

## Design / Usage: 6.2 Custom Web Application

Source Code: <https://github.com/raveenepal99/immi-tracker.git>

### Purpose:

The purpose of this web application is to manage appointments and documents effectively. This is inspired by my work (i.e. immigration agency) where there is lot of things going on with appointments and documents.

### Functional Overview:

1. Home Page
  - a. Provides a welcoming intro to the application
  - b. Encourages users to navigate using the menu
2. Appointments Page
  - a. Lists of existing appointments with date, time and details
  - b. Users can add new appointments using a form with required fields
  - c. Appointments are stored in the browser's local storage and persist between the sessions
3. Documents Page
  - a. Allows users to upload document names and files via a form
  - b. Uploaded documents are also stored in local storage of the browser
  - c. Users can delete uploaded documents easily

### Design Approach

#### Accessibility:

- Implemented ARIA labels for all form inputs and buttons to maintain accessibility
- Input fields and buttons are easily usable using a keyboard

#### Responsive Design:

- Utilised Bootstrap's grid system and responsive utilities to ensure the web app works properly on desktop, mobile devices and tablets.

#### Local Storage Implementation:

- Both appointments and documents are saved locally, ensuring data persistence

#### Structure:

- Pages are structured using Vue Router for navigation
  - o / for Home Page
  - o /*appointments* for the Appointments Page
  - o /*documents* for the Documents Page

- Components are modular and reusable

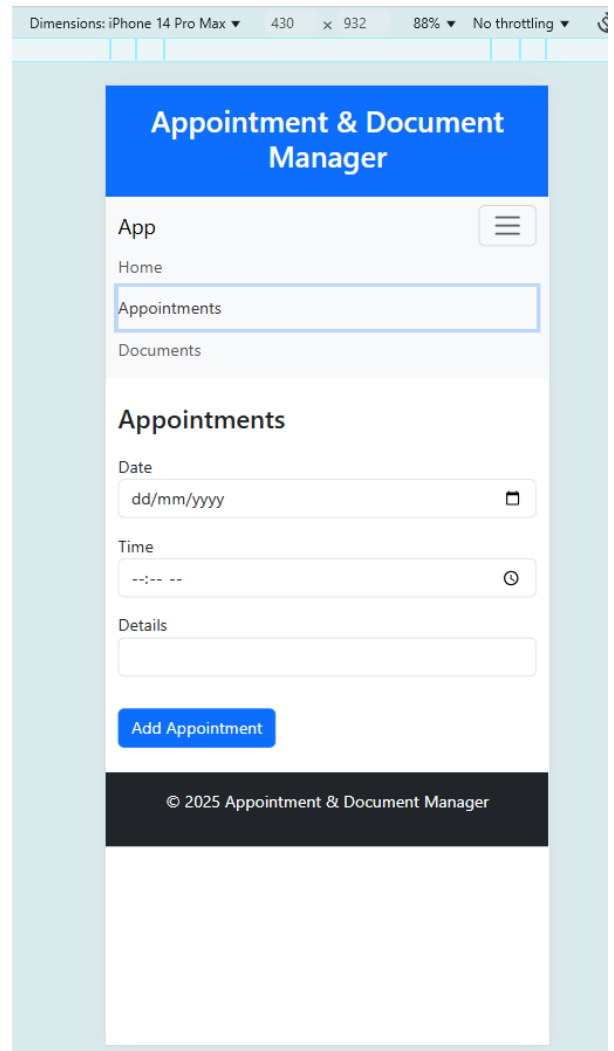
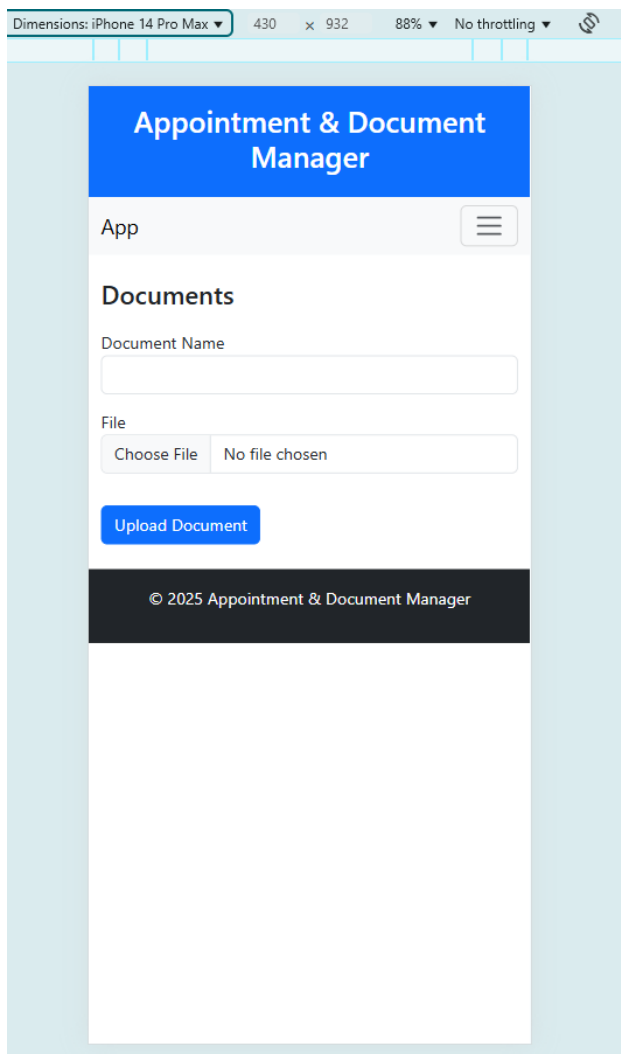
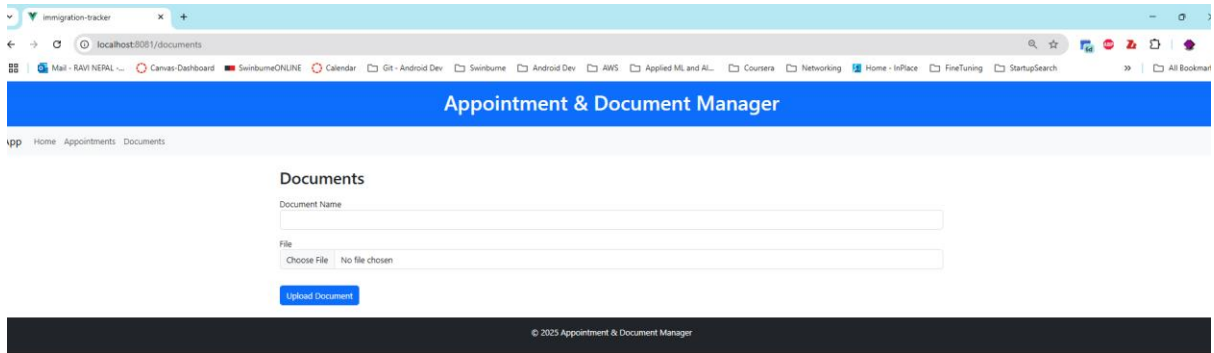
### Requirements Met:

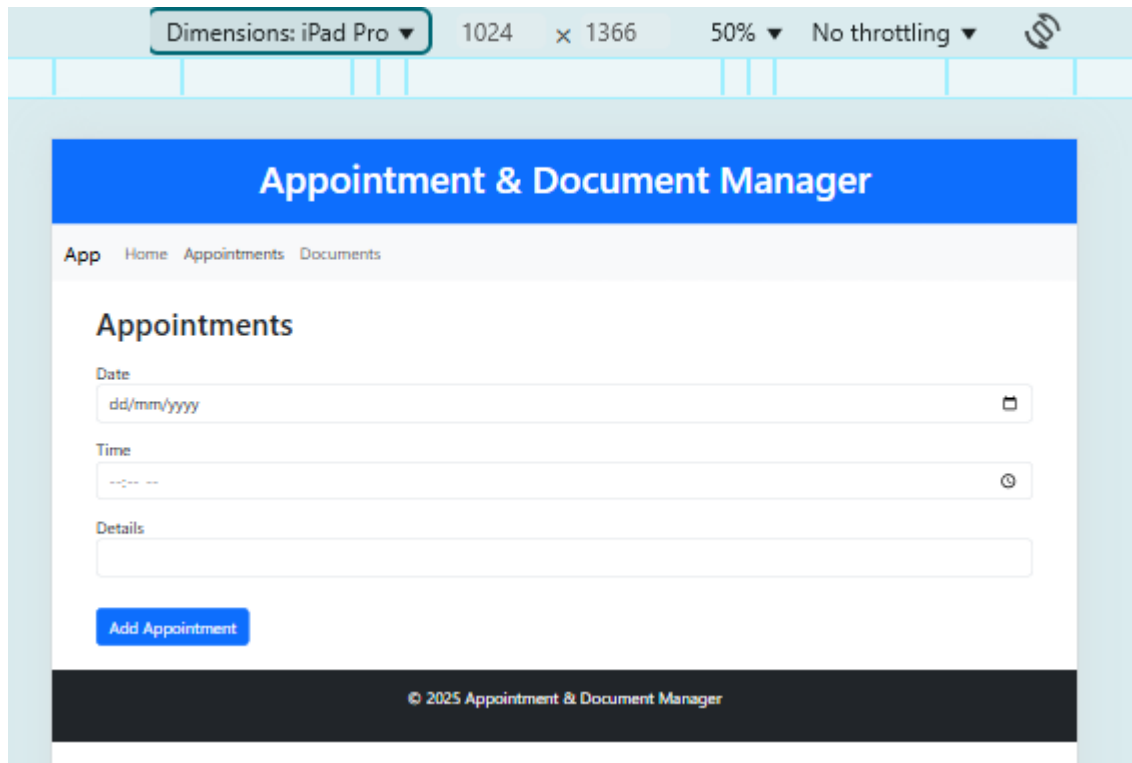
1. Row-Column Grid System
  - a. Used Bootstrap' s grid system for consistent layout design
2. Modules
  - a. Components and routing managed modularly using Vue.js and Vue Router.
3. Arrays
  - a. Managed appointments and documents using arrays in Vue.js
4. Directives and Filters
  - a. Used *v-for* and *v-model* for data binding and iteration
5. Mobile-First Design
  - a. Tested using browser developer tools to ensure proper rendering on mobile, desktop and tablet views.
6. Accessibility
  - a. ARIA labels and semantic HTML to ensure the application is accessible

### How To:

1. Navigation
  - o Click on the menu to navigate between pages
2. Adding Appointments
  - o Go to the Appointments page
  - o Fill in the date, time and details, then click *Add Appointment*
  - o The appointment appears in the list and persists even after a refresh
3. Uploading Documents
  - o Go to the Documents Page
  - o Enter the document name and upload a file, then click *Upload Document*
  - o The document name appears in the list and persists after refresh
4. Deleting Data
  - o Both appointments and documents can be removed using their respective "Delete" buttons.







## Future Enhancements

- Integration with a backend for centralised data storage
- File preview or download functionality for uploaded documents