### **Documentation for Contact List Manager App**

### **1. Introduction**

The Contact List Manager app is a web application designed to manage contact information. It includes an API with a GUI built on top of it. This document outlines the test plan, test cases, issues found during testing.

### **2. Questions and Answers**

### Q1. What additional documentation or information would have made planning the tests for this app more efficient?

* What would be the response and error message if we hit the api with invalid data, how we are handling this.
* As the components of the application are already clear but for those components which are not implemented yet what would be their flow, design and layout on application?(Get User Profile / Update User Profile / Delete User)
* Regarding performance testing, how many numbers of users could use the application at once? Checking stress points
* What is the priority for the components which are necessary but not implemented yet?

Forgot Password, Session logout, Captcha.

* Are we going to make the application responsive for cross browser and cross devices?
* What would be the GUI for thousands of records(contacts)? Pagination / scroll down and loading contacts
* GUI Mockups or Wireframes for visual representations of the application workflow.
* Would we have multiple user roles or only one (Admin, operator, user)
* Detailed user stories or requirements documentation.
* End-to-end user flow scenarios.
* Data validation rules and field constraints.
* Error message guidelines and handling mechanisms.

### Q2. Identify the components of the app that can be tested. Is there a hierarchy of importance in these components?

**Components:**

* 1. User Authentication (Login/Signup)
  2. Contact Management (Add, Update, Delete, View Contacts)
  3. User Profile Management (Get User Profile / Update User Profile / Delete User)
  4. API Endpoints
  5. GUI Functionality and Usability

Yes, the hierarchy is important for these components.

* User Authentication
* Contact Management as its core functionality of the app.
* Contact List Management important for personalization.
* Session Management
* Error Handling
* Security for protecting user data.
* UI/UX for better user experience.

### Q3. For each identified area, how would you approach testing?

* User Authentication:
  + Test various login and signup scenarios including valid and invalid inputs.
  + Test session management, login persistence, and logout functionality.
* Contact Management:
  + Validate the creation, updating, and deletion of contacts.
  + Check field validations and constraints.
  + Verify that the contact list displays the correct information.
* API Endpoints:
  + Perform API testing using tools like Postman.
  + Validate response codes, response times, and data integrity.
* User Profile Management:
  + Test viewing and updating profile details.
  + Ensure correct handling of user data and privacy.
* GUI Functionality and Usability:
  + Test the responsiveness of the UI across different devices and browsers.
  + Validate the user interface elements and overall user experience.

### **3. Functional Test Cases**

#### Test Cases for Sign Up Page

##### Positive Test Cases

* Sign up with valid information in all mandatory fields
* Sign up using a valid email address
* Sign up with a password meets complexity requirements (uppercase, lowercase, numbers, symbols)
* Sign up with confirmation password matching the original password (Not applicable)
* Successful account creation with confirmation message or email (Not applicable)

##### Negative Test Cases

* Sign up with empty mandatory fields
* Sign up with an already existing username or email
* Sign up with an invalid email format
* Sign up with a password not meeting complexity requirements (too short, weak)
* Sign up with confirmation password not matching the original password
* Sign up with special characters not allowed in username or password
* Verify behavior for exceeding character limits in any field
* Test CAPTCHA functionality if implemented (Not Applicable)
* Test Terms & Conditions checkbox functionality if we have it (Not Applicable)
* Test page layout and visual elements (proper alignment, clarity)
* Test error message clarity and user-friendliness
* Test redirection upon successful login/signup (to intended pages)
* Test responsiveness of the pages on different devices (mobile, desktop)
* Test accessibility features for users with disabilities

#### Test Cases for Login Page

##### Positive Test Cases

* Login with valid username and password
* Login after successful password reset
* Login with a valid username and password within specified character limits
* Login with special characters in the password
* Successful login with remembered username and password

##### Negative Test Cases

* Login with empty username and password fields
* Login with invalid username (non-existent user)
* Login with invalid password (wrong password)
* Login with incorrect email format (if username is email)
* Login exceeding the maximum login attempts (not applicable)
* Login with a disabled account (not applicable)
* Login functionality after account deletion (user should be redirected to signup or informed) Not Applicable
* Verify "Forgot Password" link functionality (user receives reset instructions) Not Applicable
* Try logging in with an already logged-in user account

#### Test Cases for Add Contact Form

##### Positive Test Cases

* Add contact with all fields filled: Enter valid information in all fields (first name, last name, optional birthdate, email, phone, address) and submit the form. Verify successful contact addition with confirmation message or redirection.
* Add contact with minimal information: Enter data only in mandatory fields (first name, last name) and submit. Verify successful addition if allowed by the application.
* Leave birthdate blank: Submit the form with first name, last name, and leave birthdate blank. Verify successful addition as birthdate is not mandatory.

##### Negative Test Cases

* Leave all fields blank: Submit the form without entering any data. Verify error messages appear for all mandatory fields.
* Invalid first name: Enter special characters, numbers, or symbols in the first name field and submit. Verify error messages indicating only letters and spaces are allowed.
* Invalid last name: Enter special characters, numbers, or symbols in the last name field and submit. Verify error messages indicating only letters and spaces are allowed.
* Invalid email format: Enter data that doesn't follow a valid email format (e.g., missing "@", invalid domain) and submit. Verify an error message indicating a valid email address is required.
* Invalid phone format: Enter text or special characters in the phone field and submit. Verify an error message indicating a valid phone number format.
* Invalid date format: Enter a date in a format other than yyyy-MM-dd and submit. Verify an error message indicating the correct date format is required. (Not Applicable)
* Incomplete address: Enter data only in one of the address fields (street 1 or city) and submit. Verify an error message indicating a complete address is required (Not applicable).
* Exceed character limit: Enter data exceeding the character limit in any field and submit. Verify error message indicating the character limit has been exceeded.
* Test form submission functionality across different browsers.
* Test for potential security vulnerabilities like XSS by injecting scripts in input fields.

#### Test Cases for Edit Contact

##### Positive Test Cases

* Click on contact's details to view details, modify information in one or more fields (name, email, phone, etc.), and save changes. Verify the updated information is reflected in the contact list or details page.
* Modify only mandatory fields (e.g., first name) and save. Verify successful updates even if optional fields remain unchanged.

##### Negative Test Cases

* Edit a contact, leave all fields blank, and try to save. Verify error messages appear for any fields that become mandatory after editing (e.g., email if it was initially blank).
* Make similar invalid edits as in the add contact test cases (invalid email format, exceeding character limits). Verify appropriate error messages are displayed.
* Attempt to edit a contact that doesn't exist (e.g., using a broken link or invalid ID). Verify an error message indicating the contact cannot be found. (Not Applicable)

#### Test Cases for Delete Contact

##### Positive Test Cases

* Access a contact and use the designated delete button/option. Verify confirmation prompt and successful deletion with a confirmation message or disappearance from the contact list.
* Select multiple contacts for deletion and confirm deletion. Verify all selected contacts are removed.(Not Applicable)

##### Negative Test Cases

* Initiate contact deletion but choose the "cancel" option in the confirmation prompt. Verify deletion is not performed and the contact remains.
* Try to delete a contact that doesn't exist using an invalid ID or broken link. Verify an error message indicating the contact cannot be found. (Not applicable)
* If user roles have permission limitations, try deleting a contact with an account that doesn't have edit/delete permissions. Verify an error message indicating insufficient privileges.(Not Applicable)

### **4. Issues and Suggestions**

#### Critical Issues:

* Xss vulnerability: successfully submit the add contact for with script <script>alert(document.cookie)</script><script>alert(document.cookie)</script>
* Login to the application, come back to the login screen or hit the login url, the session remains maintained.This is not the ideal practice.
* If you are not logged In and you visit pages urls, you can still able to see the UI
* Forgot password option is very necessary and there are chances users would forget the password.Remember me checkbox also be there.
* Password encryption, Automatically logout (session timeout), Two factor authentication to make it more secure.
* Captcha required to avoid multiple login hits
* Same name contact details should be allowed to add, there should be a way to differentiate between different users
* Strong password regex implementation and would better if ask to re-enter the password by adding this field
* Validation error of password is showing password

#### Issues and Suggestions:

* The user interface is not fine. Proper styling required across the application.
* The application is not responsive for mobile users
* Copyright 2024 expected
* Sign up Page: Label should be Sign Up instead of Add User. Also the URL should be …/signup. There should be asterisks for mandatory fields. The validation errors need to be improved: Add validation below to each field instead of showing it on top. ‘firstName: Path `firstName` is required’ remove the unnecessary Path path, it should be simple: First Name is Required. Same goes for all others.
* Table rows data should be either center aligned w.r.t the header columns or both of them should be left aligned.
* Edit, Delete Contact functionality should be available on the contact list page to avoid extra effort.
* Add we might have hundreds of records so there should be a search bar at top
* Mouse cursor should become a hand-icon while hovering over table records.
* Dropdown values should be coming for country and state/province fields
* Import or export contacts functionality