

# COLOR ME

IIHT

Time To Complete: 2 hr

## CONTENTS

---

1	Problem Statement	3
2	Proposed Color Me Wireframe	4
2.1	Welcome Page	4
3	Business-Requirement	7
4	Validations	7
5	Constraints	8
6	Mandatory Assessment Guidelines	8

## 1 PROBLEM STATEMENT

---

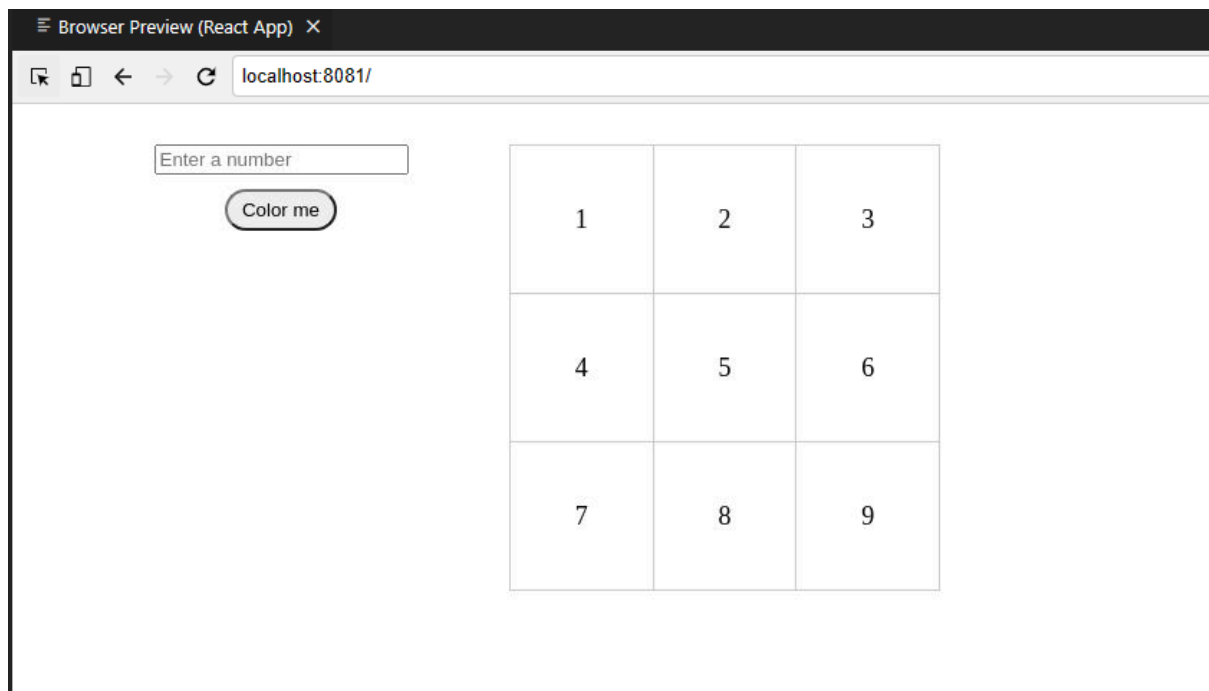
"Color Me" is a Single Page Application (SPA) that empowers users to create a 3x3 table populated with numbers ranging from 1 to 9. Users can input any number within the 1 - 9 range and utilize it to highlight a specific cell in the table with their chosen color.

## 2 PROPOSED COLOR ME WIREFRAME

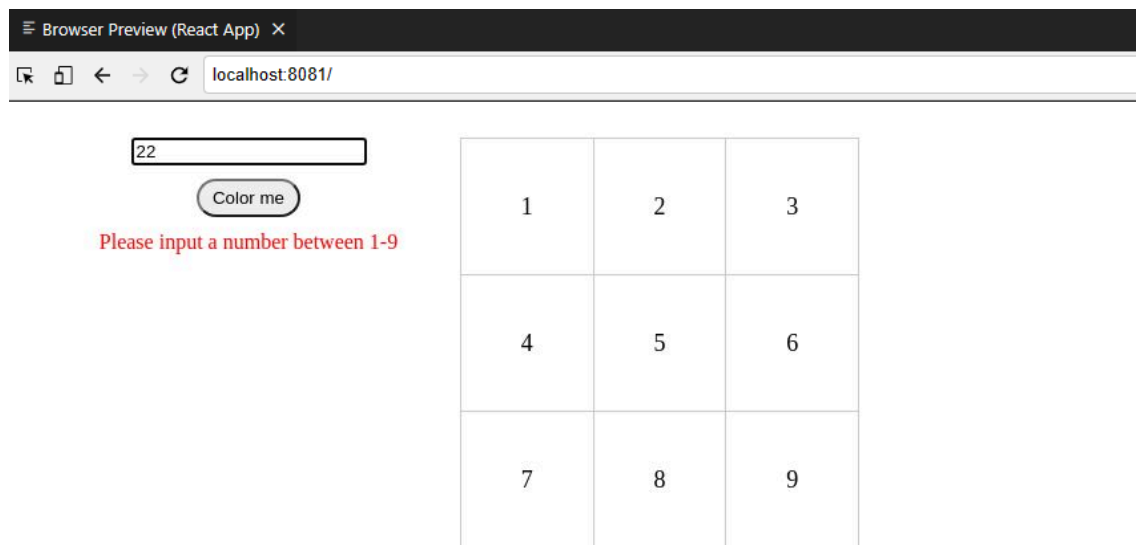
---

UI needs improvisation and modification as per given use case and to make test cases passed.

### 2.1 WELCOME PAGE



## SCREEN SHOTS



2

Color me

1	2	3
4	5	6
7	8	9

9

Color me

1	2	3
4	5	6
7	8	9

### 3 BUSINESS-REQUIREMENT:

As an application developer, develop the Color-Me (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	<p>As a user I should be able to visit the welcome page as the default page.</p> <p><b>App Component (App.js)</b></p> <p><b>Purpose:</b></p> <ol style="list-style-type: none"><li>1. Serves as the main entry point of the application.</li><li>2. Embeds the <b>ColorMe</b> component.</li></ol> <p><b>HTML Structure:</b></p> <ul style="list-style-type: none"><li>• Use a top-level fragment (<code>&lt;&gt;...&lt;/&gt;</code>) to wrap content.</li><li>• Inside it, render the <code>&lt;ColorMe /&gt;</code> component.</li></ul> <p><b>ColorMe Component</b></p> <p>Acceptance criteria:</p> <ol style="list-style-type: none"><li>1. Render a <b>text input</b> for entering a number.</li><li>2. Render a <b>button</b> labeled <b>"Color me"</b>.</li><li>3. Display a <b>3×3 grid</b> of boxes, numbered 1 to 9.</li><li>4. When the user:<ul style="list-style-type: none"><li>• Enters a valid number (1–9)</li><li>• Clicks the button:<ul style="list-style-type: none"><li>→ the corresponding box is highlighted in green.</li></ul></li></ul></li><li>5. Previously selected box should reset to white.</li><li>6. If input is invalid or empty:<ul style="list-style-type: none"><li>• Show error: <b>"Please input a number between 1-9"</b></li></ul></li></ol> <p><b>State Variables:</b></p> <ol style="list-style-type: none"><li>1. <code>selectedBox</code>: Holds the current box ID to be colored (e.g., <code>box4</code>)</li></ol>

		<p>2. <code>prevSelectedBox</code>: Stores the previously colored box ID to reset</p> <p>3. <code>errorMessage</code>: Holds the validation error message</p> <p><b>Functions &amp; Responsibilities:</b></p> <p>1. <b>handleInputChange(event):</b></p> <ul style="list-style-type: none"> <li>Triggered when the user types in the input field.</li> <li>If the value is a digit between 1–9: <ul style="list-style-type: none"> <li>→ Set <code>selectedBox</code> to the corresponding ID (e.g., <code>box3</code>)</li> <li>→ Clear <code>errorMessage</code></li> </ul> </li> <li>If invalid: <ul style="list-style-type: none"> <li>→ Clear <code>selectedBox</code></li> <li>→ Set <code>errorMessage</code> to: "Please input a number between 1-9"</li> </ul> </li> </ul> <p>2. <b>handleColorButtonClick():</b></p> <ul style="list-style-type: none"> <li>Triggered when the "Color me" button is clicked.</li> <li>If <code>selectedBox</code> is valid: <ul style="list-style-type: none"> <li>→ Reset <code>prevSelectedBox</code> background to white.</li> <li>→ Change <code>selectedBox</code> background to green.</li> <li>→ Update <code>prevSelectedBox</code>.</li> </ul> </li> <li>If <code>selectedBox</code> is not set: <ul style="list-style-type: none"> <li>→ Set <code>errorMessage</code>: "Please input a number between 1-9".</li> </ul> </li> </ul> <p><b>HTML Structure:</b></p> <p>1. <b>Overall Layout:</b></p> <ul style="list-style-type: none"> <li>Use a parent <code>&lt;div&gt;</code> with class <code>container</code></li> <li>Divide into two columns using nested <code>&lt;div&gt;</code>s: <ul style="list-style-type: none"> <li>→ Left: form controls</li> <li>→ Right: grid of boxes</li> </ul> </li> </ul> <p>2. <b>Left Column:</b></p>
--	--	--



		<ul style="list-style-type: none"> <li>• Input field: <ul style="list-style-type: none"> <li>→ Type: text</li> <li>→ Placeholder: "Enter a number"</li> <li>→ Calls <code>handleInputChange</code> on change</li> </ul> </li> <li>• Button: <ul style="list-style-type: none"> <li>→ Label: "Color me"</li> <li>→ Calls <code>handleColorButtonClick</code> on click</li> </ul> </li> <li>• Error message: <ul style="list-style-type: none"> <li>→ Shown only if <code>errorMessage</code> is not empty</li> <li>→ Styled using <code>.error-message</code> class</li> </ul> </li> </ul> <p><b>3. Right Column (3×3 Grid):</b></p> <ul style="list-style-type: none"> <li>• Use a <code>&lt;table&gt;</code> to render 3 rows and 3 columns</li> <li>• Each cell: <ul style="list-style-type: none"> <li>→ Contains a number (1 to 9)</li> <li>→ Has a unique <code>id</code>: <code>box1</code> to <code>box9</code></li> <li>→ Uses a common class <code>box</code> for styling</li> </ul> </li> </ul> <p><b>Dynamic Behavior:</b></p> <ol style="list-style-type: none"> <li>1. Initially, all boxes are white.</li> <li>2. Entering a valid number and clicking "Color me": <ul style="list-style-type: none"> <li>→ Highlights the corresponding box in green</li> <li>→ Reverts the previously selected box to white</li> </ul> </li> <li>3. Invalid input (empty, non-numeric, out of range): <ul style="list-style-type: none"> <li>→ Displays error message below the button</li> <li>→ Does not color any box</li> </ul> </li> </ol> <p><b>** Kindly refer to the screenshots for any clarifications. **</b></p>
--	--	---

## 4 VALIDATIONS

---

- All required fields must be fulfilled with valid data.
- Number field is required and on invalid number in field if entered excluding numbers from 1-9, "Please input a number between 1-9" message should be shown.
- In the starting "Color me" button should be disabled.
- Only after validating fields, "Color me" button should be enabled.
- Once we click on Color Me, that input number box must be filled with color and previously colored (if any) box must be filled with white color(default color).

## 5 CONSTRAINTS

---

- You should be able to press the "TAB" key and "SHIFT + TAB" to navigate from top field to bottom field and vice-versa.
- On clicking the "Color me" button, the particular entered number cell must get colored.
- "Color me" button will be disabled until all validations are fulfilled.
- If input is not between 1-9 then "Please input a number between 1-9" error should be shown.

## 6 MANDATORY ASSESSMENT GUIDELINES

---

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. This editor Auto Saves the code.
4. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
5. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
6. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel

of IDE, you can find Browser Preview, where you can launch the application.

**Note: The application will not run in the local browser**

7. You can follow series of command to setup React environment once you are in your project-name folder:
  - a. `npm install` -> Will install all dependencies -> takes 10 to 15 min
  - b. `npm run start` -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:8080/8081 to open project in browser -> takes 2 to 3 min
  - c. `npm run test` -> to run all test cases. **It is mandatory to run this command before submission of workspace** -> takes 5 to 6 min
8. You may also run “`npm run jest`” while developing the solution to re-factor the code to pass the test-cases.
9. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on **“Submit Assessment”** after you are done with code.