## IMG_256IMG_256

## PROJECT ON

## LEAVE MANAGEMENT SYSTEM

**(ANDROID APPLICATION)**

In Partial Fulfilment For the award

Degree of

**Bachelors of Science**

In

**Computer Science**

**Bhairab Ganguly College**

Submitted by

# Jyoti Sharma

# Poulastaa Das

# Under the supervisor of

**Prof. (Dr.) Subhranil Som** Department of Computer Science,

Principal of Bhairab Ganguly College

# AUGUST 2024

# CERTIFICATE

This is to certify that the project entitled "LEAVE MANAGEMENT SYSTEM" has been carried out by Poulastaa Das and Jyoti Sharma having roll number 6241129 20821 and 6242129 20819 respectively, under my supervision in partially fulfilment of the requirements for the Degree of Bachelor Of Science in Computer Science of the Bhairab Ganguly College under West Bengal State University during the academic year 2024.

------------------------------------------------

SUPERVISOR SIGNATURE

Prof. (Dr.) Subhranil Som

SUPERVISOR

DEPARTMENT OF COMPUTER SCIENCE

BHAIRAB GANGULY COLLEGE

------------------------------------------

HOD SIGNATURE

Dr. Pratima Biswas

HEAD OF THE DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCE

BHAIRAB GANGULY COLLEGE

## BONAFIDE CERTIFICATE

Certified that this project report “Leave Management System” is the bonafide work of “Jyoti Sharma” and “Poulastaa Das” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here in docs not part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

-------------------------------- -----------------------------

SUPERVISOR SIGNATURE HOD SIGNATURE

**Prof. (Dr.) Subhranil Som Dr. Pratima Biswas**

SUPERVISOR HEAD OF THE DEPARTMENT

PRINCIPAL OF

OF COMPUTER SCIENC

BHAIRAB GANGULY COLLEGE BHAIRAB GANGULY COLLEGE

DEPARTMENT OF COMPUTER SCIENCE

**Submitted for the 6th Semester Project Held On \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.**

**INTERNAL EXAMINAR EXTERNAL EXAMINAR**

## ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude towards all the people who have in various way, helped in the successful completion of my project.

I must convey my gratitude to our Principal Dr. Subhranil Som for giving me the constant source of inspiration and help me in preparing the project, personally correcting my work and providing encouragement throughout the project.

I also thank all my faculty members for steering me through the tough as well as easy phases of the project in a result-oriented manner with concern attention.

**------------------------------- ------------------------------**

DECLARATION

We, the undersigned, solemnly declare that the project report on “Leave Management System” is based on our own work carried out during the course of our study under the supervision of **Dr.** **Subhranil Som**. I assert that the statements made and conclusions drawn are an outcome of our research work. Furthermore, we certify that the work contained in the report is original and has been conducted by us under the general supervision of our supervisor. The work has not been submitted to any other institute for any other degree, diploma or certificate in this university or any other university in India and aboard. We have adhered to the guidelines provided by the university in writing the report. Whenever we have utilized materials (data, theoretical analysis and text) from other resources, we have appropriately credited them in the text of the report and provided their details in the references.

NAME: JYOTI SHARMA

ROLL NO: 9163

REG. NO: 1072121400012

BHAIRAB GANGULY COLLEGE

WEST BENGAL STATE UNIVERSITY SIGNATURE

NAME: POULASTAA DAS

ROLL NO: 9039

REG. NO: 1072111100006

BHAIRAB GANGULY COLLEGE SIGNATURE

WEST BENGAL STATE UNIVERSITY

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| CHAPTER NO. |  | TITLE | PAGE NO |
|  |  | ABSTRACT |  |
| 1 |  | Introduction |  |
|  | 1.1 | PROJECT AIMS AND OBJECTIVE |  |
| 2 |  | System Analysis |  |
|  | 2.1 | PROJECT AIMS AND OBJECTIVES |  |
|  | 2.1.1 | Project title |  |
|  | 2.1.2 | category |  |
|  | 2.1.3 | purpose |  |
|  | 2.1.4 | scope |  |
|  | 2.1.5 | Existing system |  |
|  | 2.1.6 | Proposed system |  |
|  | 2.1.7 | Roles of admin in the lms |  |
|  | 2.1.8 | Functional requirements for admin |  |
|  | 2.1.9 | Advantage of employee |  |
|  | 2.1.10 | Functional requirements for employee |  |
|  | 2.1.11 | Non Functional requirements for employe |  |
|  | 2.2 | Software tools |  |
|  | 2.3 | Database |  |
|  | 2.4 | Deployment |  |
|  | 2.5 | Hardware specification |  |
|  | 2.6 | Frontened technology |  |
|  | 2.7 | Backend technology |  |
| 3 |  | Iterative model |  |
|  | 3.1 | FEASIBILITY STUDY |  |
|  | 3.1.1 | TECHNICAL FEASIBILITY |  |
|  | 3.1.2 | OPERATIONAL FEASIBILIT |  |
|  | 3.1.3 | FINANCIAL FEASIBILITY |  |
|  | 3.2 | REQUIREMENT ANALYSIS |  |
|  | 3.3 | REQUIREMENT SPECIFICATIONS |  |
|  | 3.4 | SYSTEM DESIGN |  |
|  | 3.5 | TESTING |  |
|  | 3.6 | SCREENSHOTS |  |
| 4 |  | CONCLUSION |  |
| 5 |  | FUTURE SCOPE |  |
| 6 |  | REFERENCES |  |

ABSTRACT

In today's fast-paced organizational environment, efficient and effective leave management is critical for maintaining productivity and ensuring smooth operations. Traditional methods of leave management, often relying on manual processes and paper-based systems, are prone to errors, time-consuming, and lack transparency. To address these challenges, we propose the development of a Leave Management System Application.

Key Points:

Automation

User-Friendly Interface

Employee Features

Customization

Easy Management

Data Integrity and Security

Scalability

By leveraging these features, the Leave Management System Application represents a significant advancement in human resource management practices, contributing to improved productivity and operational excellence in College.

## CHAPTER 1

## INTRODUCTION

THIS CHAPTER GIVES US A BASIC INTRODUCTION AN OVERVIEW OF THE AIM AND OBJECTIVES OF THE PROJECT

A Leave Management System (LMS) is an essential tool for modern organizations to efficiently manage employee leave requests and ensure smooth business operations. This system automates the process of leave application, approval, tracking, and reporting, providing a seamless and transparent workflow for both employees and management. With an LMS, employees can easily submit their leave requests online, and managers can approve or reject these requests with just a few clicks. The system keeps track of leave balances, generates detailed reports, and ensures compliance with college policies. By implementing a Leave Management System, colleges can enhance productivity, reduce administrative burden, and improve overall employee satisfaction.

**1.1 PROJECT AIMS AND OBJECTIVES**

The Aims and Objective of the project that will be achieved after the completion of the project are discussed below-

1. User-Friendly Interface
2. Automate Leave Requests and Approvals
3. Leave Balance Tracking
4. Mobile Accessibility
5. Employee Management
6. Data Security and Privacy

## CHAPTER 2

# SYSTEM ANALYSIS

**2.1 PROJECT AIMS AND OBJECTIVES**

**2.1.1 PROJECT TITLE**

LEAVE MANAGEMENT SYSYTEM

**2.1.2 CATEGORY**

ANDRIOD APPLICATION

**2.1.3 PURPOSE**

The purpose of a leave management system is to facilitate the efficient and transparent handling of employee leave requests and approvals. It helps manage and track leave requests, approvals, and balances efficiently. It also improves communication between employees and management about leave status, and provides valuable data and insights for better decision-making and workforce planning.

**2.1.4 SCOPE**

The scope of a leave management system involves automating and managing the entire process of handling leave requests within an organization. It includes facilitating the application and approval of various types of leave, tracking leave balances, and ensuring compliance with organizational policies and legal requirements.

**2.1.5 EXISTING SYSTEM**

Existing leave management system typically includes several key components and functionalities to handle employee leave requests efficiently.

1. Web App
2. Leave types
3. Leave Request Submission
4. Approved Leave
5. Leave balance management
6. Notification and alert
7. Calendar Integration

**2.1.6 PROPOSED SYSTEM**

When proposing an enhanced leave management system, the goal is to improve upon the existing processes, leveraging technology to increase efficiency, transparency, and user satisfaction.

1. Mobile Accessibility
2. Better And Easy UI
3. Adding New Leave Types
4. Beater Management
5. Advance Notification and alert
6. Better Data Management
7. Improve Security

**2.1.7 ROLES OF ADMIN IN THE LEAVE MANAGEMENT SYSTEM**

1. Approve/reject leave: Admin can approve and reject leave of Teacher, Clark.
2. View leave: Admin can view all the employees leave.
3. Add new employee: Admin can add a new employee.
4. Remove an employee: Admin can remove any employee.
5. Edit leave balance: Admin can edit employee leave option.
6. Edit Department In-charge: Admin can change department In-charge.
7. View and download report: Admin can view employee leave and download the report.

**2.1.8 FUNCTIONAL REQUIREMENT FOR ADMIN**

### **1. User Management**

* **Add/Remove Employee**: Ability to add or remove faculty, and other staff members from the system.
* **Assign Roles**: Ability to assign roles (e.g., Departmental In-charge) and permissions.

### **2. Leave Management**

* **View Leave Requests**: Ability to view all leave requests submitted by faculty.
* **Approve/Reject Leave Requests**: Ability to approve or reject leave requests.
* **Track Leave Status**: Ability to monitor the status of leave requests (pending, approved, rejected).

### **3.** **Leave Policy Management**

* **Configure Leave Policies**: Ability to define and manage leave policies (e.g., number of leaves allowed, types of leave).
* **Update Leave Policies**: Ability to update leave policies as per college rules or regulations.

### **4. Reporting and Analytics**

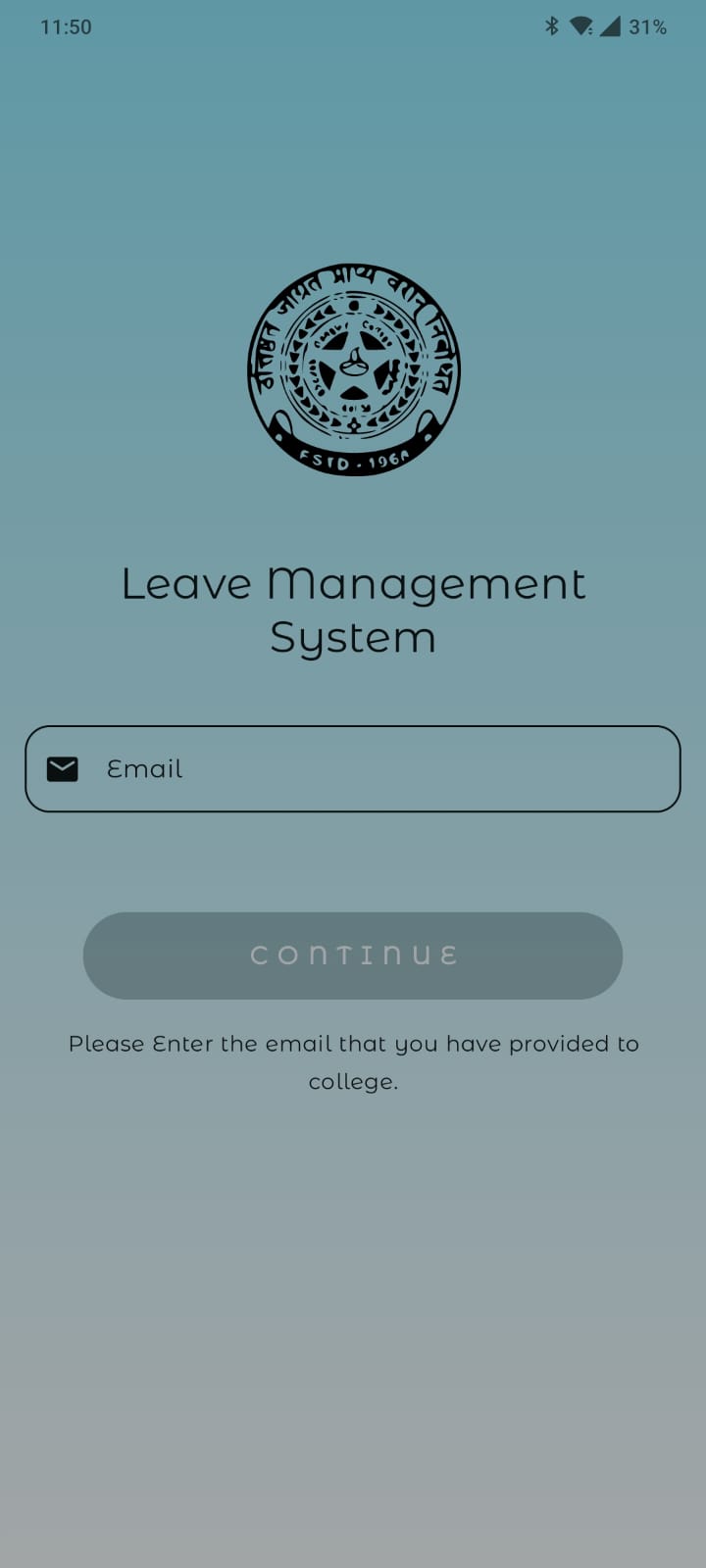
* **Generate Reports**: Ability to generate reports on leave usage, pending requests, approved requests, etc.
* **Download Reports:** Ability to download generated reports on leave usage, pending requests, approved requests, etc. (Example: PDF)

### **5. Security**

* **Access Control**: Implement role-based access control to ensure that only authorized personnel can access or modify sensitive information.

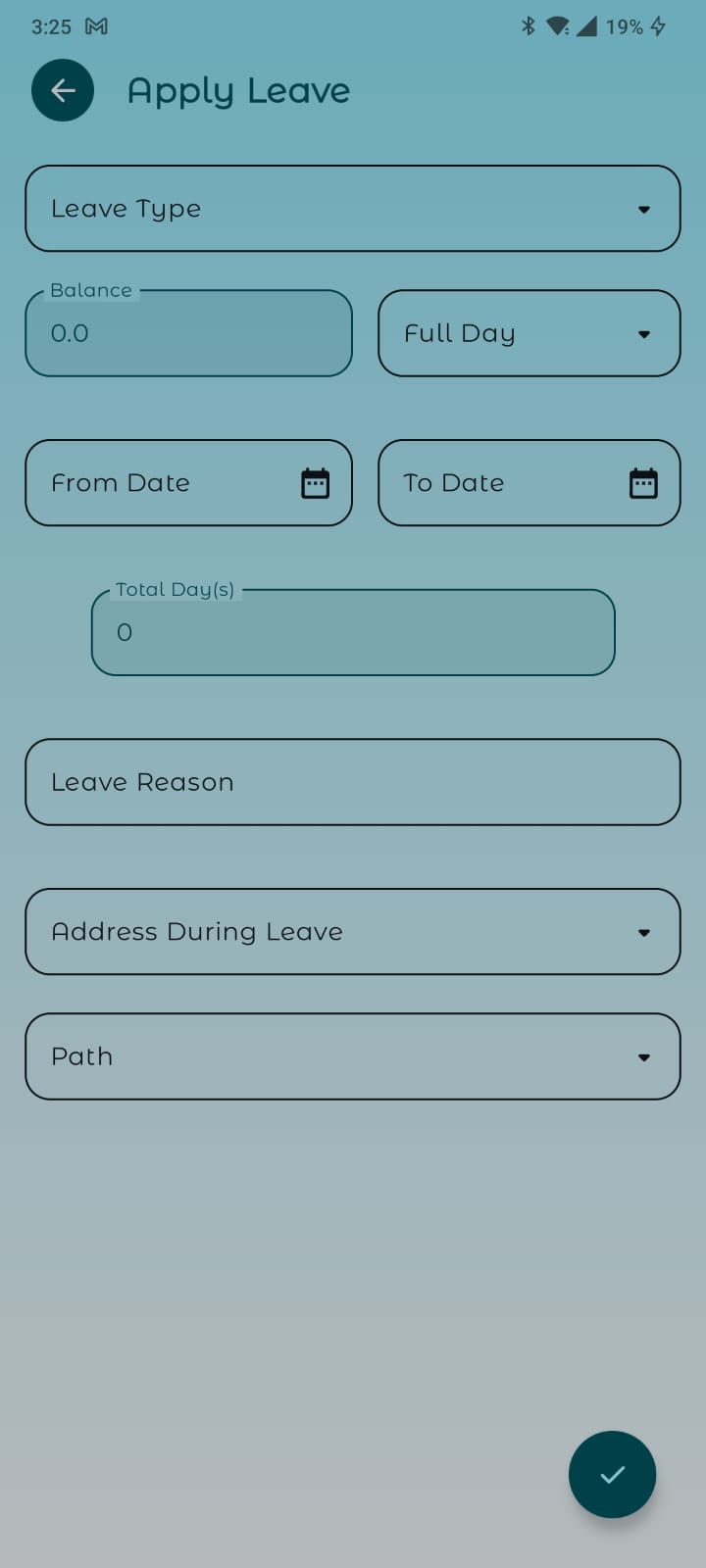
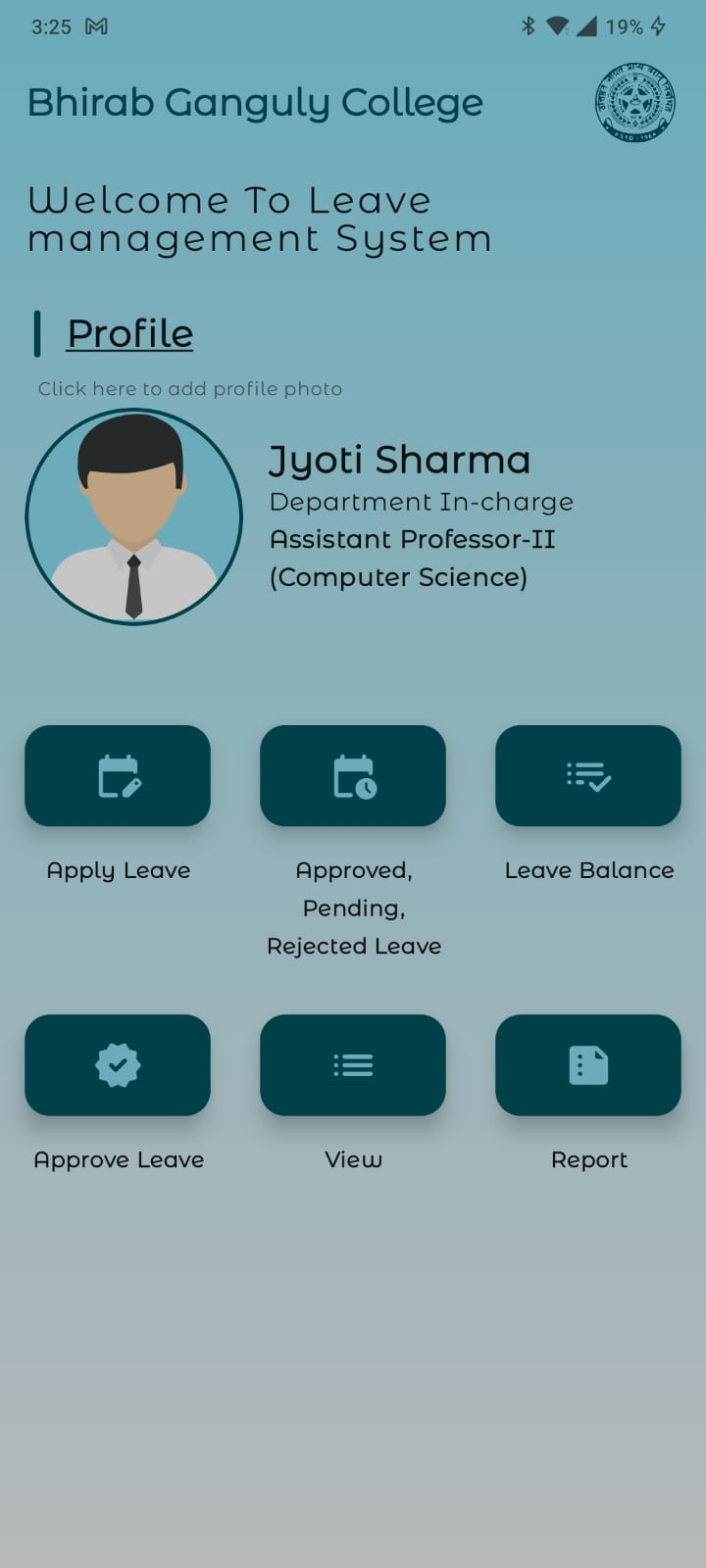
**Data Privacy**: Ensure compliance with data privacy regulations and policies.

**Admin Login**

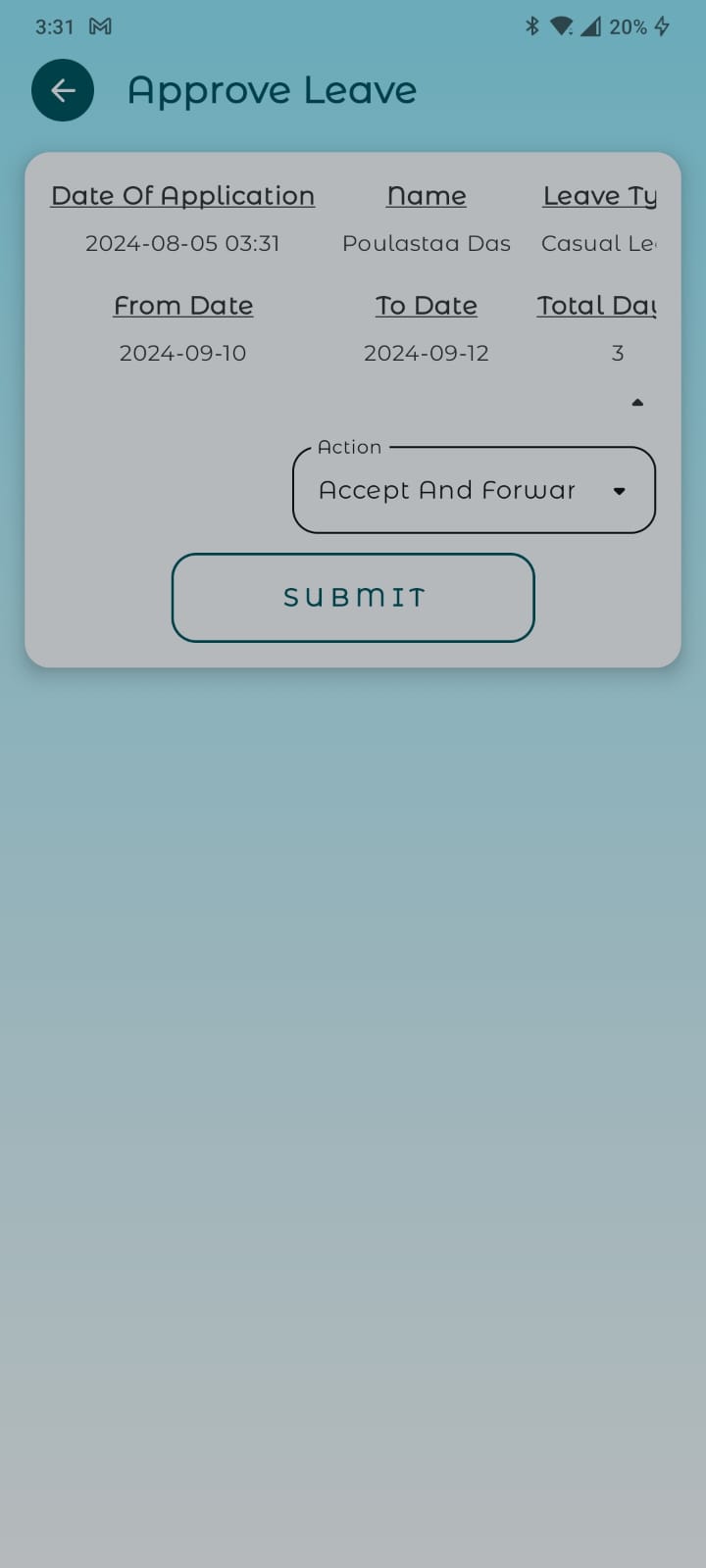
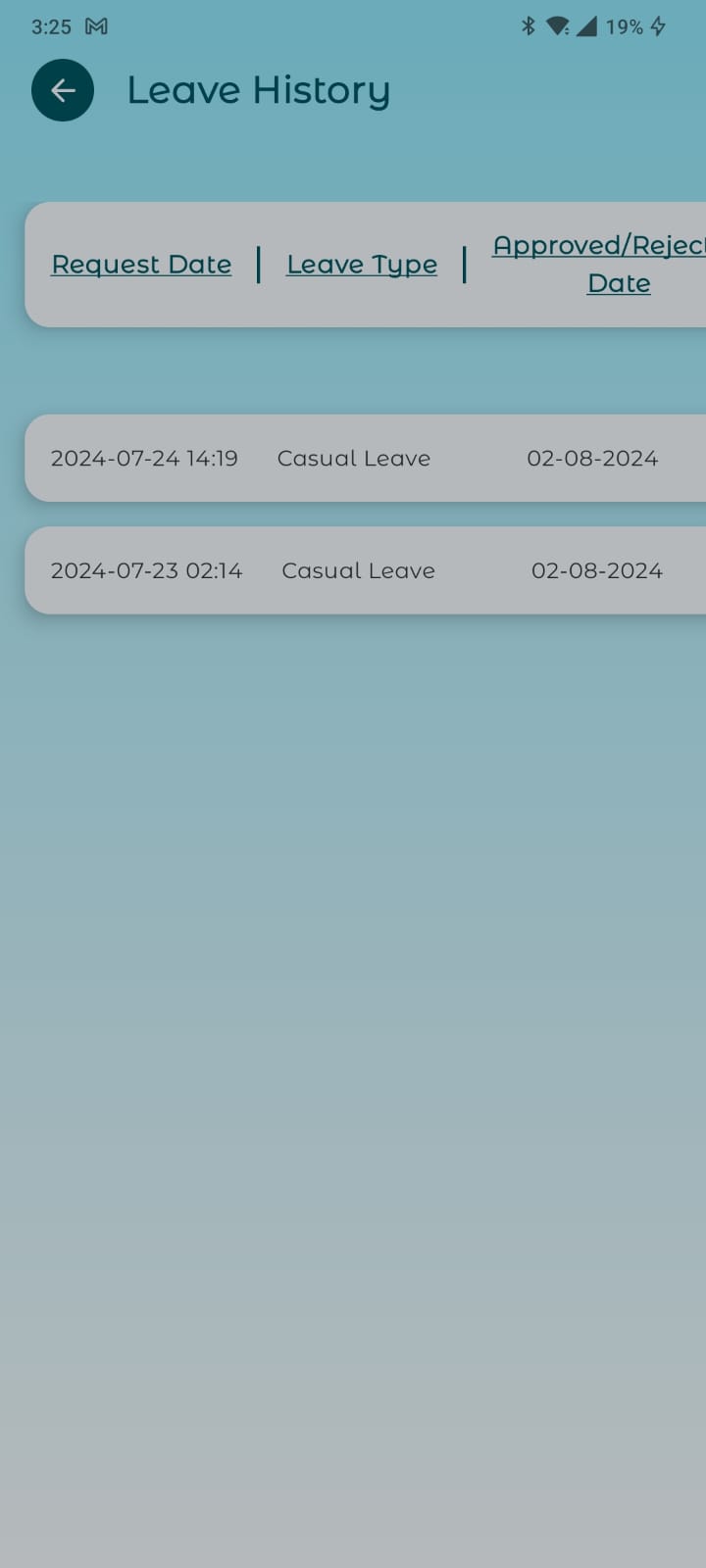


**HOD:**

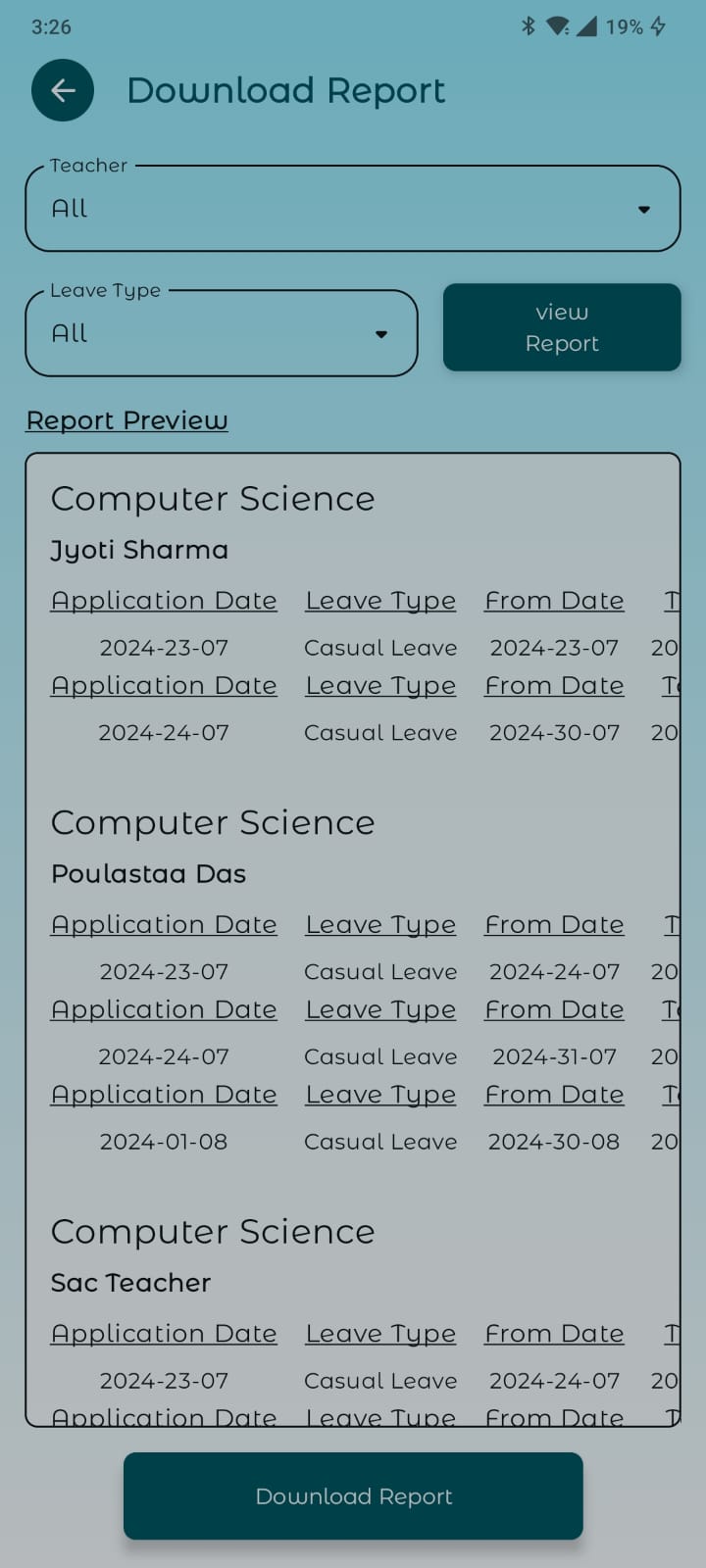
**1. Dashboard 2. Apply leave**



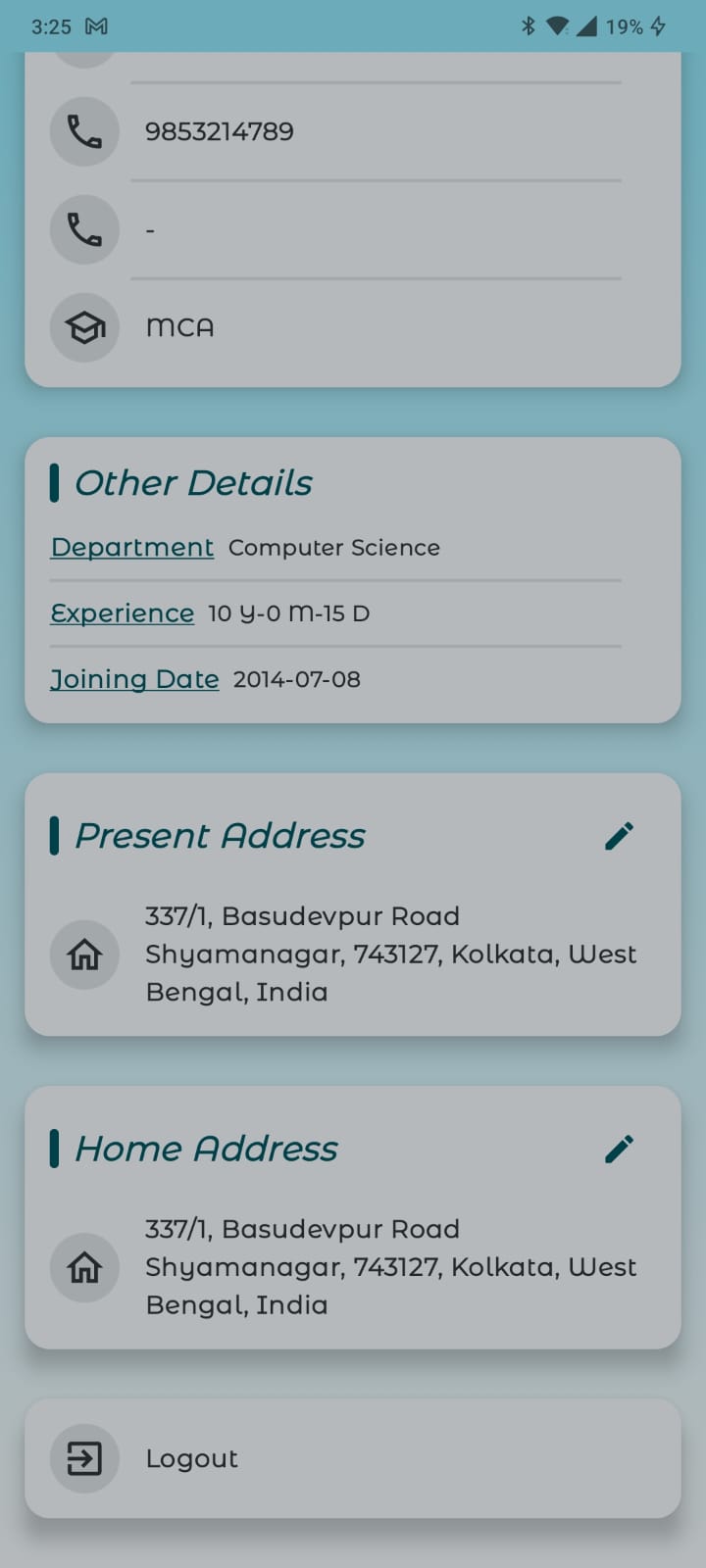
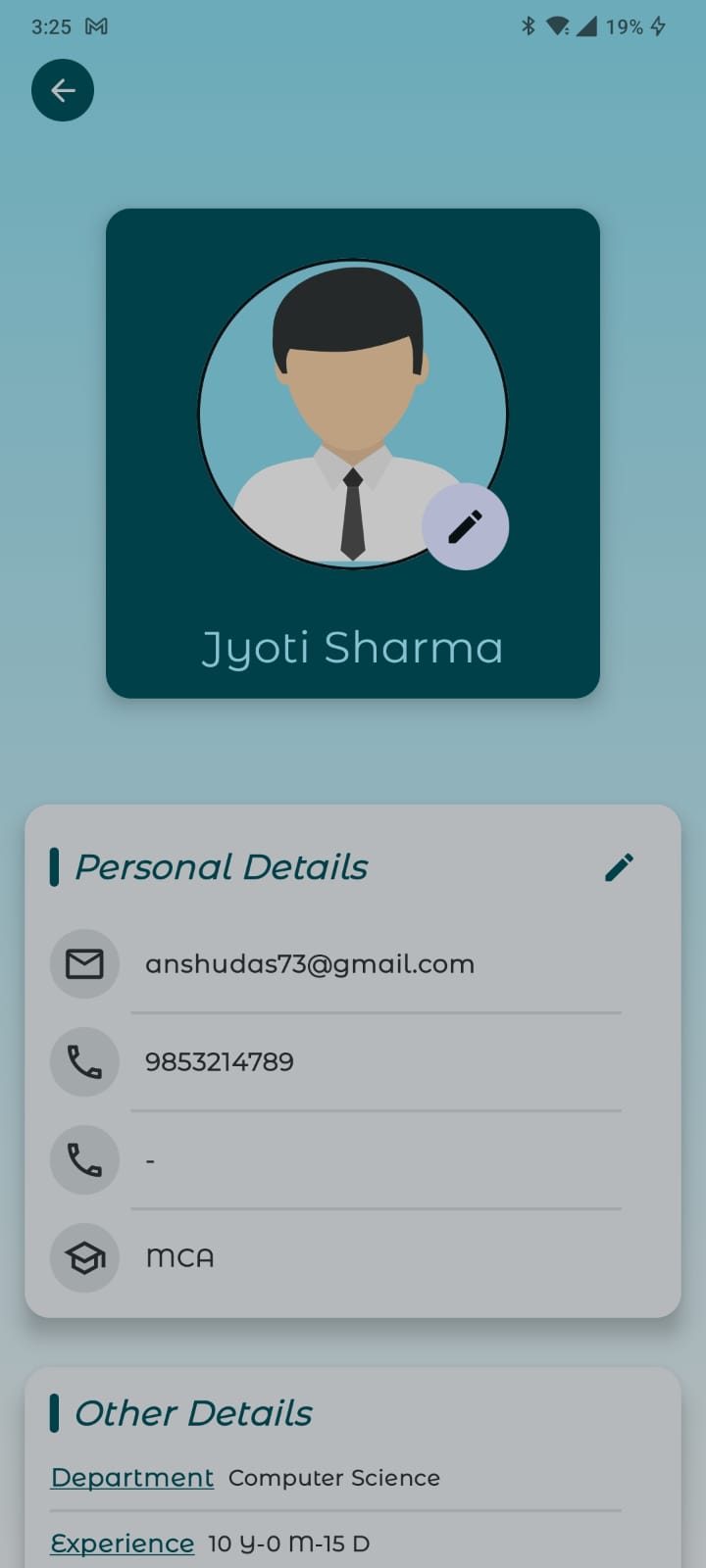
**3. Leave History 4. Approve/Reject Leave**



**6. Approved/Rejected Leave 7. View/Download Report**

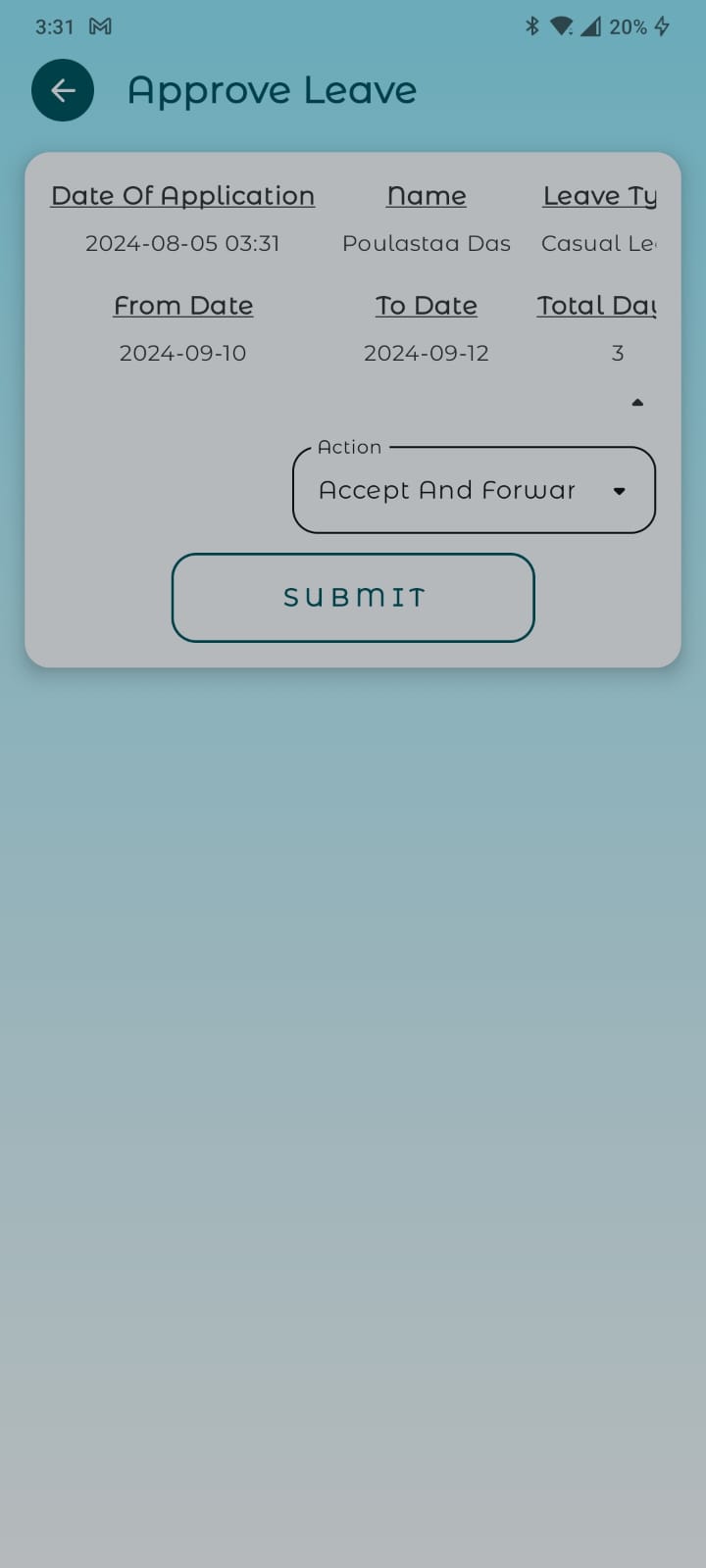
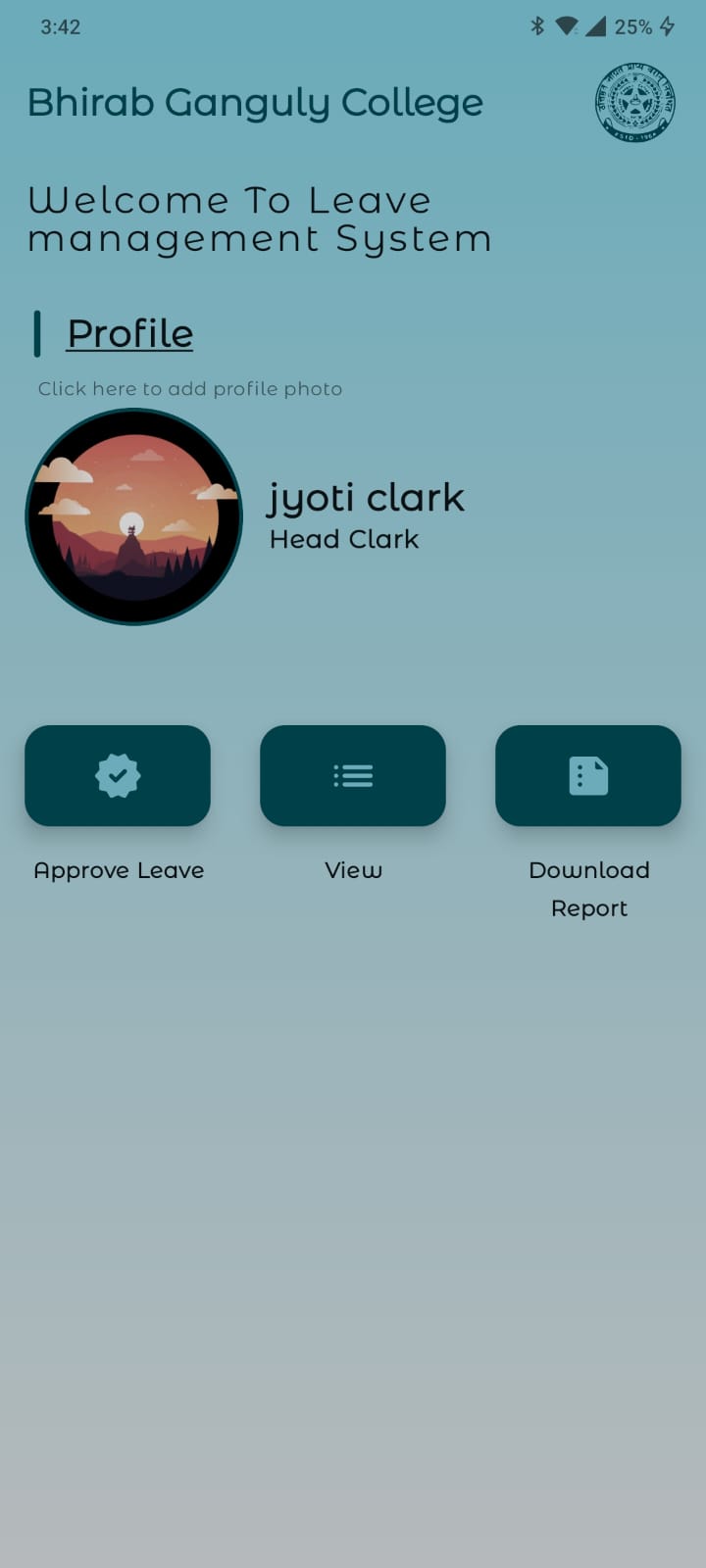


**8. Profile**

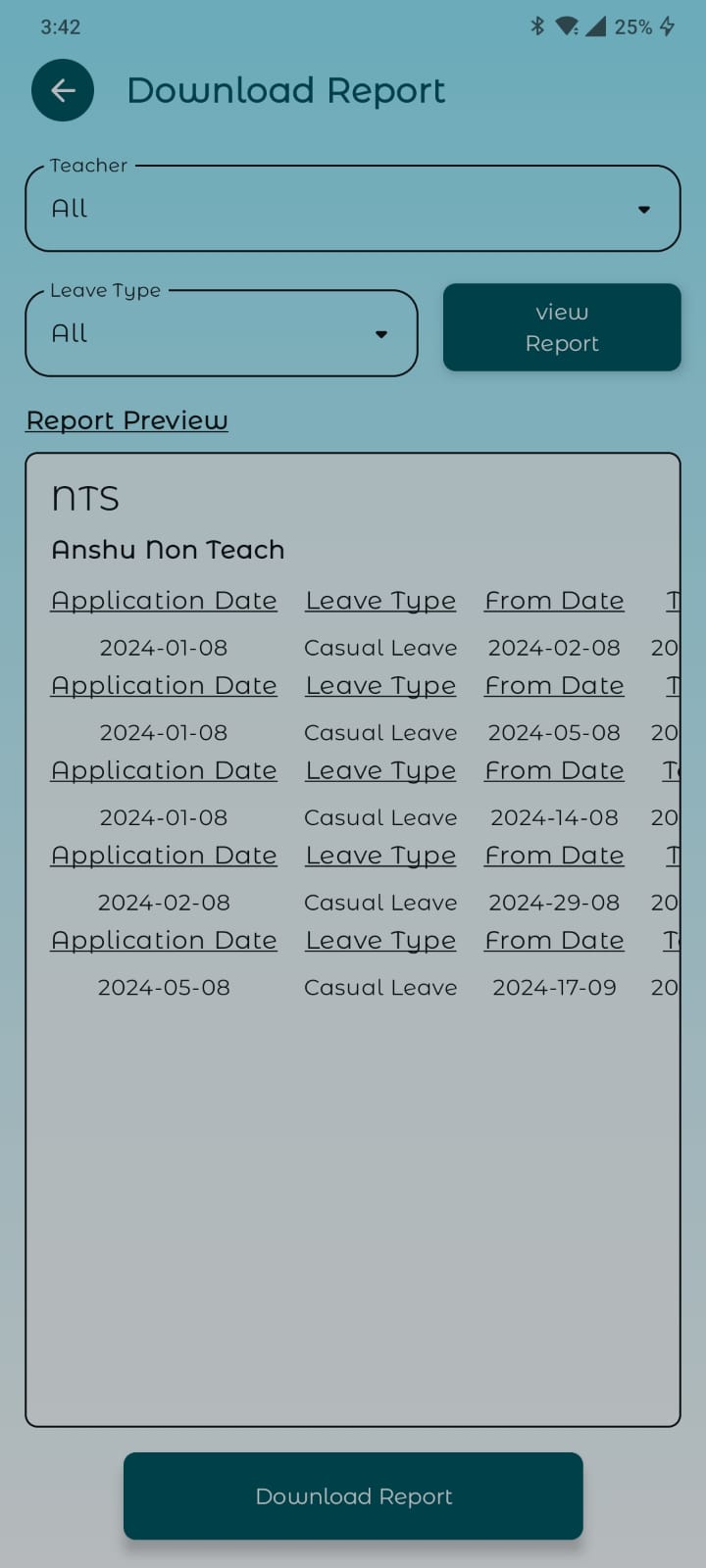
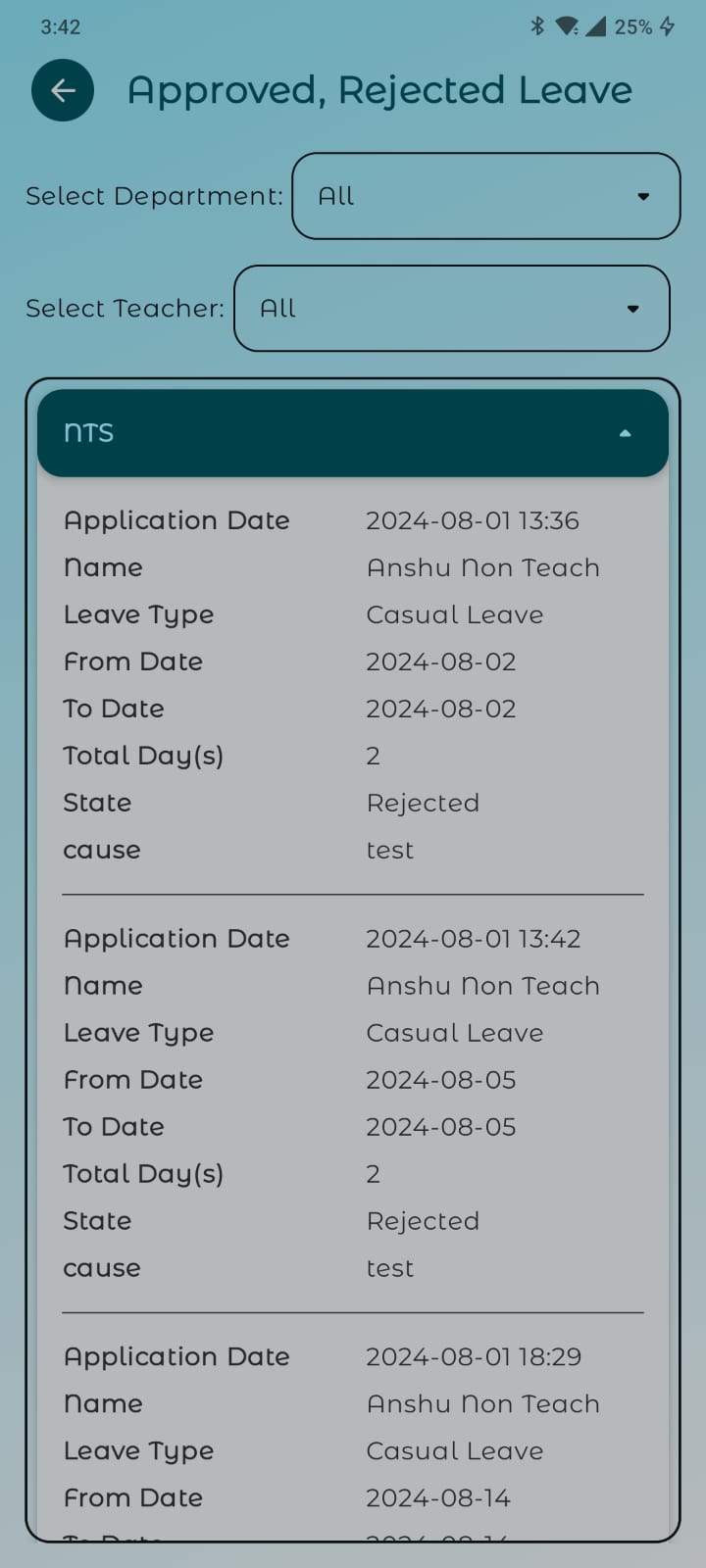


**Head Clark**

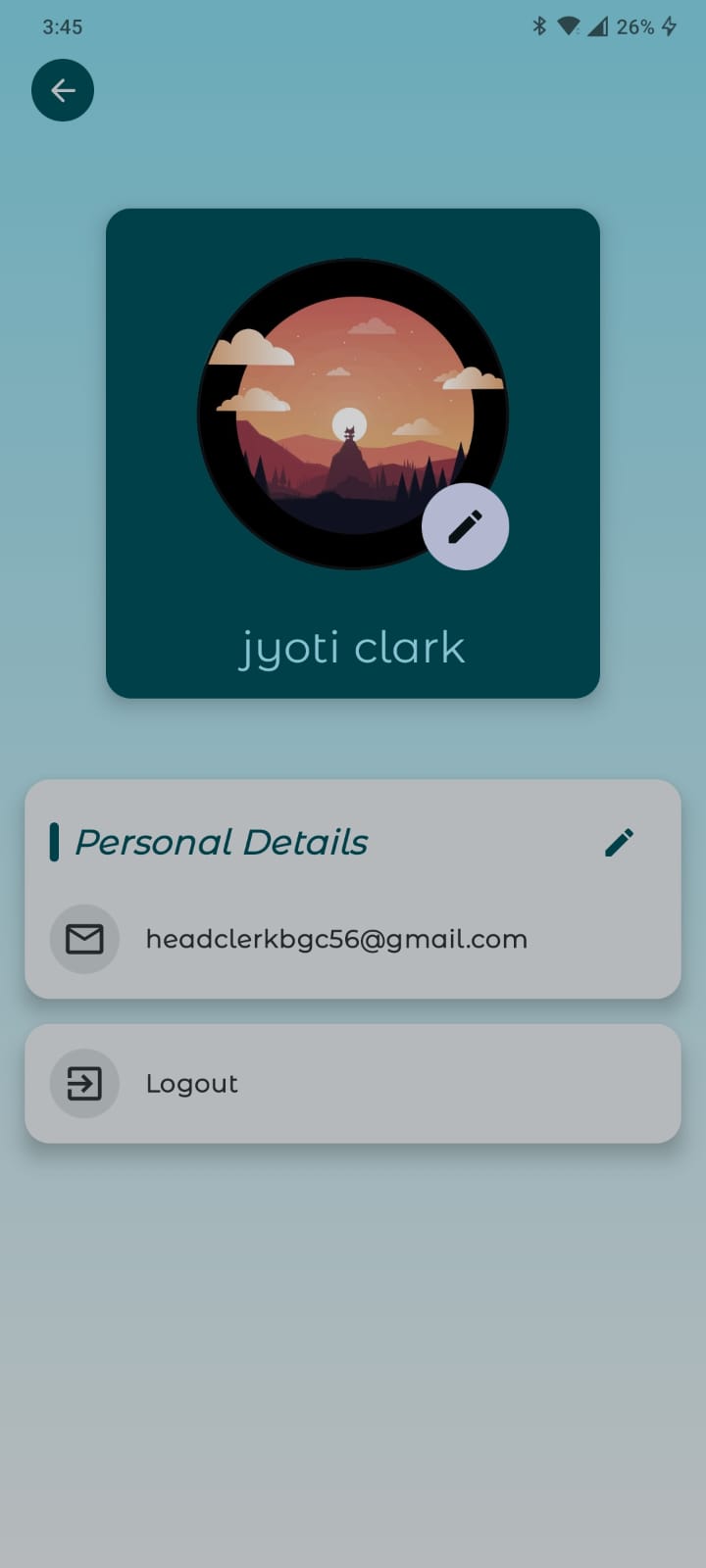
**1. Dashboard Approve/Reject Leave**



**3. Approved/Rejected Leave 4. Download Report**

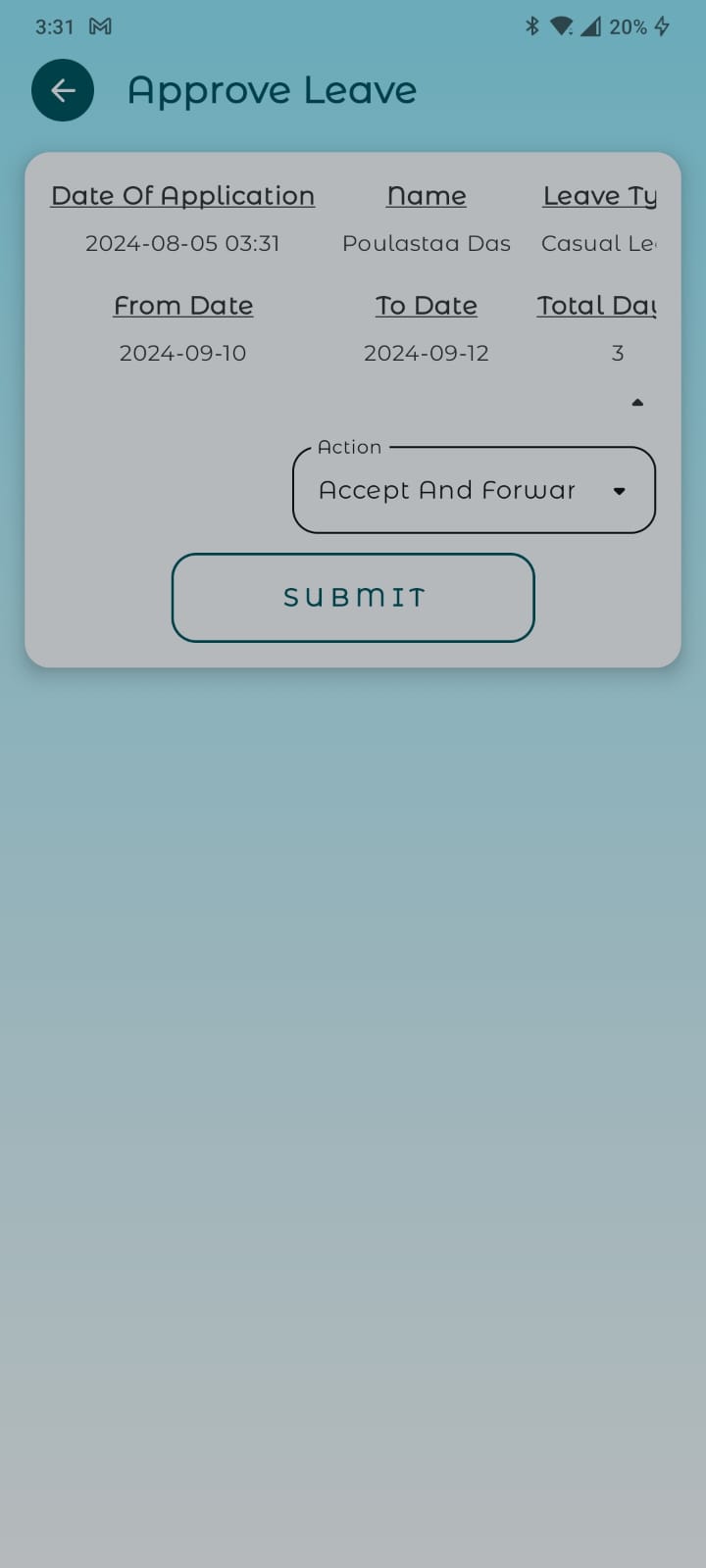
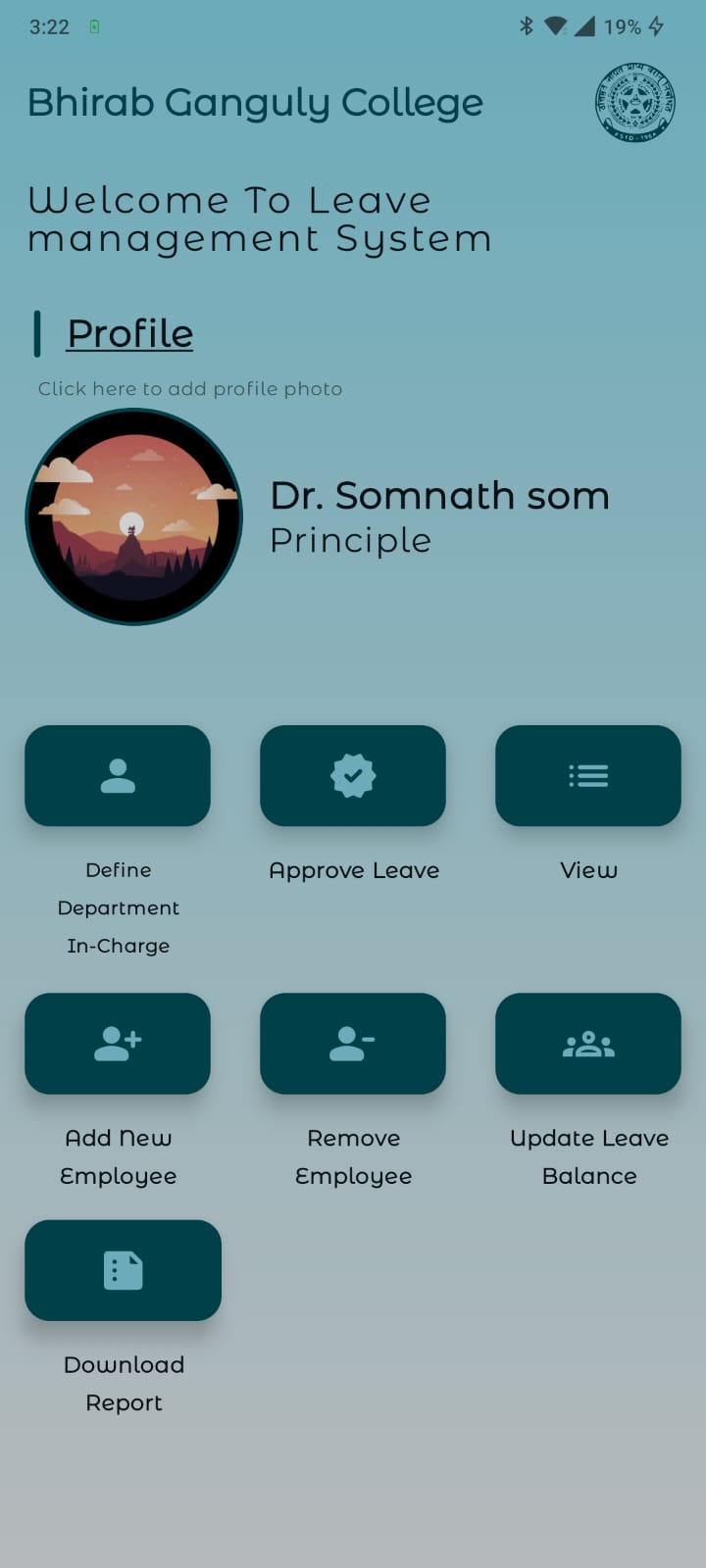


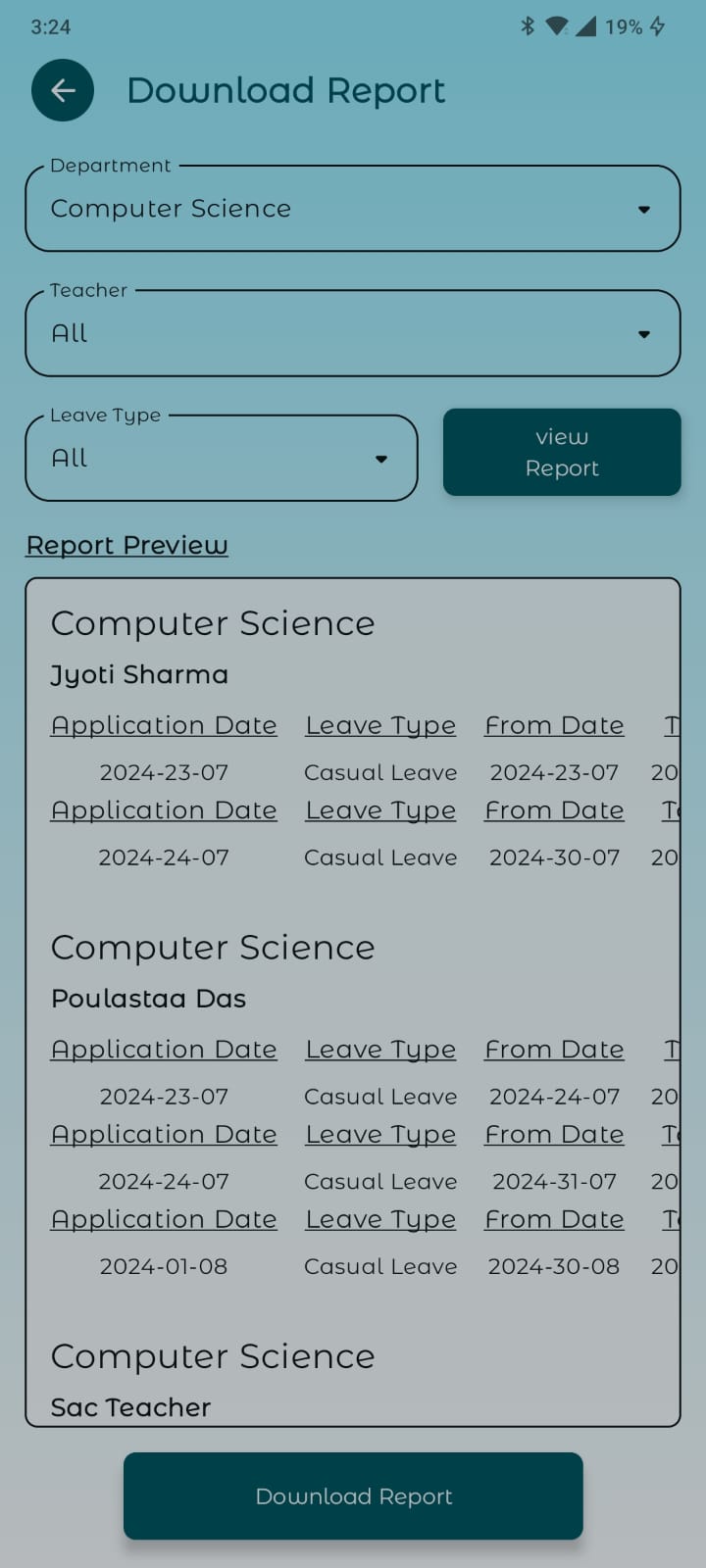
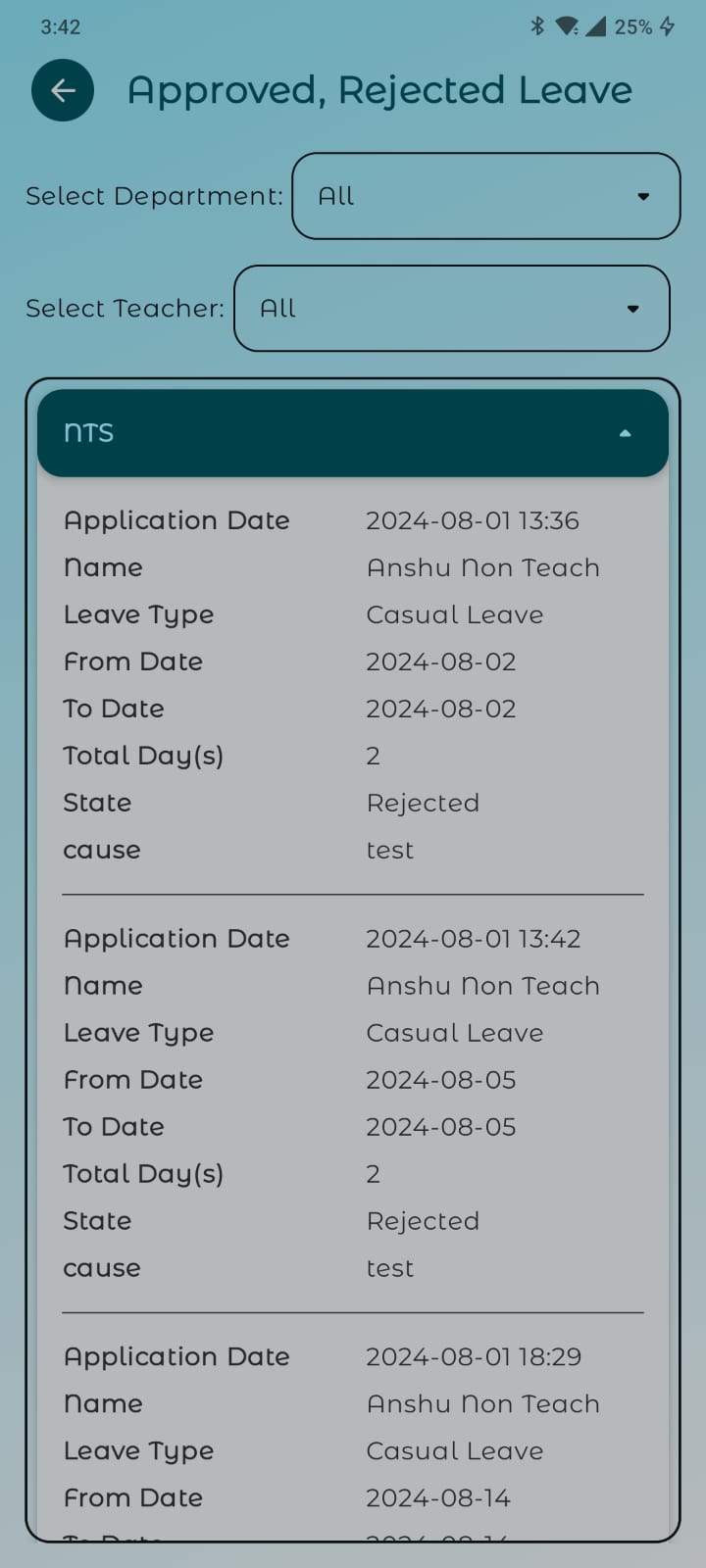
**5. Profile**



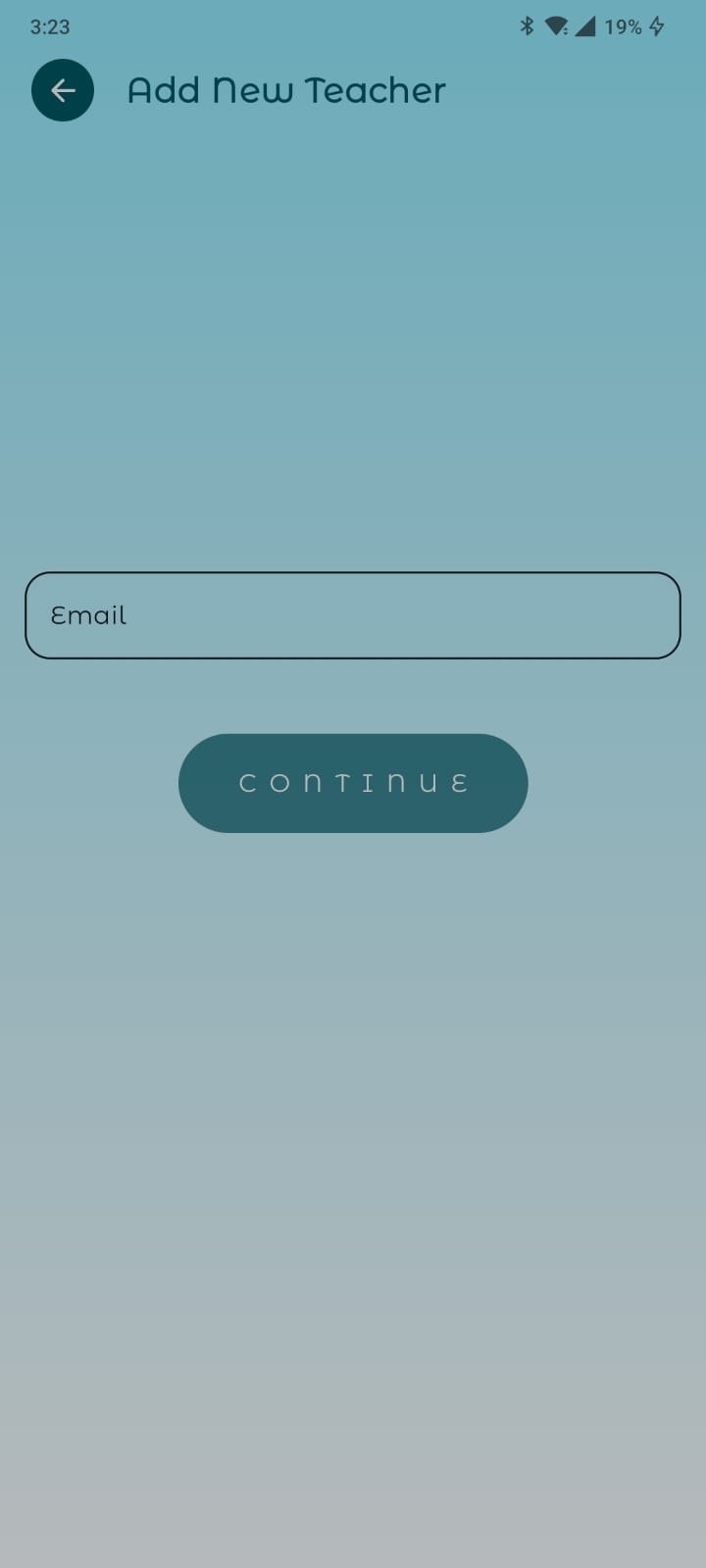
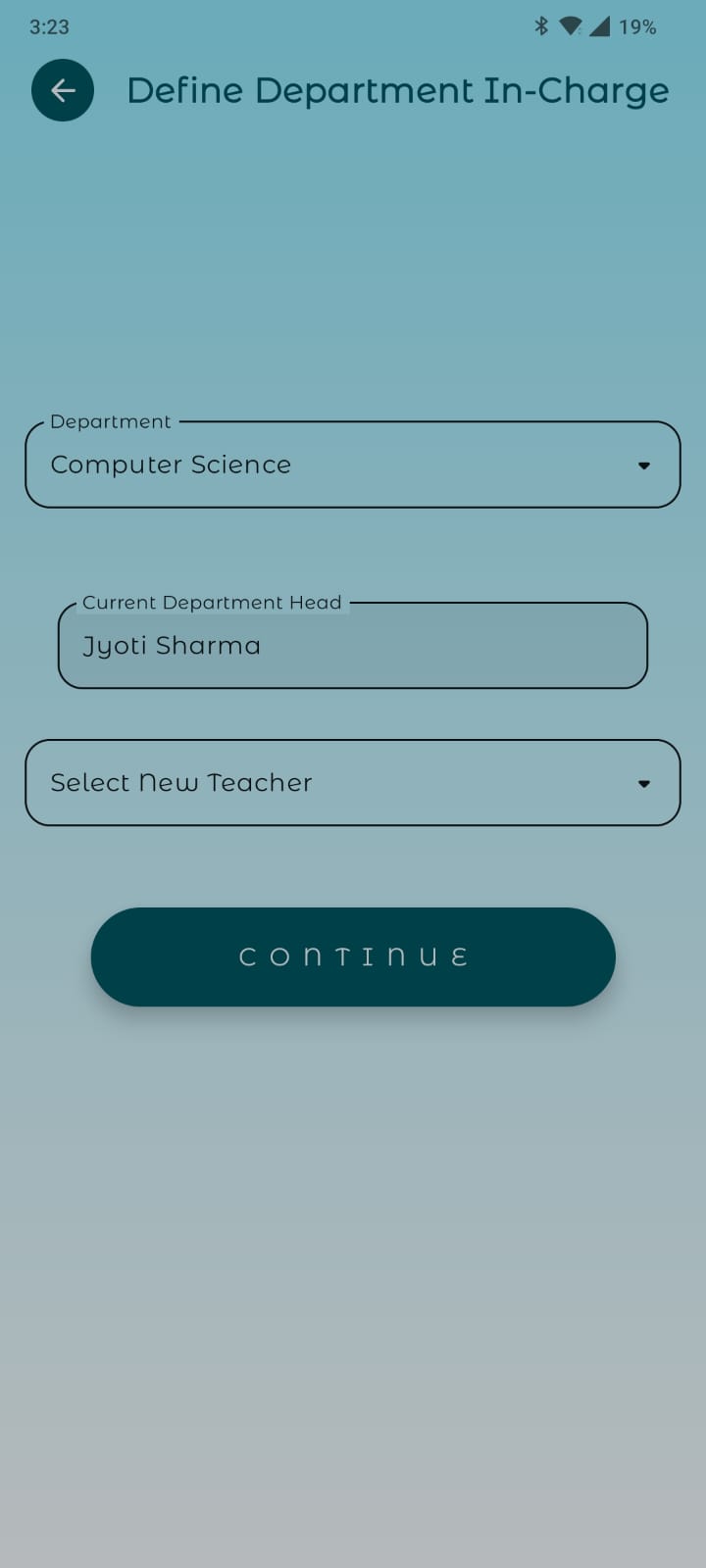
**Principal**

**1. Dashboard Approve/Reject Leave**

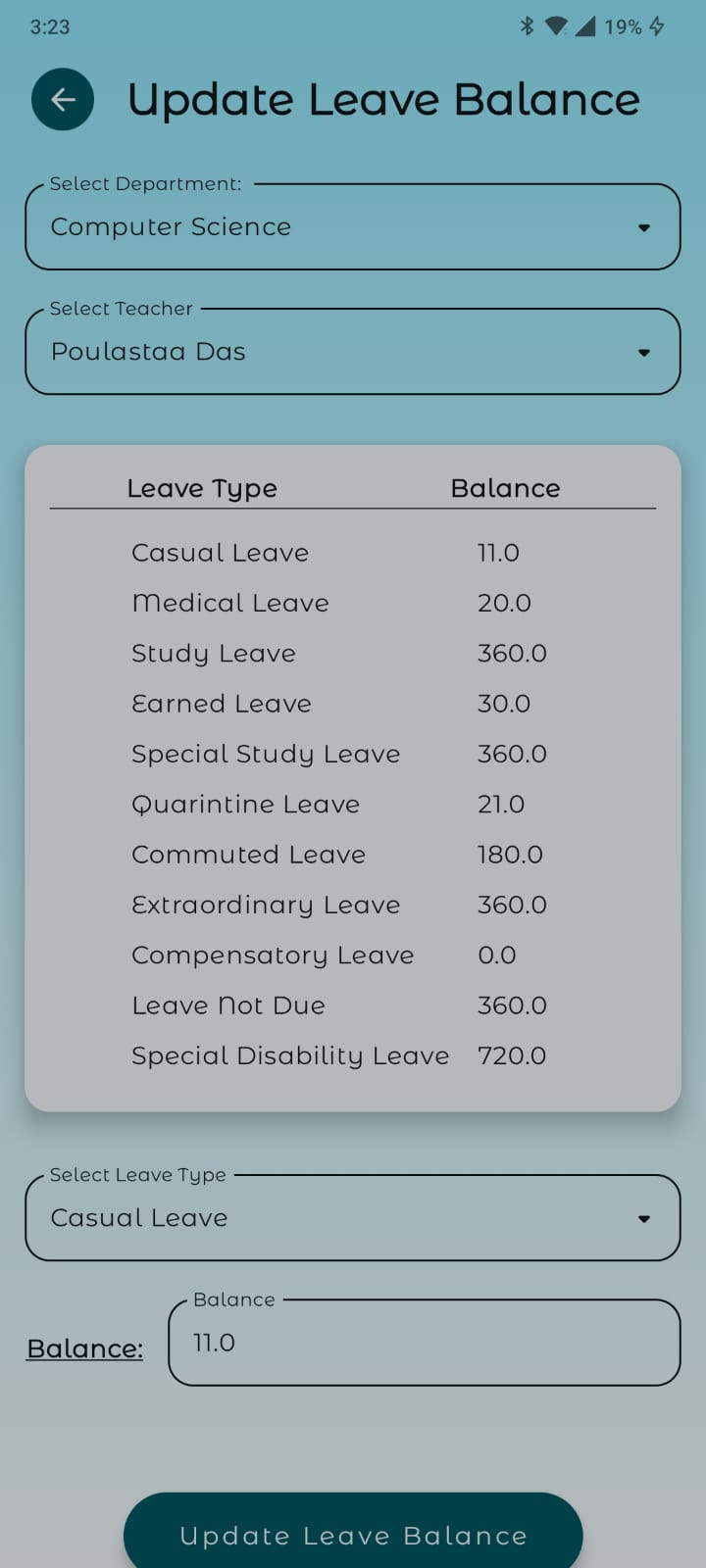
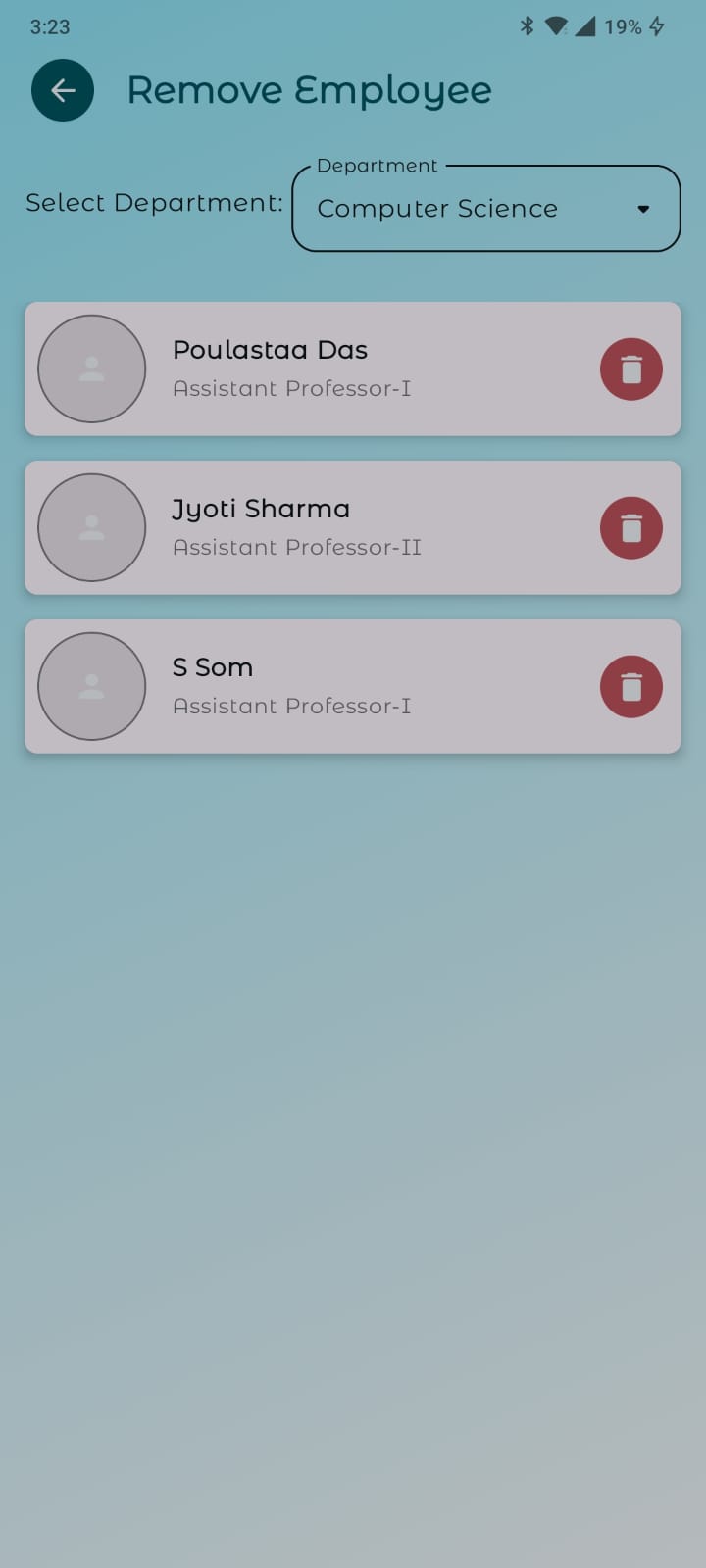


**3. Approved/Rejected Leave 4.Download Report**

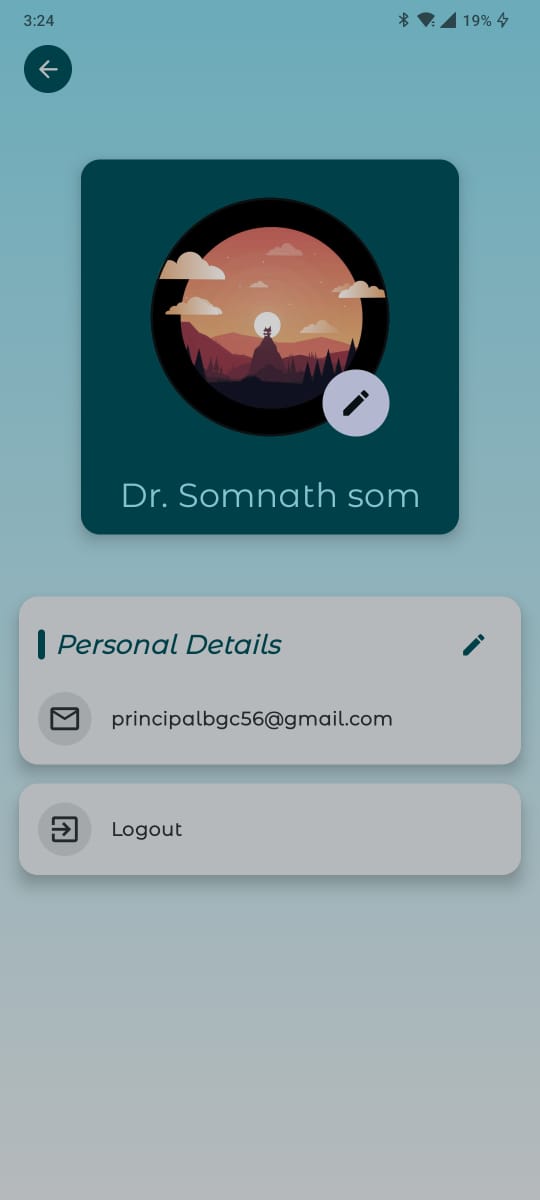
**5. Define Department In-charge 6. Add New Teacher**



**7. Remove An Employee 8. Update Leave balance**



**9. Profile**



**2.1.9 ADVANTAGE FOR EMPLOYEE**

A Leave Management System (LMS) provides several advantages to employee, making the process of managing time off more efficient and transparent. Here are some key benefits for users:

**1.Streamlined Requests:** Employee can submit leave requests through a simple and intuitive mobile interface, reducing the need for paperwork or manual processes.

**2.Real-Time Tracking:** Employee can track the status of their leave requests in real-time, seeing whether they’ve been approved, denied, or are still pending.

**3.Calendar Integration:** Our LMS platforms integrate with calendar applications, allowing users to view their leave schedule alongside their work calendar, which helps in better planning and avoiding conflicts.

**4.Self-Service:** Users can view their leave balances (e.g., vacation days, sick leave) and manage their own requests without needing to contact management directly.

**5.Automated Notifications:** Automated email notifications keep employee informed about the status of their leave requests and any changes or updates.

**6.Policy Compliance:** Our LMS platforms includes built-in leave policies and guidelines, ensuring that users are aware of college rules regarding leave entitlements and usage.

**7.Easy Documentation:** Our systems allow employee to attach supporting documents or reasons for leave, which can help in the approval process.

**8.Historical Records:** Employee can access their leave history, helping them track their past leave patterns and plan future time off more effectively.

**9.Reduced Errors:** Automation reduces the risk of errors in leave calculations and approval processes, ensuring that employee’s leave balances are accurately maintained.

**10.Enhanced Transparency:** The system often provides visibility into who is on leave and when, which can help users coordinate with colleagues and manage workloads better.

**11.Mobile Access:** Our LMS solutions offer mobile apps, allowing employee and management to request and manage leave from anywhere, at any time.

**2.1.10 FUNCTIONAL REQUIRMENTS FOR EMPLOYEE**

In a Leave Management System (LMS), functional requirements describe what the system should be able to do. List of common functional requirements for such a system, focusing on user-related functionalities:

### **1.** **User Authentication and Authorization**

* **Login and Logout**: Users must be able to log in and log out securely.
* **User Roles**: The system should support different user roles, such as Employees, HOD, Head Clark, and Principal, with different access levels.

### **2**. **Leave Application**

* **Apply for Leave**: Employees should be able to apply for different types of leave (e.g., casual leave, medical leave, maternity leave).
* **Leave Balance**: The system should display the available leave balance for each type of leave.
* **Leave Calendar**: Users should view their leave history and scheduled leaves.

### **3.** **Leave Approval Workflow**

* **Manager Approval**: Leave requests should be routed to the appropriate manager for approval.
* **Notifications**: Notifications should be sent to both the employee and the manager regarding the status of the leave request (approved, rejected, pending).
* **Multiple Approvals**: The system should support multiple levels of approval if required.

### **4.** **Leave Management**

* **Leave Cancellation**: Employees should be able to request the cancellation of an approved leave.
* **Leave Modification**: Employees should be able to modify their leave requests before approval.
* **Leave History**: Employees should access a history of their leave applications and statuses.

### **5.** **Reporting and Analytics**

* **Reports**: The system should generate reports on leave usage, pending leaves, etc., accessible to HR and managers.
* **Dashboards**: The system should offer a dashboard view of leave statistics, such as total leaves taken, most common leave types, etc.

### **6.** **Notifications and Alerts**

* **Email/SMS Notifications**: The system should send notifications and alerts for actions like leave approval, rejection, reminders for pending approvals, etc.
* **In-System Notifications**: Notifications should also be accessible within the system interface.

### **7.** **Self-Service Features**

* **Profile Management**: Employee should be able to manage their profiles, including contact information and emergency contacts.
* **Policy Access**: Employees should have access to the college's leave policies and guidelines.

### **Admin Functions**

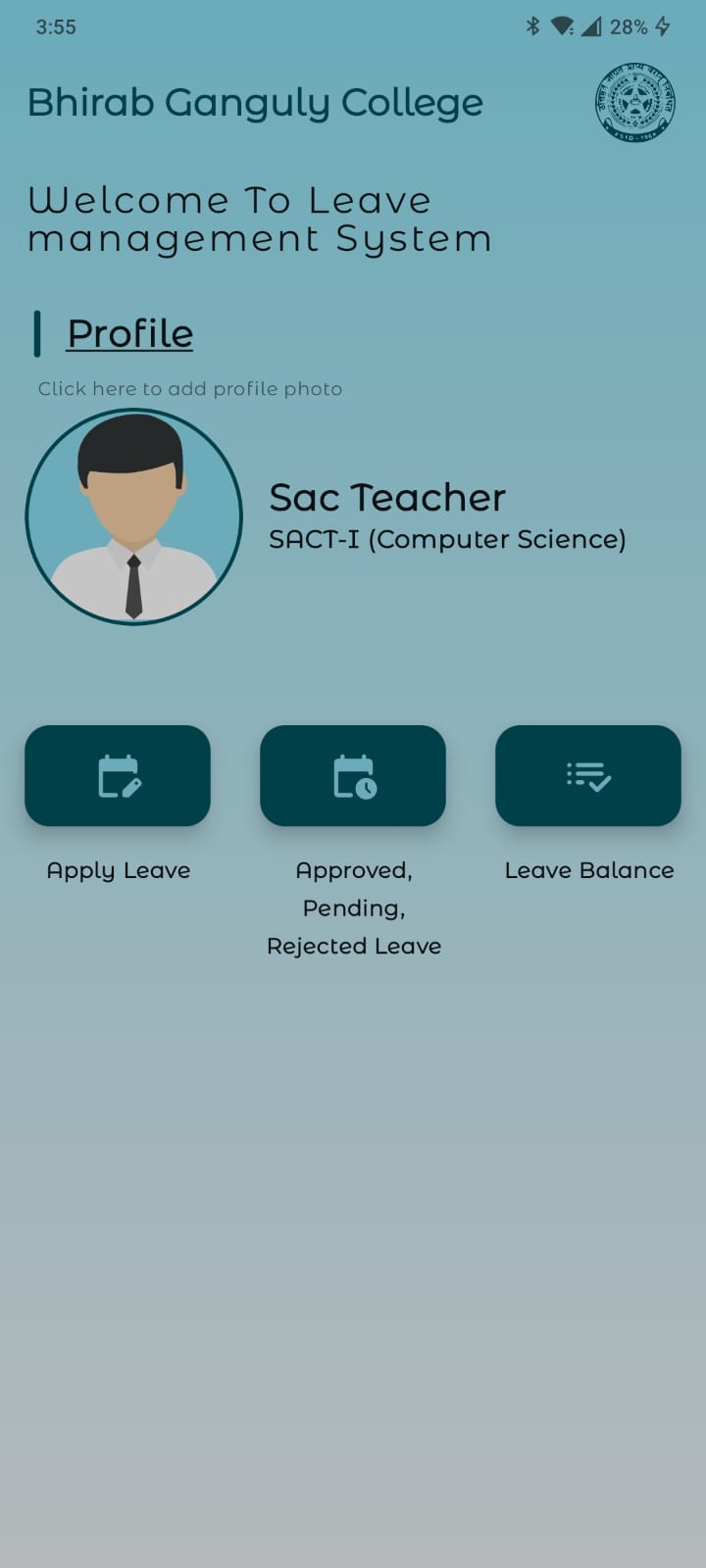
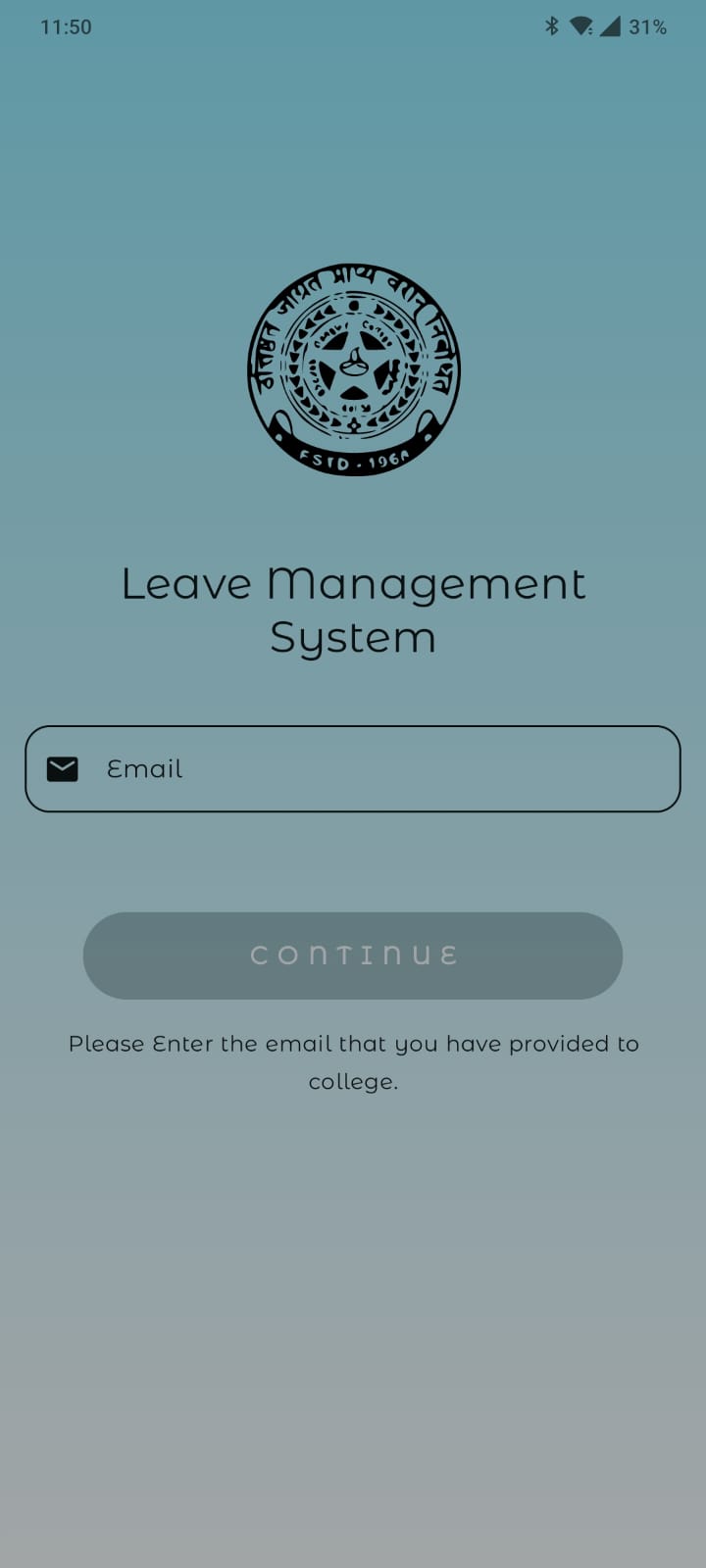
* **Approve/Reject leave:** HOD, Head Clark, Principal can approve or reject leave.
* **View leave:** HOD, Head Clark, Principal can view approved or rejected leave.
* **Edit an Employee:** Principal can add or remove old employee.
* **Edit leave balance:** Principal can edit employee’s leave balance.
* **View and download report:** HOD, Head Clark, Principal can view and download pdf report of leave.
* **Edit Departmental In-Charge:** Principal can change Departmental In charge.

### **10.** **Security and Compliance**

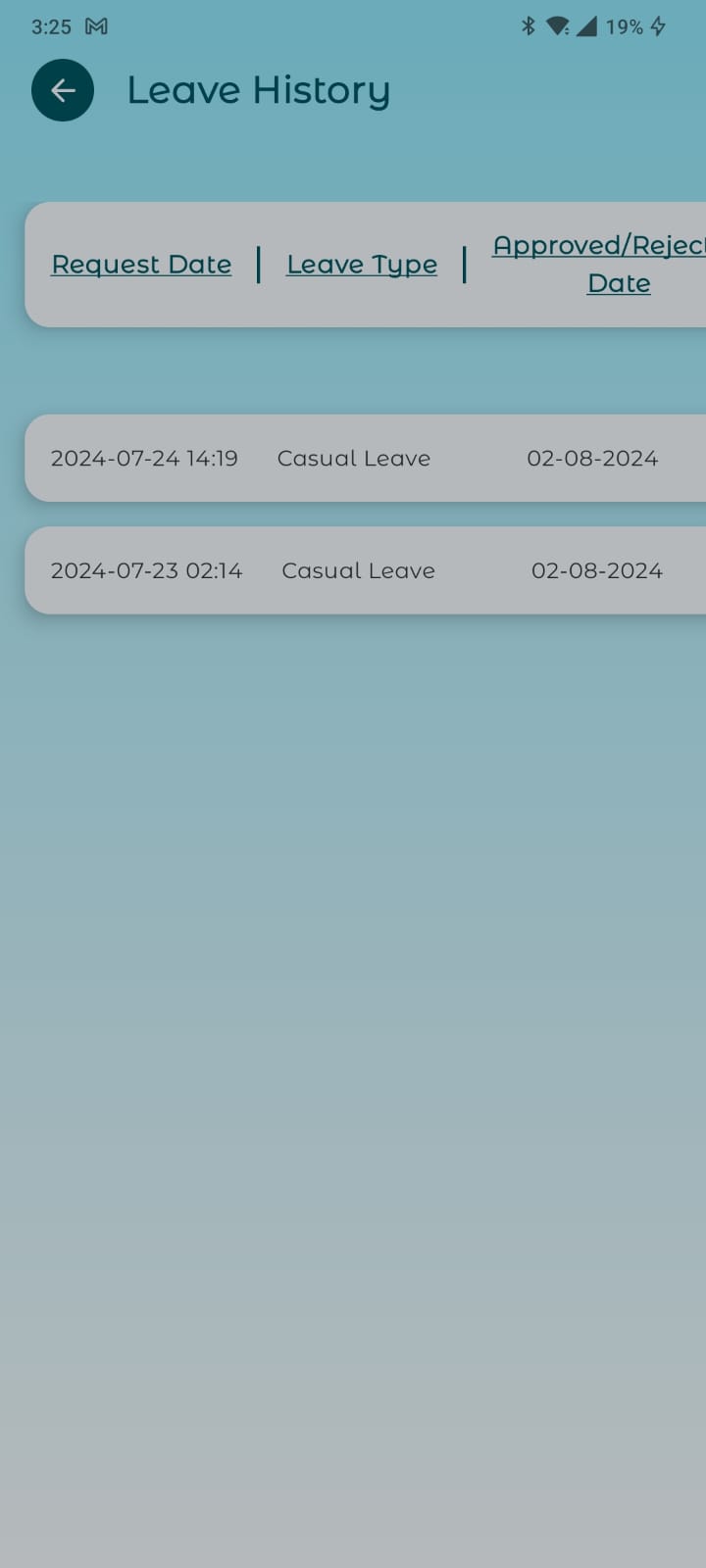
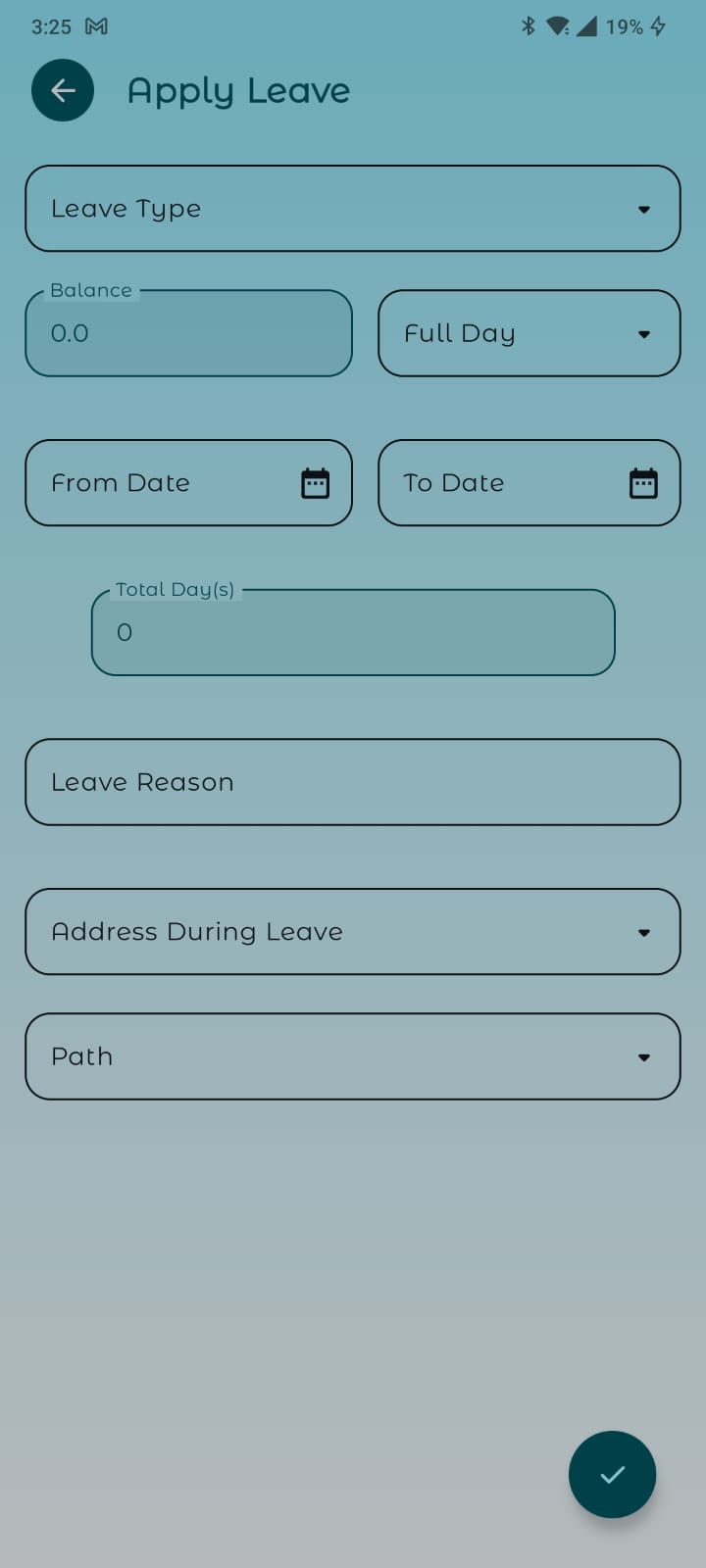
* **Data Security**: Ensure that user data is stored securely and access is controlled based on roles.
* **Compliance**: Ensure the system complies with relevant laws and regulations.

**SAC Teacher**

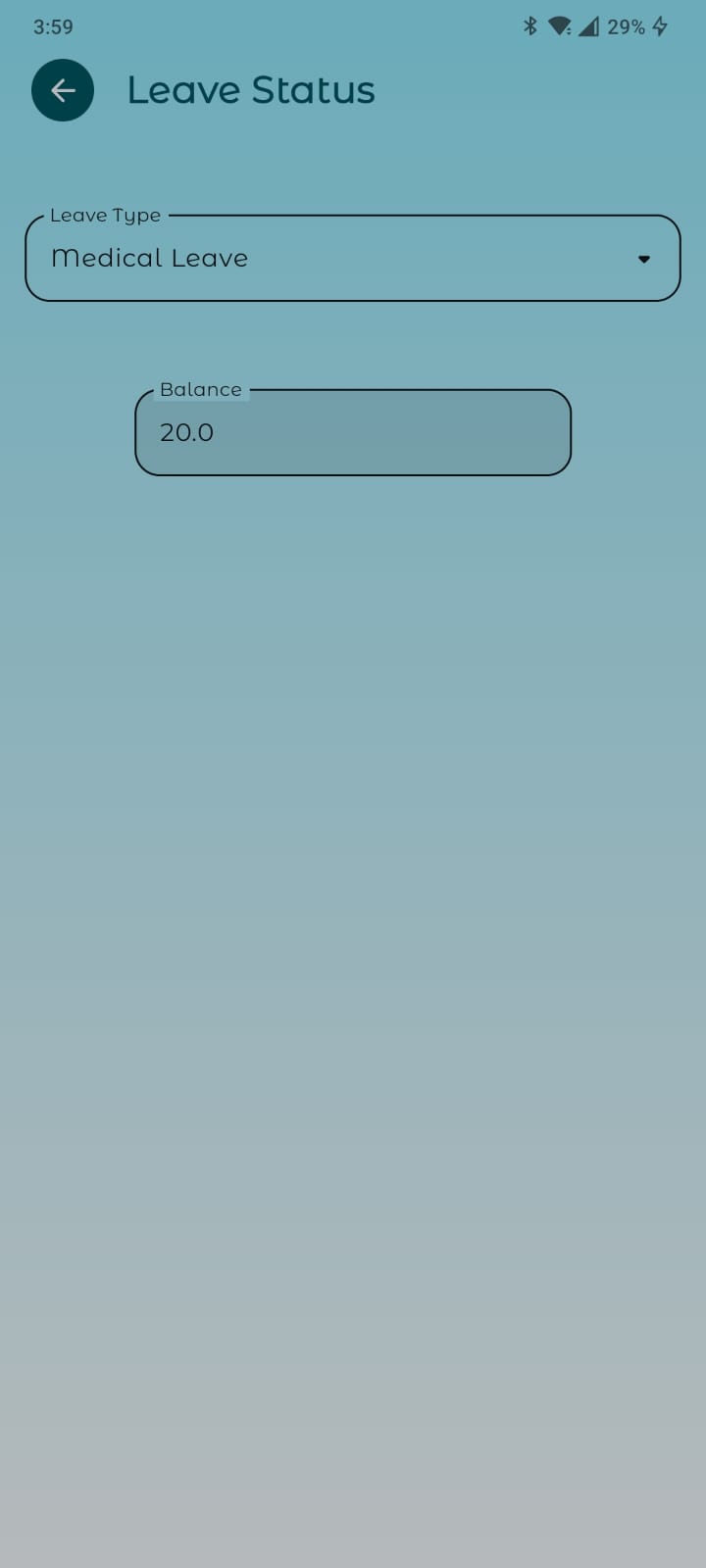
**1. Login 2. Dashboard**



**3. Apply Leave 4. Leve History**



**5. Leave balance**



**2.1.10 NON-FUNCTIONAL REQUIRMENTS OF EMPLOYEE**

Non-functional requirements define the quality attributes, performance, and constraints of a system. For a Leave Management System (LMS), non-functional requirements ensure that the system meets certain standards beyond the core functionalities. Here are some common non-functional requirements for such a system.

**1. Performance**

**Response Time:** The system should have a minimal response time for user actions, ideally under 3 seconds for most operations.

**Scalability:** The system should handle increasing numbers of users and data without performance degradation.

**Concurrent Users:** The system should support a specified number of concurrent users (e.g., 1,000 users) without performance issues.

**2. Reliability**

**Availability:** The system should have high availability, ideally 99.9% up time.

**Fail over and Recovery:** The system should have failed over mechanisms and quick recovery processes to handle system crashes or failures.

**Data Integrity:** Ensure that data is accurate and consistent, even in case of errors or failures.

**3. Security**

**Authentication and Authorization:** Implement strong authentication and authorization mechanisms to protect user data.

**Data Privacy:** Ensure compliance with data privacy regulations (e.g., GDPR, HIPAA) to protect user information.

**Audit Logging:** Maintain logs of all user activities for security auditing and monitoring.

**4. Usability**

**User Interface (UI):** The system should have an intuitive and user-friendly interface, making it easy for users to navigate and perform tasks.

**Accessibility:** The system should be accessible to users with disabilities, adhering to accessibility standards (e.g., WCAG).

**5. Maintainability**

**Modularity:** The system should be modular, allowing for easy updates, upgrades, and maintenance.

**Documentation:** Provide comprehensive documentation for users, administrators, and developers.

**Code Quality:** Follow coding standards and best practices to ensure high-quality, maintainable code.

**6. Scalability**

**Horizontal and Vertical Scaling:** The system should support both horizontal and vertical scaling to accommodate growing user bases and data volumes.

**Load Balancing:** Implement load balancing to distribute traffic evenly across servers.

**7. Backup and Recovery**

**Data Backup:** Regularly back up data to prevent data loss.

**Disaster Recovery:** Have a disaster recovery plan in place to restore services in case of a catastrophic event.

These non-functional requirements help ensure that the Leave management System is robust, user-friendly, secure, and capable of meeting the needs of its users under various conditions.

**2.2 SOFTWARE TOOLS**

* **Android Studio:** Android Studio is the official integrated development environment (IDE) for Android app development, based on IntelliJ IDEA. It offers extensive tools for building, testing, and debugging Android applications, including a rich code editor and robust emulator.
* **Android Emulator:** The Android Emulator is a tool within Android Studio that simulates Android devices on your computer, allowing developers to test and run applications in a virtual environment. It supports various Android versions and device configurations.
* **Intlej Idea:** IntelliJ IDEA is a versatile IDE developed by JetBrains, widely used for Kotlin and Java development. It offers advanced code navigation, refactoring tools, and integration with various frameworks, making it a popular choice for backend and mobile development.
* **MySQL Workbench:** MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. It provides data modelling, SQL development, and comprehensive administration tools for server configuration, user administration, and performance tuning.
* **Postman:** Postman is a popular collaboration platform for API development. It provides a user-friendly interface to design, test, and document APIs, allowing developers to send requests, analyse responses, and automate testing processes.

**2.3 DATABASE**

* **SQL(MYSQL):** SQL (Structured Query Language) is a standard language for managing and manipulating relational databases. MySQL is a widely used open-source relational database management system that uses SQL for querying and managing data. It is known for its reliability, scalability, and ease of use, making it a popular choice for web applications and various software projects.

**2.4 DEPLOYMENT**

* **PASS:** Platform as a Service

Resources:

1. CPU: Intel Xeon V3 at 4.9 gHz

2. Memory: 8gb DDR4 at 3800 mHz

3. Disk:

a. Primary- 100gb hdd

b. Secondary- 500gb storage lite server.

4. OS: UBUNTU server 24.04

**2.5 HARDWARE SPECIFICATION (MINIMUM)**

1. **CPU** - Intel Core 2 duo
2. **RAM** – 4GB DDR3
3. **OS**- UBUNTU

**2.5 FRONTEND TECHNOLOGIES**

* **Kotlin:** Kotlin is a modern, statically-typed programming language developed by JetBrains, fully interoperable with Java. It is designed to improve code readability and safety, making it a popular choice for Android development and various server-side applications.
* **Jetpack compose:** Jetpack Compose is a modern toolkit for building native Android UI. It simplifies and accelerates UI development with less code, powerful tools, and intuitive Kotlin APIs, allowing developers to create beautiful, responsive applications with ease.
* **Datastore Preferences:** DataStore is a Jetpack library that provides a modern, asynchronous API for storing data. It replaces SharedPreferences for storing key-value pairs and offers two storage solutions: Preferences DataStore for storing simple key-value pairs and Proto DataStore for storing typed objects.
* **OKHttp:** OkHttp is a popular HTTP client library for Android and Java applications. It efficiently handles network requests and responses, supports HTTP/2 and WebSocket, and provides features like connection pooling, transparent GZIP compression, and response caching to optimize network operations.

**2.6 BACKEND TECHNOLOGY**

* **Ktor:** Ktor is an asynchronous web framework built using Kotlin. It allows developers to create web applications, microservices, and REST APIs with minimal boilerplate code. Ktor provides flexibility and extensibility through its modular architecture.
* **Exposed:** Exposed is a lightweight SQL library for Kotlin that simplifies database access and interaction. It provides both DSL (Domain-Specific Language) and DAO (Data Access Object) APIs, making it easier to perform database operations and manage data models.
* **MySQL:** MySQL is a widely-used open-source relational database management system known for its performance, reliability, and ease of use. It is commonly used in web applications to store and manage data, supporting SQL for querying and managing relational data.

## CHAPTER 3

**ITERATIVE MODEL**

**3.1 FEASIBILITY STUDY**

A feasibility study for a leave management system involves analyzing the practicality and potential success of developing and implementing such a system. This study helps determine whether the project is viable and worth pursuing.

### **3.1.1 Technical Feasibility**

**System Architecture**: Overview of the system's technical structure

**Technology Stack**: Recommended technologies (programming languages, databases, frameworks)

**Security Considerations**: Data security, user authentication, authorization

### **3.1.2 Operational Feasibility**

**User Requirements**: Needs and expectations of users

**Training and Support**: Plans for training users and providing ongoing support

**Operational Workflow**: How the system will fit into current processes and workflows

### **3.1.3 Financial Feasibility**

**Cost Analysis**: Initial development costs, ongoing maintenance, and operational costs

**Benefit Analysis**: Expected benefits, including time savings, efficiency gains, and reduced errors.

## Benefits of Iterative Model

**Early Delivery and Feedback:** Features are delivered incrementally for early use, with continuous feedback ensuring the system meets user needs.

**Reduced Risk:** Smaller changes reduce errors and allow easy adaptation to new requirements.

**Better Quality Assurance:** Smaller, incremental releases enable thorough testing and continuous integration, ensuring each part of the system is well-integrated and high-quality.

**Faster Identification of Issues:** Bugs and issues can be identified and addressed more quickly due to the focused nature of each increment.

**Challenges and Mitigation**

**Rigid Phases:** The iterative model can be rigid, but the iterative approach allows for flexibility.

**Time and Cost Overruns:** Iterations can extend the timeline and increase costs; careful planning and scope management are essential.

**Complexity Management:** Managing changes and iterations can become complex; use project management tools and techniques to stay organized.

**3.2 REQUIREMENT ANALYSIS**

Requirement analysis for a Leave Management System involves identifying and documenting the needs. This process ensures that the system meets business objectives and user needs. Key aspects include:

* **User Requirements:** Understanding the needs of different user groups, such as employees, managers, HR personnel, and administrators. This includes features like applying for leave, approving requests, viewing leave balances, and generating reports.
* **Functional Requirements:** Defining the core functionalities of the system, such as leave application and approval workflows, leave balance tracking, notifications, reporting, and integration.
* **Non-Functional Requirements:** Establishing criteria for system performance, security, scalability, usability, and compliance. This includes ensuring data protection, fast response times, and an intuitive user interface.
* **User Interface and Experience Requirements:** Outlining the design and usability aspects, ensuring the system is easy to navigate and accessible to all users.
* **Reporting and Analytics Requirements:** Specifying the types of reports and analytics needed by different stakeholders, such as leave trends, utilization reports, and compliance tracking.
* **Data Management Requirements:** Defining how data will be stored, accessed, and managed, including data backup, recovery, and archiving processes.

**3.3 REQUIREMENT SPECIFICATIONS**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the requirements for a Leave Management System, which will help in managing employee leave requests and approvals efficiently.

**1.2 Scope**

The Leave Management System will automate the process of leave application, approval, tracking, and reporting within an organization.

**1.3 Definitions, Acronyms, and Abbreviations**

* LMS: Leave Management System
* HR: Human Resources
* GUI: Graphical User Interface

**2. Overall Description**

**2.1 Product Perspective**

* The Leave Management System is a standalone application that can be integrated with existing HR management systems.

**2.2 Product Functions**

* Employee leave application
* Manager leave approval/rejection
* Leave balance tracking
* Leave history and reports
* Notifications and alerts

**2.3 User Classes and Characteristics**

* Employees: Apply for leave, view leave balance, and leave history.
  + Managers: Approve/reject leave requests, view team leave schedules.
* HR: Manage leave policies, generate reports, and handle escalations.

**2.4 Operating Environment**

* Mobile application for Android platforms.

**2.5 Design and Implementation Constraints**

* System must ensure data security and privacy.
* System must be scalable to handle a large number of users.

**2.6 User Documentation**

* User manuals and help guides for employees, managers.

**2.7 Assumptions and Dependencies**

* Assumes a stable internet connection for mobile access.

**3. System Features**

**3.1 Employee Leave Application**

**3.1.1 Description and Priority**

* Employees can apply for leave by selecting leave type, start and end dates, and providing reasons.
* Priority: High

**3.1.2 Stimulus/Response Sequences**

* Employee logs in -> Navigates to Leave Application -> Fills form -> Submits application -> System sends notification to manager.

**3.1.3 Functional Requirements**

* The system shall allow employees to apply for different types of leave (e.g., annual leave, sick leave).
* The system shall validate leave dates against existing leave policies.

**3.2 Manager Leave Approval**

**3.2.1 Description and Priority**

* Managers can view and approve/reject leave requests.
* Priority: High

**3.2.2 Stimulus/Response Sequences**

* Manager logs in -> Navigates to Pending Leave Requests -> Reviews application -> Approves or Rejects -> System notifies employee.

**3.2.3 Functional Requirements**

* The system shall allow managers to view leave requests from their team.
* The system shall allow managers to approve or reject leave requests with comments.

**3.3 Leave Balance Tracking**

**3.3.1 Description and Priority**

* Employees and managers can view current leave balance.
* Priority: Medium

**3.3.2 Stimulus/Response Sequences**

* User logs in -> Navigates to Leave Balance -> Views available leave balance.

**3.3.3 Functional Requirements**

* The system shall display the leave balance for each leave type.
* The system shall update leave balance upon approval of leave requests.

**3.4 Leave History and Reports**

**3.4.1 Description and Priority**

* Users can view their leave history and generate reports.
* Priority: Medium

**3.4.2 Stimulus/Response Sequences**

* User logs in -> Navigates to Leave History -> Filters and views history -> Generates report.

**3.4.3 Functional Requirements**

* The system shall allow users to filter leave history by date range and leave type.
* The system shall generate downloadable leave history reports.

**3.5 Notifications and Alerts**

**3.5.1 Description and Priority**

* System sends notifications and alerts for leave request status changes.
* Priority: Medium

**3.5.2 Stimulus/Response Sequences**

* Leave request submitted -> System sends notification to HOD or Head Clark -> Principal approves/rejects -> System sends notification to employee.

**3.5.3 Functional Requirements**

* The system shall send email notifications for leave request submissions, approvals, rejections, and reminders.

**4. External Interface Requirements**

**4.1 User Interfaces**

* mobile-responsive design.
* Mobile app interface for Android.

**4.2 Hardware Interfaces**

* Not applicable.

**4.3 Software Interfaces**

* Integration with existing HR management systems.

**4.4 Communications Interfaces**

* HTTPS for secure communication between client and server.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system should handle concurrent users efficiently without performance degradation.

**5.2 Security Requirements**

* The system should ensure data privacy and security through authentication, authorization, and encryption.

**5.3 Software Quality Attributes**

* The system should be user-friendly, reliable, and maintainable.

**5.4 Business Rules**

* Leave policies and rules to be defined and managed by Principal.

**SYSTEM DESIGN**

LEVEL 0

Admin

USER

Admin

Manager

Authentication & Authorization

Leave management System Client

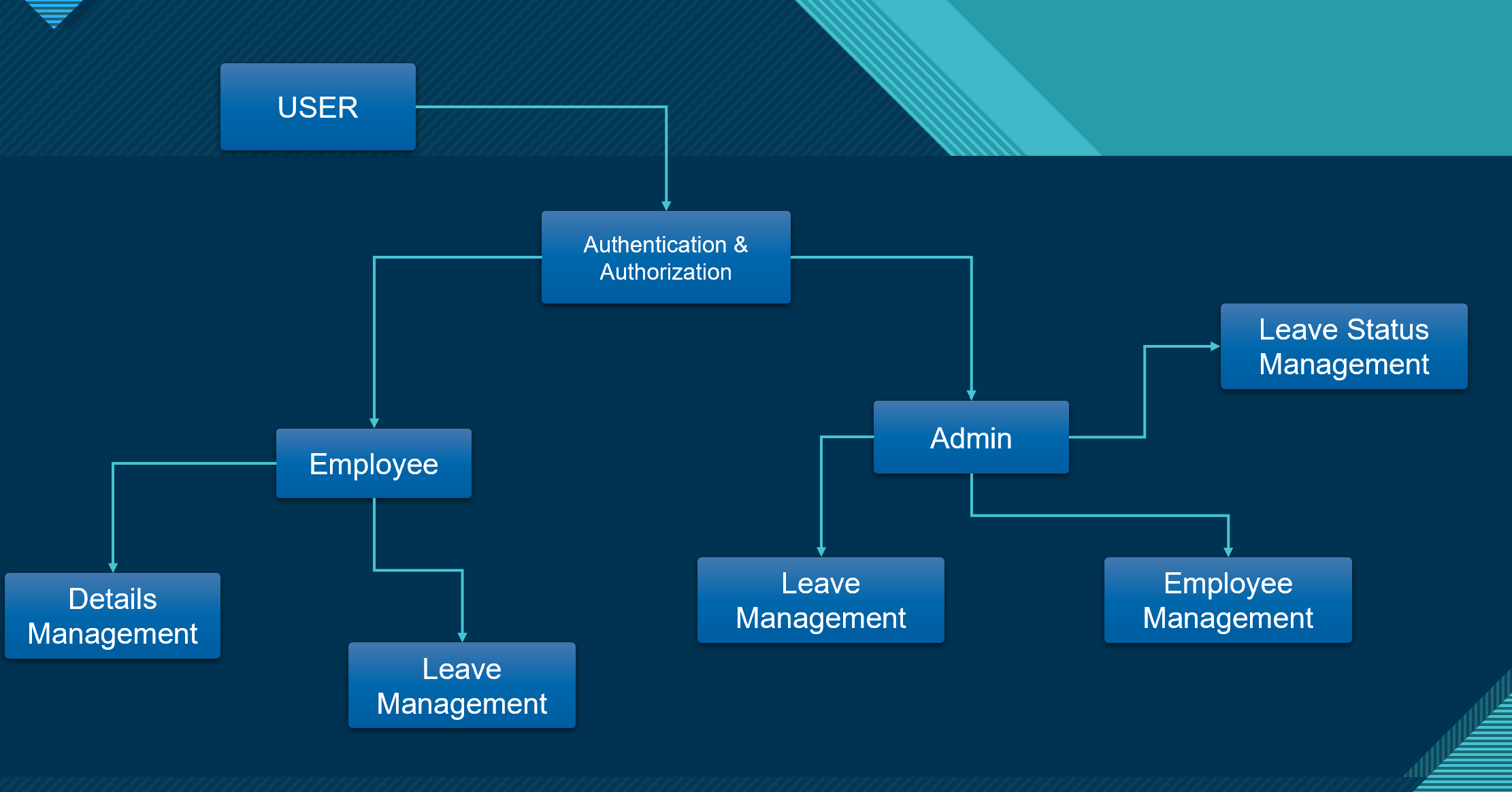
Leave

Management

Employee

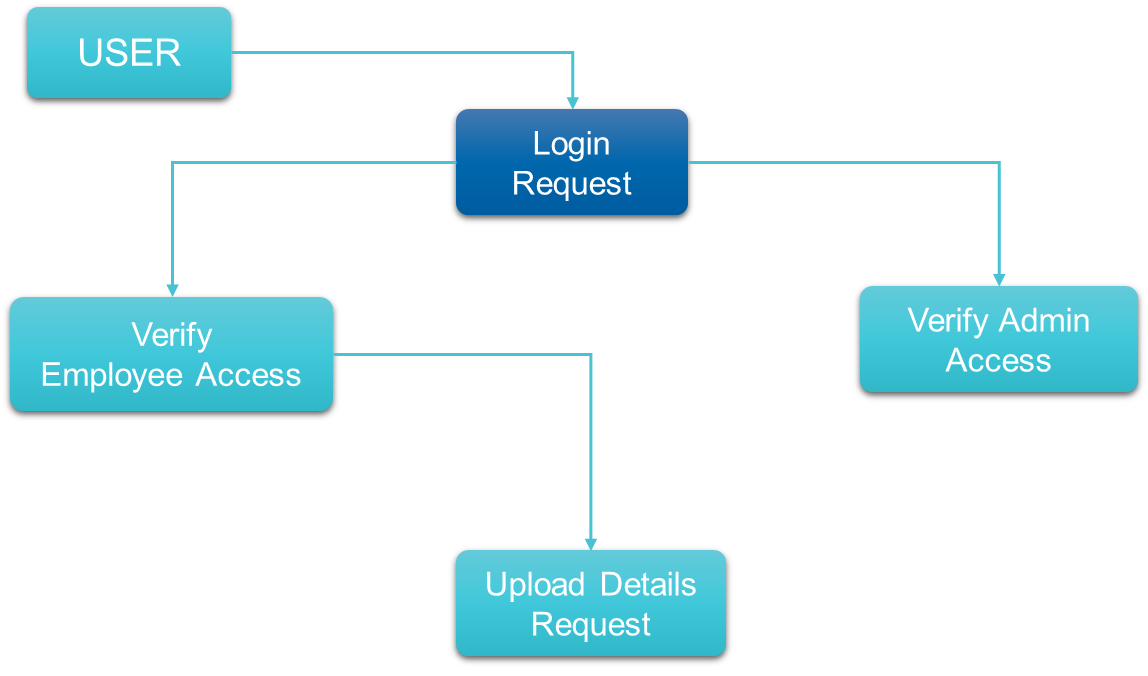
Management

LEVEL 1

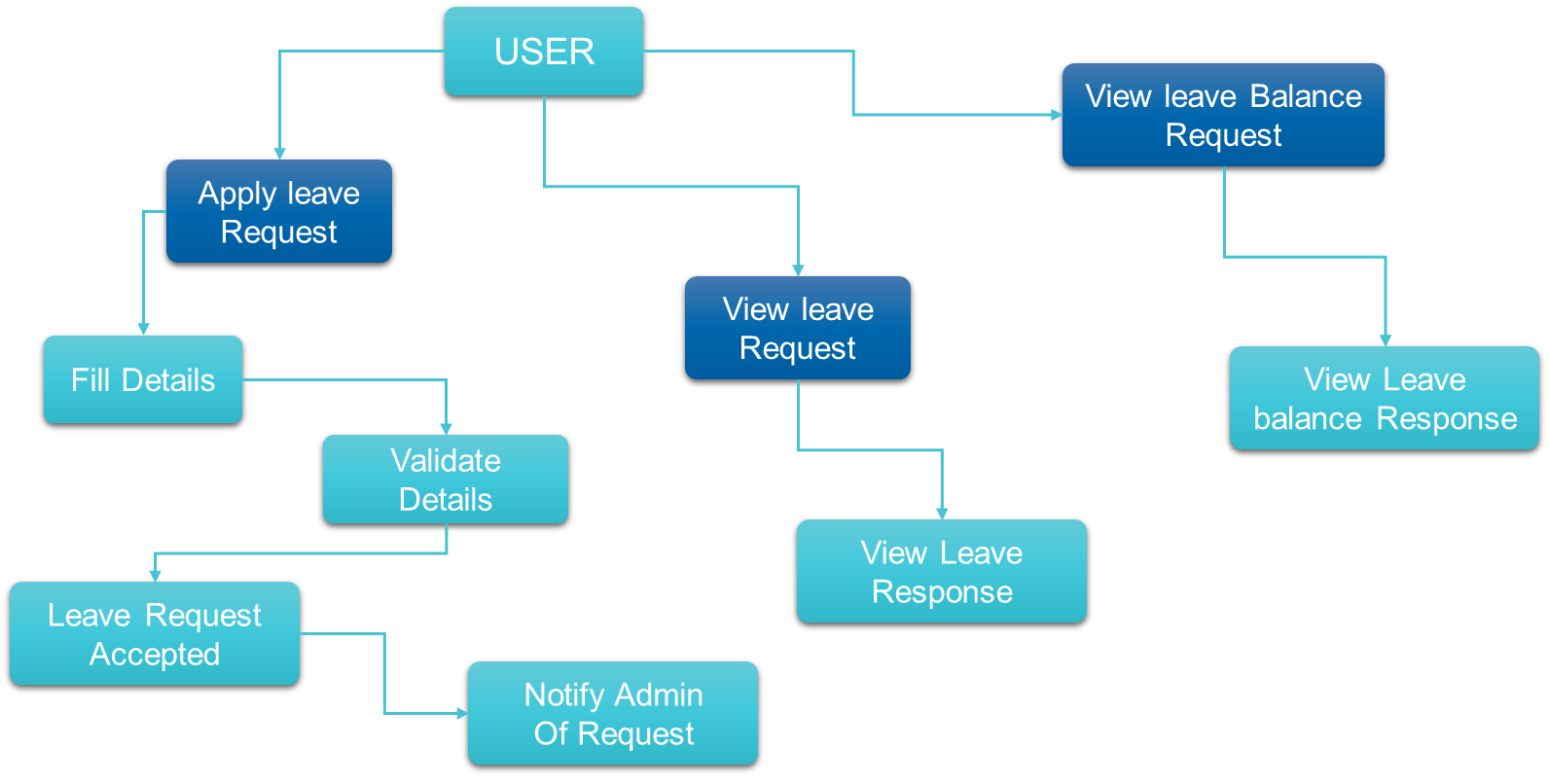


LEVEL 2

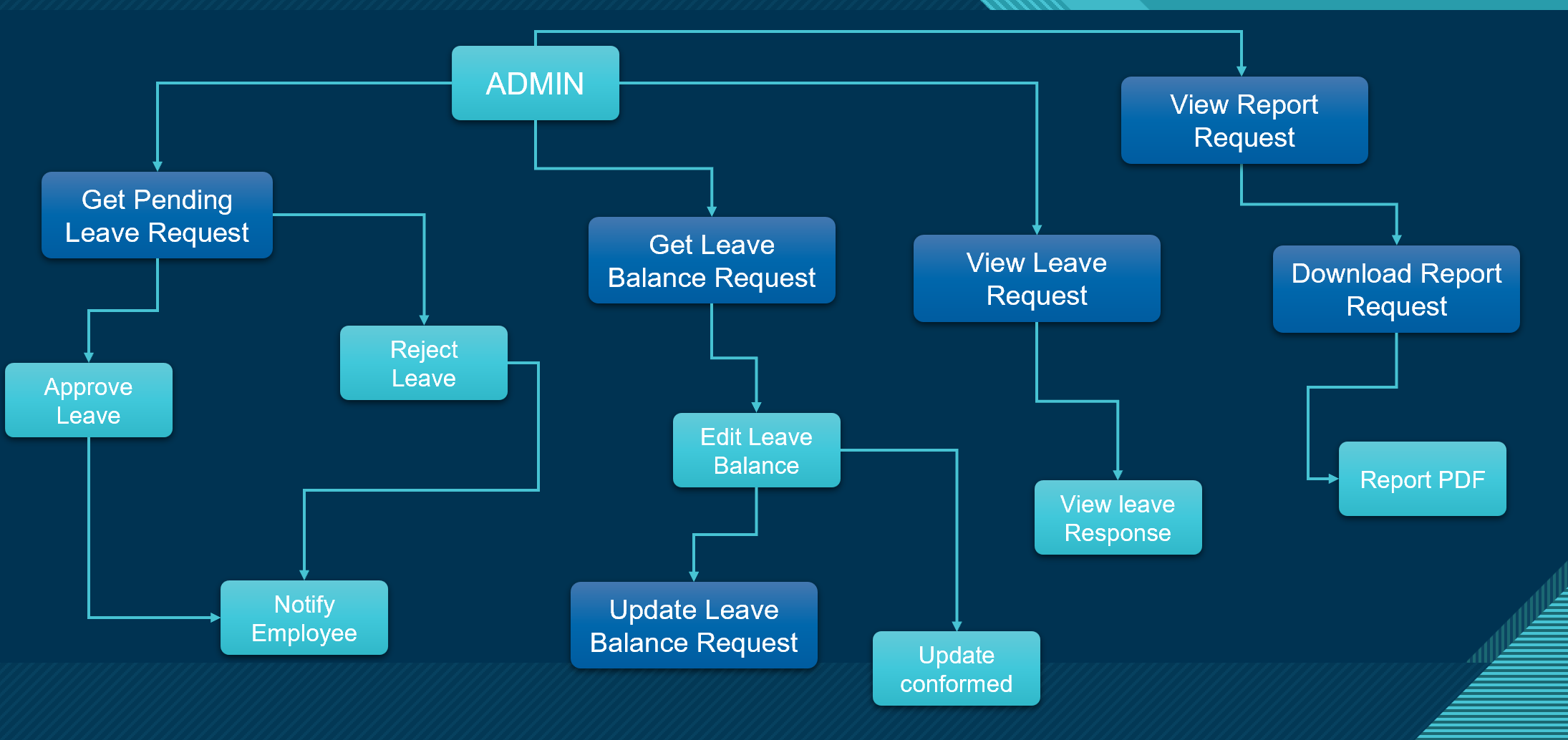
1. Employee Authentication & Authorization



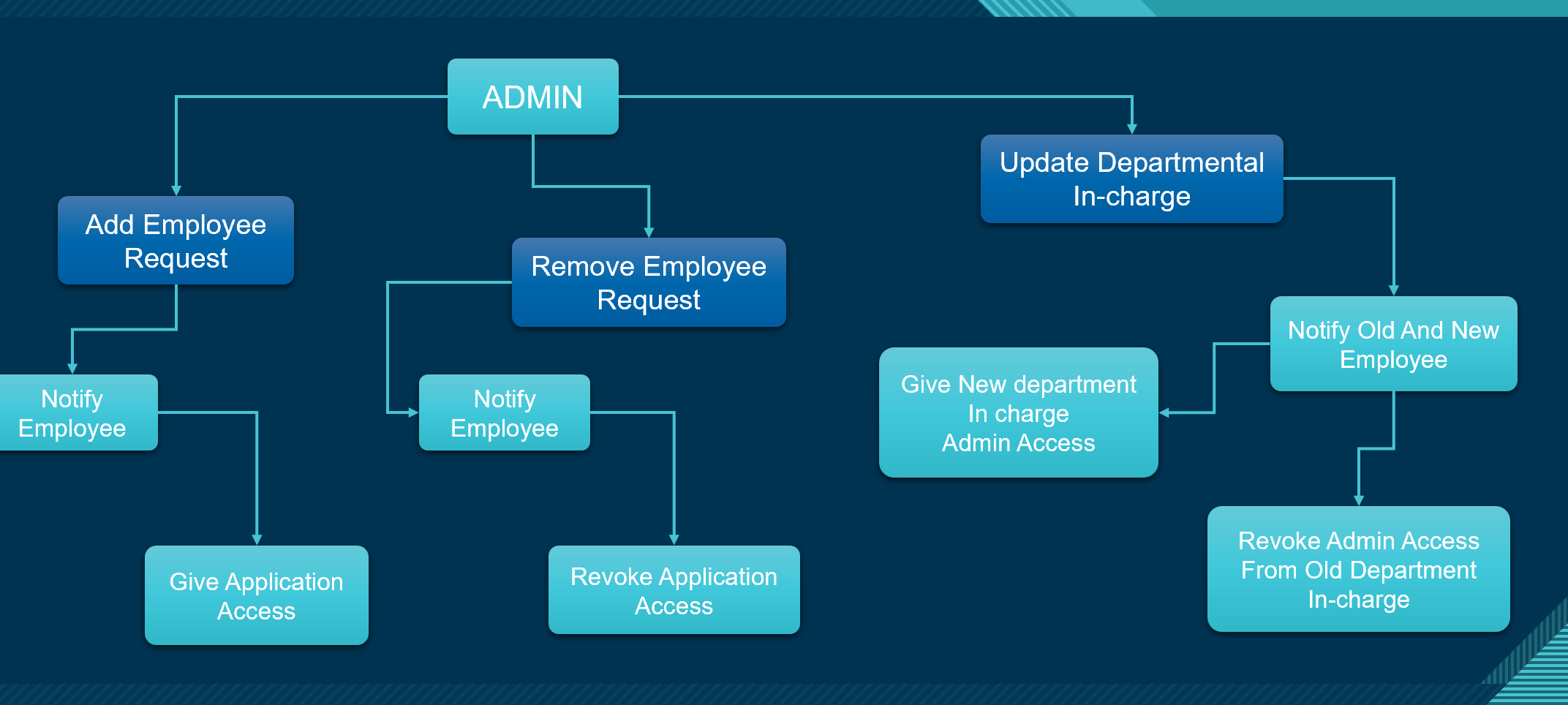
2. Leave Management



**3. Leave Status management**

****

**4. Employee Management**

****

**3.5 TESTING**

Generating comprehensive test cases for Pterodactyl game server management involves covering various aspects of the system, including user interactions, server configurations, and potential edge cases. Below are test cases organized into different categories:

### **1. User Management**

**1.1 Create User**

* **Test Case:** Verify user account creation.
  + - **Description:** Ensure that a new user can be created with valid details.
    - **Precondition:** None
    - **Test Steps:**
      * Navigate to the user registration page.
      * Enter valid user details (name, email, password, department, role, etc.).
      * Click on the 'Create' button.
    - **Expected Result:** User account should be created successfully and a confirmation message should be displayed.

**1.2 Login User**

* + **Test Case:** Verify user login functionality.
    - **Description:** Ensure that a user can log in with valid credentials.
    - **Precondition:** User account exists.
    - **Test Steps:**
      * Navigate to the login page.
      * Enter valid email and password.
      * Click on the 'Login' button.
    - **Expected Result:** User should be logged in and redirected to the dashboard.

**1.3 Delete User**

* **Test Case:** Verify user account deletion.
  + **Description:** Ensure that an existing user can be deleted.
  + **Precondition:** User account exists.
  + **Test Steps:**
    1. Navigate to the user management page.
    2. Select a user account.
    3. Click on the 'Delete' button.
    4. Confirm the deletion.
  + **Expected Result:** User account should be deleted and a confirmation message should be displayed.

**1.4 Update User**

* **Test Case:** Verify user account update.
  + **Description:** Ensure that an existing user can update their profile information.
  + **Precondition:** User account exists.
  + **Test Steps:**
    1. Navigate to the user profile page.
    2. Update profile information (name, email, department, etc.).
    3. Click on the 'Save' button.
  + **Expected Result:** User profile should be updated and a confirmation message should be displayed.

### **1. User Management**

**1.1 Create User**

* **Test Case:** Verify user account creation.
  + **Description:** Ensure that a new user can be created with valid details.
  + **Precondition:** None
  + **Test Steps:**
    1. Navigate to the user registration page.
    2. Enter valid user details (name, email, password, department, role, etc.).
    3. Click on the 'Create' button.
  + **Expected Result:** User account should be created successfully and a confirmation message should be displayed.

**1.2 Login User**

* **Test Case:** Verify user login functionality.
  + **Description:** Ensure that a user can log in with valid credentials.
  + **Precondition:** User account exists.
  + **Test Steps:**
    1. Navigate to the login page.
    2. Enter valid email and password.
    3. Click on the 'Login' button.
  + **Expected Result:** User should be logged in and redirected to the dashboard.

**1.3 Delete User**

* **Test Case:** Verify user account deletion.
  + **Description:** Ensure that an existing user can be deleted.
  + **Precondition:** User account exists.
  + **Test Steps:**
    1. Navigate to the user management page.
    2. Select a user account.
    3. Click on the 'Delete' button.
    4. Confirm the deletion.
  + **Expected Result:** User account should be deleted and a confirmation message should be displayed.

**1.4 Update User**

* **Test Case:** Verify user account update.
  + **Description:** Ensure that an existing user can update their profile information.
  + **Precondition:** User account exists.
  + **Test Steps:**
    1. Navigate to the user profile page.
    2. Update profile information (name, email, department, etc.).
    3. Click on the 'Save' button.
  + **Expected Result:** User profile should be updated and a confirmation message should be displayed.

### **2. Leave Management**

**2.1 Apply Leave**

* **Test Case:** Verify leave application submission.
  + **Description:** Ensure that a user can apply for leave.
  + **Precondition:** User is logged in.
  + **Test Steps:**
    1. Navigate to the leave application page.
    2. Select leave type, start date, end date, and reason for leave.
    3. Click on the 'Apply' button.
  + **Expected Result:** Leave application should be submitted successfully, and a confirmation message should be displayed.

**2.2 Edit Leave**

* **Test Case:** Verify leave application editing.
  + **Description:** Ensure that a user can edit a submitted leave application.
  + **Precondition:** Leave application exists.
  + **Test Steps:**
    1. Navigate to the user's leave history.
    2. Select the leave application to edit.
    3. Update the leave details.
    4. Click on the 'Save' button.
  + **Expected Result:** Leave application should be updated and a confirmation message should be displayed.

**2.3 Approve Leave**

* **Test Case:** Verify leave approval process.
  + **Description:** Ensure that a manager can approve a leave application.
  + **Precondition:** Leave application is pending approval.
  + **Test Steps:**
    1. Navigate to the leave approval page.
    2. Select the leave application to approve.
    3. Click on the 'Approve' button.
  + **Expected Result:** Leave application should be approved, and the status should change to 'Approved'.

**2.4 Reject Leave**

* **Test Case:** Verify leave rejection process.
  + **Description:** Ensure that a manager can reject a leave application.
  + **Precondition:** Leave application is pending approval.
  + **Test Steps:**
    1. Navigate to the leave approval page.
    2. Select the leave application to reject.
    3. Click on the 'Reject' button.
  + **Expected Result:** Leave application should be rejected, and the status should change to 'Rejected'.

### **3. Edit Management**

**3.1 Edit User Management**

* **Test Case:** Verify the editing of user roles and permissions.
  + **Description:** Ensure that an admin can edit user roles and permissions.
  + **Precondition:** Admin is logged in.
  + **Test Steps:**
    1. Navigate to the user management page.
    2. Select a user.
    3. Update the user's role and permissions.
    4. Click on the 'Save' button.
  + **Expected Result:** User role and permissions should be updated successfully.

**3.2 Edit Leave Balance Management**

* **Test Case:** Verify the editing of leave balances.
  + **Description:** Ensure that an admin can edit user leave balances.
  + **Precondition:** Admin is logged in.
  + **Test Steps:**
    1. Navigate to the leave balance management page.
    2. Select a user.
    3. Update the leave balance.
    4. Click on the 'Save' button.
  + **Expected Result:** Leave balance should be updated successfully.

**3.3 Edit Departmental In charge Management**

* **Test Case:** Verify the editing of departmental in-charge information.
  + **Description:** Ensure that an admin can assign or update departmental in-charge information.
  + **Precondition:** Admin is logged in.
  + **Test Steps:**
    1. Navigate to the departmental management page.
    2. Select a department.
    3. Assign or update the departmental in-charge.
    4. Click on the 'Save' button.
  + **Expected Result:** Departmental in-charge information should be updated successfully.

### **4. Viewing and Reporting**

**4.1 View Leave**

* **Test Case:** Verify the functionality of viewing leave history.
  + **Description:** Ensure that a user can view their leave history.
  + **Precondition:** User is logged in.
  + **Test Steps:**
    1. Navigate to the leave history page.
  + **Expected Result:** User should see a list of all their leave applications with statuses.

**4.2 View Report**

* **Test Case:** Verify the functionality of viewing reports.
  + **Description:** Ensure that an admin can view leave reports.
  + **Precondition:** Admin is logged in.
  + **Test Steps:**
    1. Navigate to the reports page.
    2. Select the desired report criteria (date range, department, etc.).
    3. Click on the 'Generate Report' button.
  + **Expected Result:** The system should generate the report based on the selected criteria.

**4.3 Download Report**

* **Test Case:** Verify the functionality of downloading reports.
  + **Description:** Ensure that a user can download generated reports.
  + **Precondition:** Report is generated.
  + **Test Steps:**
    1. Navigate to the reports page.
    2. Generate a report.
    3. Click on the 'Download' button.
  + **Expected Result:** The report should be downloaded in the selected format (PDF).

### **User Management Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** |
| Create User | User details (name, email, password, department, role, etc.) | User account created successfully; confirmation message displayed |
| Login User | Valid email and password | User logged in, redirected to dashboard |
| Delete User | Select user account, click 'Delete', confirm deletion | User account deleted; confirmation message displayed |
| Update User | Update user details (name, email, department, etc.), click 'Save' | User profile updated; confirmation message displayed |

### **Leave Management Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** |
| Create User | User details (name, email, password, department, role, etc.) | User account created successfully; confirmation message displayed |
| Login User | Valid email and password | User logged in, redirected to dashboard |
| Delete User | Select user account, click 'Delete', confirm deletion | User account deleted; confirmation message displayed |
| Update User | Update user details (name, email, department, etc.), click 'Save' | User profile updated; confirmation message displayed |

**Edit Management Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** |
| Edit User Management | |  | | --- | | Select user, update role and permissions, click 'Save' |  |  | | --- | |  | | User role and permissions updated successfully |
| Edit Leave Balance Management | User role and permissions updated successfully | Leave balance updated successfully |
| Edit Departmental In charge | Select department, assign/update in-charge, click 'Save' | Departmental in-charge information updated successfully |

**Viewing and Reporting Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** |
| View Leave | Navigate to leave history page | List of leave applications with statuses displayed |
| View Report | Select report criteria (date range, department, etc.), click 'Generate Report' | Report generated based on selected criteria |
| Download Report | Generate report, click 'Download' | Report downloaded in selected format (PDF) |

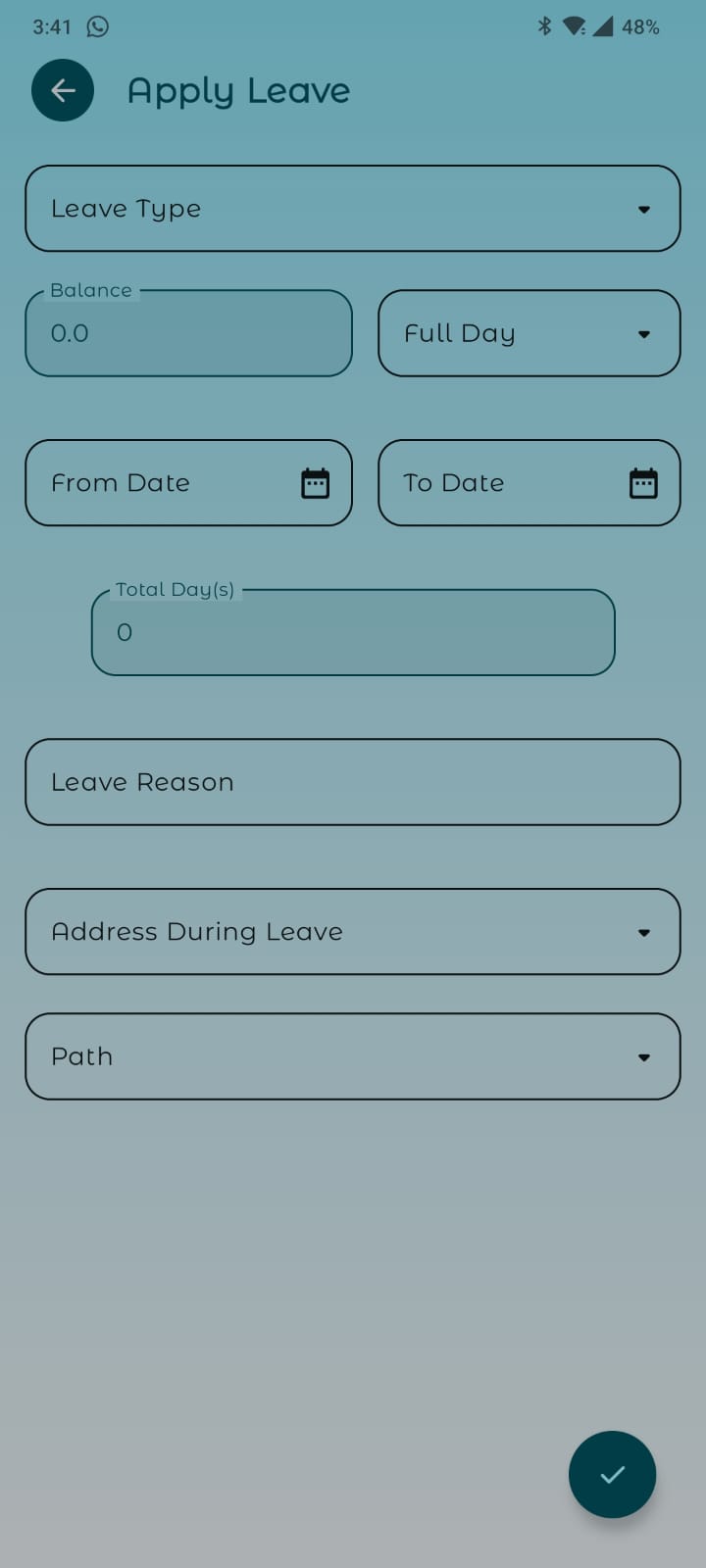
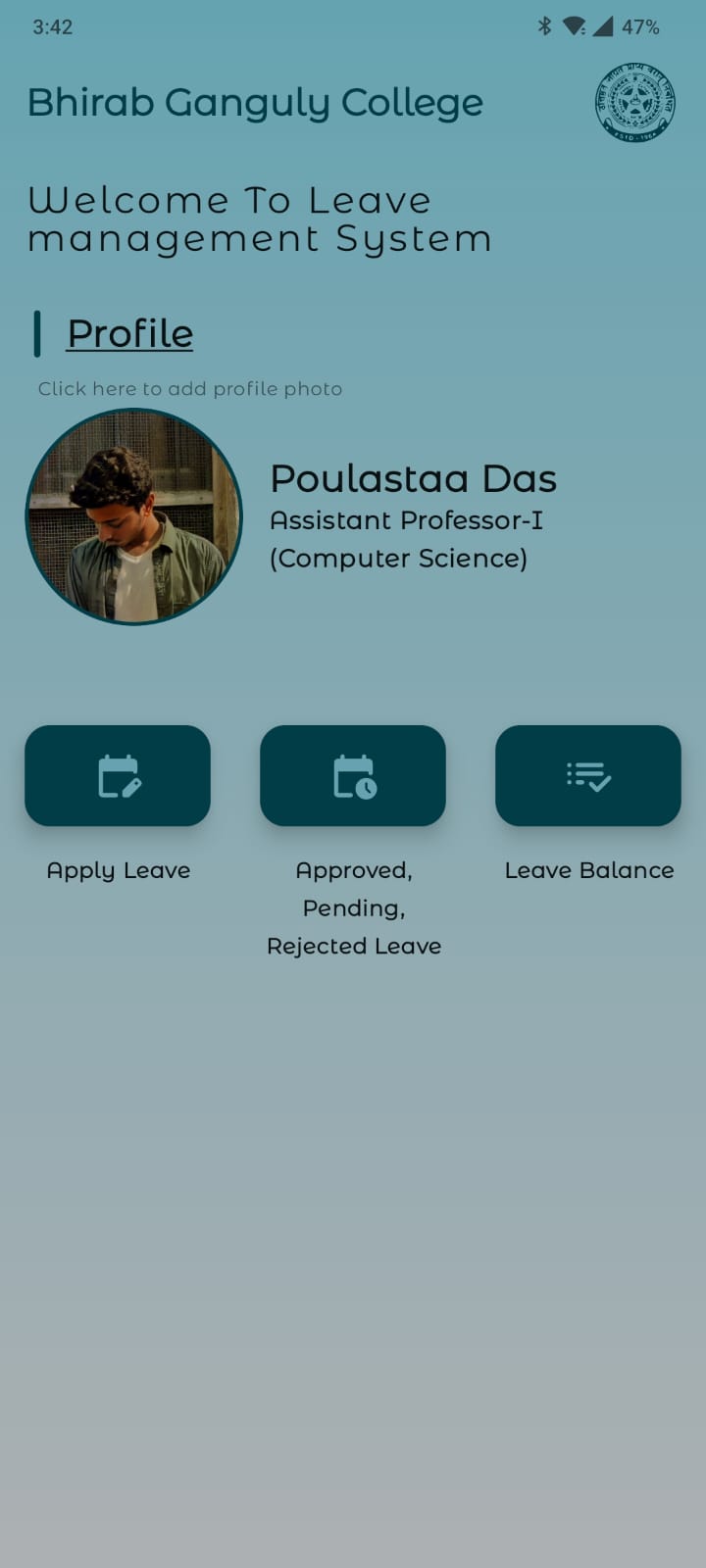
**3.6 SCREENSHOTS**

**Sac Teacher**

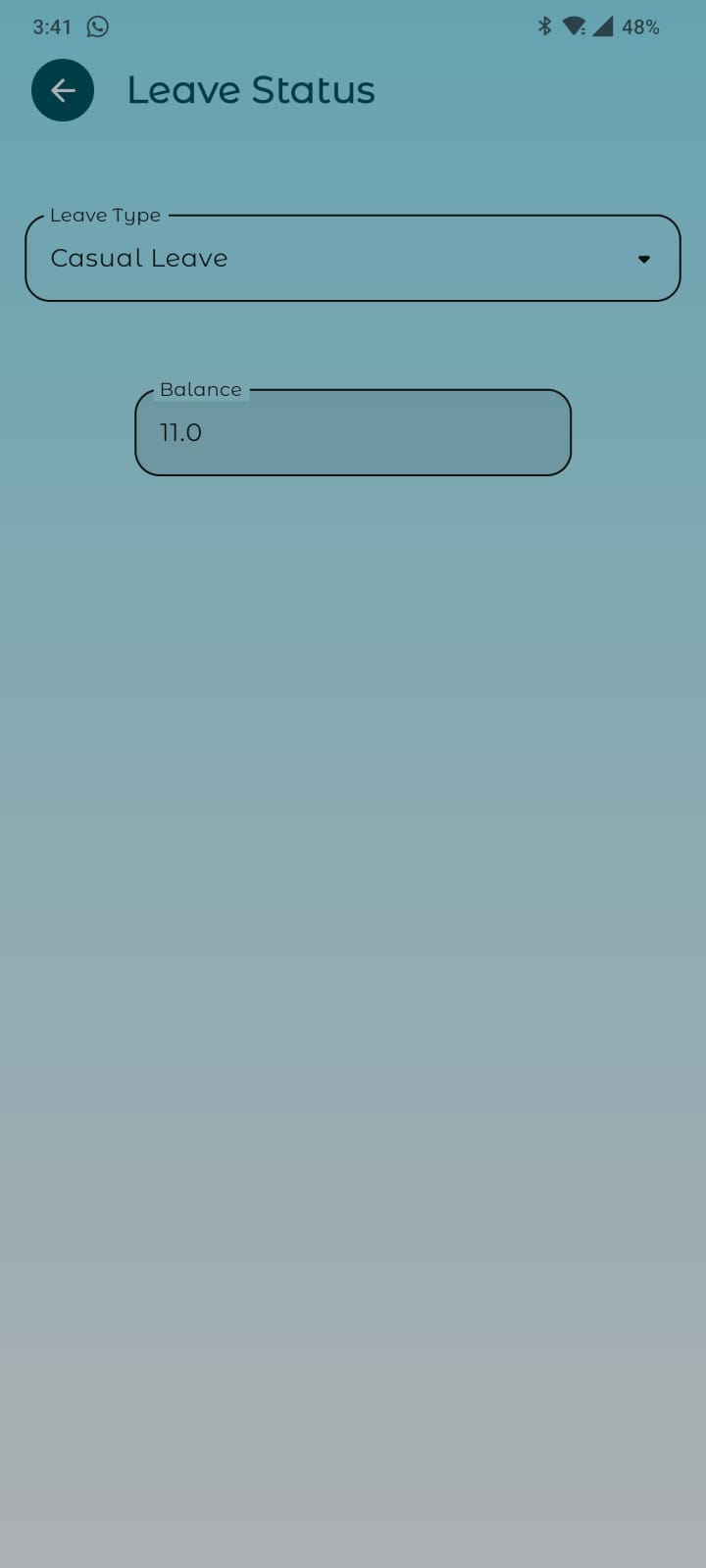
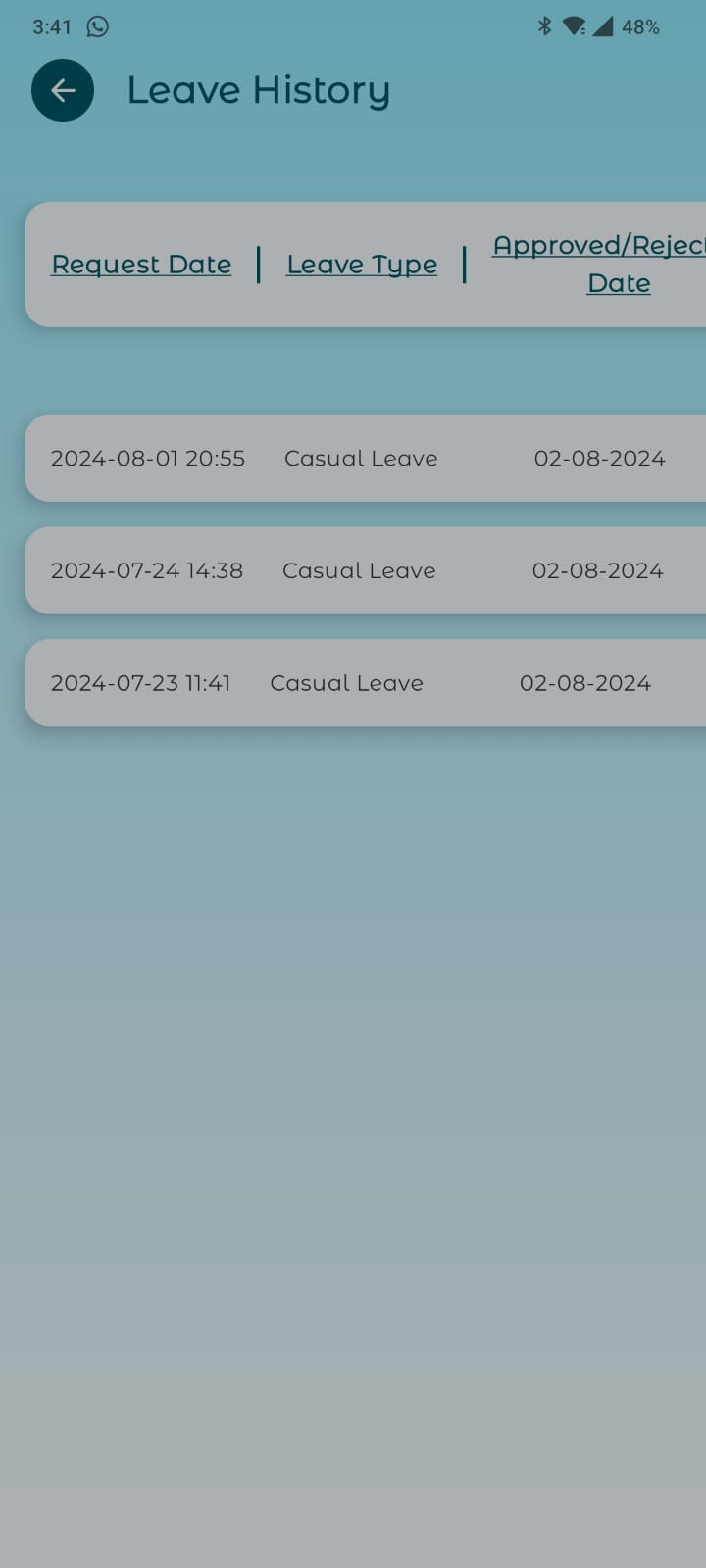
|  |  |
| --- | --- |
| **1. Login** | **2. Profile** |
|  |  |
| **3.Apply leave** | **4. Leave Balance** |
|  |  |
| **Leave history** | **SACT profile** |
| sact eave |  |

**CLIENT (Permanente Teacher)**

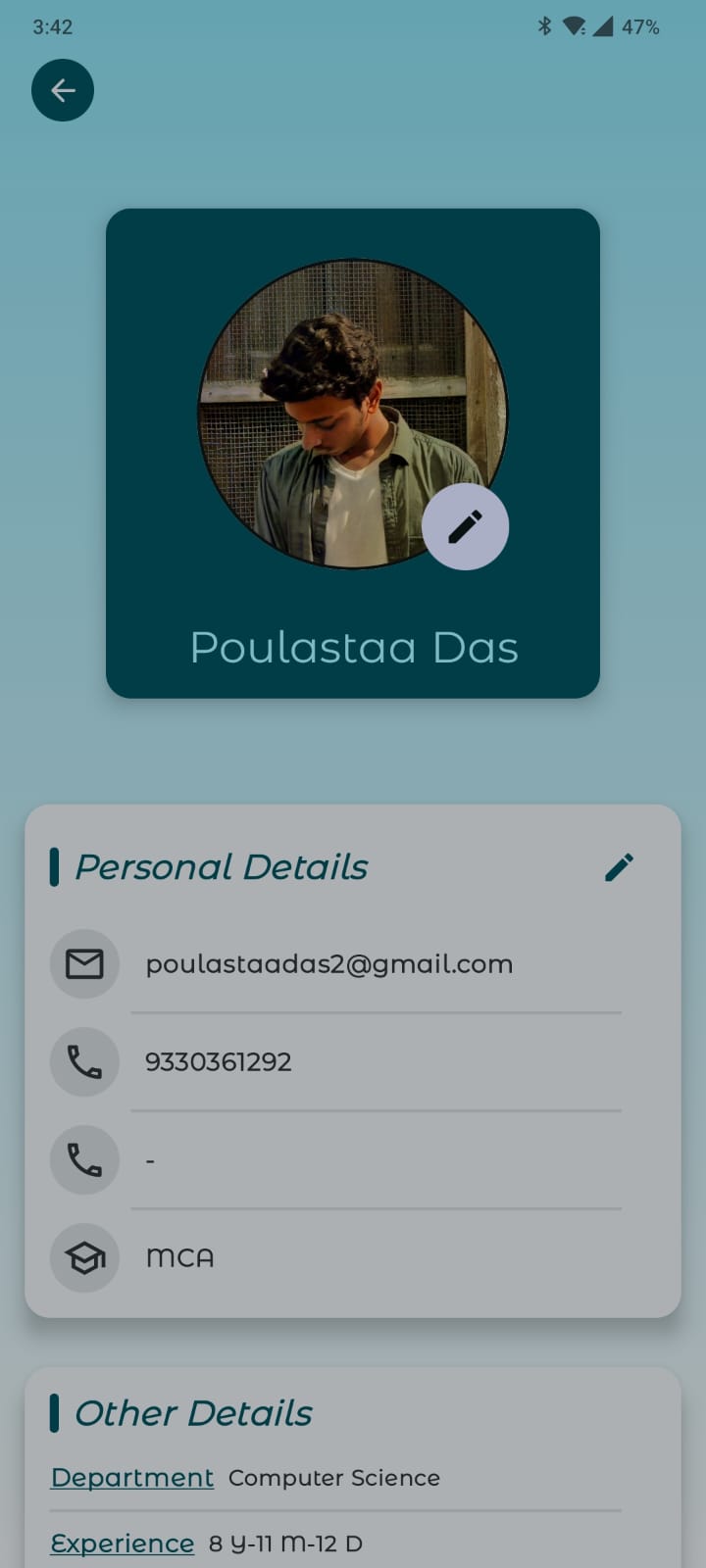
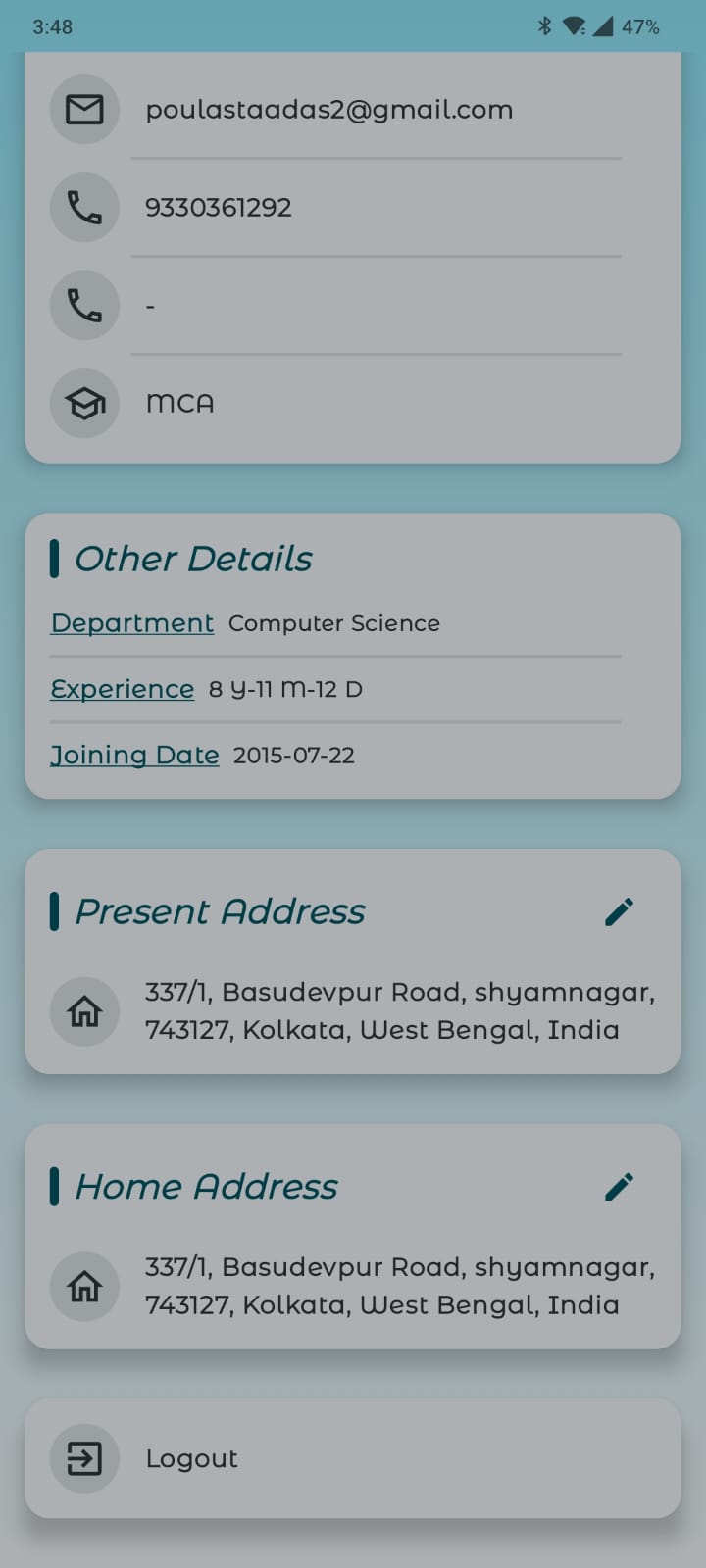
**1.Employee Dashboard 2. Employee Apply Leave**



**3.Employee View Leave Balance 4. Employee Leave History**

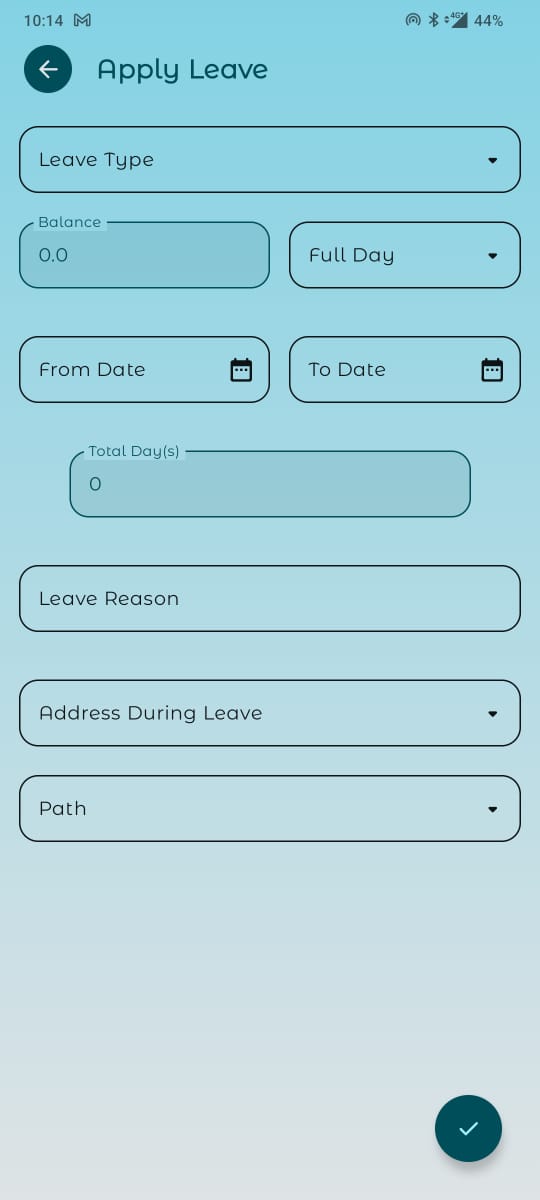
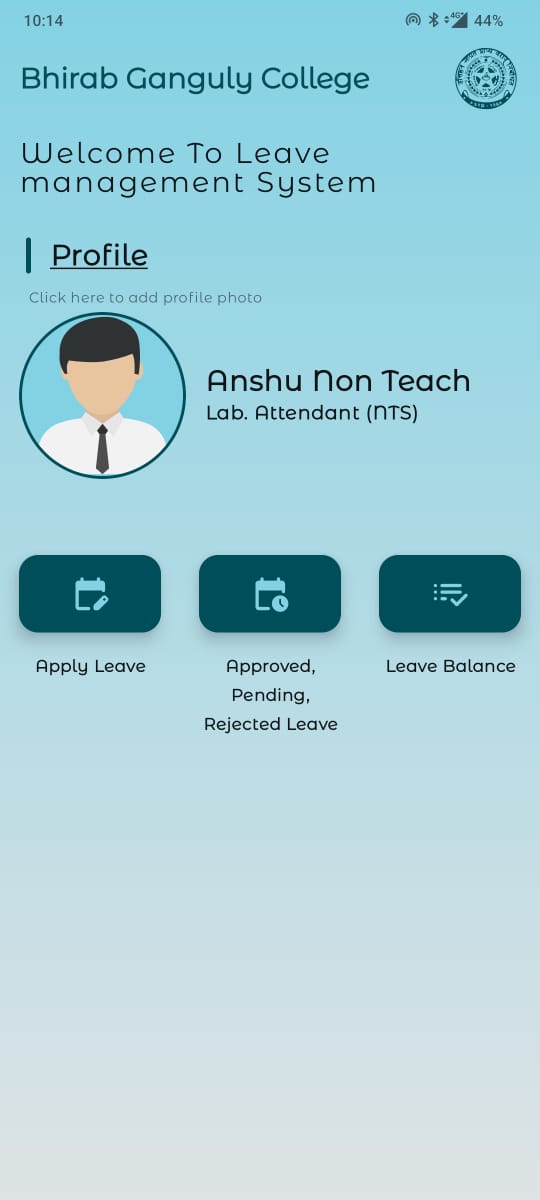
 

**5. Employee profile**

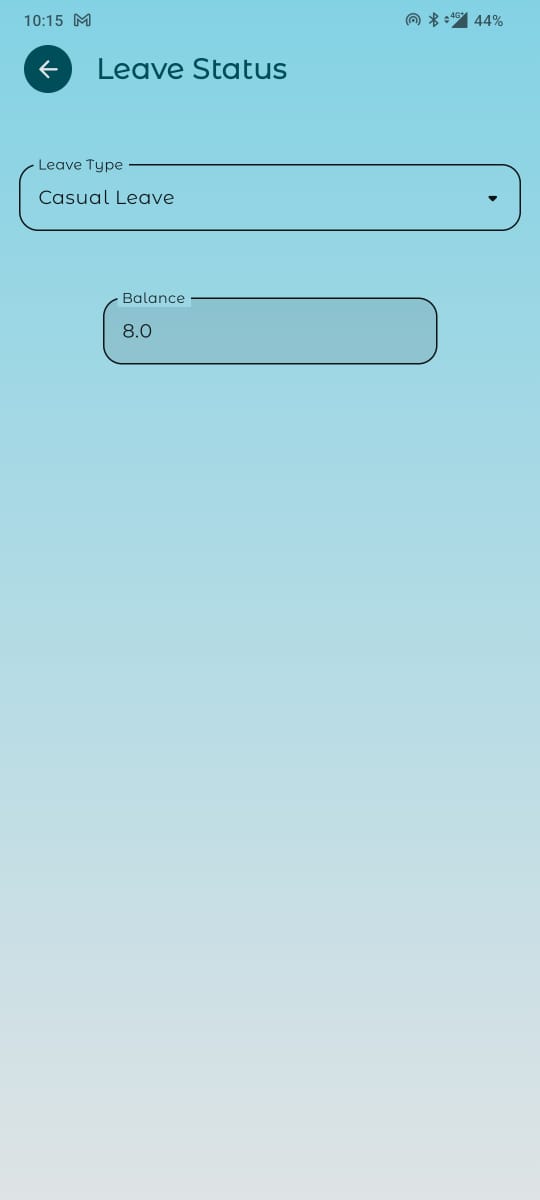
 ****

**Client Non-Teaching Staff**

**1. Profile 2. Apply leave**

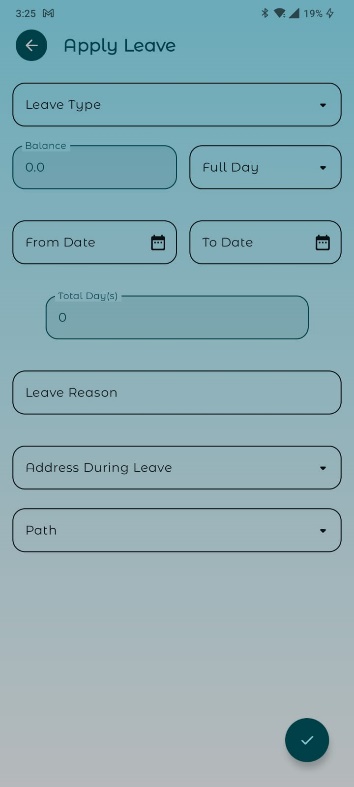
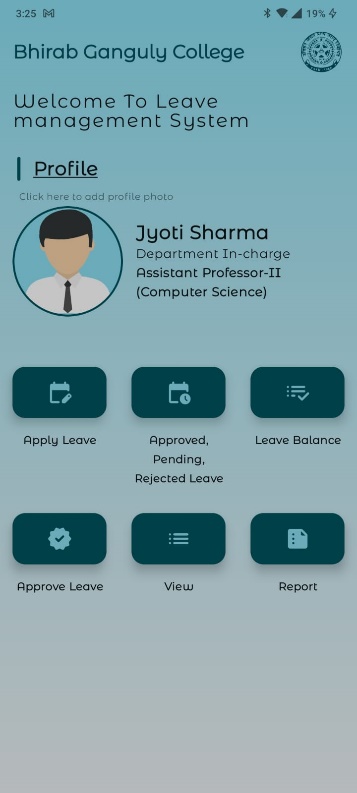


**3. Leave History 4. View Leave balance**

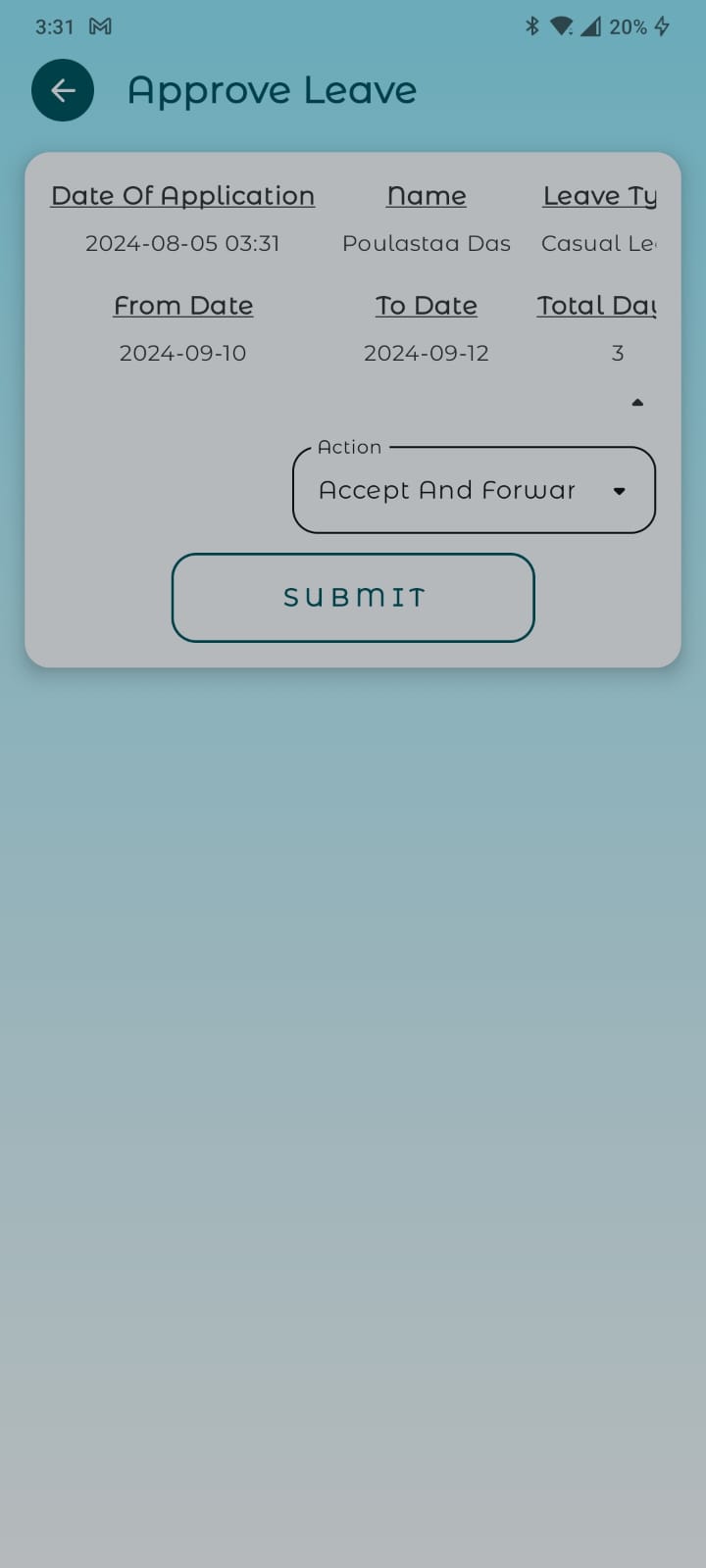
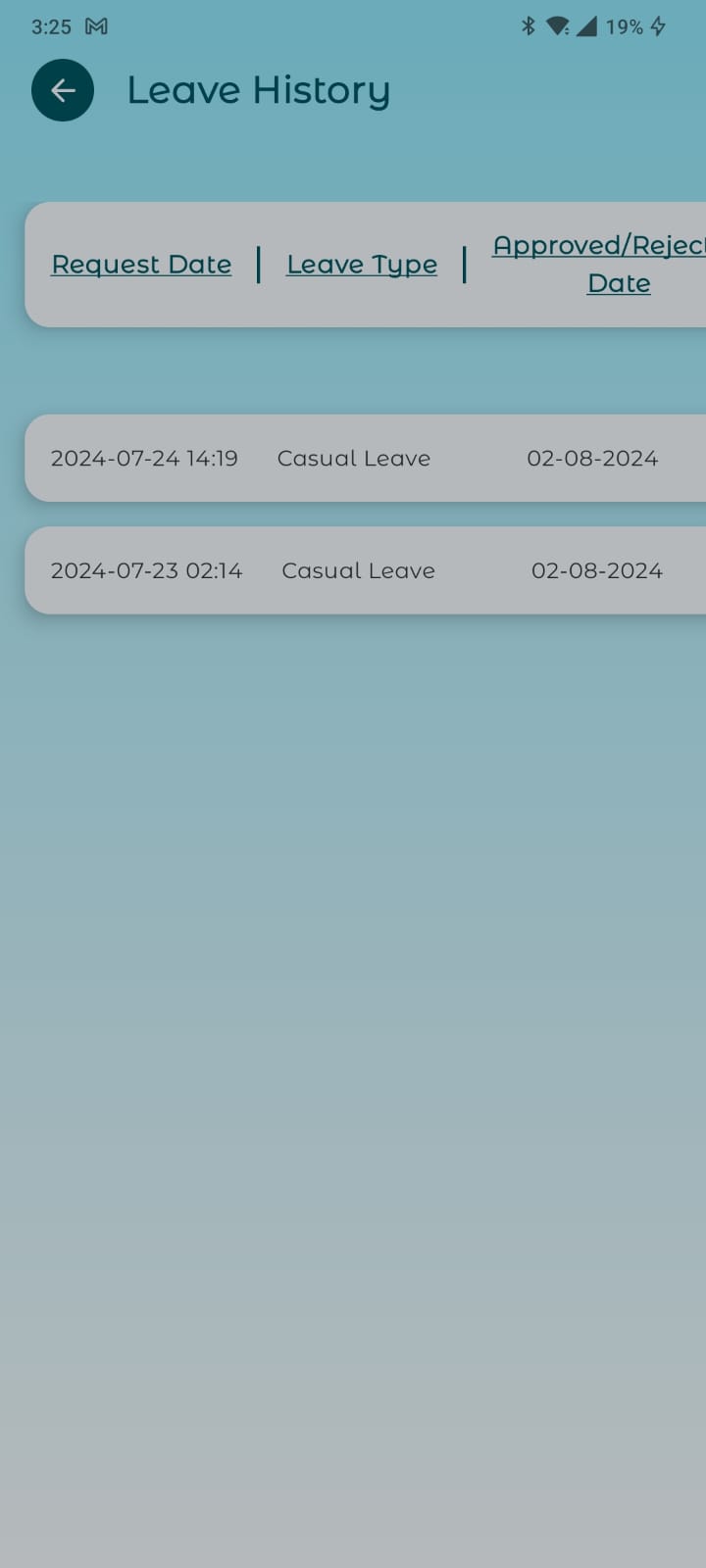


**Admin (HOD)**

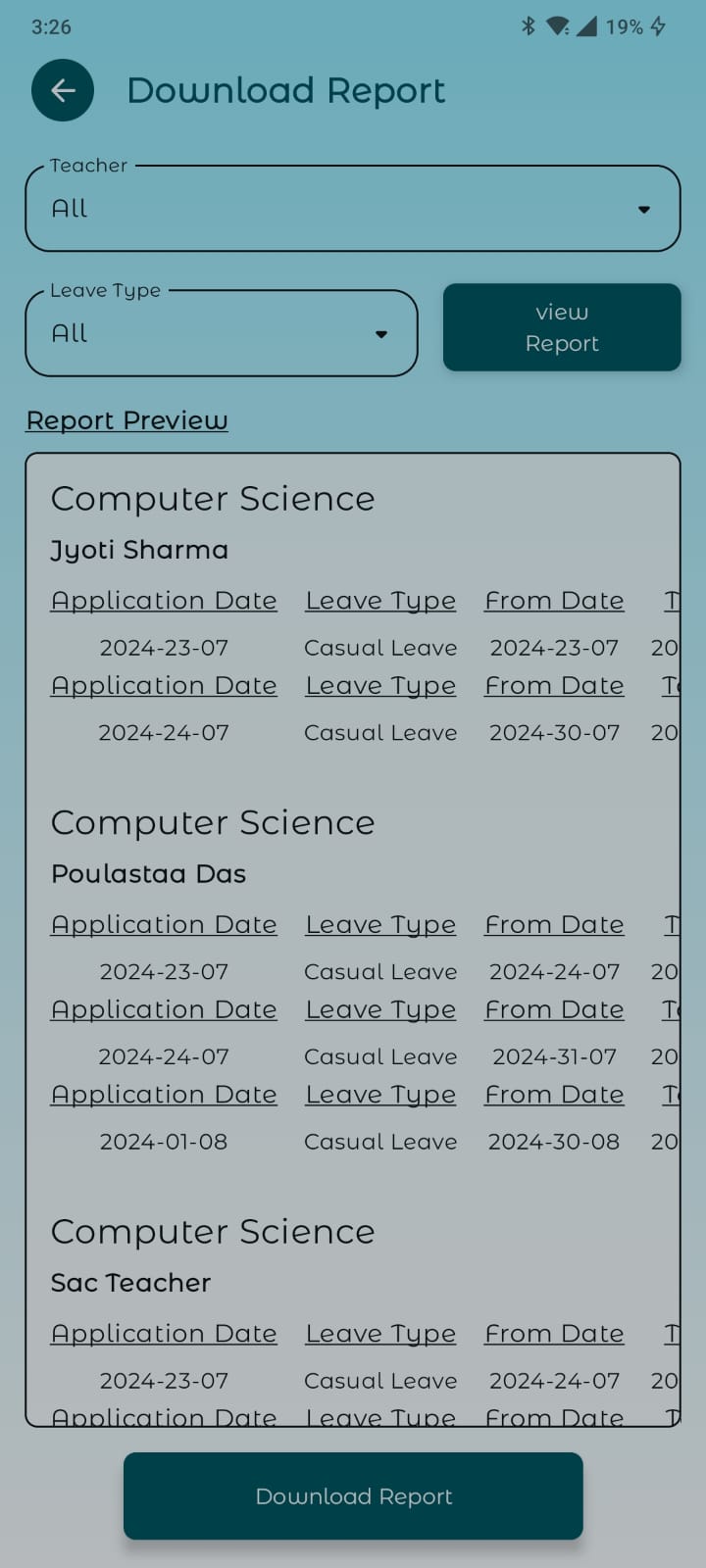
**1. Dashboard 2. Apply leave**



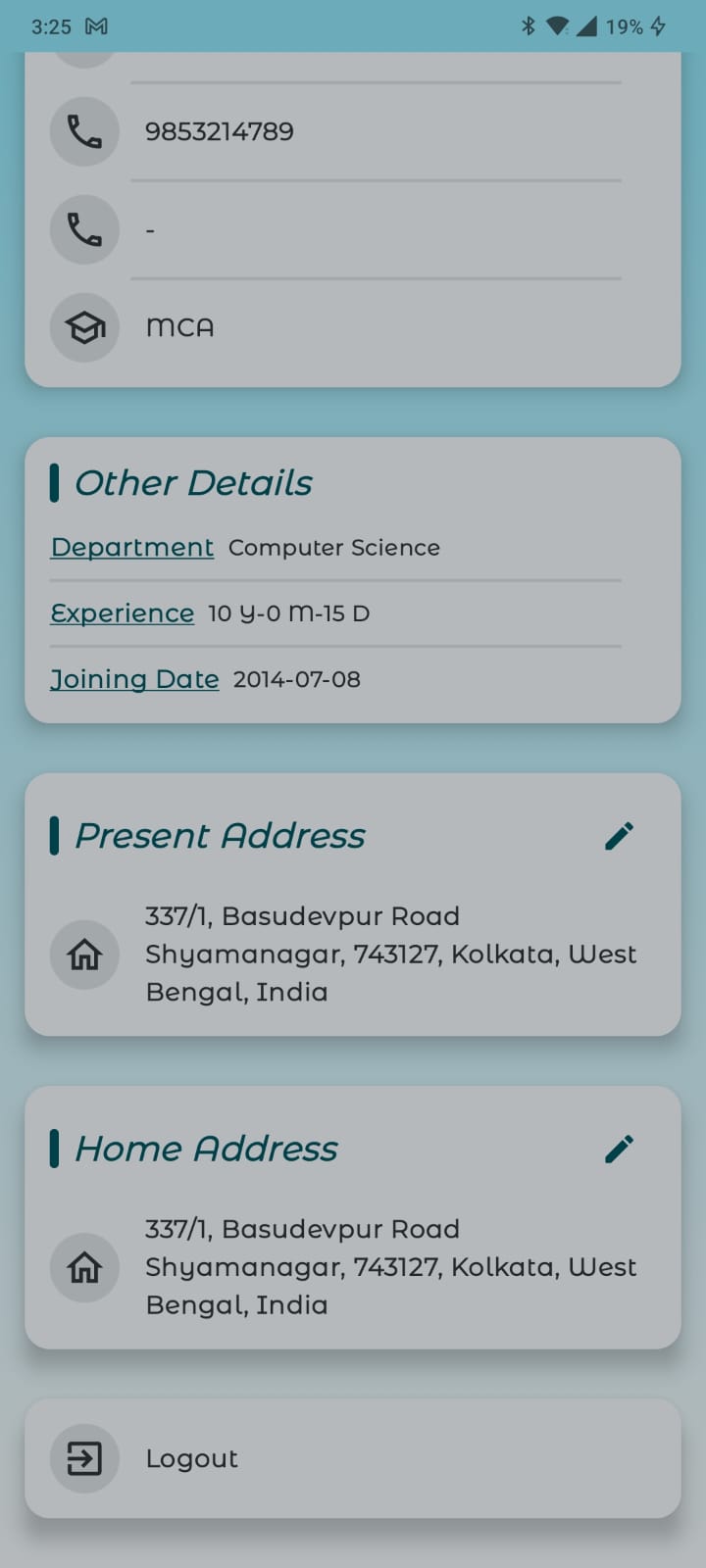
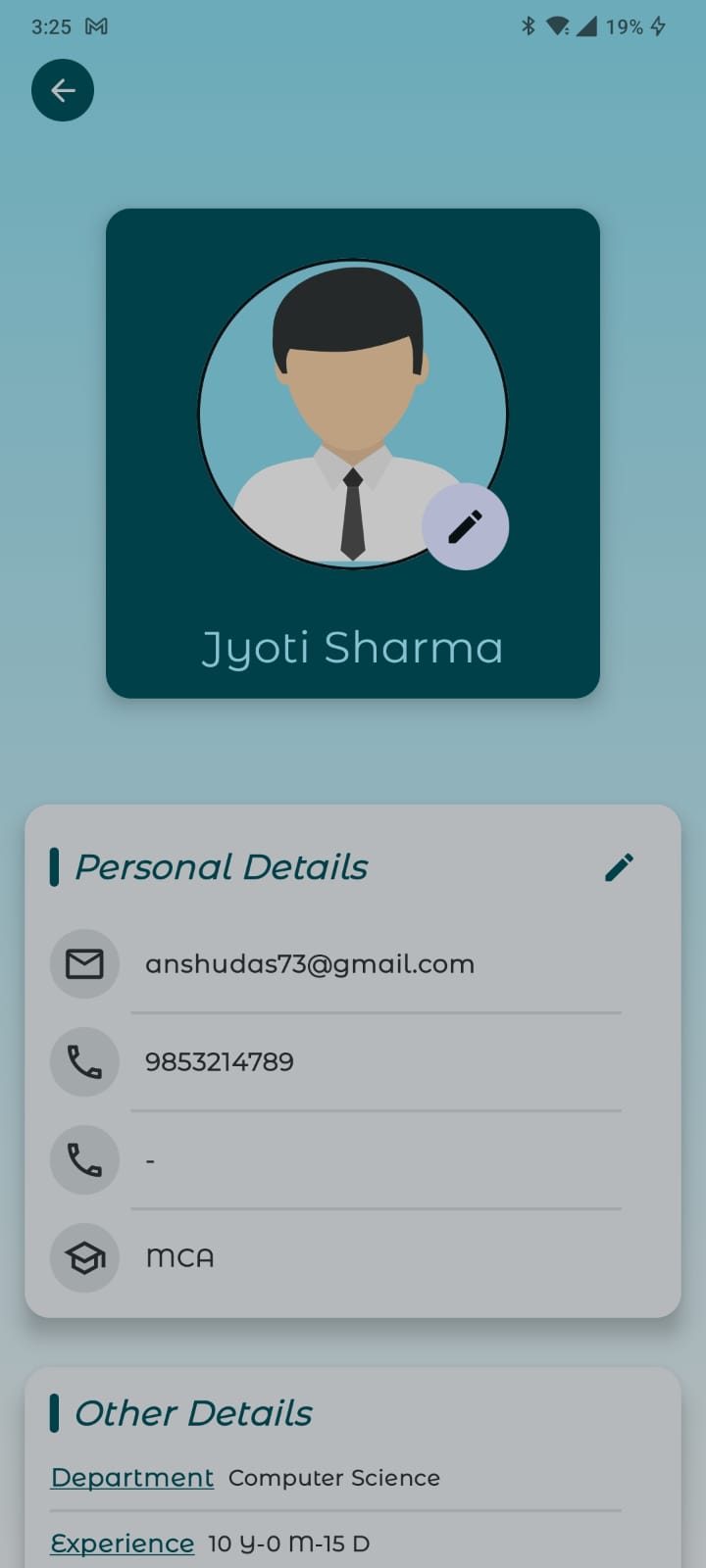
**3. Leave History 4. Approve/Reject Leave**



**6. Approved/Rejected Leave 7. View/Download Report**

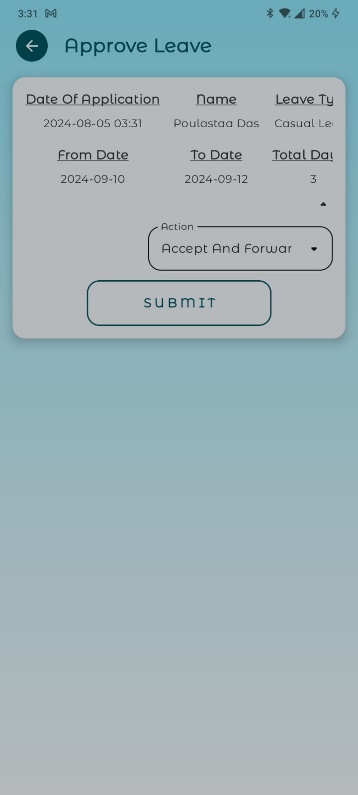
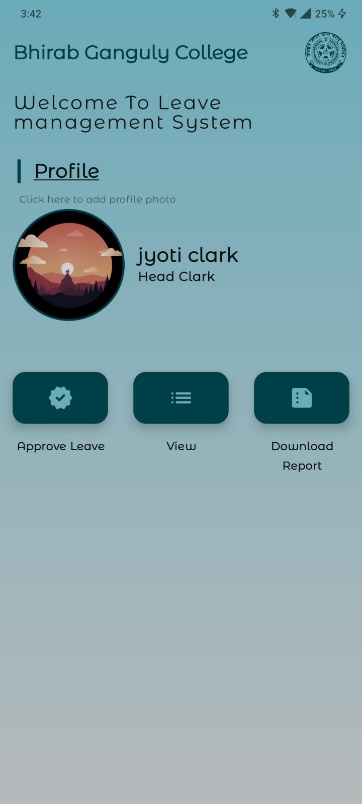


**8. Profile**

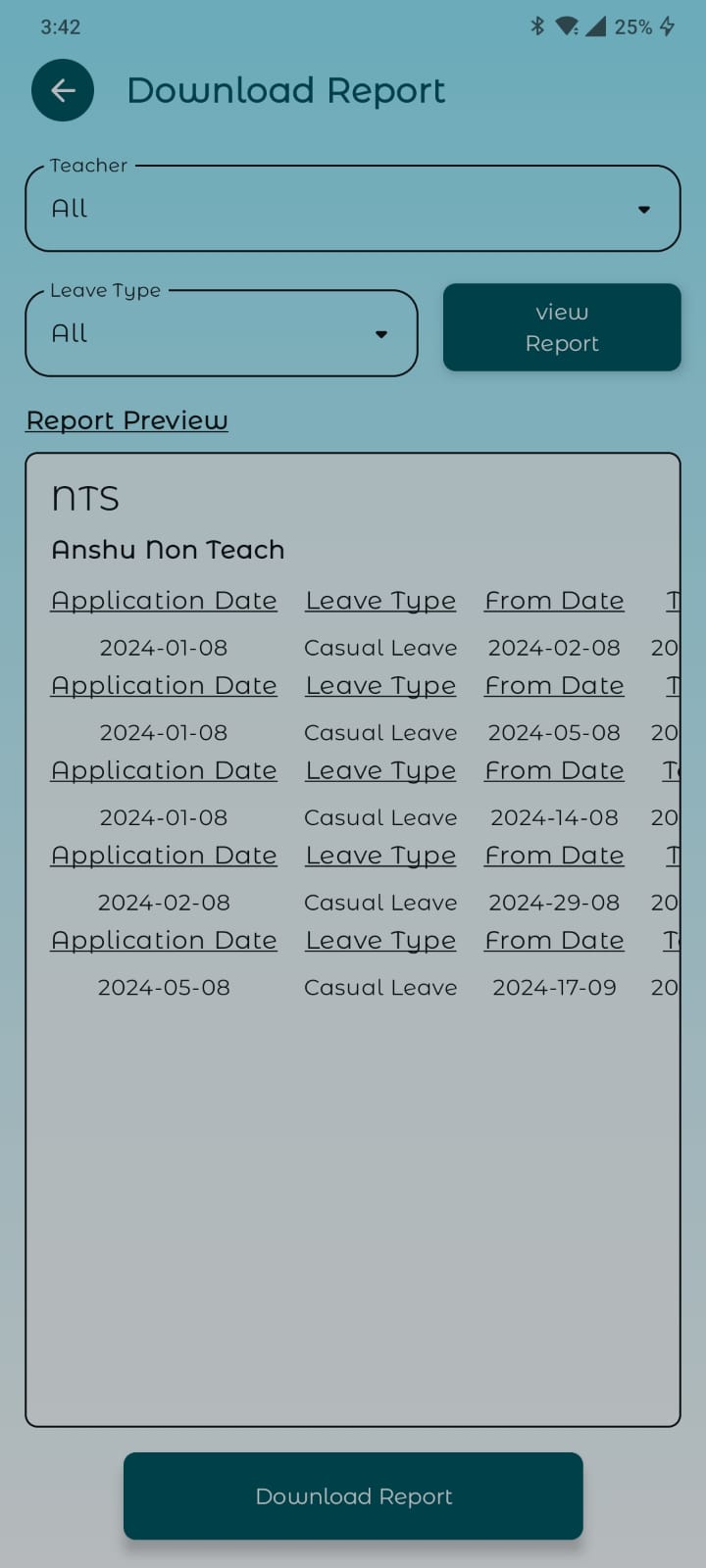
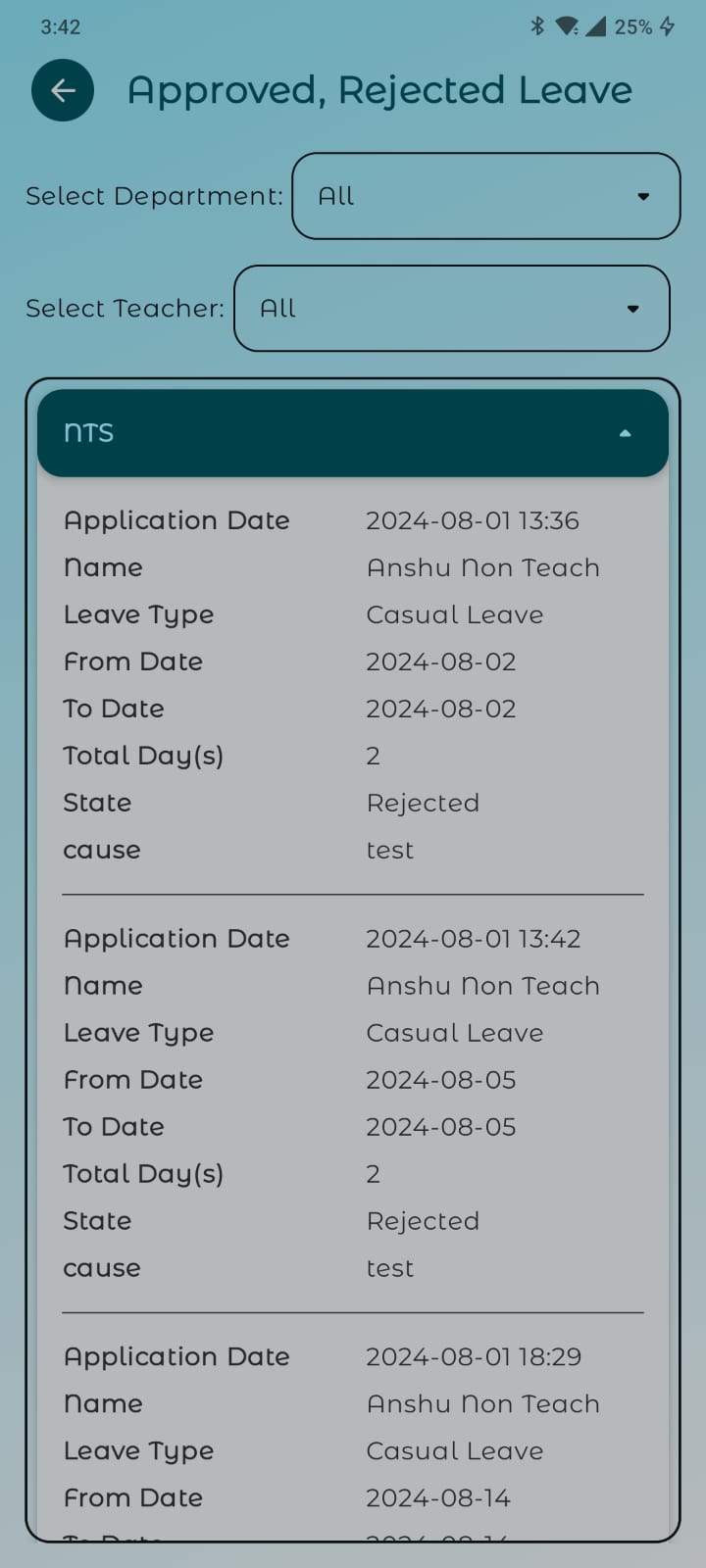


**Head Clark**

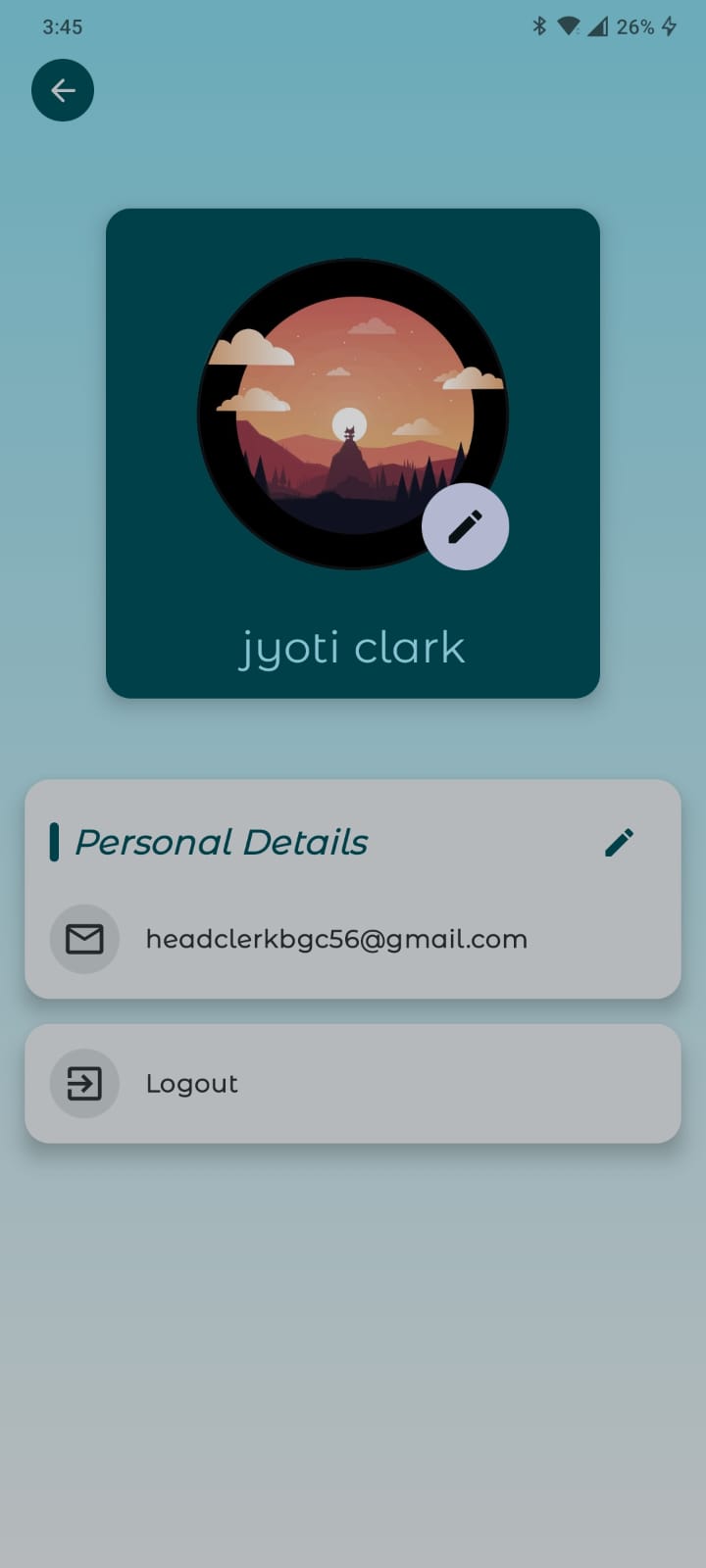
**1. Dashboard Approve/Reject Leave**



**3. Approved/Rejected Leave 4. Download Report**

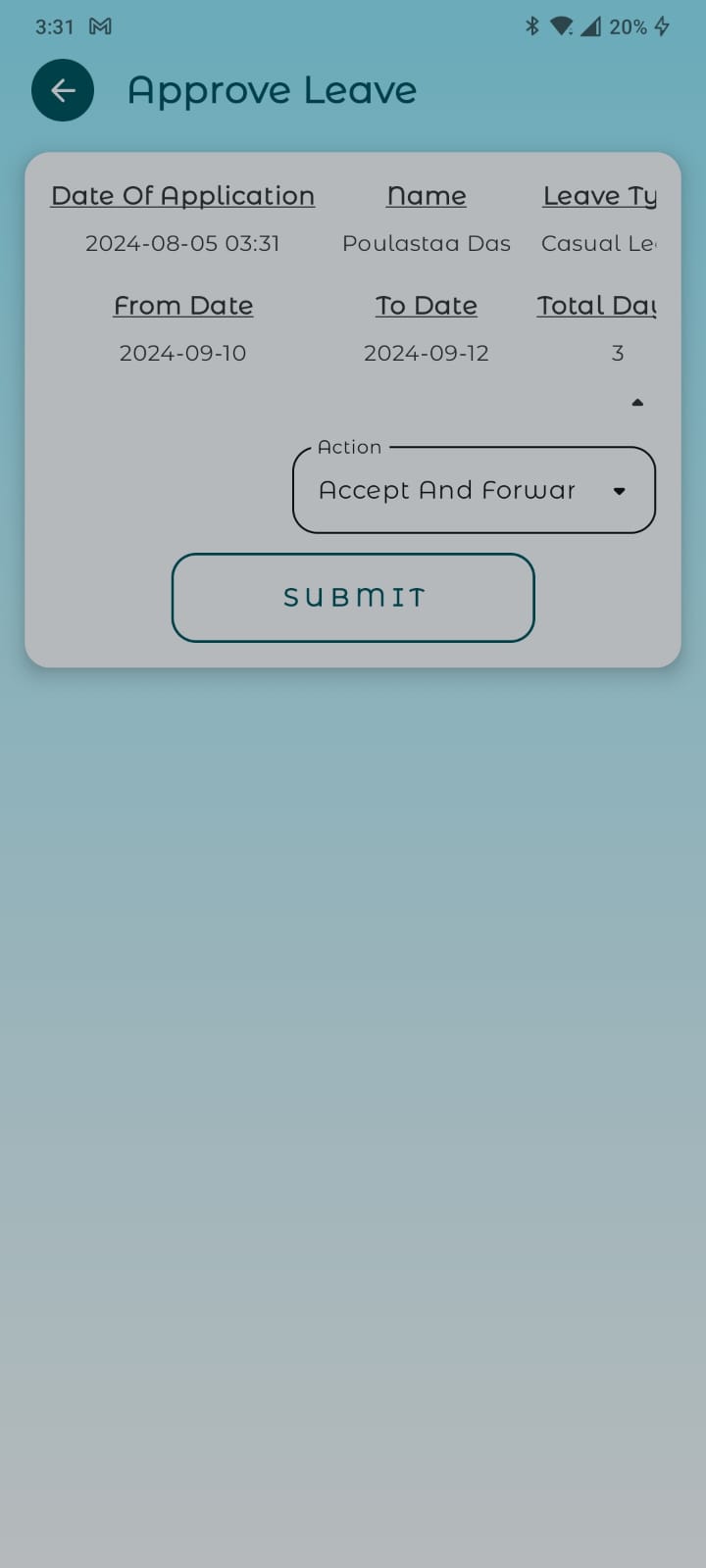
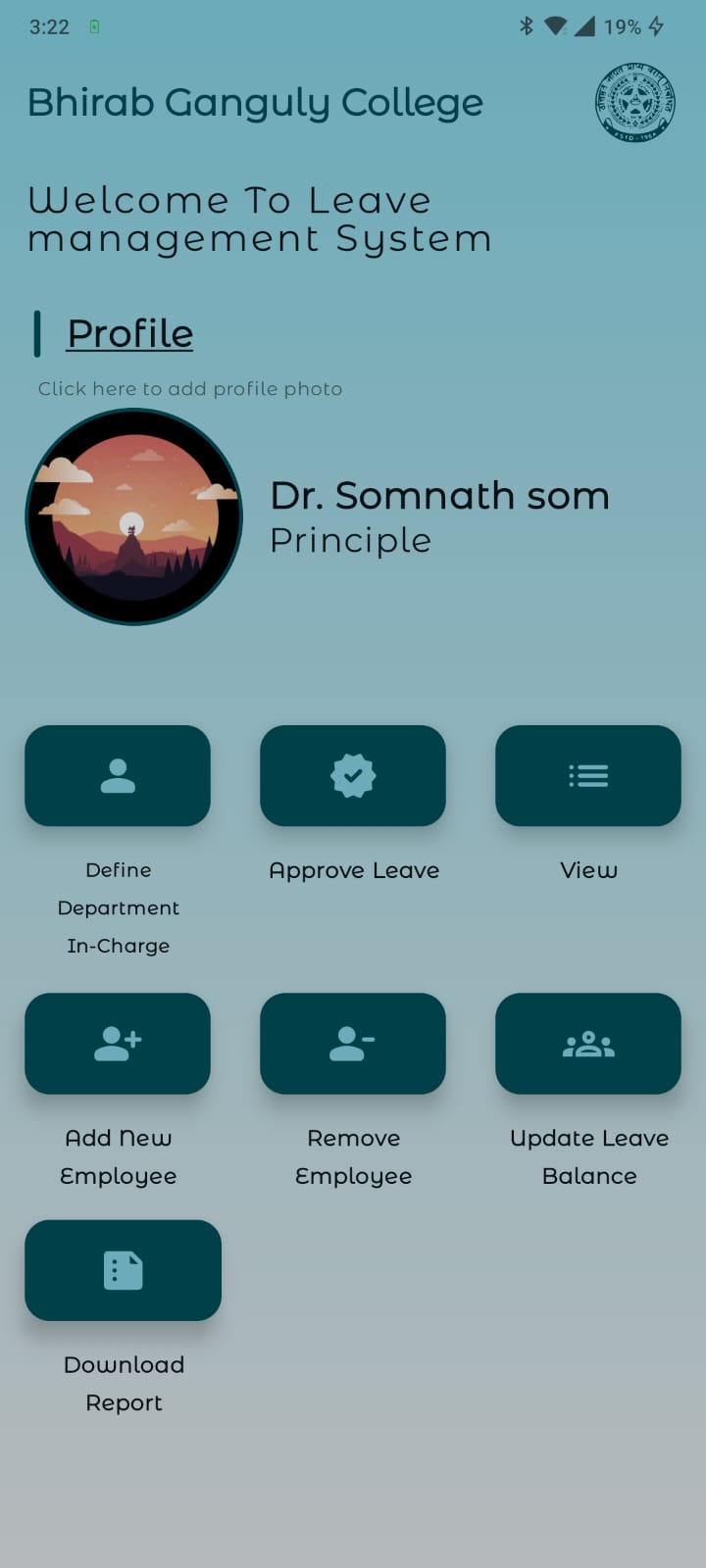


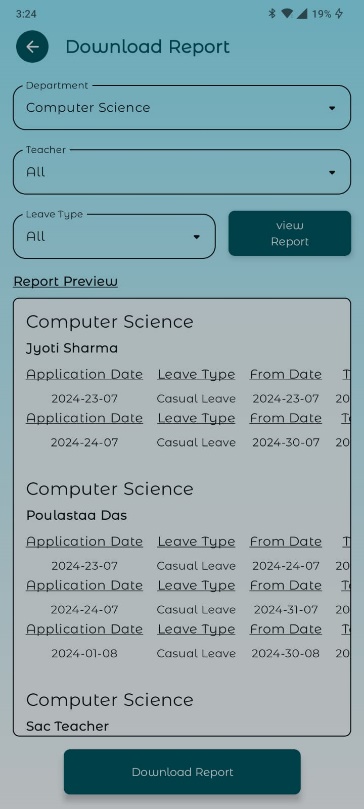
**5. Profile**



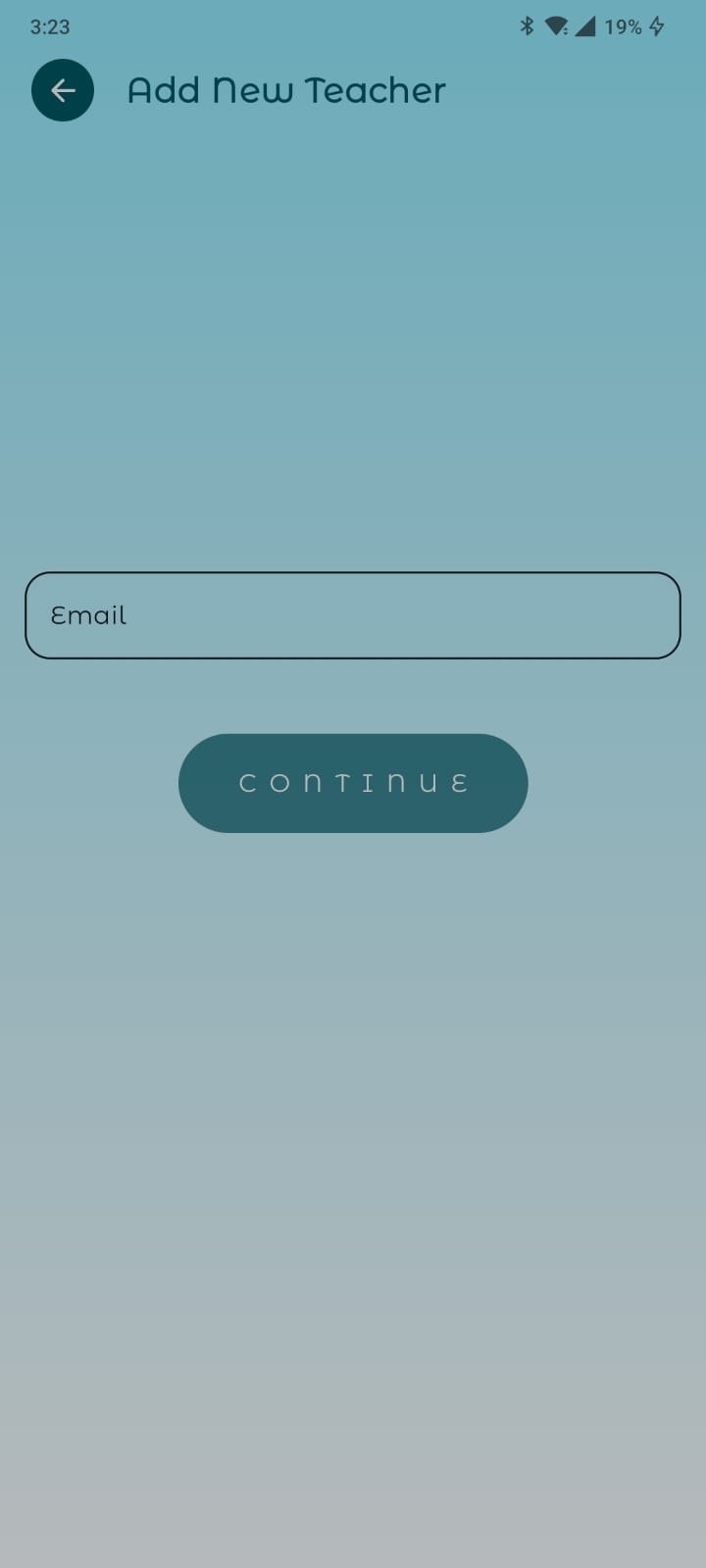
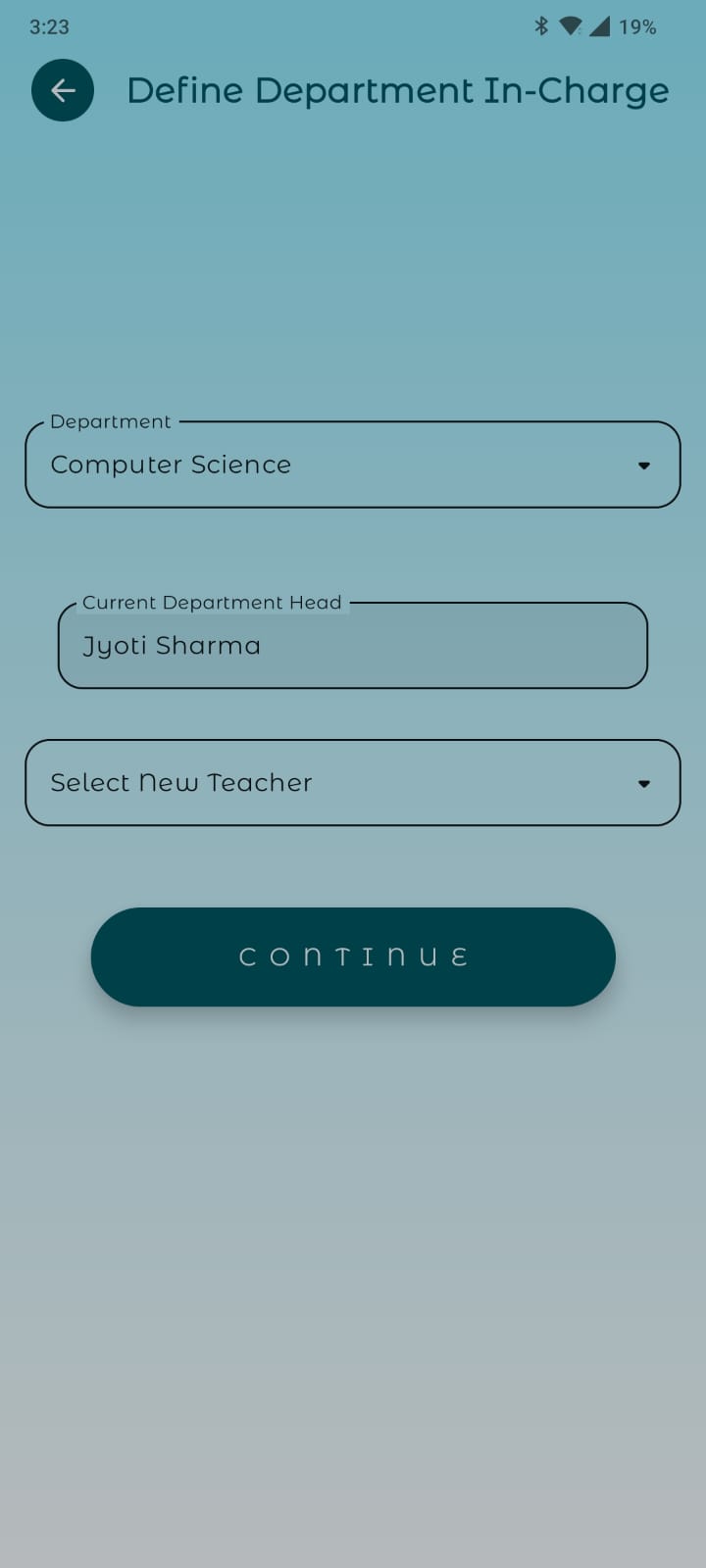
**Principal**

**1. Dashboard Approve/Reject Leave**

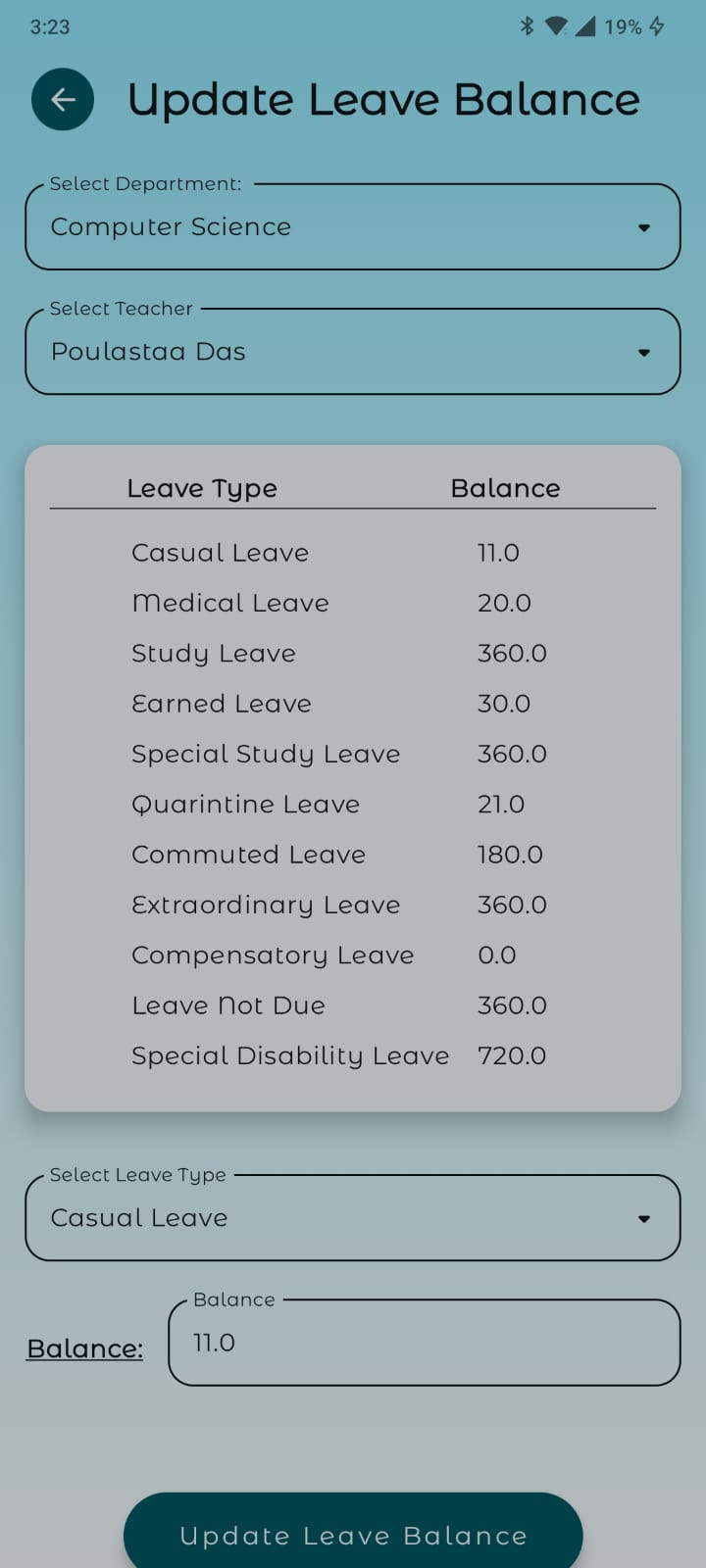
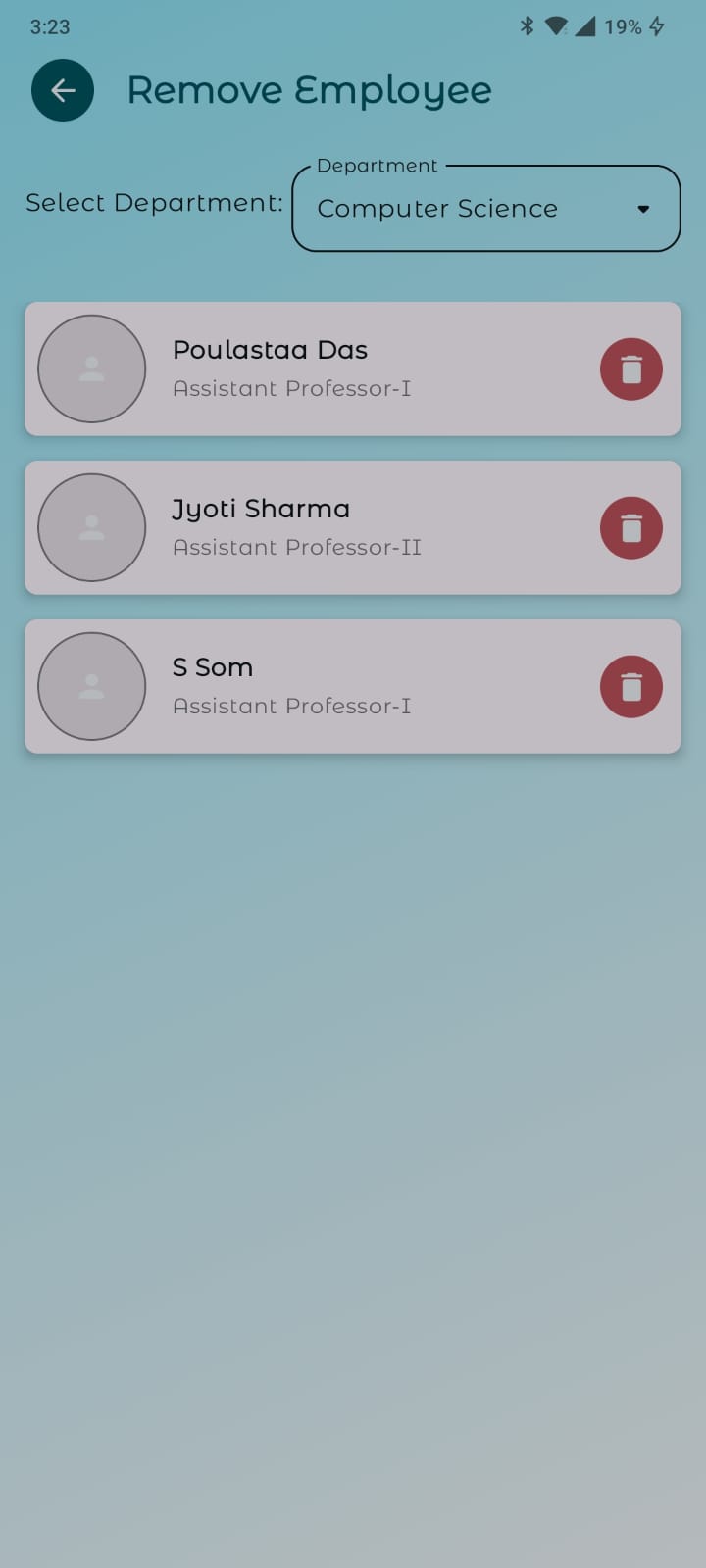


**3. Approved/Rejected Leave 4.Download Report**

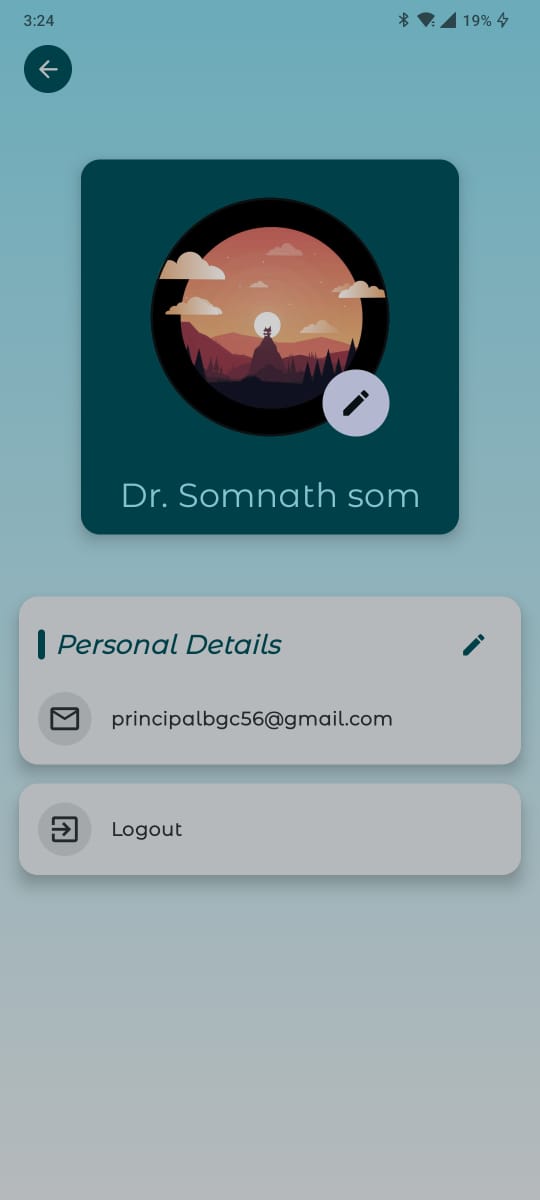
**5. Define Department In-charge 6. Add New Teacher**



**7. Remove An Employee 8. Update Leave balance**



**9. Profile**



**PROJECT LINK**

<https://github.com/POULASTAAdAS/LMS-College-Project>

**CHAPTER 4**

**CONCLUSION**

A well-designed Leave Management System is essential for efficient leave tracking, policy enforcement, and employee satisfaction. It streamlines the process of applying for, approving, and managing leave, reduces administrative burden, and ensures compliance with organizational and legal requirements. By automating leave workflows and providing accurate, real-time information, the system enhances transparency and decision-making.

It also offers valuable insights through reporting and analytics, helping organizations manage resources effectively and maintain a healthy work-life balance for employees.

**CHAPTER 5**

**FUTURE SCOPE**

**6.1 UI Improvement: -**

* **Adding New Them:** In the current context of the android application seems too much basic and consist of only one them. We are determined to add new themes and make the application more engaging.
* **Gallery Option:** We have also plan to add college-based images to promote college achievement. Which will be maintained by the admin and will be same for every employee.

**6.2 Integration with Other Systems**

* **HR and Payroll Systems**: Seamless integration with HR and payroll systems for automatic leave deduction, payroll processing, and benefits calculation.
* **Project Management Tools:** Integration with project management and scheduling tools to manage project timelines and resource availability.

**AI and Automation**

* **Chatbots and Virtual Assistants**: Implement AI-driven chatbots to handle leave requests, answer common queries, and provide instant assistance.

**6.3 Accessibility**

* **IOS and Web Apps:**  We have plans to make it accessible for IOS and web also.

### **6.4 Security and Privacy**

* **Data Security**: Enhance data security measures to protect sensitive employee information.
* **GDPR and Privacy Compliance**: Ensure the system complies with data protection regulations like GDPR.

### 6.5 **Globalization**

* **Multi-Language Support**: Provide multi-language support for global organizations.
* **Cross-Cultural Adaptability**: Adapt to diverse cultural norms and expectations regarding leave and work-life balance.

**CHAPTER 6**

**REFERENCES**

* [Kotlin](https://kotlinlang.org/)
* [Ktor](https://ktor.io/docs/welcome.html)
* [Exposed](https://ktor.io/docs/server-integrate-database.html)
* [Android Documentation](https://developer.android.com/develop)
* [Jetpack Compose](https://developer.android.com/compose)
* [MySQL Documentation](https://www.mysql.com/)
* [Postman Documentation](https://www.postman.com/)