1.Introduction/Overview: To test the functionality of the website https://thinking-tester-contact-list.herokuapp.com and verify the functionality.

2.Test Objectives:

• Test functionality of website functions.

• Verify that users can login successfully.

• Ensure there are no spelling errors once logged in to the website.

• Ensure there are no visual defects on the website.

• Verifying all categories listed on the web page works correctly.

- Test adding new contacts.
- Ensure that contact details can be modified.
- Confirm that contacts can be removed.
- Check if sorting and filter inputs work as expected.

3. Test Strategy:

• Test user login, registrations, and password reset functionality.

• Test adding, editing, and deleting contacts.

• Evaluate the usability of the website once logged in (ease of use and delays in performance).

• Test the website on various internet browsers and mobile applications to verify the website compatibility.

• Record test results and any defects.

4. Test Deliverables:

• Functional testing: verifying if user cannot log in with a valid username and invalid password.

• Integration testing: Check interactions between components.

• Usability Testing: Evaluate user experience.

• Security Testing: Identify any vulnerabilities.

• Performance testing: Review response times.

4. Test Schedule:

- Day 1: Gather requirements to define the test objectives. Create a test plan including test plan and resources needed.
- • Day 2: Verify the website link is working. Access to search functionalities. Verify different queries produce results and document any errors. Compatibility testing ensures website and functionalities work across different devices and search engines. Document any errors. Performance testing to test if there are any delays with user load.
- Day 3: Have a conversation with the developers and/or stakeholders to go over any issues encountered while testing. Integration testing to ensure that errors that were corrected did not cause any other defects.
- Day 4: Retest website and make updates as needed.

Test Creation:

1.Objective: To verify the API successfully adds a new contact to the contact list.

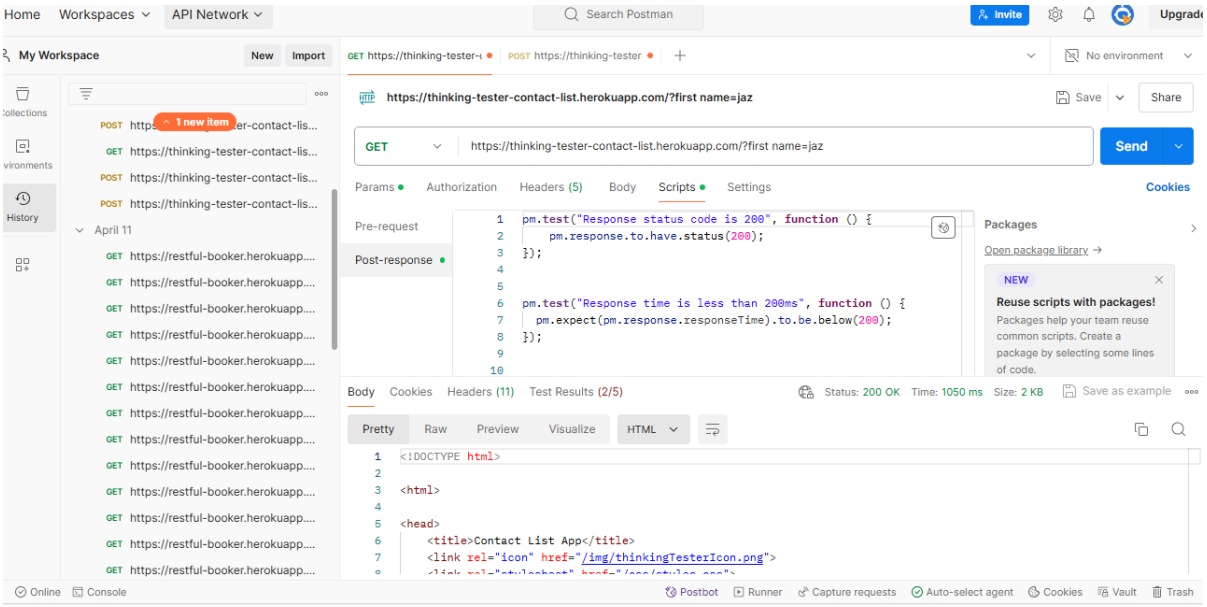
2. Title: Add Contact API Test Case

3. Description: This test case validates the functionality of the Add Contact API endpoint.

4. Test Data: Details for new contact. Name, address, email, phone number.

5. Test Steps:

1. Send a POST request to the API endpoint for adding a contact.
2. Add website https://thinking-tester-contact-list.herokuapp.com/
3. Added name Jaz
4. Verify the response status code is 200.
5. Verify that response contains newly created contact details.

6. 

2. Objective: To verify the API successfully gets the contact list.

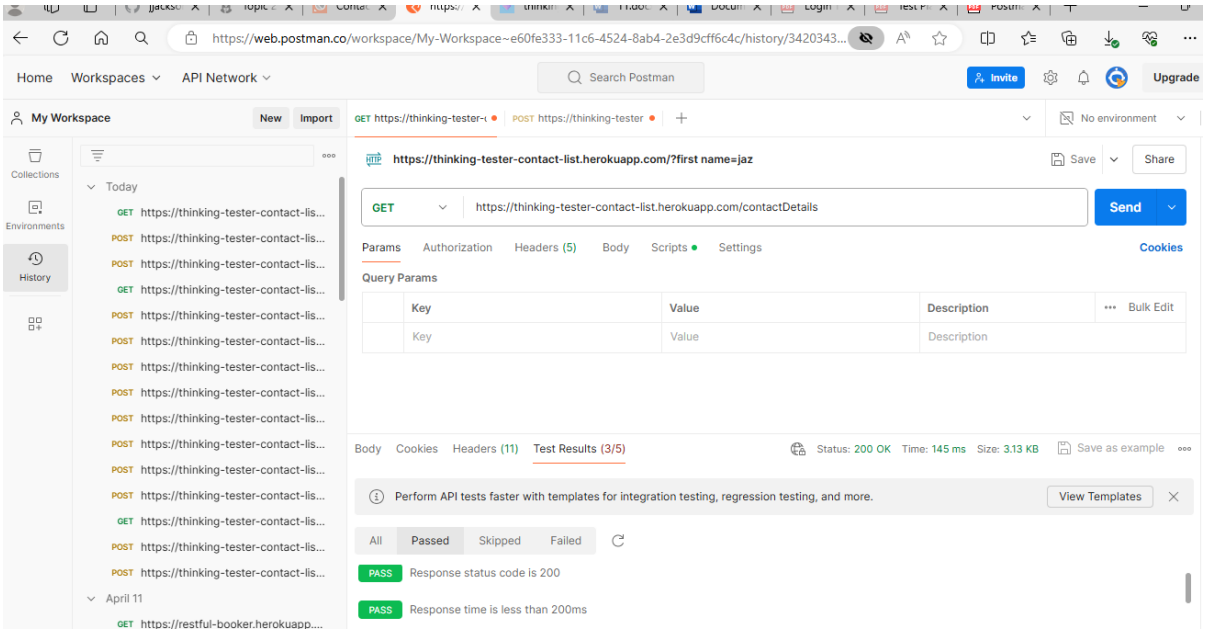
1. Title: Get Contact List API Test Case

2. Description: This test case validates the functionality of the Add Contact API endpoint.

3. Test Data: Details for new contact. Name, address, email, phone number.

4. Test Steps:

    1. Send a GET request to the API endpoint for Checking contact list.
    2. Add website https://thinking-tester-contact-list.herokuapp.com/contacts
    3. Verify the response status code is 200.
    4. Verify that response contains newly created contact details.

5. 

3. Objective: Verify the API successfully retrieves the list of contacts.

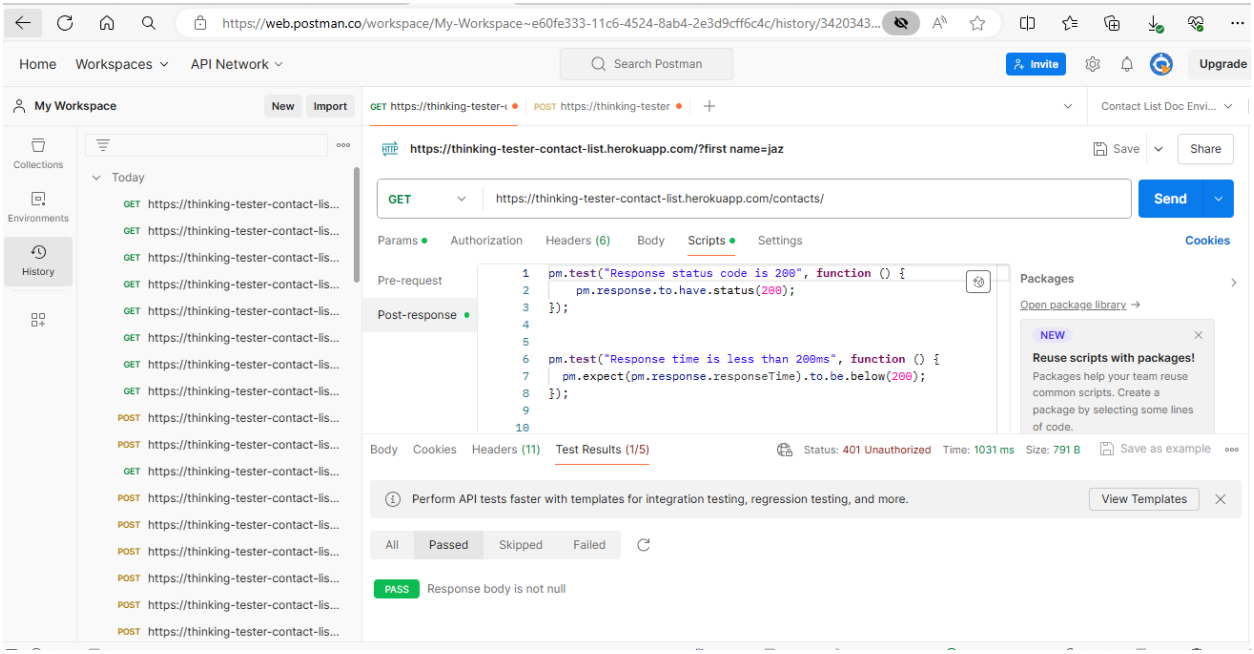
Title: Retrieve Contact List

Description: The contact list should be accessible.

Test Steps:

1. Send a GET request to the API endpoint [My Contacts (thinking-tester-contact-list.herokuapp.com).](thinking-tester-contact-list.herokuapp.com)
2. Verify the response status code is 200.

Expected results: The API returns a list of contacts



4. Objective: To update contact information

Title: Update Contact

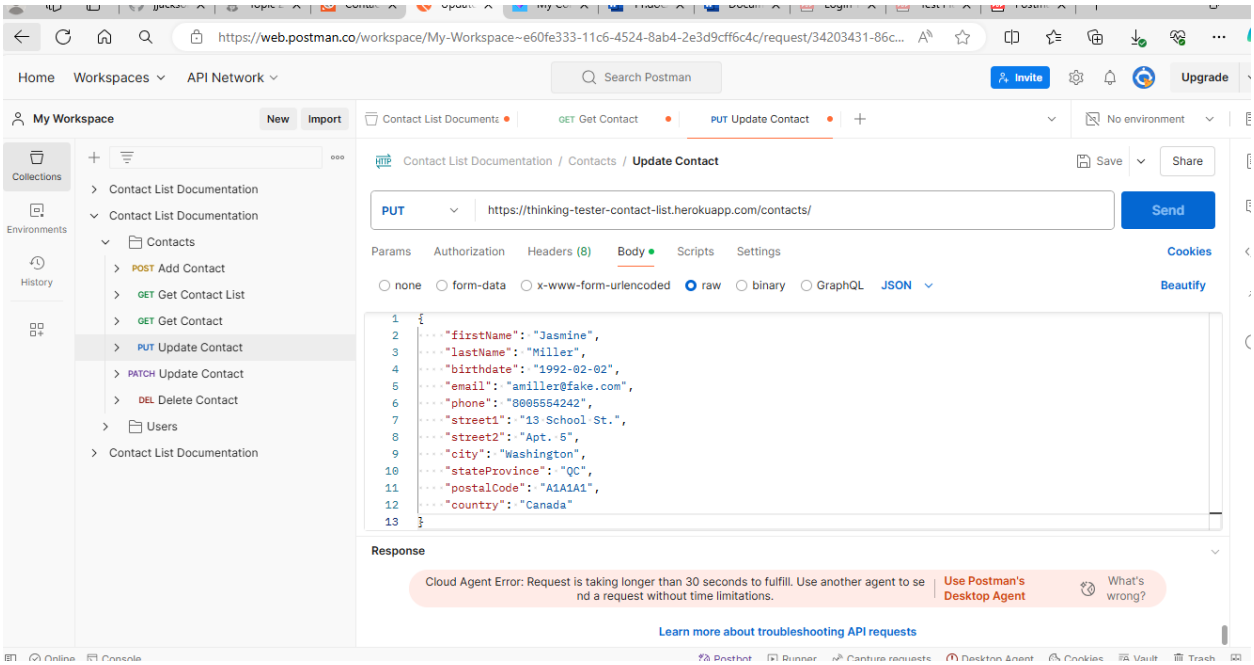
Description: Should be able to update contact information.

Test Steps:

1.Send a PUT request to the API endpoint https://thinking-tester-contact-list.herokuapp.com/contacts/.

Verify the response status code is 200.

Expected results: To update contact information with no errors. Actual result received error stating Cloud Agent Error: Request is taking longer than 30 seconds to fulfill. Use another agent to send a request without time limitations.



5. Objective: To verify the API successfully updates contact information.

Title: PATCH Update Contact

Description: This test case validates the functionality of the update Contact API endpoint.

Test Steps:

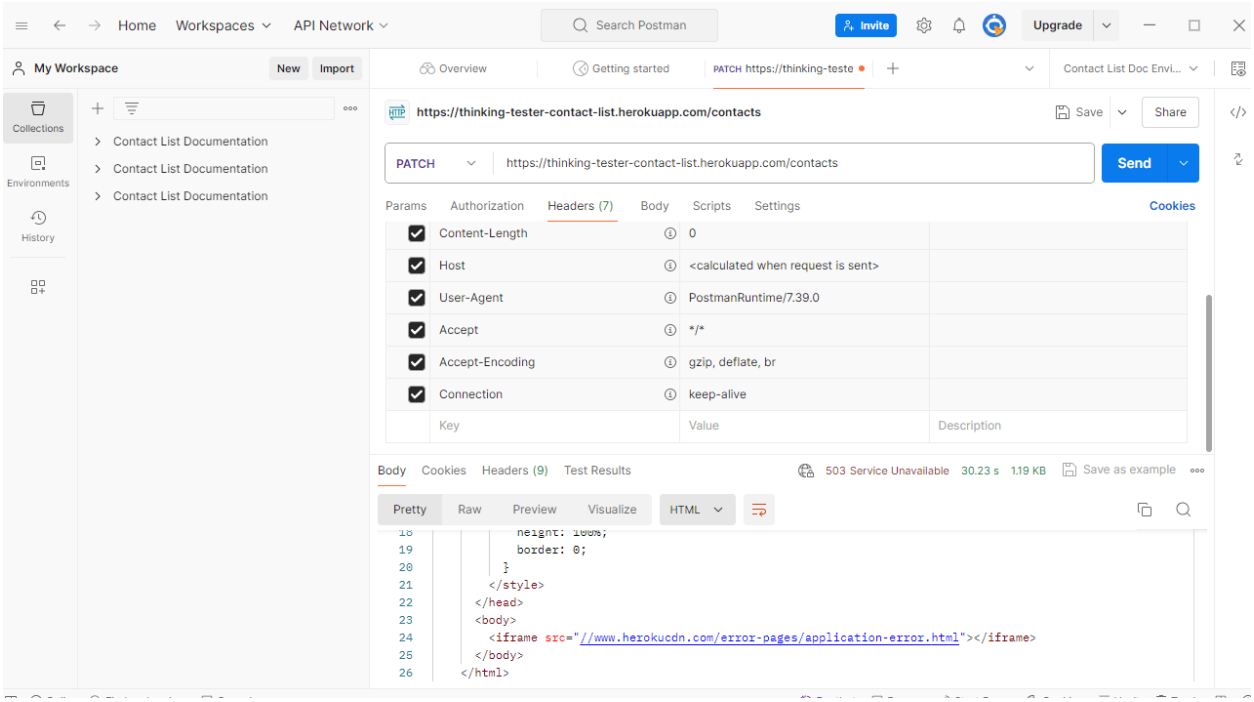
1. Send a PATCH request to the updated contact.

2. Add website https://thinking-tester-contact-list.herokuapp.com/contacts

3. Verify the response status code is 200.

4.Verify that response contains newly created contact details.

Actual results: received error code 503Service Unavailable



6. Objective: To delete contact

Title: DELETE Contact

Description: This test case validates the functionality of deleting a contact API endpoint.

Test Steps:

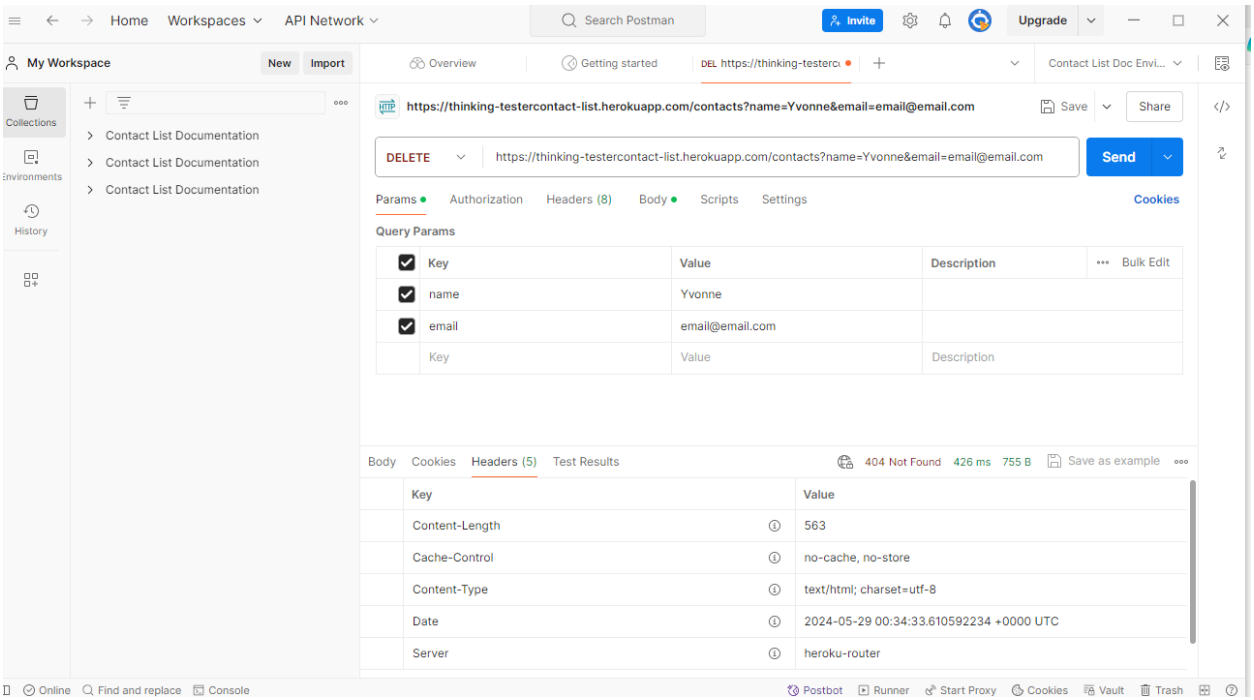
1. Send a DELETE request to the updated contact.

2. Add website https://thinking-tester-contact-list.herokuapp.com/contacts

3. Verify the response status code is 200.

4.Verify that response contains newly created contact details.

Actual results: Received error code 404 not found.



7. Objective: Post add user

Title: Add user

Description: This test case adds a user.

Test Steps:

1. Send a POST request to add a user.

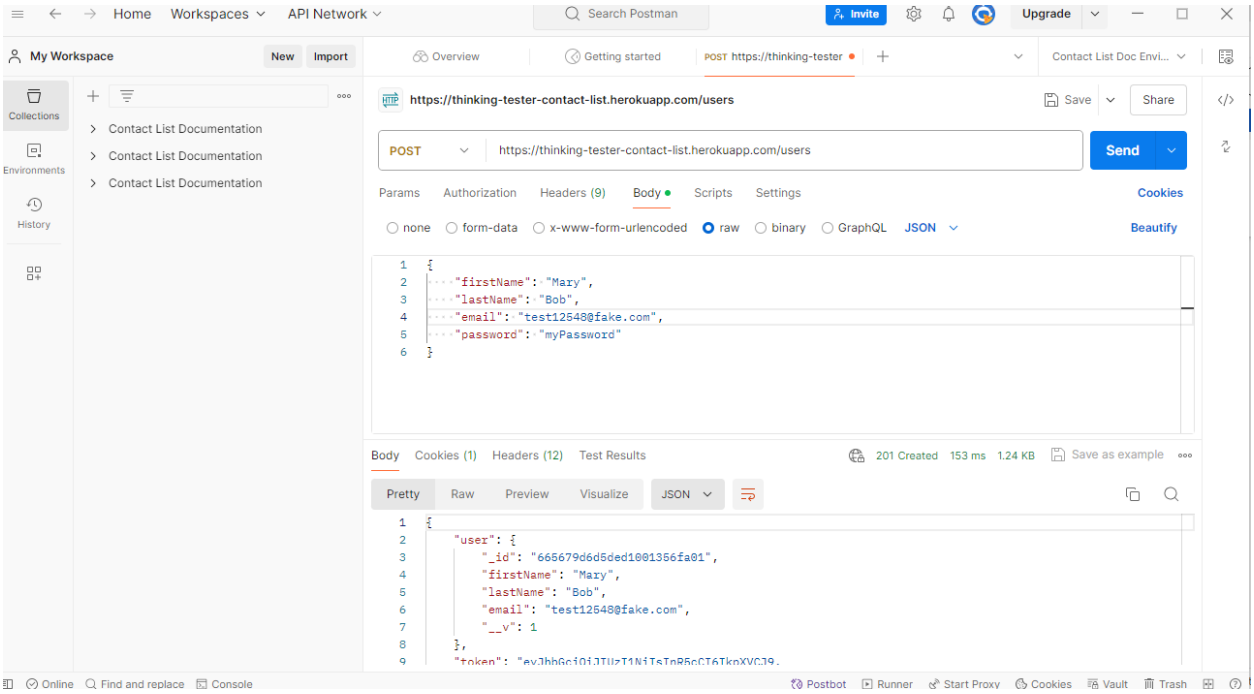
2. Add website https://thinking-tester-contact-list.herokuapp.com/contacts

3. Add username, password, and email.

3. Verify the response status code is 200.

4.Verify that response contains newly created contact details.

Actual results: Created a new user.



8. Objective: To get user profile.

Title: GET User

Description: This test case should show the user profile.

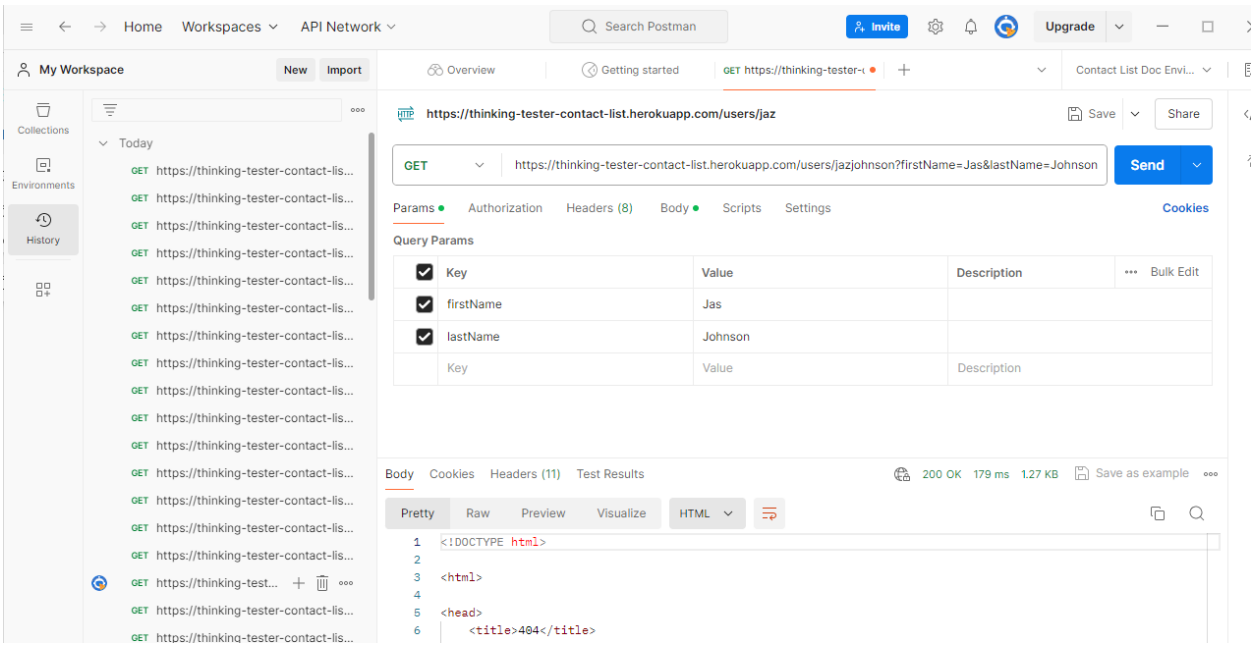
Test Steps:

1. Send a GET request to get a user.

2. Add website https://thinking-tester-contact-list.herokuapp.com/user/jasjohnson

3. Add username and password.

3. Verify the response status code is 200.



9. Objective: To update user profile.

Title: PATCH User Update

Description: This test case should be able to update user profile.

Test Steps:

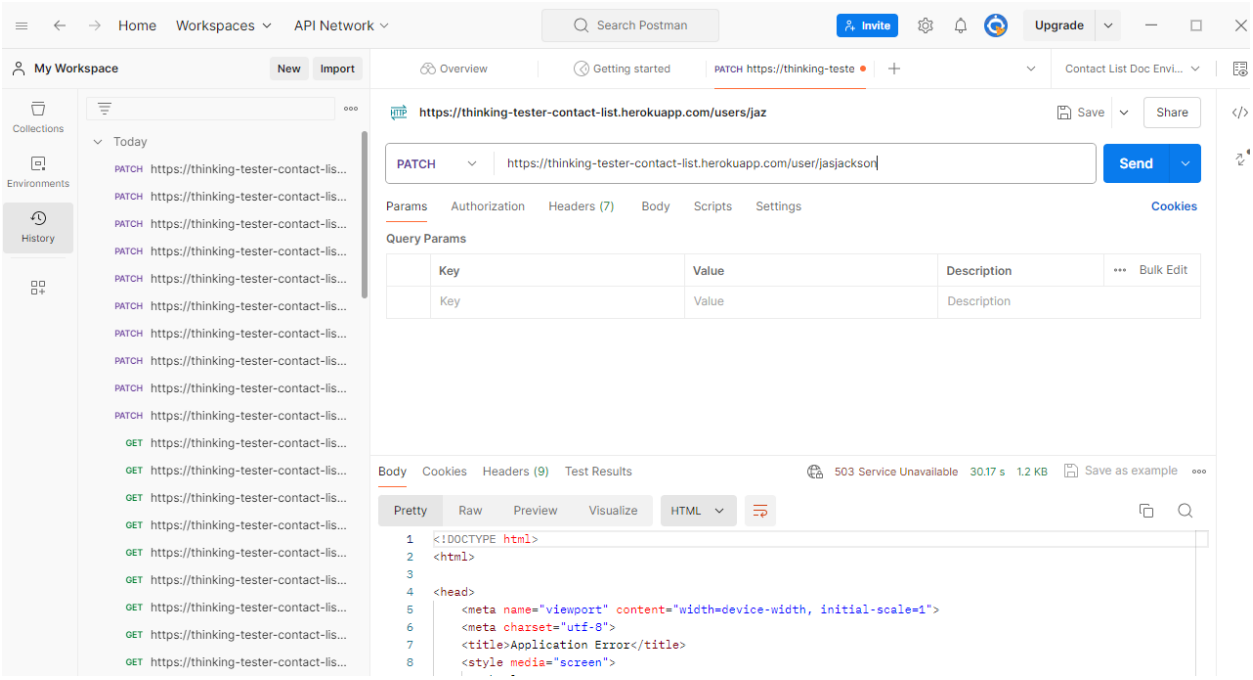
1. Send a PATCH request to get an update user profile.

2. Add website https://thinking-tester-contact-list.herokuapp.com/user/jasjohnson

3. Add username and password.

4. Verify the response status code is 200.

Actual Response: Received error code 503 service unavailable.



10. Objective: To Logout User

Title: Logout user

Description: This test case is to logout as a user.

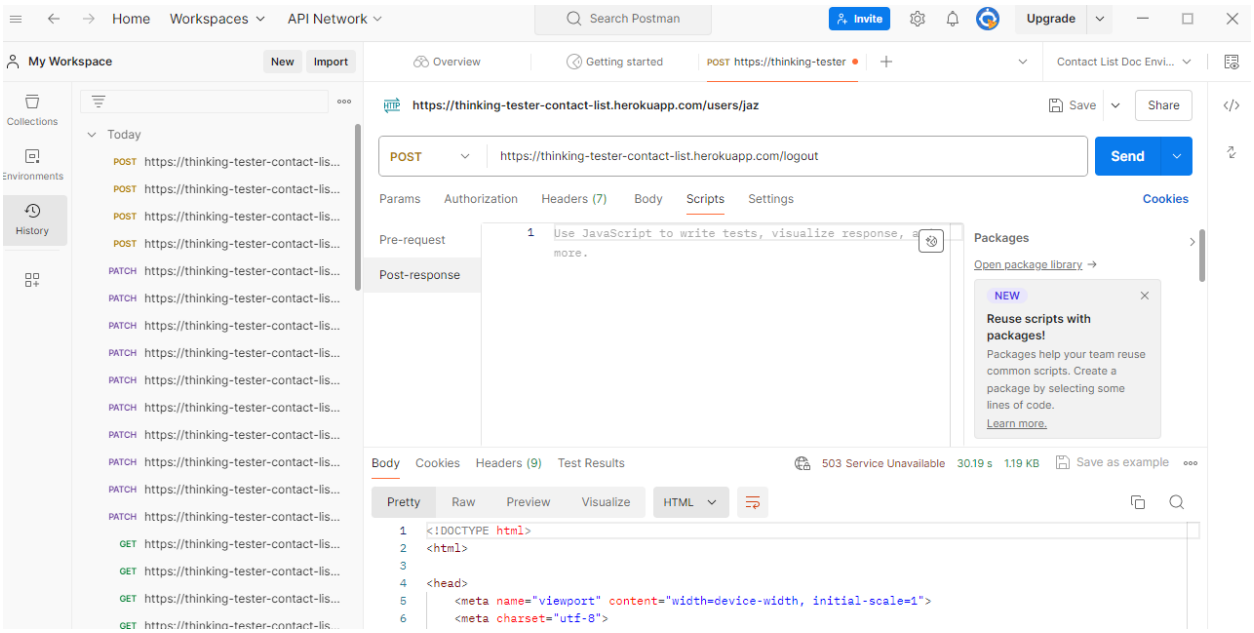
Test Steps:

1. Send a POST request to add a user.

2. Add website https://thinking-tester-contact-list.herokuapp.com/logout

3. Verify the response status code is 200.

Actual results: Received error code 503.



11. Objective: To Login User

Title: Login user

Description: This test case is to login a user.

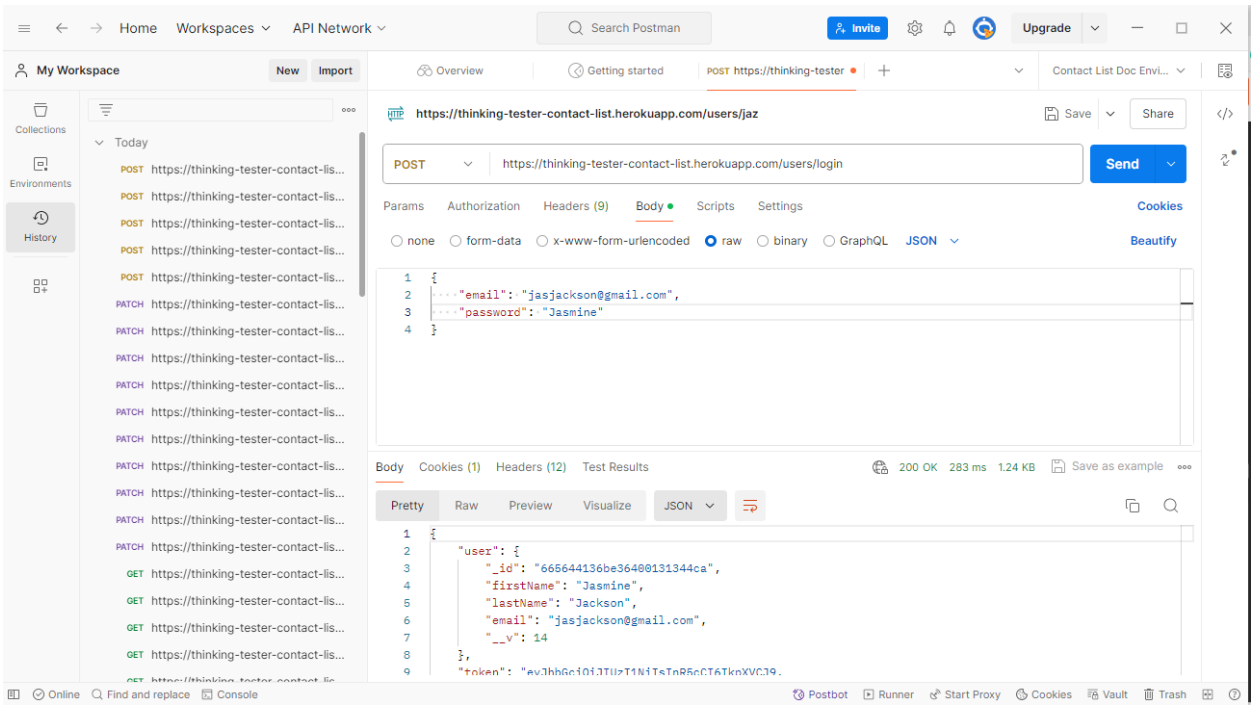
Test Steps:

1. Send a POST request to add a user.

2. Add website https://thinking-tester-contact-list.herokuapp.com/users/login

3.. Add username and password to the body.

4. Verify the response status code is 200.



12. Objective: To Delete User

Title: Delete user

Description: This test case is to delete a user.

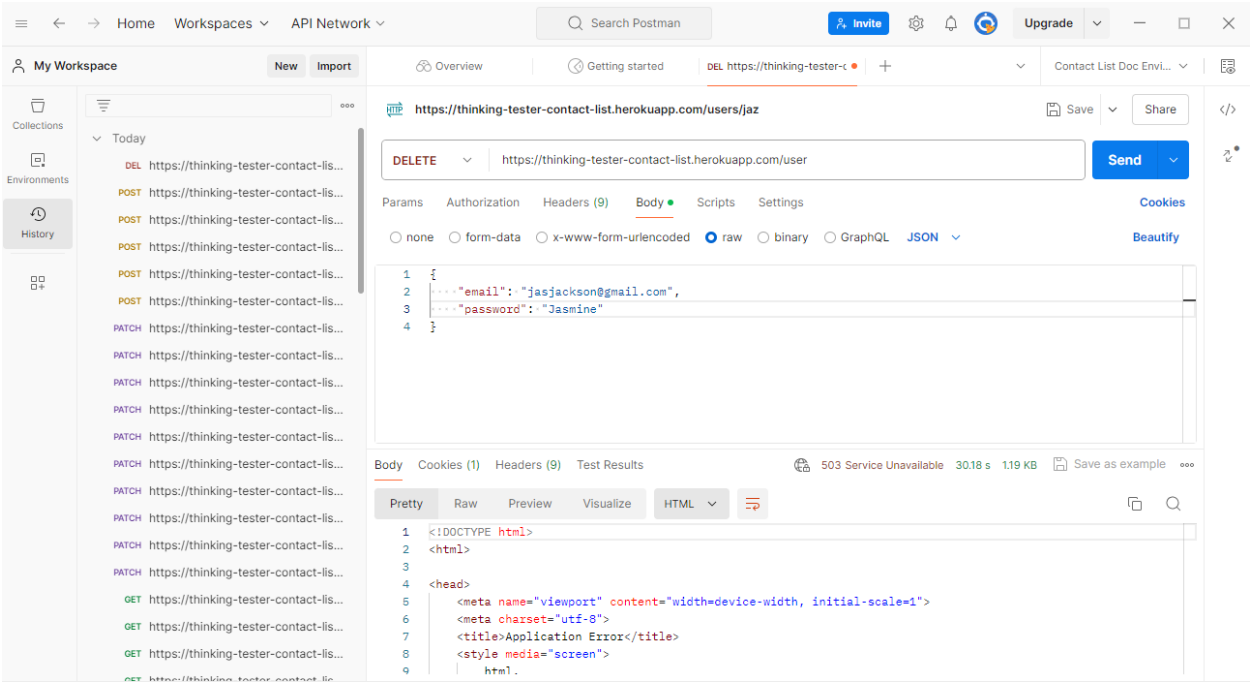
Test Steps:

1. Send a DELETE request to delete a user.

2. Add website https://thinking-tester-contact-list.herokuapp.com/user

3. Verify the response status code is 200.

Actual results: Received error code 503.



Non-Functional Tests

1.Response Time Test Case:

Objective: To evaluate the response time of the website.

Test Steps: Once logged into the website confirm the response time when creating user profiles.

Expected Results: The website should not have any lagging while using the website.

Observed Results: The system responds as expected and did not have any delays.

2.Availability Test Case:

Objective: Observe website for accessibility during business hours.

Test Steps: Login and add users during business hours.

Expected Results: The website should be accessible without any downtime or updates needed.

Observed Results: The website was accessible and had no downtime.

3.Usability Test Case:

Objective: To assess the user friendliness of the website.

Test Steps: Evaluation of the website. Including menu and buttons.

Check readability and assess any error messages.

Expected Results: The website should be easy to use.

Observed Results: Website's inputs are easy to spot and input fields have a good font size for easy readability.

4.Compatability Test Case:

Objective: Ensure website works across different browsers.

Test Steps: Test website on Chrome, Internet explorer and Edge.

Verify website responsiveness on each browser.

Expected Results: Website should function the same across all internet browsers.

Observed Results: Website was consistent across all internet browsers.

Security Test Cases

1. Authentication and Authorization Testing:

Objective: To verify only authenticated users can access the website.

Test steps: Sign up to have access to a website.

Login with valid credentials.

Confirm if not logged in you should receive an error message.

2. Input Validation Testing:
   Objective: Validate input fields to prevent common security vulnerabilities.
   Test Steps: Submit forms with malicious input.
   Ensure the application sanitizes input and rejects harmful characters.
   Verify the website displays an error message for invalid input.
3. Sensitive Data Exposure Testing:
   Objective: Check to ensure passwords are properly encrypted and not exposed.
   Test Steps: Create a login.
   Confirm that the sensitive data is transmitted securely and not visible in plain text.
4. Session Management Testing:
   Objective: Validate session time outs.
   Test Steps: Login with valid credentials and verify the session duration due to inactivity.

Attempt to perform actions after session has timed out.
Verify the website prompts for reauthentication if the session has timed out.

Jira Board