**“LEAVE MANAGEMENT SYSTEM”**

A REPORT SUBMITTED IN PARTIAL FULFILLMENT FOR

AWARD OF DIPLOMA

IN

INFORMATION TECHNOLOGY & ENGINEERING



# SESSION (2023-24)

**SUBMITTED BY:**

## Abhay Panwar (Enrollment No: IT20210001/018)

## Sujal Kumawat (Enrollment No: IT2022L003/018)

## Jayant Jangid (Enrollment No: IT20210011/018)

## Dev Sisodiya (Enrollment No:IT)

**SUBMITTED TO:**

### MR. GAURAV PALIWAL MR. GAURAV PALIWAL

(H.O.D.) (PROJECT INCHARGE)

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VIDYA BHAWAN POLYTECHNIC COLLEGE, UDAIPUR (RAJ.)**

**Approved by A.I.C.T.E., New Delhi & Affiliated by Board of Technical Education**

**Rajasthan, Jodhpur**

**“Best Polytechnic” Awarded by NITTTR Govt. of India**

**Estd.1956**



## CERTIFICATE

This is to certify Report entitled “Leave Management System” has been undertaken and written under my supervision and it describes the original project work carried out by **Abhay Panwar** **Department of Information Technology Engineering, Vidya Bhawan Polytechnic College, Udaipur.**

### MR. GAURAV PALIWAL MR. GAURAV PALIWAL

(H.O.D.) (PROJECT INCHARGE)

**DR. ANIL MEHTA**

**(Principal)**

**DATE:**

**Approved by A.I.C.T.E., New Delhi & Affiliated by Board of Technical Education**

**Rajasthan, Jodhpur**

**“Best Polytechnic” Awarded by NITTTR Govt. of India**

**Estd.1956**



## CERTIFICATE

This is to certify Report entitled “Leave Management System” has been undertaken and written under my supervision and it describes the original project work carried out by **Sujal Kumawat** **Department of Information Technology Engineering, Vidya Bhawan Polytechnic College, Udaipur.**

### MR. GAURAV PALIWAL MR. GAURAV PALIWAL

(H.O.D.) (PROJECT INCHARGE)

**DR. ANIL MEHTA**

**(Principal)**

**DATE:**

**Approved by A.I.C.T.E., New Delhi & Affiliated by Board of Technical Education**

**Rajasthan, Jodhpur**

**“Best Polytechnic” Awarded by NITTTR Govt. of India**

**Estd.1956**



## CERTIFICATE

This is to certify Report entitled “Leave Management System” has been undertaken and written under my supervision and it describes the original project work carried out by **Jayant Jangid** **Department of Information Technology Engineering, Vidya Bhawan Polytechnic College, Udaipur.**

### MR. GAURAV PALIWAL MR. GAURAV PALIWAL

(H.O.D.) (PROJECT INCHARGE)

**DR. ANIL MEHTA**

**(Principal)**

**DATE:**

**Approved by A.I.C.T.E., New Delhi & Affiliated by Board of Technical Education**

**Rajasthan, Jodhpur**

**“Best Polytechnic” Awarded by NITTTR Govt. of India**

**Estd.1956**



## CERTIFICATE

This is to certify Report entitled “Leave Management System” has been undertaken and written under my supervision and it describes the original project work carried out by **Dev Sisodiya** **Department of Information Technology Engineering, Vidya Bhawan Polytechnic College, Udaipur.**

### MR. GAURAV PALIWAL MR. GAURAV PALIWAL

(H.O.D.) (PROJECT INCHARGE)

**DR. ANIL MEHTA**

**(Principal)**

**DATE:**

**ACKNOWLEDGEMENT**

The success and final outcome of this project require a lot of guidance and assistance from many people and We are extremely fortunate to have got this all along the completion of our project work. Whatever we have done is only due to such supervision and collaboration and we would never forget to thank them.

First of all, we own huge respect of all those people who helped us with their guidance and assistance, without which the Final Year Project would not be successful. We are grateful to our Principal Sir **Mr. Anil Mehta** for providing us a great opportunity, full support and cooperation.

We are thankful to **Mr. Gaurav Paliwal** (HOD of IT) V.B.P.C. College Udaipur (Raj.) for their guidance and constant supervision as well as for providing necessary information regarding to the project report.

We extend our heartfelt gratitude to the entire **Unicloudtech** team for providing us with the opportunity to work on this project. Their unwavering support and guidance were instrumental in our ability to complete the project on time. We deeply appreciate their assistance and dedication, especially given their busy schedules.

We are thankful to all those who helped us with their guidance and assistance in completing the project. We are very delighted to get this great opportunity to show our profound sense to all our esteemed teachers. We can’t express our thanks in words for our parents who are indirectly co-operated during project.

# ABSTRACT

Managing leave applications can be challenging, especially for educational institutions where timely approvals and accurate records are crucial. Our Leave Management System offers a comprehensive solution to streamline this process, making it efficient and transparent.

Our system leverages modern web technologies to ensure a smooth user experience. The frontend interface, developed with HTML, CSS, Bootstrap, jQuery, and AJAX, is responsive and interactive. The backend, powered by the Laravel framework, ensures secure data handling and seamless integration with MySQL, managed through XAMPP.

Designed for simplicity and efficiency, our system centralizes all leave-related information, automates the leave application and approval processes, and provides clear tracking and reporting features. This enhances administrative efficiency and promotes transparency in leave management.

The system organizes all user information, including roles, leave balances, and application histories, in one place. This helps administrators and faculty easily manage and monitor leave requests. Students can apply for leave through an intuitive form with real-time validation, while faculty and administrators can approve or reject requests promptly.

Notifications and alerts are integral to our system. Users receive email notifications for leave status updates, and alerts for pending approvals and low leave balances ensure that stakeholders are always informed.

Security and privacy are prioritized in our system design, with robust measures to protect sensitive data and ensure access is limited to authorized personnel.

In conclusion, our Leave Management System is a user-friendly solution that simplifies leave management for educational institutions. It saves time, reduces administrative burden, and improves transparency, making it a valuable tool for managing leave requests efficiently.

# CHAPTER 1

**INTRODUCTION OF PROJECT**

**1.1 INTRODUCTION**

The Leave Management System is a solution designed to simplify leave management processes in educational institutions. It automates tasks such as leave application and approval, reducing administrative burden and improving efficiency. Automating leave-related tasks saves time and reduces errors, while providing clear and detailed leave information to students and faculty.

Managing leave in a college setting involves several key processes aimed at ensuring timely and accurate tracking of leave applications. Here's a detailed breakdown in a simple and descriptive manner:

1. **Leave Policy and Structure**: The college establishes various leave policies based on factors such as academic programs, faculty roles, and administrative requirements. Each policy outlines the types of leaves available, eligibility criteria, and approval workflows. These structures are often defined for different categories of users such as students, faculty, and administrative staff.
2. **User Registration and Role Assignment**: When users (students, faculty, or staff) are registered in the system, their roles and leave entitlements are determined based on the established policies. The registration system captures this information, linking each user to the applicable leave rules for the academic year.
3. **Leave Application Submission**: Users submit leave applications through an intuitive form on the system's portal. The form captures detail such as leave type, duration, and reason. Real-time validation ensures that the application meets policy requirements before submission.
4. **Approval Workflow**: Once a leave application is submitted, it enters an approval workflow. Faculty or administrative staff review and approve or reject the application based on predefined criteria. The system facilitates multi-level approvals if required, ensuring that leave requests are processed promptly.
5. **Leave Balance Tracking**: The system tracks leave balances in real-time, monitoring the status of each user's leave entitlement. This includes accounting for leave taken, pending applications, and remaining balances. Notifications are sent to users for approvals, rejections, and low leave balances.
6. **Notification and Communication**: The system sends email notifications to users for leave status updates and important alerts. This ensures that all stakeholders are informed about leave approvals, rejections, and any actions required.
7. **Reporting and Analytics**: College administrators generate regular reports on leave applications, approvals, and balances. These reports provide valuable insights for planning, forecasting, and decision-making, ensuring efficient leave management.
8. **Compliance and Record-Keeping**: The college follows regulatory requirements related to leave management and record-keeping. Detailed records are maintained to document leave transactions, audit trails, and compliance with applicable laws and regulations.

Overall, effective leave management in a college setting requires a coordinated effort across various departments, utilizing technology, clear communication, and sound administrative practices to ensure a smooth and transparent process for students, faculty, and administrators alike.

**1.2 SIGNIFICANCE OF THE PROJECT**

The Leave Management System project holds significant importance for educational institutions by transforming the way leave requests and approvals are managed. Here's a simple yet descriptive breakdown of its significance:

1. **Efficiency Enhancement**:
   * **Streamlined Processes**: The project streamlines leave-related processes, minimizing manual effort and paperwork for administrators. With automated application submissions, approval workflows, and notifications, it saves time and resources while ensuring accuracy.
   * **Real-Time Tracking**: Users can submit, approve, and track leave requests in real-time, ensuring that leave management is both efficient and up-to-date.
2. **Transparency Improvement**:
   * **Clear Communication**: By providing clear and detailed leave information to students, faculty, and administrators, the system enhances transparency. Users can easily understand their leave balances, application statuses, and approval workflows, promoting trust and accountability.
   * **Automated Notifications**: Instant notifications keep users informed about the status of their leave applications, reducing the need for manual follow-ups and inquiries.
3. **Operational Accountability**:
   * **Accurate Record-Keeping**: The system maintains accurate records of leave applications, approvals, and balances, promoting accountability and informed decision-making. Administrators can access comprehensive reports for planning and auditing purposes.
   * **Consistent Policy Enforcement**: The system ensures that leave policies are applied consistently across the institution, reducing discrepancies and ensuring fairness.

In essence, the Leave Management System project represents a significant advancement for educational institutions, bringing efficiency, transparency, operational accountability, and convenience to leave management processes. It not only simplifies administrative tasks but also enhances the overall experience for students, faculty, and administrators.

**1.3 Objective of the project**

The objective of the Leave Management System project is to automate and streamline leave-related processes in educational institutions, aiming to enhance efficiency, transparency, and accuracy in leave management. By leveraging modern web technologies and software tools such as HTML, CSS, Bootstrap, jQuery, AJAX, Laravel, and MySQL, the project seeks to simplify leave application, approval workflows, balance tracking, and reporting tasks. The ultimate goal is to create a comprehensive system that benefits students, faculty, and administrators by providing a user-friendly interface, real-time updates, and robust operational oversight. The specific objectives are:

1. **Enhance Efficiency**:
   * **Automation**: Automate the leave application and approval processes to reduce manual effort and paperwork.
   * **Real-Time Processing**: Enable real-time submission, tracking, and management of leave requests to ensure up-to-date information.
2. **Improve Transparency**:
   * **Clear Communication**: Provide clear and detailed information on leave policies, balances, and application statuses to all users.
   * **Notifications**: Implement automated notifications to keep users informed about the status of their leave applications and any required actions.
3. **Ensure Accuracy**:
   * **Accurate Record-Keeping**: Maintain precise records of leave applications, approvals, and balances to ensure data integrity.
   * **Policy Consistency**: Ensure consistent application of leave policies across the institution to avoid discrepancies.
4. **Facilitate Robust Reporting**:
   * **Comprehensive Reports**: Generate detailed reports on leave applications, balances, and trends to aid in decision-making and planning.
   * **Operational Insights**: Provide insights into leave patterns and administrative efficiency for continuous improvement.
5. **Enhance User Experience**:
   * **User-Friendly Interface**: Develop an intuitive and easy-to-navigate interface that caters to the needs of students, faculty, and administrators.
   * **Accessibility**: Ensure the system is accessible from various devices and platforms to accommodate all users.

By achieving these objectives, the Leave Management System aims to significantly improve the overall leave management process, making it more efficient, transparent, and user-friendly, while also ensuring accurate and reliable data management.

**1.4 Functionality**

The Leave Management System is designed to automate and streamline the entire leave management process in educational institutions. Below is a detailed overview of the key functionalities of the system:

1. **User Registration and Authentication**:
   * **User Roles**: Supports multiple user roles including students, faculty, and administrators.
   * **Secure Login**: Provides secure login and authentication mechanisms for all users.
   * **Role-Based Access Control**: Ensures that users can access only the functionalities pertinent to their roles.
2. **Leave Application Process**:
   * **Leave Request Submission**: Users can submit leave applications through an intuitive form.
   * **Leave Types**: Supports various types of leave such as sick leave, personal leave, and academic leave.
   * **Real-Time Validation**: Uses jQuery and AJAX for real-time form validation to ensure completeness and accuracy before submission.
3. **Leave Approval Workflow**:
   * **Multi-Level Approval**: Configurable multi-level approval workflows to accommodate different institutional policies.
   * **Approval Interface**: Faculty and administrators have a dedicated interface to review, approve, or reject leave applications.
   * **Comments and Feedback**: Allows approvers to add comments or request additional information from the applicant.
4. **Leave Balance Management**:
   * **Automatic Balance Tracking**: Automatically updates leave balances based on approved leave requests.
   * **Leave Quota Management**: Manages different leave quotas and entitlements for various user roles.
   * **Balance Inquiry**: Users can view their current leave balances and history at any time.
5. **Notifications and Alerts**:
   * **Email Notifications**: Sends automated email notifications for leave submission, approval, rejection, and balance updates.
   * **Alerts**: Generates alerts for pending approvals, upcoming leave, and low leave balances.
6. **Reporting and Analytics**:
   * **Leave Reports**: Generates detailed reports on leave applications, approvals, and balances.
   * **Usage Statistics**: Provides analytics on leave patterns, helping administrators understand trends and make informed decisions.
   * **Customizable Reports**: Allows customization of reports based on date range, leave type, department, and other parameters.
7. **User Dashboard**:
   * **Personal Dashboard**: Provides users with a dashboard displaying their leave status, upcoming leave, and application history.
   * **Administrative Dashboard**: Gives administrators an overview of leave statistics, pending approvals, and system usage metrics.
8. **System Administration**:
   * **User Management**: Allows administrators to manage user accounts, roles, and permissions.
   * **Policy Configuration**: Enables configuration of leave policies, approval workflows, and notification settings.
   * **Data Backup and Recovery**: Provides tools for data backup and recovery to ensure data integrity and availability.
9. **Security and Compliance**:
   * **Data Protection**: Implements robust security measures to protect sensitive user data.
   * **Audit Trails**: Maintains detailed logs of all system activities for auditing and compliance purposes.
   * **Compliance**: Ensures compliance with institutional regulations and legal requirements regarding leave management.

By incorporating these functionalities, the Leave Management System aims to provide a comprehensive solution that enhances the efficiency, transparency, and accuracy of leave management processes in educational institutions.

**1.5 Features**

The Leave Management System is designed to offer a comprehensive suite of features that cater to the needs of students, faculty, and administrators in educational institutions. These features aim to streamline the leave management process, enhance user experience, and ensure efficient administration. Below is an overview of the key features:

1. **User-Friendly Interface**:
   * **Intuitive Design**: A clean and easy-to-navigate interface developed using Bootstrap, ensuring a seamless user experience across all devices.
   * **Responsive Layout**: Optimized for both desktop and mobile devices, allowing users to access the system anytime, anywhere.
2. **Role-Based Access Control**:
   * **Multiple User Roles**: Supports roles such as students, faculty, and administrators, each with specific permissions and access levels.
   * **Secure Authentication**: Robust login mechanisms ensuring that only authorized users can access the system.
3. **Leave Application and Approval**:
   * **Simple Leave Application Form**: A straightforward form for submitting leave requests with fields for leave type, dates, and reasons.
   * **Multi-Level Approval Workflow**: Configurable workflows that support single or multiple levels of approval, tailored to the institution’s policies.
   * **Approval Notifications**: Automated notifications to approvers and applicants, informing them of submission, approval, or rejection status.
4. **Leave Balance Management**:
   * **Automatic Updates**: Real-time tracking and automatic updating of leave balances upon approval or rejection of requests.
   * **Leave Quotas**: Customizable leave quotas for different user roles, ensuring accurate entitlement tracking.
   * **Balance Inquiry**: Users can easily check their leave balances and leave history through their dashboard.
5. **Real-Time Notifications and Alerts**:
   * **Email Notifications**: Sends emails to users for important events such as leave submission, approval, and reminders for upcoming leave.
   * **System Alerts**: Dashboard alerts for pending approvals, low leave balances, and other critical notifications.
6. **Comprehensive Reporting and Analytics**:
   * **Detailed Reports**: Generates reports on leave applications, approvals, balances, and trends.
   * **Customizable Filters**: Allows customization of report parameters such as date ranges, leave types, departments, and user roles.
   * **Usage Analytics**: Provides insights into leave usage patterns, helping administrators in decision-making and policy adjustments.
7. **Administrative Tools**:
   * **User Management**: Administrators can add, edit, and deactivate user accounts, assign roles, and manage permissions.
   * **Policy Configuration**: Tools to configure and update leave policies, approval workflows, and notification settings.
   * **Data Backup and Recovery**: Features for regular data backups and recovery options to ensure data security and integrity.
8. **Security and Compliance**:
   * **Data Encryption**: Ensures that all sensitive data is encrypted to protect against unauthorized access.
   * **Audit Logs**: Maintains detailed logs of user activities and system changes for auditing and compliance purposes.
   * **Regulatory Compliance**: Adheres to institutional regulations and legal requirements for leave management and data protection.
9. **Integration Capabilities**:
   * **Calendar Integration**: Syncs approved leaves with institutional calendars to avoid scheduling conflicts.
   * **Third-Party Integrations**: Potential to integrate with other systems such as HR, payroll, and academic scheduling software.
10. **Support and Maintenance**:
    * **User Support**: Provides helpdesk support for users to resolve any issues or questions.
    * **Regular Updates**: Periodic updates to the system to incorporate new features, improve performance, and enhance security.

By incorporating these features, the Leave Management System ensures an efficient, transparent, and user-friendly approach to managing leave requests and approvals, significantly benefiting educational institutions.

# CHAPTER 2

**Tools & Technologies**

In the development of the Leave Management System, various tools and technologies were employed to create a robust and efficient solution. This chapter provides an in-depth analysis of the key technologies utilized, including HTML, CSS, JavaScript, Bootstrap, jQuery, AJAX, Laravel, and MySQL, highlighting their features, advantages, and disadvantages.

**2.1 HTML (HyperText Markup Language)**

HTML is the foundational markup language used to create web pages. It provides the structure for web content, defining elements such as headings, paragraphs, links, images, and other multimedia. HTML serves as the skeleton of web applications, allowing developers to embed multimedia elements and connect web pages through hyperlinks.

**Features:**

* Standard markup language for creating web pages.
* Provides the structure for web content.
* Supports multimedia elements such as images, videos, and audio.

**Advantages:**

* Easy to learn and use.
* Widely supported by all web browsers.
* Allows for the integration of various web technologies.

**Disadvantages:**

* Limited in terms of dynamic functionality.
* Requires additional languages (CSS, JavaScript) for styling and interactivity.

**2.2 CSS (Cascading Style Sheets)**

CSS is used to control the presentation and layout of web pages. It allows developers to separate content (HTML) from design, making it easier to maintain and update the look and feel of a website. CSS also supports responsive design techniques, ensuring web pages are accessible and visually appealing on various devices.

**Features:**

* Style sheet language used for describing the presentation of a document written in HTML.
* Enables the separation of content from design.
* Supports responsive design techniques.

**Advantages:**

* Enhances the look and feel of web pages.
* Promotes consistency across different pages.
* Facilitates responsive design for various devices.

**Disadvantages:**

* Can become complex to manage with larger projects.
* Browser compatibility issues can arise.

**2.3 JavaScript**

JavaScript is a high-level programming language that enables dynamic behavior on web pages. It supports various programming paradigms, including event-driven, functional, and imperative programming. JavaScript is essential for creating interactive web applications, allowing for real-time updates, form validation, and other client-side functionalities.

**Features:**

* High-level, interpreted programming language.
* Enables dynamic behavior on web pages.
* Supports event-driven, functional, and imperative programming styles.

**Advantages:**

* Enhances user experience through interactive elements.
* Runs on the client-side, reducing server load.
* Widely supported and has a vast ecosystem of libraries and frameworks.

**Disadvantages:**

* Can lead to security vulnerabilities if not properly managed.
* Debugging can be difficult in large codebases.

**2.4 Bootstrap**

Bootstrap is a front-end framework that helps in developing responsive and mobile-first web pages. It includes a set of pre-designed components and utilities, making it easier and faster to create consistent and visually appealing designs. Bootstrap's grid system facilitates layout management, ensuring responsive behavior across different devices.

**Features:**

* Front-end framework for developing responsive and mobile-first web pages.
* Includes pre-designed components and utilities.
* Utilizes a grid system for layout management.

**Advantages:**

* Speeds up development with ready-to-use components.
* Ensures consistent design and responsive behavior.
* Extensive documentation and community support.

**Disadvantages:**

* May result in websites looking similar if not customized.
* Can include unnecessary code, increasing page load times.

**2.5 jQuery**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation. It provides an easy-to-use API for AJAX and event handling, enhancing the development process by reducing the amount of code needed for common tasks and ensuring cross-browser compatibility.

**Features:**

* JavaScript library designed to simplify HTML DOM tree traversal and manipulation.
* Provides an easy-to-use API for AJAX and event handling.
* Cross-browser compatibility.

**Advantages:**

* Simplifies complex JavaScript tasks.
* Reduces the amount of code needed for common tasks.
* Enhances cross-browser compatibility.

**Disadvantages:**

* Can be slower than vanilla JavaScript for certain tasks.
* Adds an extra dependency to the project.

**2.6 AJAX (Asynchronous JavaScript and XML)**

AJAX is a technique for creating fast and dynamic web pages. It allows web pages to be updated asynchronously by exchanging small amounts of data with the server in the background. This enables partial page updates without a full reload, improving user experience and application performance.

**Features:**

* Technique for creating fast and dynamic web pages.
* Allows web pages to be updated asynchronously by exchanging small amounts of data with the server in the background.
* Supports partial page updates without a full reload.

**Advantages:**

* Improves user experience by making web applications more responsive.
* Reduces server load and bandwidth usage.
* Enhances the performance of web applications.

**Disadvantages:**

* Can complicate the debugging process.
* Requires careful management of state and data consistency.

**2.7 Laravel**

Laravel is a PHP framework for web application development that follows the MVC (Model-View-Controller) architectural pattern. It includes features like routing, authentication, and session management, and offers a built-in ORM (Object-Relational Mapping) called Eloquent for database operations. Laravel simplifies common web development tasks and promotes clean, maintainable code.

**Features:**

* PHP framework for web application development.
* Follows the MVC (Model-View-Controller) architectural pattern.
* Includes features like routing, authentication, and session management.
* Built-in ORM (Object-Relational Mapping) called Eloquent for database operations.
* Supports Blade templating engine for dynamic views.
* Integrated support for task scheduling and queuing.

**Advantages:**

* Simplifies common web development tasks.
* Promotes clean and maintainable code.
* Extensive ecosystem and community support.
* Provides robust security features like CSRF protection and SQL injection prevention.
* Easily integrates with third-party packages and services.
* Built-in testing tools facilitate the development of test-driven applications.

**Disadvantages:**

* Can have a steep learning curve for beginners.
* May introduce overhead in small projects.
* Requires familiarity with PHP and Composer (PHP package manager).

**2.8 MySQL**

MySQL is a relational database management system (RDBMS) that supports SQL (Structured Query Language) for database interactions. It offers features like indexing, transactions, and concurrency control, providing robust security features and access controls. MySQL is widely used in web applications for reliable and scalable data management.

**Features:**

* Relational database management system (RDBMS).
* Supports SQL (Structured Query Language) for database interactions.
* Offers features like indexing, transactions, and concurrency control.
* Provides robust security features and access controls.
* Supports replication and clustering for high availability and scalability.
* Compatible with various storage engines (e.g., InnoDB, MyISAM).

**Advantages:**

* Reliable and widely used in web applications.
* High performance and scalability.
* Extensive documentation and community support.
* Open-source with regular updates and improvements.
* Easy integration with various programming languages and frameworks.
* Supports complex queries and data analysis.

**Disadvantages:**

* Can be complex to set up and manage.
* May require optimization for handling large datasets.
* Certain advanced features may necessitate deeper database knowledge.

**2.9 Additional Technologies**

**2.9.1 RESTful APIs**

RESTful APIs are an architectural style for designing networked applications. They utilize standard HTTP methods (GET, POST, PUT, DELETE) and focus on stateless client-server communication. RESTful APIs support data exchange in formats like JSON and XML, enabling interoperability between different systems and platforms.

**Features:**

* Architectural style for designing networked applications.
* Utilizes standard HTTP methods (GET, POST, PUT, DELETE).
* Focuses on stateless client-server communication.
* Supports data exchange in formats like JSON and XML.

**Advantages:**

* Promotes scalability and modularity in application design.
* Enables interoperability between different systems and platforms.
* Easy to understand and implement using standard HTTP protocols.
* Wide support across different languages and frameworks.

**Disadvantages:**

* Requires proper versioning and documentation.
* Stateless nature may lead to increased overhead for maintaining context.
* Security concerns need to be addressed, such as authentication and data encryption.

**2.9.2 Version Control Systems (Git)**

Git is a distributed version control system for tracking changes in source code. It supports branching, merging, and collaborative development, providing a history of changes and enabling developers to revert to previous states. Git is essential for managing code in team environments and maintaining code integrity.

**Features:**

* Distributed version control system for tracking changes in source code.
* Supports branching, merging, and collaborative development.
* Provides a history of changes and enables revert to previous states.

**Advantages:**

* Facilitates collaboration among multiple developers.
* Ensures code integrity and history tracking.
* Supports various workflows and branching strategies.
* Widely adopted with extensive documentation and community support.

**Disadvantages:**

* Can be complex to manage with larger teams and repositories.
* Requires learning curve for beginners to understand branching and merging concepts.

**Summary**

The Leave Management System leverages a combination of HTML, CSS, JavaScript, Bootstrap, jQuery, AJAX, Laravel, and MySQL to create a robust, efficient, and user-friendly application. Each technology brings unique strengths to the project, contributing to a comprehensive solution that meets the needs of educational institutions. While there are some disadvantages to consider, the advantages provided by these tools far outweigh the challenges, ensuring the successful implementation of the system. Additionally, incorporating RESTful APIs and version control with Git further enhances the system’s scalability, maintainability, and collaborative potential.

# CHAPTER 1

**Design & Architecture**

**3.1 ER Diagram**

The Entity-Relationship (E-R) diagram for the Leave Management System provides a visual representation of the various entities (such as Admins, Departments, Employees, etc.) and their relationships within the system. This diagram serves as a blueprint for understanding the structure and interactions of the different components of the system.

Each entity in the diagram represents a distinct concept or object within the system, such as users, organizational units, or leave-related entities. The attributes associated with each entity describe the properties or characteristics of that entity.

The relationships between entities define how they are connected or associated with each other. These relationships help to illustrate how data flows and interacts within the system. For example, the "Belongs To" relationship between Employees and Departments indicates that each employee is associated with a specific department.

**Entity-Relationship (E-R) Diagram:**

**Entities:**

1. **Admins:**
   * Attributes: id (Primary Key), emp\_id, first\_name, last\_name, dpt\_id, dst\_id, dob, image, email, mobile\_no, password, created\_at, updated\_at, delete3.
   * Description: Administrators responsible for managing the leave management system. Each admin has a unique identifier (id) and employee identifier (emp\_id). Admins are associated with specific departments (dpt\_id) and designations (dst\_id). Personal information, including name, date of birth (dob), contact details, and login credentials, is stored. Timestamps are recorded for record creation and updates.
2. **Departments:**
   * Attributes: id (Primary Key), dpt\_name, status, created\_at, updated\_at.
   * Description: Represents organizational departments. Each department has a unique identifier (id) and a name (dpt\_name). Status indicates whether the department is active or inactive. Timestamps track department creation and updates.
3. **Designations:**
   * Attributes: id (Primary Key), dst\_name, status, created\_at, updated\_at.
   * Description: Different job titles or positions within the organization. Each designation has a unique identifier (id) and a name (dst\_name). Status indicates whether the designation is active or inactive. Timestamps track designation creation and updates.
4. **Employees:**
   * Attributes: id (Primary Key), first\_name, last\_name, dpt\_id, dst\_id, dob, mobile\_no, image, gender, addresses, email, password, status, leave\_taken, created\_at, updated\_at, delete1.
   * Description: Represents employees in various departments. Each employee has a unique identifier (id) and personal details such as name, date of birth (dob), contact information, and login credentials. Employees are associated with specific departments (dpt\_id) and designations (dst\_id). Attributes include employment status, total leave taken, and timestamps for record creation and updates.
5. **Leaves:**
   * Attributes: id (Primary Key), emp\_id (Foreign Key), lt\_id, start\_date, end\_date, how\_long, reason, location, approval\_pcp, approval\_hod, approval\_vc, final\_approval, comment, status, created\_at, updated\_at, delete1.
   * Description: Represents leave requests made by employees. Each leave request has a unique identifier (id) and is associated with the requesting employee (emp\_id) and leave type (lt\_id). Attributes include start date, end date, duration, reason, location, approval status by different authorities, comments, and timestamps for record creation and updates.
6. **LeaveTypes:**
   * Attributes: id (Primary Key), lt\_name, total\_leave\_year, status, created\_at, updated\_at.
   * Description: Different types of leave available to employees. Each leave type has a unique identifier (id) and a name (lt\_name). Total leave allocated per year, status, and timestamps for record creation and updates are tracked.
7. **Notifications:**
   * Attributes: id (Primary Key), name, path, status, created\_at, updated\_at.
   * Description: Notifications sent to employees or administrators. Each notification has a unique identifier (id), name, path (URL), and status (read or unread). Timestamps track notification creation and updates.
8. **Staff:**
   * Attributes: id (Primary Key), emp\_id, first\_name, last\_name, dpt\_id, dst\_id, dob, mobile\_no, image, gender, duty, status, created\_at, updated\_at, delete2.
   * Description: Represents staff members associated with different departments. Each staff member has a unique identifier (id) and personal details similar to employees. They are associated with specific departments (dpt\_id) and designations (dst\_id). Attributes include employment status, duty, and timestamps for record creation and updates.

**Relationships:**

1. **Manages:**
   * Type: Many-to-One
   * Description: Each admin manages one department, but a department can be managed by multiple admins.
2. **Belongs To:**
   * Type: Many-to-One
   * Description: Each employee belongs to one department, but a department can have multiple employees.
3. **Has Designation:**
   * Type: Many-to-One
   * Description: Each employee has one designation, but a designation can be associated with multiple employees.
4. **Submits:**
   * Type: Many-to-One
   * Description: Each leave request is submitted by one employee, but an employee can submit multiple leave requests.
5. **Uses:**
   * Type: Many-to-One
   * Description: Each employee uses one leave type, but a leave type can be used by multiple employees.
6. **Receives:**
   * Type: One-to-Many
   * Description: Each employee receives multiple notifications, but each notification is received by one employee.
7. **Part of:**
   * Type: Many-to-One
   * Description: Each staff member is part of one department.
8. **Approves:**
   * Type: One-to-Many
   * Description: Each leave request can be approved by multiple authorities, but each approval is specific to one leave request.

This refined diagram and description should provide a comprehensive understanding of the Leave Management System's structure and relationships.

**3.2 DATA FLOW DIAGRAM (DFD): -**

**3.2.1 Leave Management System:**

1. **Admin:**
   * Tasks: Manage departments, designations, and leave types, review leave applications.
   * Interactions: Set up system parameters, approve/reject leaves.
2. **Employee:**
   * Tasks: Apply for leave, view leave status.
   * Interactions: Submit leave applications, receive notifications.
3. **Leave Management System:**
   * Main processes: Leave application, approval workflow, notification management.

**Level 1 DFD:**

**Processes:**

1. **Manage Departments and Designations:**
   * Description: Admin adds, updates, or deletes department and designation information.
   * Data Stores: Department, Designation.
2. **Manage Leave Types:**
   * Description: Admin sets up different types of leave.
   * Data Stores: LeaveType.
3. **Apply for Leave:**
   * Description: Employee applies for leave.
   * Data Stores: Leave.
4. **Approve/Reject Leave:**
   * Description: Admin reviews leave applications and approves or rejects them.
   * Data Stores: Leave.
5. **Send Notifications:**
   * Description: System sends notifications to employees about leave status.
   * Data Stores: Notification.

**Data Flow:**

**Entities:**

1. **Admin:**
   * Inputs: Department details, Designation details, Leave Type details, Leave application approval/rejection.
   * Outputs: Notifications.
2. **Employee:**
   * Inputs: Leave application.
   * Outputs: Leave status, Notifications.

**Data Stores:**

1. **Department:**
   * Stores department information.
2. **Designation:**
   * Stores designation information.
3. **LeaveType:**
   * Stores leave type information.
4. **Leave:**
   * Stores leave applications.
5. **Notification:**
   * Stores notifications sent to employees.

**Detailed Processes:**

**Process 1: Manage Departments and Designations**

* **Inputs:**
  + Admin provides department and designation details.
* **Outputs:**
  + Updated Department and Designation data stores.

**Process 2: Manage Leave Types**

* **Inputs:**
  + Admin provides leave type details.
* **Outputs:**
  + Updated LeaveType data store.

**Process 3: Apply for Leave**

* **Inputs:**
  + Employee submits leave application.
* **Outputs:**
  + New entry in Leave data store.

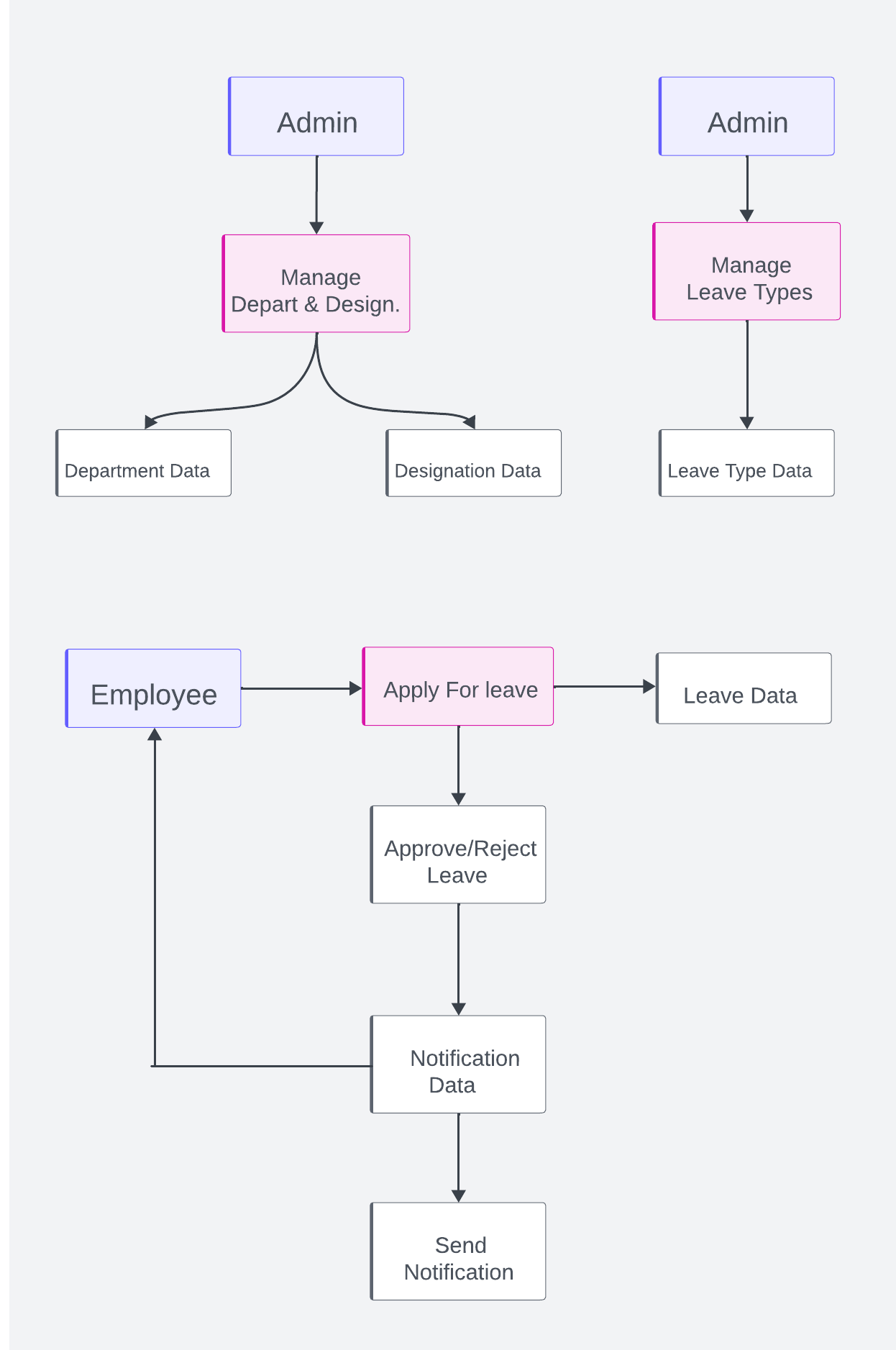
**Process 4: Approve/Reject Leave**

* **Inputs:**
  + Admin reviews leave applications.
* **Outputs:**
  + Updated Leave data store with approval status.

**Process 5: Send Notifications**

* **Inputs:**
  + Leave application status changes.
* **Outputs:**
  + New entry in Notification data store, notification sent to Employee.

**3.2.2 DFD Diagram**

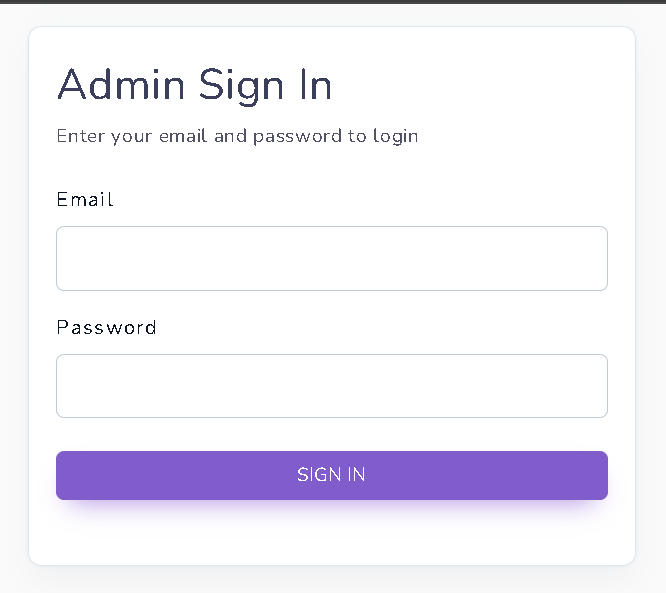


This DFD illustrates the flow of data within the Leave Management System, detailing how information is processed, stored, and communicated between admins and employees.

**CHAPTER 4:**

**SNAPSHOTS AND DESCRIPTION**

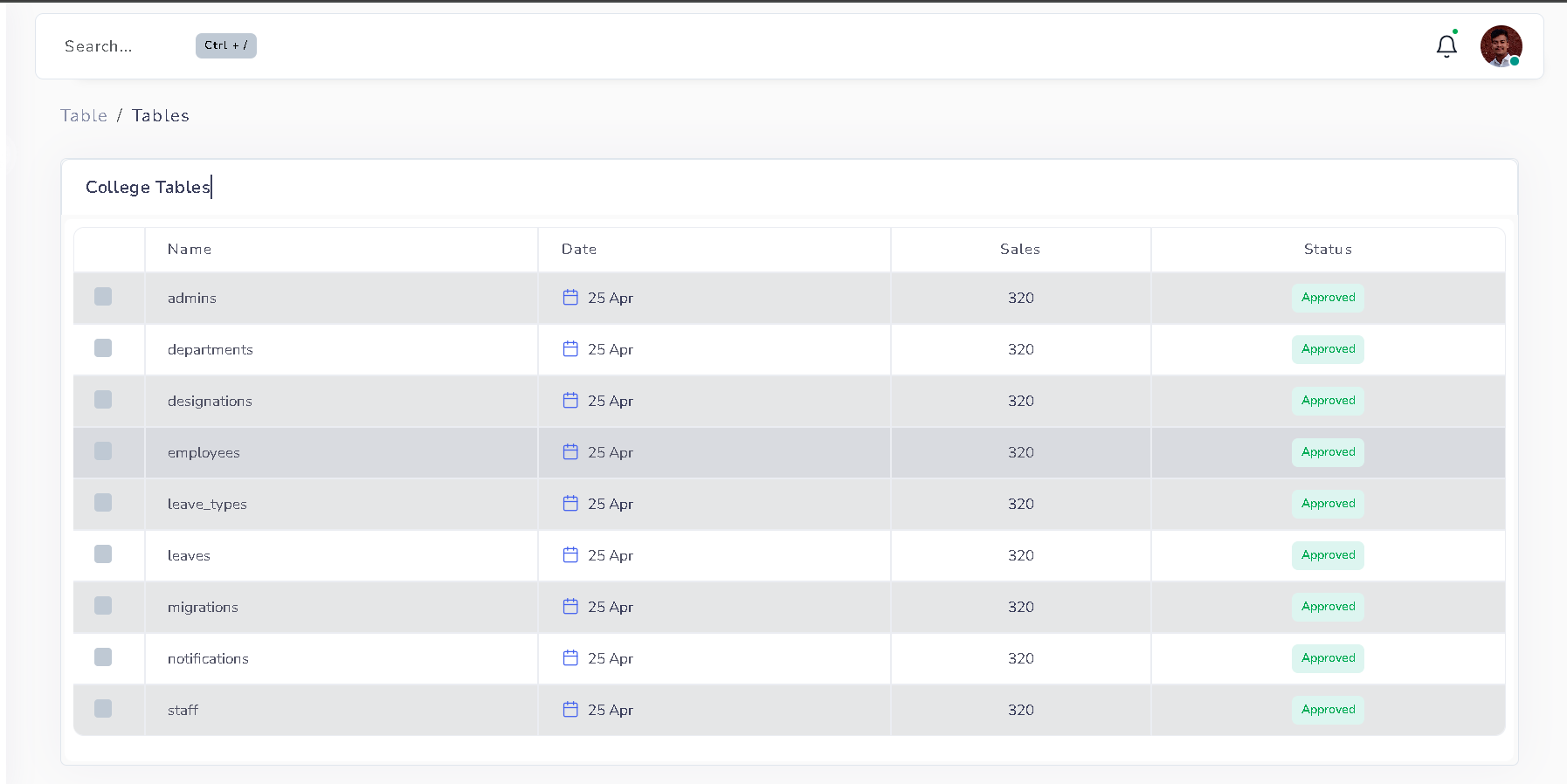
**4.1 Admin login Page**



**Description**

* **Form Action**: Submits to the admin\_login route for processing.
* **Security**: Includes a CSRF token for protection against CSRF attacks.
* **Email Input**: An input field for the admin's email address, with email validation.
* **Password Input**: An input field for the admin's password (should be type="password" for security).
* **Form Labels and Styling**: Uses Bootstrap for styling and layout, ensuring a user-friendly interface.
* **Submit Button**: A full-width button styled with Bootstrap, prompting the user to sign in.

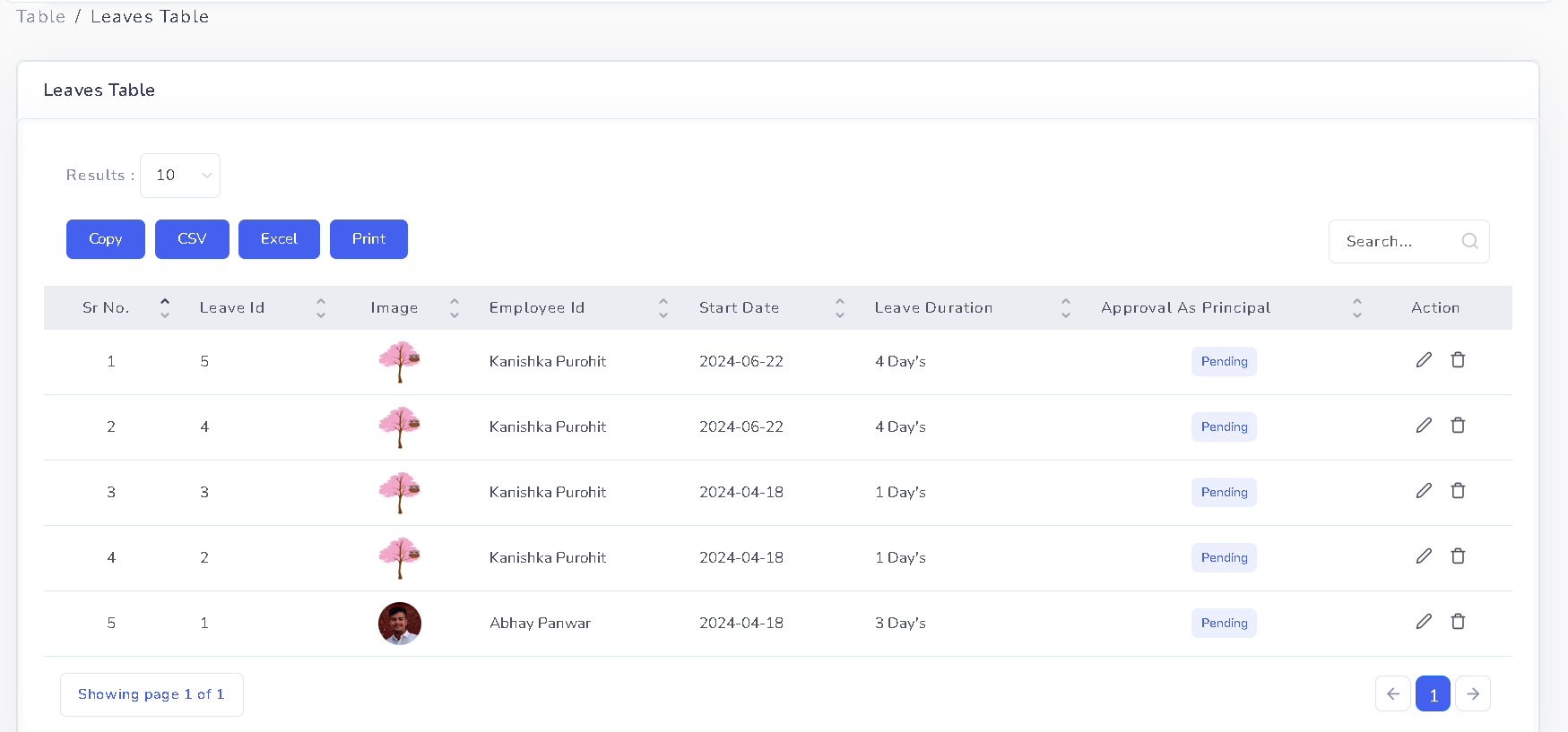
**4.2 Database Tables Page**



The provided code is a Laravel Blade template for rendering a dynamic, styled table within the Leave Management System. Here's a description of its functionality in one-line points:

* **Table Styling**: Uses Bootstrap classes for a hoverable, striped, and bordered table design.
* **Table Headers**: Defines column headers for checkbox, name, date, sales, and status.
* **Checkbox Column**: Includes a checkbox in each row for selection, initially commented out in the header.
* **Name Column**: Displays the name of each table with a clickable link to detailed data.
* **Date Column**: Uses an SVG calendar icon and a static date ("25 Apr") for visual representation.
* **Sales Column**: Centers and displays a static sales value (320) for each row.
* **Status Column**: Centers and displays the status as a badge labeled "Approved".
* **Dynamic Data**: Iterates over $employee collection to populate rows with data dynamically.
* **Blade Directives**: Utilizes Blade syntax (@foreach, {{ }}) for data binding and control structures.

**4.3 Leave Tables Page**

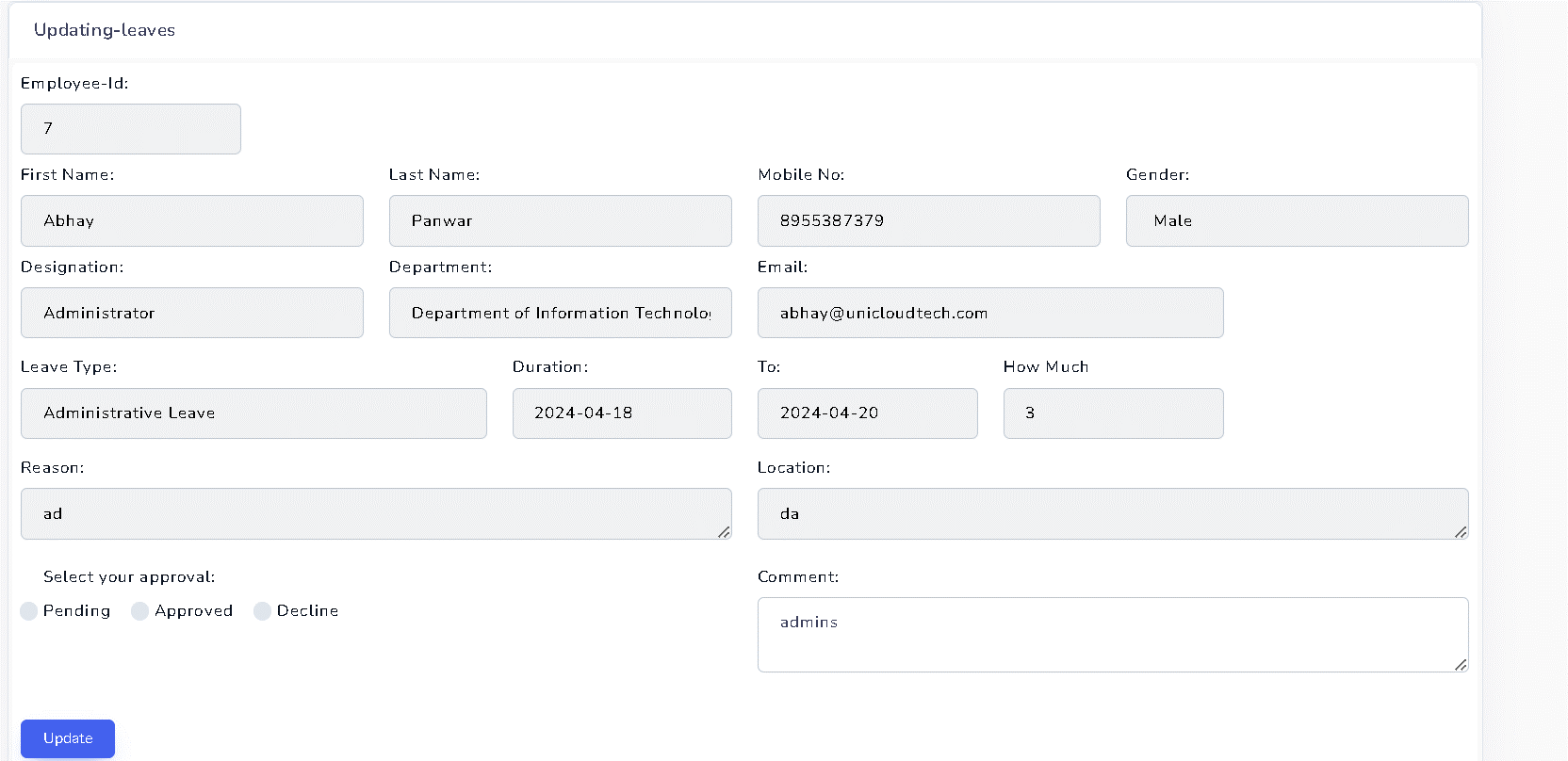


The provided code is a Laravel Blade template for rendering a detailed, interactive table to manage leave requests within the Leave Management System. Below is a description of its functionality in one-line points.

* **Container**: div with id="printarea" for defining the printable area, styled with margins and padding.
* **Table Initialization**: table element with id="html5-extension" for initializing DataTable functionalities.
* **Table Headers**: Defines columns for serial number, leave ID, image, employee ID, start date, leave duration, approval status, and action.
* **Dynamic Data Rows**: Uses @foreach ($leaves as $row) to iterate through each leave request and populate the table.
* **Checkbox Column**: First column with serial number and checkboxes for selection.
* **Leave Information**: Columns for leave ID, employee image, employee ID, start date, and leave duration.
* **Conditional Approval Columns**: Displays different approval statuses based on user roles (Principal, HOD, Vice President) using Session checks.
* **Approval Status**: Conditional badges showing approval status (Pending, Approved, SPL.Approved, Decline).
* **Action Column**: Provides action buttons (Edit, Delete) based on approval status and user roles, using icons for visual cues.
* **Bootstrap and Feather Icons**: Uses Bootstrap for styling and Feather icons for action buttons and visual elements.

This table setup enables efficient management of leave requests by displaying comprehensive leave information, approval statuses, and providing quick access to actions like edit and delete based on user roles.

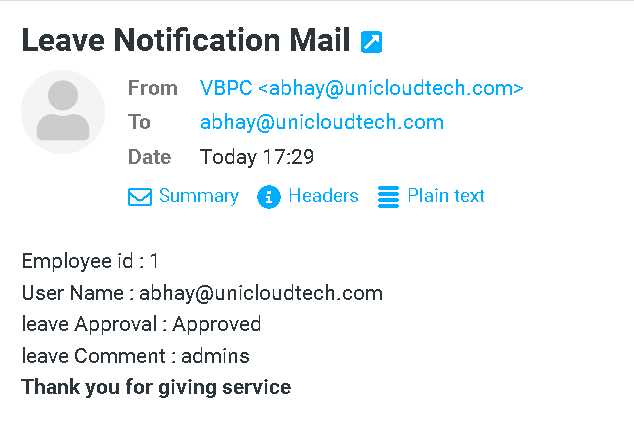
**4.4 Leave Approval Page**



The provided code is a Laravel Blade template for rendering a form that allows administrators to view and update leave request details within the Leave Management System. Below is a description of its functionality in one-line points.

* **Form Action**: Submits to the route updating-leaves-form-update with the leave ID as a parameter.
* **CSRF Protection**: Includes @csrf directive to protect against CSRF attacks.
* **Hidden Input**: Contains a hidden input field for the leave ID.
* **Employee Information**: Displays read-only fields for employee ID, first name, last name, mobile number, gender, designation, department, and email.
* **Leave Details**: Shows read-only fields for leave type, start date, end date, duration, reason, and location.
* **Approval Status**: Radio buttons to select approval status (Pending, Approved, Decline) based on session data.
* **Comment Field**: Textarea for adding or editing comments related to the leave request.
* **Submit Button**: Includes a submit button to update the leave request details.

**4.4 After Leave Approval Email has been Sent**



The screenshot shows an email notification sent after a leave approval within the Leave Management System. Below is a description of its content and functionality in one-line points.

* **Employee ID**: Displays the employee ID as "1".
* **User Name**: Shows the username as "abhay@unicloudtech.com".
* **Leave Approval Status**: Indicates the leave status as "Approved".
* **Leave Comment**: Includes a comment, "admins".
* **Closing Message**: Thanks the recipient with "Thank you for giving service".

This email efficiently communicates the leave approval status and related details to the employee, ensuring they are promptly informed.

**4.5 Process After Leave Approval or Decline**

After a leave request is either approved or declined, the Leave Management System ensures that the employee is promptly informed through multiple channels.

* **Automated Email Notification**:
  + **Recipient**: Sent to the employee's registered email address.
  + **Content**: Includes employee ID, username, approval status (approved or declined), approver comments, and a closing message.
  + **Purpose**: Provides detailed information about the leave request decision.
* **Automated SMS Notification**:
  + **Recipient**: Sent to the employee's registered mobile number.
  + **Content**: Concise update on the leave request status (approved or declined).
  + **Purpose**: Ensures quick and effective communication of the leave decision.
* **Dual-Channel Notification**:
  + **Email**: Delivers comprehensive details and formal communication.
  + **SMS**: Offers rapid and direct notification for immediate awareness.
* **Outcome**:
  + **Employee Awareness**: Employees are promptly informed about the status of their leave requests.
  + **Enhanced Communication**: Ensures that employees receive timely updates through both email and SMS.