

# Detailed Explanation for the Contact List Manager Project

---

## Project Overview

The Contact List Manager is a functional web application designed to perform essential CRUD (Create, Read, Update, Delete) operations for managing a list of contacts. The application focuses on clean code, problem-solving, and seamless user interaction. This project demonstrates full-stack development using **React** for the front-end and **Express.js** for the back-end, with data stored in a mock database or an alternative storage solution.

---

## Functionalities

### 1. Add Contacts

- **Feature:** Users can add new contacts to the system by providing their **name** and **email**.
  - **Validation:**
    - Email addresses must follow the **@gmail.com** format.
    - Duplicate email entries are prohibited. If a user attempts to add a contact with an already existing email, an error message will appear.
  - **Backend Endpoint:**
    - **POST /addContact**
    - Stores the new contact in the database.
- 

### 2. View Contacts

- **Feature:** Displays all stored contacts in a grid/table format.
  - **Backend Endpoint:**
    - **GET /getContacts**
    - Fetches all contacts from the server.
  - **Design:**
    - The table includes columns for **Name**, **Email**, and **Actions** (e.g., Delete).
    - Alternating row colors for better readability.
-

### 3. Search Contacts

- **Feature:** Users can search for contacts by name or email.
  - **Behavior:**
    - If matching contacts are found, they are highlighted in the grid (using blue borders for rows) and displayed to the user.
    - If no matches are found, an alert message (**Search Not Found**) is displayed.
  - **Backend Endpoint:**
    - `GET /searchContacts?query=<search-term>`
    - Filters contacts based on the search term provided by the user.
- 

### 4. Delete Contacts

- **Feature:** Users can delete specific contacts from the list using the **Delete** button in the grid.
  - **Backend Endpoint:**
    - `DELETE /deleteContact/:id`
    - Removes the selected contact from the database.
- 

### 5. Navigation (Home Page & Contact List Manager)

- **Feature:**
    - Users can navigate between a **Home Page** and the **Contact List Manager** page.
    - The Home Page displays a welcoming message and brief instructions on using the application.
    - The Contact List Manager allows users to manage their contacts.
  - **Design:**
    - Buttons for navigation (**Home Page** and **Contact List**) are styled in **light brown** for a clean, modern look with rounded corners and hover effects.
- 

## Deliverables

### 1. Code

- **Single GitHub Repository:**
  - The project code is structured into **frontend** and **backend** folders for clarity.
  - The repository contains clear, modular code for easy readability and maintenance.
- **Commit History:**

- Demonstrates a well-organized workflow.
  - Each commit represents a specific change (e.g., "Implemented email validation" or "Styled navigation buttons").
- 

## 2. Documentation

- **README.md:**
    - **Setup Instructions:**
      - Clone the repository: `git clone <repo-url>`.
      - Navigate to the backend folder: `cd backend`.
      - Install dependencies: `npm install`.
      - Start the server: `node server.js`.
      - Navigate to the frontend folder: `cd ../frontend`.
      - Install dependencies: `npm install`.
      - Start the application: `npm start`.
    - **Approach:**
      - **Front-End:** Built with React, focusing on reusable components, clean state management, and dynamic user interaction.
      - **Back-End:** Implemented using Express.js to handle API requests and manage the database efficiently.
      - **Design Decisions:**
        - **Validation:** Ensured that user input is validated both on the client-side and server-side for security and usability.
        - **Search Highlighting:** Added a visual indicator for matching search results to improve user experience.
        - **Responsiveness:** Designed the UI to be accessible on different screen sizes.
- 

## Project Design

### Frontend

- **React:**
  - **State Management:** Managed using `useState` and `useEffect` hooks.
  - **Components:**
    - `App.js`: The main component containing the overall structure and logic.
    - Form sections for adding and searching contacts.
    - A table for displaying the contact list.

### Backend

- **Express.js:**
    - **API Endpoints:**
      - **POST /addContact:** Adds a new contact to the database.
      - **GET /getContacts:** Fetches all contacts.
      - **GET /searchContacts?query=<search-term>:** Filters contacts based on the search term.
      - **DELETE /deleteContact/:id:** Deletes a specific contact.
- 

## Key Features

1. **Validation:**
    - Ensures emails are in the correct format (**@gmail.com**) and prevents duplicates.
  2. **Search Highlighting:**
    - Enhances usability by visually distinguishing matching rows in the table.
  3. **Responsive Design:**
    - The application is styled to look great on devices of all sizes.
  4. **Modern UI Design:**
    - Buttons are styled with hover effects and proper spacing for better accessibility.
  5. **Clean Code:**
    - Both the front-end and back-end are organized into logical components/modules.
- 

## Technologies Used

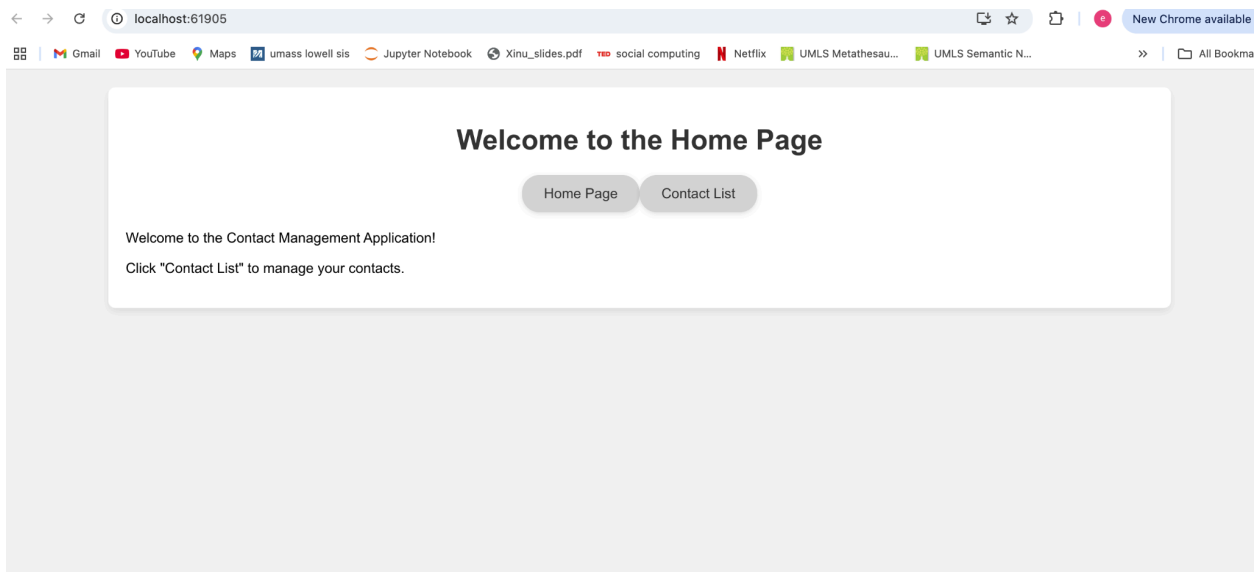
1. **Frontend:**
  - React
  - Axios (for API requests)
  - CSS (for styling)
2. **Backend:**
  - Node.js
  - Express.js
3. **Others:**
  - npm (for package management)

---

The Contact List Manager is a comprehensive solution for managing contact information efficiently. It demonstrates expertise in full-stack development, clean code practices, and modern UI/UX design. By adhering to the deliverables and documentation guidelines, this project is ready to meet the expectations of the provided requirements.

## Application Overview:

### Home screen:



### Contact List Manager Screen:

Contact List Manager

Home Page

Contact List

Enter Name

Enter Email

Add Contact

Search by Name or Email

Search

Name	Email	Actions
test	test@gmail.com	<div>Delete</div>
eswar	eswar@gmail.com	<div>Delete</div>

Jupyter NoteMetathesau...

localhost:61905 says

Name and email are required.

OK

Home Page

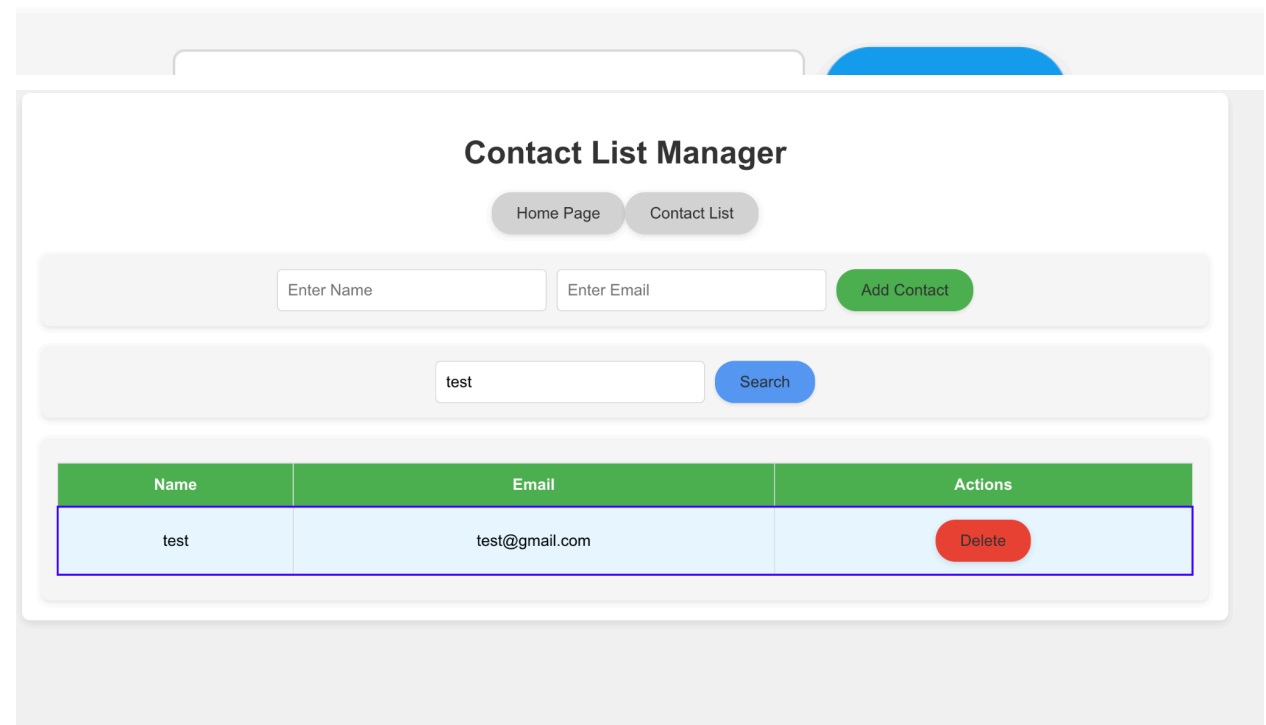
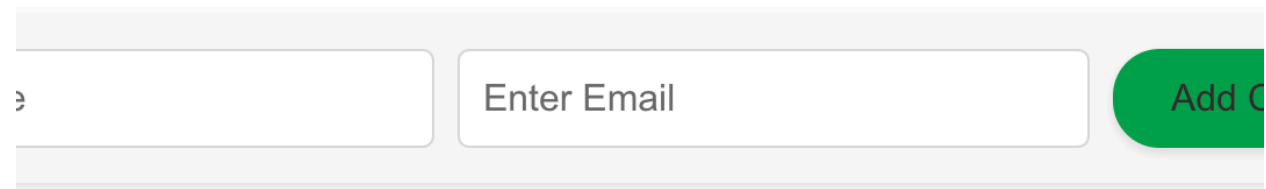
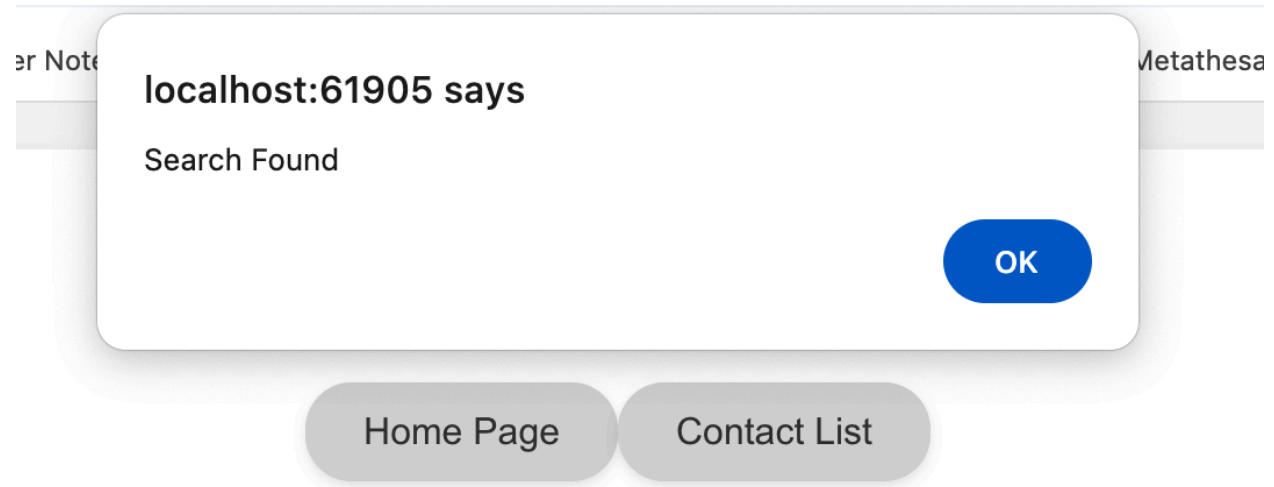
Contact List

Name

Enter Email

Add Cor

Search Functionality:



## Duplicate entries:

