

Campus Connect

Table of Contents

Abstract.....	4
Introduction.....	5
Problem Statement.....	6
Objectives.....	7
System Requirements.....	8
Hardware Requirements.....	8
Software Requirements.....	8
System Architecture.....	9
Database Design.....	10
Modules, Features and Design.....	11
User Authentication.....	12
Event Listings.....	13
Booking System.....	14
Notifications.....	15
User Interface Design.....	16
Implementation Details.....	20
Testing and Evaluation.....	21
Results and Analysis.....	22
Challenges and Solutions.....	23
Future Enhancements.....	24
Conclusion.....	25

Abstract

Campus Connect is an Android-based event ticketing application designed to streamline the process of event management and ticket booking for Goa University. The app aims to centralize event listings and ticket reservations for GU events, encouraging greater engagement among students, faculty, and campus organizations. Built using Firebase as the backend database, *Campus Connect* enables real-time event updates, secure user authentication, and seamless booking functionalities. Users can view upcoming events, book tickets, receive notifications, and manage their bookings in an intuitive interface, enhancing the overall event experience on campus. The use of Firebase ensures reliable data storage, efficient synchronization, and scalability, allowing the app to cater to large campus populations while maintaining high performance. *Campus Connect* addresses the challenges of traditional event management systems by providing a digital platform that simplifies event discovery, booking, and notifications, thereby supporting a more connected and vibrant campus life.

Introduction



Campus Connect is an event ticketing application developed exclusively for Goa University, designed to enhance event management and encourage campus-wide participation. Goa University hosts a variety of events—ranging from academic seminars and cultural festivals to sports competitions and club activities—that require a reliable, centralized system for event announcements and ticket reservations. Traditionally, managing these events has posed challenges, such as coordinating ticket sales, ensuring effective communication with students, and handling last-minute updates.

This app addresses these challenges by providing a user-friendly platform where students, faculty, and campus organizations can easily access event information, reserve tickets, and receive timely updates. Campus Connect not only streamlines ticket bookings but also enhances communication by delivering real-time notifications about events and any changes to schedules or locations. By leveraging Firebase as its backend, the app ensures secure user authentication, efficient data handling, and scalable support for large volumes of users.

With Campus Connect, Goa University aims to simplify event management processes and promote a more connected and engaging campus environment, encouraging greater student involvement and strengthening community bonds.

Problem Statement



At Goa University, managing events and promoting participation across campus presents several logistical challenges. Currently, event announcements are often fragmented, relying on posters, word of mouth, or isolated social media posts, leading to limited reach and inconsistent communication. Students frequently miss out on event updates or encounter difficulties with ticket booking due to the lack of a centralized platform. Additionally, handling last-minute changes to event schedules or locations is challenging, with no efficient system in place to notify attendees promptly.

The absence of a unified digital solution results in decreased engagement, missed opportunities for students to participate in campus activities, and inefficiencies in the administrative processes for organizers. Goa University requires a streamlined, accessible platform that consolidates event information, manages ticket reservations, and provides real-time updates to address these gaps.

Campus Connect aims to bridge these issues by offering a centralized event management and ticketing system tailored to the needs of the university, enhancing campus engagement, and improving communication and organization.

Objectives

The primary objectives of *Campus Connect*, the event ticketing app for Goa University, are as follows:

1. **Centralize Event Information:** Provide a unified platform where all Goa University events are listed, allowing students and faculty to easily access event details, schedules, and locations.
2. **Simplify Ticket Booking:** Enable users to reserve tickets for events through a seamless, intuitive booking process, reducing reliance on physical tickets and manual registrations.
3. **Enhance Communication:** Implement real-time notifications for users to receive updates on event details, cancellations, and last-minute changes, ensuring attendees are promptly informed.
4. **Ensure Secure Access and Data Management:** Use Firebase Authentication to maintain secure user logins and protect user data, offering a safe environment for all users.
5. **Increase Campus Engagement:** Encourage greater participation in campus activities by making event information and booking more accessible and convenient for students and faculty.
6. **Provide an Efficient Solution for Event Organizers:** Streamline the management process for event organizers at Goa University, making it easier to update event information, track ticket reservations, and engage with attendees.
7. **Support Scalability and Reliability:** Utilize Firebase's database capabilities to handle large numbers of users and ensure reliable performance, especially during high-demand events.

System Requirements

Outlines the requirements for both users and developers to run *Campus Connect* effectively. It includes minimum specifications to ensure smooth performance and lists Android compatibility for widespread accessibility among university students.

Hardware Requirements

- **Development Environment:** Requires a development machine with a minimum i5 processor (or equivalent), 8 GB RAM, Android Studio support, and a stable internet connection for Firebase connectivity.
- **User Devices:** Android devices running Android 5.0 or higher with at least 2 GB RAM, capable of handling moderate app activity and Firebase-driven updates in real-time.

Software Requirements

- **Development Tools:** Android Studio, Firebase SDK (including Authentication, Realtime Database, Firebase, and Cloud Messaging), and Java/Kotlin for Android development.
- **Database:** Firebase, chosen for its real-time data sync and scalability with Android.
- **Testing Tools:** Firebase Test Lab and Android Emulator for simulating various Android versions, as well as physical device testing for hands-on evaluations.

System Architecture

Client-server architecture, specifically for an Android client with a Firebase backend.

- **Client Side (Android App):** Responsible for managing the user interface, handling user interactions, and communicating with Firebase for authentication, data storage, and notifications.
- **Backend (Firebase):** Manages data storage, user authentication, and push notifications, facilitating secure and fast data handling.
- **Data Flow:** A diagram illustrates interactions between the app, Firebase Authentication, Realtime Database, and Cloud Messaging, enabling real-time communication for event updates and bookings.

Database Design

Firebase Structure: Database schema organized around collections (e.g., Users, Events, Bookings, Notifications) for efficient access and data management.

Key Collections:

- **Users:** Stores user credentials, profile information, and user role (student or faculty).
- **Events:** Holds event metadata, including eventID, name, description, date, location, and capacity.
- **Bookings:** Each booking associates a userID with an eventID, recording timestamps and attendance.
- **Notifications:** Stores notification messages for real-time delivery to users regarding event updates.

Database Schema

Users Table

```
CREATE TABLE users (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Admins Table

```
CREATE TABLE admins (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Events Table

```
CREATE TABLE events (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    description TEXT,
    image_url VARCHAR(255),
    capacity INT NOT NULL,
    price DECIMAL(10, 2), -- Can be NULL if the event is free
    date TIMESTAMP,
    created_by BIGINT,
    FOREIGN KEY (created_by) REFERENCES admins(id) ON DELETE SET NULL
);
```

Bookings Table

```
CREATE TABLE bookings (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    user_id BIGINT,
    event_id BIGINT,
    num_tickets INT NOT NULL,
    total_price DECIMAL(10, 2),
    booking_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (event_id) REFERENCES events(id)
);
```

Tickets Table

```
CREATE TABLE tickets (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    user_id BIGINT NOT NULL,
    event_id BIGINT NOT NULL,
    Name VARCHAR(255),
    department VARCHAR(255),
    class VARCHAR(255),
    Issue_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Event_date TIMESTAMP,
    paymentId BIGINT,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (event_id) REFERENCES events(id),
    FOREIGN KEY (paymentId) REFERENCES payments(id),
    FOREIGN KEY (event_date) REFERENCES events(date)
);
```

Payments Table

```
CREATE TABLE payments (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    booking_id BIGINT,
    payment_method VARCHAR(100),
    amount DECIMAL(10, 2) NOT NULL,
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (booking_id) REFERENCES bookings(id)
);
```

Modules, Features and Design

- **User Authentication:** Secure login via Firebase Authentication, offering email/password and Google login for Android.
- **Event Listings:** A real-time list of university events categorized by type, date, or location, allowing students to browse easily.
- **Booking System:** Provides users the ability to reserve tickets, track availability, and manage reservations.
- **Notifications:** Real-time alerts for new events, bookings, and updates using Firebase Cloud Messaging.

Figma: All interfaces are designed in Figma, which provides a flexible, collaborative environment for prototyping and refining the app's design.

-

User Authentication

- **Firebase Authentication Integration:** Manages secure user logins, with Firebase handling the authentication logic, minimizing the need for additional security code on Android.
- **Persistent Login:** Firebase provides token-based session management, allowing users to stay logged in between sessions, enhancing the Android app's user experience.

Welcome Back

Login to your Account

User ID

Password

Remember me

Log in

Forgot Password ?

Register

Create your new account

Full Name

Email

Password

Log in

Or Continue with

G f

Dont Have Account ? [Sign Up](#) with.

Already Have Account ? [Sign in](#)

Event Listings

- **Dynamic Listings:** Displays real-time event data from Firebase, filtering options by category (academic, cultural, etc.) to facilitate quick access.
- **Event Details Page:** Each event includes a detailed page with description, time, location, organizer info, and a booking button for reservations.

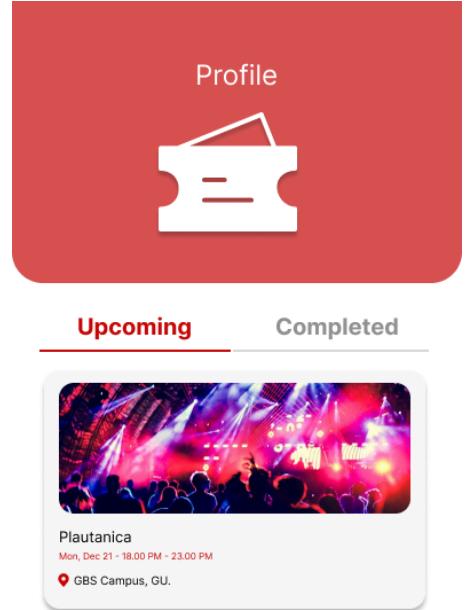
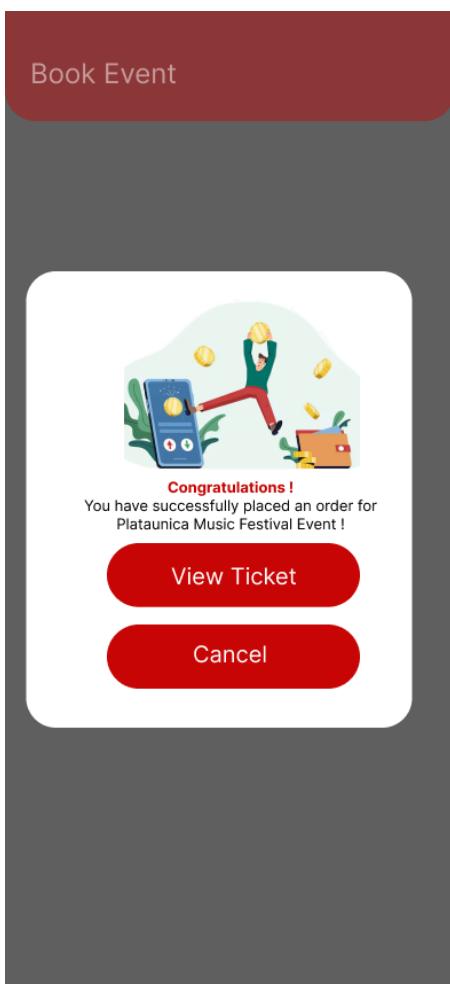
The image displays a grid of six event cards, each representing a different type of event listing. Each card includes a thumbnail image, the event name, a brief description, the date and time, and a red 'BOOK' button.

- Underground Party by GU**
Fri 20th 10am to 2pm **BOOK**
- Devfest Goa 2022**
Sun | Sand | Tech **BOOK**
- Waves**
Fri 17th 10am to 2pm **BOOK**
- Business Conference**
Fri 17th 10am to 2pm **BOOK**
- Business Marketing**
Fri 17th 10am to 2pm **BOOK**
- Electronic Music**
SATURDAY, DECEMBER 15 **BOOK**

At the bottom of the grid, there are four small icons: a house, a heart, a ticket, and a person.

Booking System

- **Booking Interface:** Users can view ticket availability and reserve or cancel bookings. Firebase's real-time updates prevent overbooking and provide instant feedback on ticket status.
- **Confirmation and Management:** Booked tickets are stored under the user's profile, allowing easy review and cancellation within the app.



Go Home



Notifications

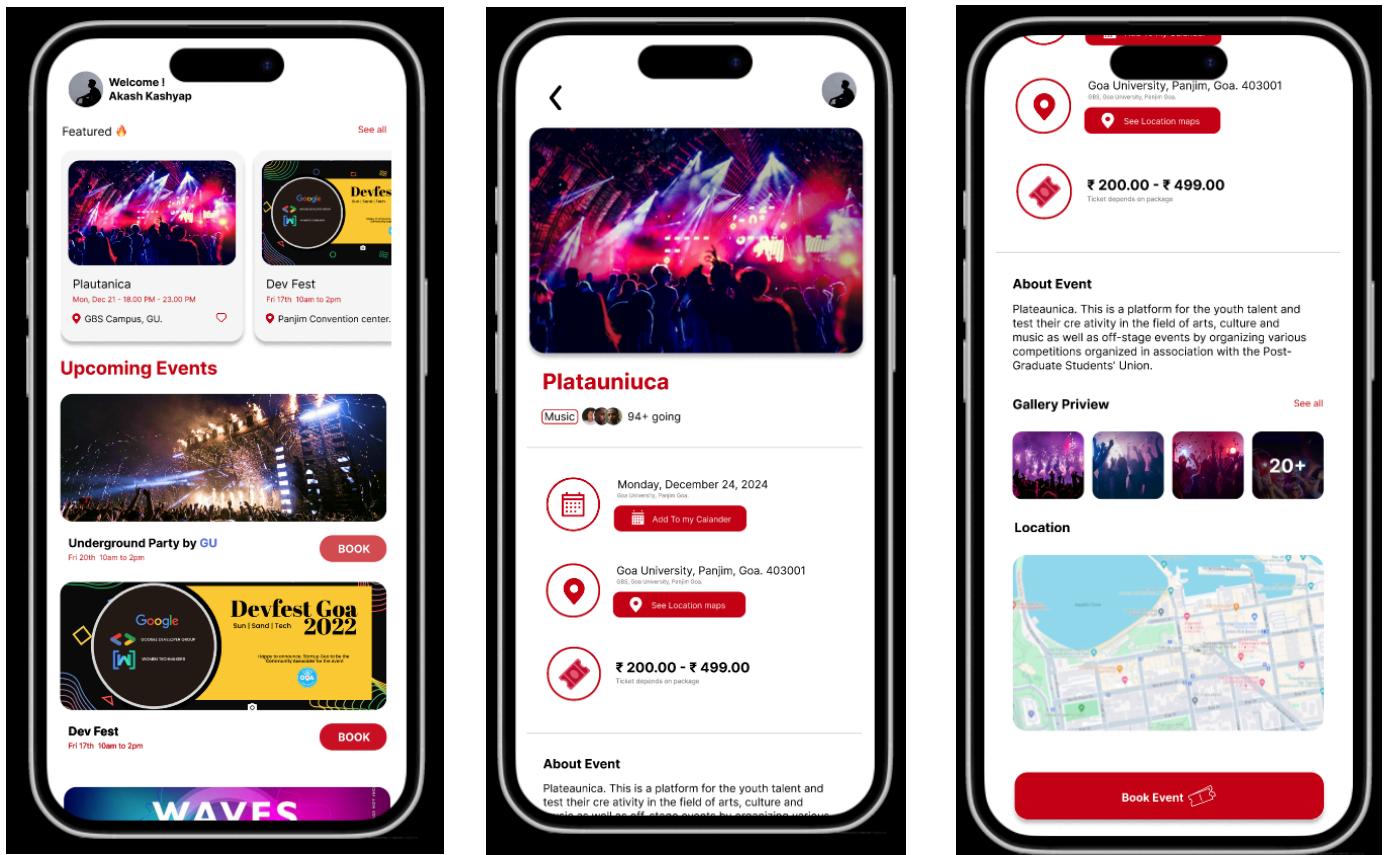
- **Firebase Cloud Messaging (FCM)**: Configured to send push notifications directly to Android devices for event announcements, booking confirmations, and reminders.
- **In-App Notifications Center**: Provides an organized notification feed within the app, displaying both push notifications and in-app messages.

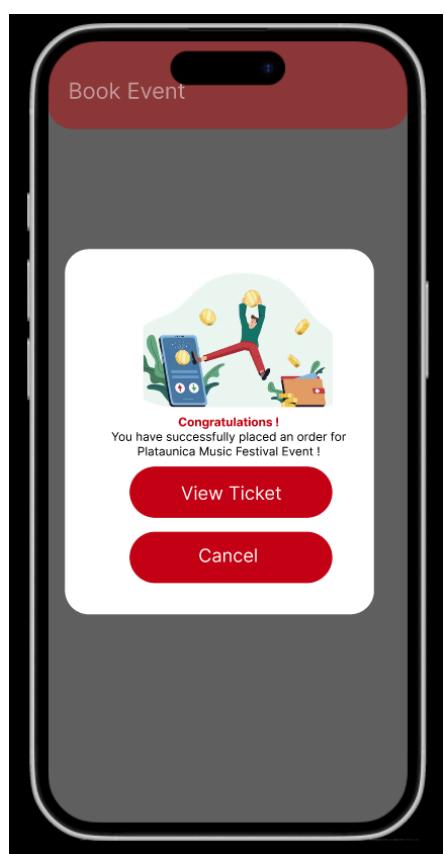
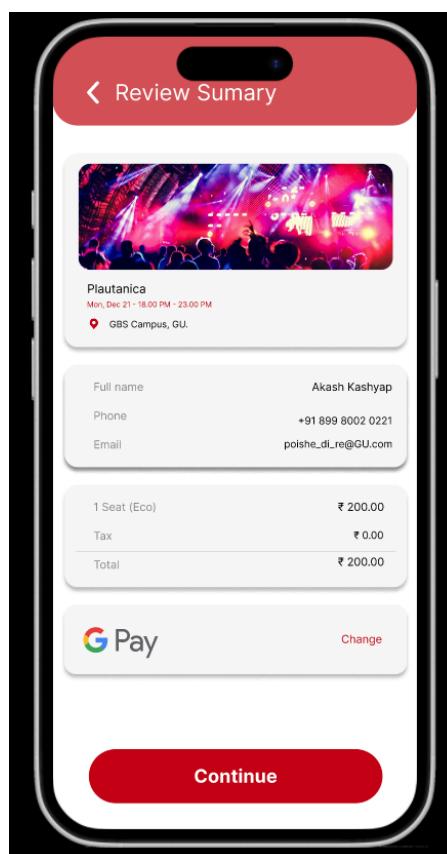
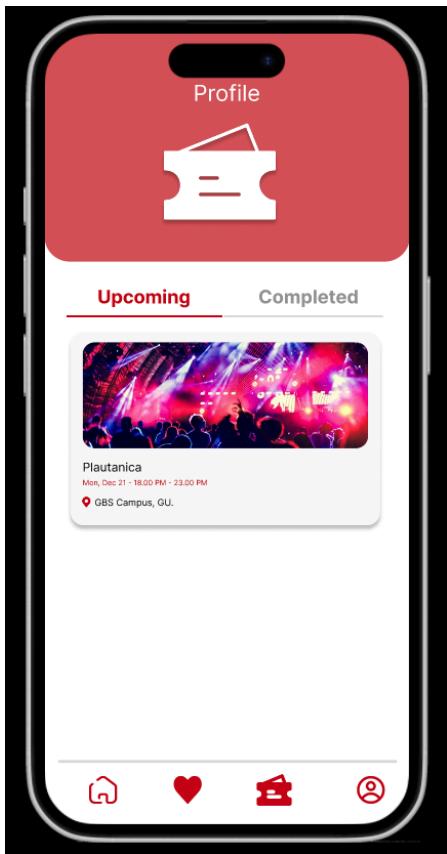
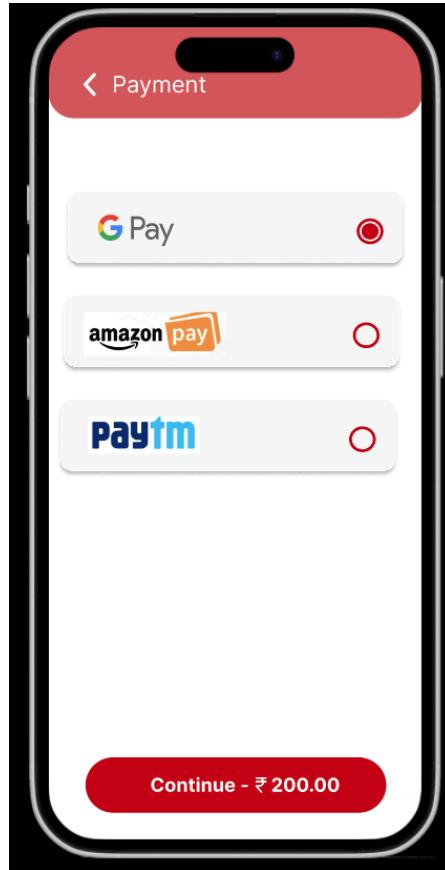
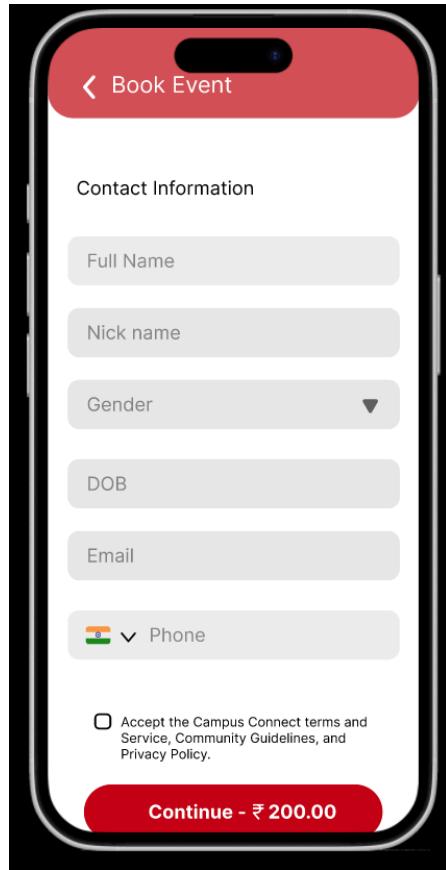
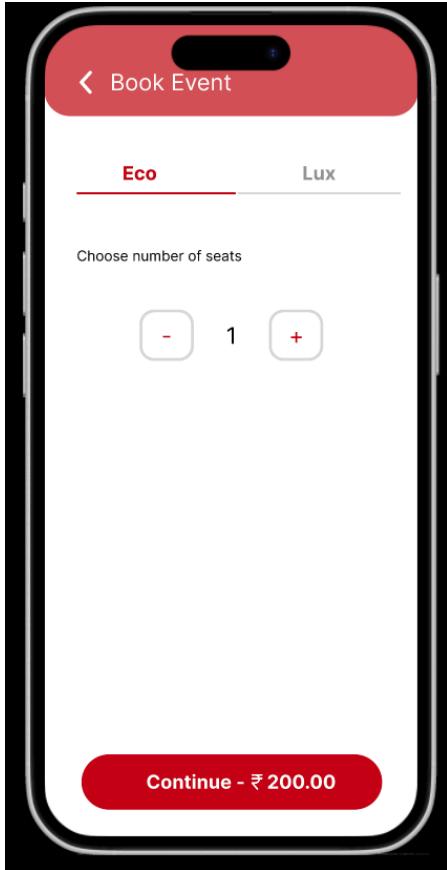
◀ Notification

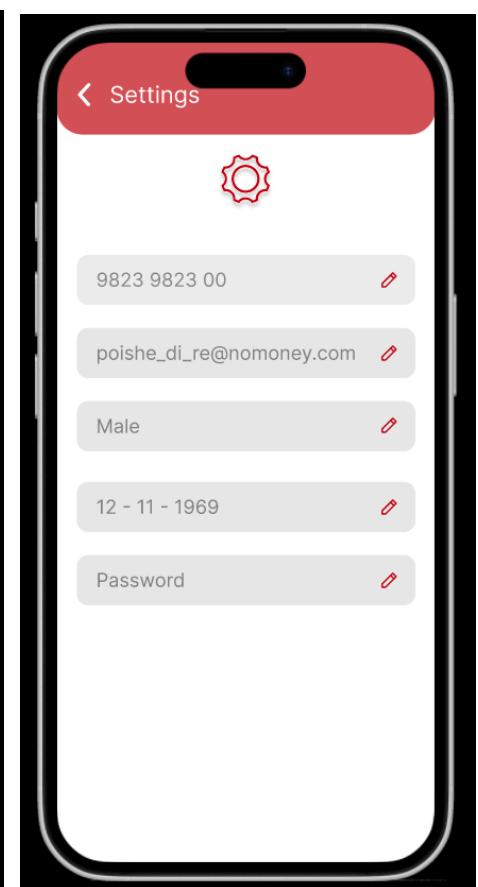
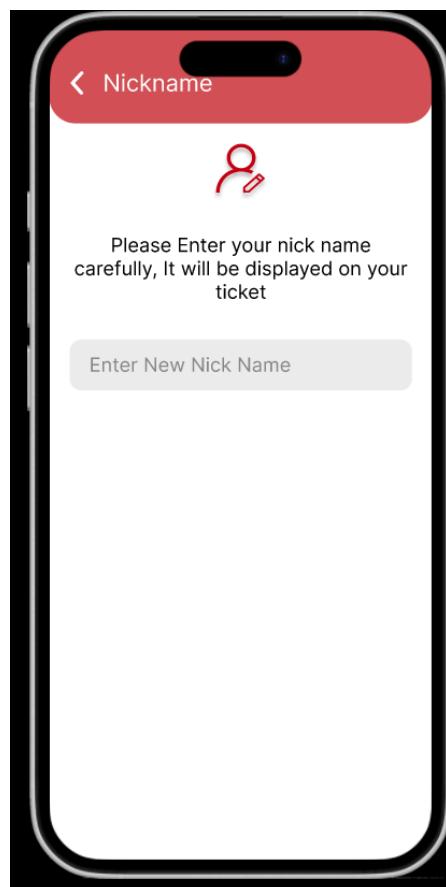
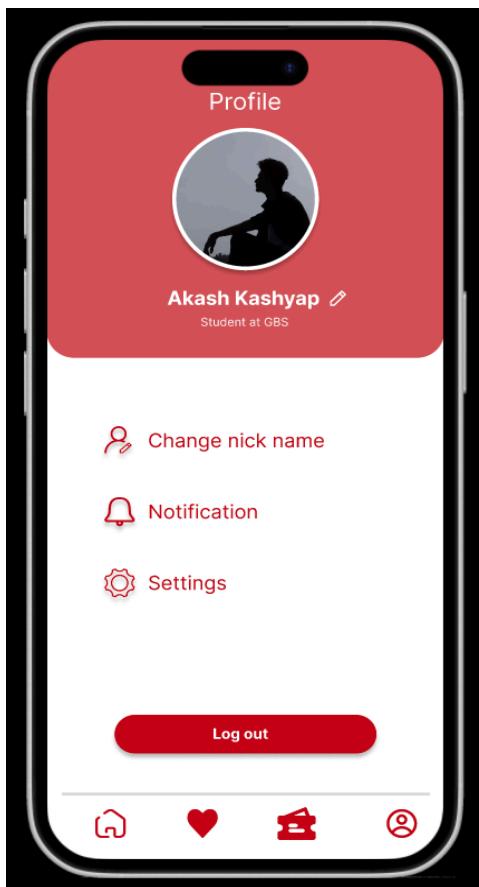
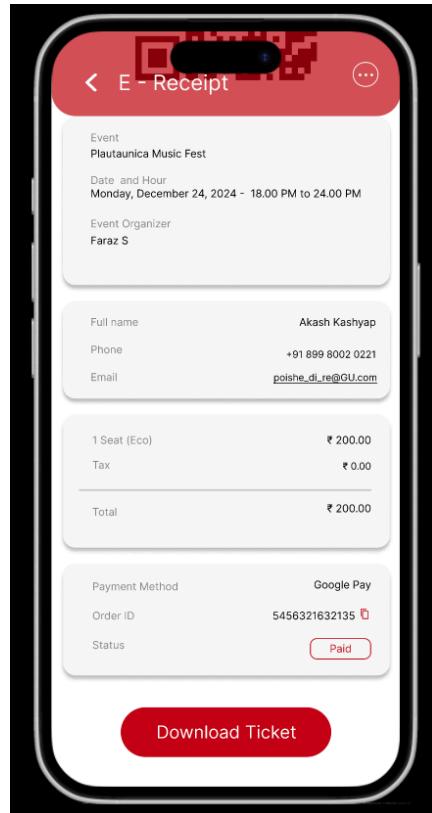
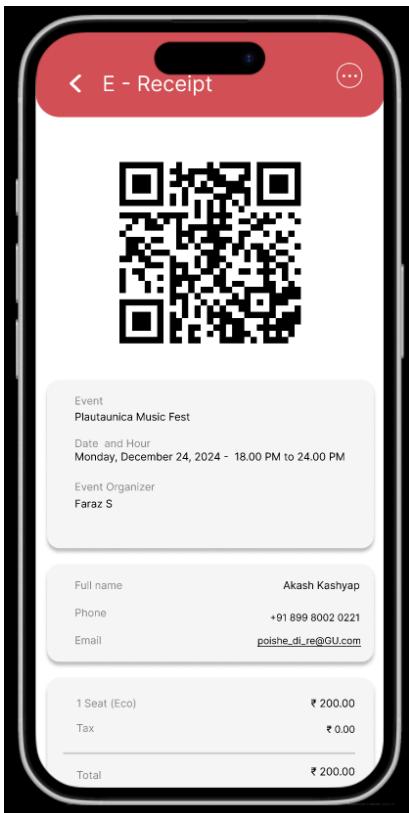


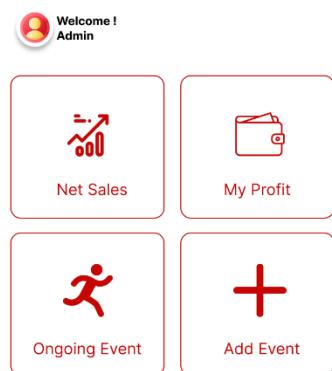
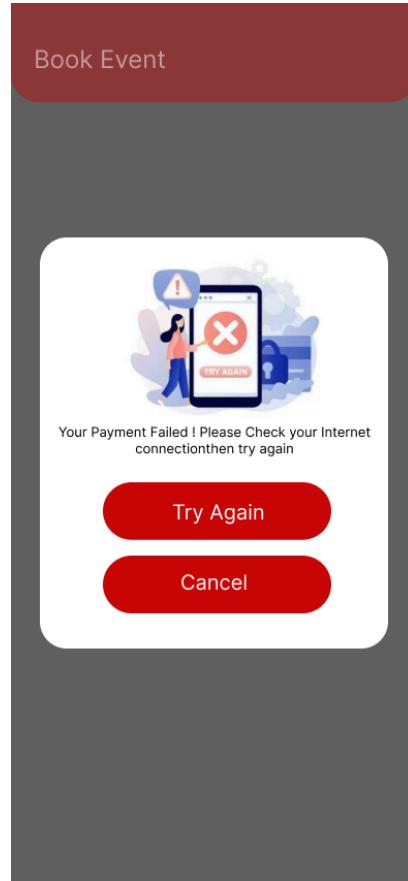
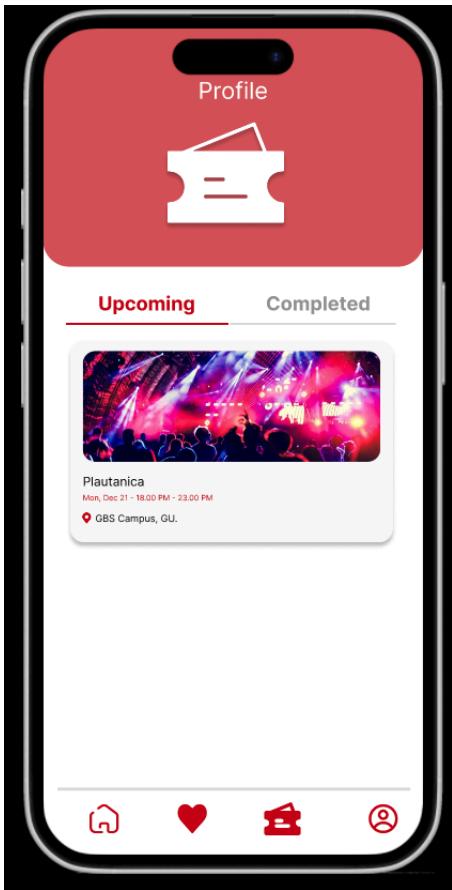
User Interface Design

- **Design Principles:** Built using Google's Material Design guidelines, ensuring a consistent, intuitive, and visually appealing experience for Android users.
- **Screens:** Login, Event Listings, Event Details, Booking Confirmation, Profile, and Notifications center screens are designed for ease of navigation and usability.
- **Mockups:** Presenting design mockups or screenshots for each screen to illustrate navigation flows and user interactions.









Implementation Details

- **Programming Language:** Implemented in Java or Kotlin, leveraging Android Studio's development tools for Android app development.
- **Firebase SDK:** Configured to integrate Firebase Authentication, Firebase, and Cloud Messaging to enable real-time data sync and secure login.
- **Development Stages:** Development progressed through UI/UX design, database setup, feature coding, and multi-stage testing on Android emulators and physical devices.

Testing and Evaluation

- **Unit Testing:** Tested individual components like login, booking, and notification to ensure each function operates as expected.
- **Integration Testing:** Verified Firebase's integration with the Android UI, checking data consistency across user sessions.
- **User Acceptance Testing (UAT):** A sample of Goa University students provided feedback on usability, which guided improvements in design and functionality.
- **Performance Testing:** Assessed app performance, load times, and data sync efficiency to ensure smooth usage across various Android devices.

Results and Analysis

- **Usage Metrics:** Tracked metrics like user engagement, ticket booking rates, and notification responses.
- **User Feedback:** High satisfaction levels were noted regarding ease of use, efficient bookings, and quick notifications. Suggestions from user feedback informed final refinements in the UI.

Challenges and Solutions

- **Real-Time Synchronization:** Managed through Firebase's Realtime Database, ensuring accurate booking status updates and preventing duplicate reservations.
- **Push Notification Reliability:** Firebase Cloud Messaging was fine-tuned to deliver timely notifications while minimizing battery impact on Android devices.
- **Offline Functionality:** Implemented limited offline capabilities so users could view previously loaded events without an internet connection.

Future Enhancements

- **Social Media Integration:** Adding features for users to share events on social media to increase visibility and participation.
- **Analytics for Organizers:** Insights for event organizers to track attendance, user demographics, and booking patterns to optimize event planning.
- **Offline Mode Expansion:** Enhanced offline functionality, allowing users to browse events and make reservations even with intermittent internet.
- **Admin Panel for Event Organizers:**
Develop a separate admin panel for event organizers to manage events more efficiently. This could include tools for tracking ticket sales, viewing feedback, and updating event details.
- **In-App Chat Feature:** Implement a chat function that allows users to communicate with each other or with event organizers. This could foster community engagement and facilitate coordination among attendees.
- **Multi-Language Support:** Add support for multiple languages to cater to a more diverse user base, making the app accessible to international students and locals who may prefer different languages.
- **Integration with Campus Services:** Explore partnerships with other campus services (e.g., transport, dining) to provide users with a comprehensive platform that addresses multiple aspects of campus life.
- **Advanced Analytics for Event Organizers:** Offer analytics tools for event organizers to assess event performance, including attendance metrics, demographics of attendees, and feedback trends. This data can help in planning future events more effectively.

Conclusion

Campus Connect effectively addresses the event management needs of Goa University, creating a centralized platform that simplifies event discovery, booking, and notifications. Built for Android and powered by Firebase, the app has improved campus engagement by offering a user-friendly and secure solution that bridges the communication gap for campus events.