



Software Requirement Specification for Contact Management

Name	Veveethan A K
Roll no	7376222IT279
Seat no	352
Project ID	28
Problem Statement	Contact Management

Phase Notes:

- ❖ Stage 1: Planning and Requirement gathering
- ❖ Stage 2: Design and Prototyping
- ❖ Stage 3: DB Designing
- ❖ Stage 4: Backend Implementation
- ❖ Stage 5: Testing & Implementation

1. Introduction

1.1. Purpose:

The purpose of the Contact Management System is to streamline the process of managing and tracking contacts made during industry visits. This system will help in recording and organizing details of each visit, including contact information, expertise, and meeting minutes. This ensures that all data is accessible and manageable for future reference and follow-up actions.

1.2. Scope of Project:

The Contact Management System aims to:

1. Record detailed information about industry visits.
2. Store contact information securely and efficiently.
3. Provide a user-friendly interface for adding, editing, and retrieving contact details.
4. Enable the management of meeting minutes and other relevant documents.
5. Offer search and filter functionalities to quickly find specific contacts or visits.

2. System Overview:

The system is developed using the MERN stack, leveraging the strengths of MongoDB for database management, Express.js and Node.js for server-side development, and React for client-side development.

It includes the following modules:

- **User Authentication:** Secure login for users.

- **Contact Management:** Adding, editing, and deleting contact details.
- **Visit Management:** Recording details of industry visits.
- **Expertise Classification:** Categorizing contacts by product and area of expertise.
- **Report Generation:** Summarizing visits and contact details.
- **Meeting Scheduling:** Arranging and tracking meetings with industry contacts.

3. Features:

Industry Visit Records:

Store details of each industry visit including the name, address, and location of the company.

Contact Information Management:

Keep records of contact persons including their name, contact number, and area of expertise.

Faculty Coordinators:

Assign and manage faculty coordinators for each visit.

Meeting Minutes:

Record and store the scope or minutes of meetings held during the visit.

Search and Filter:

Search contacts and visits by various criteria such as date, location, or expertise.

User Authentication:

Secure login and authentication to protect sensitive data.

Responsive Design:

A user-friendly interface accessible on various devices.

4. System Requirements Specification:

4.1 Functional Requirements:

User Authentication:

- Users should be able to register, log in, and log out securely.
- Access control to ensure only authorized users can add or edit contact information.

Contact Management:

- Users should be able to add, view, edit, and delete contact details.
- Ability to assign faculty coordinators to each contact.

Visit Management:

- Users should be able to record details of industry visits.
- Ability to attach meeting minutes and other relevant documents to each visit.

Search and Filter:

- Implement search functionality to find contacts or visits based on name, date, location, etc.

Data Storage:

- Store all data in a MongoDB database.

4.2. Non-Functional Requirements:**Performance:**

- The system should be able to handle multiple users and large volumes of data without performance degradation.

Usability:

- The system should be intuitive and easy to use for users with basic computer skills.

Security:

- Ensure data security through encrypted connections and secure authentication mechanisms.

Scalability:

- The system should be scalable to accommodate future growth in the number of users and data volume.

Reliability:

- High availability and minimal downtime.
- Regular backups and disaster recovery plans.

System Design

High-Level Architecture:

1. Front-End:

- React.js application interacting with the backend via API calls.

2. Back-End:

- Node.js server using Express.js to handle requests and responses.

3. Database:

- MongoDB for data storage.

8.Flowchart:

