ABSTRACT (EMS)

The Employee management system (EMS) is a comprehensive software solution designed to streamline and automate human resource management processes. The EMS is a platform for managing employee data, tracking attendance, monitoring performance and administering benefits with a user-friendly interface and robust security.

Existing system and solutions

- Workday: A cloud-based HR and financial management system that offers modules for HR, payroll, and financial management.
- SAP SuccessFactors: The only difference with workday is that it includes talent management.

METHODOLOGY

The Agile methodology can be use for developing an Employee Management System (EMS). Agile methodology refers to an iterative and incremental approach that emphasizes flexibility, collaboration, and rapid delivery. This methodology passes through the following phases;

- **Requirement Gathering**: Collect and document requirements from stakeholders which includes employees and management.
- **System Analysis**: Analyse the requirements and define the system's functional and non-functional requirements.
- **System Design**: Design the EMS architecture, including the database, user interface and security features.
- **Implementation**: Develop the EMS using selected programming languages, frameworks and tools.
- **Testing**: Conduct unit testing, integration testing and system testing to ensure the EMS meets the requirements.
- **Deployment**: Deploy the EMS to the production environment and provides training to end-users.
- **Maintenance**: Provide ongoing maintenance, updates and support to ensure the EMS continues to meet the organization's needs.

TOOLS

Development Tools

- **Integrated Development Environment (IDE):** Visual Studio Code.
- **Database system:** Xampp.
- **Version Control: Git** (GitHub).
- **UI/UX Design Tools:** Figma.
- **Use Case Tool:** StarUML.
- **Graphics and Logo Design:** Canva

Programming languages

- Frontend: HTML, CSS, JAVASCRIPT (Frameworks: React)
- Backend: PHP
- Relation Database: MySQL.

REQUIREMENTS

Functional Requirement

1. **Employee Database Management**
   - Create, update and delete employee profiles.
   - Search and filter capabilities for quick access.

2. **Payroll Management**
   - Automated salary calculation based on attendance and performance.
   - Tax deductions and generation of payslip.

3. **Attendance management**
   - Track employee attendance and absences.
   - Leave request and approval workflow.
4. **Performance management**
   - track employee performance metrics and goals.
   - 360-degree feedback system.
5. **Reporting and Analytics**
   - generate reports and provide insights on employee data.
   - Dashboard for visual representation of key metrics.
6. **User Management**
   - Use registration and authentication processes.
   - Role-base access control for HR staff and employees.

Non-functional Requirement

1. **Security**
   - Robust security measure to protect sensitive employee data.
   - Regular security audits and compliance checks.
2. **Scalability**
   - Ability to handle increased traffic and data volume.
3. **Usability**
   - User-friendly interface and intuitive.
   - Accessible to both desktop and mobile devices.
4. **Performance**
   - Fast data processing.
   - Support for concurrent users without performance degradation.
5. **Maintainability and Reliability**
   - Easy to update, modify and maintain system.
   - System uptime off 99.9% with regular backups.

Technical Requirements

1. Programming Languages (front-end (Html, CSS, JAVASCRIPT), and for the backend I will use (PHP).
2. Frameworks (back-end framework I will use Node.js with Express.js and for the front-end I will use React).
3. Database (I will be using MySQL in this project).
4. Security (SSL/TLS, encryption, and access controls).

Non-Technical Requirement

1. User Needs (easy navigation, user-friendly interface and clear instruction).
2. Operational constraints (integration with existing HR systems, Compliance with relevant laws and regulations).

Approach

PROBLEM STATEMENT

The current employee management process is manual, time-consuming, and prone to errors. The organization needs an automated system to streamline HR processes, improve efficiency and enhance employee experience.

Objectives

- Automate HR process Improve efficiency.
- Enhance Employee Experience.
- Increase Accuracy.
- Ensure data security and compliance with regulations.

KEY FEATURES

- User management (create, update and delete employee profiles).
- Leave management (manage employee leave requests and approval).
- Attendance management (track employee attendance and absences).
- Performance management (track employee performance metrics and goals).
- Reporting and Analytics (generate reports and provide insights on employee data).

PLAN OF WORK

Week1: Requirements Gathering and Planning

- I will conduct stakeholder interviews to gather requirements.
- I will also create a detailed project plan with clear milestones and timelines.
- I will also develop a project schedule with specific tasks and deliverables.
- I will also define the project scope, goal, and objectives.
- I also Identify potential risks and develop mitigation strategies.

Week 2: Design and Prototyping

- I will also develop a wireframe prototype of the system.
- I will create a visual design concept for the system.
- I will define the system architecture and technical requirement.
- I will conduct usability testing and gather feedback.
- Create a data model and database design.

Week 3: Front-end Development

- In this stage I will develop the front-end of the system using the following programming language as mention above.
- I will implement user authentication and authorization.
- I will create a responsive and mobile friendly design.
- I will develop the user interface components and layouts.
- I will also integrate the front-end with the back-end API.

Week 4: Back-end Development

- I will also develop the back-end using the programming language as mention above.
- I will also develop or implement the business logic and data processing.
- I will integrate the back-end with the front-end.
- I will develop the API endpoints and data models.
- I will also implement security measures such as authentication and authorization.

Week 5: Testing, Deployment, and Maintenance

- During this the stage I will conduct unit testing, integration testing, and system testing.
- I will deploy the system to a production environment.
- I will conduct performance testing and schedule regular updates.
- I will conduct maintenance plan and schedule regular updates.
- I will also provide training and support for end-users.

NAME	**FON VANESSA AJI**
MATRICULE	**ENG24CSE141**