BookMyTicket

Documentation

SOULUTIONS

ABSTRACT

The "**BookMyTicket**" project is a mobile application designed to streamline the movie ticket booking process, providing users with a convenient and efficient way to secure tickets. The application offers features such as user registration, movie listings, seat selection, secure payment options, booking confirmations, and finding theaters. Leveraging modern technologies and an intuitive user interface, "**BookMyTicket**" addresses the inefficiencies and inconveniences of traditional ticket booking methods. This project aims to enhance the movie-going experience by simplifying the booking process and providing valuable information at users' fingertips.

DEVELOPERS

ITBNM-2110-0002	A.M.Tharoon Naveedya
ITBNM-2110-0026	K.A.B.Y.Kanangama
ITBNM-2110-0047	R.M.S.H.Rathnayake
ITBNM-2110-0053	H.R.Tharanga
ITBNM-2110-0082	S.G.V.D.Karunathilaka
ITBNM-2110-0100	R.M.A.S.V.Ranasinghe

Table of Contents

Introduction

What is project BookMyTicket?

Background

In the modern era, technology plays a pivotal role in simplifying various aspects of life. The entertainment industry, particularly the cinema sector, has experienced a transformation with the advent of online ticket booking systems. The traditional method of standing in long queues to purchase movie tickets is gradually being replaced by the convenience of booking tickets through mobile applications and websites. This project, titled "**BookMyTicket**," aims to develop a user-friendly mobile application that facilitates seamless movie ticket booking.

Problem Statement

The existing ticket booking processes in many regions still rely heavily on manual, offline systems. This approach is not only time-consuming but also prone to errors and inefficiencies. Customers often face difficulties such as unavailability of tickets, lack of real-time information about movie schedules, and inconvenient payment methods. These issues highlight the need for a reliable, efficient, and user-centric solution to streamline the ticket booking process.

Objectives

The primary objectives of the "**BookMyTicket**" project as followings.

- To develop a mobile application that allows users to book movie tickets easily.
- To provide real-time information about movie schedules, seat availability, and pricing.
- To implement secure and multiple payment options for ticket purchases.
- To enhance the overall user experience through intuitive design and functionality.
- To integrate additional features such as user account management, booking history, and location-based cinema searches.

Scope of the Project

- **User Registration and Login –**
  - Allowing users to create accounts, log in, and manage their profiles.
- **Movie Listings –**
  - Displaying a comprehensive list of movies currently showing in various cinemas, including details such as synopsis, ratings, and showtimes.
- **Seat Selection –**
  - Enabling users to select their preferred seats from an interactive seating chart.
- **Ticket Booking –**
  - Facilitating the booking process with secure payment gateways supporting various payment methods.

- **Booking Confirmation –**
    - Providing users with digital tickets and booking confirmations via email and in-app notifications.
- **User Reviews and Ratings –**
    - Allowing users to rate and review movies they have watched.
- **Location Services –**
    - Integrating with map services to help users locate nearby cinemas.

Significance of the Project

The "**BookMyTicket**" application holds significant value for both users and cinema operators. For users, it offers a convenient method to book movie tickets, saving time and effort. The application also enhances the movie-going experience by providing easy access to movie information. For cinema operators, the app can help increase ticket sales, reduce operational costs associated with manual ticketing, and improve customer satisfaction through better service delivery.

Documentation Overview

- **Chapter 1 – Introduction**
    - Provides an overview of the "**BookMyTicket**" project, including its background, problem statement, objectives, scope, and significance.

- **Chapter 2 – Requirements and Specifications**
    - Details the software, hardware, and system requirements necessary for developing and running the "**BookMyTicket**" application.

- **Chapter 3 – System Design**
    - Describes the architectural design patterns, design diagrams (ERD, use-case, sequence, and class diagrams), and data structures used in the project.

- **Chapter 4 – Implementation**
    - Covers the detailed process of implementing the project, including the development environment, tools and technologies used, key functionalities, and integration strategies.

- **Chapter 5 – Code Structure and Explanation**
    - Explains the organization of the project's codebase, including the structure of Java packages, classes, XML layout files, PHP scripts, and other essential resources.

- **Chapter 6 – User Interface**
    - Illustrates the user interface design of the application, showcasing screenshots of key screens like login, registration, home, booking, find, and account management.

- **Chapter 7 – Features and functionalities**

  - Lists and describes the main features and functionalities of the "**BookMyTicket**" app, such as user authentication, ticket booking, account management, and movie location finder.

- **Chapter 8 – Testing**
  - Outlines the functioning of the mobile application with the demonstration.

- **Chapter 9 – Conclusion**
  - Summarizes the achievements of the project, discusses the downsides and limitations, and suggests future enhancements to improve the app's performance and user experience.

Requirements and Specifications

Platform: Android

Minimum Android Version: Android 5.0

Database: SQLite for local storage on Android device, MySQL for server-side database.

Development Environment: Android Studio for Android app development, any PHP server environment for hosting the PHP script.

Permissions: The app may require permissions for accessing the internet (for connecting to the PHP script) and possibly location services (for the "**FindFragment**").

Hardware Requirements: Standard Android device hardware with internet connectivity.

System Design

Architecture

- The app follows a standard Android application architecture, likely utilizing the Model-View-Controller (MVC) design pattern.
- Activities and Fragments are used for the UI components, with each representing a distinct screen or interaction flow.
- **SQLiteOpenHelper** is used for managing the local SQLite database on the Android device.
- HTTP requests are made to a PHP script (**connectdb_add_movie.php**) for interacting with a server-side MySQL database.

Design pattern

- **MVC (Model-View-Controller)**
  - Activities/Fragments act as controllers, XML layout files represent views, and **DBHelper** represents models for database operations.
- **Factory Pattern**
  - Multiple instances where Intent objects are created to navigate between activities or fragments.

Algorithms

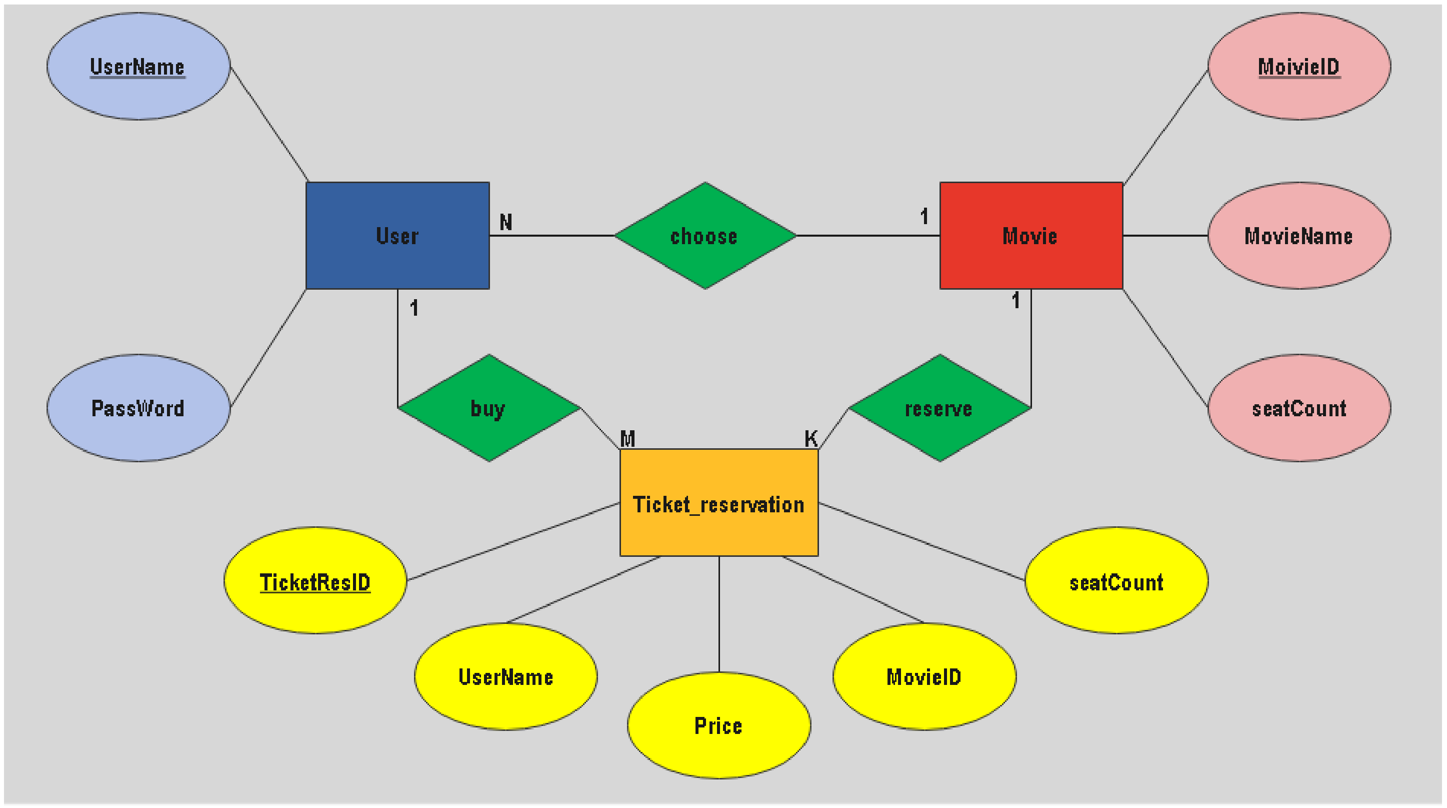
- **Authentication Algorithm –**
  - The app uses a simple username-password authentication mechanism. The **checkUnPw** method in the **DBHelper** class compares the entered username and password with those stored in the SQLite database.

- **Data Insertion Algorithm –**
  - The **insertLoginData** method in the **DBHelper** class inserts user credentials into the SQLite database.

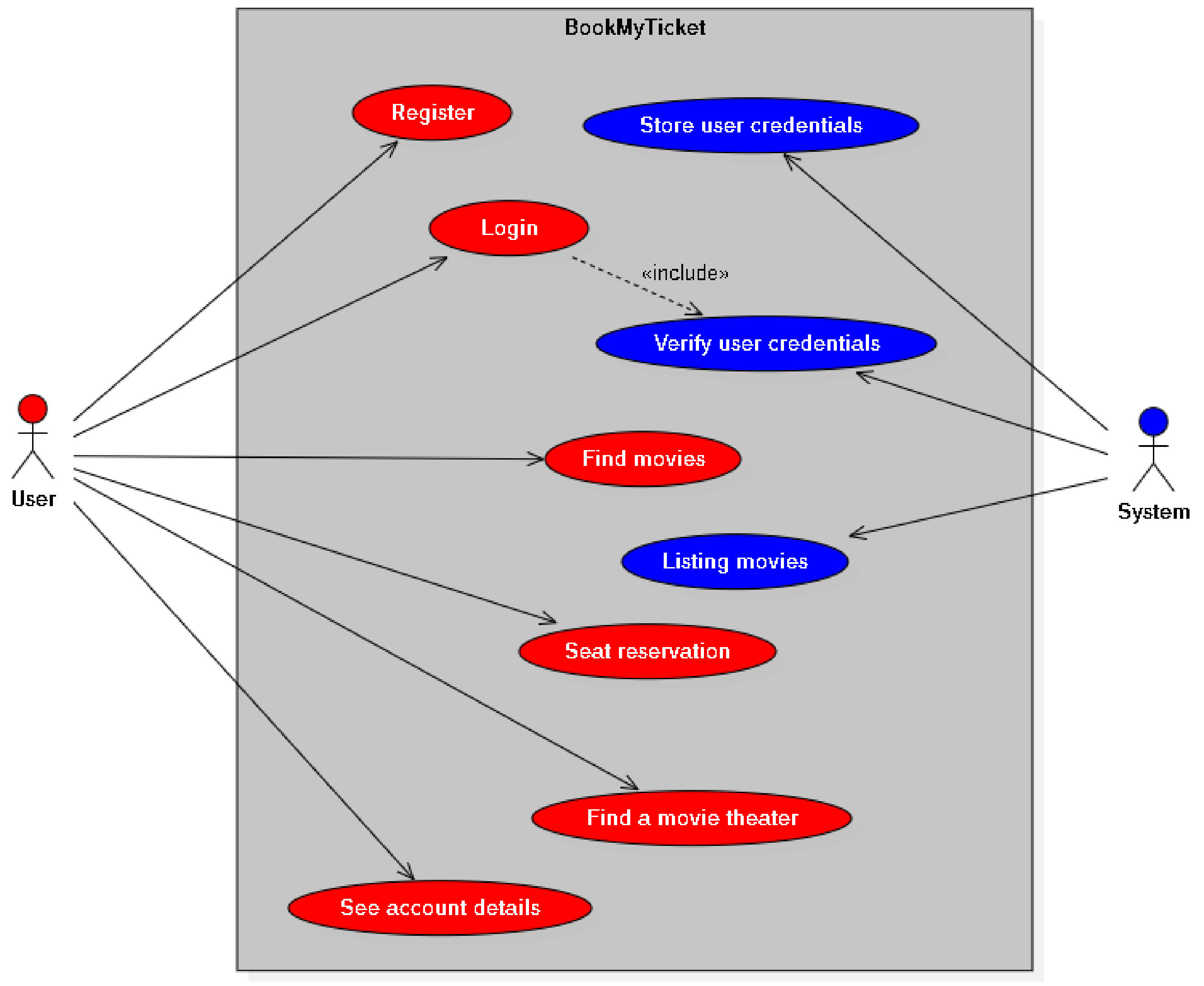
- **Mapping Algorithm –**
  - In the **FindFragment**, a simple algorithm is used to display markers on a Google Map. **LatLng** objects are created for various cinema locations, and **MarkerOptions** are used to display them on the map.

Design diagrams

ER Diagram



Use-case Diagram



Data structure

- **SQLite Database Tables –**
    - The app uses a simple database structure with tables to store user login credentials (**login table**) and possibly other data related to booking tickets (**movies table**).

Implementation

Software and Hardware Requirements for implementation

- **Software** – Android Studio, Java Development Kit (JDK), XAMPP, SQLite Browser
- **Hardware** – A computer with at least 8GB of RAM, a multi-core processor, and sufficient storage for development and testing (least 10GB space).

Tools and Technologies

- **Android Studio** – The primary Integrated Development Environment (IDE) for developing the Android application, using Java and XML.
- **XAMPP** – A free and open-source cross-platform web server solution stack package that includes Apache, MySQL, and PHP.
- **SQLite** – A lightweight, serverless, self-contained SQL database engine used for local storage within the application.
- **Google Maps API** – An API provided by Google to incorporate maps and location-based services into the application.
- **Java** – The primary programming language used for developing the application logic.
- **XML** – Used for designing the layout and user interface of the application.
- **PHP** – Used for server-side scripting to handle interactions with the remote database.

Code Structure and Explanation

This code structure of the project follows the standard conventions for organizing an Android project, with separate classes for activities, fragments, database operations, testing, and resource management
The code structure of the project can be outlined as follows.

Java Packages

- **com.example.bookmyticket** – This is the main package containing all Java classes.

Java Classes

Activities

- **MainActivity** – Main entry point of the application. Manages user login and navigation.
- **Activity_Book_Ticket** – Allows users to book tickets by selecting the number of seats.
- **CreateUserActivity** – Handles user registration by creating new accounts.
- **HomeActivity** – The main activity of the application after login. Manages navigation between different fragments.

Fragments:

- **AccountFragment** – Displays user account information.
- **BookFragment** – Allows users to browse and book movie tickets.
- **FindFragment** – Displays a map with cinema locations.
- **HomeFragment** – Displays the home screen with various options.

Database:

- **DBHelper** – Manages SQLite database operations such as user authentication and data insertion.

Testing:

- **ExampleInstrumentedTest** – Contains an instrumented test for checking the application context.
- **ExampleUnitTest** – Contains a unit test for checking addition functionality.

XML Layout Files:

- **activity_main.xml** – Layout for the main login activity.
- **activity_book_ticket.xml** – Layout for booking tickets.
- **activity_create_user.xml** – Layout for user registration.
- **home.xml** – Main layout for the home activity.
- **fragment_account.xml, fragment_book.xml, fragment_find.xml, fragment_home.xml** – Layouts for corresponding fragments.

- **ticket_book_activity.xml, activity_receipt.**xml – Additional layouts for ticket booking and receipt display.

PHP Script:

- **connectdb_add_movie.php** – Handles insertion of movie data into a MySQL database on the server.

Resource Files:

- **strings.xml, colors.xml, dimens.xml** – Resource files containing string constants, colors, and dimensions used in the layouts and activities.

Manifest File:

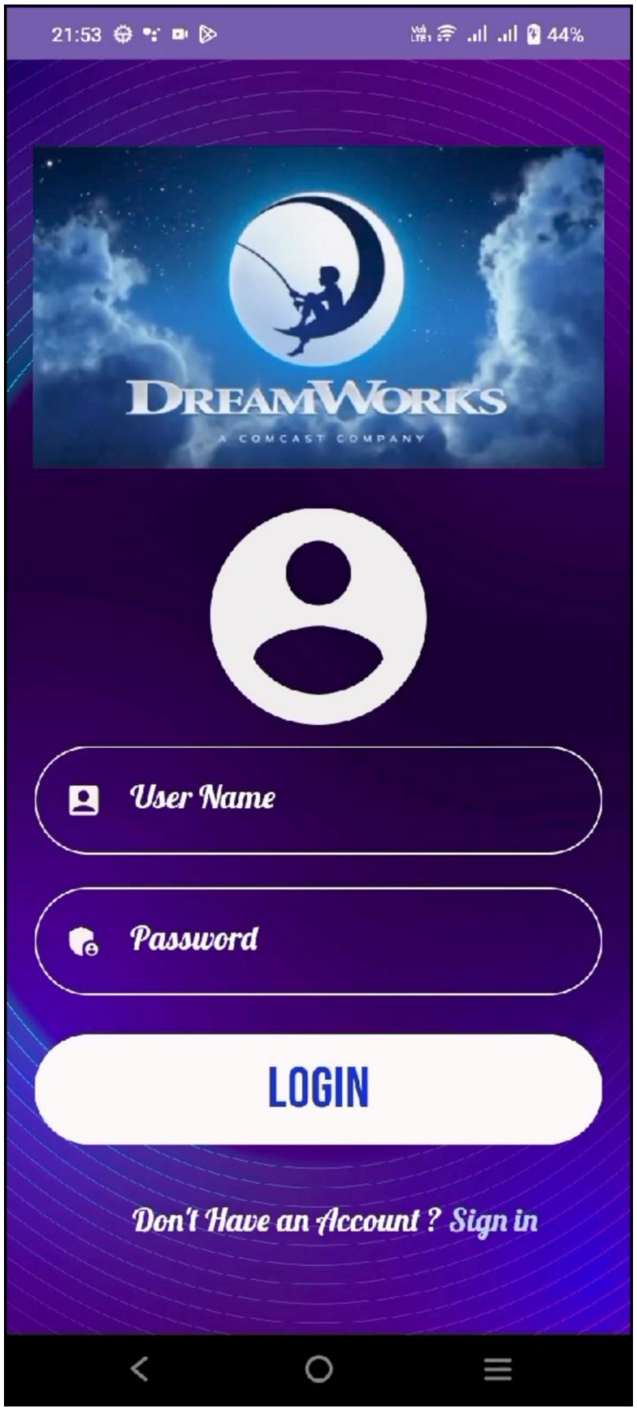
- **AndroidManifest.xml** – Contains the declaration of activities, permissions, and other essential information about the application.

Other Files:

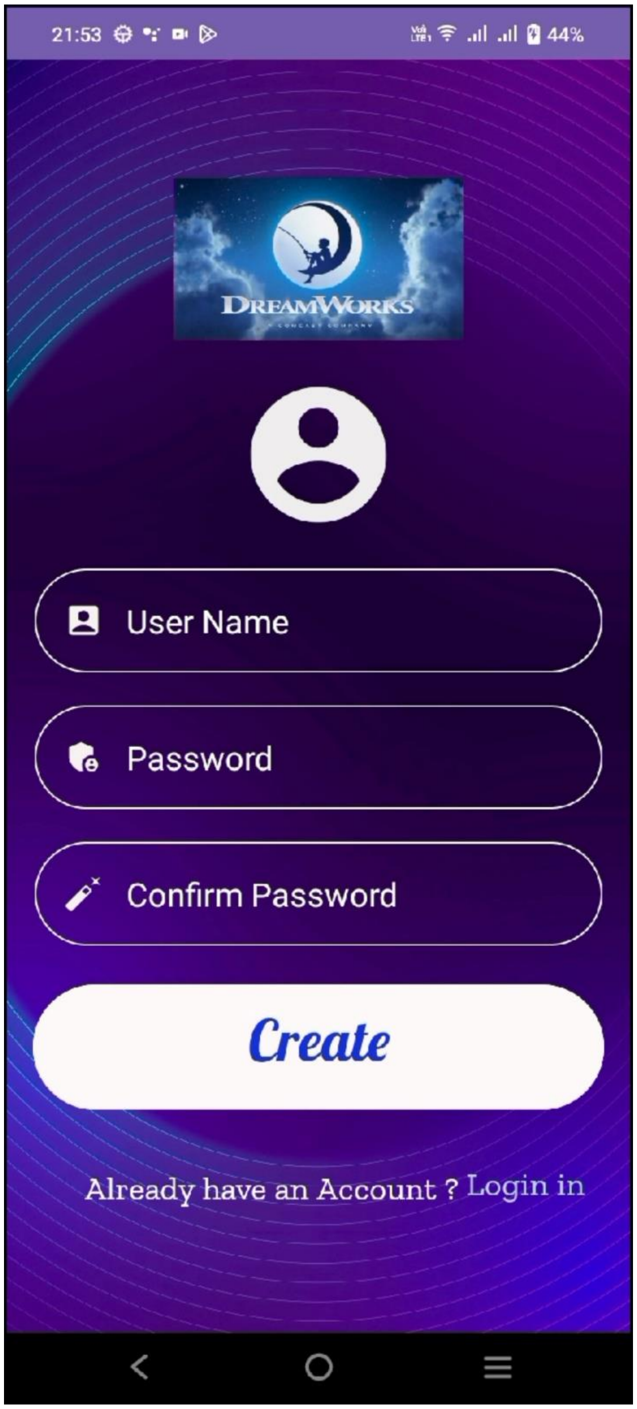
- **build.gradle** – Gradle build configuration file for the project.
- **proguard-rules.pro** – ProGuard configuration file for code obfuscation and optimization.

User Interface

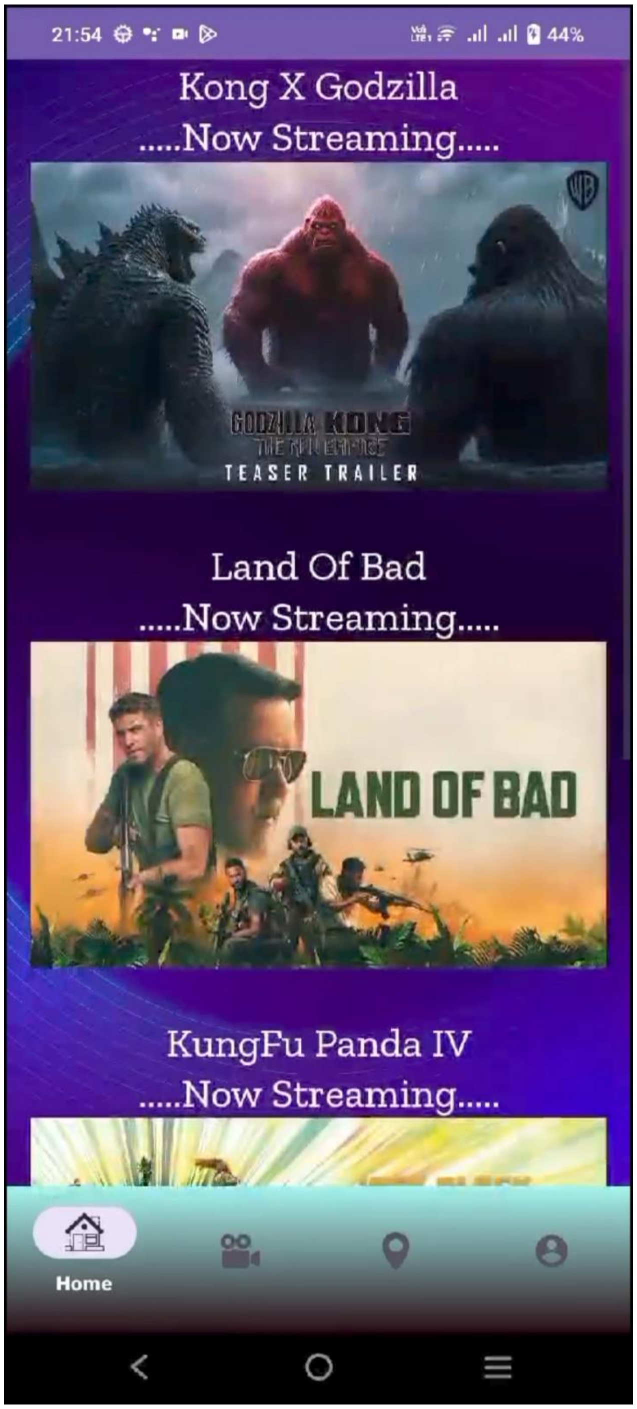
Login Screen



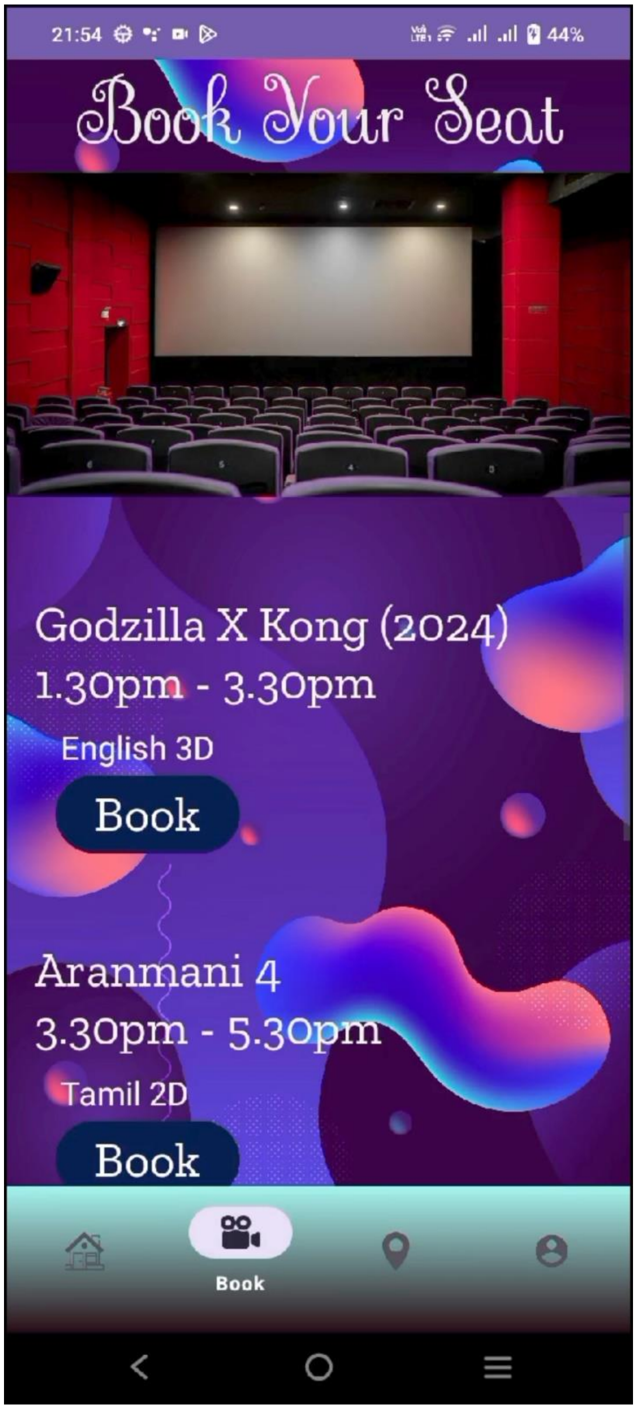
Register Screen

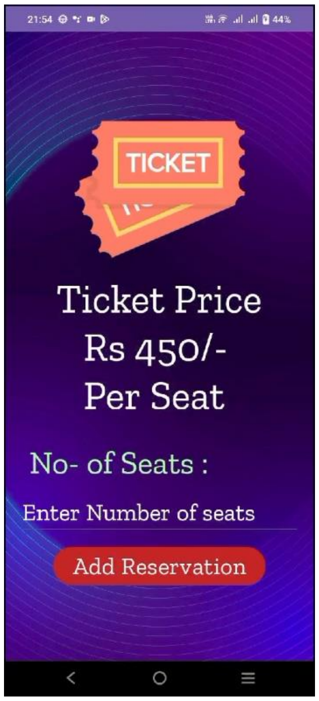


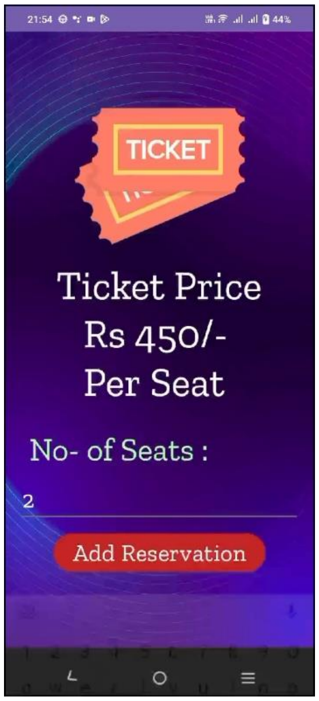
Home

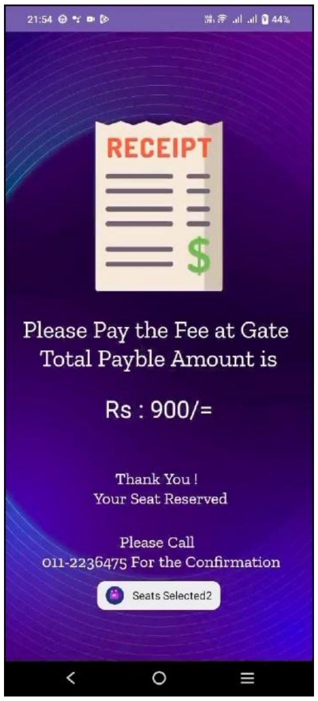

21:54
44%
Kong X Godzilla
.....Now Streaming.....
GODZILLA KONG
TEASER TRAILER
Land Of Bad
.....Now Streaming.....
LAND OF BAD
KungFu Panda IV
.....Now Streaming.....
Home

Book

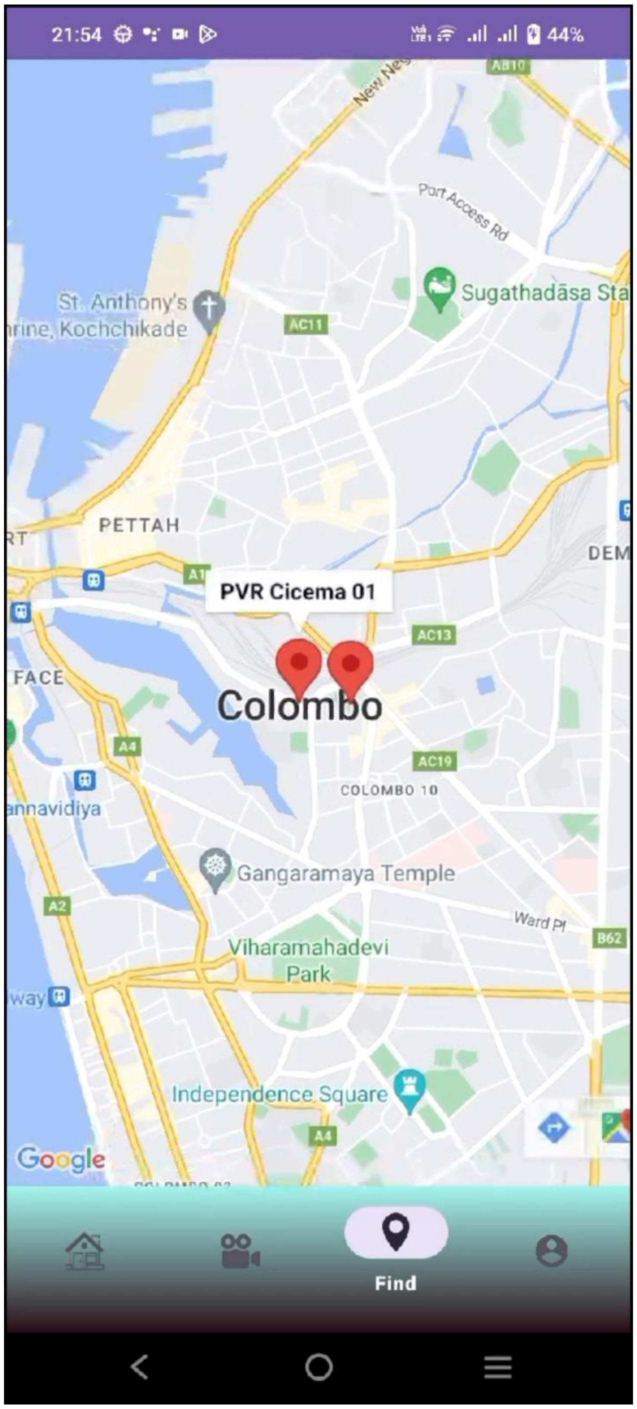

21:54
44%
Book Your Seat
Godzilla X Kong (2024)
1.30pm - 3.30pm
English 3D
Book
Aranmani 4
3.30pm - 5.30pm
Tamil 2D
Book
Book


TICKET
Ticket Price
Rs 450/-
Per Seat
No- of Seats :
Enter Number of seats
Add Reservation

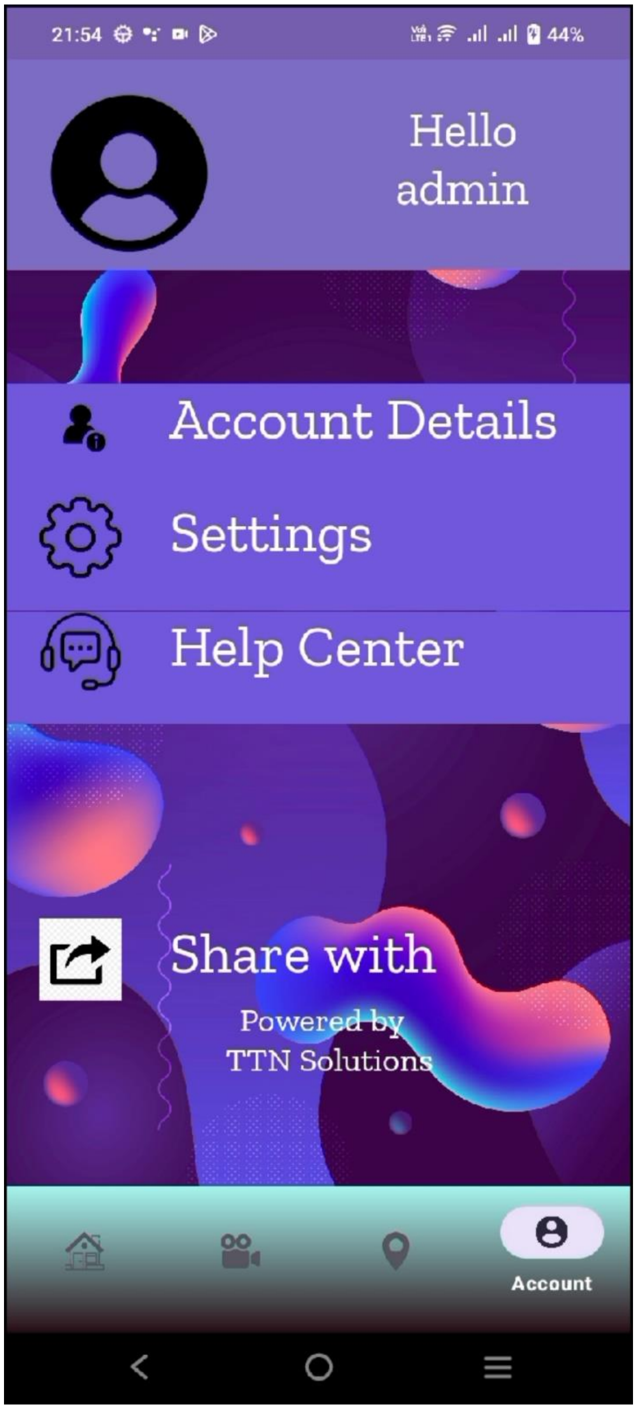

TICKET
Ticket Price
Rs 450/-
Per Seat
No- of Seats :
2
Add Reservation


RECEIPT
Please Pay the Fee at Gate
Total Payble Amount is
Rs : 900/=
Thank You !
Your Seat Reserved
Please Call
011-2236475 For the Confirmation
Seats Selected2

Find


21:54
44%
PVR Cicema 01
Colombo
PETTAH
COLOMBO 10
Gangaramaya Temple
Viharamahadevi Park
Independence Square
Sugathadāsa Sta
St. Anthony's Shrine, Kochchikade
Port Access Rd
Find

Account


21:54
44%
Hello admin
Account Details
Settings
Help Center
Share with
Powered by TTN Solutions
Account

Features and Functionalities

- **User Authentication –**
  - Users can register for an account with a username and password and registered users can log in securely to access the app's features.

- **Home Screen –**
  - Upon successful login, users are directed to the home screen. The home screen may display various options like booking tickets, viewing account details, etc.

- **Ticket Booking and Seat reservation –**
  - Users can book tickets for movies by selecting the desired number of seats. The app calculates the total fee based on the number of seats chosen. Upon confirmation, the app generates a receipt displaying the booking details.

- **Account Management –**
  - Users can view their account details, such as username. New users can register for an account and registered users can log in securely.

- **Movie Location Finder –**
  - The app provides a feature to find nearby movie locations using Google Maps. Users can view the locations of PVR Cinemas on the map.

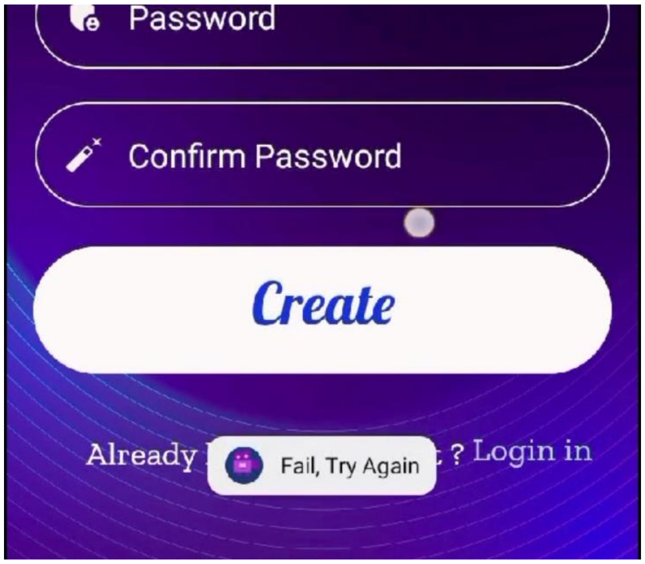
- **Database Management –**
  - The app manages user data using an SQLite database. It stores user credentials securely for authentication. Additionally, it may store movie booking details for future reference.

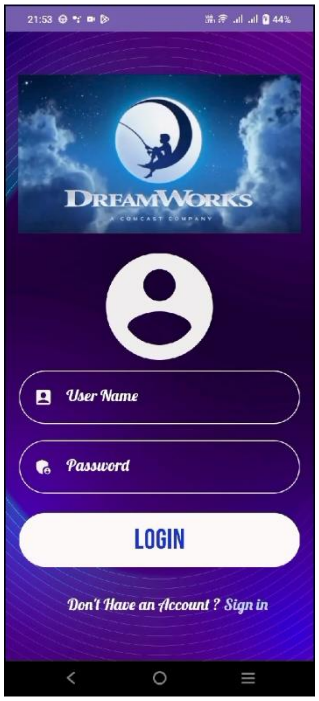
- **Navigation –**
  - The app utilizes a bottom navigation bar for easy navigation between different screens. Users can switch between home, booking, find, and account screens effortlessly.

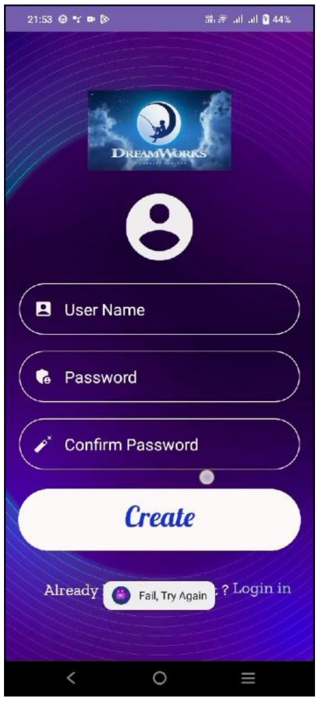
- **Toast Notifications –**
  - Toast messages are displayed to provide feedback to users, such as successful login, registration, or errors during the process.

Testing Results

First, trying to login with incorrect credentials, and it says “Fail, Try Again”

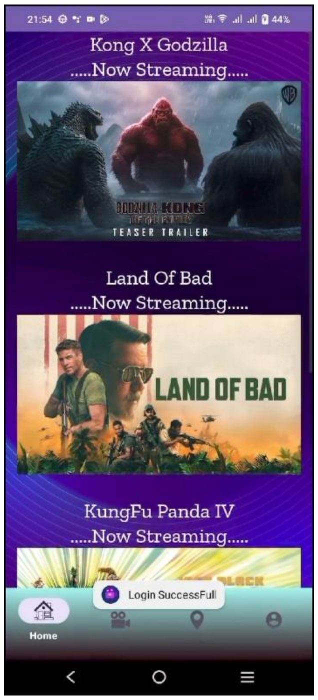


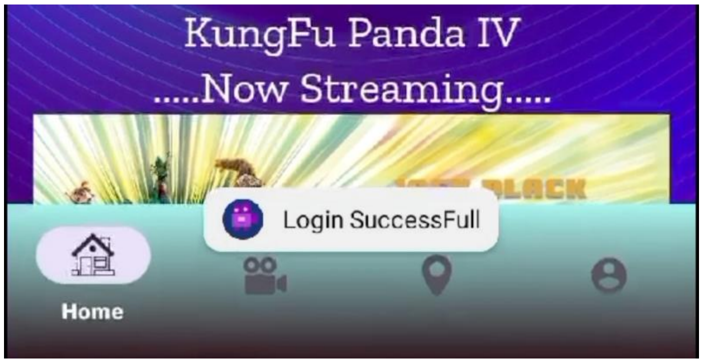




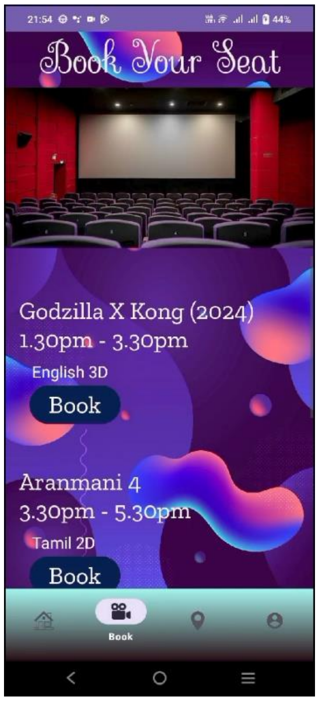
Then choose “Sign in” to create a new account, entering the credentials, press “Create” button. The “User Created” toast message can be seen.

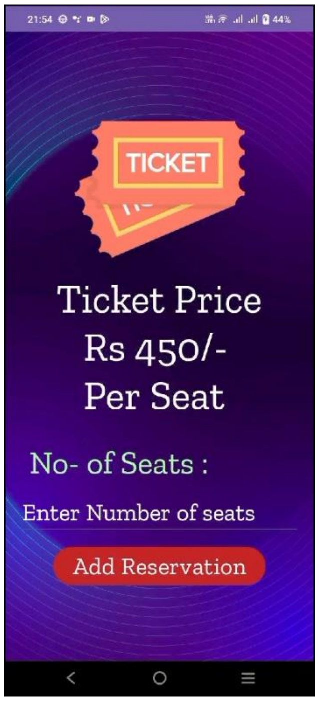
Then entering those credentials in “Login” display. Entered to “Home” screen and “Login Successful” toast message can be seen.

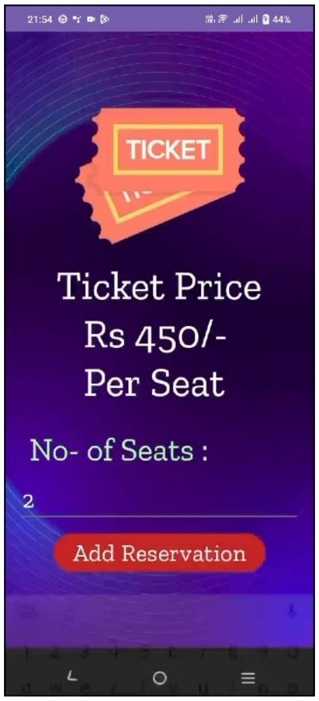




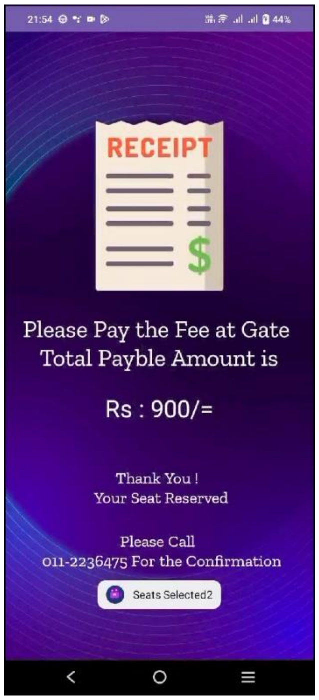
Go to "Book" screen and select (Book) a movie.
In here, entering the number of seats (giving 2) for seats reservation.







It shows amount of fee user has to pay at the gate and some other information (contact information for any confirmation). The "Seats Selected 2" toast message can be seen.



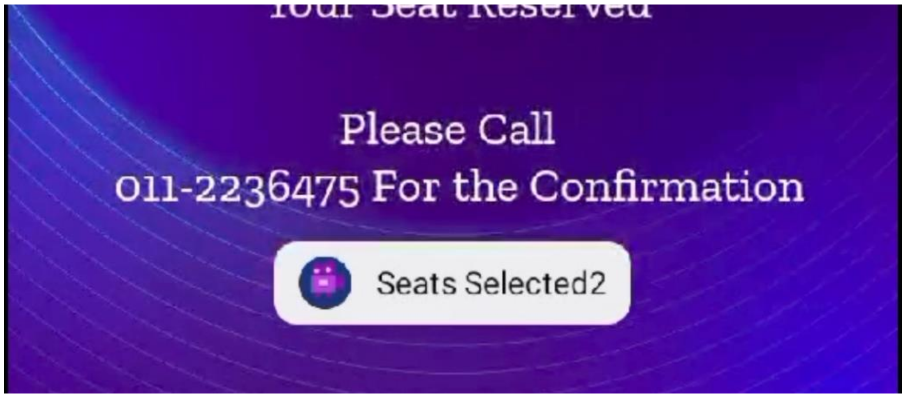
Please Pay the Fee at Gate
Total Payble Amount is

Rs : 900/=

Thank You !
Your Seat Reserved

Please Call
011-2236475 For the Confirmation

Seats Selected2

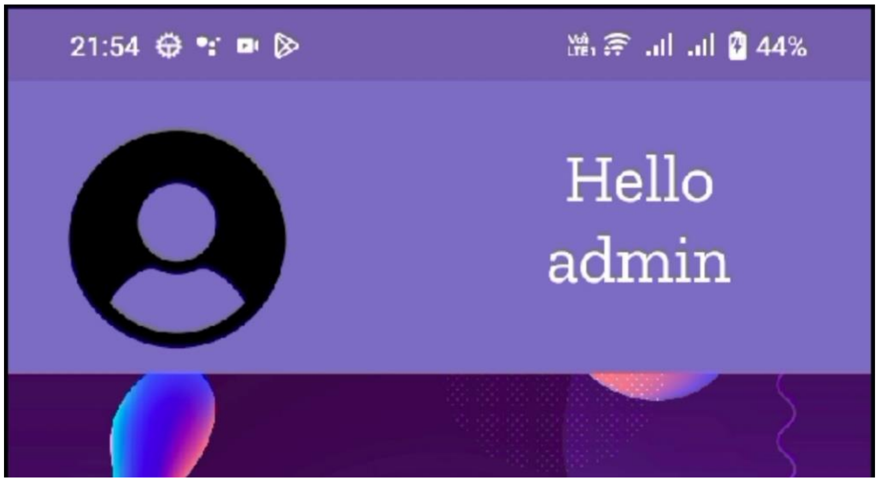


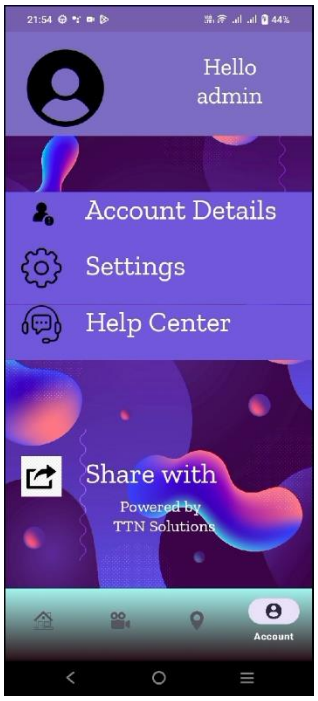
Your Seat Reserved

Please Call
011-2236475 For the Confirmation

Seats Selected2

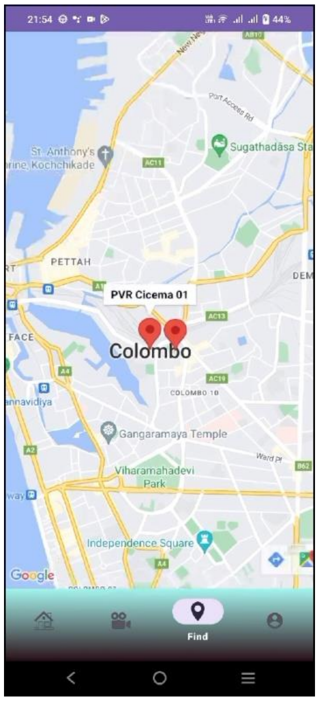
Then select "Account" option. It shows at the top right corner on the screen as "Hello admin2". The "admin2" is the username that I used to login to this app.

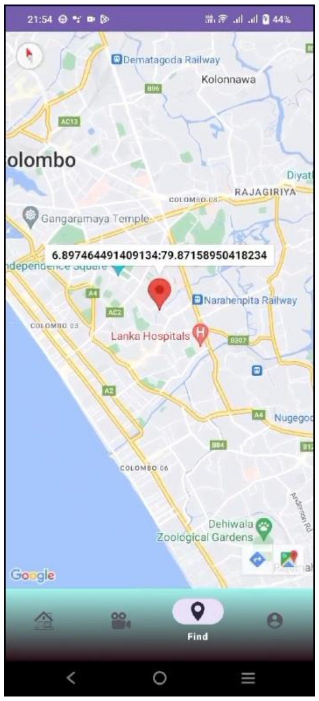


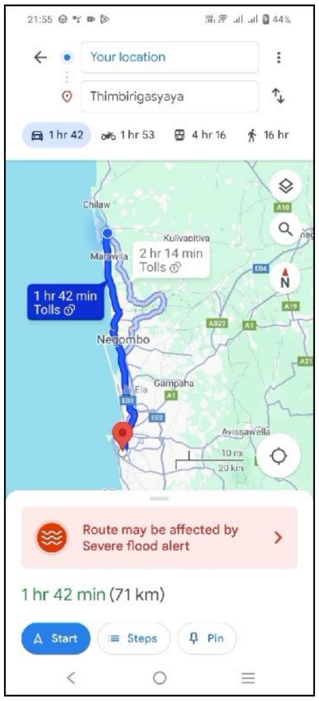


Then select "Find" option. It shows available movie theaters in the map.

By selecting one them, the user can see the distance from his/her current location, the time it takes to reach there, and the how to reach there.







Conclusion

The "**BookMyTicket**" project successfully developed a user-friendly mobile application for seamless movie ticket booking, and seat reservation; addressing key challenges associated with traditional ticket purchasing methods. By leveraging modern technologies such as Android Studio, SQLite, PHP, and Google Maps API, the app provides users with an efficient and secure platform to browse movie listings, select seats, and complete bookings with ease. Despite encountering limitations such as a relatively simple feature set, basic security measures, and a modest user interface design, the project laid a solid foundation for future improvements.

To enhance the app's competitiveness, future work should focus on implementing advanced features like user reviews, secure payment integration, robust error handling, and a more sophisticated UI/UX design. Additionally, adopting modern technologies like Progressive Web Apps (PWA), cloud-based scalability solutions, and continuous integration and deployment (CI/CD) pipelines will further optimize the app's performance, security, and user experience. Overall, "**BookMyTicket**" demonstrates significant potential to streamline the movie ticket booking process and deliver value to both users and cinema operators.

This app has able to cover many of the initial objectives. But there are some it could not achieved.

- Real-time movie listing
- Payment Integration
- User ratings/feedbacks

These require further knowledge of technical terms (APIs, frameworks, etc.) and legal background. So, these could not achieve in first 3 weeks of application development. Beside that, this application provides solid foundation for any further improvement or enhancement as it has included basic essential features and functionalities.

Key achievements

- The app appears to have a straightforward user interface with a bottom navigation bar, making it easy for users to navigate between different sections.

- The app covers essential functionalities such as user authentication, registration, booking tickets, and viewing account details, which are commonly expected in similar apps.

- Integration with Google Maps allows users to view movie theater locations, enhancing the user experience and providing additional utility.

- The app utilizes SQLite database for storing user login information, which can improve performance and offline accessibility compared to relying solely on server-side databases.

- The app provides feedback to users through toast messages, informing them about the success or failure of various actions, which can improve user experience by providing clear indications of the system's response.

Downsides

- As Android app development requires relatively powerful machines, our developers faced huge disadvantage when try to build a competitive mobile application to the market. This might be the main factor for shaping the end result.

- The app has a relatively limited scope of functionalities compared to other existing similar apps. For example, it lacks advanced features such as movie reviews, advanced search options, user profiles, or payment integration.

- While the app implements basic user authentication and registration, the security measures might be limited. For instance, there may be vulnerabilities such as SQL injection if input validation and sanitization are not implemented rigorously.

- While the user interface appears simple, the design might lack sophistication and visual appeal compared to other similar apps. Improvements in UI/UX design could enhance user engagement and satisfaction.

- The app seems to handle errors through toast messages, which may not always provide sufficient information to users for troubleshooting. Implementing more robust error handling mechanisms could improve user experience, especially in cases of unexpected errors or network failures.

- The app's architecture and design are not be optimized for scalability, which could limit its ability to accommodate a growing user base or additional features in the future. Considerations for scalability should be taken into account for long-term viability.

Future Enhancements

Enhanced User Experience (UX) Design

Implement modern and intuitive UI/UX design principles to make the app more visually appealing and user-friendly. Utilize tools like Adobe XD, Sketch, or Figma for designing prototypes and mockups, allowing for rapid iteration and feedback gathering. Employ responsive design techniques to ensure the app functions seamlessly across various devices and screen sizes.

Performance Optimization

Utilize performance optimization techniques such as code minification, image compression, and lazy loading to reduce app load times and improve responsiveness. Implement caching mechanisms, both on the client-side (e.g., browser caching) and server-side (e.g., CDN caching), to reduce server load and improve content delivery speed. Employ tools like Google PageSpeed Insights or Lighthouse to identify performance bottlenecks and prioritize optimization efforts.

Security Enhancements

Integrate robust security measures such as HTTPS, data encryption, and secure authentication mechanisms (e.g., OAuth 2.0, JSON Web Tokens) to protect user data and prevent unauthorized access. Implement secure coding practices and conduct regular security audits to identify and address potential vulnerabilities in the app's codebase. Consider utilizing security-focused libraries and frameworks such as OWASP Dependency-Check, Bcrypt, or SQL Injection Prevention libraries to mitigate common security risks.

Scalability and Reliability

Architect the app using scalable and resilient design patterns such as microservices architecture or serverless computing to handle increasing user loads and ensure high availability. Leverage cloud platforms like AWS, Google Cloud Platform, or Microsoft Azure to dynamically scale resources based on demand and improve reliability. Implement automated monitoring and alerting systems (e.g., Prometheus, Grafana) to proactively identify and address performance issues or downtime.

Integration with Third-Party Services

Explore opportunities to integrate with popular third-party services and APIs (e.g., payment gateways, social media platforms, analytics tools) to enhance the app's functionality and provide additional value to users. Utilize API management platforms like Apigee or AWS API Gateway to streamline API development, versioning, and access control.

Adoption of Progressive Web App (PWA) Technologies

Consider converting the app into a Progressive Web App (PWA) to provide users with a native-like experience, including offline access, push notifications, and installation prompts. Leverage technologies such as Service Workers, Web App Manifests, and IndexedDB to enable PWA features and improve user engagement and retention.

Continuous Integration and Deployment (CI/CD)

Implement CI/CD pipelines using tools like Jenkins, Travis CI, or GitHub Actions to automate the build, test, and deployment processes. Adopt DevOps practices to facilitate faster release cycles, improve code quality, and reduce time-to-market for new features and updates.