Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Project Kit

Title of the Project

Leave Management System

Abstract of the Project

A Leave Management System is a software solution designed to streamline the process of managing employee leave requests and approvals. It automates tracking leave balances, requests and approvals, ensuring transparency and consistency. The system minimizes manual errors and improves efficiency. It helps HR teams and managers maintain compliance with policies while enhancing employee satisfaction through a seamless leave management experience.

Generic keyword:

Leave Management, Attendance Tracking, Employee Leave Tracking, Time-off Management.

Specific Technology keywords:

HTML, CSS, PHP, Laravel Framework, Web-Based Application, Database Management.

Functional Components of the Project:

 User Authentication and Authorization
- Login, registration, and logout functionality.
- Role-based access control (Admin, HR, Manager, Employee).

 Employee Management
- Add, update, and view employee details.
- Assign employees to departments and roles.

 Leave Type Management
- Define leave types (e.g., Sick Leave, Vacation, Casual Leave).
- Configure leave policies and limits for each type.

 Leave Application and Approval
- Employees can apply for leave with necessary details.
- Managers or HR can approve, reject, or request modifications.
- Track the status of leave requests (Pending, Approved, Rejected).

 Leave Balance and Allocation
- Display leave balances to employees.
- Allocate leave days annually or periodically based on policies.
- Deduct leave days upon approval of requests.

 Notifications and Alerts
- Send notifications to employees and managers for leave updates.
- Alerts for pending leave requests or approvals.

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Functionality:

Users of the system:

Employees:

Role: User Registration and Profile Management, Leave Application, Leave Balance Monitoring, Request Tracking, Notifications and Alerts.

Administrators:

- Role: Policy Definition and Management, User Role and Access Control, Employee and Department Management, Leave Type and Allocation Management, Approval Workflow Configuration, Monitoring and Reporting, System Administration and Maintenance.

The core functionality of the Leave Management System includes:

 User Authentication and Role Management: Secure login and role-based access control.
 Leave Application and Tracking: Employees submit and track leave requests.
 Leave Approval Workflow: Managers and HR approve or reject requests.
 Leave Balance Management: Display and update real-time leave balances.
 Leave Type and Policy Management: Define leave types and policies.
 Notifications and Alerts: Automated alerts for leave status and approvals.
 Reporting and Analytics: Generate leave reports and attendance trends.

Steps to start-off the project:

The following steps will be helpful to start off the project –

1. Requirement Gathering**:** Collect requirements from stakeholders.
2. Feasibility Analysis**:** Assess technical and operational feasibility.
3. Project Planning**:** Define scope, timeline, and budget.
4. Design Architecture**:** Develop HLD and DLD for the system.
5. Technology Selection**:** Choose suitable tools and platforms.
6. Database Design**:** Create schema and relationships.
7. Development**:** Build core modules as per design.
8. Testing**:** Perform unit, integration, and system testing.
9. Deployment**:** Deploy the system on servers or cloud.
10. Training**:** Train users and document processes.
11. Go-Live**:** Launch the system and monitor performance.
12. Maintenance**:** Provide support and regular updates.

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Hardware Requirements:

Category	Component	Specification
Server	Processor	Quad-core or higher (e.g., Intel Xeon, AMD Ryzen)
	RAM	8 GB or higher (16 GB for large setups)
	Storage	500 GB SSD or more for database and logs
	Network	High-speed internet (1 Gbps recommended)
	Operating System	Windows Server, Linux, or cloud platforms (AWS, Azure)
Client Devices	Processor	Dual-core or higher (e.g., Intel i3 or above)
	RAM	4 GB or more
	Storage	10 GB free space
	Operating System	Windows, macOS, Linux, Android, or iOS
	Browser	Chrome, Firefox, Safari, or Edge (latest versions)
Backup & Recovery	Backup Server / Device	External device or cloud-based storage
	RAID Storage Configuration	To prevent data loss
	Uninterruptible Power Supply (UPS)	For handling power outages
Network	Router and Firewall	For secure network access
	VPN Access	For remote employees or administrators

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Software Requirements:

Category	Component	Specification
Server Software	Operating System	Windows Server
	Web Server	Apache, Nginx, or IIS,
	Database	MySQL
	Programming Language	HTML, CSS, PHP,Laravel
	Cloud Platform	AWS,Google Cloud
Client Software	Web Browser	Chrome, Firefox, Safari, or Edge
	Operating System	Windows
Development Tools	IDE / Code Editor	Visual Studio, Eclipse, PyCharm, or VS Code
	Version Control System	Git, GitHub, or GitLab

Manpower requirements:

2 to 3 students can complete this in 4–6 months if they work fulltime on it.

Milestones and Timelines

No.	Milestone name	Milestone Description	Time-line	Remarks
1.	Requirement Gathering	Collect and document all project requirements.		Engage with all stakeholders.
2.	Feasibility Analysis	Assess technical and financial feasibility of the project.		Evaluate risks and alternatives.
3.	Project Planning	Define project scope, timeline, and resource allocation.		Create detailed project plan.
4.	Design Phase	Develop high-level and detailed design documents.		Include user interface and architecture design.

5.	Technology Selection	Choose technology stack and tools for development.		Finalize hardware and software requirements.
6.	Database Design	Create database schema and relationships.		Ensure normalization and data integrity.
7.	Development Phase	Build core modules and functionalities.		Follow Agile methodology for iterative development.
8.	Testing Phase	Conduct unit, integration, and system testing.		Ensure all functionalities work as intended.
9.	Deployment	Deploy the system on the production environment.		Prepare for Go-Live.
10.	Training	Train end-users and administrators on system usage.		Prepare training materials.
11.	Go-Live	Official launch of the Leave Management System.		Monitor system performance.
12.	Post-Implementation Review	Evaluate project success and gather feedback.		Identify areas for improvement.

Guidelines and Reference:

Object Oriented Modelling and Design with UML- Michael Blaha, Jams Rumbaugh.

Software Engineering, Seventh edition Ian Sommerville.

Spring Boot Reference Document docs.spring.io.

Java - www.sun.com

Wikipedia - www.wikipedia.com

Database Management Systems - Navathe.

Complete Reference - J2EE - Keogh.