу також перевагою є зручний та нативний інтерфейс додатку.

Функціонал мобільного додатку **"менеджер записів".**

- Створення дописів(завдань) та їх редагування;
- Можливість використовувати на всіх платформах, таких я к Android IOS і навіть ПК;
- Синхронізація з каталогом фільмів, та перегляду детальної інформації;
- Широка база для подальшого розвитку та додавання нових можливостей;

Однією з ключових переваг мобільних додатків є простота та зручність використання в сучасному світі. Великим аспектом в сучасній розробці програмного забезпечення є бізнес, адже без фінансів нові технології не матимуть змогу розвиватися з тією швидкістю як зараз, адже це дає змогу створювати та удосконалювати. В результаті розвитку, я б навіть сказав еволюції з’явився така технологія як React Native, дозволяє створювати кросплатформених додатків. І саме за допомогою неї, в даній роботі буде та детально описано принцип її роботи та створено мобільний додаток.

1.Актуальність тематики роботи, огляд технологій та рішень.

1.1 Основні технології розробки мобільних додатків

Мобільні рограми – це програмне забезпечення, розроблене для роботи на смартфоні, комп'ютері, планшеті або інших електронних пристроях. Програми мають дизайн, який призначений для певної функції. Більшість програм стосуються бізнесу або послуг, але це не завжди. Наприклад, ігрові програми часто не стосуються бізнесу, що стосується користувача.

Через обмежені апаратні ресурси ранніх мобільних пристроїв мобільні додатки уникали багатофункціональності. Однак, навіть якщо пристрої, які використовуються сьогодні, набагато складніші, мобільні додатки залишаються вузько функціональними. Ось як власники мобільних додатків дозволяють споживачам вручну вибирати саме функції, які повинні мати їхні пристрої.

Щоб краще зрозуміти процес створення мобільного додатка, давайте детальніше розглянемо всі різні технологічні міркування, які власники бізнесу повинні враховувати перед створенням програми, отжде на бізнес розрахована найбільша частина всіх ПО.[1]

Нативні програми

Нативні програми – це програми, створені для певної мобільної платформи. Наприклад, оригінальні програми Samsung доступні лише на пристроях Samsung, а користувачі можуть отримати доступ лише до програм інших постачальників мобільних телефонів на пристроях власної марки. Нативні програми можуть запропонувати більшу продуктивність порівняно з більш узагальненими програмами, які надають ту саму послугу.

Найважливішим недоліком нативних програм є їх вартість. Щоб створити, підтримувати та підтримувати програму для Android та iOS, вам в

основному потрібні дві команди розробників. Як ви можете собі уявити, це може призвести до підвищення ціни на проект.

Веб-програми

Для використання веб-програми потрібний доступ до Інтернету, і всі дані програми зберігаються в режимі онлайн. Код цього типу програми - Javascript, HTML5 або CSS. Перевага веб-програми полягає в тому, що вона потребує менше місця у пам'яті на вашому пристрої. Прикладами такого типу мобільних додатків є Notebook, Netflix та Dropbox.

Гібридні програми

Ці програми створені з використанням таких веб-технологій, як JavaScript, CSS та HTML 5. Чому їх називають гібридними? Гібридні додатки в основному працюють як веб-програми, замасковані під рідну оболонку.

Гібридні програми — це програми, які об'єднують ефективність родних та веб-додатків і підтримують обидва типи технологій. Ці програми швидко і легко створюються розробником, але вони часто мають нижчу продуктивність і, отже, надають менш цінний досвід для кінцевого користувача.

Таблиця 1.1 – Порівняльна характеристика основних видів мобільних додатків

	Нативні	Гібридні	PWA
Можливість перевикористання коду	Код розробляється окремо для кожної платформи	Можливе перевикористння коду	Можливе перевикористння коду
Доступ до функцій пристроїв	Найбільш повний	Обмежений доступ	Дуже низький
Модель розповсюдження	Завантаження в магазині додатків	Завантаження в магазині додатків	Доступ по URL
Продуктивність	Висока	Низька	Низька
Підтримка пристроями	Висока	Висока	Середня
Популярність	Висока	Середня	Середня
Підтримка зовнішніх бібліотек	Висока	Середня	Висока

1.3 Статистика мобільних додатків

Щоб надати ширше уявлення про сучасну сцену мобільних додатків, ось найважливіші статистичні дані, які показують поточний ландшафт мобільних пристроїв і його майбутнє.

- Середній користувач мобільних додатків у Сполучених Штатах має понад 100 програм, встановлених на своєму пристрої.
- Звичайний користувач мобільного телефону перевіряє свій смартфон 63 рази на день**.**
- 87% користувачів перевіряють свій телефон принаймні за годину до сну. З них 69% перевіряють свій телефон принаймні за п’ять хвилин до сну.
- 79% користувачів відмовляться від цифрового продукту лише через один день використання.
- Сьогодні на мобільні додатки припадає понад 57% усіх цифрових медіа **.**
- До 2021 року майже 7 мільярдів людей у всьому світі використовують мобільні пристрої.
- До 2022 року кількість завантажень мобільного додатка або рік сягне 258 мільярдів. Це велике збільшення порівняно з 2017 роком, коли ця цифра досягла 168 мільярдів.
- До того ж року споживчі витрати магазину додатків зростуть на 92% і досягнуть 157 мільярдів доларів у всьому світі. [10] [11] [12] [13]

1.4 Розробка мобільних додатків

Розробка мобільних додатків – це процес, який багато в чому черпає з традиційної розробки програмного забезпечення. Однак він зосереджений на створенні програмного забезпечення, яке використовує переваги унікальних можливостей апаратного забезпечення мобільних пристроїв.

Найпростіший сценарій створення мобільного додатка — це взяти програму для настільних комп’ютерів та імпортувати її на мобільний

пристрій. Однак, оскільки програма стає більш надійною, ця техніка може стати проблематичною.

Наприклад, програми, які використовують функції на основі місцеперебування, такі як карти, завжди створюються з нуля з урахуванням мобільних пристроїв. Служби на основі розташування, які надаються в настільному додатку, мають менший сенс, оскільки користувачі настільних комп'ютерів не пересуваються.

Сучасні смартфони та планшети оснащені такими функціями, як Bluetooth [15], ближній зв'язок (NFC) [16], GPS[17], гіроскопічні датчики, камери та багато іншого**.** Розробники можуть використовувати ці функції для створення додатків із такими технологіями, як віртуальна або доповнена реальність, сканування штрих-кодів, послуги на основі місцеперебування та багато іншого. Найуспішніші та популярні мобільні додатки найкращим чином використовують функції смартфона.Проблема апаратного забезпечення в мобільних пристроях створює ще одну складність:

Хоча розробники, які створюють програми для iOS, можуть очікувати, що програми будуть запускатися лише на двох типах пристроїв (iPhone та iPad), розробники Android не можуть сказати те саме. Фактично для них кожен смартфон і планшет може працювати на різному обладнанні та різних версіях операційної системи.

1.5 Процес створення мобільного додатку

Для розробки мобільного додатку, можа вибрати один із чотирьох різних варіантів:

- Створіть внутрішню команду розробників,
- Найняти спеціалізоване агентство з розробки програмного забезпечення,
- Покладайтеся на досвід фрілансерів.
- І найскладніший, це самостійно

Створення внутрішньої команди

Якщо вирішуєте створити для свого проекту внутрішню команду розробників, ви матимете повний контроль над напрямком розробки вашого мобільного додатка. Однак це обійдеться дорогою ціною.

Зрештою, доведеться платити не лише за зарплату розробникам, а й за ряд накладних витрат, таких як робочий простір, пільги та пільги, обладнання, ліцензії на програмне забезпечення та багато інших. Це може бути особливо важко, якщо ви перебуваєте в регіоні, де наймати мобільних розробників дорого. Якщо ви виберете цей варіант, очікуйте дуже високої ціни на ваш додаток. Підвищиться і ризик вашого проекту, тому що ви будете відповідати за роботу своєї команди.

Наймання фрілансера

Цей варіант, безумовно, найдешевший з усіх, окрім самостійної розробки, але ви там витрачаєте свій час. Найнявши одного фрілансера для створення вашого мобільного додатка, ви отримаєте задоволення від спрощеного процесу розробки. Спілкування може бути простіше, і ви не зіткнетеся з проблемами співпраці, оскільки лише одна особа відповідатиме за створення вашої програми.

Однак знайти надійного та надійного фрілансера – проблема. Досягнення та успіх всього свого проекту завдяки навичкам та досвіду однієї людини. Навіть якщо ви найняте талановитого мобільного розробника, він може бути дійсно талановитим у розробці серверної частини та мати обмежені знання про розробку інтерфейсу. Якщо ви створите програму, яка працює добре, але має непривабливий інтерфейс користувача, ваш продукт постраждає.

Найняти фрілансера — це економічно вигідний варіант, але він супроводжується багатьма різними ризиками, які можуть поставити під загрозу процес створення вашої програми на кожному окремому етапі.

Найняти компанію з розробки програмного забезпечення

Сьогодні існує багато агентств з розробки програмного забезпечення, які спеціалізуються на розробці мобільних пристроїв. Об’єднавшись із такими компаніями, ви делегуєте завдання створення свого додатка групі професіоналів, які можуть надати вам широкий спектр послуг, таких як:

- UX/UI дизайн [18],
- розробка продукту,
- розробка бекенда та інтерфейсу,
- тестування,
- забезпечення якості (QA),
- та управління проектами.

Основна перевага цього варіанту полягає в тому, що ви можете скористатися перевагами колективного досвіду та знань, отриманих командою, реалізуючи проекти, подібні до ваших**.** Ви можете підтвердити їхній досвід і навички за допомогою портфоліо, а також попросити рекомендацій клієнтів. Цей варіант пропонує чудовий баланс ціни та якості, особливо якщо ви вирішите передати свій проект команді, яка знаходиться в регіоні, де темпи розробки нижчі, ніж локально.

Отже, це все, що стосується порівняння 3 вищевказаних моделей, окрім самостійної розробки, приклад якої буде наведений нижче, але не ватро забувати те що ви можете приймати учать не тільки як замовник, а як і розробник, так що ви можете бути як і розробником так і замовником, ПО, а можете створити його собі самостійно.

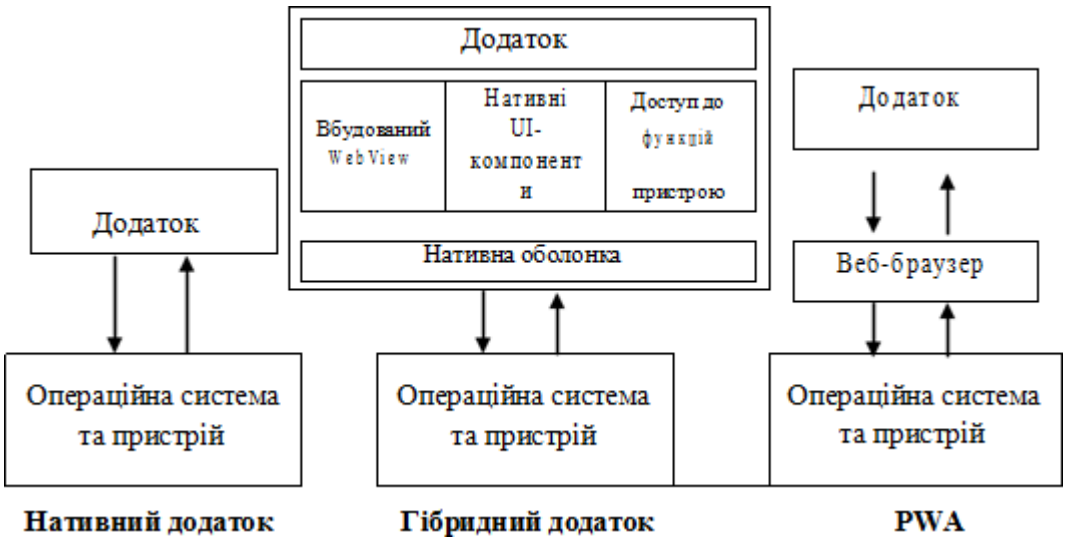

Рисунок 1.1 - Основні підходів реалізації мобільних додатків

2. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1Мови програмування для розробки додатків Android

Отже, у двох словах, програми для смартфонів є невід'ємною частиною нашого повсякденного життя. Їх можна використовувати для створення зв'язків, отримання інформації чи просто розваги! І хоча створення програми для смартфона також весело, воно вимагає трохи більше вказівок, зокрема щодо вибору правильної мови програмування

Хоча Kotlin є офіційною мовою для Android, існує багато інших мов, які можна використовувати для розробки додатків для Android. Нижче наведено докладну інформацію про них.

1. Java

Загалом, Java — чудова мова, щоб відчути всі переваги розробки додатків для Android. Однак це може бути трохи складним для новачків, які вважають за краще почати з чогось простішого, а потім повернутися до цього.

2. Kotlin

Тепер Kotlin є офіційною мовою для розробки додатків для Android, оголошеної Google у 2019 році. Kotlin — це кросплатформна мова програмування, яка може використовуватися як альтернатива Java для розробки додатків для Android. У 2017 році він також був представлений як вторинна «офіційна» мова Java. Kotlin може взаємодіяти з Java і працює на віртуальній машині Java.

Єдина суттєва відмінність полягає в тому, що Kotlin видаляє зайві функції Java, такі як винятки нульового покажчика. Це також усуває необхідність закінчувати кожен рядок крапкою з комою.Коротко кажучи, Kotlin набагато простіше для початківців у порівнянні з Java, і його також можна використовувати як «точку входу» для розробки додатків для Android.

3. C++

C++ можна використовувати для розробки додатків Android за допомогою набору Android Native Development Kit (NDK). Однак програму не можна створити повністю за допомогою C++, і NDK використовується для реалізації частин програми у рідному коді C++. Це допомагає використовувати бібліотеки коду C++ для програми за потреби.

Хоча C++ в деяких випадках корисний для розробки додатків для Android, його набагато складніше налаштувати і він набагато менш гнучкий. Це також може призвести до більшої кількості помилок через підвищену складність. Отже, краще використовувати Java, ніж C++, оскільки він не забезпечує достатнього виграшу, щоб компенсувати необхідні зусилля.

4. C#

C# дуже схожий на Java, тому він ідеально підходить для розробки додатків Android. Як і Java, C# також реалізує збір сміття, тому є менше шансів витоку пам'яті. І C# також має більш чистий і простий синтаксис, ніж Java, що робить кодування з ним порівняно простіше.

Раніше найбільшим недоліком C# було те, що він міг працювати тільки в системах Windows, оскільки використовував .NET Framework. Однак цю проблему вирішив Xamarin. Android — це кросплатформна реалізація загальномовної інфраструктури. Тепер, Xamarin. Інструменти Android можна використовувати для написання нативних програм Android і обміну кодом між різними платформами.

5. Python

Python можна використовувати для розробки прикладом цього є Kivy, бібліотека Python з відкритим кодом, яка використовується для розробки мобільних додатків. Він підтримує Android, а також заохочує швидку розробку додатків (як на мене, це безпрограшна ситуація!). Проте, недоліком цього є те, що для Kivy не буде вбудованих переваг, оскільки він не підтримується в оригінальному форматі.

6. HTML, CSS, JavaScript

Програми для Android можна створювати за допомогою HTML, CSS і JavaScript за допомогою фреймворку Adobe PhoneGap, що працює на основі Apache Cordova Фреймворк PhoneGap в основному дозволяє використовувати навички веб-розробки для створення гібридних програм, які відображаються через «WebView», але упаковуються як програма.

Хоча фреймворку Adobe PhoneGap достатньо для виконання основних завдань у сфері розробки додатків для Android, він навряд чи вимагає багато програмування, за винятком JavaScript. І оскільки для створення гідного додатка потрібно багато роботи, краще використовувати інші мови з цього списку, якщо ви хочете, щоб вас називали справжнім розробником Android (Так…Це справа!). Але якщо вам подобається Javascript, ви можете вивчити React Native, який є фреймворком з відкритим вихідним кодом, який зараз дуже потрібний і на основі якого в даній роботі буде створено мобільний додаток, детальна інформація та код буде в наступних розділах. Ви можете розробляти красиві та потужні гібридні програми з нативною реакцією, що означає, що ваша програма буде працювати як на Android, так і на iOS. Розробка гібридних додатків стає настільки популярною, що вивчення react-native може допомогти вам стати вашим кар'єром у розробці програмного забезпечення.

7. Dart

Ігнорувати Dart як мову програмування в сучасному контексті було б все одно, що ігнорувати горилу в кімнаті (тому що слон — це java). Dart — це мова програмування з відкритим вихідним кодом, яка керує фреймворком Flutter, який сьогодні набуває великої популярності через його здатність надавати красиві та продуктивні програми для Інтернету, настільних комп'ютерів і мобільних пристроїв за менший час. Ключова перевага Dart полягає в тому, що він розроблений Google як мова, оптимізована для клієнтів для швидких програм на будь-якій платформі. Dart в основному зосереджується на тому, щоб полегшити розробку інтерфейсу користувача для розробників за допомогою таких функцій, як гаряче перезавантаження, що дозволяє розробникам миттєво бачити зміни під час роботи над програмою. Dart також відомий своєю високою

продуктивністю, він компілює машинний код ARM і x64 для мобільних пристроїв, настільних комп'ютерів і серверів. І до JavaScript для веб-програм.
У нас є ще одна мова програмування для розробки додатків Android, тобто
8 Corona

Corona — це набір для розробки програмного забезпечення, який можна використовувати для розробки додатків Android за допомогою Lua. Він має два режими роботи, а саме Corona Simulator і Corona Native. Corona Simulator використовується для безпосереднього створення програм, тоді як Corona Native використовується для інтеграції коду Lua з проектом Android Studio для створення програми з використанням рідних функцій.

Хоча Lua трохи обмежена в порівнянні з Java, вона також набагато простіша і має легше навчання. Крім того, існують функції монетизації збірки, а також різні активи та плагіни, які збагачують досвід розробки додатків. Corona в основному використовується для створення графічних програм та ігор, але цим не обмежується.

2.2 Платформи для розробки мобільних додатків

Коли справа доходить до створення додатків для мобільних пристроїв або веб-сайтів, напевно, кожен подумає про складні процеси програмування або дизайн інтерфейсу користувача (UI). Це правда, але сьогодні ви можете легко обійти ці процеси, оскільки у вас вже є ці функції на вибір, які надаються постачальниками платформ для розробки додатків. Ці найкращі платформи розробки мобільних додатків схожі на універсальне середовище інструментів, які дозволяють створювати, тестувати, оптимізувати, налагоджувати, розгортати та підтримувати мобільні додатки. Залежно від розробника ці платформи будуть мати різні функції, наприклад, ви можете використовувати платформу для створення мобільного додатка від А до Я, але іноді вам доведеться

об’єднати кілька окремих платформ для інтерфейсу, сервера. розробка або додаткові рівні безпеки, сповіщення та зберігання даних тощо.

Microsoft Xamarin

Багато користувачів віддають перевагу Xamarin, оскільки він не тільки простий у використанні, ідеально підходить для пристроїв Android, iOS та Windows, але він також надає послуги розробки між платформних мобільних додатків, тобто, використовуючи лише базу коду C#, ви можете створювати рідну програмау для кількох платформ. Усі ці функції принесуть вам великий прибуток з точки зору витрат на розробку та часу виходу на ринок.

Flutter

Стильний і оригінальний – це лише деякі з компліментів, які ми можемо використовувати, описуючи цей пакет SDK з відкритим кодом Google. Він високо відомий своєю універсальністю в побудові інтерфейсу користувача та характеристиками, такими як створення унікальних, інноваційних безшовних анімацій, серед іншого. Ця платформа використовує мову Dart для створення найпотужніших програм. Тому Flutter чудово підходить для створення 2D-ігор або магазину одягу для систем iOS або Android.

Adobe PhoneGap

Якщо ви шукаєте різні платформи розробки мобільних додатків, які відповідатимуть вашим майбутнім стратегіям розширення, то не варто ігнорувати безкоштовну платформу PhoneGap з відкритим кодом. За допомогою цієї опції ви можете створити просту мобільну програму, використовуючи CSS3, HTML5, Javascript, і покращити її розробку за допомогою різних інтегрованих бібліотек. Використання програми також можна розширити за допомогою архітектури плагінів, що стає більш корисним для користувачів мобільних телефонів.

Sencha

Не дивно, що Sencha входить до нашого списку найкращих платформ для розробки мобільних додатків. Він уже надзвичайно популярний завдяки потужному пакету даних і багатим ресурсам інтерфейсу користувача (понад 115 компонентів), що підходить для тих, хто шукає кросплатформну програму з інтенсивним використанням даних. Програма написана на базі коду HTML5 і може бути встановлена на різних операційних системах, включаючи Android, iOS, Blackberry, Windows, Kindle і Tizen.

Appcelerator

Appcelerator зробив досить хорошу роботу, адаптувавши потужну та високопродуктивну модель мобільного додатка для корпоративних користувачів. Це фреймворк для розробки Titanium SDK, який допомагає скоротити час створення мобільного додатка для операційних систем, таких як iOS та Android. Хмарне сховище також є більш широким, не кажучи вже про інтеграцію зі сторонніми бібліотеками, спрямованими на підтримку бізнесу в різних регіонах глобального розгортання.

Ionic

Дві функції, які нам подобається в Ionic, — це простота використання та кросплатформенність. Це означає, що ви можете легко створити його в кількох операційних системах за допомогою HTML, CSS і Javascript. Крім того, якщо у вас є потреба створити більш просунуту мобільну програму для електронної комерції з більш корисними функціями та більшою інтерактивністю, тоді Ionic задовольнить вас інтегрованим AngularJS.

React Native

І знову React Native, адже тема цієї роботи тісно пов’язана з ним, і далі його ще більше. React Native може не сильно відрізнятися від перерахованих вище платформ розробки мобільних додатків, однак, якщо ви спробуєте створити мобільний додаток на цьому інструменті, ви не пошкодуєте. Він належить Facebook, є кросплатформною платформою розробки з відкритим кодом,

а також інтегрується зі сторонніми бібліотеками. Це також може скоротити час розробки додатків на різних платформах за допомогою функцій, які легко вивчати та розробляти.

2.3. Вибір платформи під потрібні цілі

IOS — це закрита платформа. Це означає, що Apple має повний контроль над своєю мовою програмування, програмами, які вона публікує, і пристроями, які вона поширює. Вони мають суворі обмеження, і будь-який відхід може призвести до загибелі стану вашої цифрової дитини. Тому потрібно дотримуватися цих правил і бути готовим до постійних змін. Це інноваційна компанія, лідер думок, який постійно шукає досконалості. З іншого боку, розробники завжди вміють спілкуватися з дизайнерами і створювати бездоганний продукт, який буде сприйнятий магазином на 100%.

Час від часу вам доведеться оновлювати свою програму до останньої операційної системи. Все, що їм потрібно зробити, це перейти в налаштування та почати оновлення. Як власник продукту, це означає, що вам потрібно буде оновлювати програму, щоб вона відповідала останнім потребам.

Платформа для розробки Android

Згідно з останніми даними Android є однією з найкращих платформ розробки мобільних пристроїв у світі. Тому що велика кількість людей користується смартфонами Android. Створивши програму для Android, ви отримаєте доступ до більшої кількості потенційних споживачів. Ця операційна система, як і iOS, надзвичайно адаптивна, оскільки одна програма може працювати на різних пристроях, включаючи смартфони, планшети, годинники, телевізійні приставки і навіть автомобілі. У Google Play також є широкий спектр програм. Тут кожен знайде щось на смак. Таким чином, опублікувавши програму на цьому ринку, ви можете бути впевнені у великій кількості перших послідовників.

Між платформні програми

Кросплатформний додаток — це комбінація HTML5 та рідних плагінів. Так званий підхід «запустить один запустіть скрізь» може здатися досить економічно ефективним рішенням. Appcelerator.com, одна з найпопулярніших кросплатформних фреймворків, обіцяє на 20% швидше розробку в порівнянні з рідними мовами, повторне використання до 90% розробленого коду при підтримці кількох платформ і значне зниження витрат.

Існує кілька кросплатформних фреймворків, кожна з яких має власний набір обмежень. Перш ніж виносити рішення на користь однієї системи над іншою, уважно оцініть її можливості. Крім того, не всі функції можна легко інтегрувати за допомогою кросплатформного фреймворку.

Найкращі платформи для розробки мобільних додатків складаються з багатоканальних можливостей надсилання, безпеки, плати та здібностей узгодження серверів. Проте, у будь-якому випадку, вибираючи між інструментами, розробники можуть домагатися вибору з найвищими шансами та кінцем, в той час як CIO або IT-керівники розуміють важливість підтримки інтересів організацій на фундаментальному рівні, і в цьому напрямку потрібно думати про весь образ витрат, рентабельності дизайнера, безпеки, керівників і це лише верхівка айсберга.Виведення програми на ринок вимагає серйозних досліджень і стратегічного планування.

3. ВИБІР МОВИ ПРОГРАМУВАННЯ, ТА ТЕХНОЛОГІЙ

3.1 Що таке React Native

React Native — це фреймворк JavaScript з відкритим вихідним кодом, розроблений для створення програм на кількох платформах, таких як iOS, Android, а також веб-додатків, використовуючи ту саму кодову базу . Він заснований на React і привносить всю свою славу в розробку мобільних додатків. Обидві фреймворки: ReactJS (веб) і React Native були реалізовані Facebook. React Native був проектом хакатону, спрямованим на вирішення найбільшої болючої точки компанії — підтримки двох кодових баз для свого додатка. Проблема з підтримкою двох кодових баз для такого великого додатка? Дублювання роботи і, часом, вирішення однієї і тієї ж проблеми двома різними способами. React Native — це пряма відповідь на ці проблеми. React Native використовує JavaScript для компіляції користувальницького інтерфейсу додатка, але використовує подання нативної ОС. Для більш складних функцій він дозволяє реалізувати код на рідних мовах ОС (Swift і Objective-C для iOS, і Java і Kotlin для Android).

Рисунок 3.1. Офіційний сайт React Native

3.2 Переваги React Native

Ми згадували, що React Native використовує JavaScript для створення інтерфейсу програми. На відміну від своїх конкурентів (наприклад, Ionic), React Native покладається не на веб-перегляди , а на реальні матеріали, надані нативними платформами. Він має вбудований доступ до власних представлень і компонентів, а також може використовувати написаний на власному коді та надавати доступ API до функцій ОС у програмі. Питання в тому, як саме це відбувається? React Native використовує концепцію «міста» , яка дозволяє здійснювати асинхронний зв'язок між елементами JavaScript і Native – концепція мосту лежить в основі гнучкості React Native. Нативні та JavaScript-елементи — це абсолютно різні технології, але вони здатні спілкуватися. Цей тип архітектури пропонує переваги використання багатьох властивих ОС функцій , але також має важливі проблеми; наприклад, постійне використання мостів всередині програми може значно сповільнити її продуктивність. Якщо ви створюєте програму, яка включає багато подій, багато даних тощо. React Native може бути не найкращим варіантом

3.3 Досвід роботи

Якщо ви коли-небудь розробляли для мобільних пристроїв раніше, ви можете бути здивовані, наскільки легко працювати з React Native.Команда React Native вклала у фреймворк потужні інструменти розробника та значущі повідомлення про помилки, тому робота з надійними інструментами є природною частиною вашого досвіду розробки.

Наприклад, оскільки React Native — це «просто» JavaScript, вам не потрібно перебудовувати свою програму, щоб побачити ваші зміни; замість цього ви можете натиснути Command+R, щоб оновити програму, так само як і будь-яку іншу веб-сторінку. Усі ці хвилини, витрачені на очікування створення вашої програми, можуть дійсно додатися, і навпаки, швидкий цикл ітерацій React Native виглядає як знахідка.

Крім того, React Native дозволяє скористатися перевагами інтелектуальних інструментів налагодження та звітів про помилки. Якщо вам зручно користуватися інструментами розробника Chrome або Safari, вам буде приємно дізнатися, що ви також можете використовувати їх для мобільної розробки. Аналогічно, ви можете використовувати будь-який текстовий редактор для редагування JavaScript: React Native не змушує вас працювати в Xcode для розробки для iOS або Android Studio для Android.

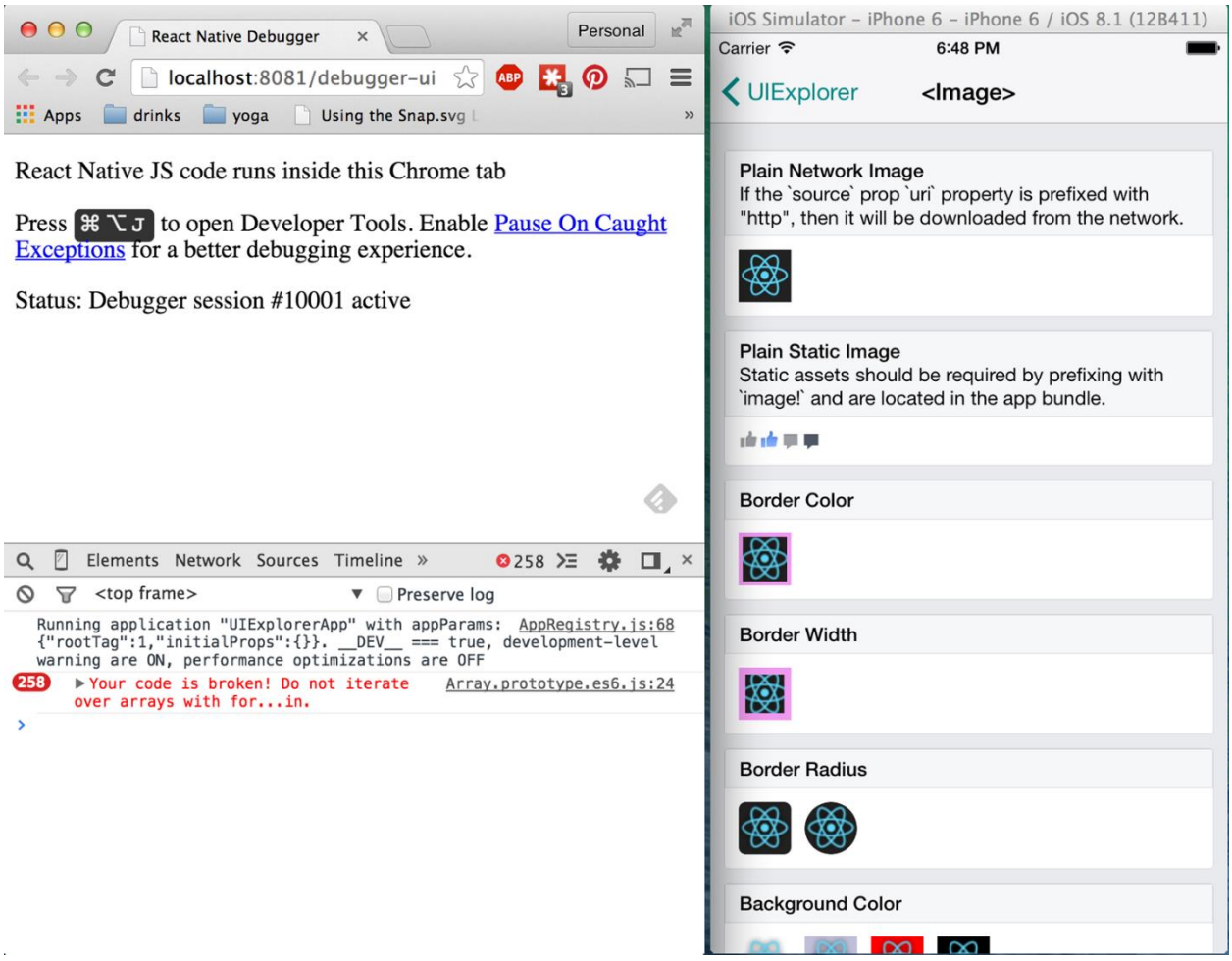

Рисунок 3.2. *Використання налагоджувача Chrome*

Окрім щоденного покращення вашого досвіду розробки, React Native також може позитивно вплинути на цикл випуску продукту. Наприклад, Apple дозволяє завантажувати зміни поведінки програми на основі JavaScript без необхідності додаткового циклу перевірки.

Усі ці невеликі переваги заощаджують вам і вашим колегам-розробникам час і енергію, дозволяючи зосередитися на більш цікавих частинах вашої роботи та бути більш продуктивними в цілому.

3.4 Повторне використання коду та обмін знаннями

Робота з React Native може різко скоротити ресурси, необхідні для створення мобільних додатків.Будь-який розробник, який знає, як писати код React, тепер може орієнтуватися на Інтернет, iOS та Android, і всі вони мають однакові навички. Усуваючи необхідність «розшарування» розробників на основі їх цільової платформи, React Native дозволяє виконувати ітерації швидше та ефективніше ділитися знаннями та ресурсами.

Крім обміну знаннями, багатою частиною вашого коду також можна поділитися. Не весь код, який ви пишете, буде кросплатформним, і залежно від того, яка функціональність вам потрібна на конкретній платформі, іноді вам може знадобитися зануритися в Objective-C або Java. Але повторно використовувати код на різних платформах напрочуд легко з React Native.

3.5 Ризики та недоліки

Як і в будь-якому випадку, за допомогою React Native не позбавлений недоліків, і чи підходить React Native для вас, залежить від вашої індивідуальної ситуації.

Оскільки React Native запроваджує ще один рівень у ваш проект, це також може зробити налагодження більш волосистим, особливо на перетині React і хост-платформи.

React Native ще молодий, і тут застосовуються звичайні застереження, пов’язані з роботою з новими технологіями. Але в цілому, я думаю, ви побачите, що переваги переважують ризики.

React Native — це захоплююча структура, яка дає змогу веб-розробникам створювати надійні мобільні додатки, використовуючи наявні знання JavaScript. Він пропонує швидшу розробку для мобільних пристроїв і ефективніший обмін кодом на iOS, Android і в Інтернеті без шкоди для кінцевого користувача або якості програми. Компроміс у тому, що це нове, і все ще триває робота. Якщо ваша команда може впоратися з невизначеністю, пов’язаною з роботою з новою технологією, і хоче розробляти мобільні додатки для більш ніж однієї платформи, вам варто звернути увагу на React Native.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

Для створення програм на базі React Native можна використовувати утиліту Expo cli – це окремий додаток для iOS та Android, що містить у собі всі React-компоненти, вбудовані у фреймворк за замовчуванням. Перед початком пороби потрібно налаштувати робоче місце, а точніше завагтажити потрібні програми такі як Node.js, VS Code, npm та Android Studio. Це мінімальний список програм для комфортної розробки, які порібно налаштувати ред розробкой програм.

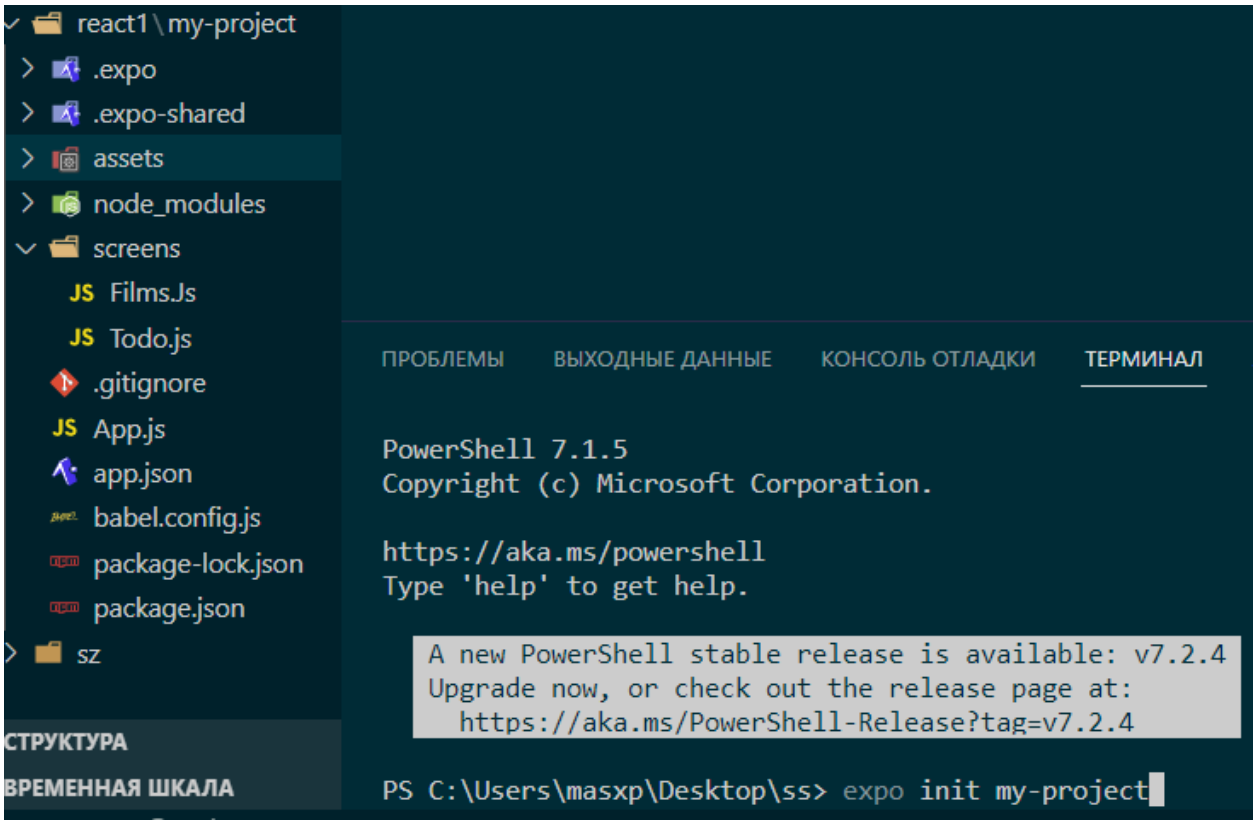

Рисунок 4.1 створення проекту

Початок створення мобільного додатку , за допомогою npm завантажуємо всі потрібні пакети та завантажуємо EXPO за допомогою команд npm i -g expo-cli expo init my-project ствторюємо стартовий проек, та чекаємо на повне завантаження

→

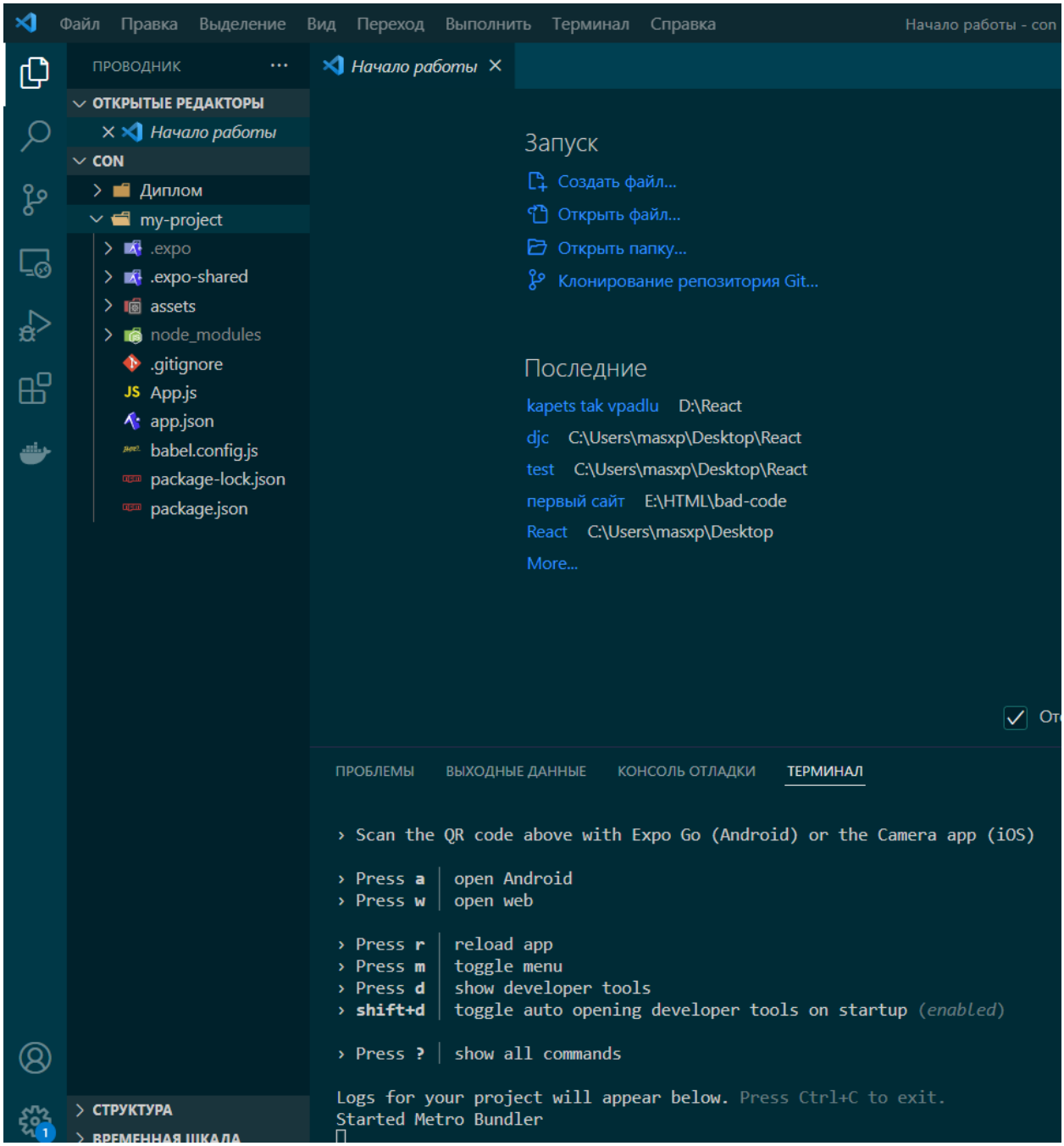

Рисунок 4.2. *Вибір емулятора в терміналі*

Після повного завантаження, обертаю стартову версію котра буде в проекті (без TypeScript) та чикаю поки буде збиратись проект.

Після цих маніпуляцій запускаємо наш проект, і обираємо потрібний нам емулятор в консолі (*Рисунок* 4.2). Якщо ван не подобається користуватись терміналом для запуску емулятора є альтернатива, бо після запуску відкривається localhost і в ньому також присутні всі види емулятору(*Рисунок 4.3)* .

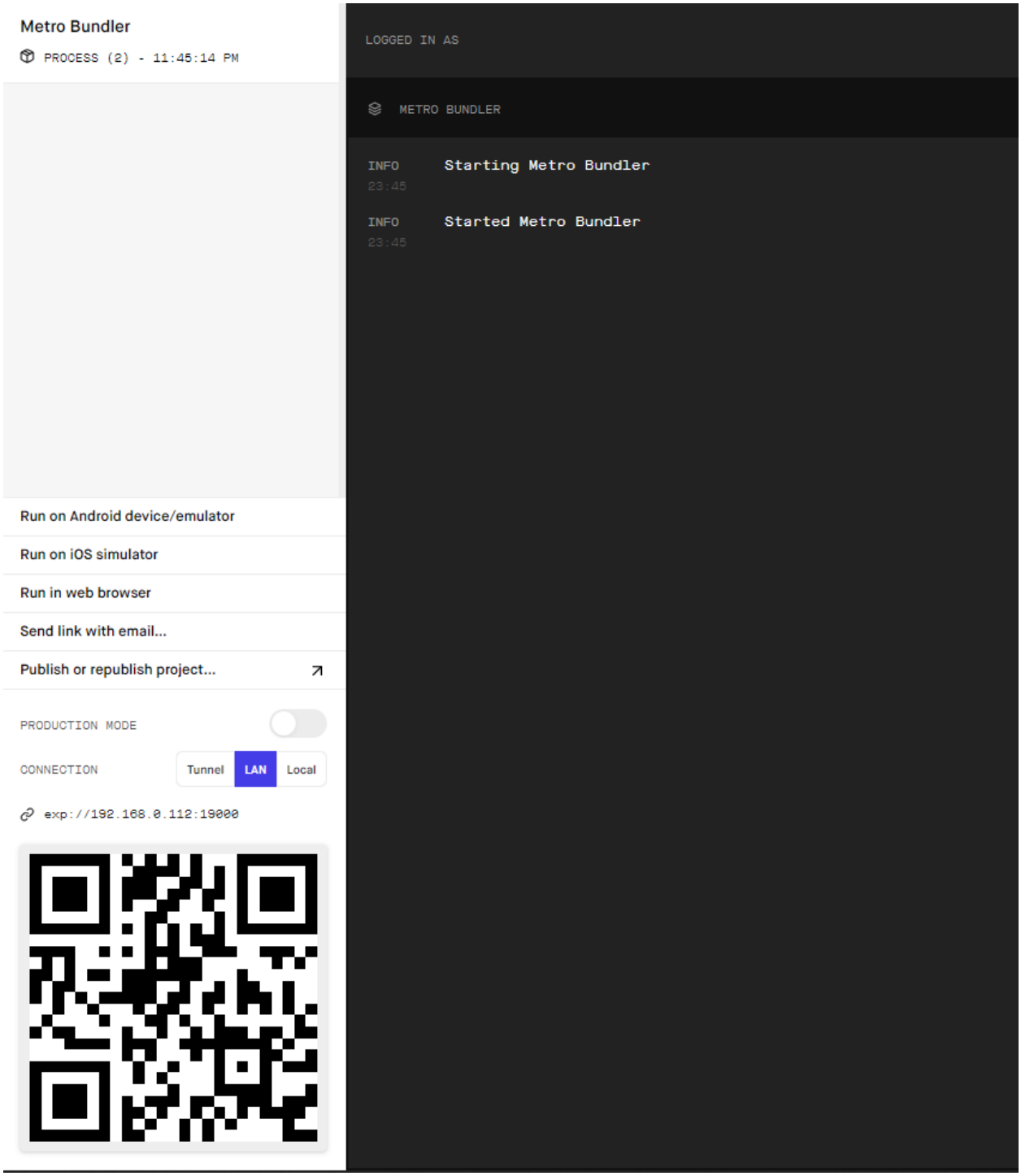

Рисунок 4.3. Вибір емулятора на localhost

Після запуску веб сервера у на доступний вибір емулятора(*Рисунок 4.3)*, обираємо потрібний емулятор, в моєму випадку це Android емулятор та Web browser.

1 Android емулятор - це досить зручний спосіб, якщо порівнювати з іншими, але він потребує високого навантаження на систему та завантаження та

налаштування Android Studio та образ вашого мобільного пристрою, і запуск на емуляторі самого Android емулятору.

2 В роботі буде представлений 2 спосіб як основний запуск проекту, в браузері. Запуск в браузері не потребує детальної налаштування, після вибору цього пункту в браузері відкривається нова вкладка з мобільним додатком, і не потрібує високого навантаження на систему.

3 Запуск на телефоні за допомогою мобільного додатку Expo

Це теж досить простий спосіб, для нього потрібно всього завантажити Expo на свій телефон, і відпанувати QR-код.

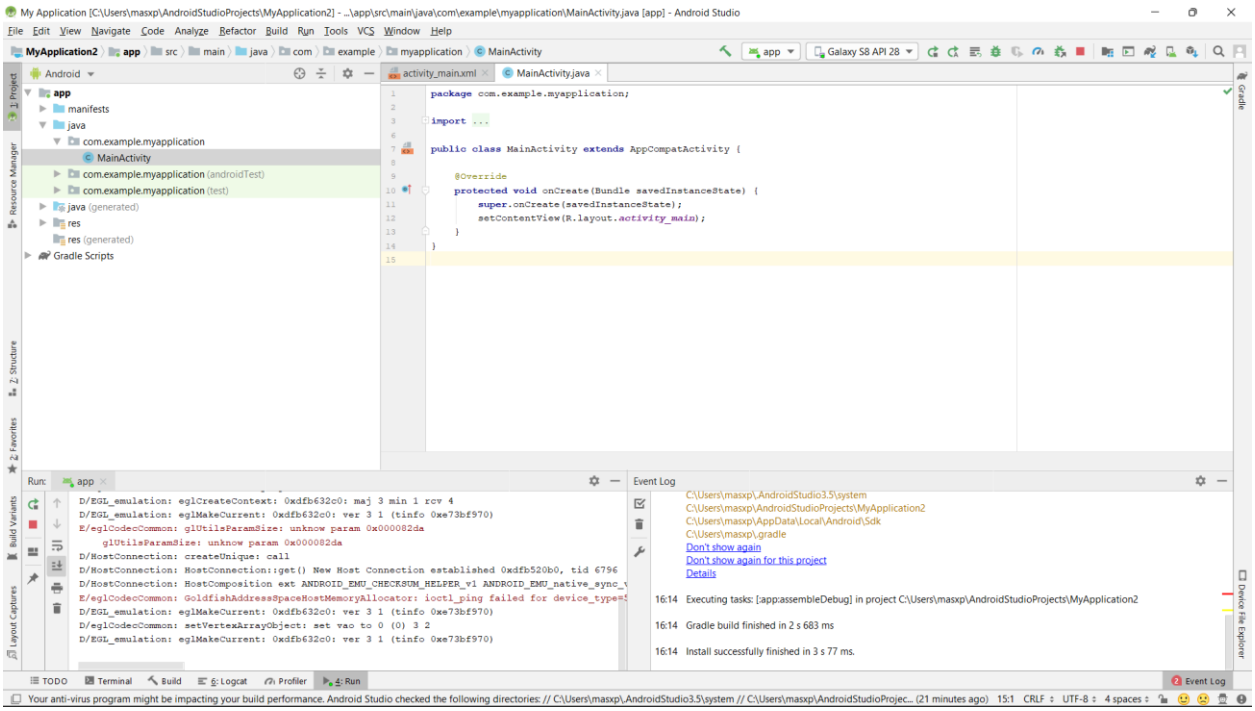

Рисунок 4.4. Створення файлу проекту та нового AVD (Android Studio)

Для попереднього першого запуску портібно відкрити Android Studio та запустити Android Virtual Device. (*Рисунок* 4.4)

Рисунок 4.5.Запуск образу телефона (Android 9.0)

Після запуску Android Virtual Device ми можем спостерігати наш емулятор Android 9.0 в образі потрібної моделі смартфону.

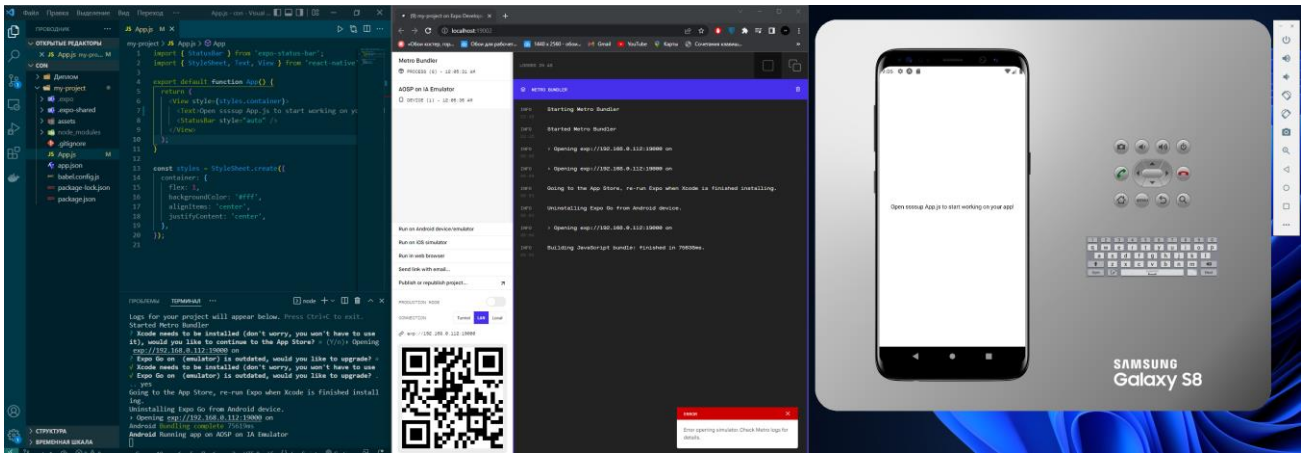

Рисунок 4.6.Підключення стартового пакету Expo

Після підключення до емулятора стартового пакету Expo ми можем спостерігати наш додаток, в якому всі зміни відбуваються в онлайн режимі. В цьому і є великий плюс цього методу, що ми бачимо як виглядає додаток на телефоні використовуючи тільки ПК.(*Рисунок* 4.6)

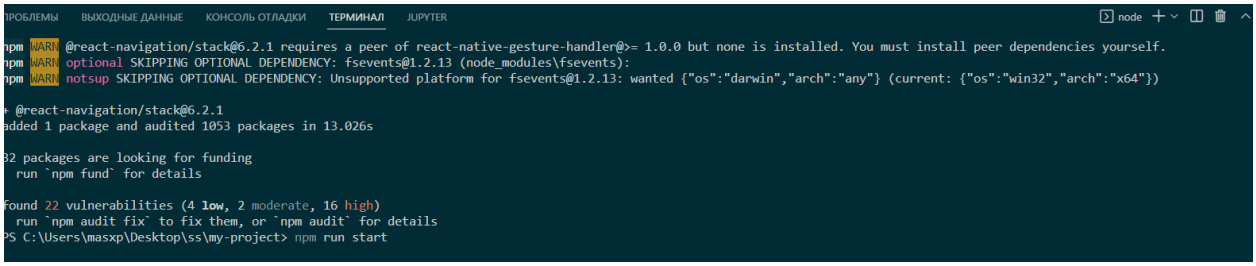

Рисунок 4.7. завантаження бібліотек

За допомогою команд додаємо всі потрібні бібліотеки (наприклад навігації) та чекаємо на загрузку всіх потрібних файлів.(*Рисунок* 4.7)

Рисунок 4.8. початок створення дизайну

В головному вікні створюємо початковий дизайн та навігацію, на головній сторінці, і також встановлюємо потрібні бібліотеки.

Після створення початкової навігації додаємо потрібні файли логотип, картинки та інші JS файли.

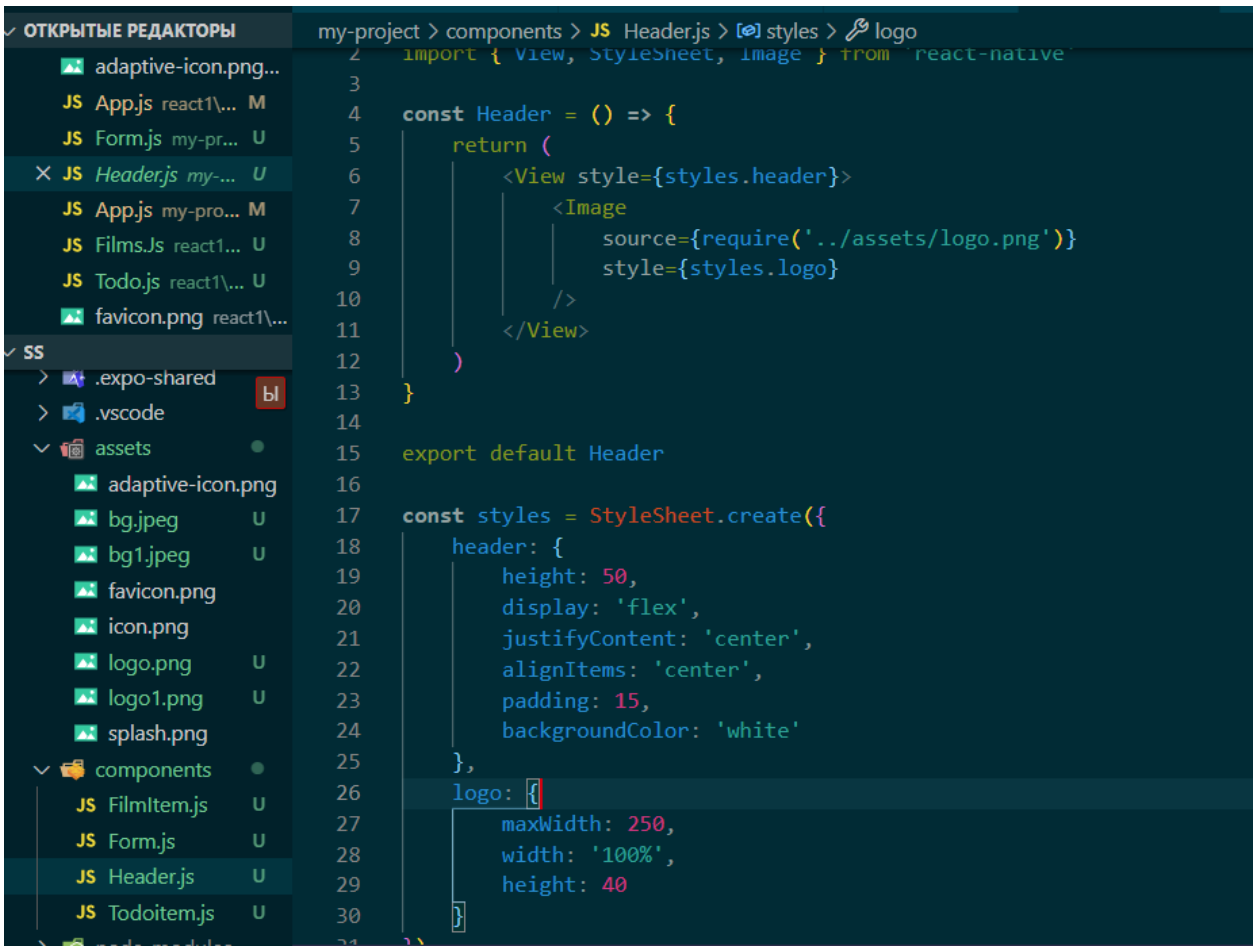

Рисунок 4.9. створення хеддеру

Створюємо потрібні нам js файли та наповняємо їх кодом та потрібною логікою.

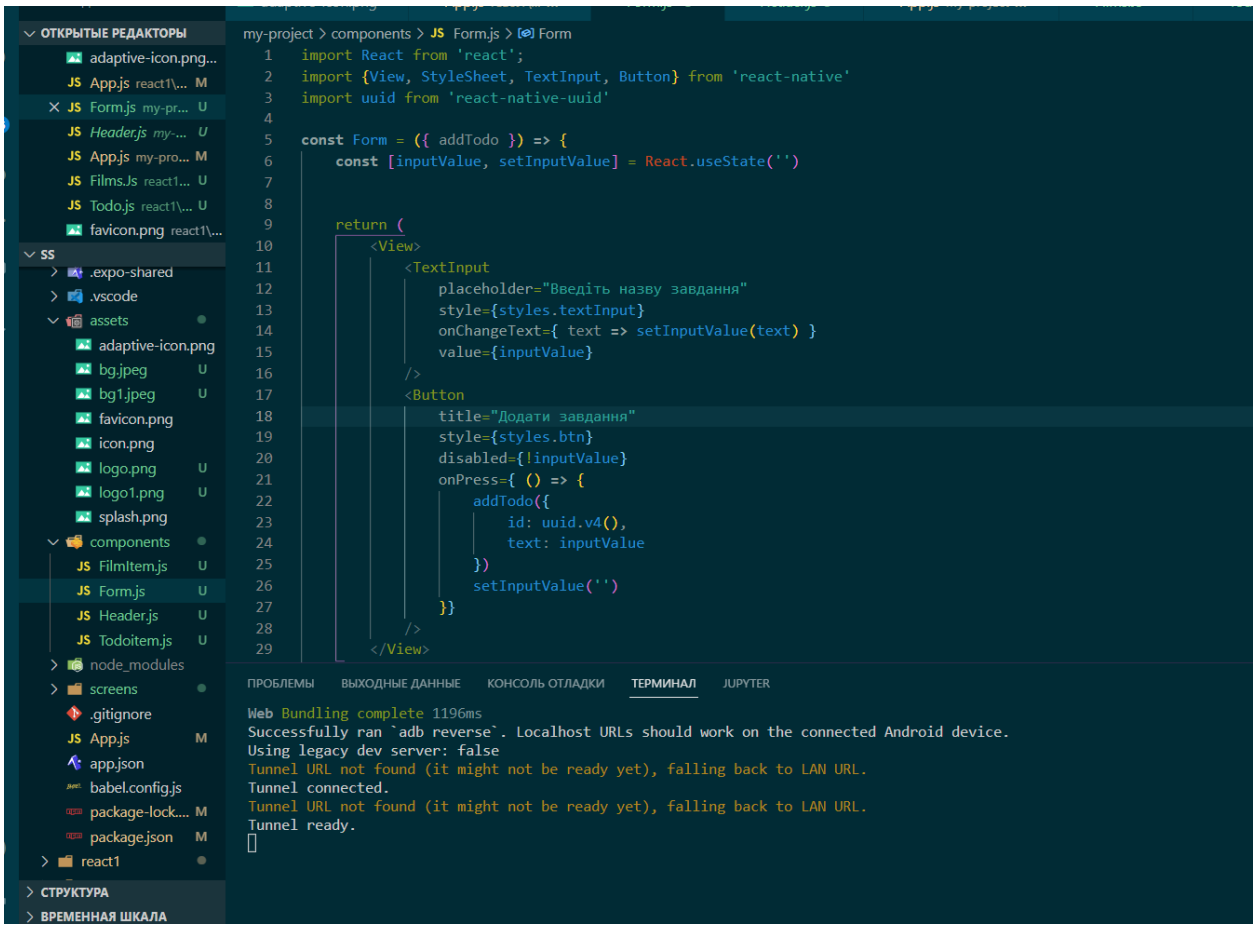

Рисунок 4.10.

Додаємо кнопки на головний екран на, та строїмо логіку на додавання та видалення і нформації в мобільний додаток.

Рисунок 4.11.

Додаємо API для передачі повної інформації фільмів і пишемо потрібний код.

Рисунок 4.12.

Додаємо модальні вікна для фільмів, для комфортного перегляду інформації.

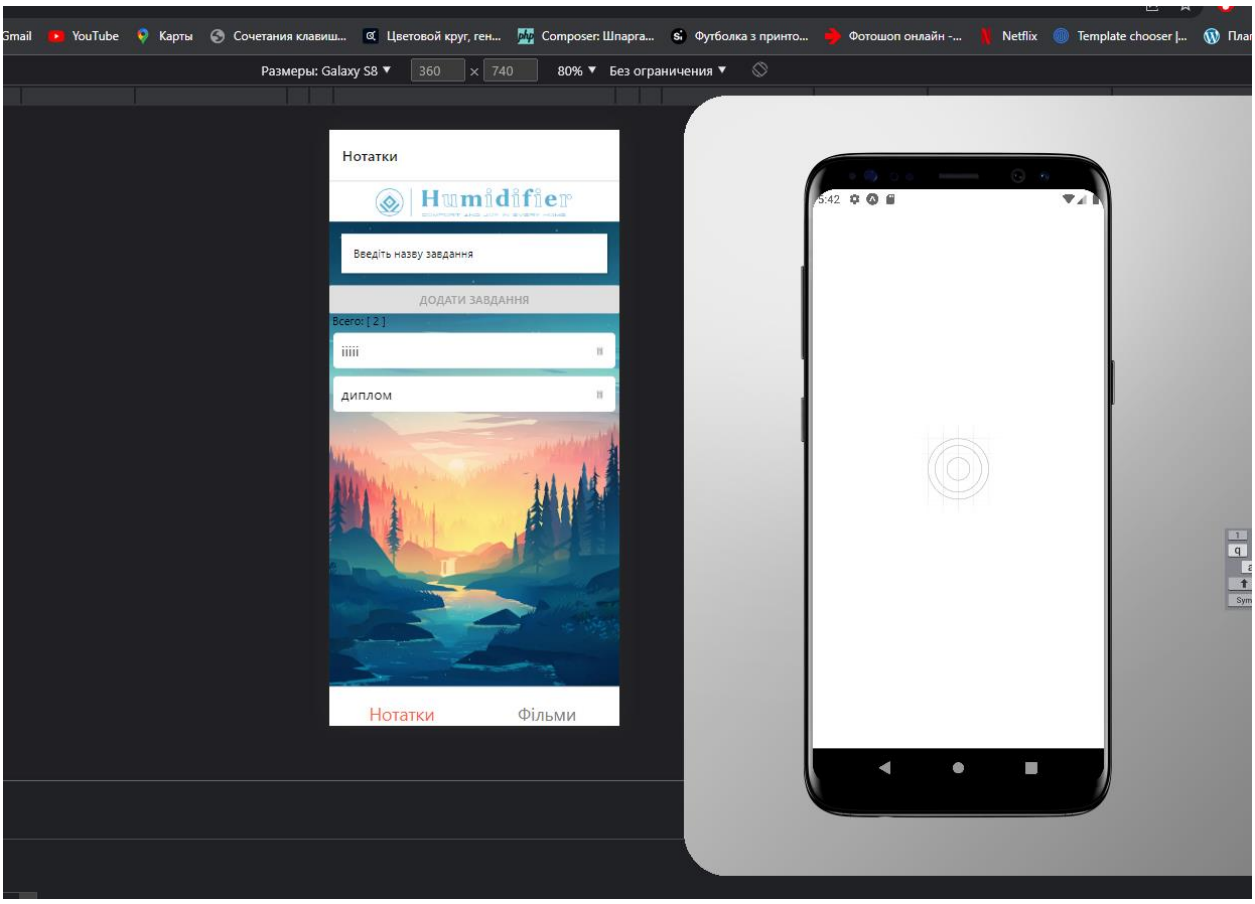

Рисунок 4.13. мобільний додаток в браузері

Запуск мобільного додатку в браузері, далі буде наведений приклад в емуляторі. Але подальша робота буде в браузері через надмірне навантаження на систему емулятором.

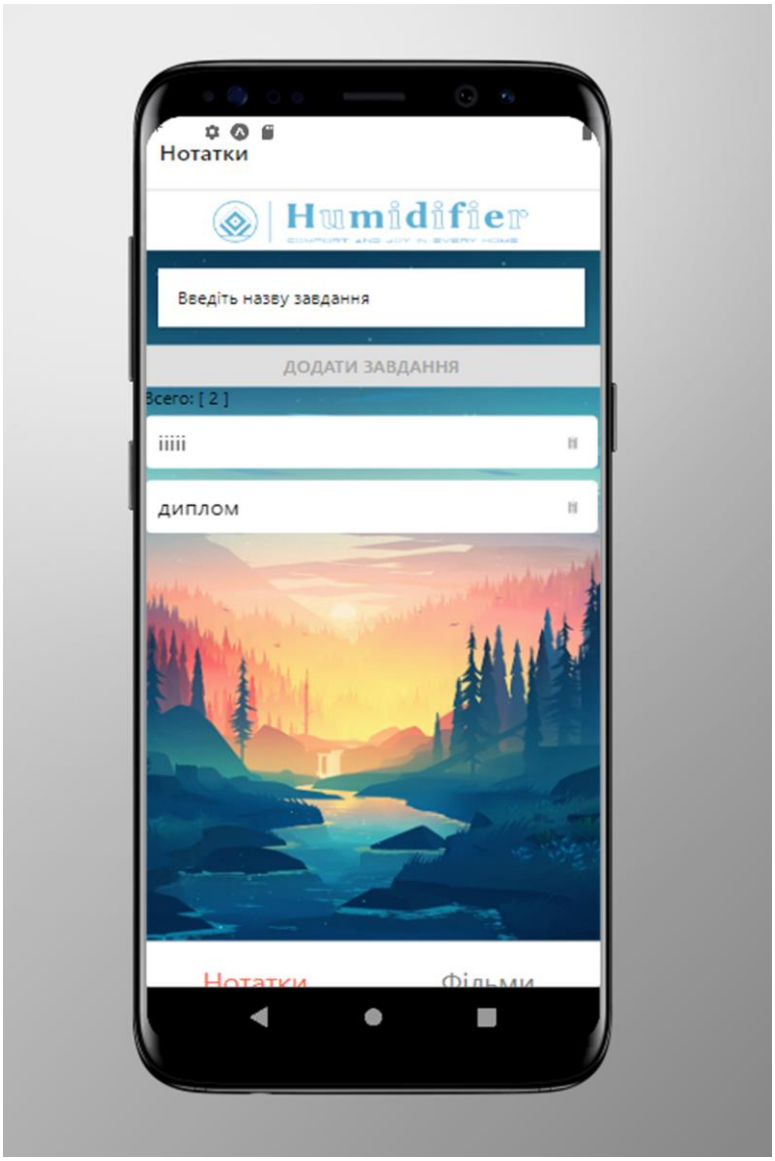

Рисунок 4.14.Приклад роботи в емуляторі

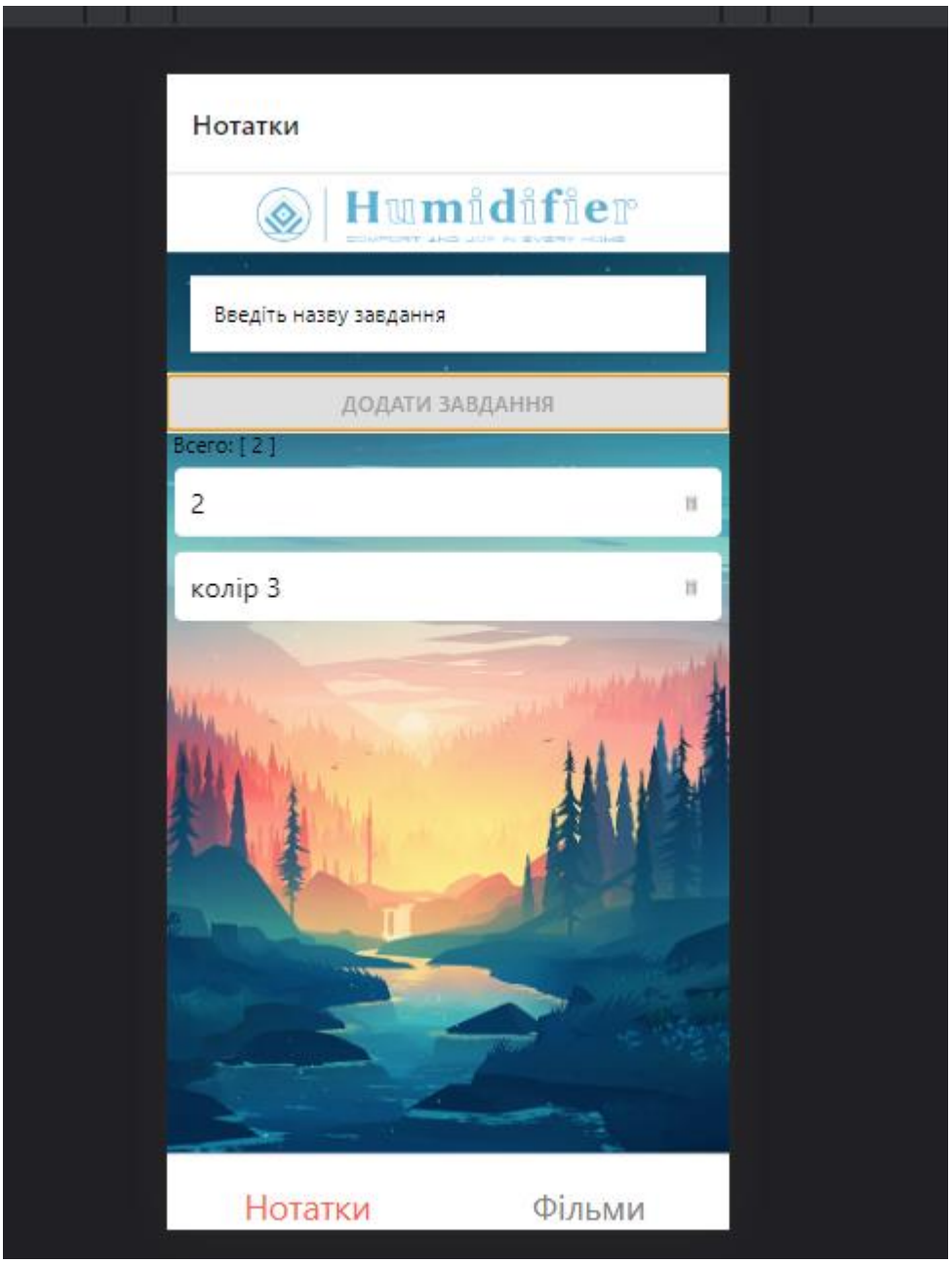

Рисунок 4.15.Тестування додатку додавання завдань

Початок тесту додатку за домогою додавння та видалення інформації.

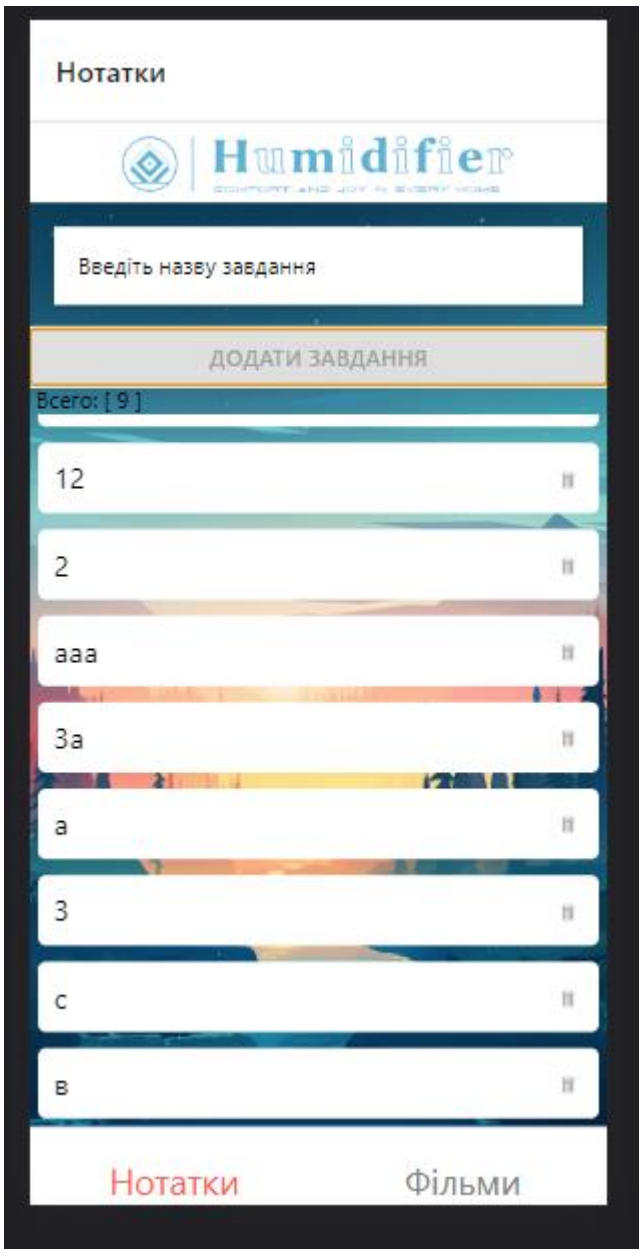

Рисунок 4.16.Тест функцій додатку

Продовження тесту.

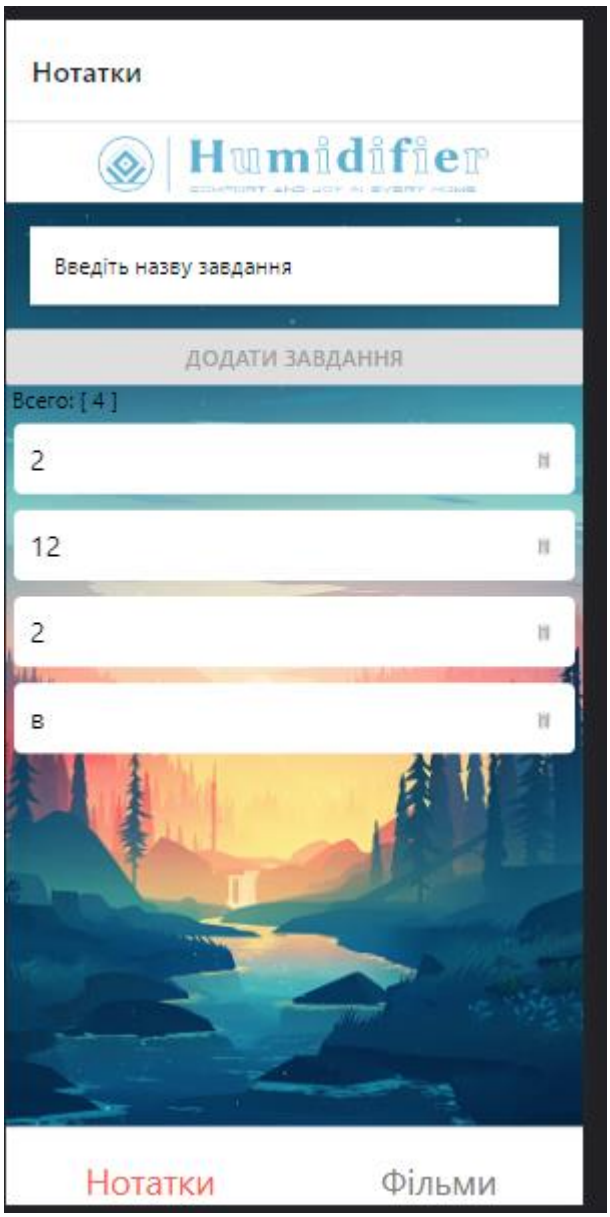

Рисунок 4.17.Видалення завдань

Приклад видалення інформації.

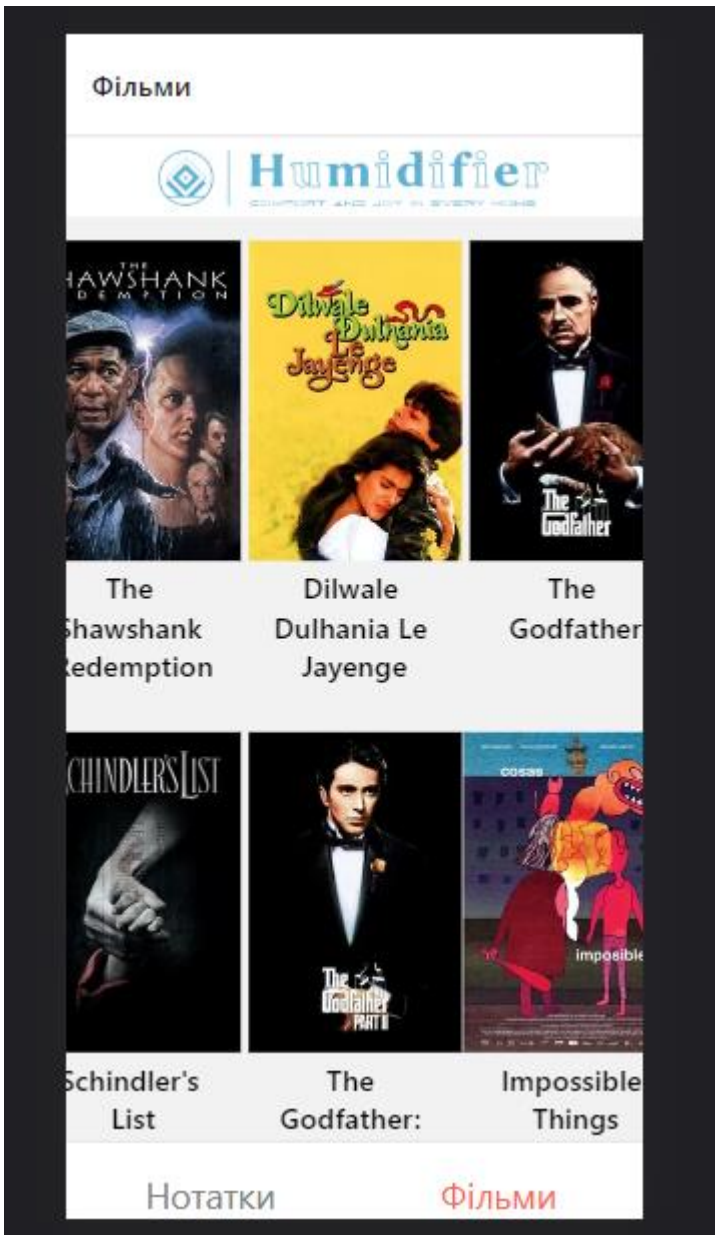

Рисунок 4.18 друга частина з фільмами

Демонстрація роботи другої частини додатка.

Рисунок 4.19.Модальне вікно з інформацією.

Демонстрація роботи модального вікна з інформацією.

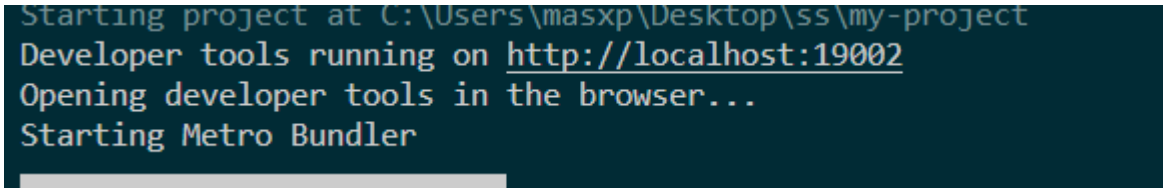

Рисунок 4.20. локальний хостинг

ВИСНОВОК

У дипломній роботі було розроблено та створено мультиплатформенний мобільний додаток “ **менеджер записів**” для смартфонів з ОС Android та IOS. Виділено недоліки та плюси сучасних мов програмування, та методики розробки мобільних додатків як . Android та IOS. Також були наведені основні типи мобільних додатків та технології для їх створення.

На першому етапі виконання роботи було проведено аналіз технологій розробки, типів та процес створення мобільних додатків.

У другому розділі даної роботи були описані основні теоретичні відомості. Розглянуто мови програмування для Android, та платформи розробки.

У третьому розділі даної роботи були описані основні технології

У четвертому розділі створено графічну частину та сам додаток “**менеджер записів**” та протестований с різних ОС та приладах.

СПИСОК ЛІТЕРАТУРИ

1. Типи мобільних додатків. [Електронний ресурс] — Режим доступу: https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types
2. Програми та додатки для Android [Електронне джерело]. – Режим доступу: URL http://android-phones.ru/. – Назва з екрану.
3. Android Emulator [Електронне джерело]. – Режим доступу: URL http://developer.android.com/tools/help/emulator.html. – Назва з екрану.
4. Google Android [Електронне джерело]. – Режим доступу: URL http://www.androidtalk.ru/google-android/. – Назва з екрану.
5. Все про Android [Електронне джерело]. – Режим доступу: URL http://androiddocs.ru/ . – Назва з екрану
6. Архітектура Android-додатків [Електронне джерело]. – Режим доступу: URL http://habrahabr.ru/post/141201/. – Назва з екрану.
7. App Store Resource Center. – Електрон. дан. – Режим доступу: https://developer.apple.com/appstore/index.html
8. Google Maps SDK for iOS. – Електрон. дан. – Режим доступу: https://developers.google.com/maps/documentation/ios/?hl=ru
9. Офіційний сайт React Native. – Електрон. дан. – Режим доступу: https://reactnative.dev/docs/components-and-apis
10. Офіційний сайт exp– Електрон. дан. – Режим доступу https://docs.expo.dev/
11. Microsoft Xamarin [Електронне джерело]. –https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin
12. Офіційний сайт NPM. – Електрон. дан. – Режим доступу: https://docs.npmjs.com/
13. Самоучитель Java [Електронний ресурс] – режим доступу: http://bit.ly/2ZaB6l8
14. Разработка под iOS и Android: рейтинг языков программирования 2020 [Електронний ресурс] — Режим доступу: https://apptractor.ru/reyting-yazyikovprogrammirovaniya-2020
15. Martin Fowler — GUI Architectures. Часть 2 [Електронний ресурс]. — 2009 — Режим доступу: https://habr.com/post/53536/.

16.7. Reuven M. Lerner Multiprocessing in Python [Електронний ресурс] / Reuven M. // Lerner, 2018. – Режим доступу:
17. Facebook, Communication between native and React Native [Електронний ресурс] / Facebook. ¾ Режим доступу: https://faceboo45 81 k.github.io/react-native/docs/communication-ios. ¾ Дата доступу: 22.11.2019.
18.. Turskyi, V. MOLE-RPC [Електронний ресурс] / Viktor Turskyi. ¾ Режим доступу: https://www.npmjs.com/package/mole-rpc. ¾ Дата доступу: 20.11.2019.

Додаток 1. App.js

```
import React from 'react';
import { StyleSheet, Text, View, SafeAreaView } from 'react-native';
import Todo from './screens/Todo'
import Films from './screens/Films'

import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator
        screenOptions={{
          tabBarAllowFontScaling: true,
          tabBarLabelStyle: {
            fontSize: 22,
          },
          tabBarIcon: () => {
            return
          },
          tabBarActiveTintColor: 'tomato',
          tabBarInactiveTintColor: 'gray',
        }}
      >
        <Tab.Screen name="Нотатки" component={Todo} />
        <Tab.Screen name="Фільми" component={Films} />
      </Tab.Navigator>
    </NavigationContainer>
  );
}

const styles = StyleSheet.create({
  text: {
    color: 'blue',
    fontSize: 35,
    textAlign: "center"
  }
})
```

Додаток 2. Todo.js

```
import React from 'react';
import {FlatList, SafeAreaView} from 'react-native';
import Header from '../components/Header';
import axios from 'axios'
import FilmItem from '../components/FilmItem';

export default function Films() {
  const [films, setFilms] = React.useState([])

  React.useEffect(() => {
    const getFilms = async () => {
      const res = await axios.get('https://api.the-
moviedb.org/3/movie/top_rated?api_key=578ae1e94516c4bfc0b4aa5bbb8ab157&l
anguage=en-US&page=1')
      setFilms(res.data.results)
    }
    getFilms()
  }, [])

  return (
    <SafeAreaView>
      <Header />
      <FlatList
        columnWrapperStyle={{flex: 1, justifyContent: 'space-around'}}
        numColumns={3}
        contentContainerStyle={{ paddingBottom: 100 }}
        data={films}
        keyExtractor={item => item.id}
        renderItem={( { item } ) => (
          <FilmItem
            film={item}
          />
        )}
      />
    </SafeAreaView>
  );
}
```

Додаток 3. Films

```
import React from 'react';
import { StyleSheet, SafeAreaView, ImageBackground, FlatList, Text} from 'react-native';
import Form from '../components/Form';
import Header from '../components/Header';
import TodoItem from '../components/Todoitem';

export default function Todo() {

  const [todoItems, setTodoItems] = React.useState([])

  const addTodo = (newItem) => {
    setTodoItems([
      ...todoItems,
      { ...newItem }
    ])
  }

  const deleteTodo = (id) => {
    setTodoItems((newList) => {
      return newList.filter( todoItems => id != todoItems.id )
    })
  }

  return (
    <SafeAreaView style={styles.app}>
      <Header />
      <ImageBackground
        source={require('../assets/bg.jpeg')}
        style={styles.bg}
      >
        <Form addTodo={addTodo} />

        {
          todoItems.length
          ? <Text> Всего: [ {todoItems.length} ] </Text>
          : null
        }

        <FlatList
          data={todoItems}
          keyExtractor={ item => item.id }
          renderItem={ ({ item }) => (
```

```
          <TodoItem
            item={item}
            deleteTodo={deleteTodo}
          />
        )}
      />

    </ImageBackground>
  </SafeAreaView>
 );
}

const styles = StyleSheet.create({
 bg: {
  flex: 1,
 },
 app: {
  flex: 1
 }
})
```