

1. Introduction

The Web-Based Police Duty Scheduling project is a modern solution designed to revolutionize the way law enforcement agencies manage and organize officer duty schedules. This online platform leverages the power of the web to provide a user-friendly interface for efficient scheduling, enabling administrators to streamline the allocation of shifts while considering officer availability and organizational requirements. By transitioning from manual to web-based scheduling, this project aims to enhance flexibility, reduce administrative burden, and optimize the deployment of police personnel, ultimately contributing to improved operational effectiveness and officer well-being.

2. Problem Definition & Scope

1. Duty Repetition to Single Policemen:

In an offline duty assigning system using registers, the manual nature of the process increases the likelihood of duty repetition for a single policeman.

Lack of visibility into historical duty assignments can lead to inadvertent scheduling errors, resulting in officers being assigned the same duties frequently.

2. Data is Not Getting Stored Digitally:

The register-based system relies on physical records, making it susceptible to damage, loss, or unauthorized access.

Lack of digital storage prevents efficient data retrieval, analysis, and reporting, hindering the ability to track duty histories, trends, and performance metrics.

3. Manual Duty Assigning System is Confusing and Complicated:

The manual process of duty assignment using registers can be prone to human errors, leading to confusion and complications.

Officers and administrators may find it challenging to navigate through handwritten entries, increasing the risk of mistakes in duty assignments.

4. Offline Duty Scheduling System is Time-Consuming:

The manual nature of the offline duty scheduling system is time-consuming, requiring administrators to manually review and update duty assignments.

Time spent on manual scheduling could be better utilized for strategic planning, training, or addressing other critical tasks within the law enforcement agency.

5. Register-Based System Not Available Globally:

The traditional register-based system may not be universally accessible or compatible with the evolving technological landscape.

Law enforcement agencies with diverse infrastructures or operating in different regions may face challenges in implementing a standardized register-based system globally.

Scope of Project

The future scope of the Web-Based Police Duty Scheduling Project involves several avenues for improvement and expansion. One key aspect is the integration of more advanced reporting and analytics features, allowing law enforcement agencies to derive valuable insights from historical scheduling data. Enhanced reporting capabilities could aid administrators in making informed decisions about resource allocation, identifying trends, and optimizing scheduling processes. Additionally, exploring the implementation of

biometric authentication and mobile verification could add an extra layer of security to the system, ensuring the integrity and confidentiality of duty schedules. The project can also explore facilitating the efficient handling of officer requests for time off and minimizing disruptions in duty schedules. As technology advances, the project should stay adaptable to new web-based frameworks and technologies, ensuring continued compatibility and accessibility across various devices and platforms. Regular user feedback and collaboration with law enforcement professionals will be crucial for refining the system to meet the unique and evolving needs of different agencies, ultimately contributing to improved operational efficiency and officer satisfaction

2. Goals

1.Automation: Implement an automated scheduling system to streamline duty assignments and minimize manual effort.

2.Digital Management: Establish a secure digital database for efficient data storage, retrieval, and analysis.

3.Real-time Updates: Provide features for real-time updates and notifications to enhance communication and responsiveness.

4.Compliance: Ensure compliance with labor regulations, union agreements, and organizational policies.

5.Time Efficiency: Reduce manual scheduling time, allowing administrators to allocate time strategically.

6.Global Accessibility: Design a universally accessible system for law enforcement agencies worldwide.

3. Objectives

1.Transparency: Foster transparency in duty assignments by providing a clear and accessible system that allows both administrators and officers to easily understand and access their schedules.

2.Adaptability: Develop a system that can adapt to dynamic operational needs, allowing for quick adjustments in response to emergencies, unforeseen events, or changes in officer availability.

3.Compliance: Ensure compliance with labor laws, union agreements, and organizational policies regarding working hours, overtime, and rest periods, reducing the risk of legal issues.

4. **Communication:** Facilitate effective communication within the law enforcement agency by providing real-time updates and notifications regarding changes in duty schedules or other relevant information.
5. **Officer Satisfaction:** Improve officer satisfaction by considering preferences and creating fair and balanced duty schedules, contributing to a positive working environment.
6. **Resource Optimization:** Optimize the allocation of police personnel to prevent issues like fatigue and burnout, ultimately enhancing the agency's ability to maintain public safety.

Major Constraints

1. **Technology:** The system relies heavily on technical glitches or issues can impact the system's effectiveness.
2. **Internet connectivity:** Police Duty Scheduling system requires a stable and reliable internet connection, which can be a challenge in some areas.
3. **Data security:** With sensitive customer and financial data being stored online, data security is a major concern for online car service station management systems.
4. **User adoption:** The system may not be as effective if it is not adopted by all police stations, including mechanics and administrative staff.
5. **Integration with Police stations:** Building partnerships with police stations and accessing their data for matching purposes might face resistance or technical challenges due to differing systems and policies

6. Outcomes

1. **Optimal Workforce Utilization:** Maximizing the use of available personnel to meet operational needs efficiently.
2. **Employee Satisfaction:** Designing schedules that take into account officer preferences and work-life balance to enhance job satisfaction.
3. **Reduced Overtime Costs:** Minimizing the need for overtime by creating efficient duty schedules.

4.Improved Performance and Safety: Ensuring that officers are not overworked to maintain high levels of alertness, performance, and overall safety.

5.Adaptability and Flexibility: Creating a scheduling system that can adapt to changes in staffing requirements or unexpected events.

6.Compliance and Accountability: Ensuring that the scheduling system complies with legal regulations and department policies, and holding administrators accountable for any deviations.

7. Enhanced Communication: Facilitating effective communication between officers and scheduling administrators to address concerns and changes promptly.

7. Reduced Burnout: Designing schedules that minimize stress and fatigue, reducing the risk of burnout among officers.

3. Software Requirements Specification (SRS)

1. Team Members (Stake Holders):

1. Police
2. Chaturshringi police station Staff Members

3. Scope:

- This system can be used in police station as well as corporate world.
- Policemen can use the portal at anytime and anywhere also it is web-based application.
- Admin and Attendance masters has privileges to create, modify, view schedules and policemen view schedules.
- The system will provide functionalities to register attendance master, policemen to manage scheduling reports for officers and Staff members

4. Functional Requirements:

➤ Admin

- Admin will be able to login himself. Then admin will register attendance masters.
- Admin can also view the daily duty reports. If any emergency occurs, then admin can arrange emergency meet.
- Admin has authority to accept or reject officers as well as attendance master leave.
- Admin should be able to create different types of shifts (day,night,overtime).

➤ Attendance Masters:

- Attendance master will login himself first, then do many activities.
- Attendance master can change his own authentication credentials.
- Attendance master will register all policemen.

- Attendance master will insert the list of activities with locations.
- Attendance master assign duty to policemen through schedule sheet and authority to modification in schedule sheet.
- Attendance master can view list of present officers and staff members.
- Attendance master can generate scheduled duty report.
- He maintained a historical record of schedules and they may use for auditing purpose.

➤ **Policemen :**

- Policemen will login himself and view scheduled duty report.
- Policemen can change his own authentication credentials.
- Policemen can apply for leave.

5. Non-Functional Requirements:

A. Security

Registered officers will allow to see the generated reports. The website should have authentication process for admin, attendance master, police officers while logging. System will automatically log off all user after some time due to inactiveness. System will internally maintain secure communication channel between Servers (Web Servers, App Servers, database Server). Sensitive data will be always encrypted across communication.

B. Reliability

Continuous updates are maintained, continuous Administration is done to keep system operational. During peak hours system will maintain same user experience by managing load balancing.

C. Availability

Uptime: 24*7 available, 99.999%

D. Maintainability:

A database software will be used to maintain System data Persistence. IT operations team will easily monitor and configure System using administrative tools provided by Servers. Separate environment will be maintained for system for isolation in production, testing, and development.

E. Portability:

PDA: Portable Device Application Portable device application system will be provided portable user interface through users will be able to access online web-based system. System can be deployed to single server, multi-server, to any OS, Cloud (Azure or AWS or GCP).

F. Accessibility:

Only registered admin, attendance master and officers will be able to login on website after authentication.

G. Durability:

The system will maintain officer's details. The system will implement backup and recovery for retaining officers' data over the time. The system will use cache for faster data retrieval and improved performance.

H. Efficiency:

Maximum number of officers will login to website at same time. System will be able to manage all transactions with isolation.

I. Modularity:

System will design and developed using reusable, independent scenarios in the form of modules. These modules will be loosely coupled and highly cohesive.

J. Scalability:

The system will provide a consistent user experience to users irrespective of load.

K. Safety:

The officer's login page will be secure from malicious attacks and phishing. Separate environments will be maintained for the system for isolation in production, testing, and development.

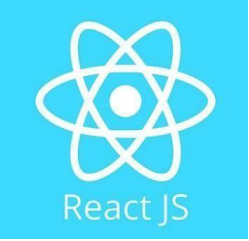


6. Constraints

- a) The web application should support common web browsers such as Chrome, Safari, Firefox, etc.
- b) User Device Compatibility
- c) Compliance with relevant data protection regulations.
- d) Content Moderation
- e) Budget Constraints

4. Performance Requirements

Performance requirements for a recipe website would include quick loading times for smooth navigation, dependable and secure data storage to protect user data, and capacity to manage many concurrent users looking for recipes. A pleasant user experience depends on mobile compatibility for access while on the go, as well as user-friendly navigation and an intuitive recipe interface. To improve the recipe-sharing platform, it is imperative to track user interaction, recipe popularity, and other relevant indicators using comprehensive reporting and analytics tools.

1. Software Requirements

FRONT END	BACKEND	DATABASE
		
REACT JS	J2EE	MySQL

A) Frontend Technologies:

React:

Version: 18.2.0 (Latest stable release)

Description: JavaScript library for building user interfaces.

HTML, CSS, JavaScript:

Utilized for structuring web pages, styling, and client-side scripting.

Bootstrap:

Version: 5.3.2 (Latest stable release)

Description: Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

B) Backend Technologies:

Spring Boot:

Version: 2.7.14 (Latest stable release)

Description: Java-based framework for building robust and scalable backend applications.

Hibernate:

Version: Latest stable release

Description: Object-relational mapping (ORM) tool for data management.

C) Database technologies:

MySQL:

Version: 8.0 (Latest stable release)

Description: Relational database management system for storing application data.

D) Development Environment:

Integrated Development Environment (IDE): Eclipse, VS Code

Description: An IDE for coding, testing, and debugging application components.

E) Git:

Version: 2.42.0.windows.2 (Latest stable release)

Description: Distributed version control system for tracking changes in source code.

F) Web Server:

Apache Tomcat:

Version: 9.0 (Latest stable release)

Description: Web server and servlet container for hosting the Java-based application.

G) Deployment Platform:

Frontend = Vercel

Vercel builds a frontend-as-a-service product—they make it easy for engineers to deploy and run the user facing parts of their applications.

Backend = Railway

Railway is the cloud for building, shipping, and monitoring applications.

2. Hardware Technologies (Minimum)

Server Side			
Monitor	Processor	RAM	Disk Space
Resolution 1024x768	Intel or AMD 2GHZ OR HIGHER	4GB	10GB
Client Side			
Monitor	Processor	RAM	Disk Space
Resolution: 1024x768	Intel or AMD 1GHZ DUAL CORE MIN	2 GB	128 GB

A) Server

A robust server with sufficient processing power and memory to handle concurrent user requests and ensure smooth application performance.

B) Storage

Adequate storage capacity for the database and multimedia content (images, videos) associated with recipes.

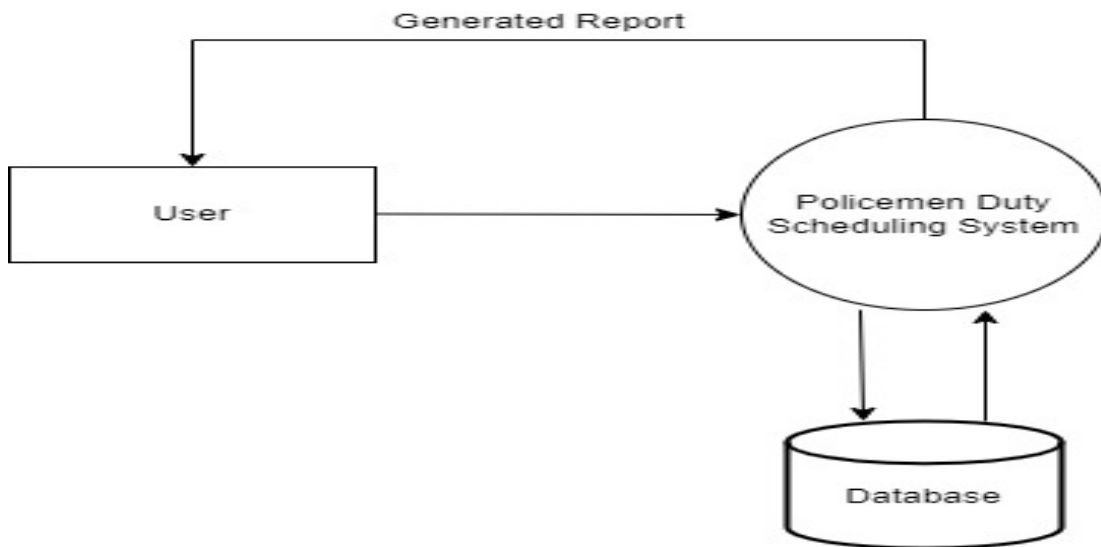
C) Network Infrastructure:

Reliable internet connectivity and network infrastructure to support data transmission between the frontend and backend components.

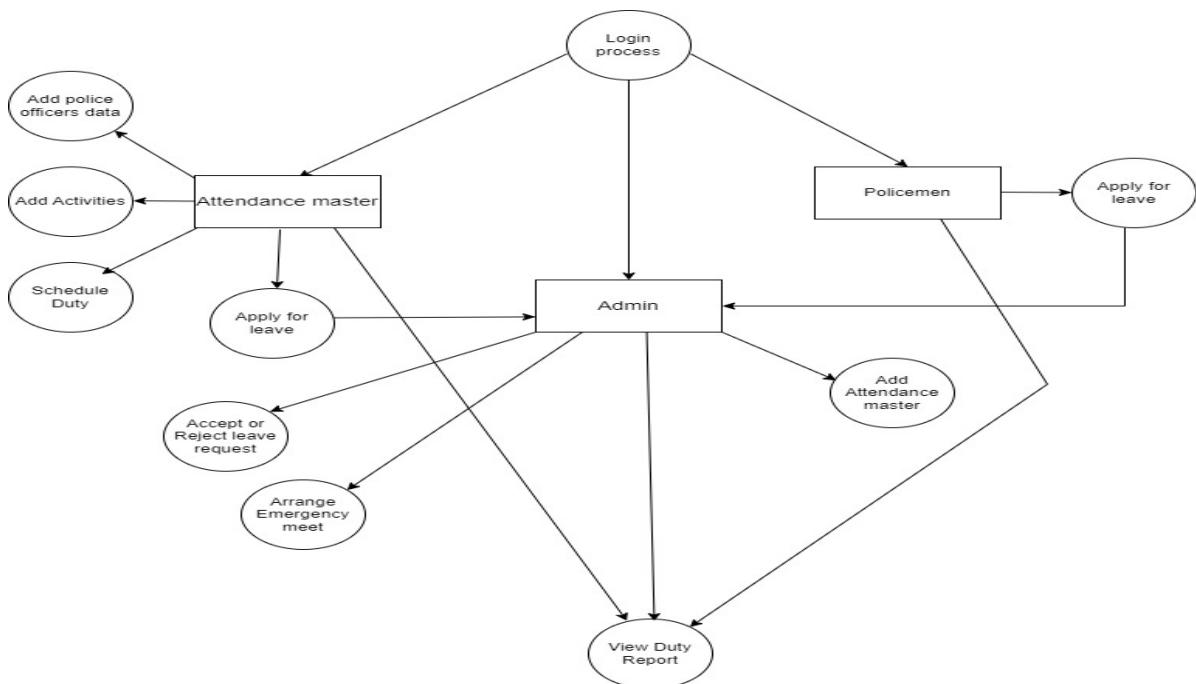
5. UML Diagrams

1. Data Flow Diagram

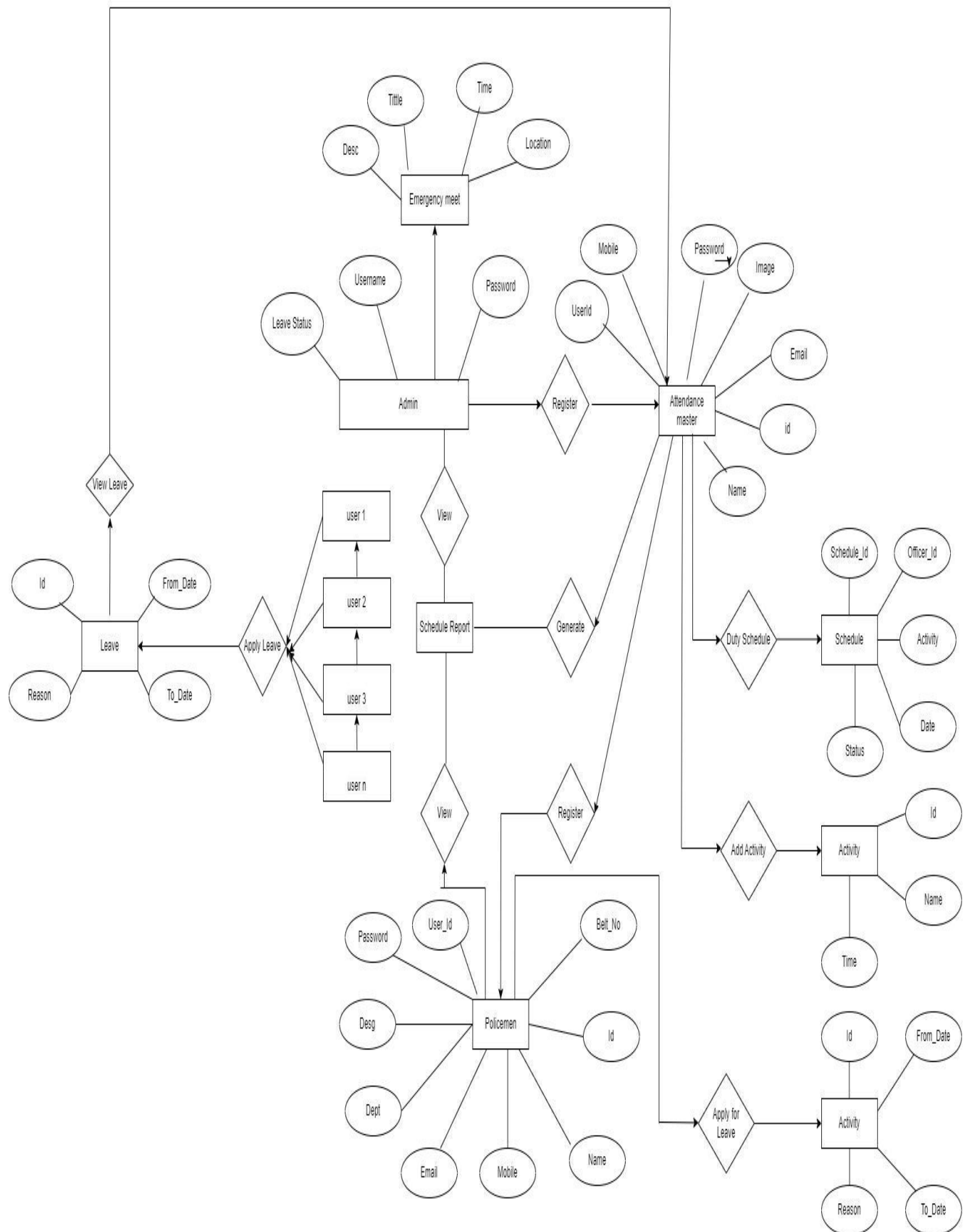
DFD Level 0



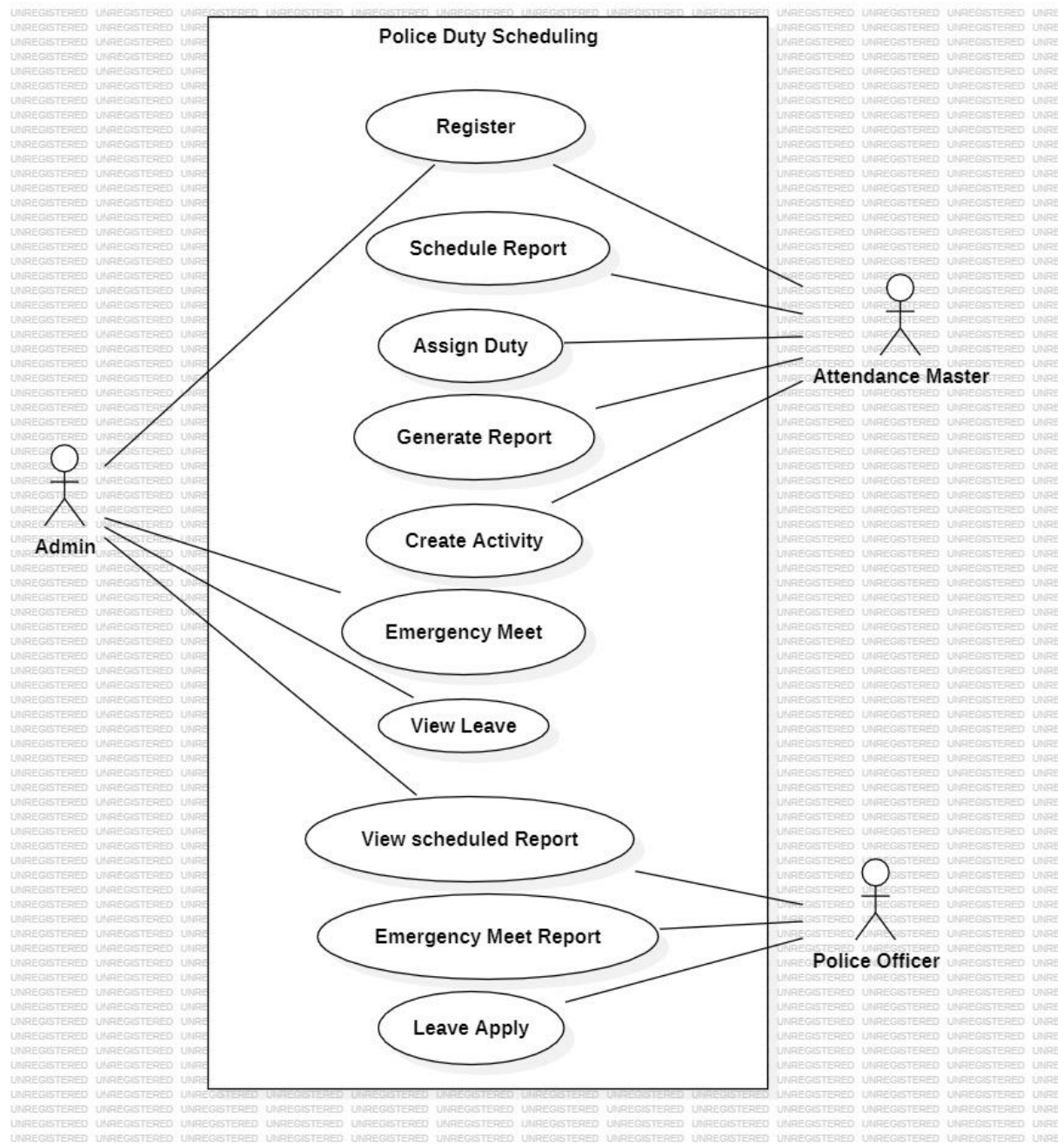
DFD Level 1



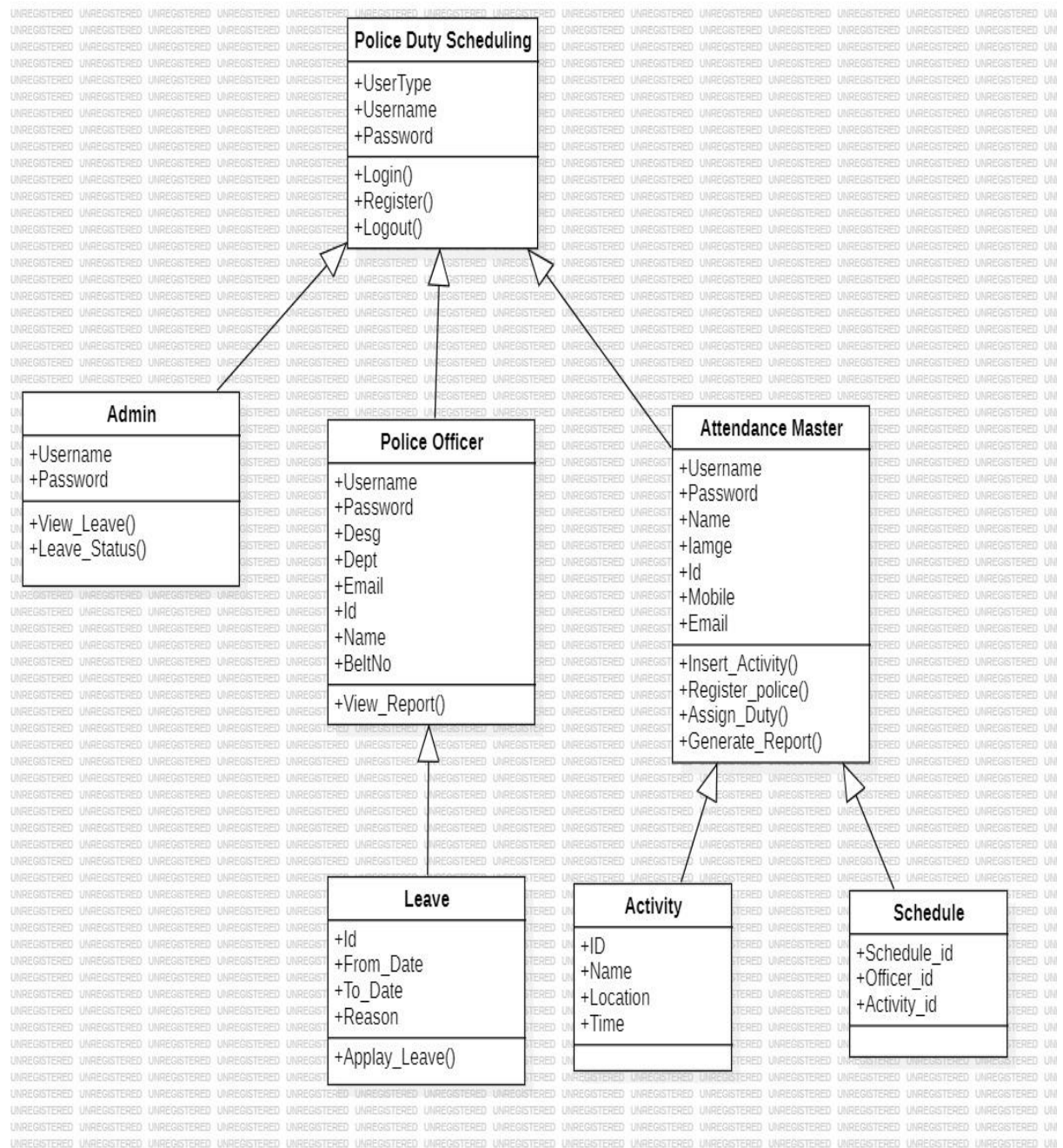
2. Entity Relation Diagram (ERD)



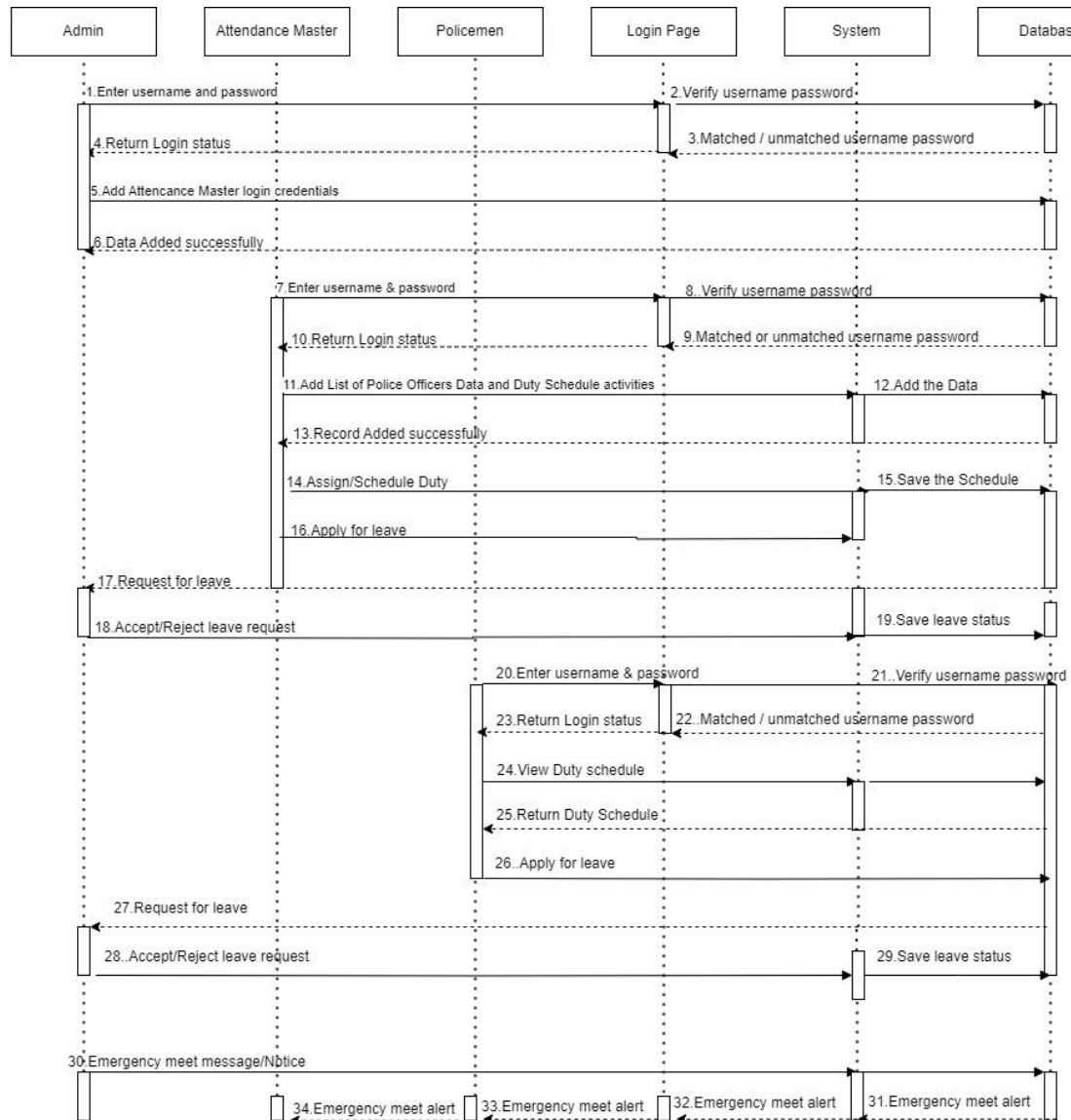
3. Use Case Diagram



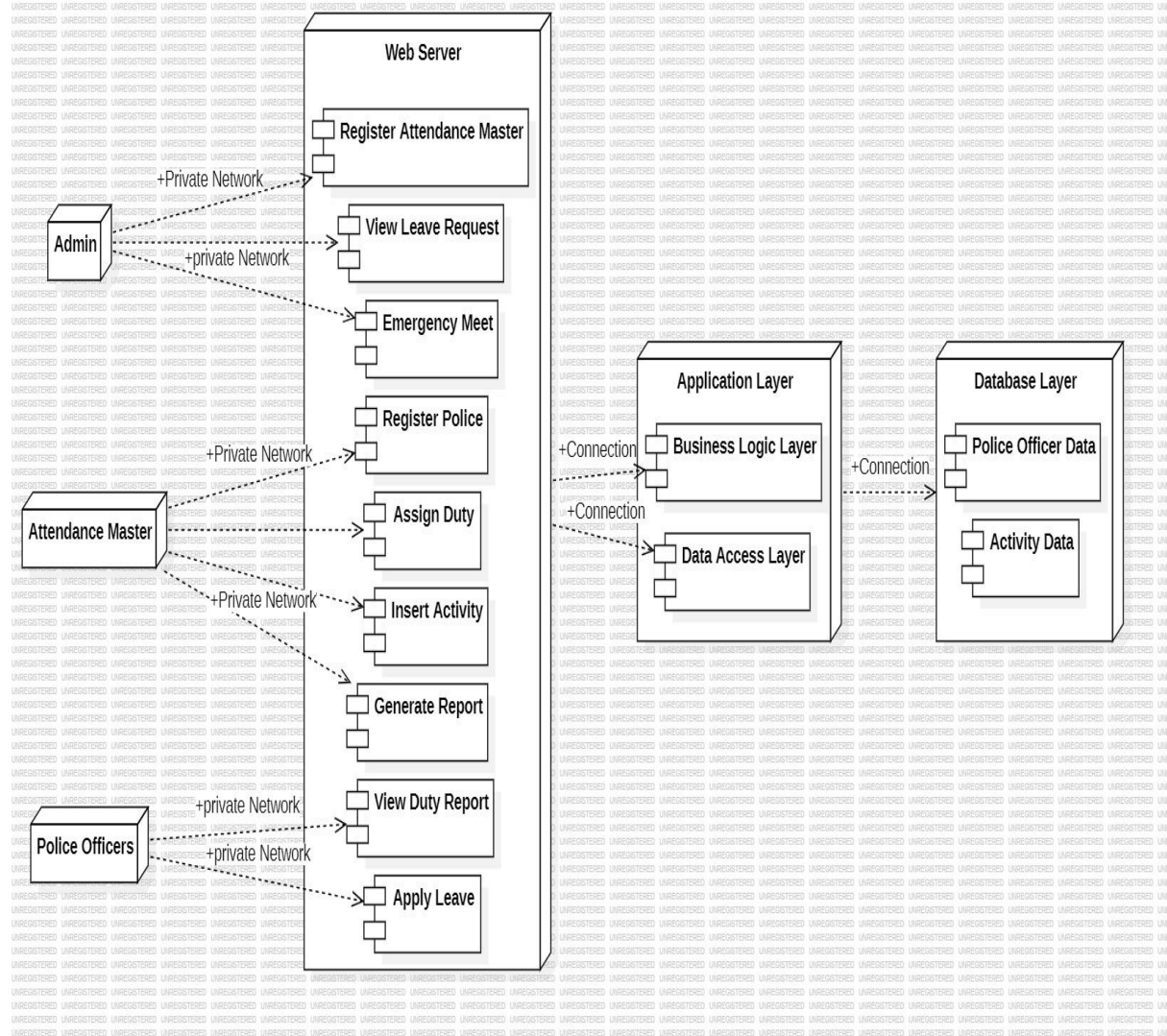
4. Class Diagram



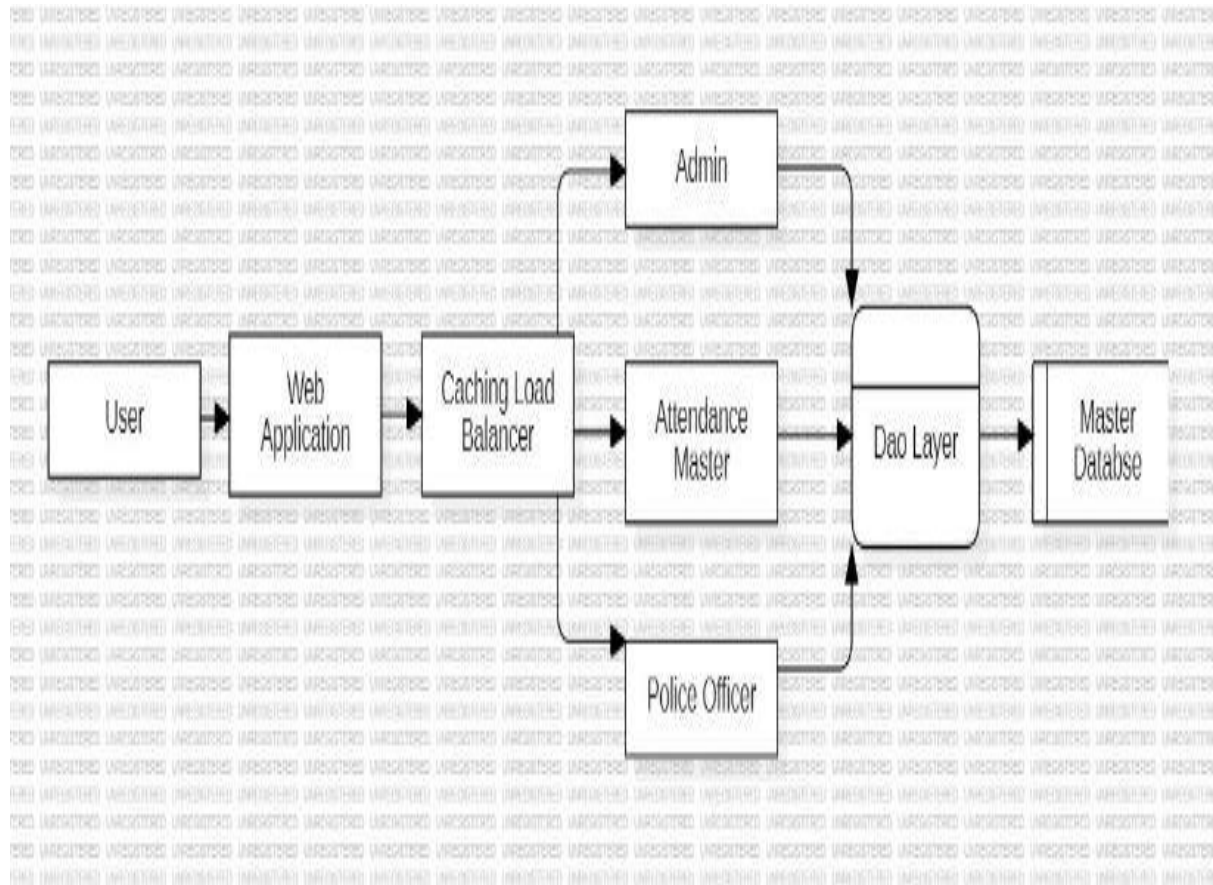
5. Sequence Diagram



6. Deployment Diagram



6. System Architecture



The system architecture for Police Duty Scheduling can follow a typical three-tier architecture, separating the application into three main components: the presentation layer (frontend), the application layer (backend), and the data layer (database). Here is an overview of the system architecture:

1. Presentation Layer (Frontend):

Technology: React (HTML, CSS, JavaScript)

Description: The user interface where users interact with the application. React is used for building responsive and dynamic components, allowing for an interactive and engaging user experience.

2. Application Layer (Backend):

Technology: Spring Boot (Java)

Description: The backend logic and application processes are managed using the Spring Boot framework. It handles user requests, business logic, and communication with the database.

3. Data Layer (Database):

Technology: MySQL

Description: The MySQL database stores user profiles, recipes, and related data. It interacts with the backend to retrieve and update information.

4. External APIs (Optional):

Description: Additional external APIs, if integrated, provide services such as nutritional information retrieval. These APIs can be accessed by the backend for enhanced functionality.

5. Cloud Service (Optional):

Description: Cloud platforms like AWS, Google Cloud, or Azure can be used for hosting the application, providing scalability, flexibility, and reliable infrastructure.

6. Development Environment:

Technology: IntelliJ IDEA, Eclipse (Optional)

Description: Integrated Development Environments (IDEs) are used during the development phase for coding, testing, and debugging application components.

7. Version Control:

Technology: Git

Description: Git is employed for version control, allowing multiple developers to collaborate on the codebase, track changes, and manage the development workflow.

8. Web Server (Optional):

Technology: Apache Tomcat (or similar)

Description: A web server may be used to host the React frontend files and serve them to users' web browsers.

7. Test Cases

Testing for a web application like Police Duty Scheduling involves verifying various aspects of its functionality, usability, and security. Below are examples of test cases covering different scenarios:

User Registration:

- a) Verify that users can successfully register with valid information.
- b) Check for appropriate error messages when invalid or incomplete information is provided.
- c) Ensure the uniqueness of usernames and email addresses during registration.

Login Functionality:

- a) Confirm that registered users can log in with correct credentials.
- b) Test the handling of incorrect login attempts, ensuring appropriate error messages are displayed.
- c) Check for session management, including proper session expiration and user logout.

Record Submission:

- a) Test the ability to submit a new record with all required fields.
- b) Verify that users can upload images or multimedia content related to the record.
- c) Check for appropriate error handling when required fields are left blank or invalid data is entered.

Record Search and Discovery:

- a) Test the search functionality by searching for record based on keywords, ingredients, or cuisine types.
- b) Verify that search results are relevant and displayed correctly.
- c) Check for pagination or infinite scrolling functionality for a large number of search results.

Community Interaction:

- a) Test the ability to like, comment on, and share record
- b) Verify that notifications are triggered for user interactions.
- c) Check for proper display of user-generated content within the community.

Responsive Design:

- a) Test the application's responsiveness on different devices, including desktops, tablets, and smartphones.
- b) Verify that the layout adjusts appropriately to various screen sizes.

Security Testing:

- a) Test for SQL injection vulnerabilities by entering malicious input.
- b) Ensure that user authentication and authorization mechanisms are robust.

- c) Verify that sensitive information is appropriately protected, especially during data transmission (HTTPS).

Cross-Browser Compatibility:

- a) Test the application on different browsers (Chrome, Firefox, Safari, Edge) to ensure consistent behavior.
- b) Check for compatibility with different browser versions.

Performance Testing:

- a) Test the application's response time under different levels of user load.
- b) Verify that the application performs well in terms of speed and responsiveness.

Error Handling:

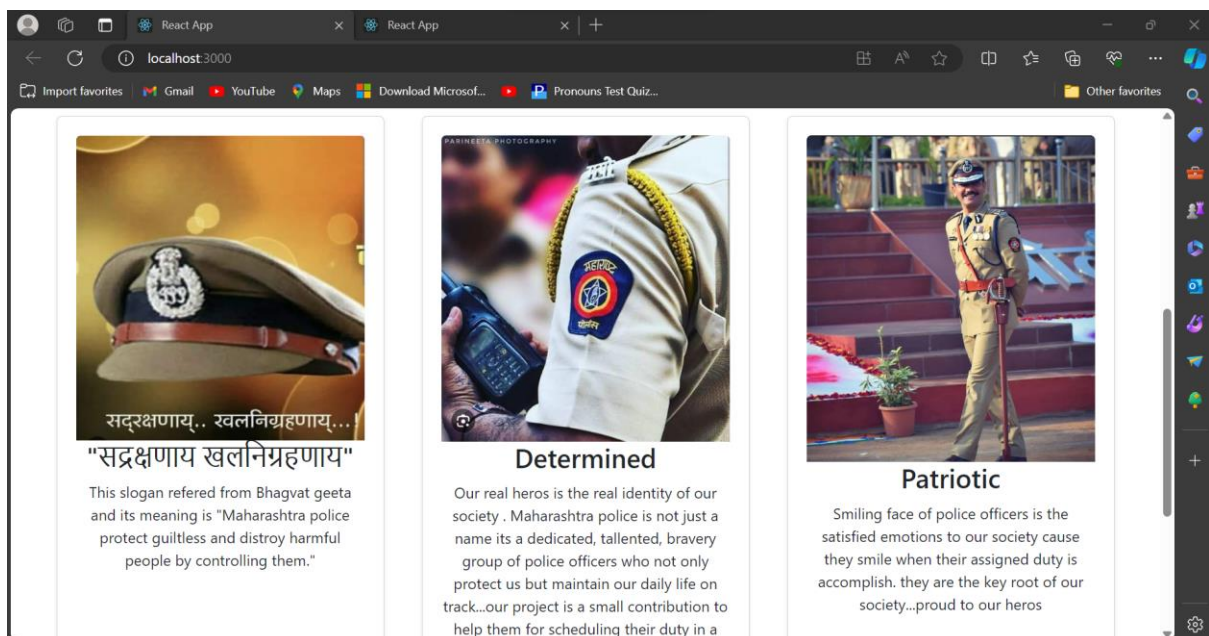
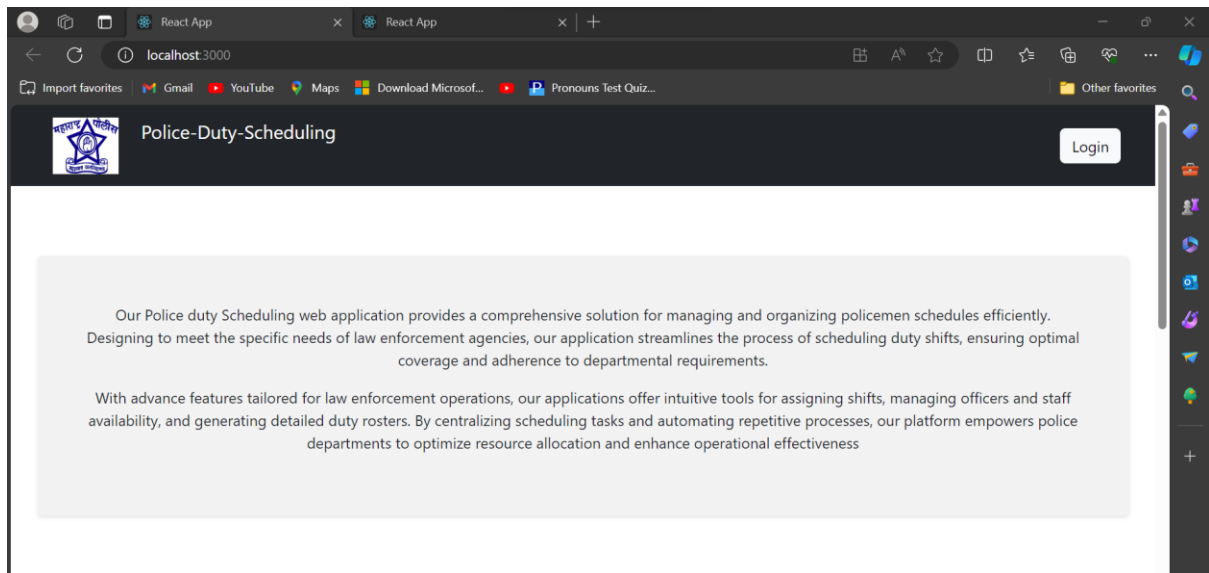
- a) Test how the application handles unexpected errors or exceptions.
- b) Verify that users receive clear and helpful error messages.
- c) Compatibility with External APIs (if applicable):

If the application integrates with external APIs for features like nutritional information, test the integration for accuracy and reliability.

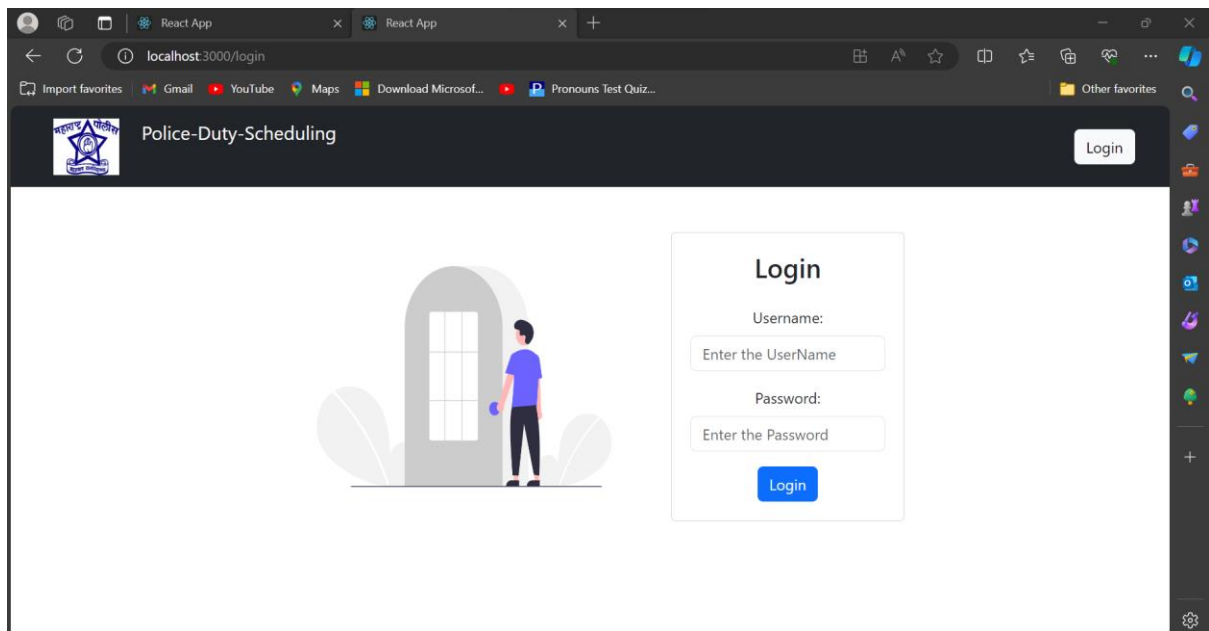
These test cases cover various functional and non-functional aspects of the FlavourFeed web application, ensuring a thorough evaluation of its performance and user experience. Adapt and expand these test cases based on the specific features and requirements of your project.

8. Screenshots

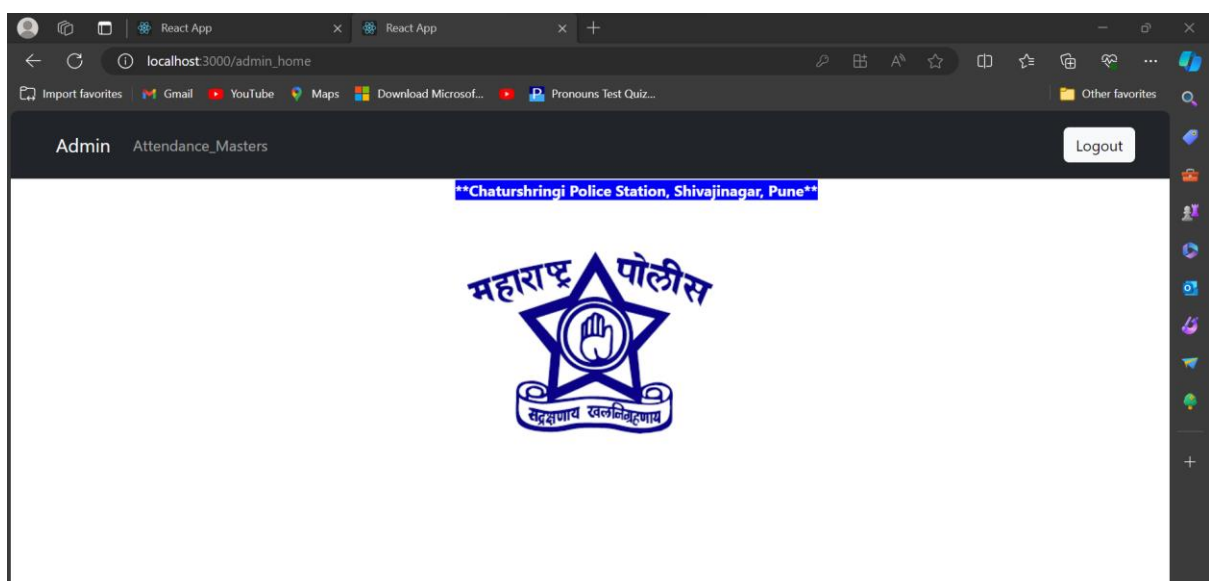
Home/Landing Page



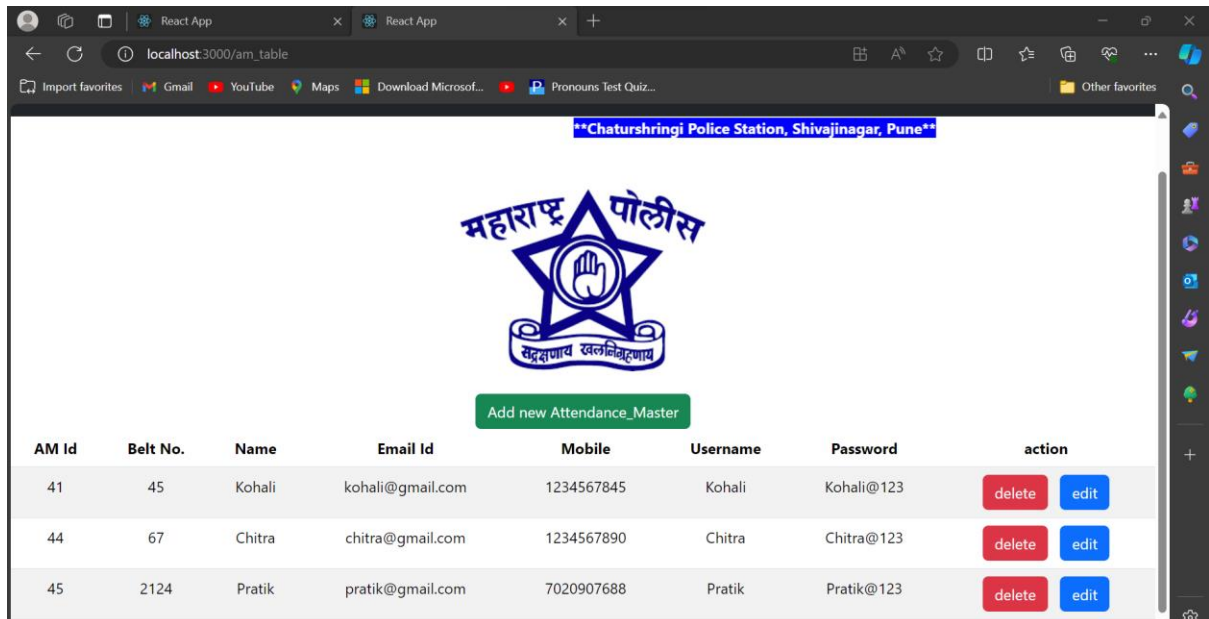
Login Page



Admin Home



Admin Functionality



Attendance Master registration

Belt No. :

Enter the Belt Number

Name :

Enter the Attendance_Master Name

Email Id :

Enter your email (e.g., example@gmail.com)

Mobile :

Enter the Mobile Number

Username :

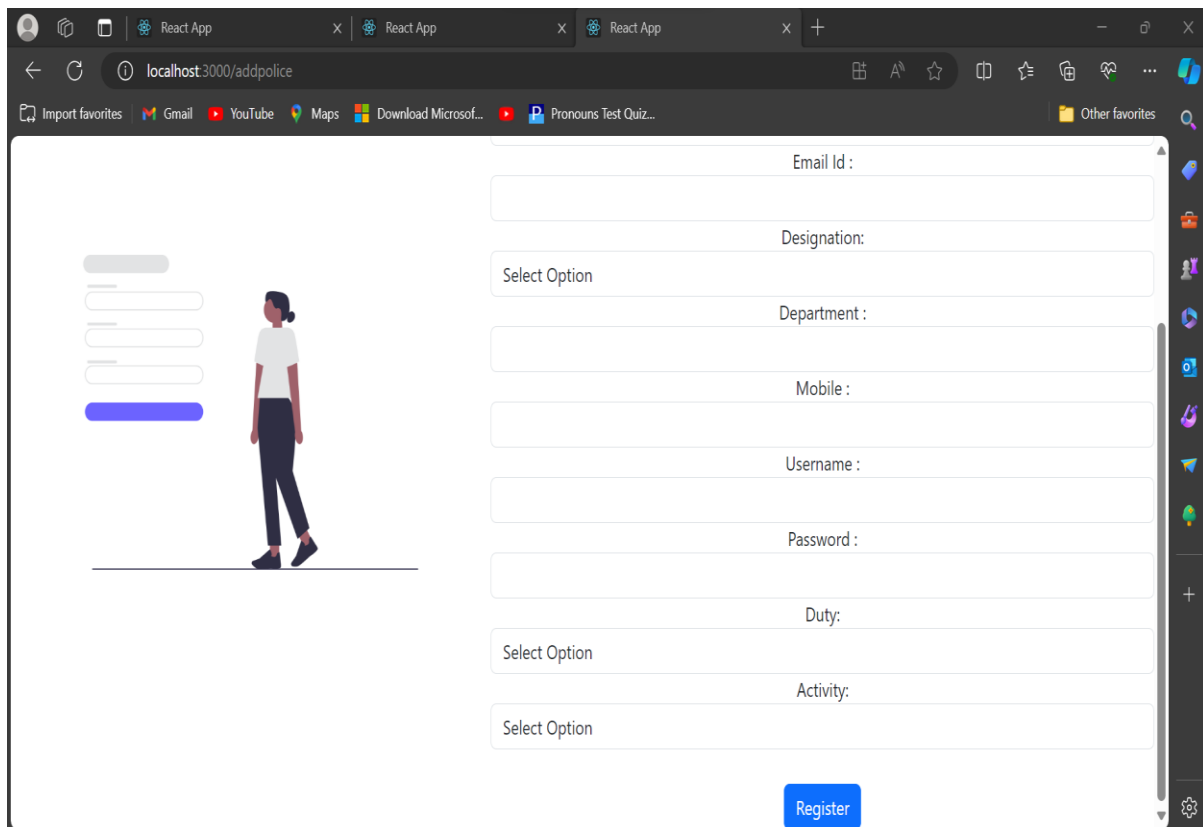
Enter the Username

Password :

Enter the Password

[Register](#)

Police Registration



localhost:3000/addpolice

Email Id :

Designation:

Select Option

Department :

Mobile :

Username :

Password :

Duty:

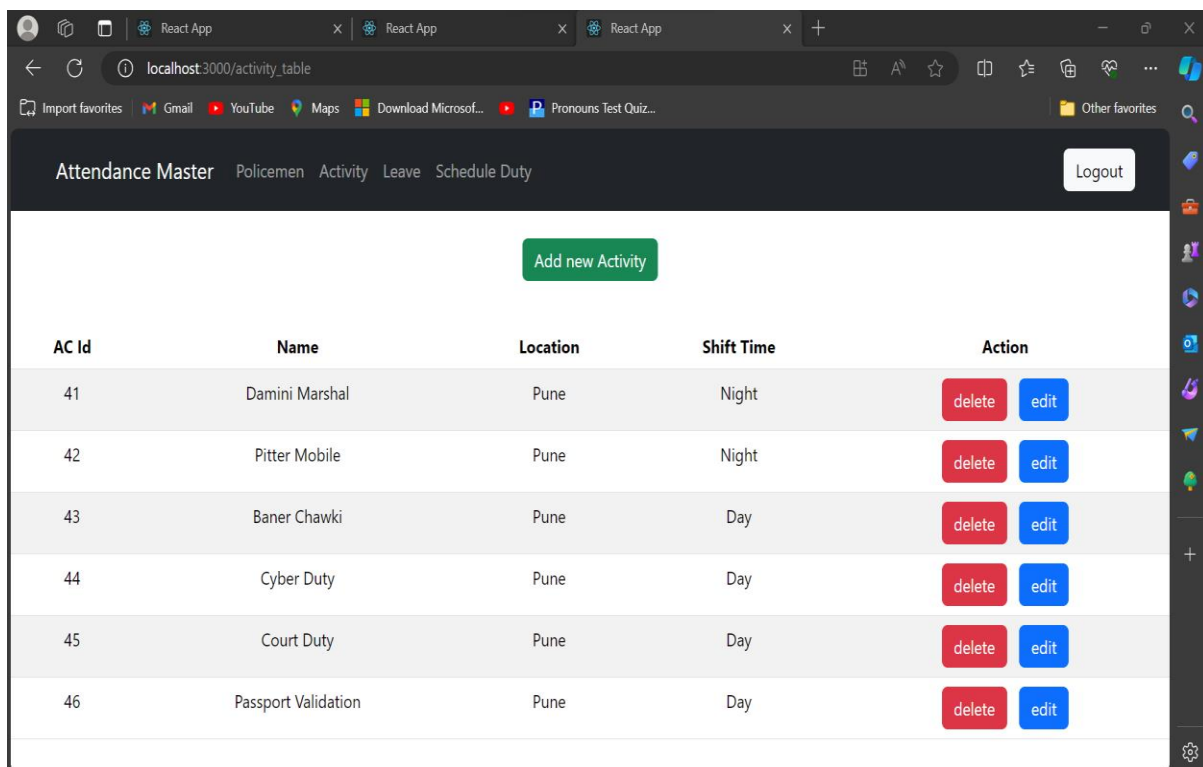
Select Option

Activity:

Select Option

Register

Activity Schedule



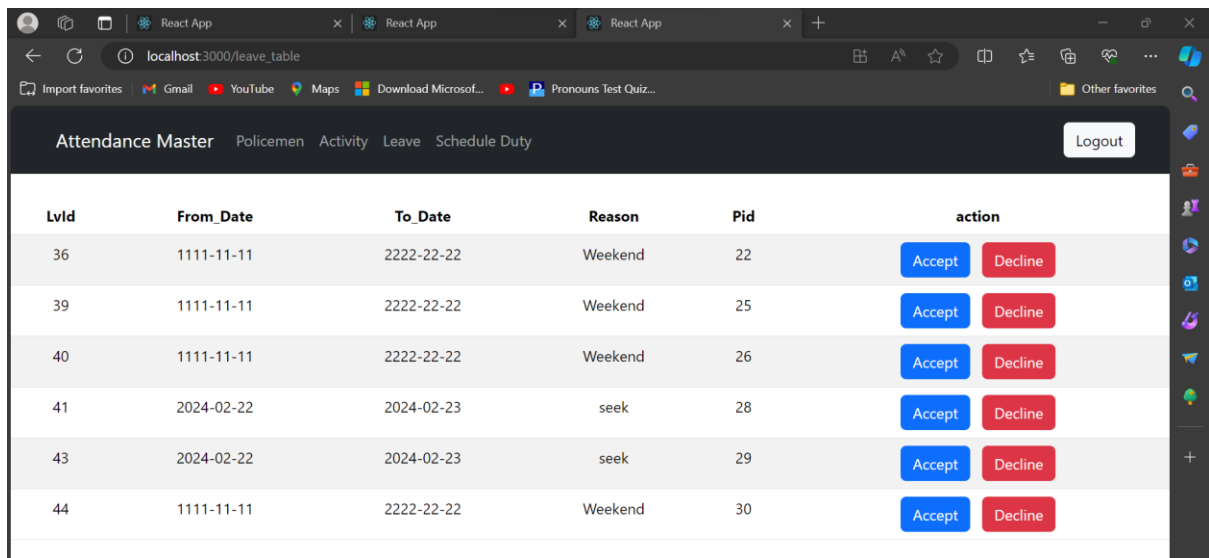
localhost:3000/activity_table

Attendance Master Policemen Activity Leave Schedule Duty Logout

Add new Activity

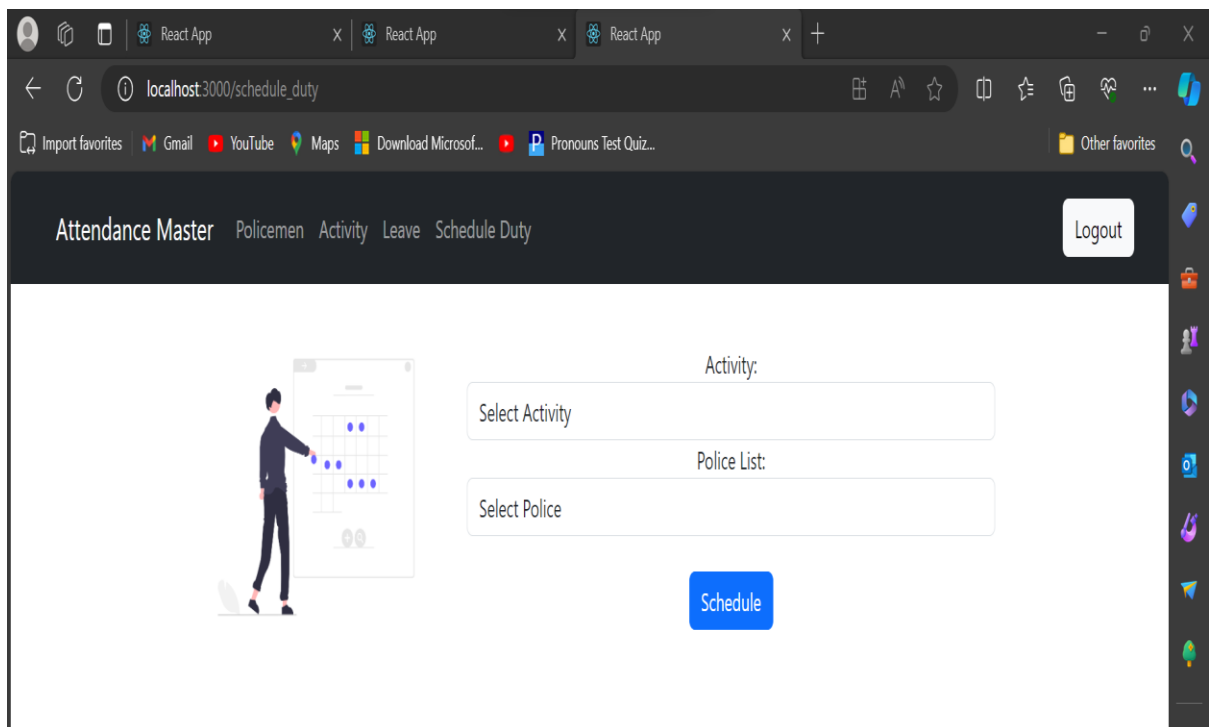
AC Id	Name	Location	Shift Time	Action
41	Damini Marshal	Pune	Night	<button>delete</button> <button>edit</button>
42	Pitter Mobile	Pune	Night	<button>delete</button> <button>edit</button>
43	Baner Chawki	Pune	Day	<button>delete</button> <button>edit</button>
44	Cyber Duty	Pune	Day	<button>delete</button> <button>edit</button>
45	Court Duty	Pune	Day	<button>delete</button> <button>edit</button>
46	Passport Validation	Pune	Day	<button>delete</button> <button>edit</button>

Authority to accept or reject Leave



Lvid	From_Date	To_Date	Reason	Pid	action
36	1111-11-11	2222-22-22	Weekend	22	<button>Accept</button> <button>Decline</button>
39	1111-11-11	2222-22-22	Weekend	25	<button>Accept</button> <button>Decline</button>
40	1111-11-11	2222-22-22	Weekend	26	<button>Accept</button> <button>Decline</button>
41	2024-02-22	2024-02-23	seek	28	<button>Accept</button> <button>Decline</button>
43	2024-02-22	2024-02-23	seek	29	<button>Accept</button> <button>Decline</button>
44	1111-11-11	2222-22-22	Weekend	30	<button>Accept</button> <button>Decline</button>

Authority of Scheduling Duty



Activity:

Police List:

Schedule

Police Apply Leave

The screenshot shows a web browser window with the URL `localhost:3000/police_leave`. The application has a dark header with the text "Policeman Apply_Leave Duties" and a "Logout" button. The main content area features an illustration of a policeman on the left. To the right, there is a form with three input fields: "From Date :" (placeholder: YYYY-MM-DD), "To Date :" (placeholder: YYYY-MM-DD), and "Reason :". Below these fields is a blue "Apply Leave" button.

Duty Details

The screenshot shows a web browser window with the URL `localhost:3000/police_duty`. The application has a dark header with the text "Policeman Apply_Leave Duties" and a "Logout" button. The main content area features the title "Duty Details" and a table with the following data:

PID	Activity	Location	Shift
31	Damini Marshal	Pune	Night

Below the table, there is an illustration of a person walking next to a document icon.

9. References

- [1] React. (2023). <https://reactjs.org/>
- [2] Spring Framework. (2023). <https://spring.io/projects/spring-framework>
- [3] Hibernate ORM. (2023). <https://hibernate.org/orm/>
- [4] MySQL Database. (2023). <https://www.mysql.com/>
- [5] Apache Tomcat. (2023). <http://tomcat.apache.org/>
- [6] React Testing Library. (2023). <https://testing-library.com/docs/react-testing-library/intro/>
- [7] Git. (2023). <https://git-scm.com/>