**Contact List Application Analysis**

1. **Overview**

The Contact List Application is designed for managing user contacts through a web-based interface. It uses a REST API to perform operations on user and contact data. The API endpoints are documented via Postman, and the application provides UI forms for interaction.

**Technologies Used:**  
 **Frontend:** HTML, CSS, JavaScript  
 **Backend:** REST APIs  
 **Database:** Likely MYSQL database for user and contact data.  
 **Testing Tool:** Postman for API testing and documentation.  
 **Purpose:** To demonstrate how APIs integrate with a web application for

managing user and contact data.

**2** **System Architecture**  
  
 The application architecture can be summarized as follows:  
  
 **2.1 Frontend**

• Provides the interface for user interactions (e.g., login, contact list, add contact).  
 • Makes HTTP requests to the backend API using forms and buttons.  
  
**2.2 Backend**

• Implements Restful endpoints to handle (user authentication, contact management).   
 • Validates input, processes requests, and interacts with the database.  
  
**2.3 Database**

• Stores user information( name, email, password) and contact data ( name, phone).

**3 API Documentation and UI Interaction**  
  
 The API consists of endpoints grouped into two main categories:  
 1. **Contacts API**  
 2. **Users API**

Below is a breakdown of each API method, Action, and how it is triggered via the UI.

**3.1 Contacts API**

|  |  |  |
| --- | --- | --- |
| **API** | **Triggers in UI** | **Action** |
| Post – Add contact | Click on “Add New Contact” button. | Submit a form with contact details( name ,phone, email , etc) to add new contact. |
| GET – Get Contact List | Open “Contact list” page. | Fetching and display the list of all contacts. |
| GET – Get Contact | Click on any contact name in contact list page. | Fetching and display the details of the selected contact. |
| PUT – Update Contact | Click on “Edit Contact” button on contact details page. | Submit the updated contact information to the API. |
| PATCH – Update Contact | Edit specific field of the contact. | Update the modified field without affecting others. |
| DEL – Delete Contact | Click on “Delete Contact” button on contact details page. | Send a request to delete the selected contact. |

**3.2 Users API**

|  |  |  |
| --- | --- | --- |
| **API** | **Triggers in UI** | **Action** |
| POST – Add User | Click on “Sign Up” button on contact list app page. | Submit the registration form with user details to create new user. |
| GET – Get User Profile | Click on any user name in contact list page. | Fetching and display the details of the selected user. |
| PATCH – Update User | Click on edit button or modify a specific fields(e .g email). | Submit updated user information to the API. |
| POST – Log Out User | Click on “Log out” button. | Invalidates user session and redirects to log in page. |
| POST – Log In User | Submit the log in form ( email, password). | Authenticates the user and starts a session. |
| DEL – Delete User | Click on delete button. | Send a request to delete the user account. |

**4 Detailed API Analysis**

**4.1 Contact API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Endpoint** | **HTTP Method** | **Parameters** | **Auth-Token** | **Request Body** | **Response Status** |
| /contacts | POST | None | Bearer{{token}} | {  "first Name":"John",  "last Name":"Doe", "birth date":"1970-01-01",  "email":"jdoe@fake.com",…} | 201 Created A new resource was created successfully. |
| /contacts | GET | None | Bearer{{token}} | None | 200 OK Request successful. The server has responded as required. |
| /contacts/{id} | GET | None | Bearer{{token}} | None | 200 OK Request successful. The server has responded as required. |
| /contacts{id} | PUT | None | Bearer{{token}} | {   "first Name":"Amy",  "last Name":"Miller",  "birth date":"1992-02-02","email": [amiller@fake.com](mailto:amiller@fake.com),...} | 200 OK Request successful. The server has responded as required. |
| /contacts{id} | PATCH | None | Bearer{{token}} | {      "first Name": "Anna"  } | 200 OK Request successful. The server has responded as required. |
| /contacts{id} | DELETE | None | Bearer{{token}} | None | 200 ok, { "message": "Contact deleted " } |

**4.2 Users API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Endpoint** | **HTTP Method** | **Parameters** | **Auth-Token** | **Request Body** | **Response Status** |
| /users | POST | None | Bearer{{token}} | {      "first Name": "Jakob",      "last Name": "Doe",      "email": "Jakob@Brown.com",      "password": "Jakob2345@"  } | 201 Created A new resource was created successfully. |
| /users/me | GET | None | Bearer{{token}} | None | 200 OK Request successful. The server has responded as required. |
| /users/me | PATCH | None | Bearer{{token}} | {"first Name":"Updated", "last Name": "Username", "email": "test2@fake.com",  "password": "my New Password"} | 200 OK Request successful. The server has responded as required. |
| /users/me/Login | POST | None | - | {      "email": "Jakob@Brown.com",      "password": "Jakob2345@"  } | 200 OK Request successful. The server has responded as required. |
| /users/me/Logout | POST | None | Bearer{{token}} | None | 200 OK Request successful. The server has responded as required. |
| /users/me | DELETE | None | Bearer{{token}} | None | 200 ok, { "message": "Contact deleted " } |

**5 Testing**  
• **Functional Testing:**  
• Verify API endpoints using Postman.  
• Test UI interactions for expected behaviors.  
• **Negative Testing:**  
• Test invalid inputs, duplicate records, and unauthorized actions.  
• **Performance Testing:**  
• Check the system’s behavior under high loads.