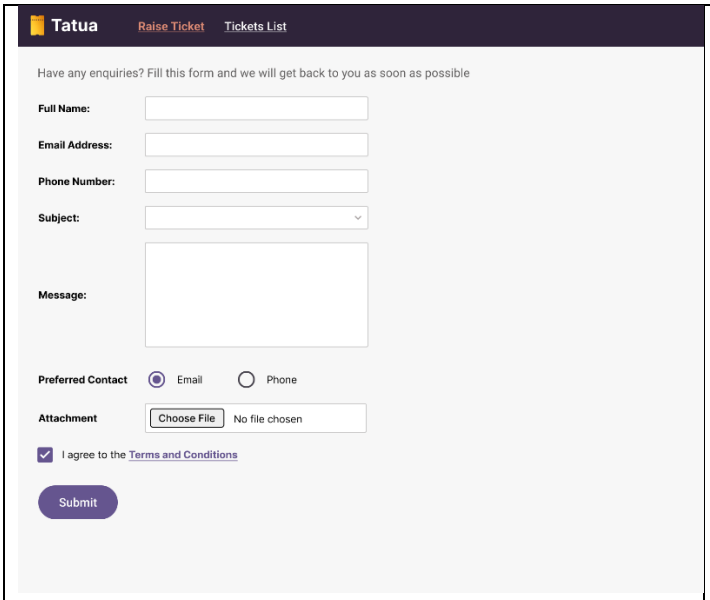
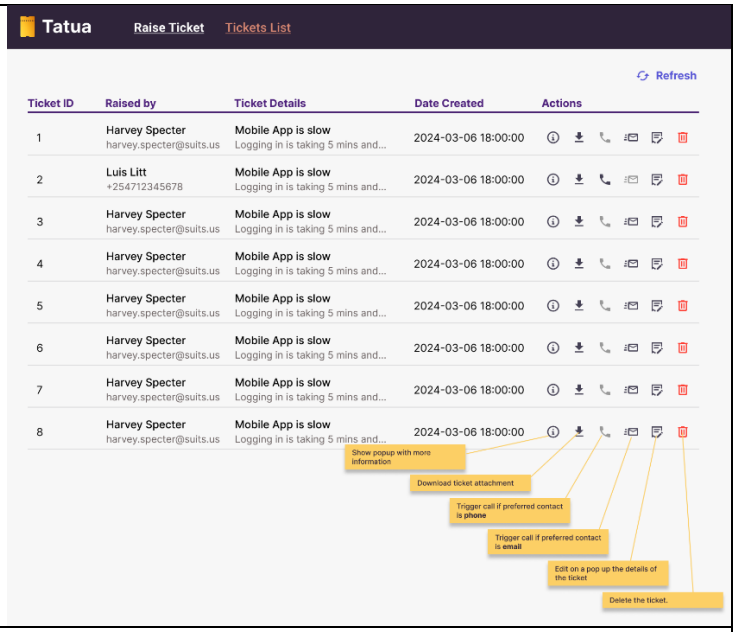
Task 7: Tatua: Ticketing System





Project Overview

Create a web application that allows users to:

1. Raise a ticket via the form shown in the image above
2. View all submissions in the data table format shown above
3. Persist data between browser sessions

Create a Github repo called **tatua-ticketing-app**

Concepts

1. JavaScript Data Types: Objects, Arrays
2. Form Elements
3. Tables and Buttons
4. Storage & Data Persistence
    a. In Memory
    b. Session Storage
    c. Local Storage

Development Process

Create a branch named part1-memory-storage

After initializing the repository, create a branch named ***part1-memory-storage*** from ***main*** to implement the in-memory version of your application:

1. Create the exact HTML form shown in the screenshot
2. Implement form styling to match the visual design
3. Add client-side validation for all fields
4. Handle form submission events
5. Create the data table structure shown in the screenshot
6. Implement JavaScript to store form submissions as objects in an array
7. Render submitted data in the table
8. Add delete functionality for entries

Once the in-memory version is implemented and tested, merge the ***part1-memory-storage*** branch into ***main***.

Create a branch named part2-session-storage

After merging the first phase, create a branch named ***part2-session-storage*** from your updated ***main*** to implement session-based persistence:

1. Modify your application to store submission data in sessionStorage
2. Implement logic to load saved data when the page loads
3. Ensure delete operations update both the display and sessionStorage
4. Test that data persists between page refreshes but is cleared when the browser session ends

Once the sessionStorage implementation is complete and tested, merge the ***part2-session-storage*** branch into ***main***.

Create a branch named part3-local-storage

After merging the second phase, create a branch named ***part3-local-storage*** from your updated ***main*** to implement permanent persistence:

1. Modify your application to store submission data in localStorage
2. Implement logic to load saved data when the page loads
3. Ensure delete operations update both the display and localStorage

Once the localStorage implementation is complete and tested, merge the ***part3-local-storage*** branch into ***main***.

Technical Requirements

- Use semantic HTML5 elements for the form structure
- Match the exact visual layout from the screenshot
- Use proper JavaScript objects and arrays for data management
- Implement form validation with user feedback
- Create clean, well-documented code with appropriate comments

Deliverables

1. A repository with all three branches showing your development process
2. A working contact form application with in-memory, sessionStorage, and localStorage implementations
3. A brief README

This project will demonstrate your ability to use Git for feature-based development while implementing a complete web application that handles form data and different types of browser storage mechanisms.