“LEAVE MANAGEMENT SYSTEM”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:
Dr. Sanwta Ram Dogiwal
Associate Professor-2

Submitted By :
Yashpal Siyag (21ESKIT308)
Sonu Saini (21ESKIT300)
Rahul Ahuja (21ESKIT302)

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2024-2025

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Yashpal Siyag**, a student of B.Tech(Information Technology) VIII semester has submitted her Project Report entitled **”Leave Management System**” under my guidance.

Mentor	**Coordinator**
Dr. Sanwta Ram Dogiwal	Dr. Richa Rawal
Associate Professor-2	Associate Professor-1
Signature............	Signature............

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur
Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Sonu Saini**, a student of B.Tech(Information Technology) VIII semester has submitted his Project Report entitled **”Leave Management System”** under my guidance.

Mentor	**Coordinator**
Dr. Sanwta Ram Dogiwal	Dr. Richa Rawal
Associate Professor-2	Associate Professor-1
Signature............	Signature............

Swami Keshvanand Institute of Technology,

Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr. Rahul Ahuja**, a student of B.Tech(Information Technology) VIII semester has submitted his Project Report entitled **”Leave Management System”** under my guidance.

Mentor	**Coordinator**
Dr. Sanwta Ram Dogiwal	Dr. Richa Rawal
Associate Professor-2	Associate Professor-1
Signature............	Signature............

DECLARATION

We hereby declare that the report of the project entitled **”Leave Management System”** is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of **Dr. Sanwta Ram Dogiwal**(Dept. of Information Technology) and coordination of **Dr. Richa Rawal**(Dept.of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology. It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members	**Signature**
Yashpal Siyag (21ESKIT308)	
Sonu Saini (21ESKIT300)	
Rahul Ahuja (21ESKIT302)	

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor **Dr. Sanwta Ram Dogiwal** .He has been a guide, motivator, source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator **Dr. Richa Rawal** for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to **Dr. Anil Chaudhary**, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:
Yashpal Siyag (21ESKIT308)
Sonu Saini (21ESKIT300)
Rahul Ahuja (21ESKIT302)

Abstract

The Leave Management System is a web-based application developed to simplify and automate the leave application and approval process within an organization. The system provides a centralized platform where employees can submit leave requests, check their leave balances, and track approval status, while administrators or managers can review, approve, or reject these requests and maintain proper records.

This project has been implemented using HTML and CSS for the front-end design, the Laravel PHP framework for the back-end logic, and MySQL as the database management system. The use of Laravel ensures secure, scalable, and structured development with built-in functionalities for routing, authentication, and database interactions. The system supports different user roles (admin and employee) and leave type categorization.

The Leave Management System ensures accessibility from anywhere, improves communication between staff and management, minimizes paperwork, and reduces administrative workload. This project showcases the practical application of full-stack web development skills and provides an effective digital solution to a real-world organizational challenge.

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement and Objective

In many organizations, including educational institutions and corporate offices, the leave management process is still handled manually or through basic spreadsheets, which often leads to inefficiencies such as delayed approvals, loss of records, miscommunication, and difficulty in tracking leave balances. This traditional approach lacks transparency, consumes valuable administrative time, and is prone to human errors. To address these issues, there is a need for a centralized, automated, and user-friendly system that simplifies leave tracking and approval workflows.

The objective of the Leave Management System is to provide an efficient digital solution that automates the entire leave management process. It enables employees to apply for leave online and track their application status, while allowing administrators to view, approve or reject leave requests and manage leave records with ease. The system is developed using HTML and CSS for the front-end, Laravel PHP framework for the back-end, and MySQL as the database.

It ensures role-based access, maintains accurate leave records, enhances transparency, and minimizes paperwork. Additionally, it supports report generation and cloud deployment for anytime, anywhere access. Overall, the system aims to streamline administrative tasks and improve communication between staff and management..

1.2 Literature Survey /Market Survey/Investigation and Analysis

Leave management is a critical aspect of human resource (HR) operations in any organization. Traditional leave management methods, such as paper forms or spreadsheets, are time-consuming, error-prone, and difficult to maintain. As organizations grow, managing leave manually becomes increasingly complex, leading to operational inefficiencies. The demand for automated leave management solutions has thus grown significantly, prompting the development of various digital systems.

Several commercial and open-source leave management systems are currently available in the market, including platforms like Zoho People, Keka HR, and greytHR. These systems offer advanced features such as automated leave approval workflows, integration with payroll systems, attendance tracking, and mobile access. However, these tools often come with subscription costs and may offer features that are either excessive or not customizable for smaller organizations or institutions.

Open-source solutions like OrangeHRM provide flexibility but require technical expertise for customization and maintenance. Moreover, not all solutions are localized or adapted to specific institutional policies and needs. This gap presents an opportunity to develop a lightweight, cost-effective, and customizable leave management system tailored to the requirements of educational institutions or small-to-medium enterprises.

In our analysis, we found that users prioritize simplicity, reliability, and transparency in a leave management system. Therefore, our project focuses on building a user-friendly, secure, and scalable web-based application using HTML, CSS, Laravel (PHP Framework), and MySQL. The system aims to meet the functional needs of both employees and administrators while being easy to deploy in a cloud environment for remote access and centralized control.

1.3 Introduction to Project

The Leave Management System is a web-based application developed to automate and simplify the process of managing employee leave within an organization. Traditional manual methods are often inefficient and error-prone, leading to delays and miscommunication. This system allows employees to apply for leave online and track their application status, while administrators can review, approve, or reject requests and generate reports.

Built using HTML, CSS, Laravel (PHP framework), and MySQL, the system ensures secure data handling and smooth functionality. Designed for deployment in a cloud environment, it offers easy access, reduces paperwork, and enhances transparency and efficiency in leave management.

1.4 Proposed Logic / Algorithm / Business Plan / Solution / Device

1.4.1 Logic

The Leave Management System operates with a simple workflow. Employees log in to the system, fill out and submit their leave requests. These requests are stored in the database with a ”Pending” status. Administrators then log in to review pending requests, check the employee’s leave balance, and either approve or reject the leave. Once the decision is made, the employee is notified, and if approved, the leave balance is automatically updated. This process ensures transparency, efficiency, and accurate leave tracking.

1.4.2 Algorithm

- **Login:** Employee/Admin logs in.
- **Employee Request:** Employee submits leave request (stored as ”Pending”)
- **Admin Review:** Admin views requests and checks leave balance.
- **Approve/Reject:**

  If balance is sufficient, approve and update status to ”Approved”.

  If balance is insufficient, reject and update status to ”Rejected”.
- **Notify:** Employee is notified of the decision.
- **Update Balance (if approved):** Deduct leave days from balance.

- **End**.

1.4.3 Solution

The Leave Management System provides an automated, digital solution to streamline the process of managing employee leaves. By moving away from manual processes, the system reduces errors, ensures transparency, and increases efficiency. Employees can easily apply for leave online, while administrators can quickly review, approve, or reject requests. With automated leave balance updates and real-time notifications, the system improves communication and minimizes delays. The use of Laravel (PHP framework) for back-end development, MySQL for data storage, and cloud deployment ensures scalability, security, and accessibility. This solution simplifies leave management, saving time for both employees and administrators.

1.5 Scope of the Project

The Leave Management System aims to automate the process of leave management within an organization, making it more efficient, transparent, and accessible. The scope of this project includes the development of a web-based application that allows employees to easily submit leave applications, view their leave balances, and track the status of their requests. On the other hand, administrators will have the ability to review all leave requests, approve or reject them based on company policies, and generate reports on leave utilization and employee attendance.

The system will feature role-based access control, ensuring that employees can only access their personal leave information and submit requests, while administrators will have access to the entire database to manage leave requests and generate necessary reports. Additionally, the leave types will be configurable by the administrator, allowing the system to accommodate different leave policies (e.g., annual leave, sick leave, etc.).

A key aspect of the system is its integration with MySQL for secure and efficient data storage. The leave records, employee details, and leave balances will be stored in the database, ensuring that all data is up-to-date and accessible in real-time. The back-end of the application will be powered by the Laravel PHP framework, providing a secure and scalable platform to handle user authentication, business logic, and database interactions.

The system will also include an automated leave balance update feature, which will be adjusted each time a leave request is approved. Notification systems will inform employees about the status of their leave requests, ensuring that they are always updated in real time. The system will be accessible from any device through a cloud-based deployment, ensuring remote access and eliminating the need for physical paperwork.

This project will primarily serve small to medium-sized organizations or educational institutions that require a simple yet effective leave management solution. It will reduce manual errors, improve transparency, and cut down on the time spent on administrative tasks. The project aims to provide a user-friendly experience while supporting essential leave management processes, making it a reliable tool for any organization to manage employee leaves efficiently.

Chapter 2
Software Requirement Specification

2.1 Overall Description

The Leave Management System is a web-based application designed to automate the leave request, approval, and record-keeping processes within an organization. It provides a centralized platform for employees and administrators to manage leave applications, balances, and reports efficiently and accurately. The system ensures secure login, role-based access, and real-time data updates.

2.1.1 Product Perspective

This system is a standalone web application that can be integrated with HR or payroll systems if required in the future. It follows a client-server model where the client–side is built using HTML and CSS, and the server-side is developed using Laravel (PHP framework), with MySQL as the database.

2.1.1.1 System Interfaces

- **Database Interface:** The system interacts with a MySQL database for storing user details, leave records, and system logs.
- **Authentication Interface:** Laravel's built-in authentication system manages secure login and user sessions.
- **Cloud Deployment Interface:** Supports integration with cloud services for hosting and remote access.

2.1.1.2 User Interfaces

- **Employee Panel:**
  Login/Register

  Apply for Leave

  View Leave Status

  View Leave Balance

- **Admin Panel:**
  Login

  View and Manage Leave Requests

  Approve/Reject Applications

  View Leave Reports

  Configure Leave Types and Policies

2.1.1.3 Hardware Interfaces

- **Client Side:**
  Any device (desktop/laptop/mobile) with a modern web browser (Chrome, Firefox, Edge)

  Minimum 2 GB RAM

- **Server Side:**
  Web server (e.g., Apache)

  At least 4 GB RAM, 20 GB storage

  PHP (with Laravel), MySQL installed

2.1.1.4 Software Interfaces

- **Frontend:**HTML, CSS, JavaScript (optional enhancements)

- **Backend:** PHP with Laravel Framework

- **VSCode:** Used as the primary IDE for development and debugging.
- **Database:**MySQL
- **Operating System:** Windows/Linux (for deployment)

2.1.1.5 Communications Interfaces

- HTTP/HTTPS protocol for communication between client and server.
- SMTP or Laravel notification system for email alerts (optional).
- Cloud access via internet.

2.1.1.6 Memory Constraints

- Minimal memory usage on the client side
- Server should have at least 4 GB RAM and optimized database queries to ensure performance
- Storage depends on number of users and leave records but is scalable

2.1.1.7 Operations

- Accessible via browser from any internet-enabled device
- System starts with login authentication
- Different dashboards based on user roles
- Operations include leave request, review, approval/rejection, report generation, and leave balance updates

2.1.1.8 Project Functions

- Employee registration and authentication
- Leave request form and submission
- Admin review and decision on leave applications

- Notification of leave status to employees
- Automatic leave balance management
- Role-based access to system features
- Report generation for leave history and statistics

2.1.1.9 User Characteristics

- **Employees:** Basic computer and internet usage knowledge
- **Administrators:** Familiar with HR or leave policies and basic system operations

2.1.1.10 Constraints

- Requires internet access for cloud deployment
- Limited scalability unless hosted on a scalable cloud infrastructure
- Dependent on browser compatibility
- Only supports English UI (unless localization is added)

2.1.1.11 Assumption and Dependencies

Assumptions

- Users have basic knowledge of using web applications
- Administrators are authorized to manage leave policies
- Users provide accurate information during registration and leave application

Dependencies

- Laravel framework (must be properly installed and configured)
- PHP environment
- MySQL database

- Web server (Apache/Nginx)
- Cloud server or local deployment environment

Chapter 3
System Design Specification

3.1 System Architecture

The system follows a multi-tier architecture composed of four major layers:

Presentation Layer (Client-side)

API/Controller Layer (Laravel Controllers)

Processing Layer (Business Logic)

Data Layer (MySQL Database)

3.1.0.1 Presentation Layer(Fronted)

- **Technologies Used:** HTML, CSS, Bootstrap (optional), basic JavaScript
- **Purpose:** Handles the user interface and interactions
- **Features:**
- Login/Registration page
- Dashboard for employees and admins
- Leave application form
- Leave history and balance display
- Admin panel for request management and reporting
- **UI Characteristics:**
- Responsive design for desktops and mobiles
- Simple navigation for all user roles
- Error and success message alerts

3.1.0.2 Processing Layer(Business Logic Layer)

- **Components:** Laravel services, middleware, controller logic
- **Functionality:**
- Leave request processing
- Checking leave balance
- Handling approval/rejection logic
- Notification generation
- Role-based access management
- **Security:**
- Uses Laravel's middleware for authentication
- Role checks for admin/user functionalities

3.1.0.3 Data Layer(Database Layer)

- **Database:** MySQL
- **Purpose:** Stores all persistent data including:
- User information (name, email, role, etc.)
- Leave types and policies
- Leave applications (status, dates, reasons)
- Leave balance and history
- **Structure:**
- Tables: users, leaves, leave types, notifications
- Relations: Users to Leaves (One-to-Many)

3.1.0.4 Monitoring & Logging

- **Error Logging:**
- Laravel automatically logs errors to storage/logs/laravel.log
- Custom logging can be added for key actions (login, leave actions)
- **Monitoring:**
- Laravel Telescope (optional) can be integrated for real-time app monitoring
- Web server logs (Apache/Nginx) for access and error tracking
- **Admin Insights:**
- Admin dashboard can show leave trends and status reports

3.1.0.5 Additional Components(Optional)

- **Notification System**
- Laravel supports notifications via email, database, or SMS (if configured).
- Used to alert users on leave status updates.
- **Session Management**
- Laravel's session management keeps track of logged-in users and their roles.
- **Security Measures**
- CSRF protection on forms
- Password hashing using Laravel's bcrypt
- Input validation and sanitization
- Role-based access control

3.2 Module Decomposition Description

The Leave Management System is broken down into several interrelated functional modules. Each module handles specific responsibilities to maintain modularity, clarity, and scalability of the application.

1. **User Authentication Module**

   Purpose: Handles secure login and registration of users.

   Functions:

   Employee and Admin login

   Password encryption

   Session management

   Technologies: Laravel Auth System

2. **User Management Module**

   Purpose: Manages user data and roles.

   Functions:

   Create/edit/delete user profiles (admin only)

   Assign roles (admin/employee)

   View user list and details

3. **Leave Application Module**

   Purpose: Allows employees to apply for different types of leave.

   Functions:

   Submit leave request form

   Select leave type and date range

   Add reason for leave

   View submitted applications

4. **Leave Approval Module**

   Purpose: Enables administrators to manage and process leave requests.

   Functions:

   View pending leave requests

   Approve or reject requests

   Add admin remarks or notes

   Filter/search leave records

5. **Leave Balance Management Module**

   Purpose: Tracks and updates employee leave balances.

   Functions:

   Auto-update balance after approval

   Prevent over-application

   Display remaining leave to users

6. **Notification Module**

   Purpose: Sends real-time alerts to users.

   Functions:

   Notify employees of leave status (approved/rejected)

   Optional: email or in-app alerts via Laravel Notification System

7. **Reporting Module**

   Purpose: Provides reports and summaries for admin analysis.

   Functions:

   Generate monthly/annual leave reports

   Filter data by employee, status, or date

   Export to PDF/Excel (optional)

8. **Configuration and Settings Module**

   Purpose: Allows the admin to set up leave types and system settings.

   Functions:

   Add/edit/delete leave types (e.g., sick, casual)

   Set leave policies and limits

   Manage public holidays (optional)

9. **Database Module**

   Purpose: Manages all persistent data storage and access.

   Functions:

   Maintain tables for users, leaves, balances, and logs

   Handle relational mappings (Laravel Eloquent ORM)

10. **Security Module**

    Purpose: Ensures secure and authorized access.

    Functions:

    CSRF protection

    Role-based access control

    Input validation and sanitation

Interaction Flow:

1. Employee logs in, views leave balance, and applies for leave.
2. System records the request as pending and notifies the admin.
3. Admin logs in, reviews leave applications, and approves or rejects them.
4. System updates leave status and balance, and notifies the employee.
5. Employee checks the updated status; Admin can generate reports as needed.

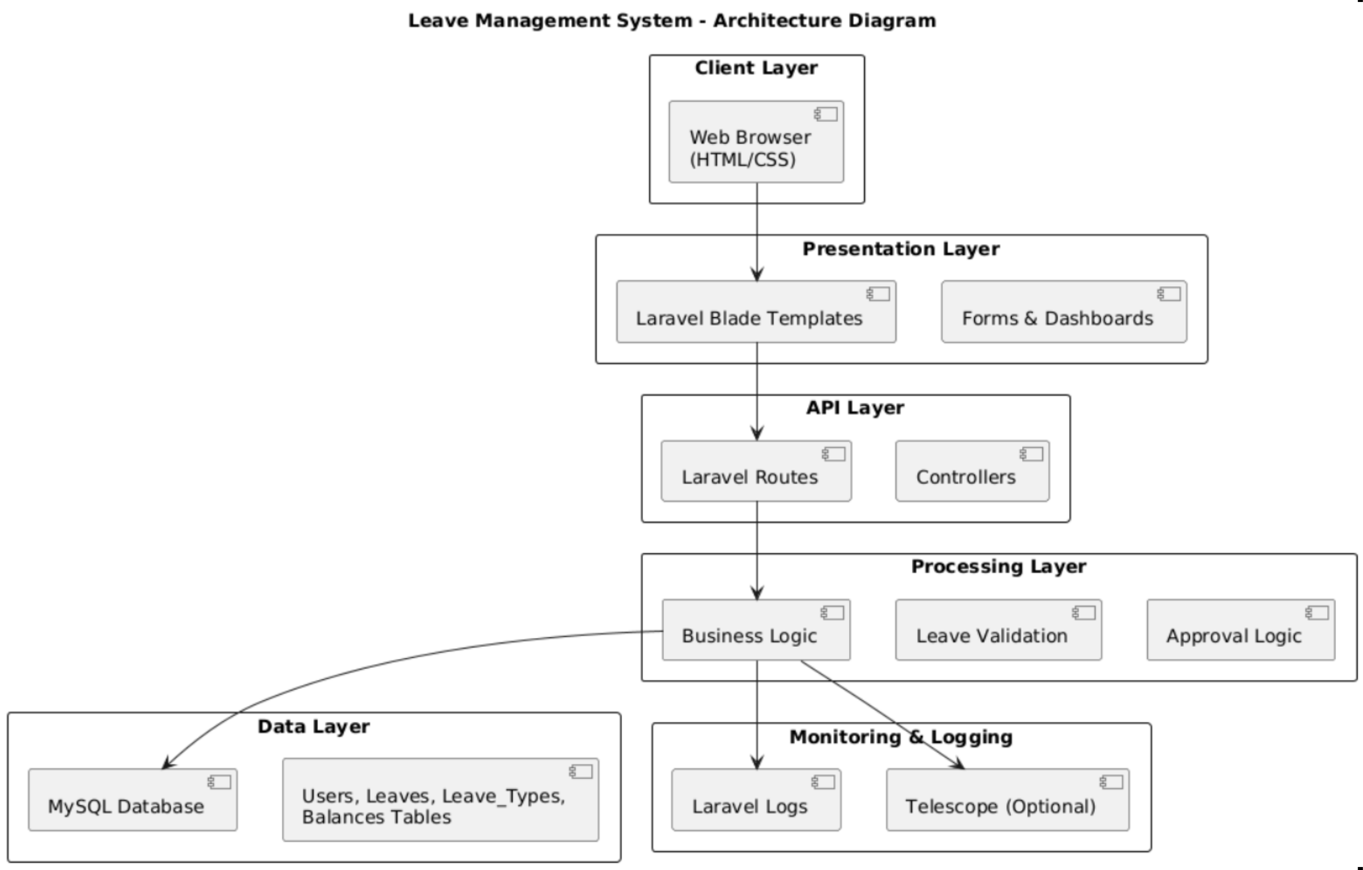


Figure 3.1: Architecture Diagram

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

A use case diagram shows how users (actors) interact with a system and what functionalities (use cases) are available to them. In the Leave Management System:

Actors:

- **Employee:** Can log in, apply for leave, check leave status and balance.
- **Admin:** Can log in, view leave requests, approve/reject leave, generate reports, and manage leave types.
- **Use Cases:** Represent actions like ”Apply for Leave”, ”Approve Leave”, ”View Reports”, etc.

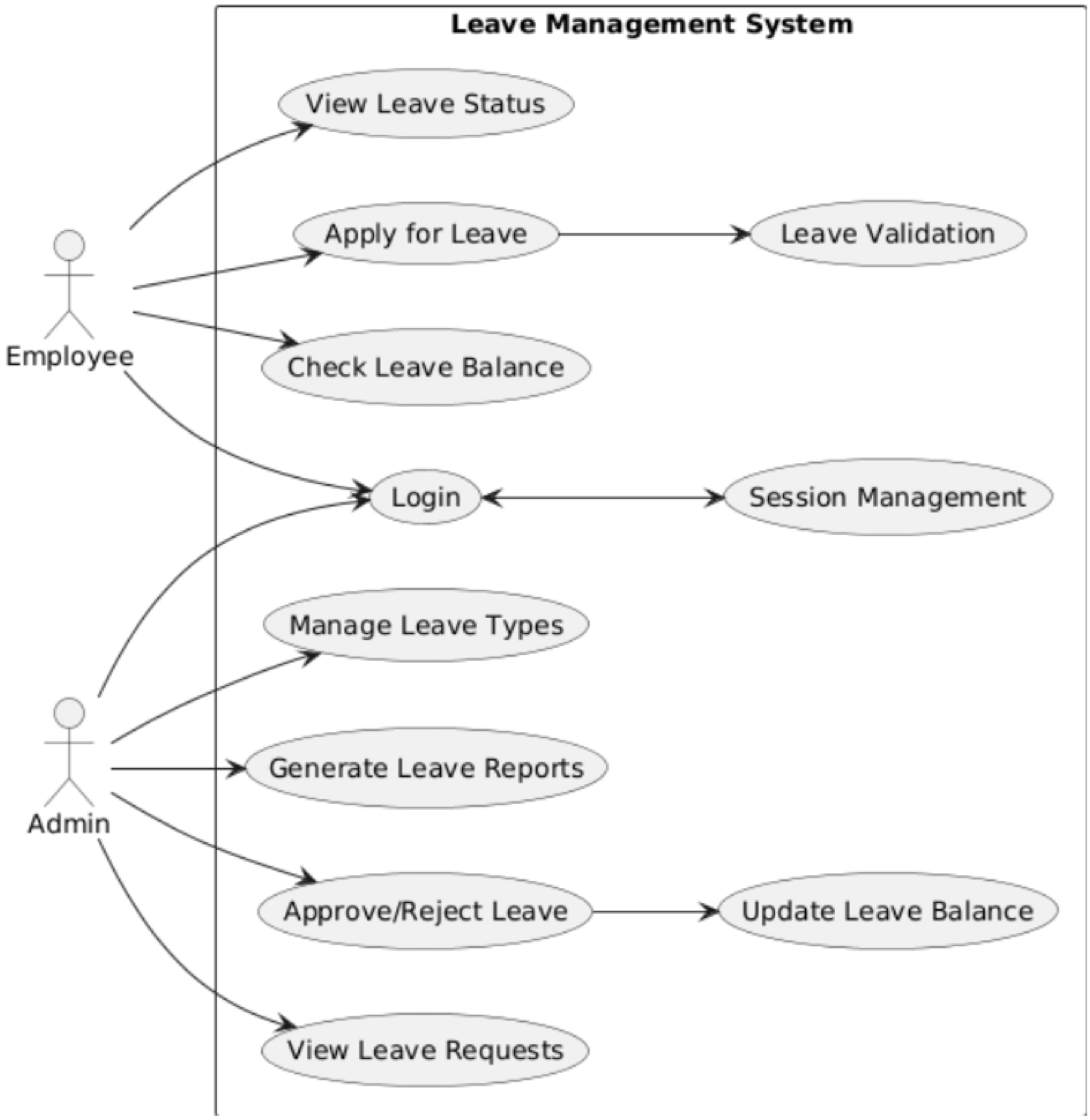


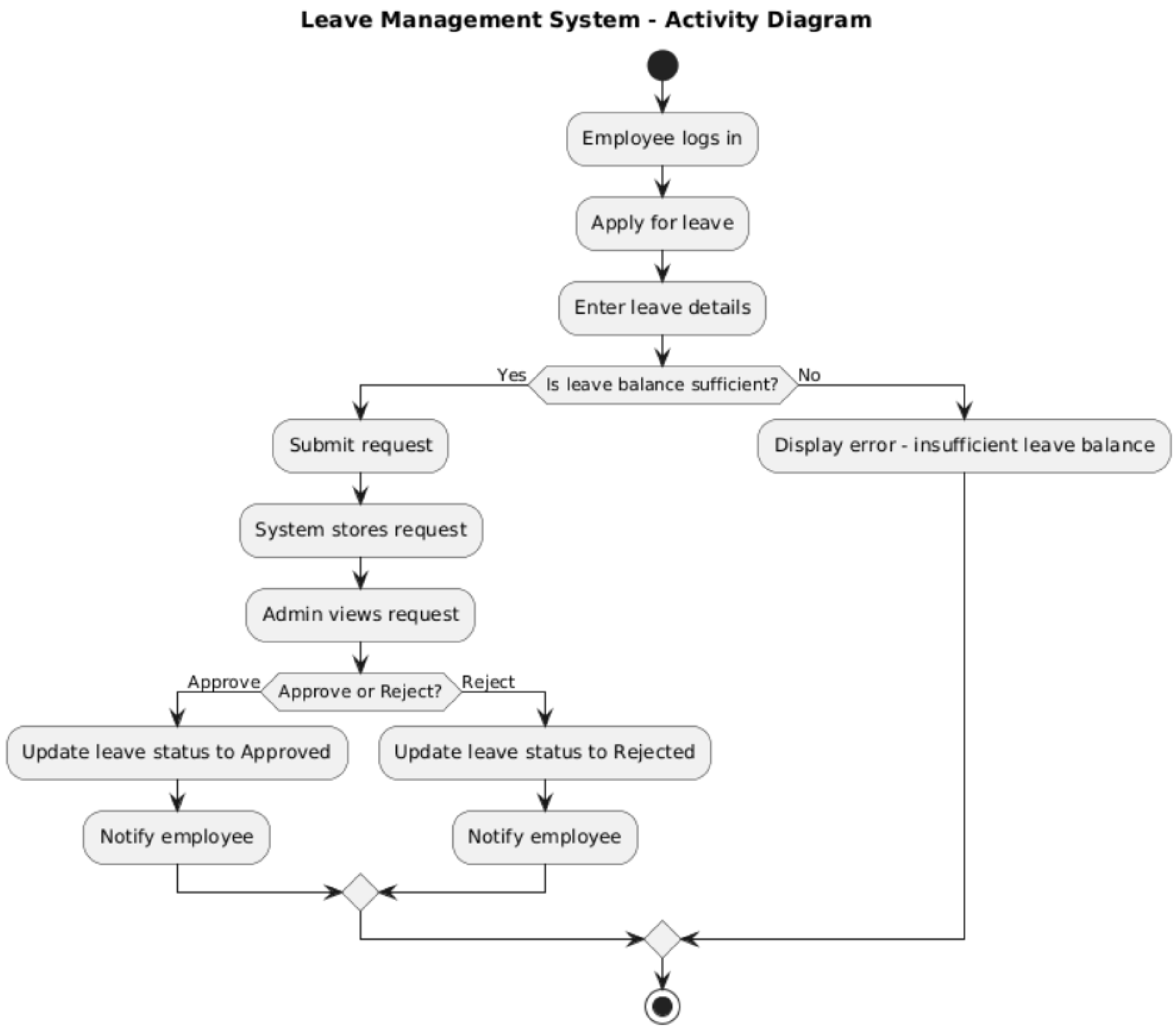
Figure 3.2: Use Case Diagram

3.3.2 Activity Diagram

An Activity Diagram represents the flow of activities or actions in a system. It shows the sequence, conditions, and decisions involved in a process. In the Leave Management System, the activity diagram helps visualize how a leave request is handled from start to end.

Key Elements in LMS Activity Diagram

- **Start Node:** Marks the beginning of the process (e.g., Employee logs in).
- **Actions:** Steps like applying for leave, validating the request, admin reviewing it, and updating status.
- **Decision Nodes:** Conditions like ”Is leave balance sufficient?” or ”Approve or Reject?”.
- **End Node:** Represents the completion of the process (e.g., Notification sent to the user).



Leave Management System - Activity Diagram
Employee logs in
Apply for leave
Enter leave details
Is leave balance sufficient?
Yes
No
Submit request
Display error - insufficient leave balance
System stores request
Admin views request
Approve or Reject?
Approve
Reject
Update leave status to Approved
Update leave status to Rejected
Notify employee
Notify employee

Figure 3.3: Activity Diagram

3.3.3 Data-Flow Diagram

A Data Flow Diagram (DFD) illustrates how data moves through a system. It shows external entities, processes, data stores, and data flows between them. In a Leave Management System, the DFD helps visualize how information such as leave requests, approvals, and user data flows between the user, admin, and database.

Main Components in LMS DFD:

- **External Entities:**
- Employee and Admin who interact with the system.
- **Processes:**
- Login, Apply for Leave, Review Leave Request, Generate Reports.
- **Data Stores:**
- User Database, Leave Requests, Leave Balances.
- **Data Flows:** Data like login credentials, leave application forms, approval decisions, and reports.

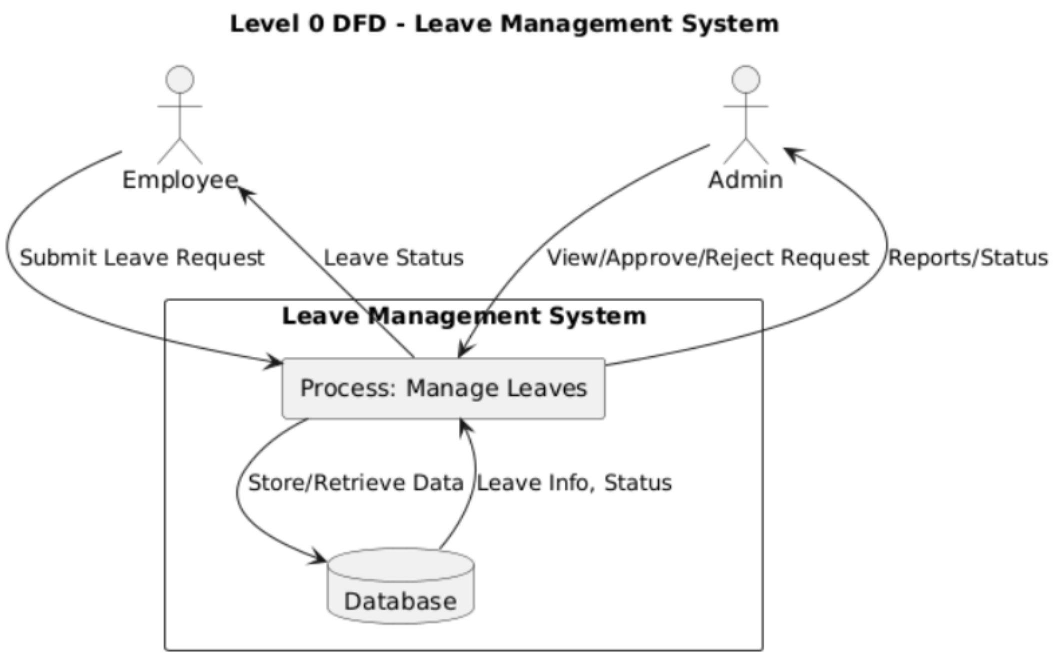


Figure 3.4: Data Flow Diagram (DFD) Level 0

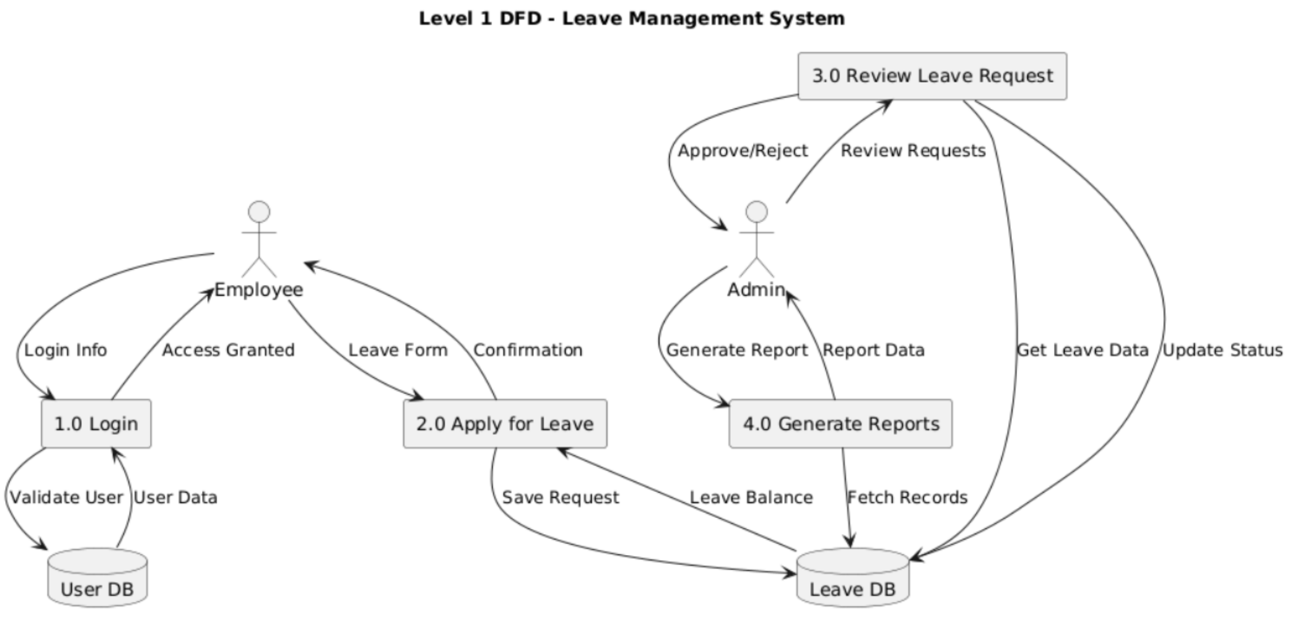


Figure 3.5: Data Flow Diagram (DFD) Level 1

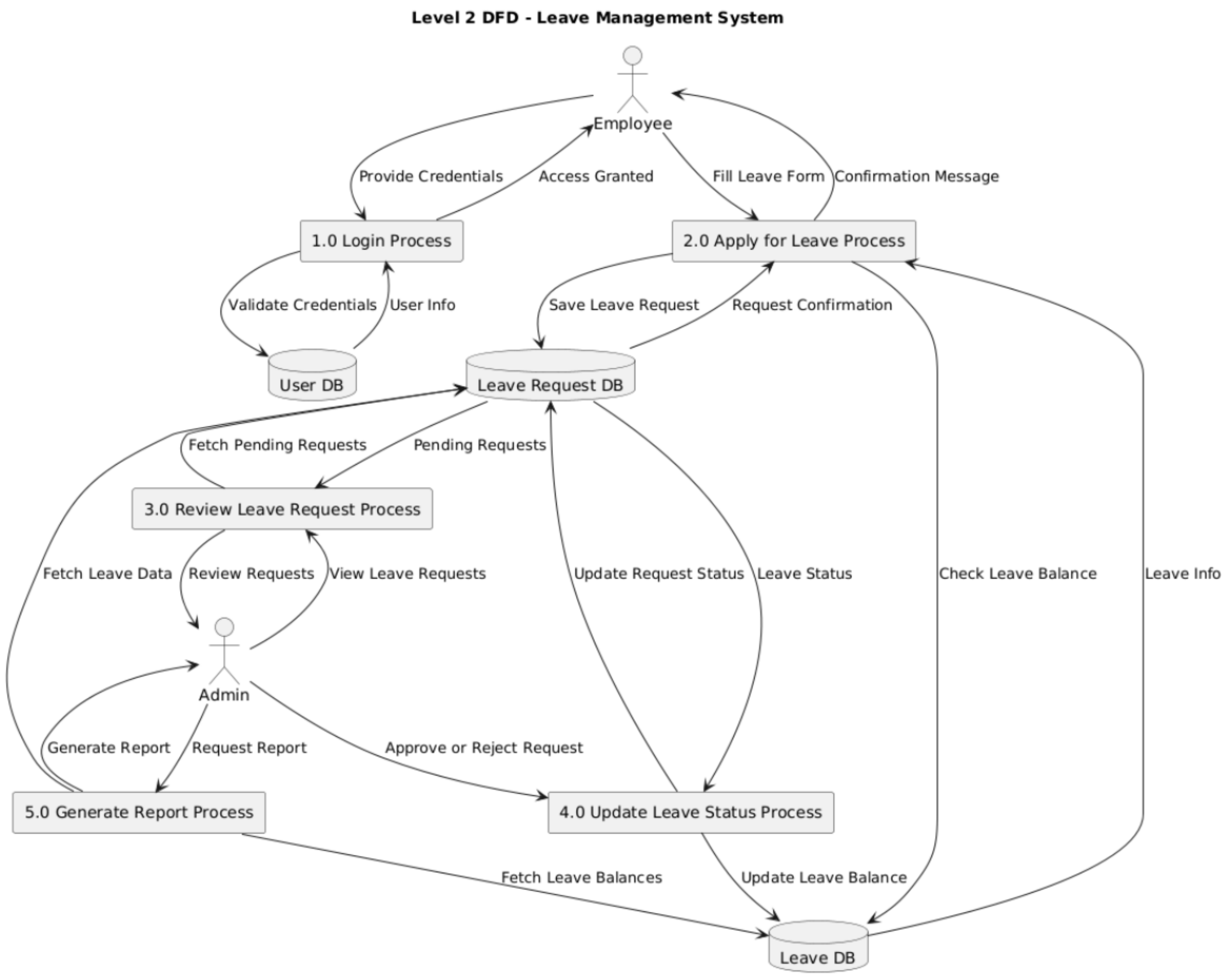


Figure 3.6: Data Flow Diagram (DFD) Level 2

3.3.4 ER Diagram

The ER diagram illustrates the relationships between the various entities in the system. The main entities include:

- Entities include Employee, Leave, Admin, and Leave Type.
- Attributes describe entities (e.g., Employee ID, Leave Status).
- Relationships show how entities interact (e.g., Employee applies for Leave, Admin approves Leave).
- Cardinality specifies how many instances of one entity relate to another (e.g., One Employee can have multiple Leaves).

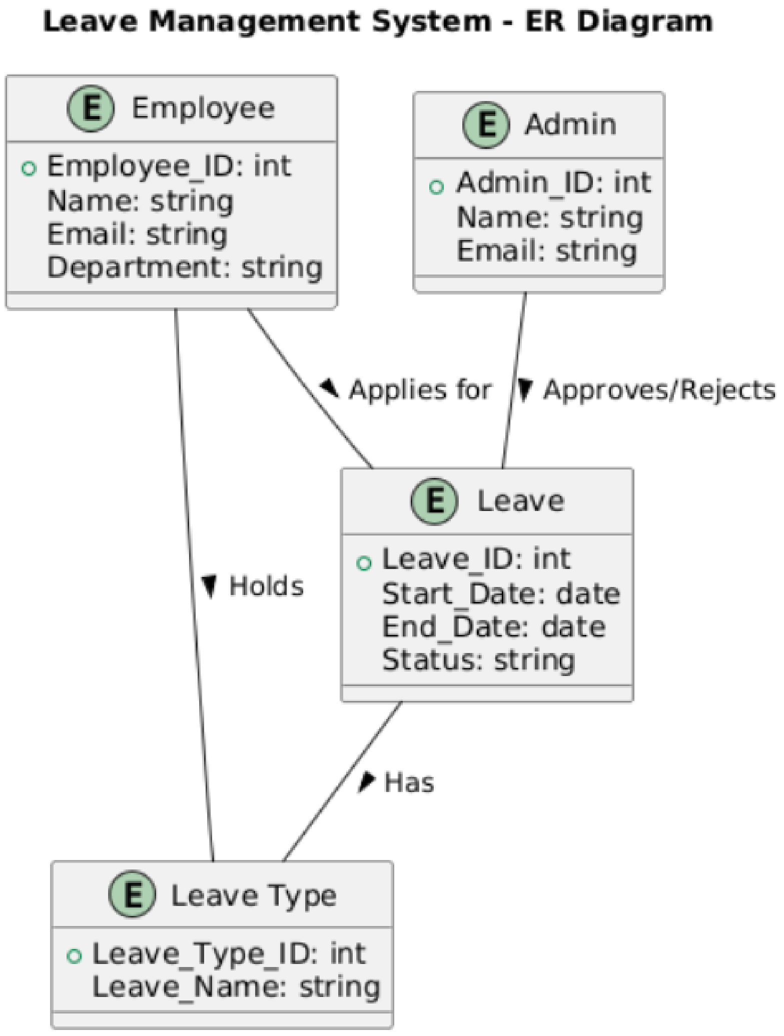


Figure 3.7: ER Diagram

3.3.5 Class Diagram

The Class Diagram shows the structure of the system in terms of classes, their attributes, methods, and relationships (associations).

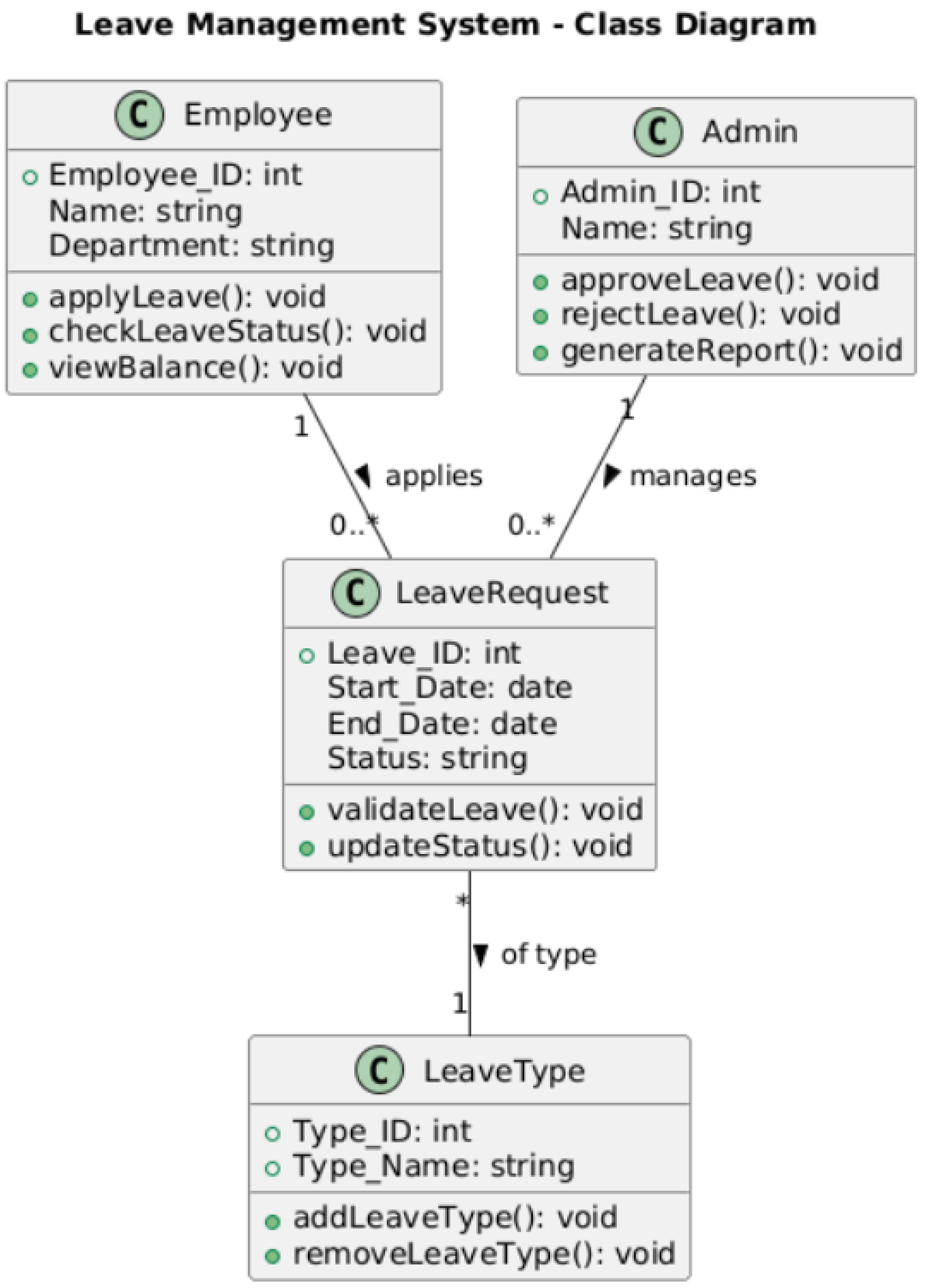


Figure 3.8: Class Diagram

3.3.6 Communication Diagram

The Communication Diagram shows how objects (such as users, the system, and external components) interact with each other through messages.

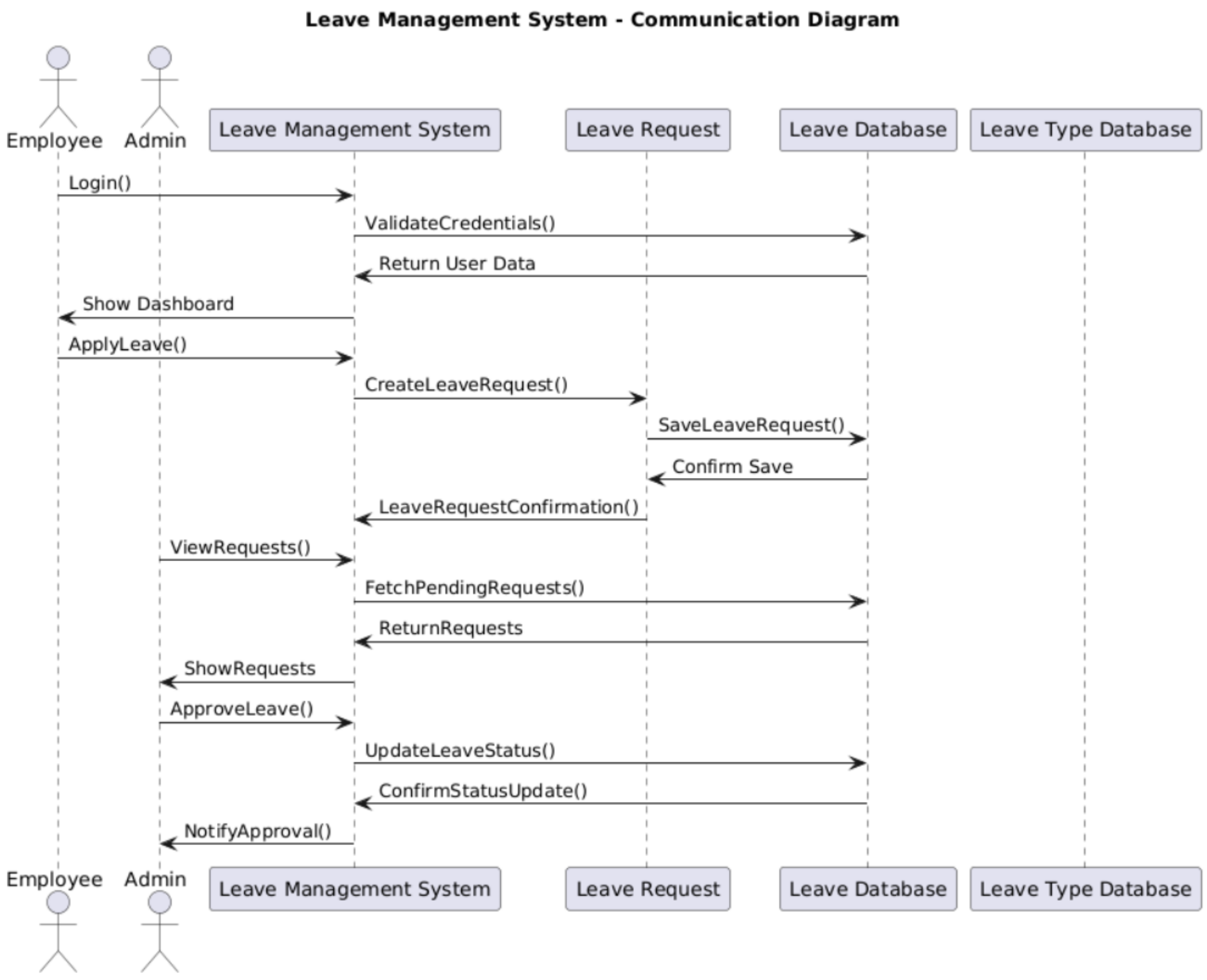


Figure 3.9: Communication Diagram

3.3.7 Sequence Diagram

A Sequence Diagram is a type of UML diagram that illustrates how objects in a system interact in a specific sequence. It shows the order of messages exchanged between objects and actors in a particular process.

- Actors (e.g., Employee, Admin) and Objects (e.g., LMS, Leave Database).
- Messages (arrows) represent communication between objects.

- Lifelines show the duration of an object's existence.
- Activation Bars indicate when an object is processing a message.

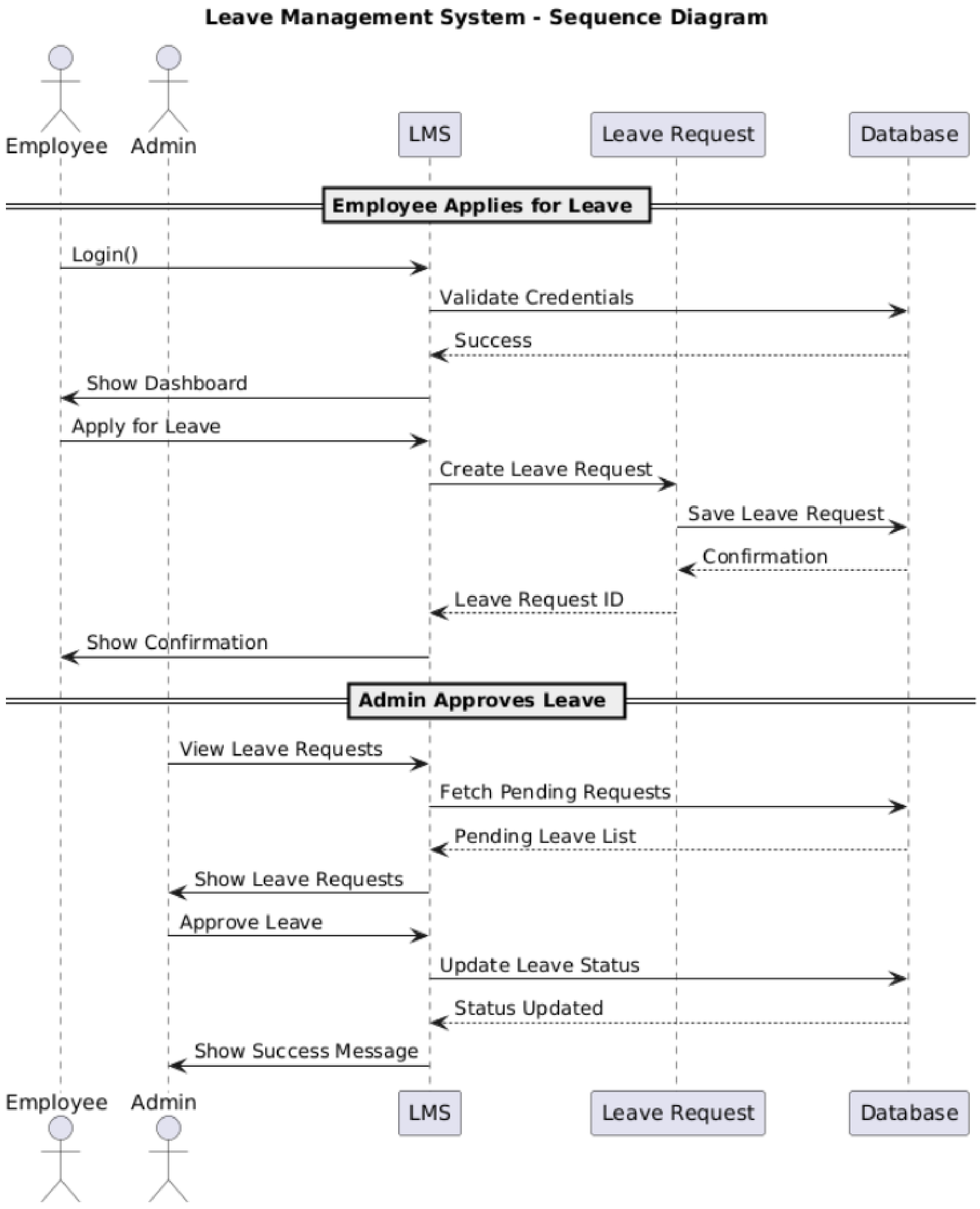


Figure 3.10: Sequence Diagram

3.3.8 Component Diagram

A Component Diagram shows the modular structure of a system by displaying its components, their interfaces, and dependencies. In a Leave Management System, it helps visualize how different parts (like frontend, backend, database, and services) are connected and interact.

- Interface (HTML/CSS): Used by employees and admins to interact with the system.
- Application Layer (Laravel/PHP): Handles business logic like leave processing and validation.
- Database Component (MySQL): Stores user data, leave requests, leave types, etc.
- Authentication Module: Verifies login and roles.
- Leave Management Module: Manages creation, approval, and status of leave requests.

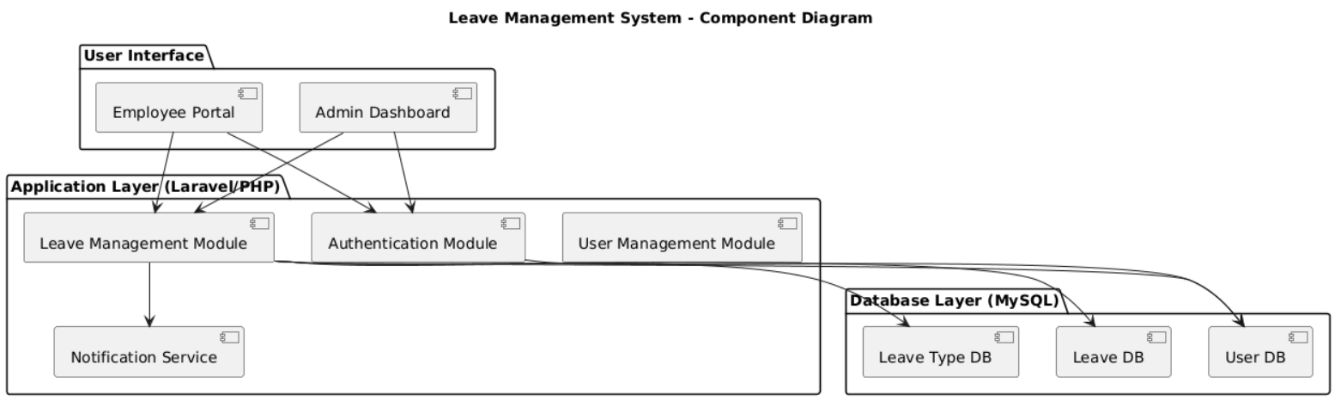


Figure 3.11: Component Diagram

3.3.9 Deployment Diagram

A Deployment Diagram shows the physical arrangement of hardware (nodes) and software (artifacts) in a system. It illustrates how different system components are deployed across servers, clients, and devices, and how they interact.

- **In LMS, the diagram typically includes:**
- Client Node: Devices like desktops or mobiles used by employees/admins to access the system via a browser.
- Web Server Node: Hosts the Laravel/PHP application, handling business logic and API requests.
- Database Server Node: Runs the MySQL server to store and retrieve data (user info, leave requests, etc.)

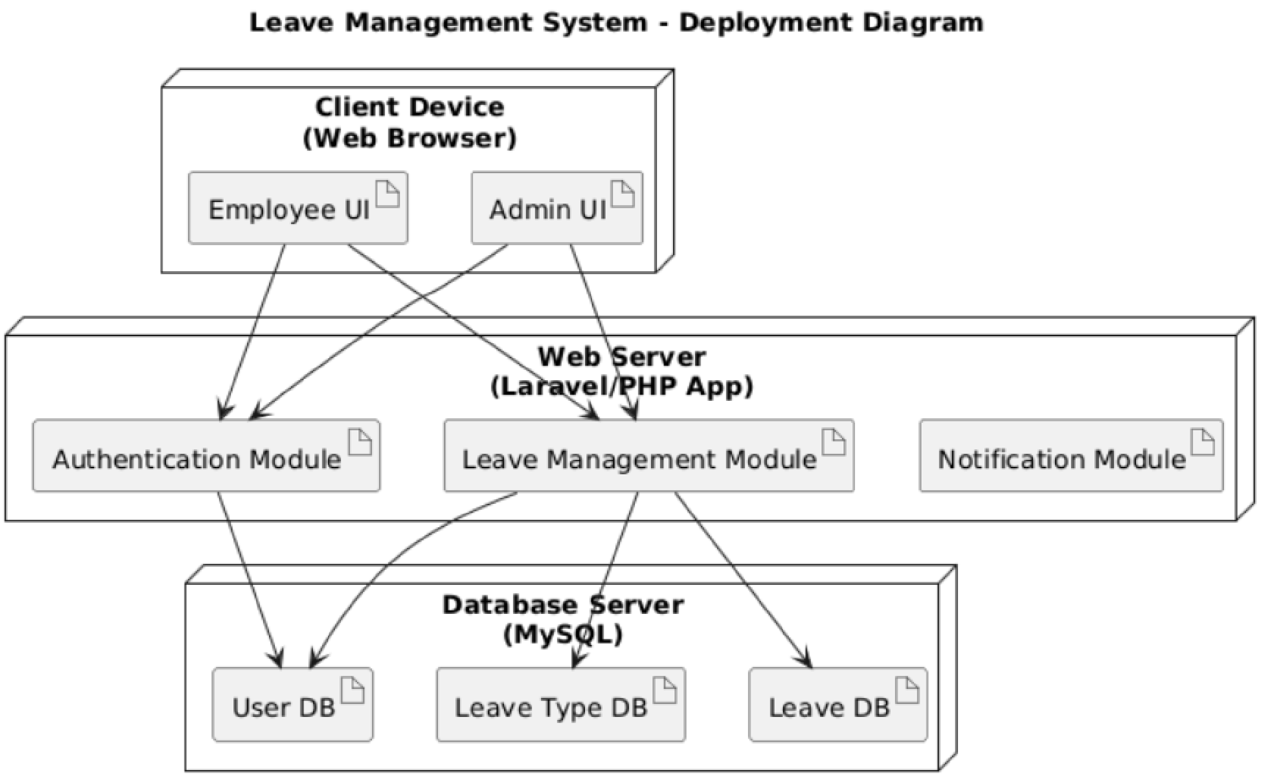


Figure 3.12: Deployment Diagram

3.3.10 Business Process Model

A Business Process Model (BPM) represents the workflow or sequence of business activities carried out to achieve a specific goal. In the context of a Leave Management System, it illustrates how leave requests are handled from initiation to approval or rejection.

- **Key Steps in LMS BPM:**

- Employee logs in to the system.
- Applies for leave by filling in details like dates and type.
- System validates the request.
- Leave request is forwarded to the Admin or Manager.
- Admin reviews and approves/rejects the request.
- Employee is notified of the decision.

The business process model shows the end-to-end process flow of the system.

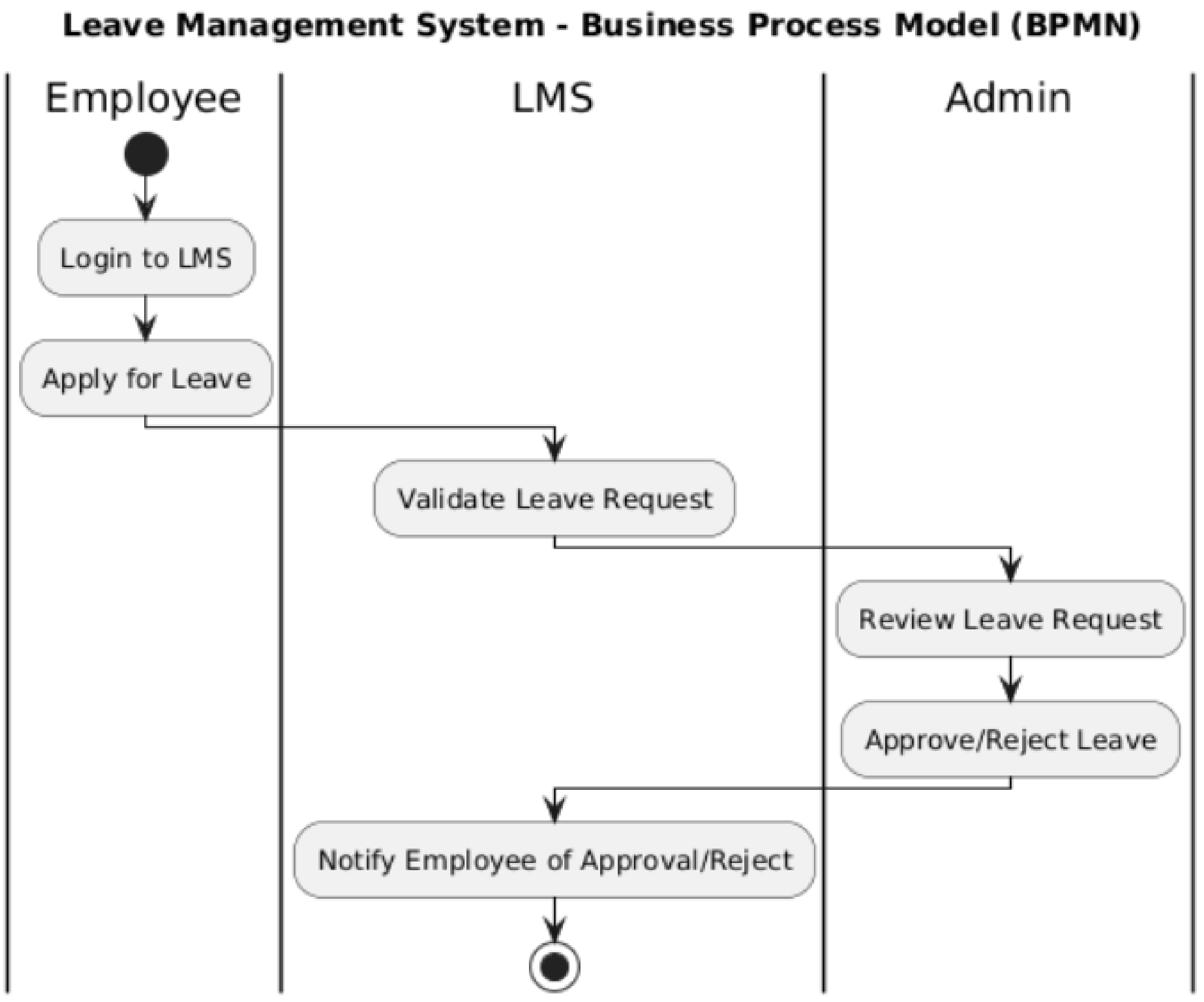


Figure 3.13: Business Process Model

Chapter 4

Methodology and Team

4.1 Introduction to Waterfall Framework

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In ”The Waterfall” approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as an input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

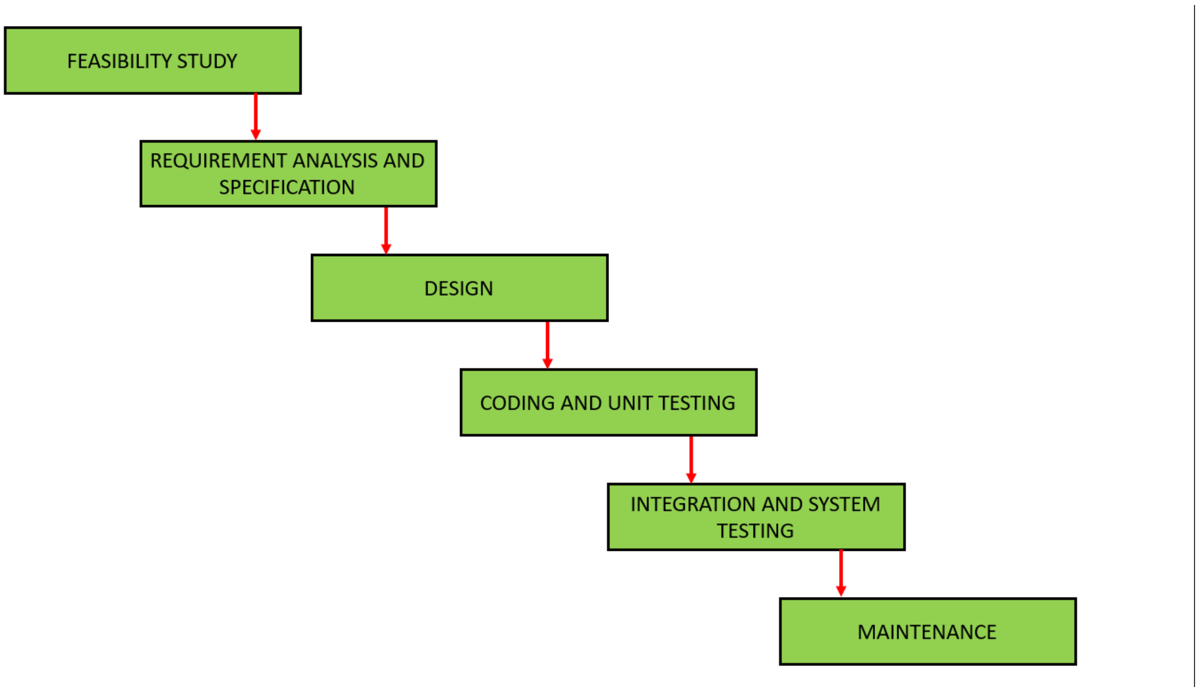

FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

Figure 4.1: WaterFall model

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

Waterfall Model Pros & Cons

Advantage The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

4.2 Team Members, Roles & Responsibilities

The project was completed by a team of three students, with well-defined roles assigned to each member.

- **Yashpal Siyag (21ESKIT308) — Frontend Developer**

- Design and implement user interfaces using HTML, CSS, and JavaScript.
- Create responsive pages for employee and admin dashboards, login/signup, and leave application forms.
- Ensure user-friendly design and proper input validation on the client side.
- **Sonu Saini (21ESKIT300) — Backend Developer**

- Develop the core logic using the Laravel PHP framework.
- Handle routing, controller logic, authentication, and leave processing workflows.
- Integrate frontend with backend via proper endpoints and manage security.
- **Rahul Ahuja (21ESKIT302) — Database Manager**

- Design and manage the MySQL database schema for users, leave requests, roles, etc.
- Write optimized queries, handle migrations, and ensure data integrity.
- Assist in database connectivity with the Laravel backend and manage backups.

Chapter 5

Centering System Testing

The designed system has been testing through following test parameters.

5.1 Functionality Testing

Functional testing focuses on verifying that each function of the LMS operates in accordance with the defined specifications and requirements. It ensures the software behaves as expected under specific conditions.

- **Purpose:** To validate the core functionalities of the system such as leave application, approval workflows, user login, and role management.
- **Scope in LMS:**
- Login/Logout functionality for employees and admins.
- Leave request process – submission, approval, rejection.
- Role-based access – only admins can approve leave; employees can only request.
- Leave history and status display.
- Input validation – ensuring no incorrect or incomplete form submissions.
- Email or notification system – confirming actions like approval or rejection.
- **Approach:**
- Use test cases to validate each user action.
- Check expected vs actual output for all functional modules.

- Perform positive and negative testing (e.g., submitting invalid leave dates).
- **Example:**
- Test Case: Employee logs in $\rightarrow$ navigates to leave form $\rightarrow$ submits request $\rightarrow$ admin receives the request for action $\rightarrow$ employee gets a notification after admin's decision.

5.2 Performance Testing

Performance testing determines how the LMS performs under expected and peak workloads. It measures system stability, responsiveness, speed, and resource usage.

- **Purpose:** To ensure the LMS works efficiently without performance bottlenecks, especially when accessed by many users or during peak operations (e.g., around holidays).
- **Scope in LMS:**
- Response time of login, dashboard, and leave application pages.
- Load testing for concurrent users submitting or viewing leave data.
- Stress testing to assess the system's breaking point.
- Database performance when handling large amounts of leave data.
- **Approach:**
- Use tools like Apache JMeter, LoadRunner, or browser dev tools for monitoring.
- Simulate high user traffic and measure performance metrics.
- **Example:**
- Simulate 500 employees logging in and applying for leave simultaneously to test server response and database handling capacity.

5.3 Usability Testing

Usability testing evaluates the LMS's user interface (UI) and user experience (UX) to ensure it is easy, intuitive, and satisfactory for end users.

- **Purpose:** To ensure that both technical and non-technical users (employees and admins) can navigate and use the system without confusion or extensive training.
- **Scope in LMS:**
- Clarity of menus, buttons, and form fields.
- Logical navigation across pages (e.g., from dashboard to leave application).
- Consistency in design (colors, fonts, layout).
- Accessibility across devices (responsive design).
- User guidance in case of errors or missing input.
- **Approach:**
- Conduct testing with real users (e.g., fellow students or staff).
- Use feedback to improve interface and fix confusing elements.
- Track actions to see how easily tasks are completed.
- **Example:**
- Give a user the task of applying for leave. If they can complete it easily without external help, the system passes the usability test.

Chapter 6
Test Execution Summary

The testing of the Leave Management System was conducted to ensure the functionality, performance, and usability of the application. Functional testing included validating core features such as login, leave application, approval, and role-based access, where 18 out of 20 test cases passed and the remaining minor issues were fixed. Performance testing confirmed that the system responded well under simulated load conditions without lag. Usability testing showed that the user interface was mostly intuitive, with slight adjustments made based on feedback for better user experience. Overall, the system met the expected quality standards, and all critical modules performed as required, making the LMS stable and ready for deployment.

graphicx

Table 6.1: Test Execution Summary for Leave Management System

Test Case ID	Module	Test Scenario	Expected Result	Status	Remarks
TC01	Login	Valid user login	Dashboard should open	Passed	-
TC02	Login	Invalid login credentials	Error message should display	Passed	-
TC03	Leave Application	Apply leave with valid inputs	Leave request submitted	Passed	-
TC04	Leave Application	Apply leave with past date	Show validation error	Passed	-
TC05	Leave Approval	Admin approves leave	Status updates to ”Approved”	Passed	-
TC06	Leave Approval	Admin rejects leave	Status updates to ”Rejected”	Passed	-
TC07	Dashboard	Employee views leave history	Leave records should display	Passed	-
TC08	Dashboard	Admin views all requests	All requests display correctly	Passed	-
TC09	Usability	Navigation across modules	Smooth and intuitive experience	Passed	UI updated
TC10	Performance	50 concurrent users apply	System responds without lag	Passed	Load handled well
TC11	Security	Unauthorized admin access	Access denied message	Passed	-
TC12	Compatibility	Access on mobile	Responsive layout adapts	Passed	-

Chapter 7

Project Screen Shots

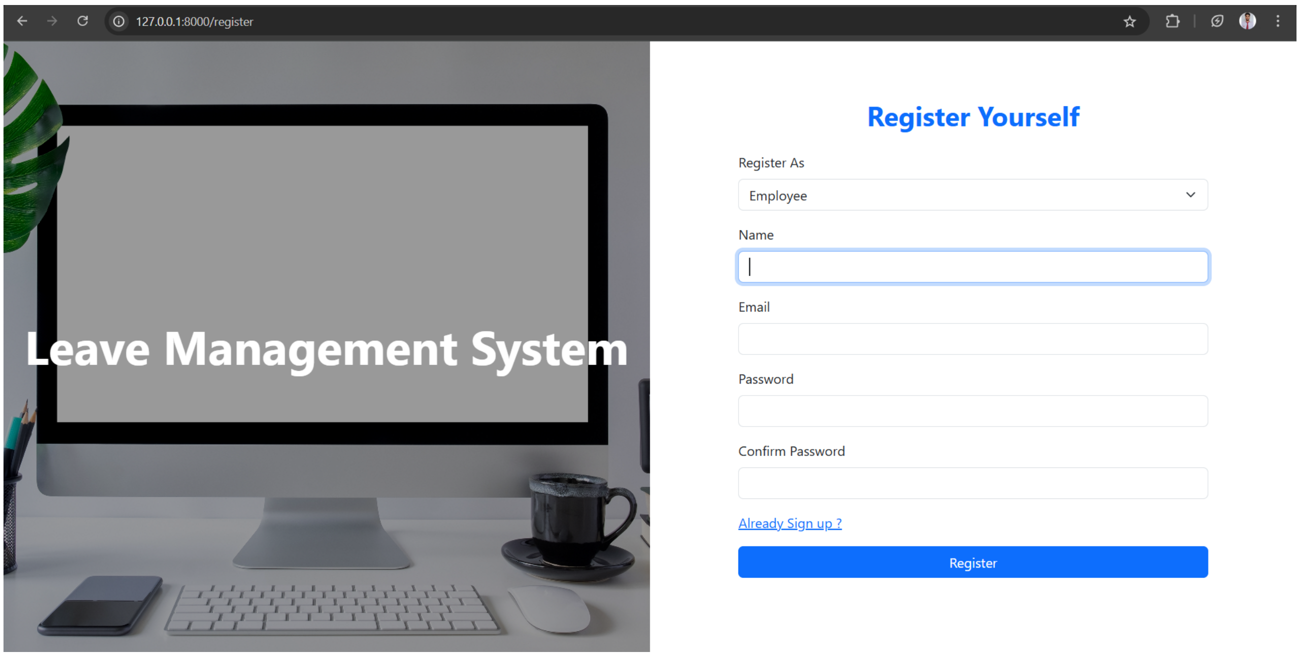


Figure 7.1: Register Page

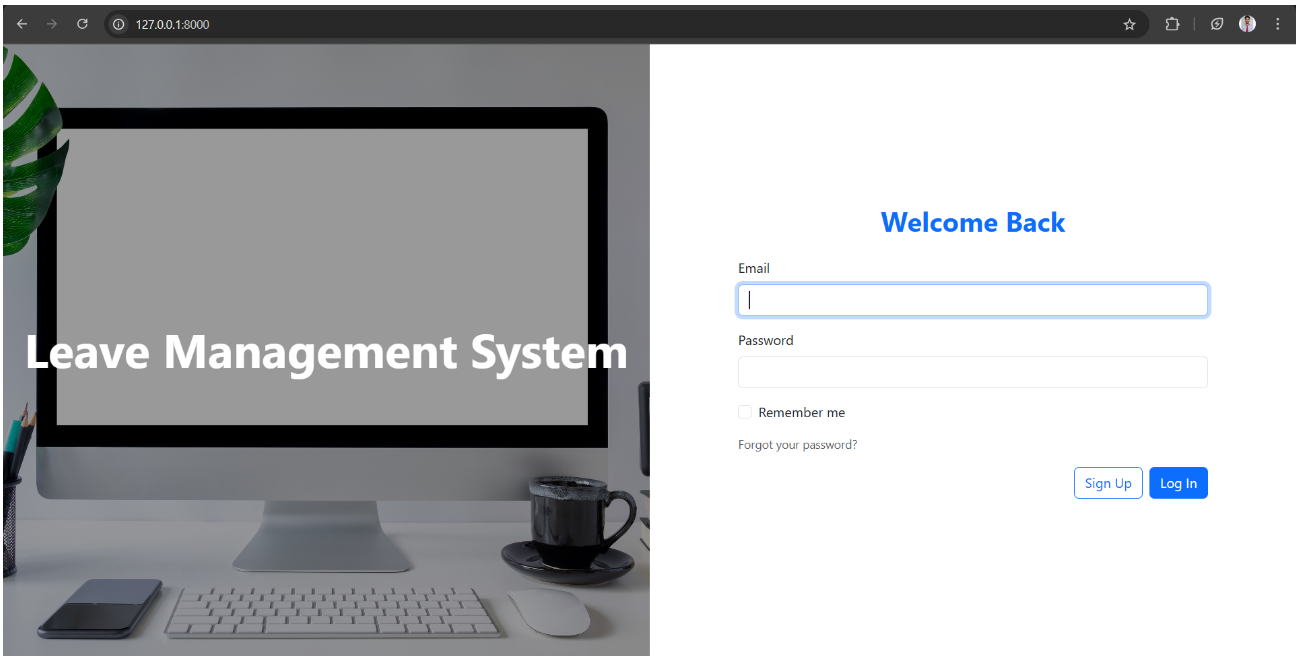
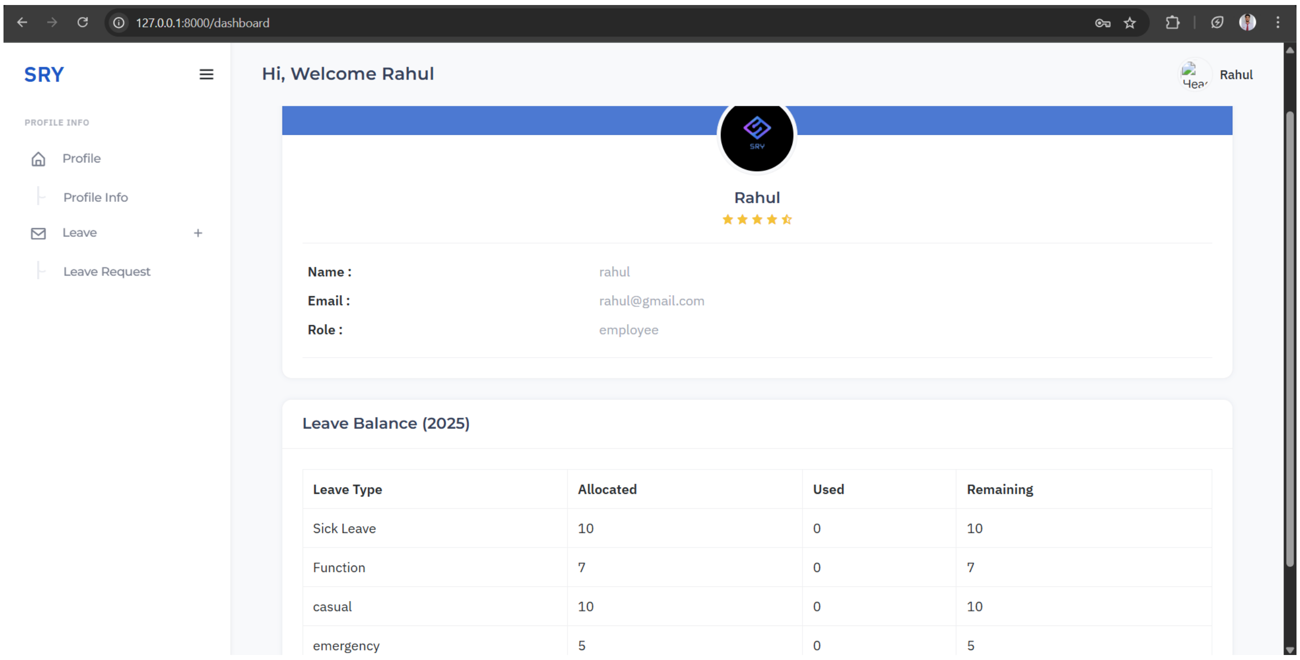


Figure 7.2: Login Page


127.0.0.1:8000/dashboard

SRY

PROFILE INFO

Profile

Profile Info

Leave

Leave Request

Hi, Welcome Rahul

Rahul

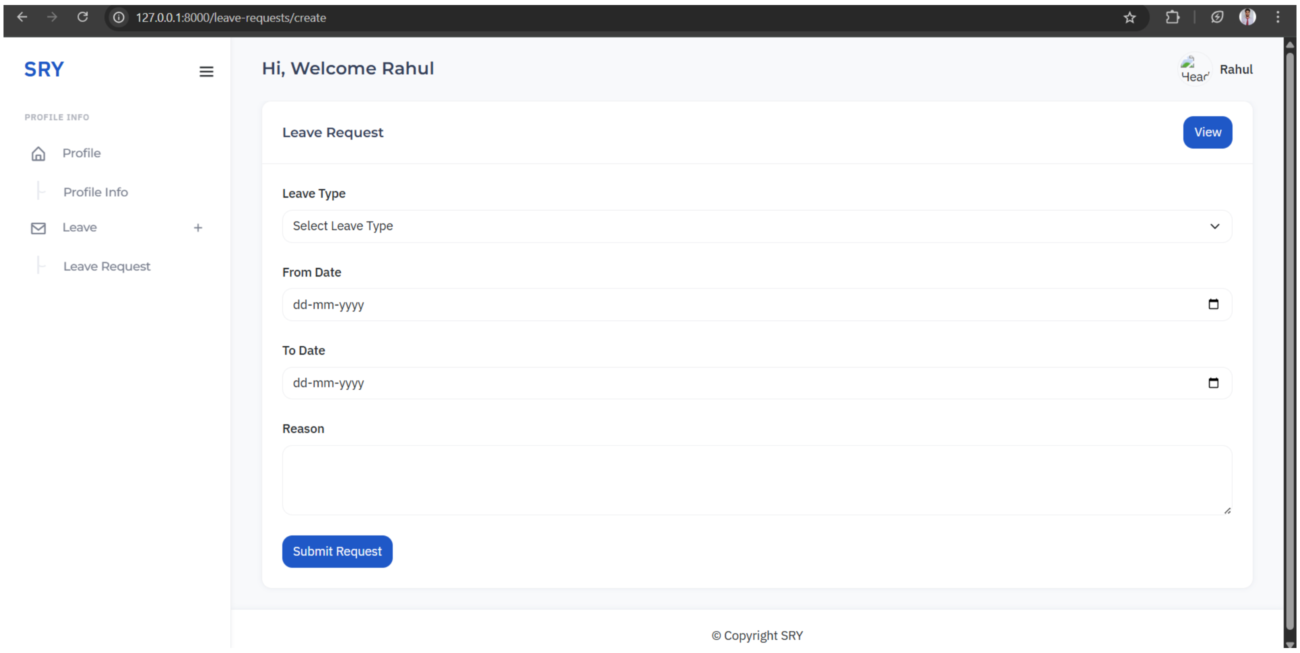
Rahul

Name :	rahul
Email :	rahul@gmail.com
Role :	employee

Leave Balance (2025)

Leave Type	Allocated	Used	Remaining
Sick Leave	10	0	10
Function	7	0	7
casual	10	0	10
emergency	5	0	5

Figure 7.3: Employee Dashboard


127.0.0.1:8000/leave-requests/create

SRY

PROFILE INFO

Profile

Profile Info

Leave

Leave Request

Hi, Welcome Rahul

Rahul

Leave Request

View

Leave Type

Select Leave Type

From Date

dd-mm-yyyy

To Date

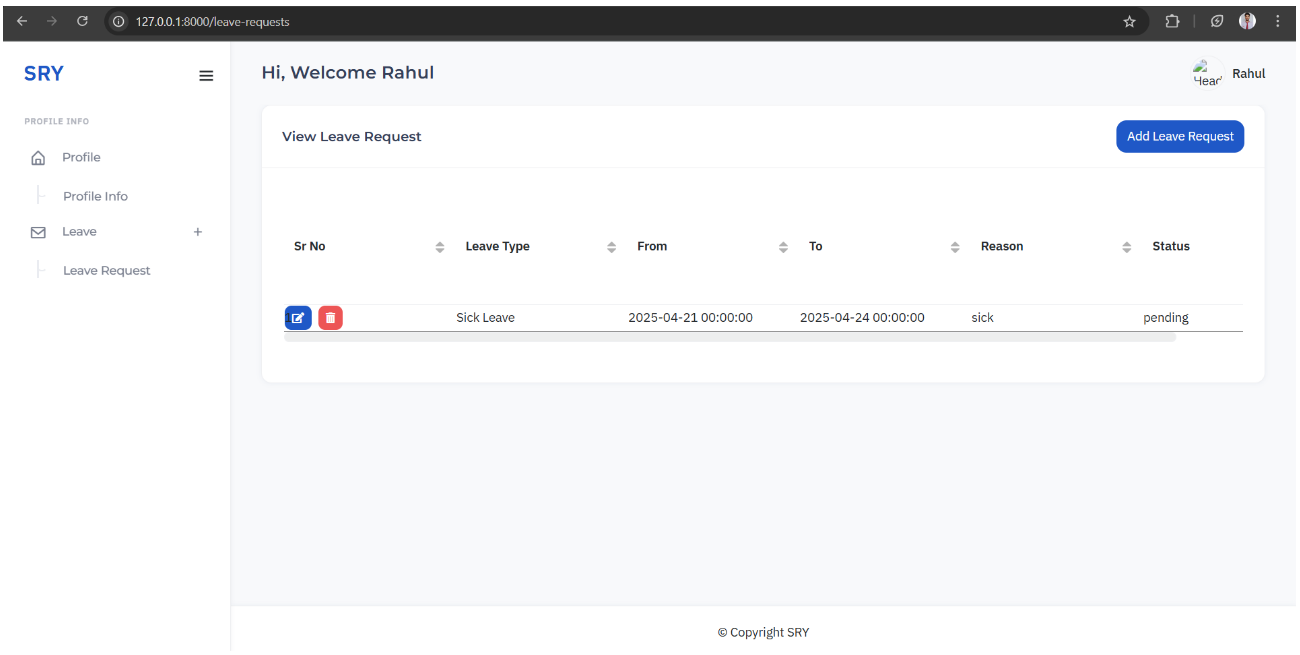
dd-mm-yyyy

Reason

Submit Request

© Copyright SRY

Figure 7.4: Leave Request Page


127.0.0.1:8000/leave-requests

SRY

Hi, Welcome Rahul

Rahul

PROFILE INFO

Profile

Profile Info

Leave

Leave Request

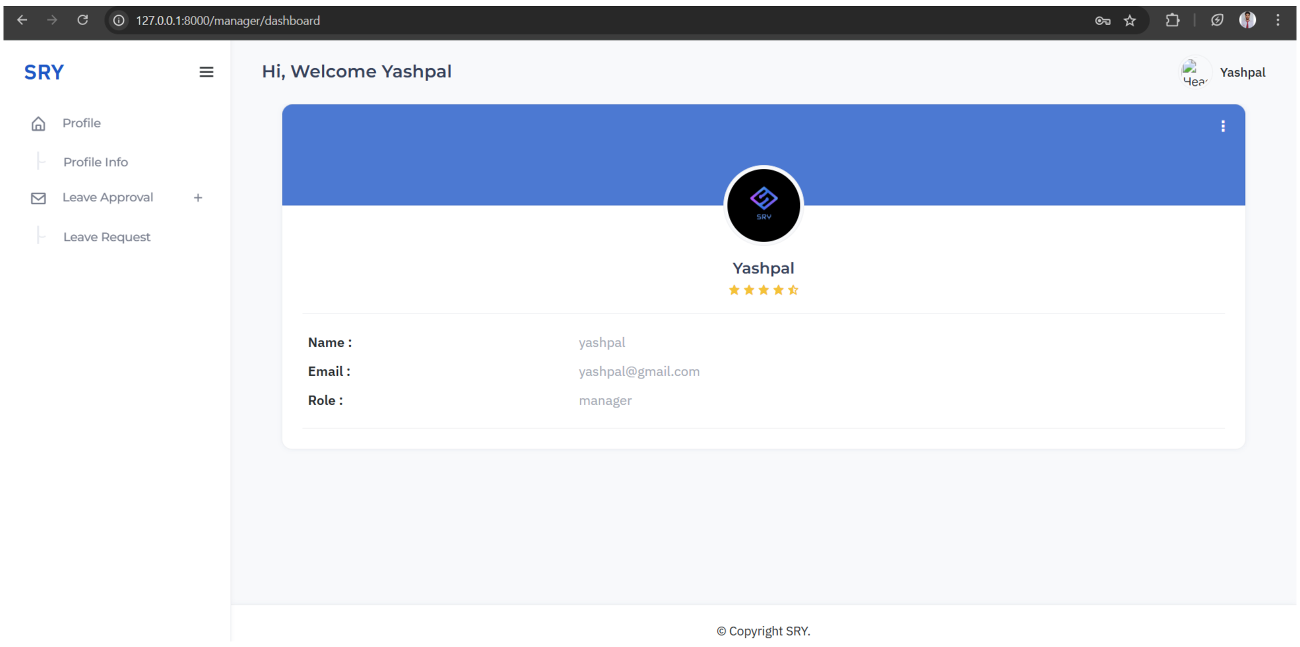
View Leave Request

Add Leave Request

Sr No	Leave Type	From	To	Reason	Status
	Sick Leave	2025-04-21 00:00:00	2025-04-24 00:00:00	sick	pending

© Copyright SRY

Figure 7.5: Employee View Leave Request


127.0.0.1:8000/manager/dashboard

SRY

Hi, Welcome Yashpal

Yashpal

Profile

Profile Info

Leave Approval

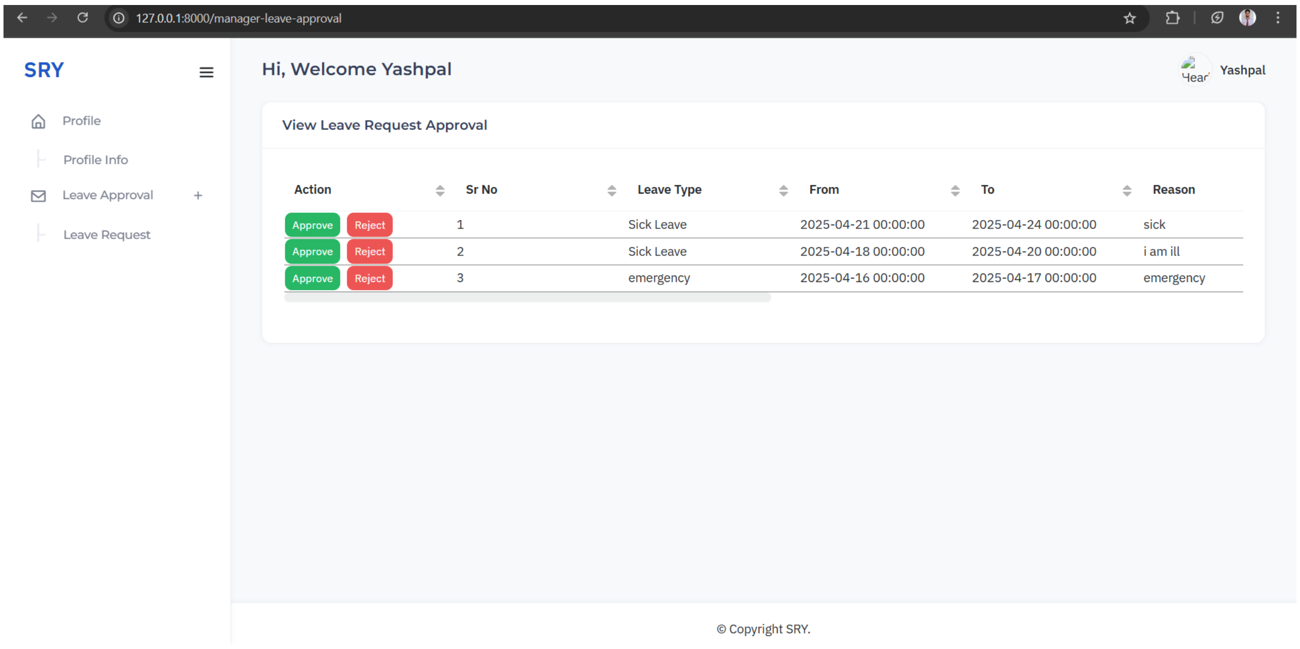
Leave Request

Yashpal

Name :	yashpal
Email :	yashpal@gmail.com
Role :	manager

© Copyright SRY.

Figure 7.6: Manager Dashboard



Action	Sr No	Leave Type	From	To	Reason
Approve Reject	1	Sick Leave	2025-04-21 00:00:00	2025-04-24 00:00:00	sick
Approve Reject	2	Sick Leave	2025-04-18 00:00:00	2025-04-20 00:00:00	i am ill
Approve Reject	3	emergency	2025-04-16 00:00:00	2025-04-17 00:00:00	emergency

Figure 7.7: Manager Approve/Reject Leave Request

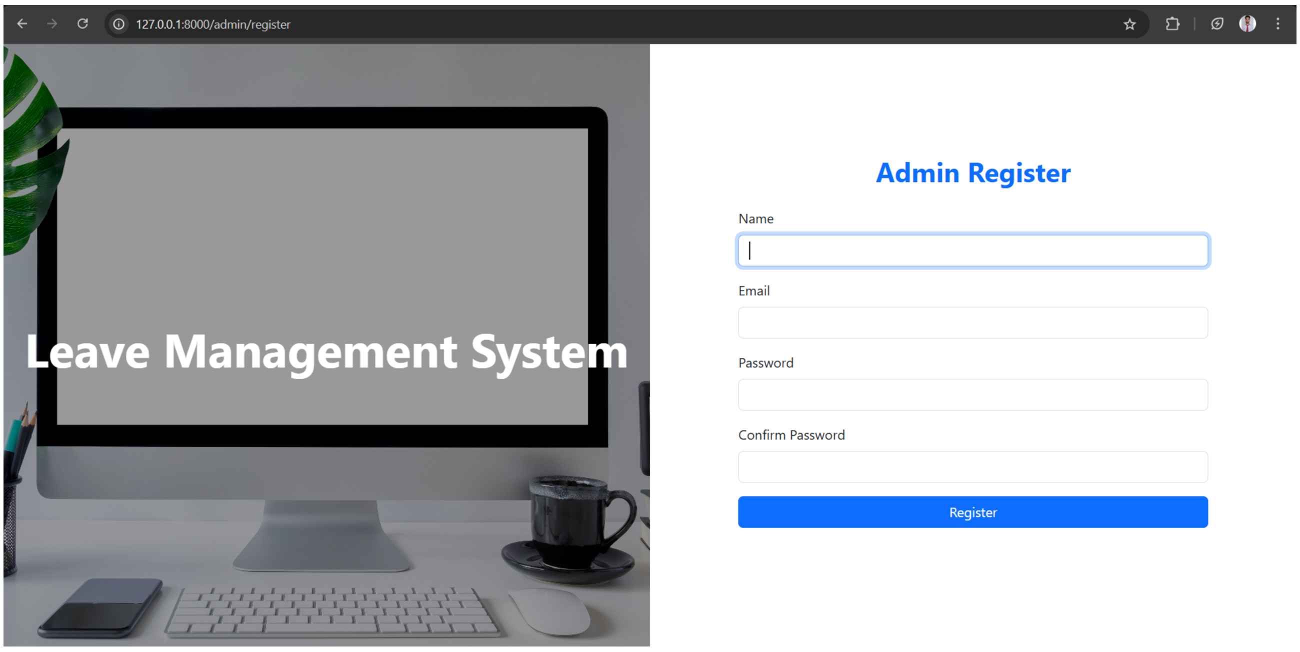


Figure 7.8: Admin Register

Chapter 8
Project Summary and Conclusions

8.1 Project Summary

The Leave Management System is a web-based application developed to automate the process of leave requests and approvals in an organization. It aims to replace the manual, paper-based leave management system, which is often prone to errors and inefficiencies, with a streamlined, digital solution that is easy to use and manage.

The system leverages the following key technologies:

- **Fronted:** HTML, CSS, and JavaScript (for dynamic interactions)
- **Backend:** Laravel PHP framework (for routing, controllers, and business logic)
- **Database:** MySQL (to manage and store leave records, user information, and other system data)

The system is divided into user roles: employees can apply for leave, check their leave balance, and view the status of their requests, while administrators can approve, reject, or modify leave requests. The application provides a responsive interface, making it accessible on both desktop and mobile devices.

The development team consisted of three students, each taking responsibility for a specific part of the system: one focused on the frontend, another on the backend, and the third on database management. The system was thoroughly tested for functionality, performance, and usability, ensuring it met the required standards.

8.2 Conclusion

The Leave Management System (LMS) successfully achieves its goal of simplifying the leave application and approval process within organizations. By eliminating paper-based leave management, it improves efficiency, accuracy, and record-keeping. Employees can easily apply for leave and track the status of their requests, while administrators can manage all leave applications with ease.

The technologies used, particularly Laravel for backend development and MySQL for database management, allowed for the creation of a scalable and secure system. The application is fully functional, user-friendly, and adaptable to various organizational needs. The development team has gained valuable experience in modern web technologies and full-stack development.

Looking forward, the system could be further enhanced by adding features such as automated email/SMS notifications, mobile app support, analytics dashboards, and integration with HR or payroll systems. These improvements would further streamline leave management and improve user engagement. Overall, the LMS serves as an effective solution for modernizing leave management in any organization.

Chapter 9

Future Scope

The Leave Management System (LMS) has the potential to be further enhanced and expanded to meet evolving organizational needs and technological advancements. Some of the key areas where the system could be improved and extended in the future include:

- **Mobile Application Development**
  Mobile app support could be developed to allow employees and administrators to manage leave requests and approvals on the go. This would improve accessibility and convenience for users, especially in remote or field-based environments.
- **Email and SMS Notifications**
  Automated email and SMS notifications for leave approvals, rejections, or reminders for pending requests would increase user engagement and provide better communication throughout the process.
- **Integration with Payroll and HR Systems**
  The LMS can be integrated with existing HR management and payroll systems to automate the calculation of leave balances and payroll deductions based on leave taken. This would enhance the efficiency of payroll processing and reduce manual errors.
- **Analytics and Reporting**
  Analytics dashboards can be added to provide insights on leave patterns, usage statistics, and reports for both employees and administrators. This feature would help organizations track employee leave trends and optimize resource planning.

- **Role-Based Access Control (RBAC) Enhancements**
  Further enhancements to the role-based access control (RBAC) system could allow for more granular permissions and customizable workflows. This would cater to larger organizations with complex hierarchical structures.

- **AI-based Leave Prediction and Suggestions**
  Implementing AI and machine learning algorithms could help predict peak leave periods, suggest optimal leave times based on workload, and prevent operational disruptions by analyzing leave patterns across the organization.

- **Leave Policy Management**
  A module for leave policy management can be introduced, enabling administrators to define different types of leave (e.g., sick leave, casual leave, vacation) with configurable policies for each type (e.g., eligibility, approval hierarchy, and quotas).

- **Multi-Language Support**
  Adding multi-language support would make the system more accessible for employees from diverse regions or non-English-speaking backgrounds, improving its adoption across various geographical locations.

- **Cloud Integration**
  Moving the system to a cloud-based environment can provide better scalability, disaster recovery, and data redundancy. It would also allow for seamless updates and improved performance, especially for organizations with large numbers of users.

- **AI-based Fraud Detection**
  Advanced features such as fraud detection could be integrated into the system to monitor unusual leave patterns, such as employees frequently taking leaves on specific dates, and flag suspicious activity to the administrators.

By integrating these enhancements, the Leave Management System could evolve into a more comprehensive tool that offers advanced functionalities to help organizations streamline not only leave management but also overall employee management and resource planning.

References

[1] Laravel, *Laravel Documentation*, Laravel, 2025. [Online]. Available: https://laravel.com/docs. [Accessed: Apr. 21, 2025].

[2] MySQL, *MySQL Reference Manual*, Oracle, 2025. [Online]. Available: https://dev.mysql.com/doc/. [Accessed: Apr. 22, 2025].

[3] W3Schools, *W3Schools Online Web Tutorials*, W3Schools, 2025. [Online]. Available: https://www.w3schools.com. [Accessed: Apr. 23, 2025].

[4] PHP Manual, *PHP: Hypertext Preprocessor Documentation*, The PHP Group, 2025. [Online]. Available: https://www.php.net/manual. [Accessed: Apr. 23, 2025].

[5] GitHub, *GitHub - Version Control and Collaboration*, GitHub, 2025. [Online]. Available: https://github.com. [Accessed: Apr. 25, 2025].

[6] MDN Web Docs, *Mozilla Developer Network Web Docs*, Mozilla, 2025. [Online]. Available: https://developer.mozilla.org/en-US/. [Accessed: Apr. 25, 2025].