UI-API Mapping: Contact List App API Testing

Index

Introduction

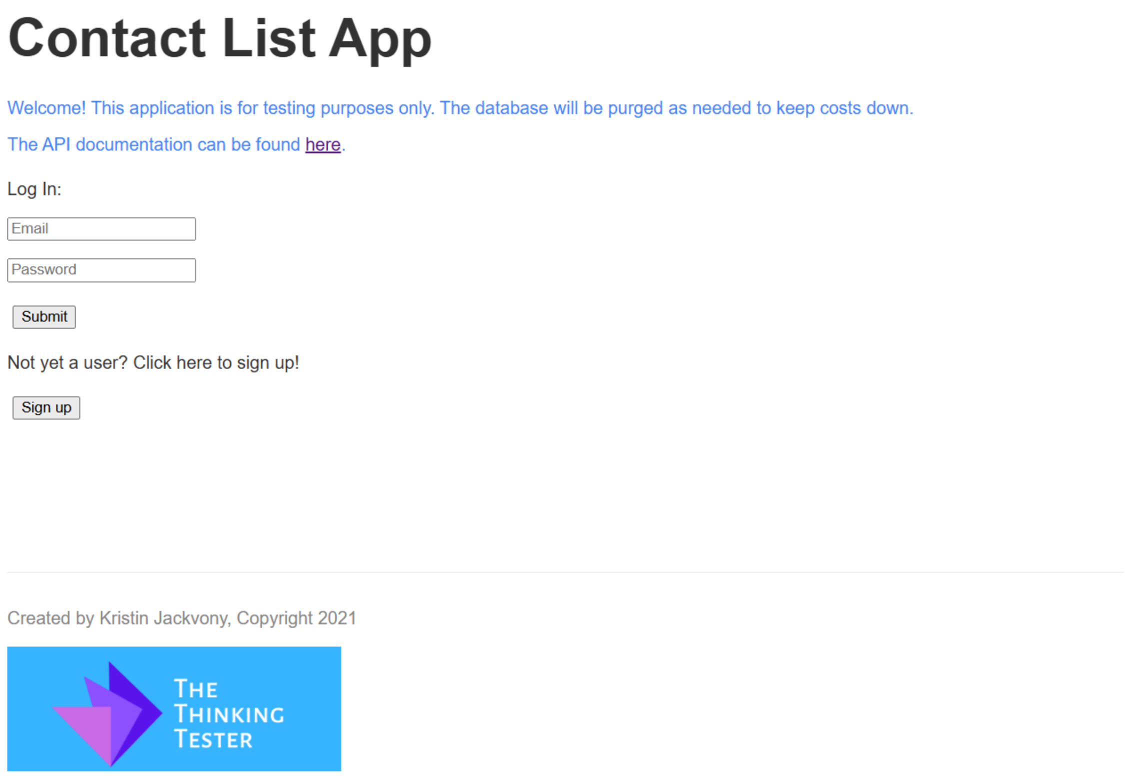
This document details the UI-API mapping for the "Contact List App"

https://thinking-tester-contact-list.herokuapp.com/

A live web application. While users interact with its UI, all core functionalities are powered by a RESTful API. This mapping provides precise details (endpoints, methods, payloads) of API requests triggered by specific user actions, serving as a crucial guide for API testing and ensuring independent backend functionality.

A- Users

1- Login User



Contact List App

Welcome! This application is for testing purposes only. The database will be purged as needed to keep costs down.

The API documentation can be found here.

Log In:

Not yet a user? Click here to sign up!

Created by Kristin Jackvony, Copyright 2021

Method: POST

Endpoint : https://thinking-tester-contact-list.herokuapp.com/users/login

Purpose

Authenticates a user by validating their email and password. Upon successful authentication, the server issues a token to authorize future requests.

UI Context

This operation is triggered when the user submits the login form on the application's initial screen. The form includes fields for email and password, along with a "Submit" action.

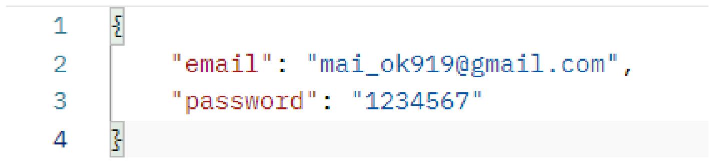
1.1 Successful Login

UI & API Workflow :

Step 1 : The application loads & User accesses the login screen which contains : Email & Password fields along with submit button .

Step 2 : User enter credentials (email & Password) .

Step 3 : User clicks Submit, then the frontend triggers a POST request to the login API with user's credentials :



```json
{
    "email": "mai_ok919@gmail.com",
    "password": "1234567"
}
```

Step 4 : The server receives the request and validates the credentials against the user database & Determines if the credentials match an existing user.

Step 5 : If the login details are correct, the server sends back a 200 OK **response** along with an **auth token** and some basic user info.

Once the frontend receives the response:

- The token is included **both in the response body** (token) and **as a cookie**.
- The user is then **automatically redirected** to the Contact List Dashboard.
- From now on, this token will be included in all API requests to keep the user logged in and allow access to protected features.

Response Body :

```
{
  "user": {
    "_id": "687e6dd77bc5660015e3b6fa",
    "firstName": "Mai",
    "lastName": "Okasha",
    "email": "mai_ok919@gmail.com",
    "__v": 4
  },
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdlNmRkNzdiYzU2NjAwMTVlM2I2ZmEiLCJpYXQiOjE3NTMxMTg3MDF9.26QLS5O9aaf47b6CEcABrWTuwW-Gi-zlUgj9C3bra54" }
```

1.2 Failed Login – User doesn’t exist

Ui & API Workflow :

Step 1 : User fills the login form with an email and password that do not match any existing account.

Step 2 : User clicks the submit button.

Step 3 : A POST request is sent to the login API endpoint : https://thinking-tester-contact-list.herokuapp.com/users/login

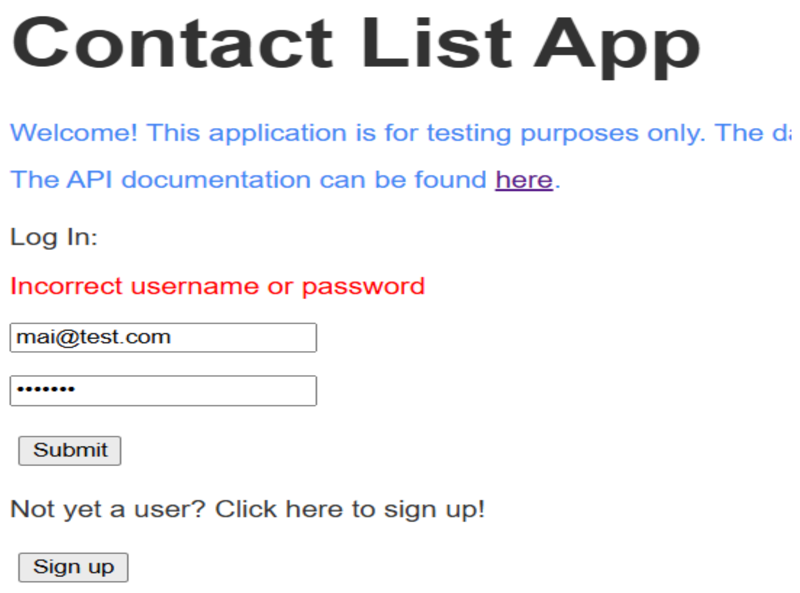
with the entered credentials :

```
{

   "email": "mai_fuad@gmail.com",

   "password": "1234567"

}
```

Step 4 : The server checks the provided email against its user database and does not find a match.

Step 5 : The server will send back a `401 Unauthorized` response .

Step 6 : Ui will receive the response & display an error message => "Incorrect username or password " as shown below :


Contact List App

Welcome! This application is for testing purposes only. The d

The API documentation can be found here.

Log In:

Incorrect username or password

mai@test.com

•••••••

Not yet a user? Click here to sign up!

1.3 Failed Login – Empty Fields

Ui & API Workflow :

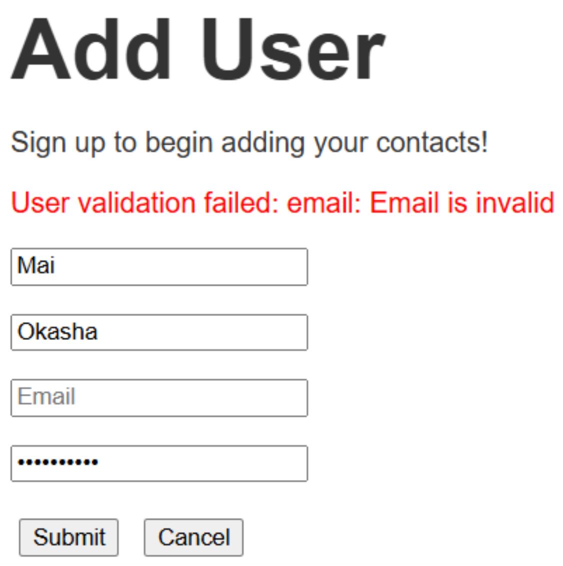
Step 1 : User clicks the Submit button on the Login screen without filling in one or both required fields (email or password).

Step 2 : Post Request with empty fields is sent from ui to server with payload

```
{
  "email" : " ",
   "password": "1234567"
}
```

Step 3 : Server receives the request and detect missing required fields then responds with `401 Unauthorized` and an empty body request .

Step 4 : Frontend receives the error response and shows an error message on the UI as shown below :



Add User

Sign up to begin adding your contacts!

User validation failed: email: Email is invalid

Mai
Okasha
Email
••••••••••

1.4 Failed Login – Incorrect Password

Ui & API Workflow :

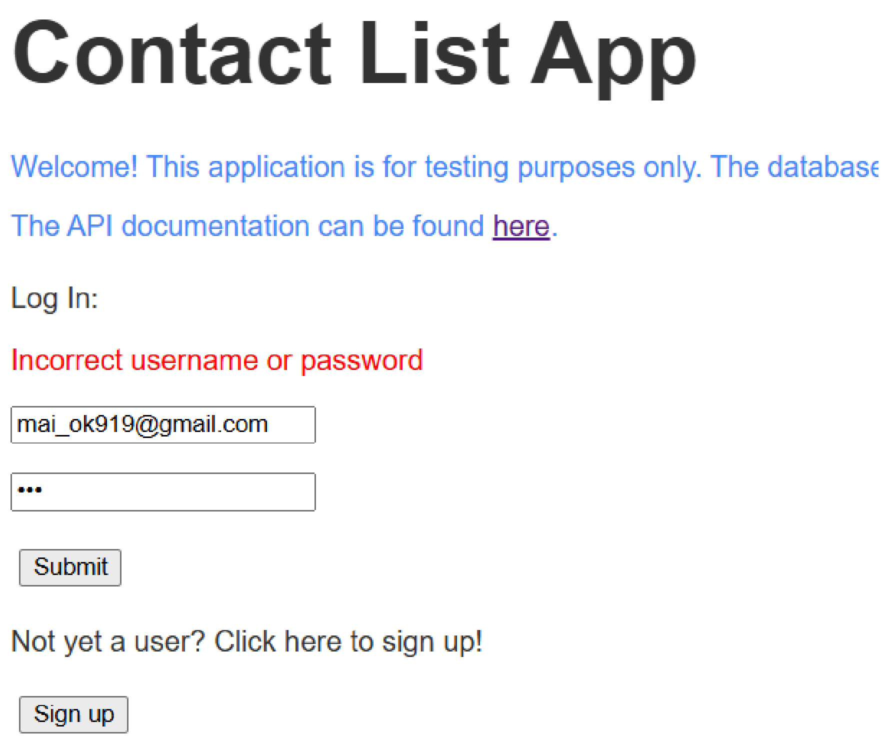
Step 1 : User enters a valid email but an incorrect password in the login form.
Step 2 : User clicks the submit button.
Step 3 : Frontend sends a POST request to the login endpoint with the entered credentials.
Step 4 : Server receives the request and checks the credentials.

Since the password is incorrect, it returns:

Status: 401 Unauthorized

And an empty body response .

Step 5 : Ui display en error message as shown below :



Contact List App

Welcome! This application is for testing purposes only. The database

The API documentation can be found here.

Log In:

Incorrect username or password

mai_ok919@gmail.com

•••

Not yet a user? Click here to sign up!

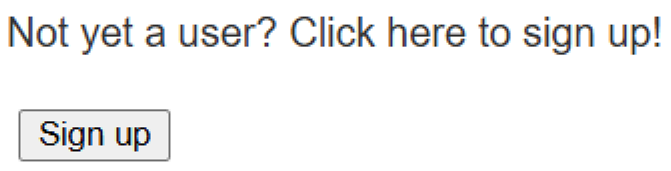
2- Sign Up

Purpose :

Add a new user by collecting and validating their first & last name, email, and password. Upon successful registration, the server creates the account and issues an authentication token for future requests.

Ui Context :

Users without an existing account should click the 'Sign Up' button on the login screen to create a new account



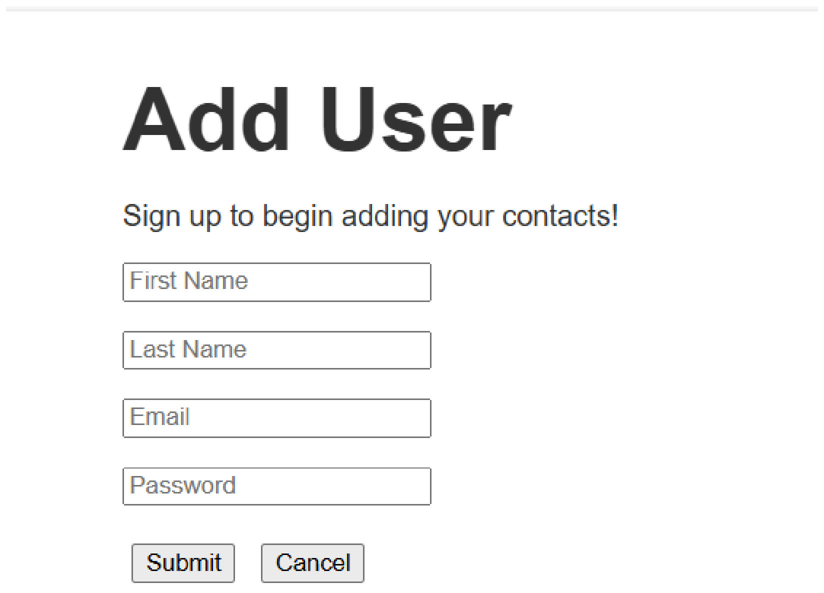
Then User will navigated to the sign up screen or **Add User** Screen which has a form for following fields :

First Name , Last Name , Email , Password

Then Following buttons :

Submit , Cancel

As shown below :


Add User

Sign up to begin adding your contacts!

2.1 Successful Sign Up

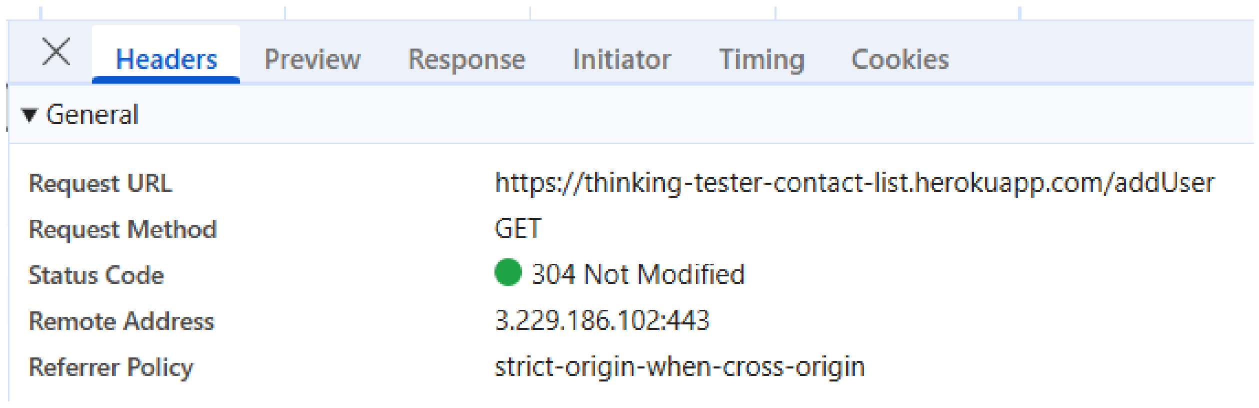
Method : Post

Endpoint : https://thinking-tester-contact-list.herokuapp.com/users

UI & API Workflow :

Step 1 : User clicks into the sign up button from the Login Screen .

Step 2 : User navigated to Add user screen and Frontend sent a Get Request with add user endpoint (Just to get Add user screen displayed) .



Request URL	https://thinking-tester-contact-list.herokuapp.com/addUser
Request Method	GET
Status Code	● 304 Not Modified
Remote Address	3.229.186.102:443
Referrer Policy	strict-origin-when-cross-origin

Step 3 : User fills out their email and password, then clicks the Submit button.

Step 4 : Frontend sends a Post Request to the server to the Sign Up API endpoint with the following payload

```
{

  "firstName": "Mai",

  "lastName": "Okasha",

  "email": "mai_ok919@gmail.com",

  "password": "1234567"

}
```

Step 5 : The server receives the request with its payload then validates the email .

If the email is new it creates the user and responds with 201 Created

- A response body containing an **auth token** and user details .

  ```
  {

    "user": {

      "_id": "687e806a40c4990015bf6a8e",

      "firstName": "Mai",

      "lastName": "Okasha",

      "email": "mai_okasha919@gmail.com",

      "__v": 1
  ```

```
    },

    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdlODA2YTQwYzQ5OTAwMT
ViZjZhOGUiLCJpYXQiOjE3NTMxMjA4NzR9.vpfmkUYdSEvQyUFT6wechTcb0ey9zx5lPft
gYd_U9dA"

}
```

Step 6 : User will be navigated to Contact List Screen .

2.2 Failed Sign Up - Email is already exist

Ui & API Workflow :

Step 1 : User fills in the sign up form with their details, including an email that is already registered.

Step 2 : User clicks the "Submit" button to create an account.

Step 3 : Frontend sends a POST request to the API endpoint:

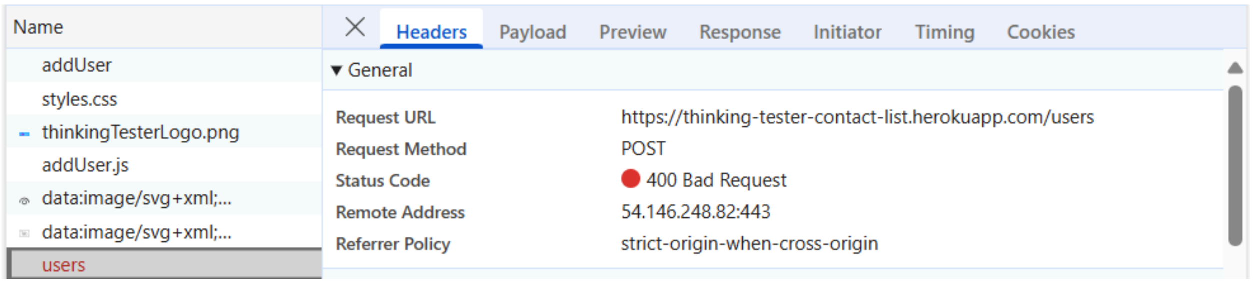
https://thinking-tester-contact-list.herokuapp.com/users

Step 4 : The server checks the submitted email against existing records.

Step 5 : Since the email is already registered, Server will respond with a

400 Bad Request and a

{"message":"Email address is already in use"} .

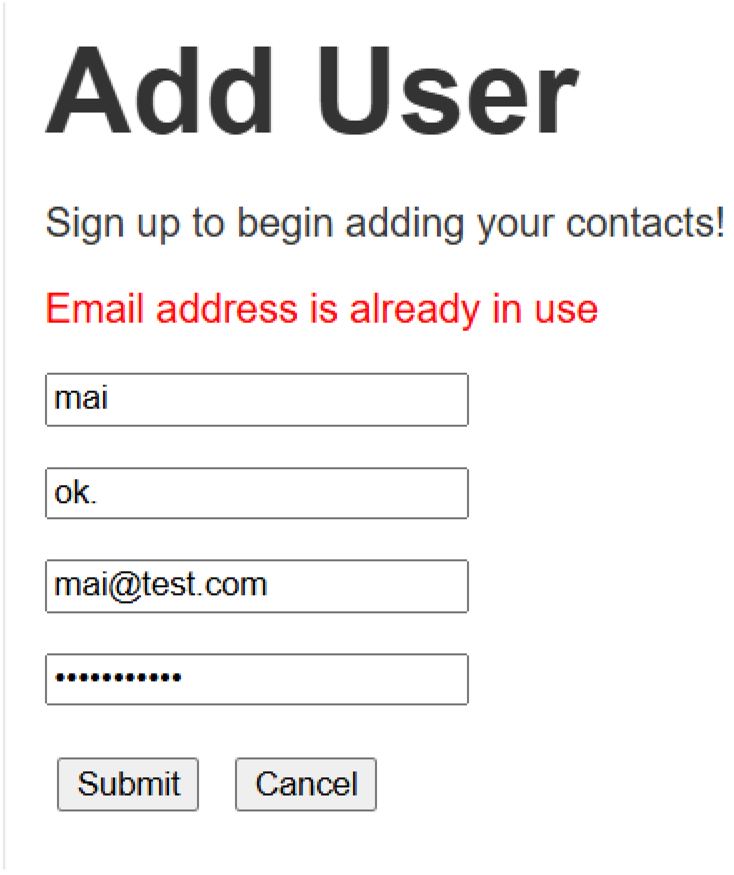


Name
addUser
styles.css
thinkingTesterLogo.png
addUser.js
data:image/svg+xml;...
data:image/svg+xml;...
users

▼ General

Request URL	https://thinking-tester-contact-list.herokuapp.com/users
Request Method	POST
Status Code	● 400 Bad Request
Remote Address	54.146.248.82:443
Referrer Policy	strict-origin-when-cross-origin

Step 6 : User will stay at Add user screen and UI will show an error message as shown below :



Add User

Sign up to begin adding your contacts!

Email address is already in use

mai

ok.

mai@test.com

••••••••••••

2.3 Failed Sign Up – Short Password

Ui & API Workflow :

Step 1 : User fills in the sign up form but enters a short password (Less than 7 characters) .

Step 2 : User clicks the "Submit" button to create an account.

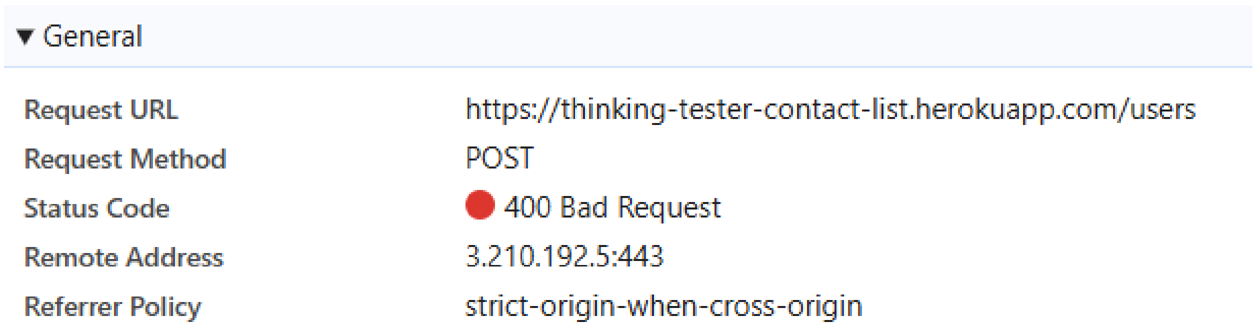
Step 3 : Frontend sends a POST request to the API endpoint:

https://thinking-tester-contact-list.herokuapp.com/users

Step 4 : Server validates the password length .

Step 5 : The server responds with a 400 Bad Request and Response body include :

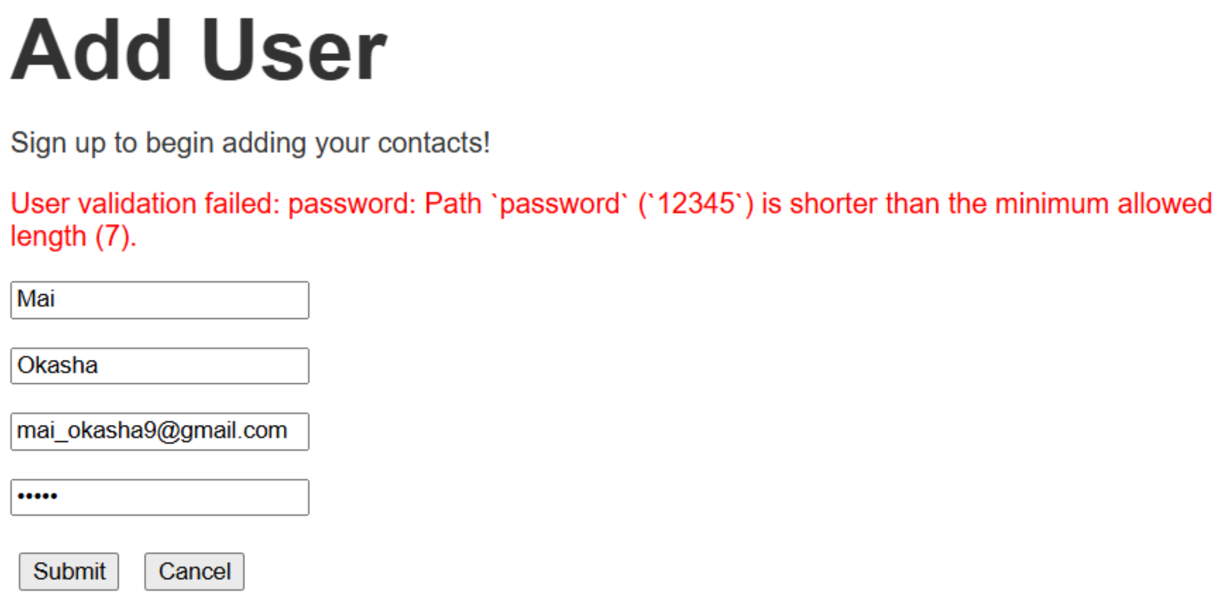
```
"message": "Path `password` (`12345`) is shorter than the minimum allowed length (7).",

"type": "minlength"

  "minlength": 7,

            "path": "password",

            "value": "12345"

        },
```



▾ General

Request URL	https://thinking-tester-contact-list.herokuapp.com/users
Request Method	POST
Status Code	● 400 Bad Request
Remote Address	3.210.192.5:443
Referrer Policy	strict-origin-when-cross-origin

Step 6 : The frontend remains on the Sign Up screen and shows an error message as shown below :


Add User

Sign up to begin adding your contacts!

User validation failed: password: Path `password` (`12345`) is shorter than the minimum allowed length (7).

Mai

Okasha

mai_okasha9@gmail.com

•••••

2.4 Failed Sign Up – Missing Fields

Ui & API Workflow :

Step 1 : User submits the form without filling in all the required fields.

Step 2 : User clicks the "Submit" button.

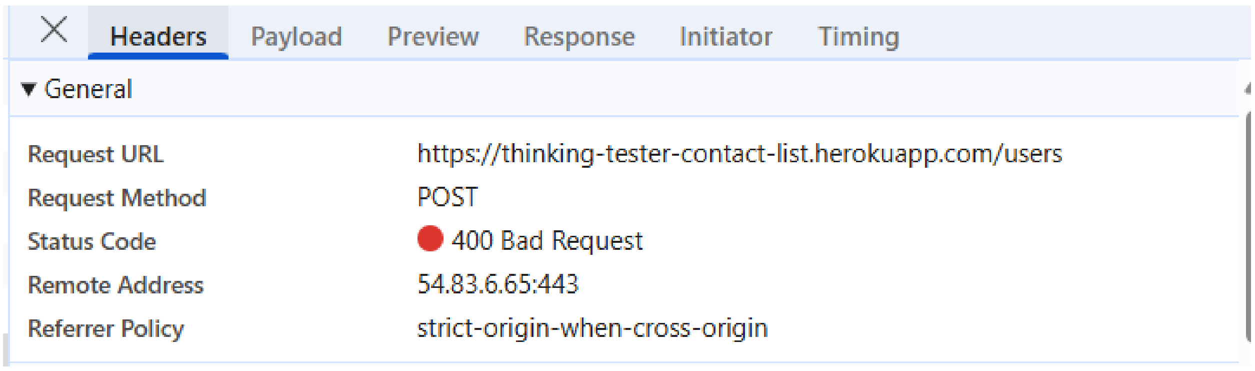
Step 3 : Frontend sends a POST request to the API endpoint:

https://thinking-tester-contact-list.herokuapp.com/users

Step 4 : The server checks for missing required fields.

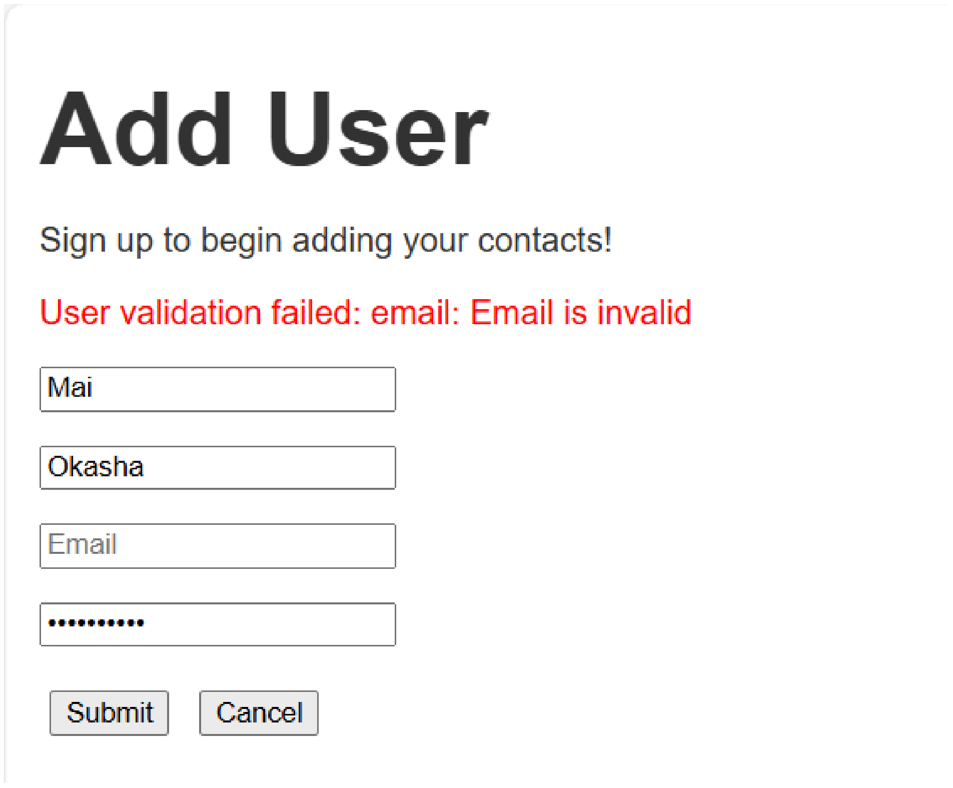
Step 5 : Server Responds with 400 Bad Request and Response body include :

```
"message": "Email is invalid",
        "properties": {
           "message": "Email is invalid",
           "type": "user defined",
           "path": "email",
           "value": "",
           "reason": {}
        },
```



Request URL	https://thinking-tester-contact-list.herokuapp.com/users
Request Method	POST
Status Code	● 400 Bad Request
Remote Address	54.83.6.65:443
Referrer Policy	strict-origin-when-cross-origin

Step 6 : Ui display en error message as shown below :


Add User

Sign up to begin adding your contacts!

User validation failed: email: Email is invalid

Mai

Okasha

Email

••••••••••

2.4 Failed Sign Up – Invalid Email Format

UI & API Workflow :

Step 1 : User types an email in an invalid format and enters first & last name and password .

Step 2 : User clicks the submit button .

Step 3 : Ui sends a Post request with payload :

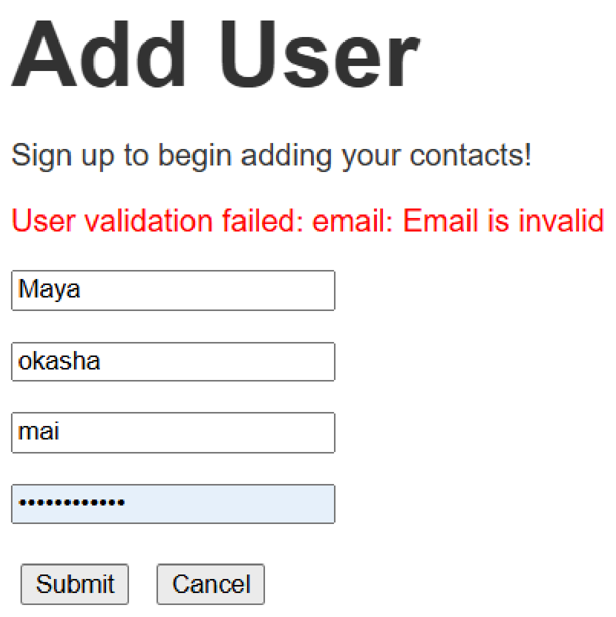
{firstName: "May", lastName: "okasha" , email: "mai" , password: "12345233558h"}

Step 4 : Server receive the request then respond with a **400 Bad Request**

And a message :

message: "User validation failed: email: Email is invalid"

Step 5 : Ui will display an error massage as shown below :


Add User

Sign up to begin adding your contacts!

User validation failed: email: Email is invalid

Maya

okasha

mai

••••••••••••

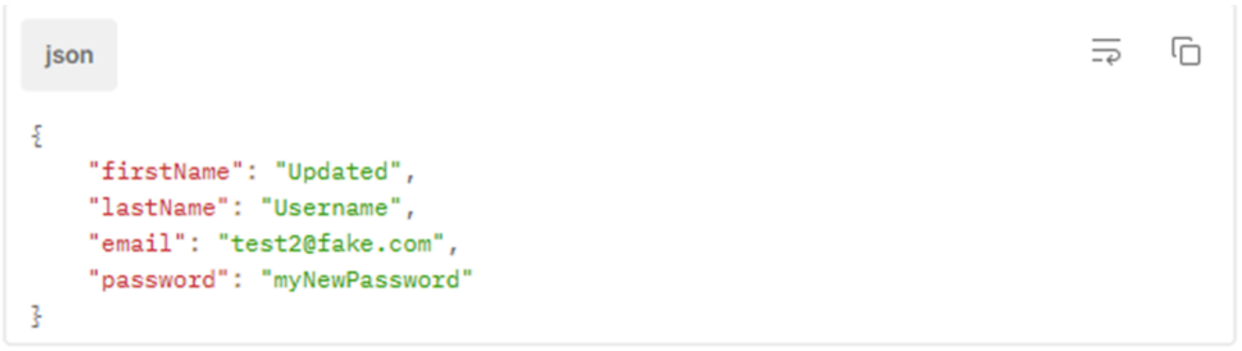
3. Get User Profile Request is not available through the UI.

4. Update user

Method: PATCH

URL : https://thinking-tester-contact-list.herokuapp.com/users/me

Request Payload:



```json
{
    "firstName": "Updated",
    "lastName": "Username",
    "email": "test2@fake.com",
    "password": "myNewPassword"
}
```

Purpose:

Updates a user information (First name, last name, email, password) through API.

UI Context:

The user cannot perform this operation through the UI and it must be done using an API tool like Postman or Swagger. Once executed, it will update the user's login credentials, and the user will no longer be able to log in using their old credentials.

UI & API Workflow:

Step1: open Postman and set the request URL to:

https://thinking-tester-contact-list.herokuapp.com/users/me

Choose the PATCH method.

Step 2: In the request payload (body), enter the new login credentials.

```
{
    "firstName": "Updated",
    "lastName": "Username",
    "email": "test2@fake.com",
    "password": "myNewPassword"
}
```

Step 3: In the request header, include the authorization token that was previously obtained.

Step 4: Click Send to submit the request.

Step 5: Verify that the response returns as 200 OK status code, which indicates that the update was successful.

Step 6 going back to the Contact List App main page, Log in using the newly updated credentials to confirm that the changes have taken effect.

5. Log Out User

Method: **POST**

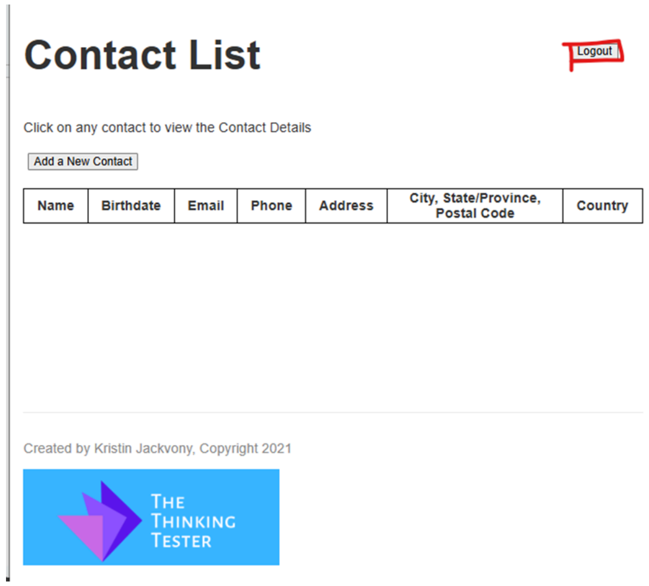
URL: https://thinking-tester-contact-list.herokuapp.com/users/logout

Purpose

Logs the user out from the application through the UI, Consequently expiring the authentication token and ending the current session.

UI Context

This action is triggered when the user clicks the 'logout ' button in the main page of the Contact List app, Once clicked, the user is logged out and the session is terminated.



Contact List

Click on any contact to view the Contact Details

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country

Created by Kristin Jackvony, Copyright 2021

UI & API Workflow:

Step 1: While logged into the Contact List App, locate the "Logout" button, usually found in the top navigation bar.

Step 2: Click on the Logout button.

Step 3: The application will invalidate the current session token, effectively ending the user session.

Step 4: The user is redirected to the login page, and must re-enter their credentials to log in again.

6. Delete user

Method : DELETE

URL : https://thinking-tester-contact-list.herokuapp.com/users/me

Purpose

Deletes the user account through API, The user will no longer be able to log in after this operation.

UI Context

The user cannot perform this operation through the UI so it must be done using an API tool like Postman or Swagger. Once executed, it will delete the user account, and the user will no longer be able to access the system using their credentials.

UI & API Workflow:

Step 1: Open Postman and set the request URL to:

https://thinking-tester-contact-list.herokuapp.com/users/me

Choose the DELETE method.

Step 2: In the request headers, include the authorization token that was previously obtained during login or registration.

Click Send to submit the delete request.

Step 3: Verify that the response returns a 200 OK status code, stating that the user account has been successfully deleted.

Step 4: Go back to the Contact List App login page.

The user will no longer be able to log in using the deleted credentials, as the account has been removed from the system.

B. Contacts

1. Add New Contact

Method: POST

Endpoint:https://thinking-tester-contact-list.herokuapp.com/contacts

Purpose

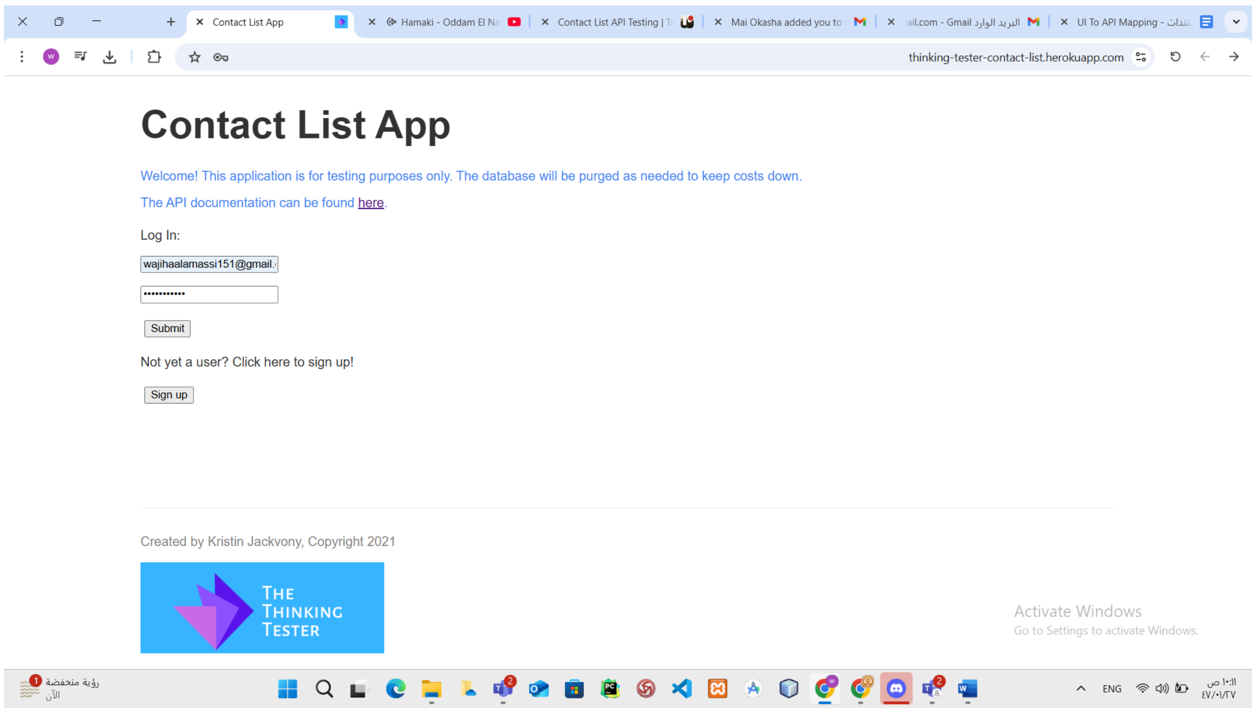
Adds a new contact for the authenticated user. All fields related to a contact are submitted via the request body.

UI Context

This action is triggered when the user clicks the 'Add New Contact' button in the Contact List dashboard and submits the form.

UI & API Workflow:

Step 1: User navigates to the Contact List dashboard after login.



Contact List App
thinking-tester-contact-list.herokuapp.com

Contact List App

Welcome! This application is for testing purposes only. The database will be purged as needed to keep costs down.

The API documentation can be found here.

Log In:

wajihaalamassi151@gmail.

•••••••••••

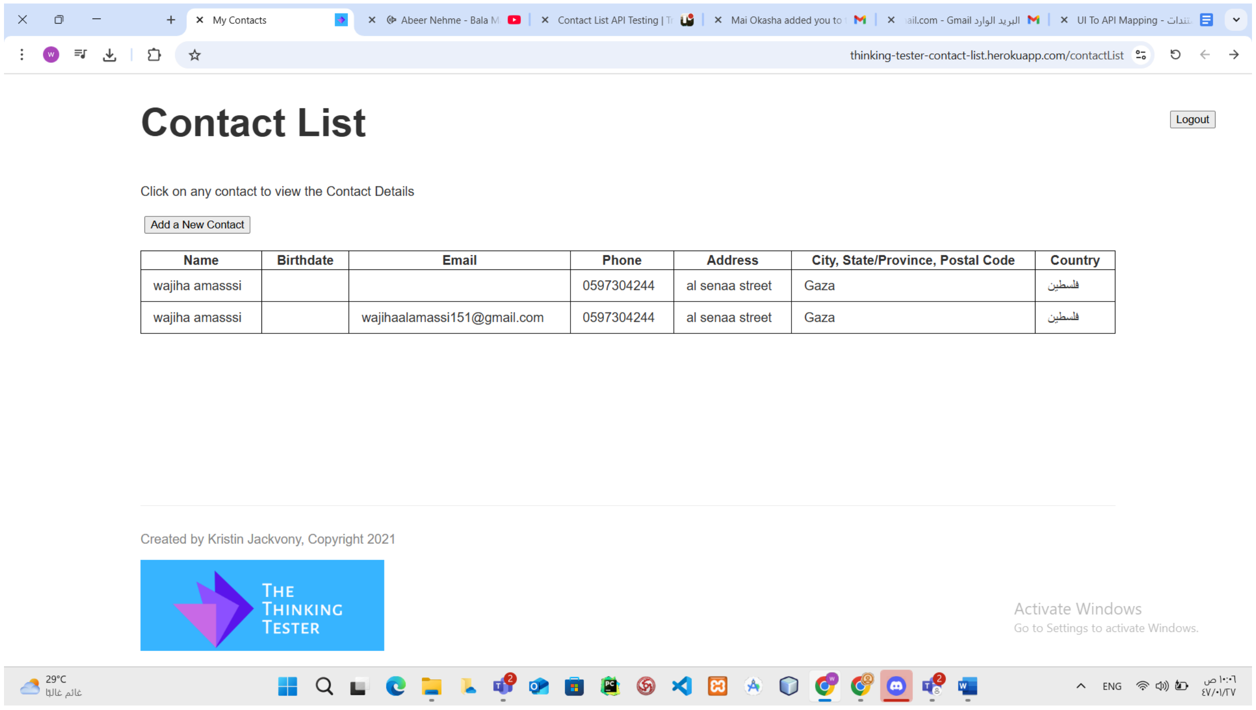
Submit

Not yet a user? Click here to sign up!

Sign up

Created by Kristin Jackvony, Copyright 2021

THE THINKING TESTER
Activate Windows
Go to Settings to activate Windows.



Contact List

Logout

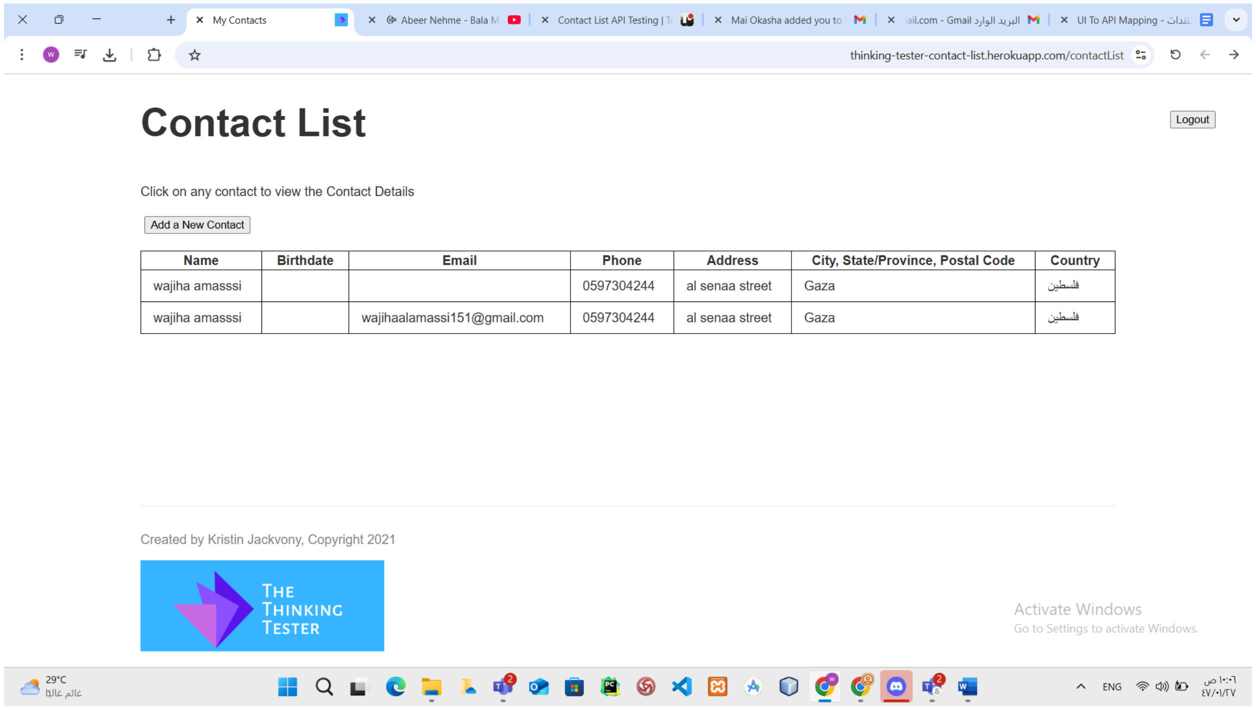
Click on any contact to view the Contact Details

Add a New Contact

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country
wajiha amasssi			0597304244	al senaa street	Gaza	فلسطين
wajiha amassi		wajihaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين

Created by Kristin Jackvony, Copyright 2021

Step 2: User clicks on the 'Add New Contact' button.



Contact List

Logout

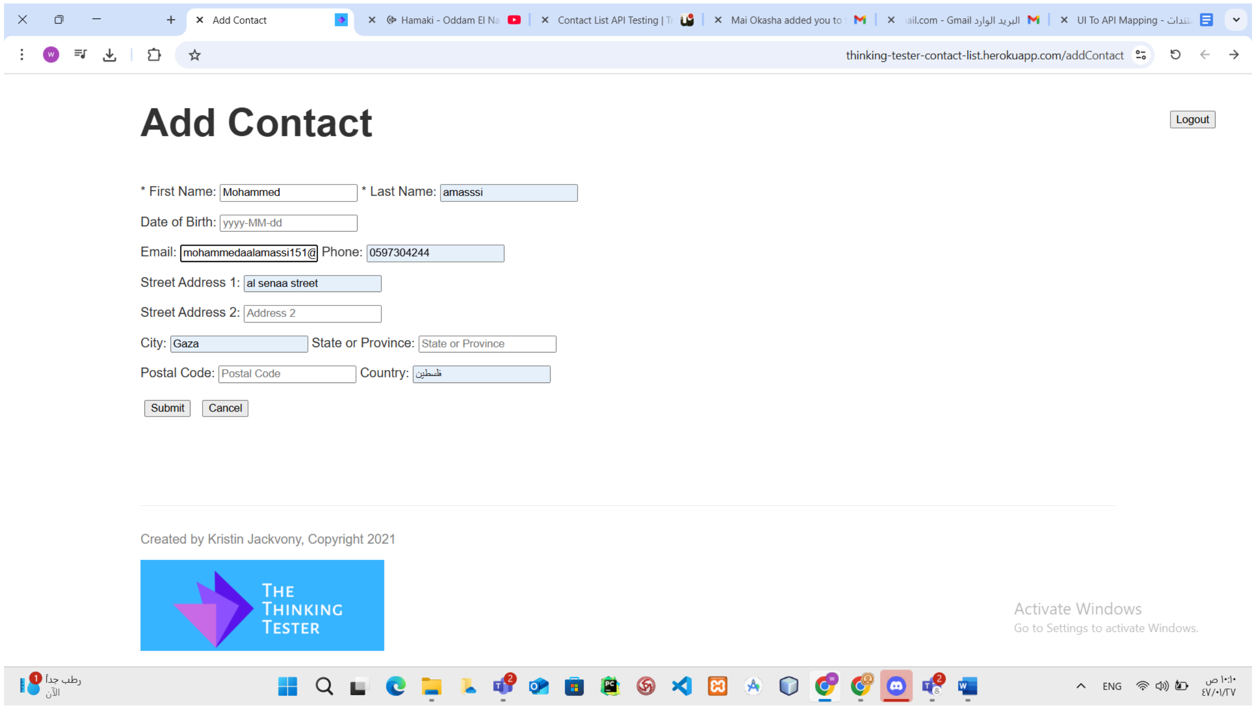
Click on any contact to view the Contact Details

Add a New Contact

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country
wajiha amassi			0597304244	al senaa street	Gaza	فلسطين
wajiha amassi		wajihaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين

Created by Kristin Jackvony, Copyright 2021

Step 3: UI displays a form to enter contact details like first name, last name, email, phone, address, etc.


Add Contact
Hamaki - Oddam El Na
Contact List API Testing | T
Mai Okasha added you to
ail.com - Gmail البريد الوارد
UI To API Mapping - تنفيذ
thinking-tester-contact-list.herokuapp.com/addContact
Logout
Activate Windows
Go to Settings to activate Windows.
ENG

Add Contact

* First Name: Mohammed * Last Name: amassi

Date of Birth: yyyy-MM-dd

Email: mohammedaalamassi151@ Phone: 0597304244

Street Address 1: al senaa street

Street Address 2: Address 2

City: Gaza State or Province: State or Province

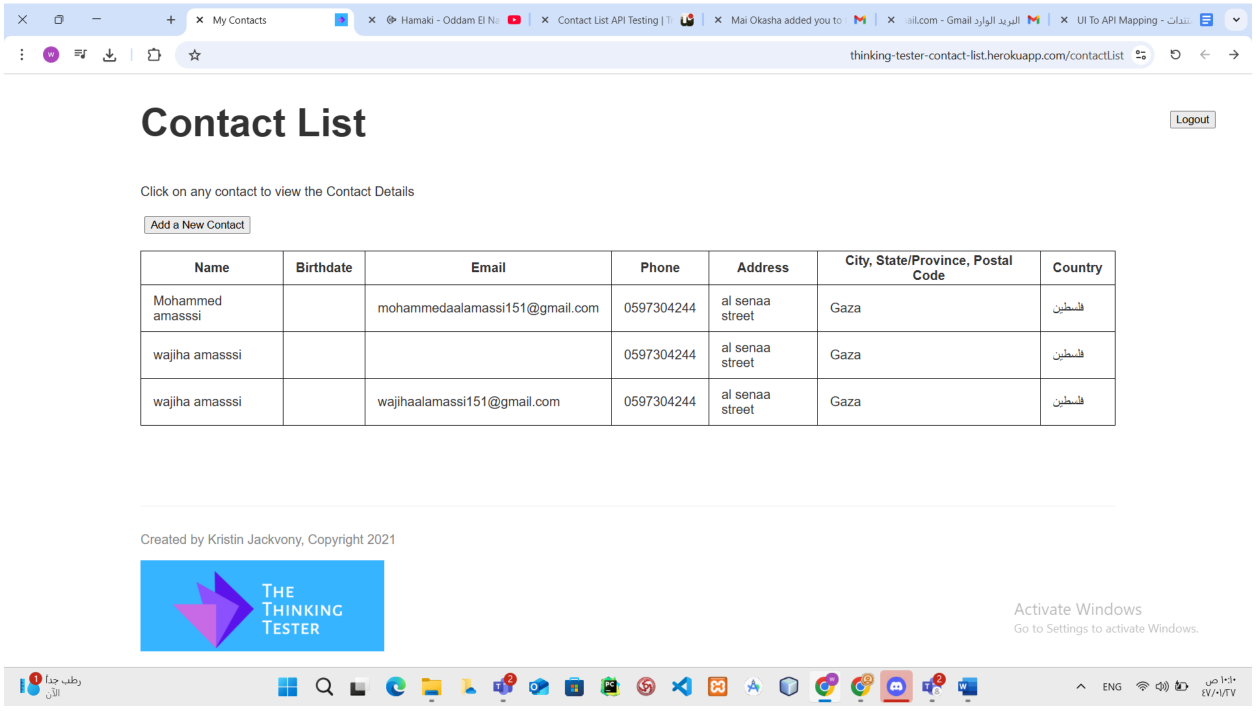
Postal Code: Postal Code Country: فلسطين

Submit Cancel

Created by Kristin Jackvony, Copyright 2021

THE THINKING TESTER

Step 4: User fills the form and clicks 'Submit'.


My Contacts
Hamaki - Oddam El Na
Contact List API Testing | T
Mai Okasha added you to
ail.com - Gmail البريد الوارد
UI To API Mapping - تنفيذ
thinking-tester-contact-list.herokuapp.com/contactList
Logout
Activate Windows
Go to Settings to activate Windows.
ENG

Contact List

Click on any contact to view the Contact Details

Add a New Contact

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country
Mohammed amassi		mohammedaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين
wajiha amassi			0597304244	al senaa street	Gaza	فلسطين
wajiha amassi		wajihaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين

Created by Kristin Jackvony, Copyright 2021

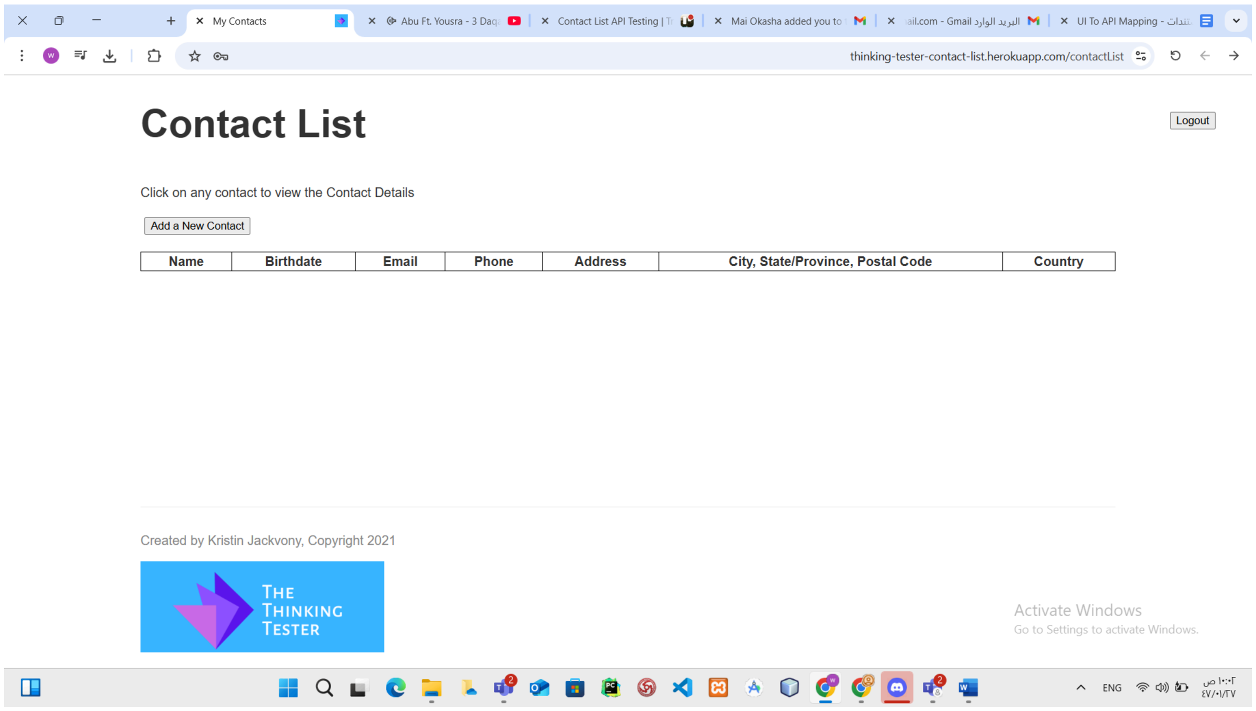
THE THINKING TESTER

Step 5: UI sends a POST request to the Add Contact API endpoint with all entered data in the request body and Authorization header with token.

Step 6: Server validates data and token.

Step 7: If data is valid, the server creates the contact and responds with 201 Created and contact info.

Step 8: UI redirects back to contact list with the new contact added.


My Contacts
Abu Ft. Yousra - 3 Daq
Contact List API Testing | T
Mai Okasha added you to
ail.com - Gmail البريد الوارد
UI To API Mapping - تنزيل
thinking-tester-contact-list.herokuapp.com/contactList
Logout
Add a New Contact
THE THINKING TESTER
Activate Windows
Go to Settings to activate Windows.
ENG

Contact List

Click on any contact to view the Contact Details

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country

Created by Kristin Jackvony, Copyright 2021

2. Get Contact List

Method: GET

Endpoint: https://thinking-tester-contact-list.herokuapp.com/contacts

Purpose

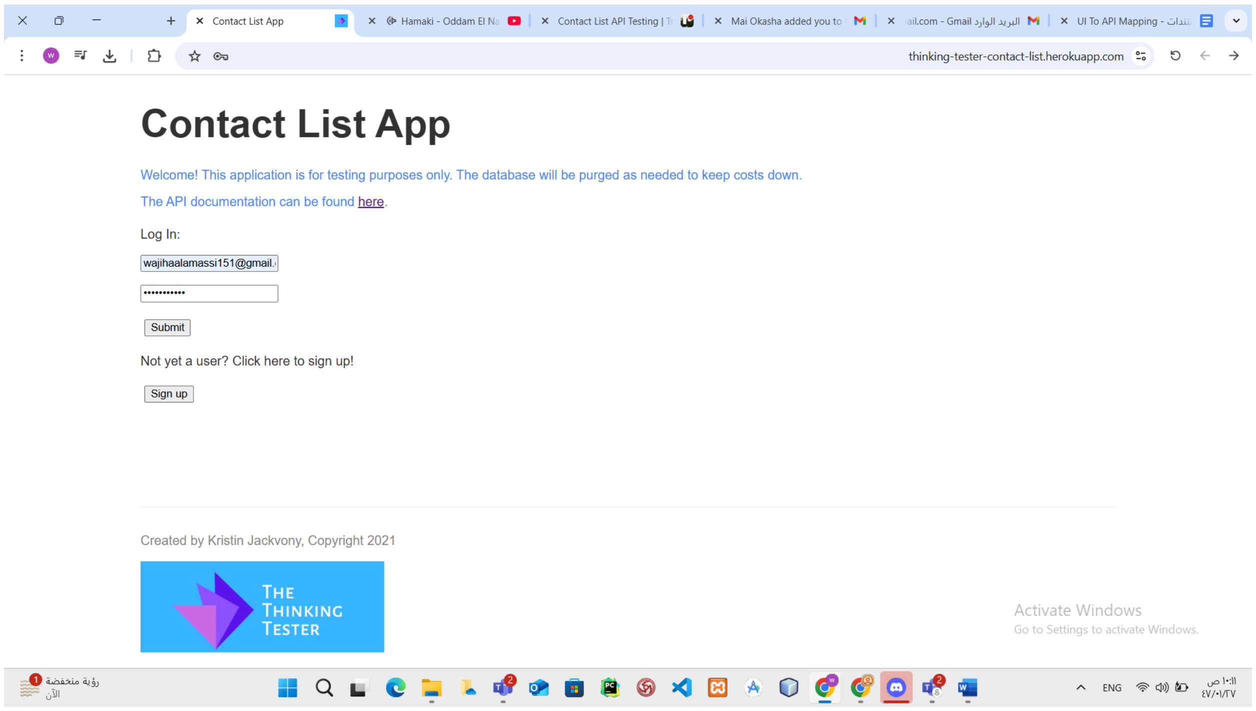
Retrieves the list of all contacts created by the logged-in user.

UI Context

This occurs when the Contact List screen is loaded or refreshed.

UI & API Workflow:

Step 1: User logs in or is redirected to the Contact List screen.



Contact List App

Welcome! This application is for testing purposes only. The database will be purged as needed to keep costs down.

The API documentation can be found here.

Log In:

wajihaalamassi151@gmail.

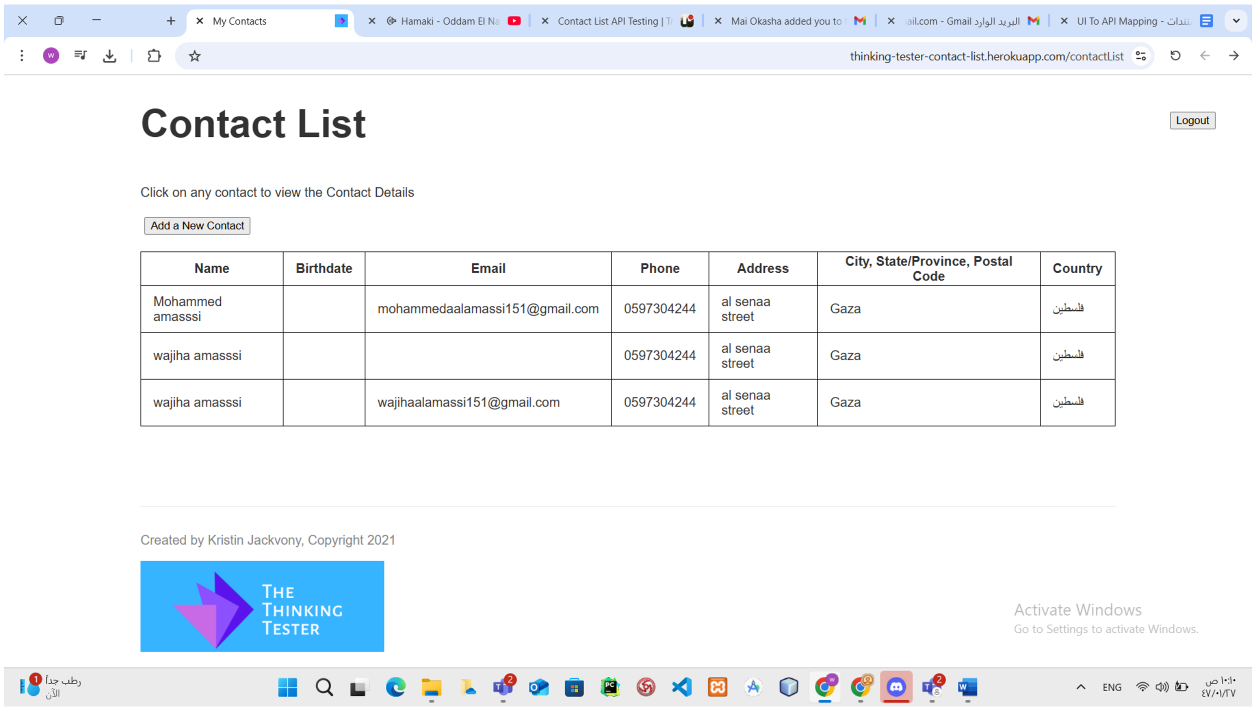
•••••••••••

Not yet a user? Click here to sign up!

Sign up
thinking-tester-contact-list.herokuapp.com

Created by Kristin Jackvony, Copyright 2021

The Thinking Tester
Activate Windows
Go to Settings to activate Windows.


My Contacts
Hamaki - Oddam El Na
Contact List API Testing | T
Mai Okasha added you to
ail.com - Gmail البريد الوارد
UI To API Mapping - تنزيلات
thinking-tester-contact-list.herokuapp.com/contactList
Logout
Add a New Contact
THE THINKING TESTER
Activate Windows
Go to Settings to activate Windows.
ENG

Contact List

Click on any contact to view the Contact Details

Name	Birthdate	Email	Phone	Address	City, State/Province, Postal Code	Country
Mohammed amasssi		mohammedaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين
wajiha amasssi			0597304244	al senaa street	Gaza	فلسطين
wajiha amasssi		wajihaalamassi151@gmail.com	0597304244	al senaa street	Gaza	فلسطين

Created by Kristin Jackvony, Copyright 2021

Step 2: Frontend sends a GET request to the contact list endpoint with a valid Authorization token.

Step 3: Server validates the token.

Step 4: If the token is valid, the server returns an array of contact objects (200 OK).

Step 5: UI parses and displays all contacts in a structured table or card format.

3. Get Single Contact

Method: GET

Endpoint:
https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Purpose

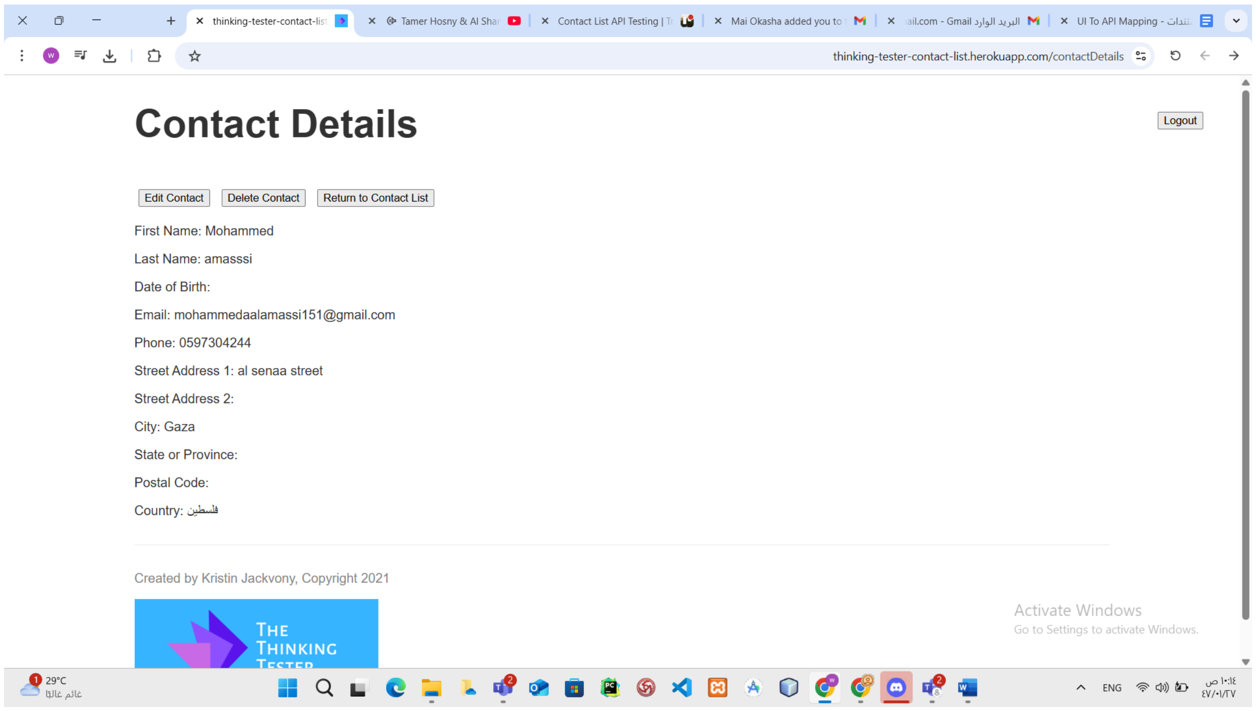
Fetches a specific contact's details using its unique ID.

UI Context

This action may occur when a user clicks 'Edit' or 'View Details' on a specific contact.

UI & API Workflow:

Step 1: User clicks on a contact card or Edit button.



Contact Details

First Name: Mohammed

Last Name: amasssi

Date of Birth:

Email: mohammedaalamassi151@gmail.com

Phone: 0597304244

Street Address 1: al senaa street

Street Address 2:

City: Gaza

State or Province:

Postal Code:

Country: فلسطين

Created by Kristin Jackvony, Copyright 2021

Step 2: UI sends a GET request to the API with the contact ID appended to the endpoint and includes the Authorization token.

Step 3: Server verifies the token and contact ID.

Step 4: If valid, it returns contact data (200 OK); otherwise, 404 or 401.

Step 5: UI uses the data to populate the Edit form or display it to the user.

4. Update Contact (Full Update – PUT)

Method: PUT

Endpoint:
https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Purpose:

Replaces all fields of an existing contact with new values provided in the request body. It overwrites the entire contact resource.

UI Context:

Triggered when a user selects a contact from the contact list, clicks the **Edit** button, updates the information in all fields of the Edit Contact form, and clicks **Submit**.

UI & API Workflow – Successful Full Update

Step 1:
The user clicks on the **Edit** button on a contact card.

Step 2:
UI sends a **GET** request to fetch the contact's current data using the contact's ID and populates the edit form.

Step 3:
The user edits the full contact details (first name, last name, phone, address, etc.).

Step 4:
User clicks **Submit**, and the UI sends a **PUT** request to:

https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Payload example:

```
{
  "firstName": "Hazem",
  "lastName": "Test",
  "email": "hazem@example.com",
  "phone": "1234567890",
  "address": "123 Street"
}
```

Headers:

Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdlNmRkNzdiYzU2NjAwMTVlM2I2ZmEiLCJpYXQiOjE3NTMxMTg3MDF9.26QLS5O9aaf47b6CEcABrWTuwW-Gi-zlUgj9C3bra54

Step 5:
Server validates the token and payload.

Step 6:
If valid, server replaces the contact record and returns:

- **Status:** 200 OK
- **Body:** Updated contact data.

Step 7:
UI redirects back to the contact list and displays the updated contact.

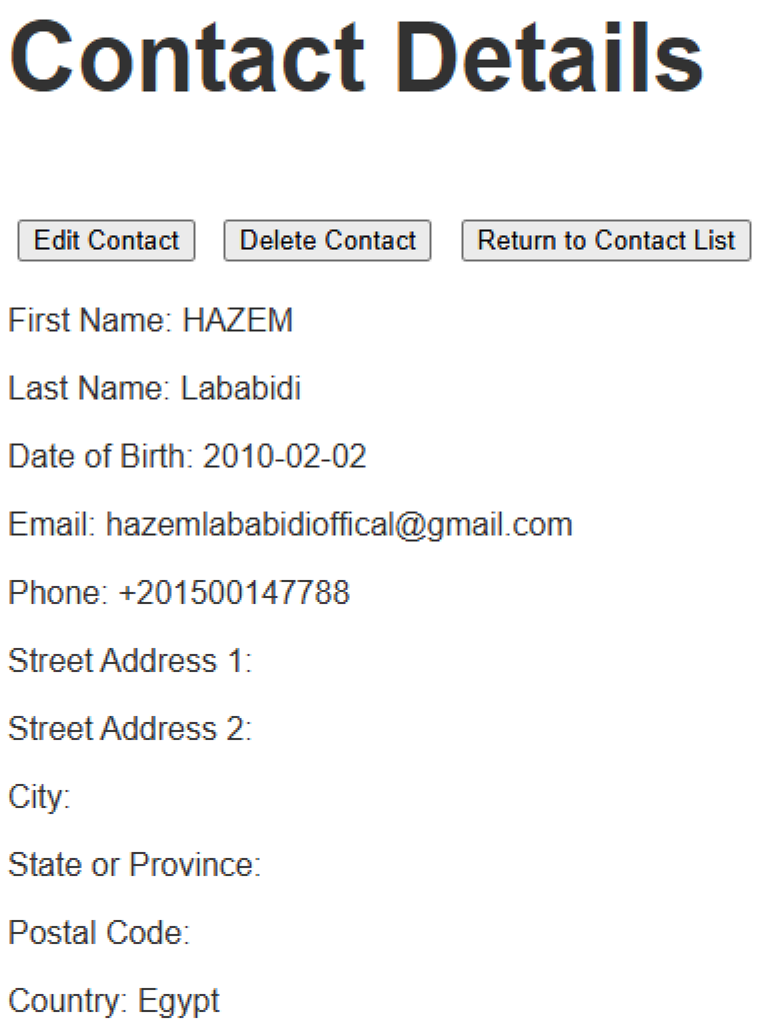
UI & API Workflow – Failed Update (Invalid Fields)

- If a required field is missing or in the wrong format, the server responds with:
    - **Status:** 400 Bad Request

Body:

```
{
 "message": Email is invalid"
}
```

- UI displays a relevant error message and keeps the user on the edit screen.



Contact Details

First Name: HAZEM

Last Name: Lababidi

Date of Birth: 2010-02-02

Email: hazemlababidioffical@gmail.com

Phone: +201500147788

Street Address 1:

Street Address 2:

City:

State or Province:

Postal Code:

Country: Egypt

5. Partial Update Contact (PATCH)

Method: PATCH

Endpoint:
https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Purpose:

Updates only specific fields of an existing contact without modifying the entire resource.

UI Context:

This can be triggered when the UI allows editing only one or a few fields (e.g., updating just the phone number or email) and then submits.

UI & API Workflow – Successful Partial Update

Step 1:
User clicks **Edit** and modifies only selected fields (e.g., just the phone number).

Step 2:
UI sends a **PATCH** request to:

https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Payload example:

```
{
  "phone": "9988776655"
}
```

Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdlNmRkNzdiYzU2NjAwMTVlM2I2ZmEiLCJpYXQiOjE3NTMxMTg3MDF9.26QLS5O9aaf47b6CEcABrWTuwW-Gi-zlUgj9C3bra54

Step 3:
Server validates the token and the fields provided.

Step 4:
If successful, server returns:

- **Status:** 200 OK
- **Body:** Updated contact data with the modified field(s).

Step 5:
UI refreshes the contact list or displays the updated details.

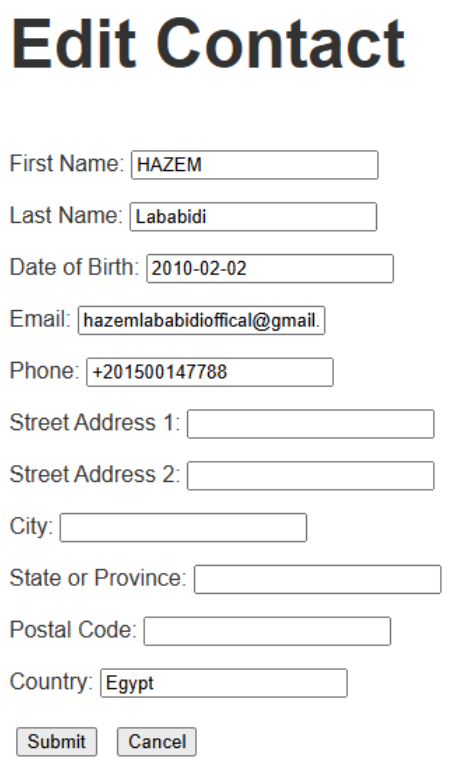
UI & API Workflow – Failed Partial Update

- If an invalid field is submitted (e.g., wrong email format):
  - **Status:** 400 Bad Request

Message:

```
 {
  "message": "Email is invalid"
}
```

- UI highlights the error field and shows a validation message.



Edit Contact

First Name:	HAZEM
Last Name:	Lababidi
Date of Birth:	2010-02-02
Email:	hazemlababidioffical@gmail.
Phone:	+201500147788
Street Address 1:	
Street Address 2:	
City:	
State or Province:	
Postal Code:	
Country:	Egypt

6. Delete Contact

Method: DELETE

Endpoint:
https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Purpose:

Deletes an existing contact using its unique ID.

UI Context:

This occurs when a user clicks the **Delete** icon or button on a specific contact in the contact list and confirms the deletion.

UI & API Workflow – Successful Delete

Step 1:
User clicks the **Delete** icon on a contact card & Show popup to confirm.

Step 2:
User confirms, and the frontend sends a **DELETE** request to:

https://thinking-tester-contact-list.herokuapp.com/contacts/{contactId}

Headers:

Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdlNmRkNzdiYzU2NjAwMTVlM2I2ZmE
iLCJpYXQiOjE3NTMxMTg3MDF9.26QLS5O9aaf47b6CEcABrWTuwW-Gi-zlUgj9C3bra54

Step 4:
Server validates the token and contact ID.

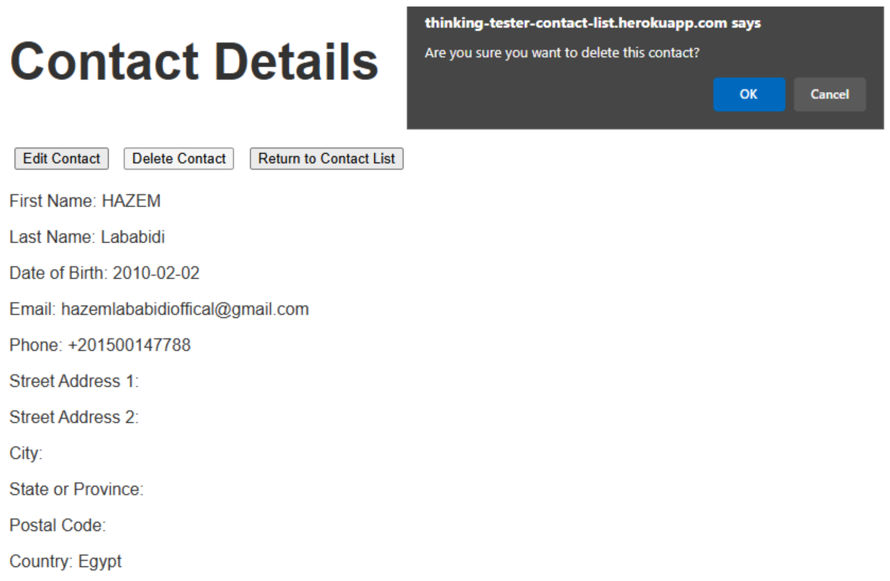
Step 5:
If successful:

- **Status:** 200 OK or 204 No Content
- No body content.

Step 6:
UI removes the contact from the list and shows a success toast/snackbar.

6.1 UI & API Workflow – Failed Delete

- If the contact does not exist or token is missing/invalid:
  - **Status:** 401 Unauthorized or 404 Not Found
    UI Message: *"Contact not found"* or *"Session expired"*


Contact Details

thinking-tester-contact-list.herokuapp.com says

Are you sure you want to delete this contact?

First Name: HAZEM

Last Name: Lababidi

Date of Birth: 2010-02-02

Email: hazemlababidiofficial@gmail.com

Phone: +201500147788

Street Address 1:

Street Address 2:

City:

State or Province:

Postal Code:

Country: Egypt