**Contact Management System**

A full-stack application to manage contacts, allowing users to create, update, delete, and view a list of contacts with features like duplicate detection for email and names. The app is built using **React** on the frontend, **Node.js** with **Express** on the backend, and **MongoDB** for the database.

**Features**

* **Add Contacts:** Users can add new contacts with fields like first name, last name, email, phone number, company, and job title.
* **Edit Contacts:** Modify details of existing contacts.
* **Delete Contacts:** Remove contacts from the database.
* **Validation:** Prevent duplicate contacts based on email or first name + last name.
* **Pagination & Sorting:** Paginate and sort the contact list by columns like first name, email, etc.

**Setup Instructions**

Follow these steps to set up and run the project on your local machine.

**1. Clone the Repository**

git clone https://github.com/bestcoder22/Contact-Management---Mini-Feature-of-a-CRM.git

cd contact-management

**2. Backend Setup**

**Install Dependencies**

Navigate to the Backend directory and install the dependencies:

npm install

**Configure the Database**

1. Create a MongoDB Atlas database (or use a local MongoDB instance).
2. Update the connection string in the server.js file:

Const uri="mongodb+srv://<username>:<password>@cluster.mongodb.net/<database-name>";

**Run the Backend Server**

Start the backend server:

node server.js

The backend will run on http://localhost:5000.

**Database Schema**

The MongoDB schema for a contact:

const contactSchema = new mongoose.Schema({

firstName: { type: String, required: true },

lastName: { type: String, required: true },

email: { type: String, required: true, unique: true },

phoneNumber: { type: String, required: true },

company: { type: String },

jobTitle: { type: String },

});

**3. Frontend Setup**

**Install Dependencies**

Navigate to the Frontend directory and install the dependencies:

npm install

**Run the Frontend**

Start the React development server:

npm start

The frontend will run on http://localhost:3000.

**Technical Decisions**

**Frontend**

* Built with **React** for component-based UI.
* **Material-UI (MUI)** for styling and UI components.
* **Axios** for making API requests to the backend.
* Added **pagination** and **sorting** for improved user experience.

**Backend**

* Built with **Node.js** and **Express**.
* **MongoDB** for data persistence.
* Validation and duplicate checks are implemented in the backend to ensure data integrity.

**Validation**

* **Duplicate email check**: Prevents adding or updating contacts with the same email.

**How the Application Works**

**1. Adding Contacts**

* Fill out the form with required fields (first name, last name, email, etc.).
* Validation prevents duplicate entries in the backend.
* New contacts are saved to the database and displayed in the frontend.

**2. Editing Contacts**

* Click the edit icon next to a contact to open the pre-filled form.
* Make changes and save them; the changes are updated in the database and reflected in the frontend.

**3. Deleting Contacts**

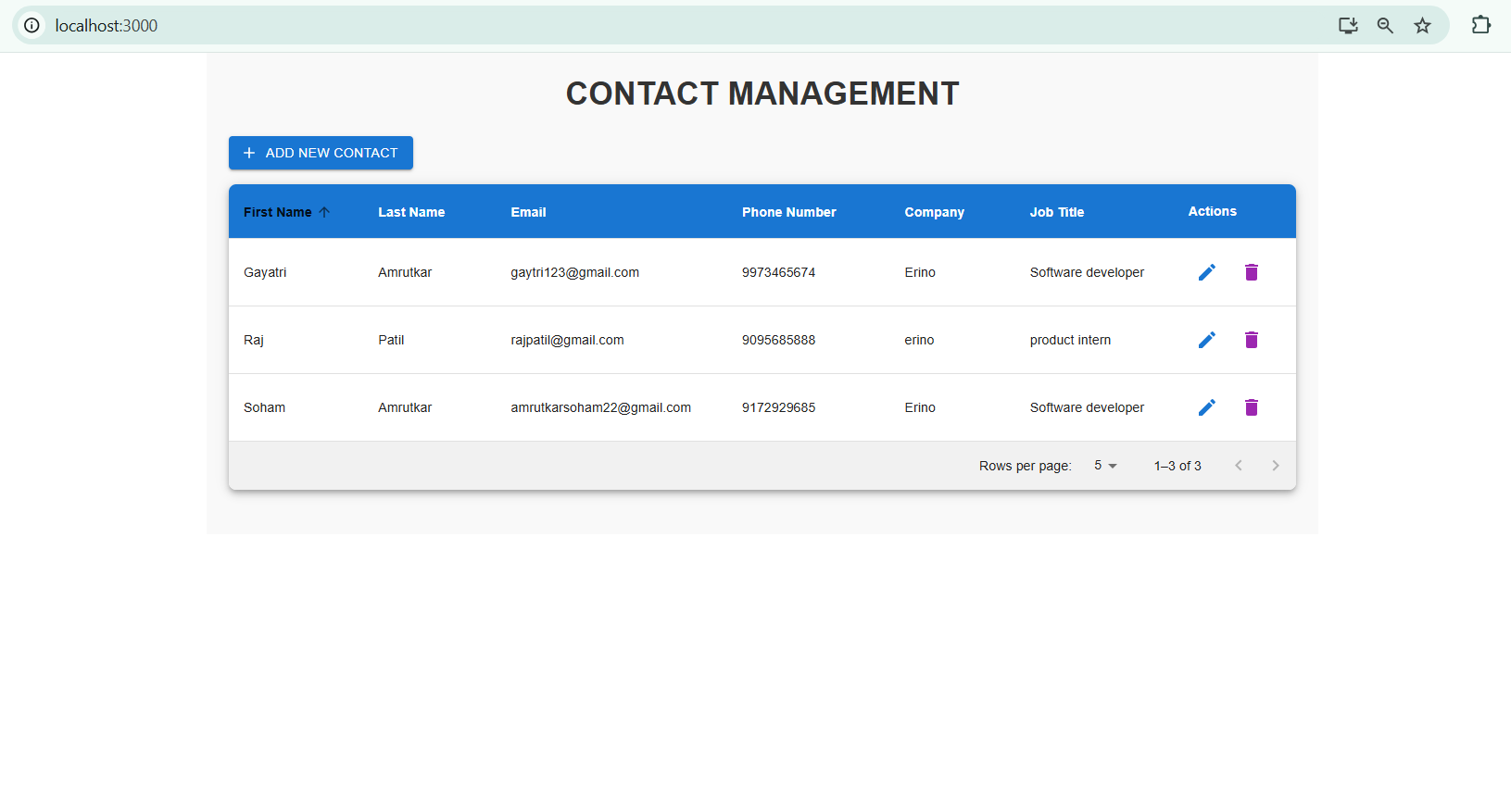
* Click the delete icon next to a contact to remove it from the database.

**4. Pagination and Sorting**

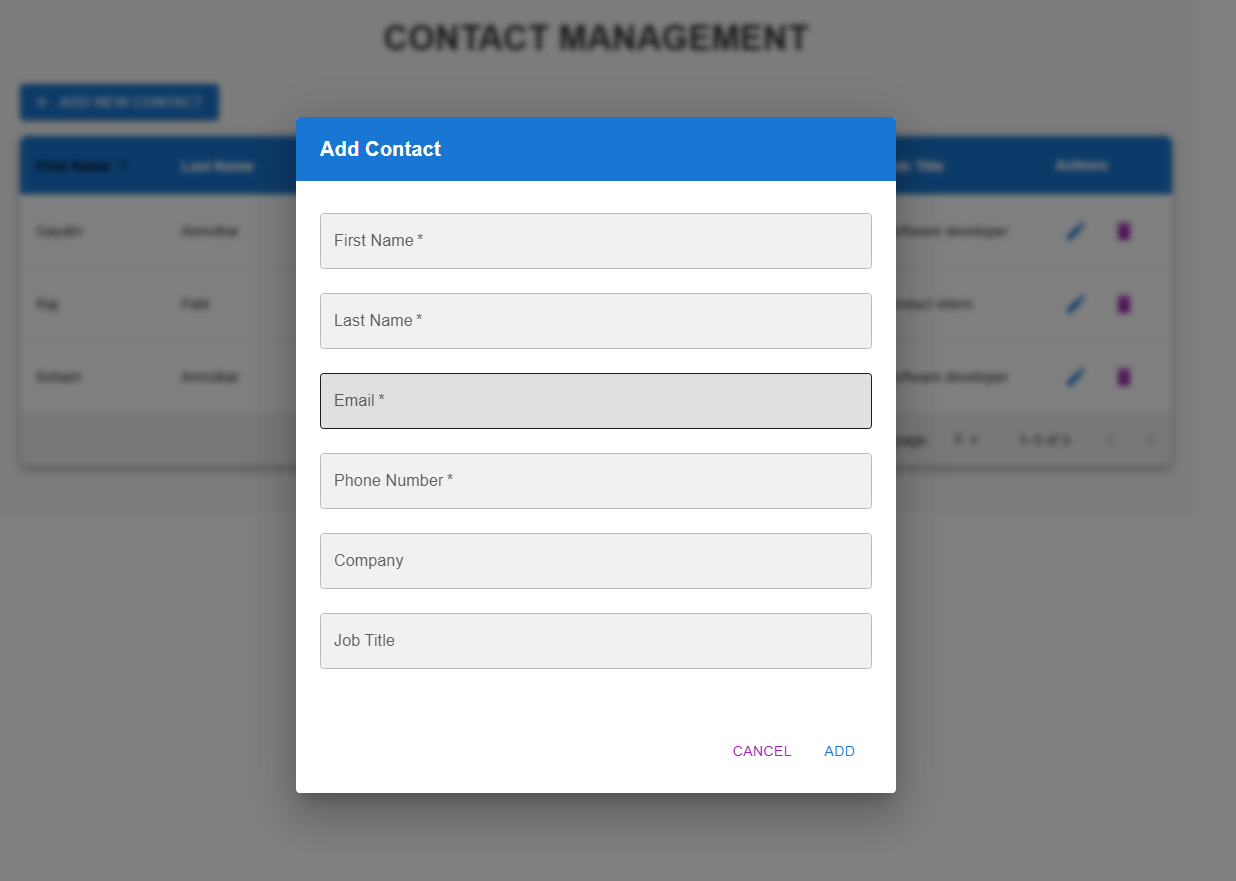
* Contacts are paginated and sortable by column headers for easy navigation.

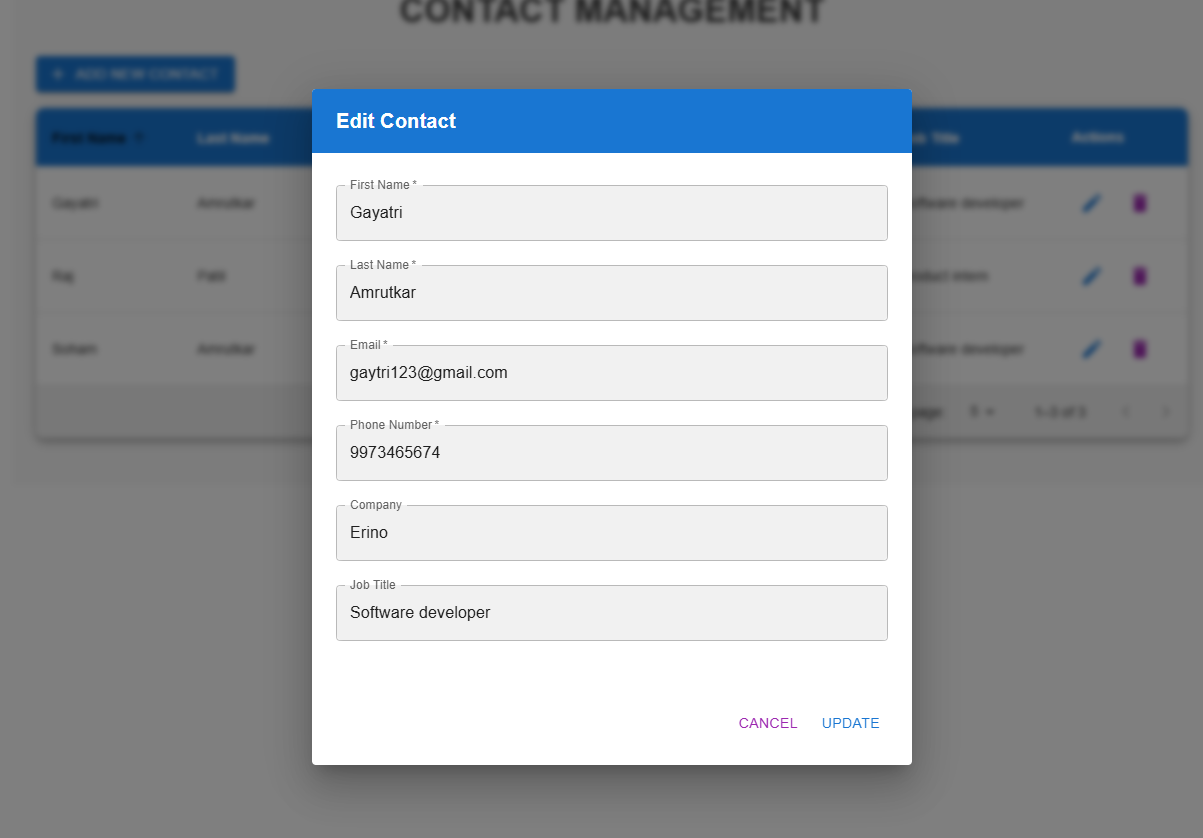
**Screenshots**

1. **Contact List**

****

**2. Add/Edit Contact**

****

****