Topic :- SmartE (Smart Evolution)

1. **Introduction**

This Software Requirements Specification (SRS) document defines the functionality and requirements of the SmartE website, a web platform that allows employees to register, log their attendance, apply for leaves, and track their work details. The website is developed using ReactJS for the frontend and Python Django for the backend.

1.1 Purpose

The purpose of this website is to streamline employee management by providing a centralized platform for attendance logging, leave management, departmental assignments, and salary calculations based on attendance and leave records.

1.2 Scope

The SmartE website allows the following functionalities:

- Employee registration and login
- Attendance tracking (automatically records time and date upon login)
- Viewing and managing department and post assignments
- Leave application submission and approval workflow
- Leave deductions from employee leave balance (CL)
- Monthly salary calculation based on attendance and leave
- AI powered Employee Help Chatbot for handling employee queries

2. Functional Requirements

2.1 User Registration and Login

User Type: Employee

Functionality:

- Employees must register with personal details such as name, email, department, and post.
- Login credentials (email and password) are used for authentication.
- Employees can reset their passwords via email if they forget their login credentials.

2.2 Attendance Tracking

User Type: Employee

Functionality:

- When employees log in, the system records the current date and time as their attendance for that day.
- The attendance data (date, time, employee ID) is stored in the database for future reference and payroll calculations.

2.3 Department and Post Information

User Type: Employee

Functionality:

- Each employee can view which department they belong to and their assigned post.
- This information is available on the employee's dashboard after logging in.

2.4 Leave Management

User Type: Employee

Functionality:

- Employees can apply for leaves (e.g., Casual Leave (CL), Sick Leave (SL), etc.).
- The leave application must specify the start and end dates for the leave.
- Once submitted, the leave application is forwarded to the respective department manager for approval.

2.5 Leave Approval and Virtual Signature

User Type: Manager (Department Head)

Functionality:

- Managers can view all pending leave requests and review the details of the employee's leave application.
- Managers can approve or reject the leave request.
- Upon approval, the manager digitally signs the leave request using a virtual signature (e.g., drawing, click to sign).
- Once the leave request is approved, the system automatically updates the employee’s leave balance and records the leave data.

2.6 Leave Deduction from CL

User Type: Employee

Functionality:

- Upon the approval of leave, the system deducts the corresponding number of leave days from the employee's available Casual Leave (CL) balance.
- The leave balance is updated in real time and displayed to the employee on their dashboard.

2.7 Monthly Salary Calculation

User Type: Employee, Manager

Functionality:

- At the end of the month, the system calculates the salary based on:
- Number of days the employee has logged attendance.
- Number of leaves taken and deducted from the CL balance.
- The salary calculation is done automatically, taking into account the absence due to leaves and attendance data.
- The final salary is available to the employee on their dashboard for review.

2.8 AI Employee Help Chatbot

User Type: Employee

Functionality:

- The chatbot assists employees with queries related to:
- Attendance Queries: "What’s my attendance record this month?"
- Leave Management: "How many vacation days do I have left?"
- Payroll Information: "When is the next payday?"
- Company Policies: "What is the company's remote work policy?"
- IT Support: "How do I reset my password?"
- Personalized Notifications: Reminders for upcoming meetings, deadlines, etc.
- The chatbot uses Natural Language Processing (NLP) to understand and respond to queries.
- It integrates with the existing system to fetch real time data for accurate responses.

3. Non Functional Requirements

3.1 Performance Requirements

- The website should be responsive and accessible across all devices (mobile, tablet, desktop).
- The system should handle up to 1000 concurrent users.
- The database should store attendance and leave data efficiently for easy retrieval.
- The chatbot should respond to queries within 2 seconds for a seamless user experience.

3.2 Security Requirements

- All passwords should be stored securely using encryption (e.g., bcrypt).
- Sensitive data such as leave information and salary details should be encrypted and protected.
- The system should implement role based access control to ensure only authorized users can access sensitive features (e.g., managers approving leave).
- The chatbot should be secure, ensuring that no sensitive data is exposed through conversational interfaces.

3.3 Usability Requirements

- The user interface should be intuitive and easy to use.
- Employees should be able to navigate the website and perform actions (e.g., applying for leave, viewing attendance) without prior training.
- The website should provide clear feedback after each action, such as confirmation messages when a leave is applied or when attendance is successfully recorded.
- The chatbot interface should be simple, allowing employees to ask questions naturally.

3.4 Availability Requirements

- The website should have an uptime of 99.9% to ensure uninterrupted access for employees and managers.
- The system should provide real time updates to the leave balance and attendance data.
- The chatbot should be available 24/7 to assist employees with their queries.

4. System Architecture

4.1 Frontend

Technology Stack: ReactJS

- The frontend will be developed using ReactJS to build a responsive and dynamic user interface.
- Users will interact with the system through web pages where they can register, login, view attendance, apply for leaves, etc.
- The chatbot interface will also be integrated into the frontend for seamless access.

4.2 Backend

Technology Stack: Python Django

- Django will serve as the backend framework to handle business logic, database interactions, and API endpoints.
- Django's ORM will be used to manage database transactions such as storing attendance records, leave data, employee profiles, etc.
- Django Rest Framework (DRF) will be used to expose RESTful APIs for frontend backend communication.
- The chatbot backend will use Python based NLP libraries to process and respond to employee queries.

4.3 Database

Technology Stack: PostgreSQL or MySQL (Relational Database)

- Employee details, attendance records, leave data, and salary information will be stored in a relational database.
- Database tables will include entities such as employees, attendance, leave applications, and departments.

4.4 Chatbot Technology

Technology Stack:

- NLP Framework: Rasa, Dialogflow, or OpenAI GPT APIs
- Frontend Integration: ReactJS chatbot UI components
- Backend Integration: Django APIs to fetch real time data
- Deployment: Hosted on cloud platforms like AWS, Azure, or Google Cloud for scalability

5. Data Flow and Use Cases

5.1 Data Flow Diagram

1. Employee Registration and Login:

- Employee registers with personal details.
- The system verifies credentials during login.

2. Attendance Logging:

- The system records the current date and time when the employee logs in.
- Attendance data is stored in the database.

3. Leave Application:

- Employee applies for leave.
- Leave request is forwarded to the manager for approval.

4. Leave Approval:

- Manager reviews and approves/rejects the leave request.
- Upon approval, the leave balance is updated, and salary calculation is adjusted.

5. Salary Calculation:

- The system calculates the employee’s salary based on attendance and leave deductions.

6. Chatbot Interaction:

- Employee asks a query.
- The chatbot processes the query using NLP.
- The chatbot retrieves relevant data from the backend and responds to the employee.

5.2 Use Case Diagram

- Actors: Employee, Manager, System, Chatbot
- Use Cases:
- Employee: Register, Login, Apply for Leave, View Attendance, View Salary, Ask Chatbot for Information
- Manager: Approve/Reject Leave, View Employee Leave Requests
- System: Store Attendance, Track Leave Balance, Calculate Salary
- Chatbot: Answer Queries, Provide Notifications

6. Conclusion

This SRS document outlines the key features and requirements for the SmartE website, aimed at overseeing employee attendance, leave applications, and salary calculations. With ReactJS for the frontend and Python Django for the backend, the website will provide an efficient, user friendly platform for both employees and managers to streamline HOD processes. The addition of an AI powered Employee Help Chatbot will further enhance the user experience by offering quick, accurate responses to employee queries and providing personalized notifications.