Contact Management System

Introduction

The **Contact Management System** is a web-based application designed to help users organize, manage, and retrieve contact information efficiently. It allows users to store contacts, categorize them into groups, and manage detailed information such as phone numbers, emails, and notes.

The purpose of this system is to provide a **centralized and structured way** to maintain personal or professional contacts. By using a relational database, it ensures data integrity, easy retrieval, and scalability for future enhancements.

Database Design

ER Diagram

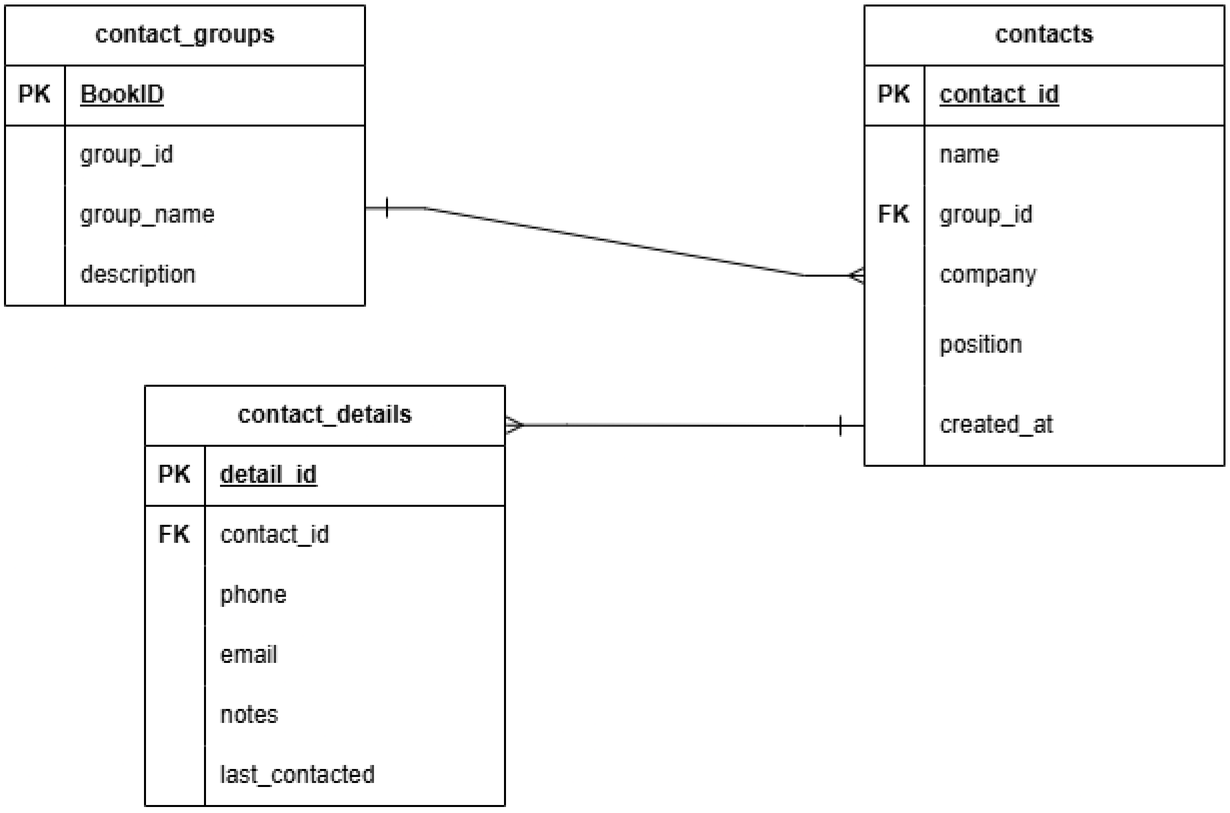

Table descriptions

Table	Purpose
contact_groups	Defines categories or groups (e.g., "Work", "Family").
contacts	Stores the primary information of a person (Name, Company, Position).
contact_details	Stores specific communication data (Phone, Email, Notes).

Relationships

1. **contact_groups → contacts**
   - One-to-Many (1 group → many contacts)
   - One group can contain many contacts, but a contact belongs to only one group.
2. **contacts → contact_details**
   - One-to-Many (1 contact → many contact details)
   - While a contact can technically have multiple detail entries in this schema, it is usually used to store specific info for one person.

Web Interface

Key Pages

1. **Home/Dashboard**
   - Overview of all contacts and groups.
2. **Contact Groups Page**
   - Add, edit, or delete groups.
   - List of all group with description
3. **Contacts Page**
   - List all contacts with their group, company, and position.
   - Add, edit or delete contact details.
4. **Contact Details Page**
   - View multiple phone numbers, emails, notes, and last contacted date.
   - Add new details or edit existing ones.
5. **Add/Edit Forms**
   - Forms with input validation for creating or updating contacts and details.

Key Functionalities

1. CRUD Operations: Create, Read, Update, Delete contacts, groups, and details.
2. Group Categorization: Contacts can be grouped for easier organization.
3. Data Integrity: Foreign key constraints ensure proper relationships.

Challenges and Learning

Well for me the most challenging is the start where you don't know how you system will go so you need to do a lot research and look for example so you can have an idea, Second, the javascript is kind of hard mostly in big projects so I just minimize it cause its so sensitive so I need it to split in their own file for easier debugging. Third, connecting the database actually at first I think that I cant use xampp and use php for backend that's why I use node.js which more complicated but very interesting. Overall, I learn how to connect database using node.js in server.js file, how foreign key constraint to ensure the relationship, manipulating the table using sql code, and debug the error in the overall system.