# CHAPTER - 1 PROJECT DEFINITION

### INTRODUCTION:

The Hostel Management System Application is a comprehensive and efficient software solution designed to streamline and automate the management of hostels, providing a seamless experience for both hostel administrators and residents. With a wide range of functionalities, this application aims to enhance the overall hostel management process, promoting convenience, transparency, and accountability.

In today's fast-paced world, managing hostels manually can be a daunting and time-consuming task for hostel authorities. The Hostel Management System Application aims to address these challenges by providing a user-friendly platform that simplifies various administrative tasks and ensures a smooth flow of operations.

The application's core functionalities include managing student profiles, facilitating fee management, handling new admissions, and automating room allotment processes. Hostel administrators can easily create and maintain student profiles with essential details, making it easier to keep track of residents and their respective room allocations. Additionally, the system streamlines the fee management process, enabling residents to pay their fees conveniently through the application, and allowing administrators to monitor and track payment records efficiently.

For hostel administrators, handling new admissions is made effortless with the application's intuitive interface, which guides them through the admission process and maintains all relevant information in a centralized database. Moreover, the room allotment feature ensures optimal utilization of hostel facilities, making room allocation a seamless and organized procedure.

The application also caters to the needs of hostel residents, providing functionalities such as leave management, food ordering, and complaint registration. Residents can request leaves through the application, streamlining the approval process and ensuring better communication between hostel authorities and residents. Ordering food becomes hassle-free with the ability to place food orders within the application, offering a convenient and quick way to cater to residents' dietary needs.

Moreover, the complaint registration feature allows residents to report any issues they encounter within the hostel, enabling administrators to promptly address and resolve problems, thereby enhancing resident satisfaction.

The application also incorporates an attendance tracking system, allowing hostel authorities to monitor the attendance of residents efficiently. Additionally, residents can receive important notifications, such as event announcements, deadlines, and general hostel-related information, ensuring effective communication within the hostel community.

Furthermore, the Hostel Management System Application manages light bill records, simplifying billing procedures and facilitating transparent financial transactions.

### PROBLEM STATEMENT:

The existing hostel management system at CHARUSAT University faces numerous challenges that hinder its efficiency and effectiveness. Manual management of resident profiles, fee collection, room allotment, and other administrative tasks consumes a significant amount of time and resources. This traditional approach leads to errors, delays, and difficulties in maintaining accurate records. Additionally, the lack of a streamlined communication system between hostel administrators and residents often results in miscommunication and dissatisfaction among residents.

Furthermore, the absence of an integrated complaint management system leads to delays in addressing resident grievances, impacting their overall living experience. The current system also lacks a centralized platform for resident attendance tracking, hindering administrators' ability to monitor and manage student attendance effectively.

To address these issues, there is a pressing need for a modern and user-friendly Hostel Management System Application that automates administrative tasks, facilitates online fee payments, optimizes room allocation, and streamlines communication between administrators and residents. By creating a comprehensive solution that enhances transparency, reduces manual efforts, and improves communication, the proposed application aims to revolutionize the hostel management process, ensuring a seamless and satisfactory experience for all stakeholders involved.

### OBJECTIVE:

The primary objective of the Hostel Management System Application is to provide a technologically advanced and user-friendly platform that automates and streamlines various hostel management tasks, aiming to enhance efficiency, transparency, and communication within the hostel ecosystem. The specific objectives of the application are as follows:

Automation and Simplification: To automate manual processes such as resident profile creation, fee management, room allotment, and attendance tracking, reducing the administrative burden on hostel authorities and ensuring accurate and timely execution of tasks.

Online Fee Payment: To enable residents to conveniently pay their hostel fees through the application, promoting cashless transactions and eliminating the need for physical fee collection, thus enhancing financial management.

Efficient Room Allotment: To optimize room allocation based on resident preferences, availability, and other relevant factors, ensuring equitable distribution of hostel resources and enhancing resident satisfaction.

Streamlined Communication: To establish a seamless communication channel between hostel administrators and residents, allowing for important notifications, event announcements, and updates to be disseminated effectively.

Complaint Management: To provide a centralized platform for residents to register and track complaints, enabling swift resolution of issues and fostering a positive living experience within the hostel.

Attendance Tracking: To implement a robust attendance tracking system for residents, aiding hostel administrators in monitoring and managing student attendance, leading to improved hostel discipline.

Data Security and Privacy: To prioritize data security and privacy by implementing encryption and access controls, safeguarding sensitive information from unauthorized access.

User-Friendly Interface: To offer an intuitive and easy-to-navigate interface that caters to users with varying technical expertise, ensuring a pleasant and productive user experience.

Real-Time Insights: To provide hostel administrators with real-time data and reports on various aspects of hostel management, empowering them to make informed decisions and implement strategic improvements.

Scalability and Adaptability: To design the application in a scalable and adaptable manner, allowing it to accommodate the needs of hostels of different sizes and catering to potential future enhancements and updates.

By achieving these objectives, the Hostel Management System Application aims to revolutionize hostel management, optimizing administrative processes, fostering effective communication, and ultimately enhancing the overall living experience for hostel residents and administrators alike.

# CHAPTER - 2 DESCRIPTION

### 2.1 DESCRIPTION:

In general, creating a mobile application is a difficult and time-consuming task. There are numerous frameworks available for creating a mobile application. Android provides a native framework based on the Java programming language, whereas iOS provides a native framework based on the Objective-C / Swift programming language.

To develop an application that supports both operating systems, we must code in two different languages and use two different frameworks. There are mobile frameworks that support both operating systems to help with this complexity. These frameworks range in complexity from simple HTML-based hybrid mobile application frameworks (which use HTML for user interface and JavaScript for application logic) to complex language-specific frameworks (which do the heavy lifting of converting code to native code). Regardless of their simplicity or complexity, these frameworks always have numerous drawbacks., one of them being their slow performance.

In this case, **Flutter** - a simple and high-performance Cross-Platform framework developed by Google and based on the **Dart Programming Language** - provides high performance by rendering the UI directly in the operating system's canvas rather than through a **native framework**.

Flutter also provides a plethora of ready-to-use widgets (UI) for building a modern application. These widgets are optimized for mobile environments, and creating an application with them is as simple as creating HTML.

# CHAPTER – 3 FLUTTER SETUP

### Flutter SDK:

* + 1. Navigate to the official website of **Flutter**.
    2. Then, download the latest and stable version of **Flutter SDK**.
    3. Alternatively, we can also use the **git clone** command.
    4. Then, add flutter to the **PATH** environment Variable.
    5. Run **flutter doctor** command in any command line editor to check if the path is updated or nor.

### Android Setup:

* + 1. Navigate to the **Android** **Studio** download page.
    2. The Installation wizard will automatically download all the required **Android SDKs**, command Line tools and **Android SDK build- tools**
    3. Again try to run **flutter doctor** to confirm that flutter has located the **path** of android studio.
    4. If the command cannot locate it, then run **flutter config --android-studio-dir**

**<directory>** to set the directory that Android Studio is installed to.

### Set up Android Device:

* + 1. Enable **Developer Options** and **USB Debugging** on your device.
    2. For windows, install the **Google USB driver**, link is provided in the setup guide.
    3. Using a **USB cable**, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
    4. In the terminal, run the **flutter devices** command to verify that Flutter has recognized the connected android device. By default, flutter uses the version of android SDK where your adb tool is based.

### Set up the Android Emulator:

* + 1. First, Navigate to the flutter setup console.
    2. Enable **VM Acceleration** on your machine.
    3. Launch **Android Studio**, click the **AVD Manager** icon, and select **Create Virtual Device…**
       1. In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device…**. (The **Android** submenu is only present when inside an Android project.)
       2. If you do not have a project open, you can choose **Configure > AVD Manager** and select **Create Virtual Device…**
    4. Choose a device definition and select **Next**.
    5. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86\_64* image is recommended.
    6. Under Emulated Performance, select **Hardware - GLES 2.0** to enable [hardware acceleration.](https://developer.android.com/studio/run/emulator-acceleration)
    7. Verify the AVD configuration is correct, and select **Finish**.
    8. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

# CHAPTER - 4 SOFTWARE AND HARDWARE

# REQUIREMENTS

### 4.1 SOFTWARE REQUIREMENTS:

* + 1. Flutter 3.7.10: - Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase.
    2. Firebase: - Firebase is a platform developed by Google for creating mobile and web applications. Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase’s cloud. The product assists software developers in building real-time, collaborative applications.
    3. VS Code (IDE): - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
    4. Android Studio SDK: - Android Studio is the official integrated development environment (IDE) for Google’s Android operating system, built on JetBrains’ IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (EADT) as the primary IDE for native Android application development.
    5. Android and iOS Emulator
    6. Firebase Database
    7. Realtime Database
    8. Postman

# CHAPTER - 5 MAJOR FUNCTIONALITY

### 5.1 MAJOR FUNCTIONALITY:

**1. Admin:**

- New Admission: Manage the process of admitting new students to the hostel.

- Room Allocation: Allocate rooms to students based on their preferences and availability.

- Leave Approval: Review and approve leave requests from students, staff, and other users.

- Complaint Forwarding: Handle and forward complaints from various users to the relevant authorities.

- Food Menu Setup: Manage and update the food menu for different meal types and days.

- Generate Notifications: Send notifications to users regarding important updates or events.

- Add User Expense: Record and track expenses incurred by different users in the system.

- Manage Fees: Track and manage student fees payments.

- Manage Light Bill: Handle the billing and payment for electricity usage in the hostel.

**2. Student:**

- Add Leave: Request for leave from the hostel and track its approval status.

- Complaints: Lodge complaints related to hostel facilities or other concerns.

- Fees Details: View and check details of the hostel fees and payment history.

- Light Bill Details: Access information about their electricity usage and bills.

**3. Rector:**

- Room Allocation Request: Send requests to the manager or admin for room allocations.

- Leave Approval: Approve or reject leave requests from students or staff.

- Attendance Management: Take attendance of students and staff in the hostel.

- Add Expense: Record expenses for hostel maintenance or other purposes.

- Manage Student Details: Update and maintain student information under their supervision.

- Manage Counsellor Details: Manage information and assignments of counsellors.

- Update Food Menu: Collaborate with admin to update the food menu.

- Food Count: Keep track of the number of students opting for different meals.

**4. Parent:**

- Leave Approval: View and approve leave requests for their child staying in the hostel.

- Child Light Bill: Access and monitor their child's electricity usage and bills.

- Manage Child Details: Update and maintain information about their child in the system.

- Get Notifications: Receive notifications related to their child's hostel activities and updates.

**5. Staff:**

- Complaint Handling: Receive complaints from users and resolve issues promptly.

**6. Counsellor:**

- Student Details: Access information and manage students under their assigned batch.

- Leave Approval: Approve or disapprove leave requests from their assigned students.

**7. Food Manager:**

- Update Food Details: Manage food-related information, such as menu and inventory.

- Food Count Verification: Verify students' hostel affiliation through QR codes for meals.

- Generate Unique QR Codes: Generate unique QR codes for each day and meal type.

**8. Helpers:**

- Specific duties and functionalities based on their roles within the hostel.

These functionalities allow the various users to efficiently manage and interact with the hostel system, ensuring smooth operations and enhanced communication.

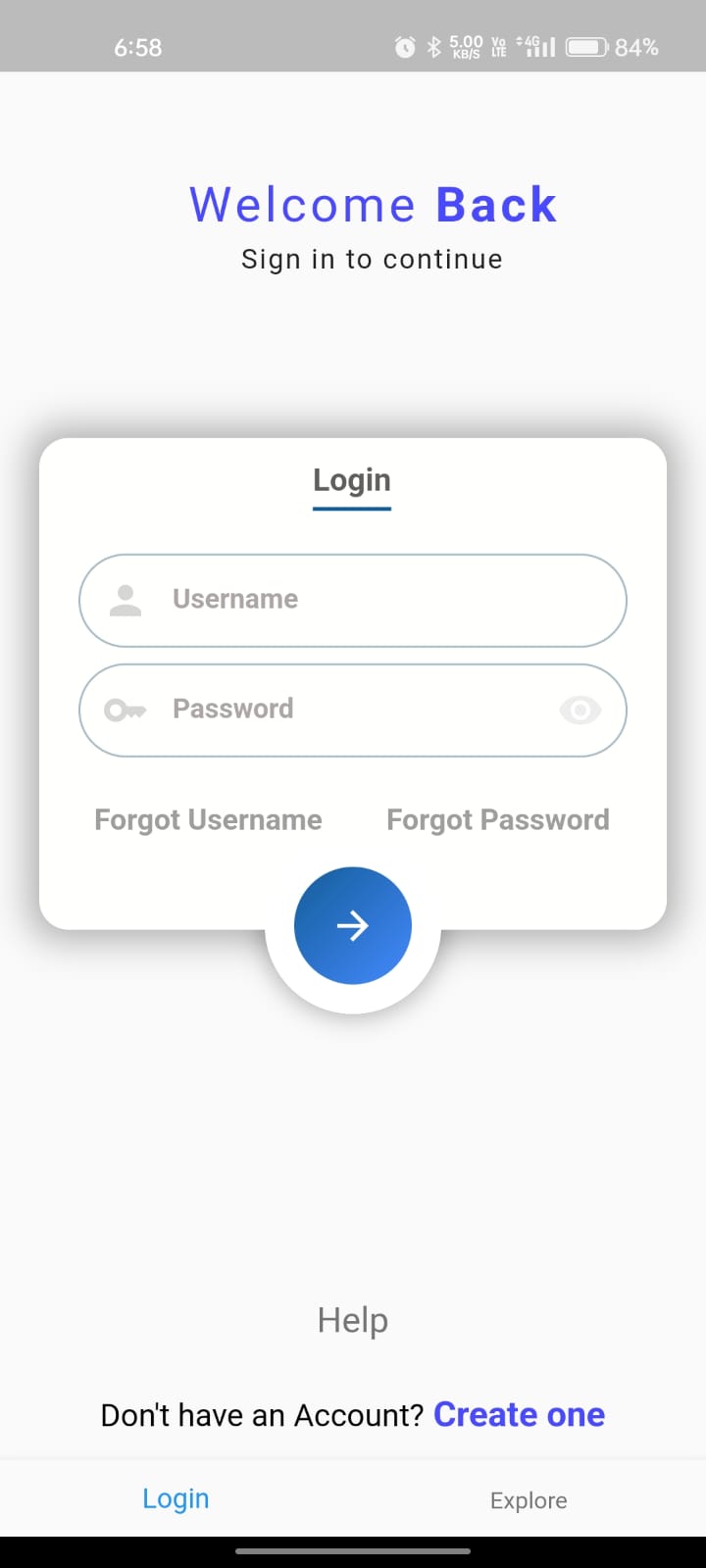
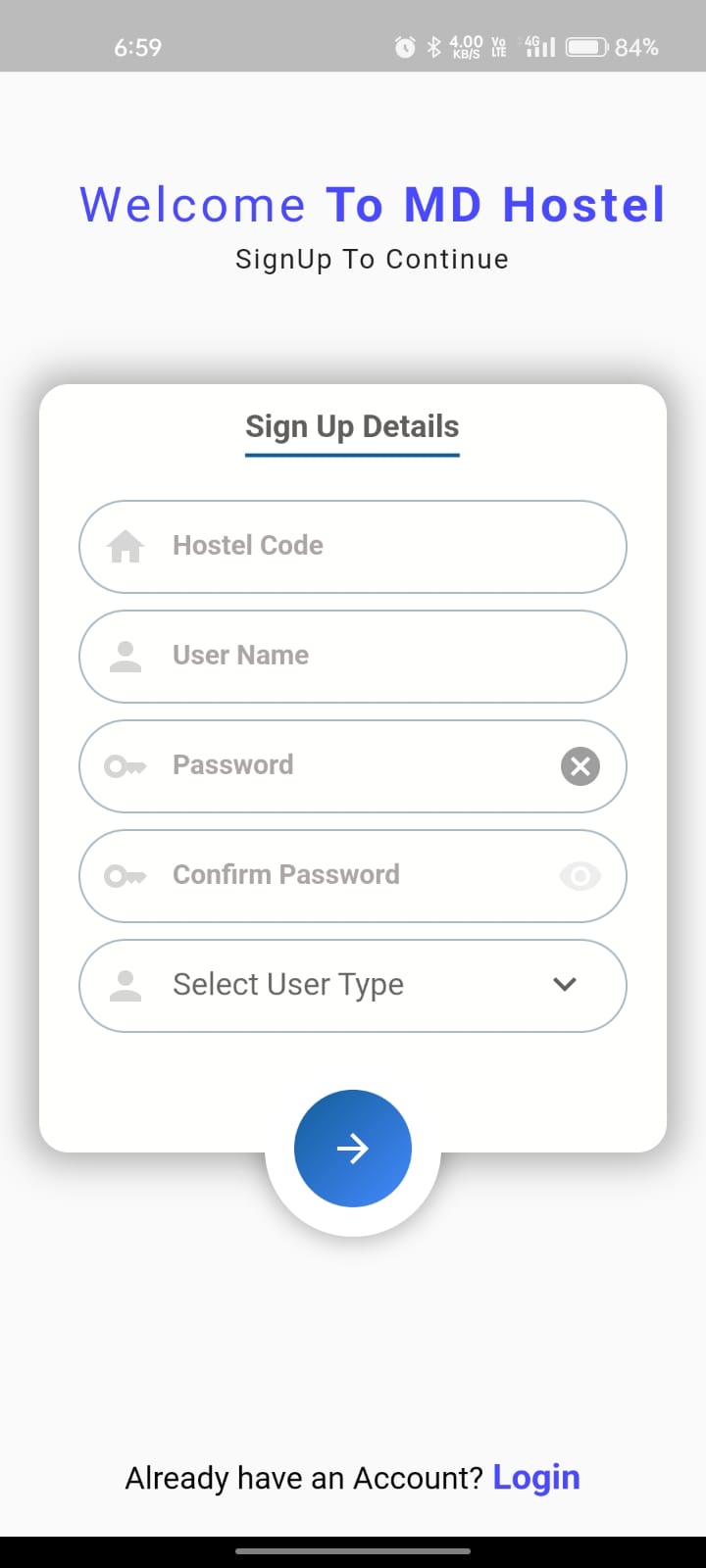
# CHAPTER – 6 Flow Chart

# 6.1 Flow Chart of project:

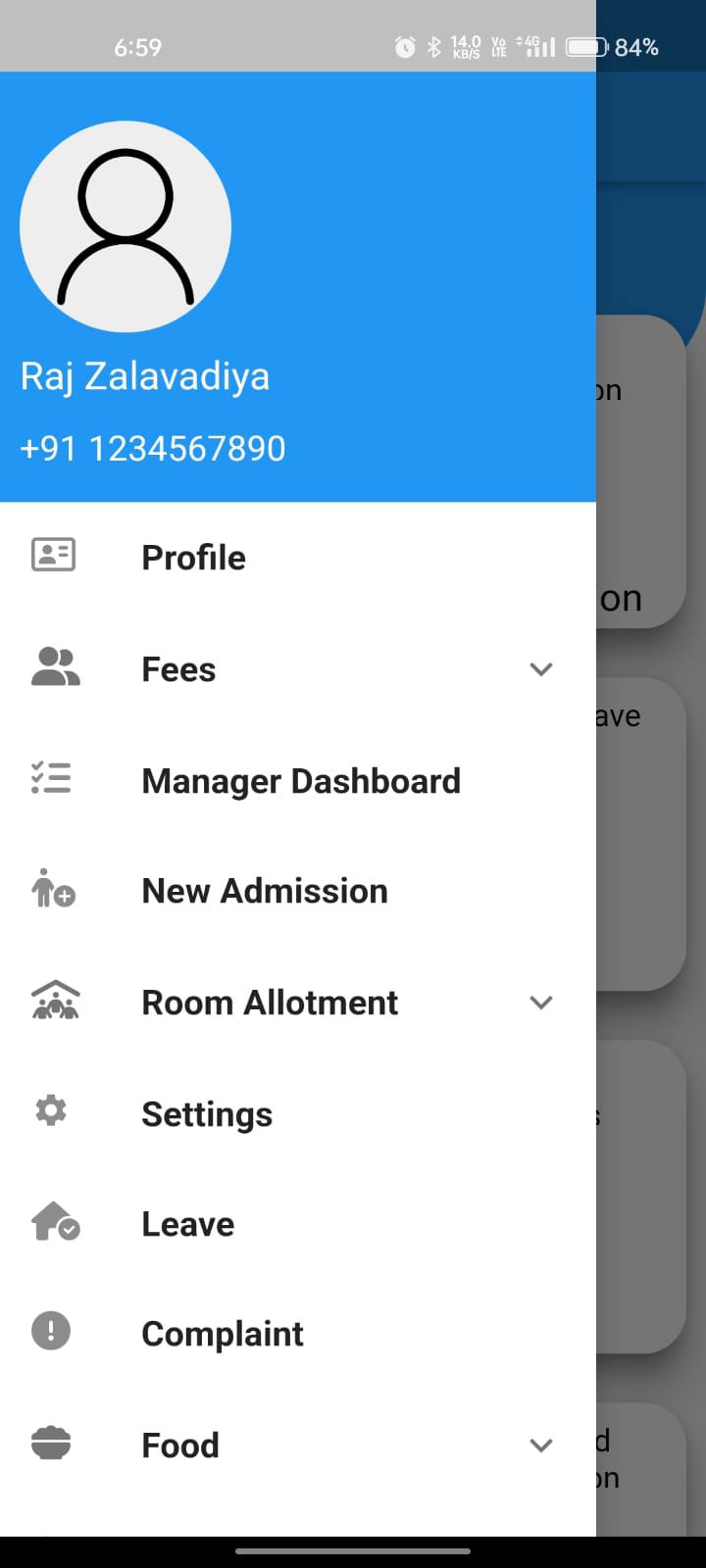
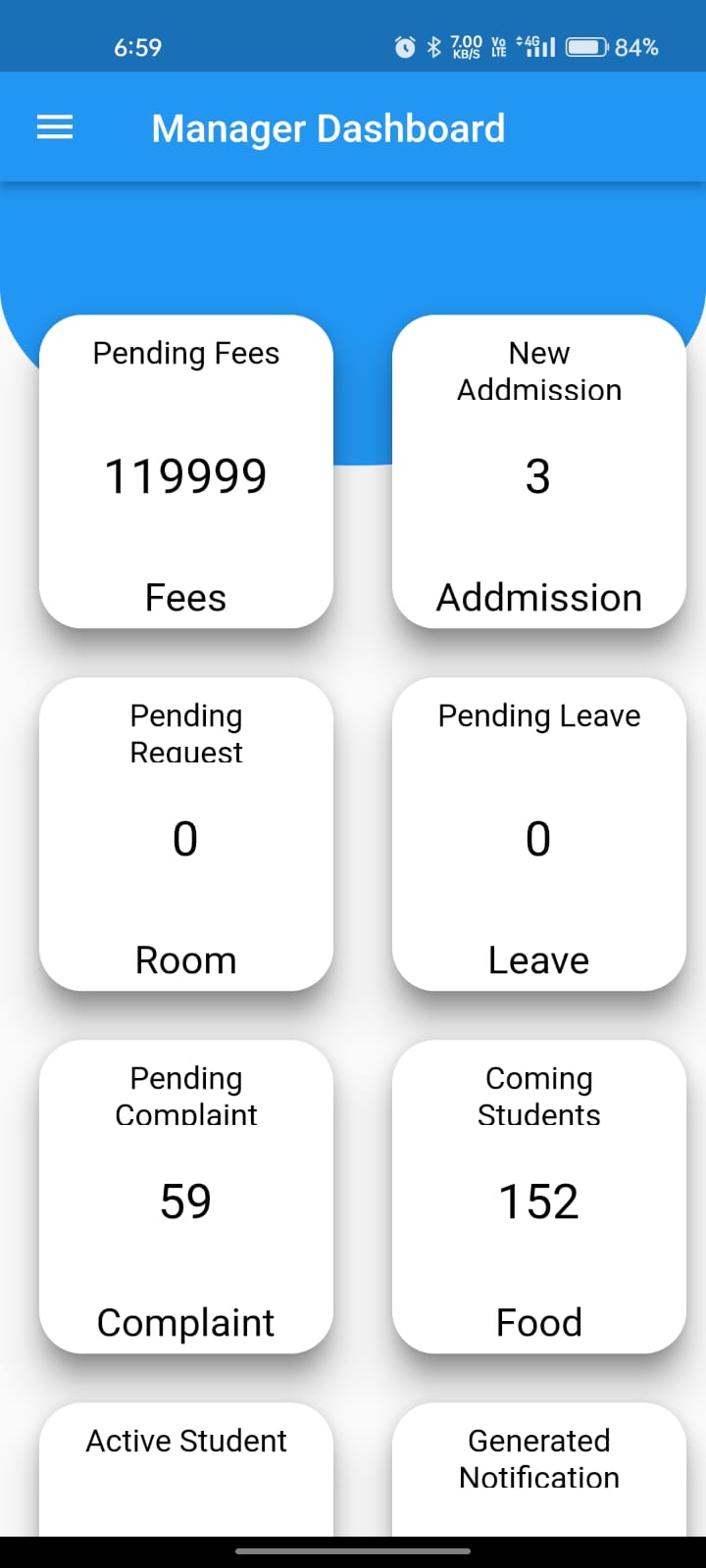
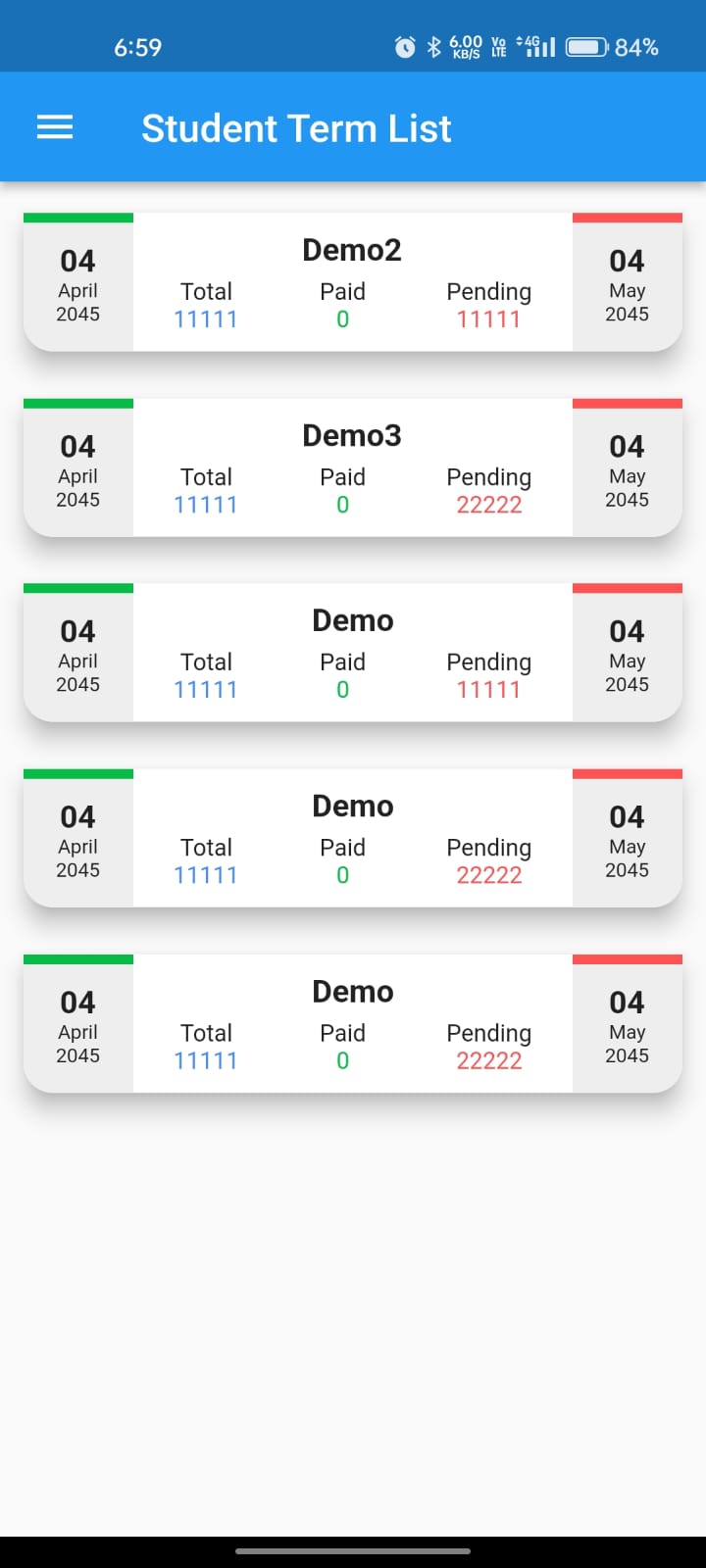
# ..

# CHAPTER - 7 SCREENSHOTS

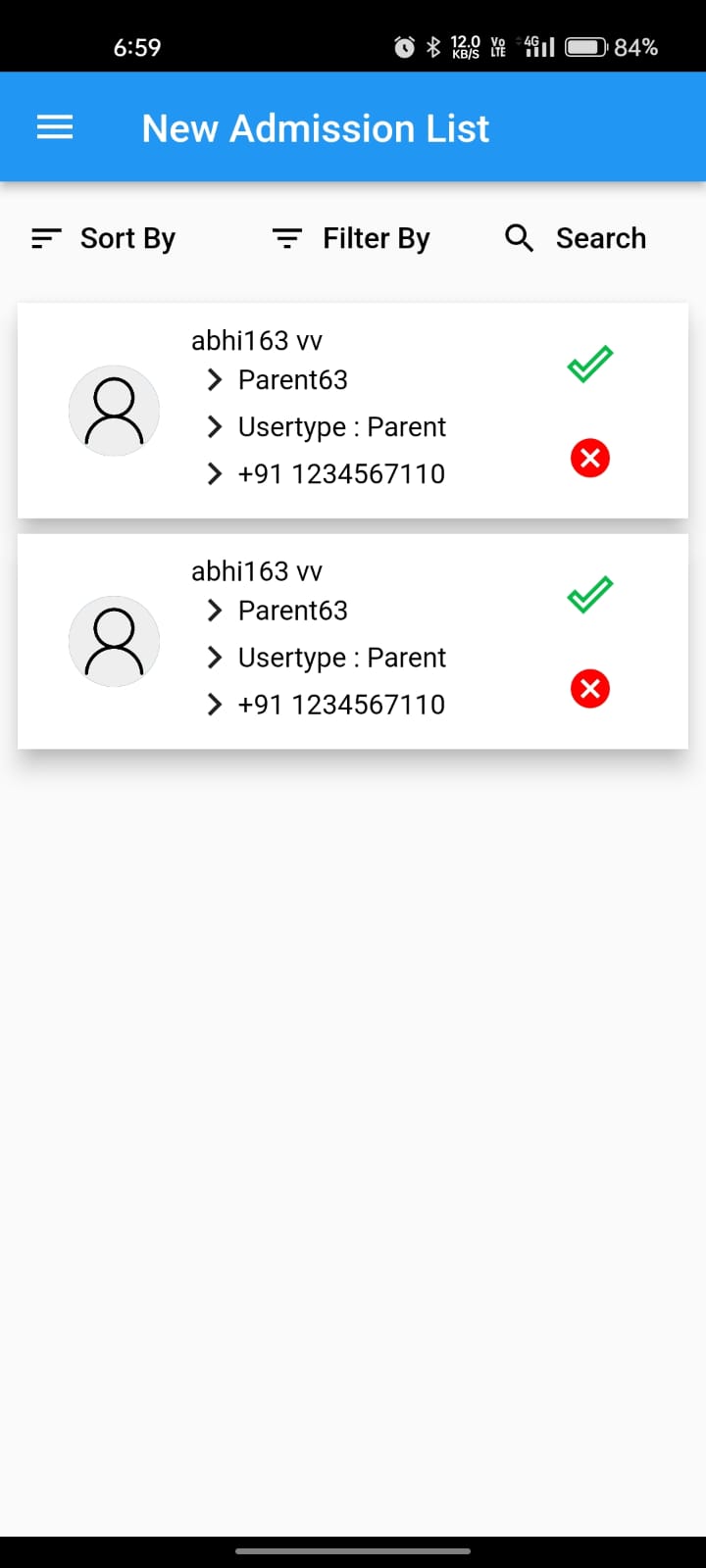
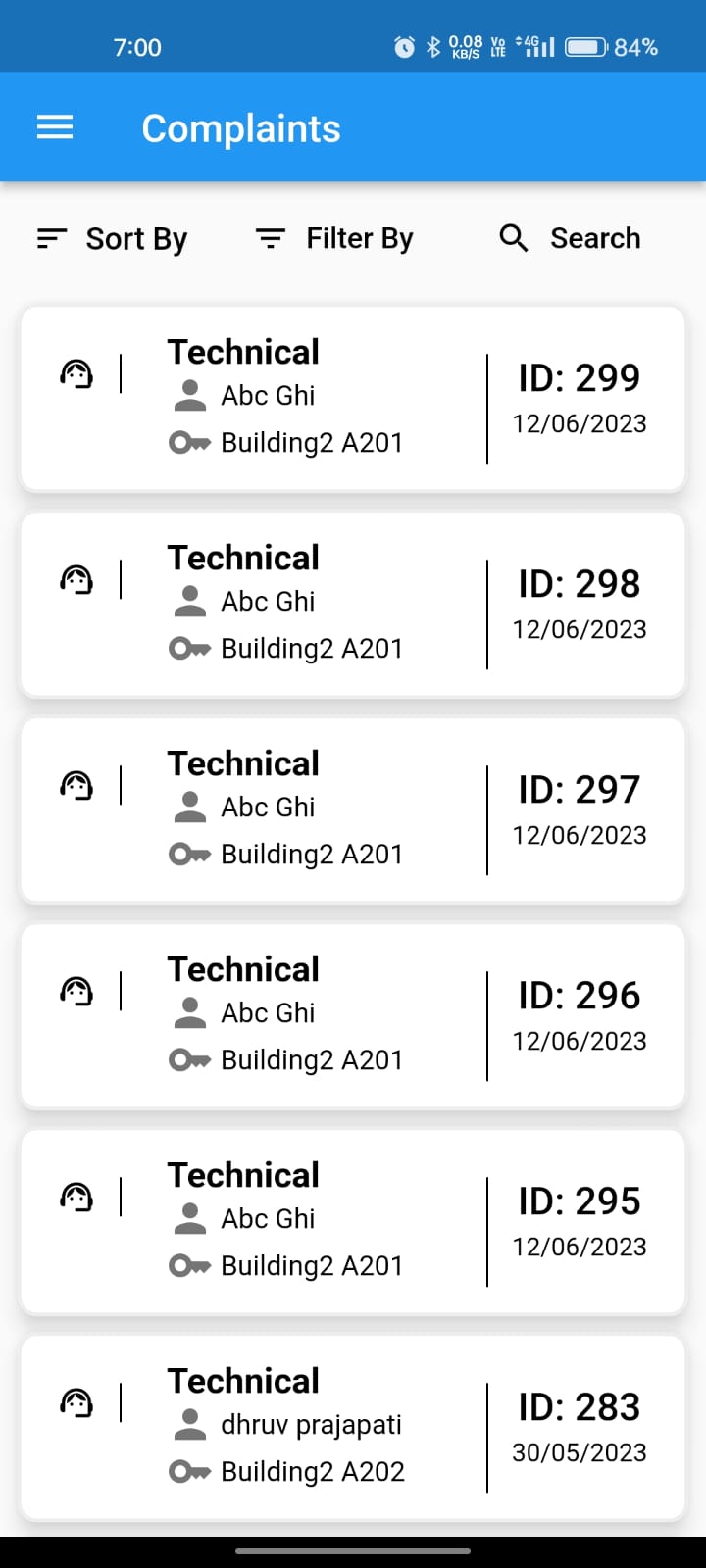
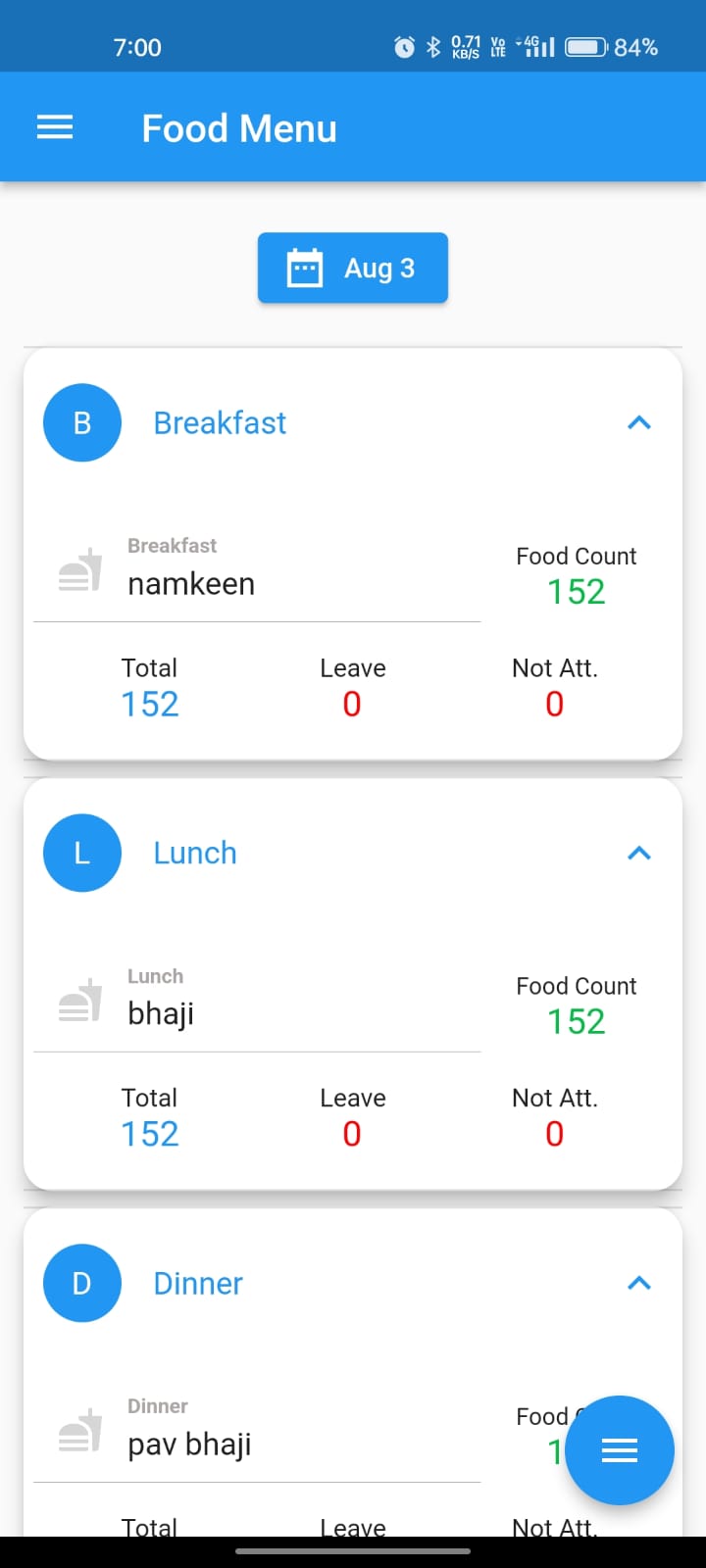
# 7.1 SCREENSHOTS:

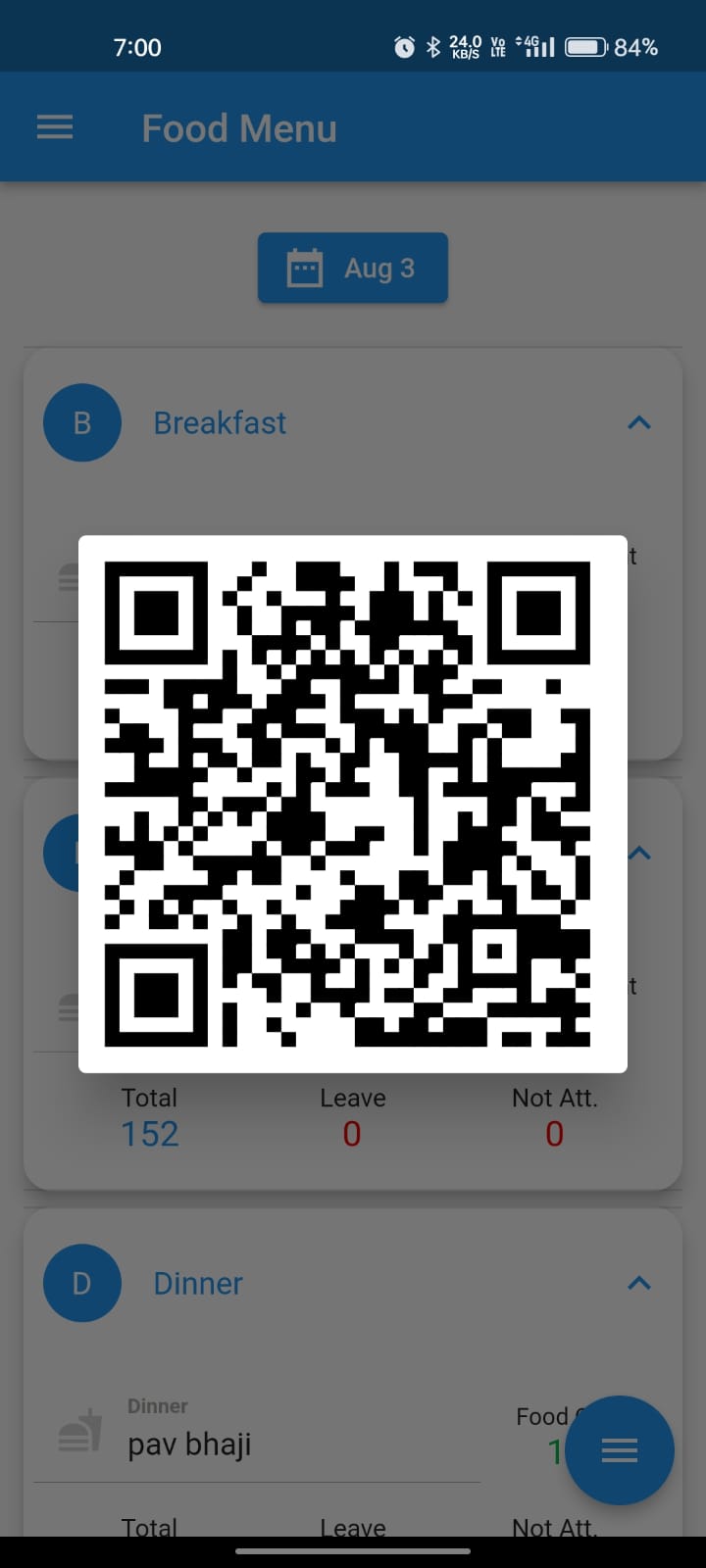
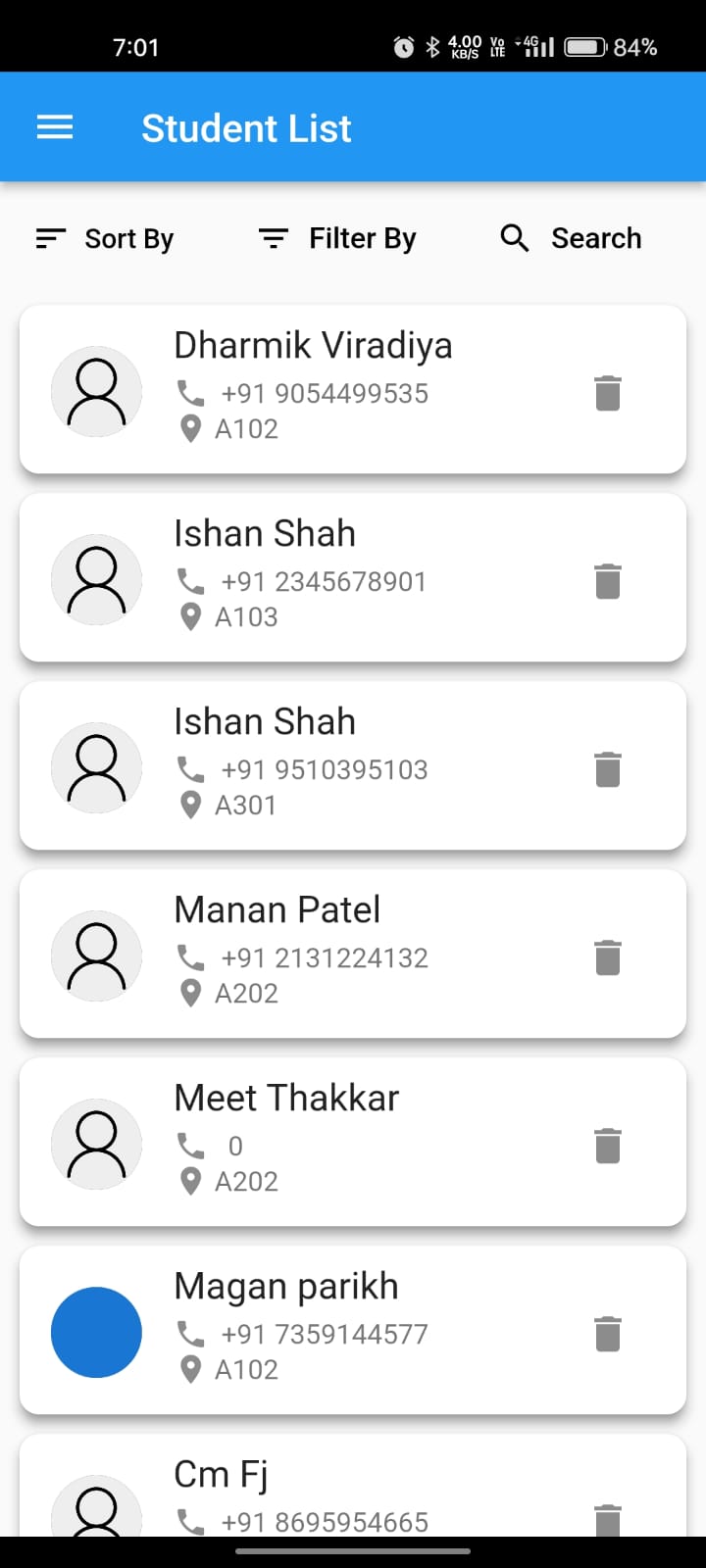
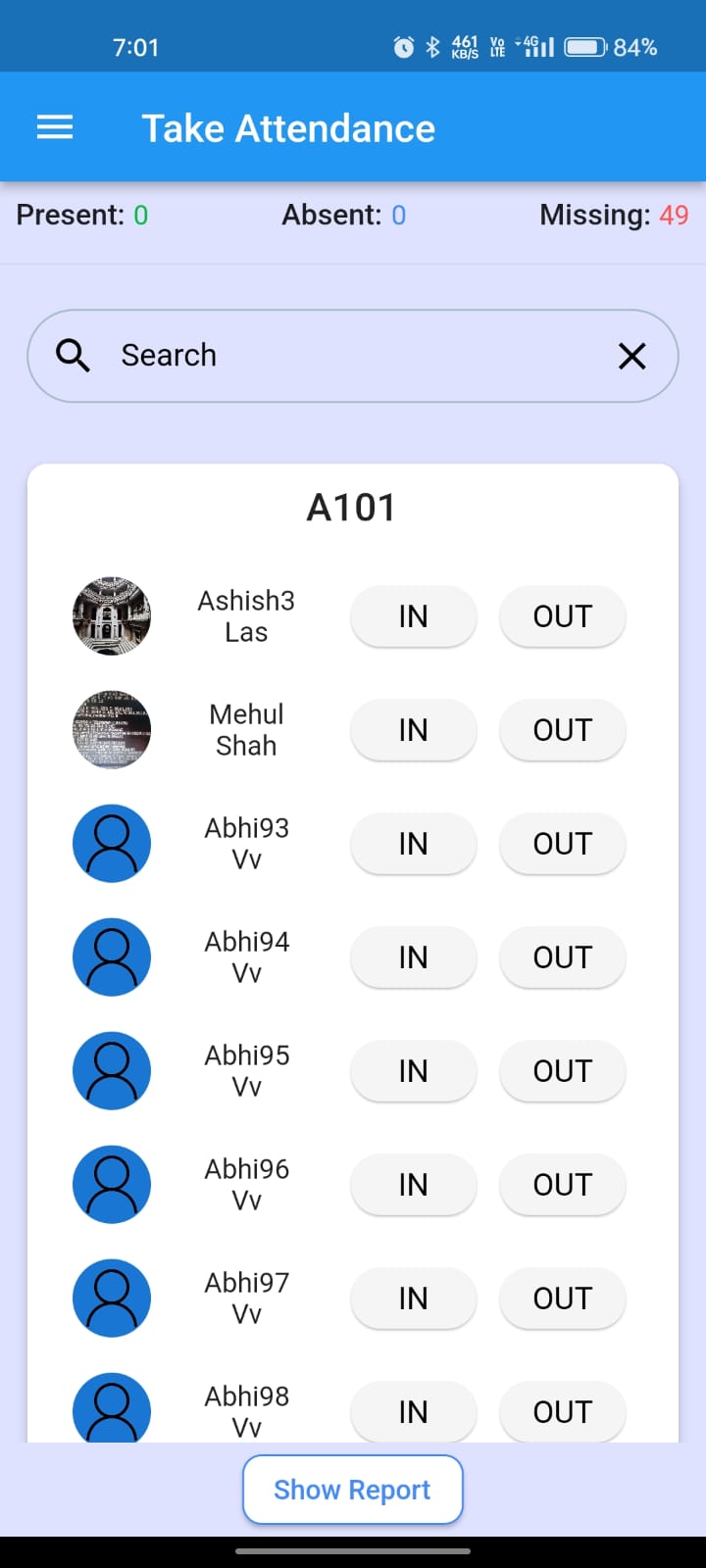
**Fig. 7.1.1 Splash Screen Fig. 7.1.2 Log In Screen Fig. 7.1.3 Sign up**

  ** **

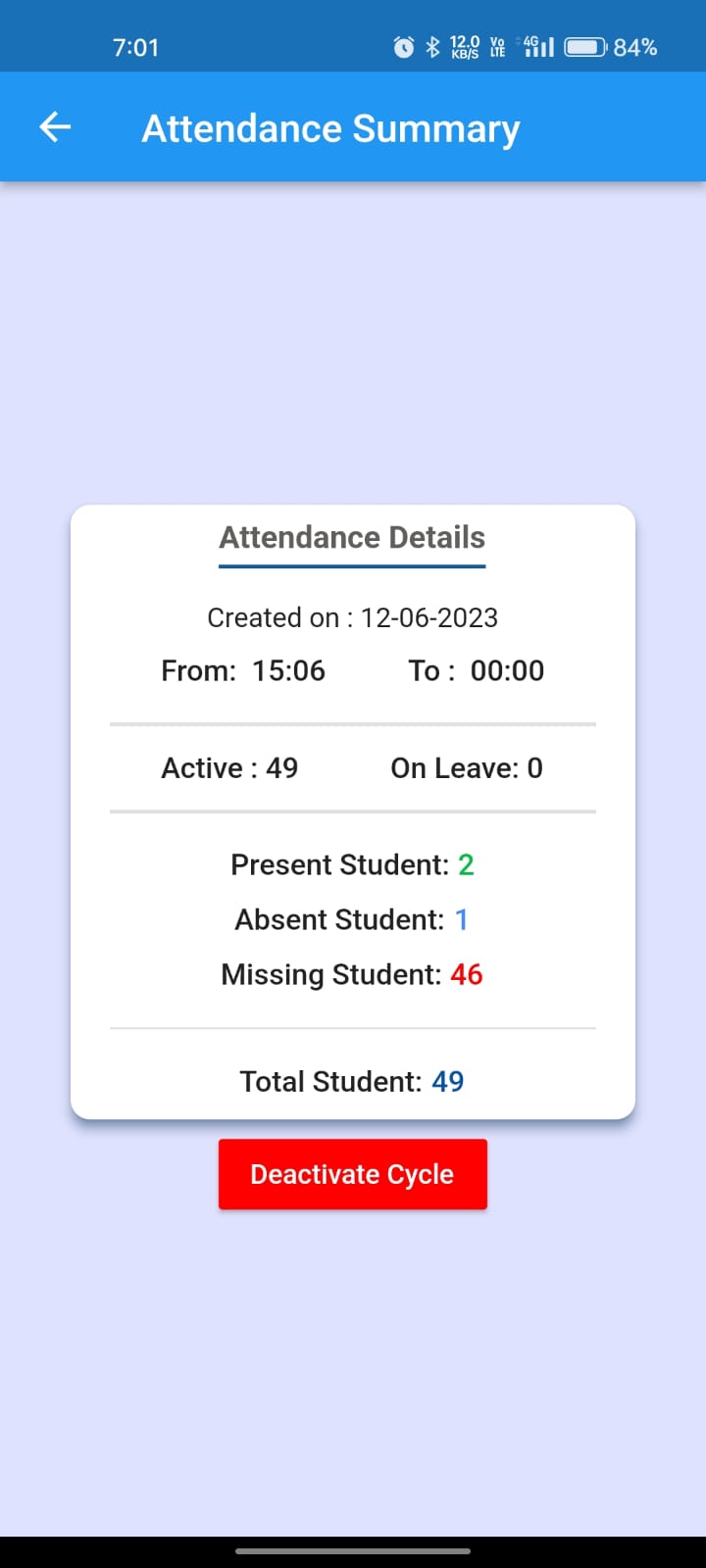
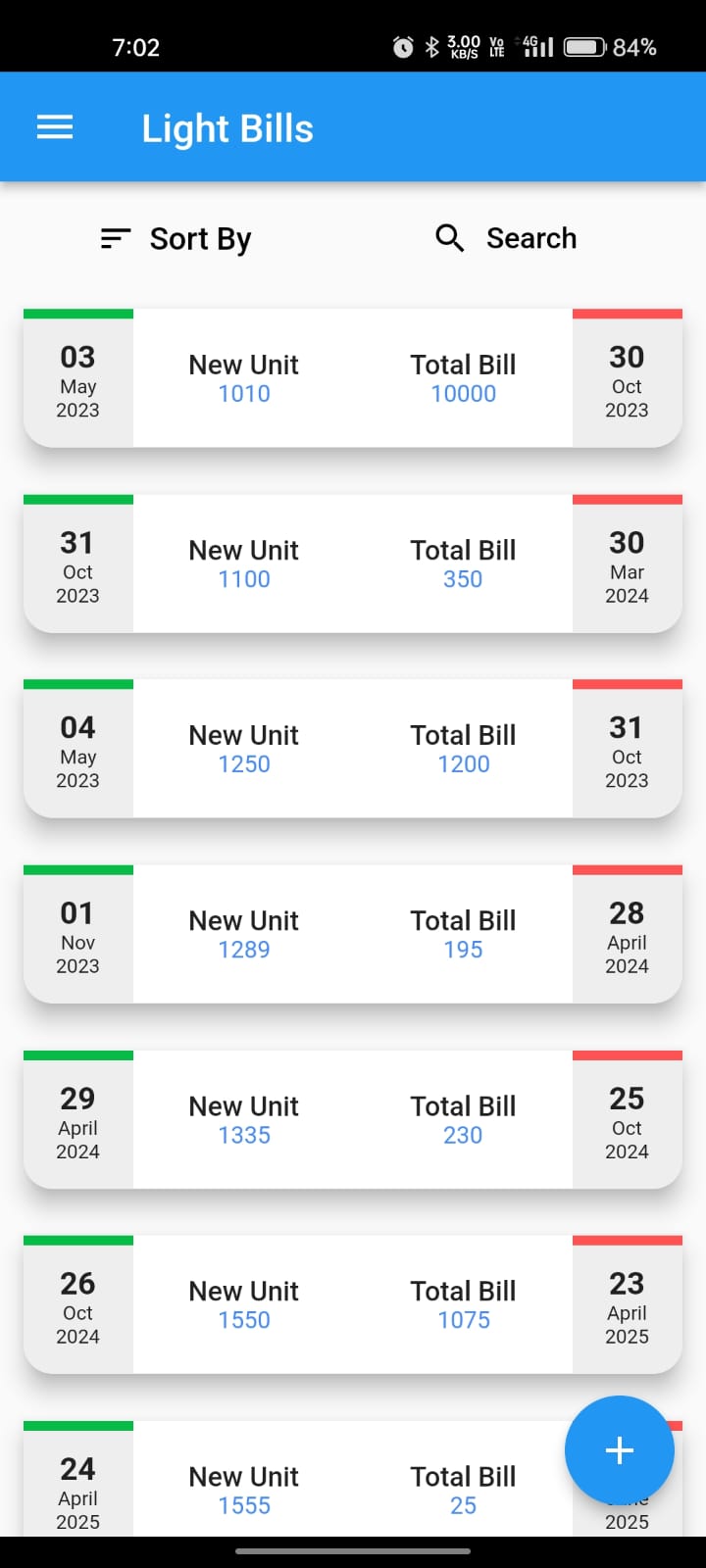
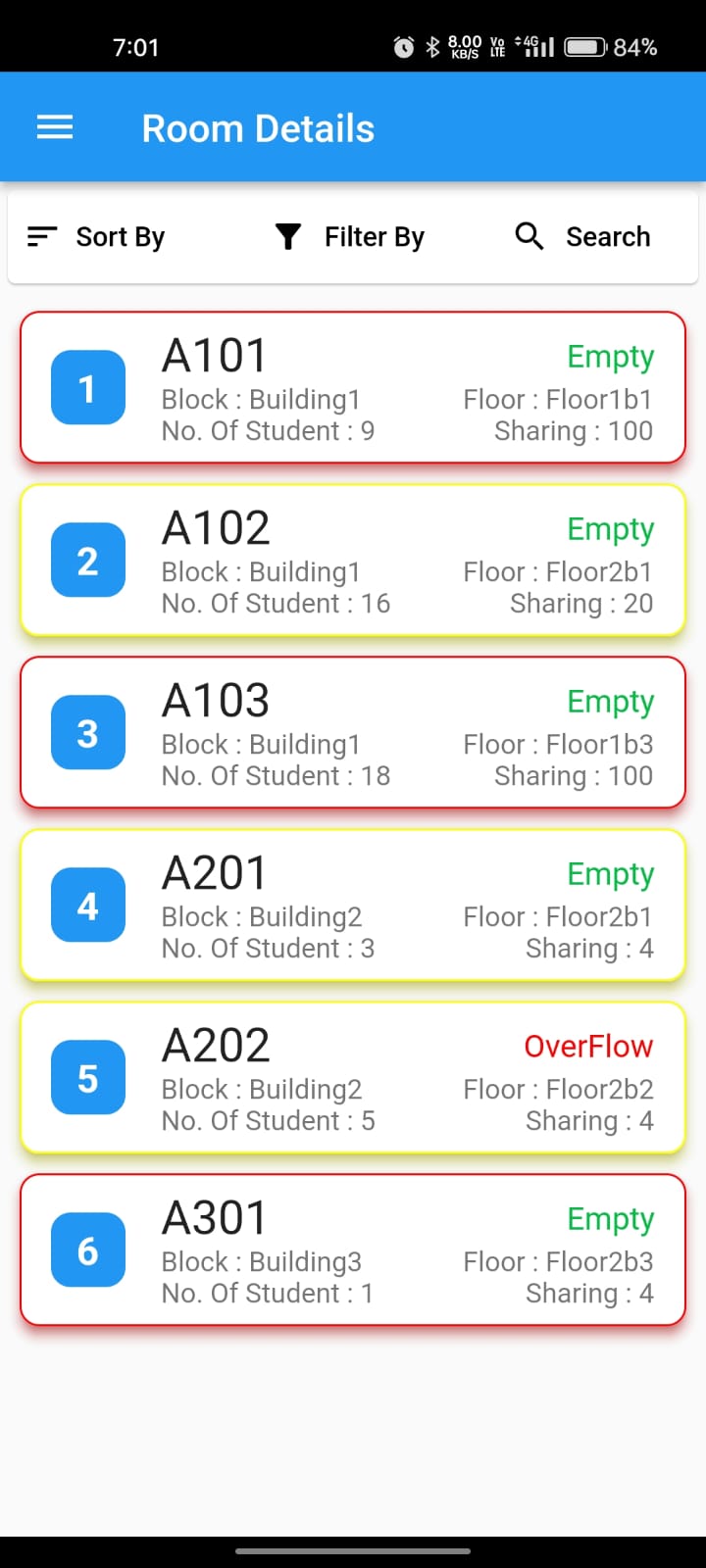
**Figure 7.1.4 Manager Drawer Fig. 7.1.5 Manager Dashboard Fig 7.1.6 Student Term List**

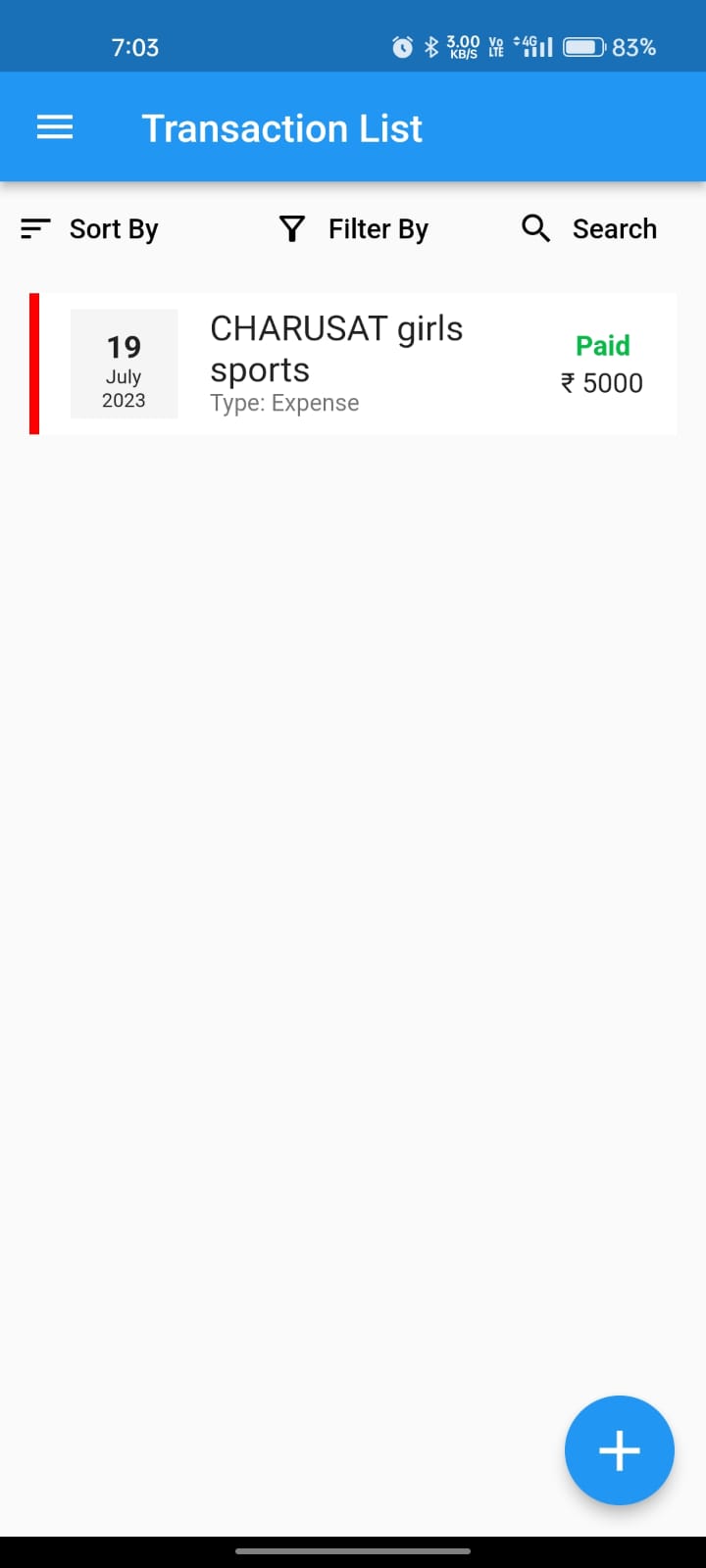
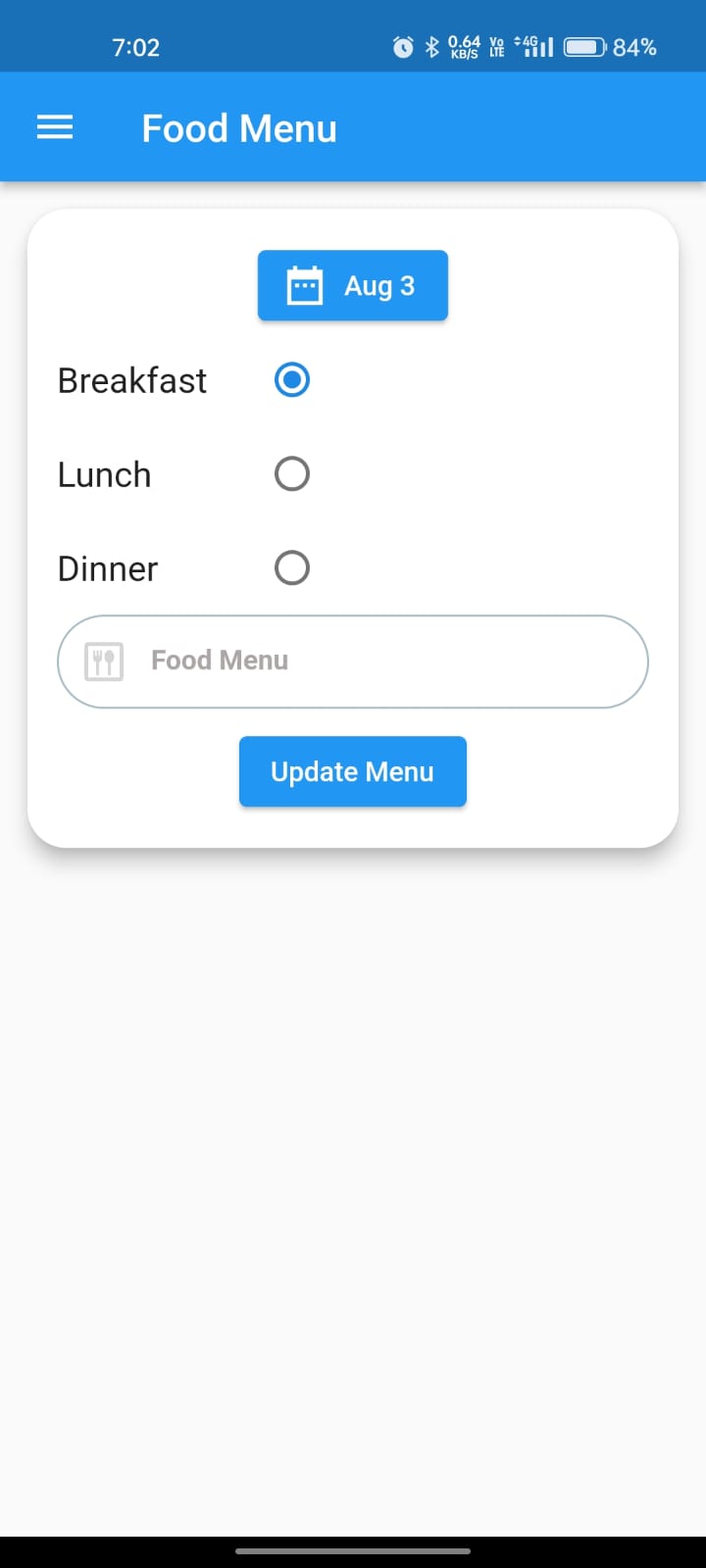
**Fig. 7.1.7 New Admission Fig. 7.1.8 Complaint list Fig. 7.1.9 Food Menu**

**  **

**Fig. 7.1.10 QR for Verification Fig. 7.1.11 Student List Fig. 7.1.12 Attendance List**

**Fig. 7.1.13 Analysis of Attendance Fig. 7.1.14. Light Bill Fig.7.1.15 Room Details**



**Fig. 7.1.16 Leave Section Fig 7.1.17 Food Update Fig.7.1.18 Expanse List**

­­­

# CHAPTER - 8 LIMITATIONS OF PROJECT

### 8.1 LIMITATIONS:

An appropriate limitation for the hostel management app could be the potential reliance on digital technology and internet connectivity. Since the app handles critical tasks such as leave approvals, attendance tracking, and food verification through QR codes, any disruption in internet connectivity or technical issues could temporarily hinder the system's functionality. This limitation may particularly affect users in areas with unreliable internet access or during network outages. To address this, the app could be designed to have offline capabilities and store essential data locally, ensuring some level of functionality even when internet connectivity is unavailable. Additionally, users should be made aware of the potential impact of internet dependencies and the need for backup measures to minimize disruptions in essential hostel operations.

### 8.2 CHALLENGES:

Complexity and Scope: Building a comprehensive app with multiple modules and functionalities can be complex. Ensuring a seamless integration of all features and managing interactions between different user roles can be challenging.

User Adoption: Convincing all stakeholders, including staff, students, parents, and other users, to adopt the app and make it a part of their routine can be difficult. Resistance to change or unfamiliarity with technology may hinder user acceptance.

Data Security and Privacy: Dealing with sensitive information such as student details, fees, and personal data requires robust data security measures. Ensuring data privacy and protecting against potential breaches is crucial.

User Experience (UX): Designing an intuitive and user-friendly interface is essential for the app's success. If the UX is confusing or not well-optimized, users may find it challenging to navigate and utilize the app effectively.

Maintenance and Updates: Regular maintenance and updates are necessary to keep the app running smoothly and to address any issues or bugs that may arise. Ensuring timely updates and addressing user feedback can be resource-intensive.

Integration with Existing Systems: Integrating the app with any existing systems or databases in the hostel can be a challenging task. Compatibility issues and data migration may need to be addressed.

Scalability: The app should be designed to accommodate potential growth and increasing user demands. Ensuring scalability to handle a growing number of hostels, users, and data is crucial.

Training and Support: Proper training and support for all users are necessary for them to fully utilize the app's functionalities. Providing ongoing support and addressing user queries and issues can be time-consuming.

Cost: Developing and maintaining the app can be costly. Budget constraints may impact the app's features or cause delays in implementation.

Regulatory Compliance: Depending on the location and jurisdiction, the app may need to comply with specific legal and regulatory requirements related to data protection and user privacy.

# CHAPTER - 9 OUTCOME

### OUTCOME:

1. Streamlined Operations: The app optimizes and streamlines various hostel operations, making tasks like admissions, room allocations, attendance tracking, and fee management more efficient and organized.

2. Improved Communication: The app facilitates seamless communication between administrators, staff, students, parents, and other users. Instant notifications and updates enhance communication and reduce delays in information dissemination.

3. Enhanced User Experience: A well-designed and user-friendly app improves the overall experience for all users. It reduces manual paperwork and simplifies complex processes, making it easier for users to perform their tasks.

4. Increased Transparency: The app promotes transparency in hostel management by providing real-time information on room allocations, leave approvals, fee status, and more. Users can easily access and track their relevant data.

5. Data Accuracy and Security: The app centralizes data, reducing the chances of errors and discrepancies. Robust security measures protect sensitive information, ensuring data privacy and integrity.

6. Efficient Resource Management: The app helps optimize resource allocation and utilization. From managing food menus to handling expenses, the app assists in effectively managing resources.

7. Better Parental Involvement: Parents can stay informed about their child's hostel activities, academic progress, fees, and attendance, leading to increased parental involvement and support.

8. Effective Complaint Handling: With a structured complaint management system, issues can be addressed promptly, leading to improved student satisfaction and a positive hostel environment.

9. Increased Productivity: By automating repetitive tasks, the app frees up staff and administrators to focus on more critical and strategic aspects of hostel management, thereby increasing overall productivity.

10. Scalability and Growth: A successfully implemented app can be easily scaled to accommodate additional hostels or more users. As the system expands, it can efficiently handle the increasing workload.

11. Data Analytics and Insights: The app can generate valuable data analytics and insights, enabling administrators to make data-driven decisions for hostel improvements and optimizations.

# CHAPTER - 10 REFERENCES

### 10.1 REFERENCES:

* + 1. <https://docs.flutter.dev/>
    2. <https://www.tutorialspoint.com/dart_programming/index.htm>
    3. <https://youtube.com/playlist?list=PLEFAsRdcqVXfzxJofnP7AG8MQ4AO8b_OR>
    4. <https://www.youtube.com/watch?v=OGe8TwGKabs&list=PL8kbUJtS6hyal7Uw7wTeYmv7yiNPH5kOq&ab_channel=HarshH.Rajpurohit>
    5. <https://console.firebase.google.com/project/we-chat-19cdc/firestore/data/~2Fusers~2F1hKygA5TZ0Uc6B4bBYG1Zptw1vK2>