## Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

**Background Color Selector**

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Raveesh Samkaria Shweta Wadhwa

Roll Number:2210990847

Group - 13

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**Abstract**

This project presents a front-end website enabling users to effortlessly customize their browsing environment through color palette selection. Leveraging HTML, CSS, and JavaScript, the platform provides intuitive tools for choosing plain colors or gradients, complemented by a dark mode toggle. Emphasizing simplicity and functionality, the project offers a seamless interface where users can effortlessly switch between plain color mode and gradient mode, fostering creativity and enhancing the digital experience.Through this project, users are invited to explore the realm of color possibilities, fostering creativity and personalization in their online interactions.

窗体底端

窗体顶端

窗体底端

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Section** | **Page No** |
| **1.** | Introduction | 4 |
| **2.** | Problem Statement | 5 |
| **3.** | Technical Details | 6 |
| **4.** | File Structure | 7 |
| **5.** | Result | 16 |
| **6.** | References | 23 |

**Introduction**

In today's digital landscape, the importance of a captivating online presence is undeniable, especially for businesses aiming to effectively engage their audience. This report chronicles the development journey of crafting a front-end website dedicated to the dynamic realm of color palette customization, leveraging the fundamental languages of HTML, CSS, and JavaScript. Just as a finely curated dish tantalizes the palate, this website endeavors to captivate users through its visually enticing design and user-friendly interface.

Through the fusion of artistic flair and technical prowess, the website aspires to provide visitors with an immersive experience, inviting them to explore the boundless possibilities and creative expressions inherent in color selection. With a focus on user engagement and accessibility, every aspect of the website is meticulously tailored to spark curiosity and inspire users to experiment with diverse color schemes. From intuitive color pickers to interactive features, each element is designed to elevate the browsing experience and empower users to personalize their digital environment.

This introduction sets the stage for a deeper exploration of the development process, delving into key design decisions, technical implementations, and innovative solutions. By documenting the journey from conception to execution, this report not only showcases the harmonious blend of design and technology but also underscores the commitment to delivering exceptional digital experiences. Through the lens of this project, the convergence of creativity, functionality, and digital ingenuity comes to life, inviting users to embark on a journey of color exploration and customization in a dynamic online setting.

**Problem Statement**

In the digital realm of color palette customization, the creation of an engaging online platform poses a multifaceted challenge that extends beyond mere aesthetics. The project endeavors to provide users with a seamless experience, where they can effortlessly tailor their digital environment to reflect their individual preferences and tastes. Achieving this goal requires a delicate balance between simplicity, functionality, and accessibility, ensuring that users of all backgrounds and abilities can interact with the website with ease.

One of the primary challenges lies in striking the right balance between simplicity and functionality. While the website should offer a wide range of customization options, it must do so in a way that is intuitive and user-friendly. Complex interfaces or convoluted processes may deter users from exploring the full potential of the platform. Therefore, careful attention must be paid to the design and layout of the website to ensure that users can navigate effortlessly and find the tools they need to personalize their color palette.

Another crucial aspect is responsiveness across different devices and screen sizes. In an era dominated by mobile devices, it is essential that the website functions seamlessly on smartphones, tablets, and desktops alike. This necessitates the implementation of responsive design principles, where the layout and content of the website adapt dynamically to the user's device, ensuring a consistent and optimized experience regardless of the screen size.

Furthermore, catering to the diverse preferences and accessibility needs of users presents a significant challenge. Beyond the visual aspect, considerations must be made for users with color vision deficiencies or visual impairments. Providing alternative color options, high contrast modes, and accessible features such as keyboard navigation can enhance the exclusivity of the website and ensure that all users can engage with the platform effectively.

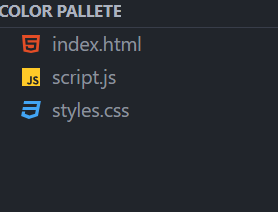
In summary, the project's objective is to develop a front-end website for color palette customization that not only offers a rich array of customization options but also prioritizes simplicity, functionality, and accessibility. By addressing these challenges comprehensively, the website aims to empower users in personalizing their digital environment and fostering a sense of ownership and satisfaction in their online interactions.

**Technical Details**

1. ****HTML Structure:**** The foundation of the color palette website is built using semantic HTML markup. Each page is meticulously structured with appropriate HTML tags, ensuring intuitive navigation and readability for visitors. Semantic markup enhances accessibility by providing clear and meaningful structure to the content, allowing users to easily understand the purpose of each element on the page.
2. ****CSS Styling:**** Cascading Style Sheets (CSS) play a crucial role in styling the layout, typography, and color schemes of the color palette website. Custom CSS rules are carefully applied to maintain a cohesive visual design across different pages and components. By prioritizing optimized load times and code simplicity, the CSS ensures that the website delivers a visually appealing user experience while also enhancing performance.
3. ****Bootstrap Integration:**** The Bootstrap framework is seamlessly integrated into the color palette website to streamline development and enhance responsiveness. Leveraging Bootstrap's prebuilt CSS components and JavaScript plugins, the website implements features such as navigation bars, buttons, forms, and responsive grid layouts. This integration enables rapid development and ensures a consistent appearance across various devices and screen sizes, thanks to Bootstrap's robust responsive grid system.
4. ****JavaScript Functionality:**** JavaScript (JS) adds interactive and dynamic features to the website, enhancing user engagement and customization options. JS functionalities include color selection, gradient generation, dark mode toggle, and buttons for generating random colors/gradients. Event listeners detect user interactions, triggering corresponding actions to provide an intuitive user experience. Additionally, JS dynamically updates color display text and manages the visibility of certain elements based on user actions, contributing to a seamless and engaging browsing experience.

**File Structure: -**

1. **index.html:** Serving as the central entry point, this HTML file outlines the structure and content of the web pages. It encompasses sections like the header, navigation bar, main content area, footer, and any additional elements necessary for a cohesive user experience.
2. **styles.css:** This CSS file houses the styling rules governing the visual presentation of the website. It includes directives on colors, fonts, layout properties, and other stylistic elements, ensuring a consistent and visually appealing design across the site. Both general styling and responsive design considerations are encapsulated within this file.
3. **script.js:** The script.js file contains JavaScript code that adds interactivity and dynamic functionality to the website. It handles tasks such as color selection, gradient generation, dark mode toggling, and other user interactions. By implementing event listeners and DOM manipulation, this file enhances the user experience and adds depth to the website's functionality.



**Code:-**

***Figure 1 HTML - head tag***

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Color Palette</title>

    <!-- Bootstrap CSS -->

    <link

      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

      rel="stylesheet"

    />

    <link

      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"

      rel="stylesheet"

    />

    <link rel="stylesheet" href="styles.css" />

**Figure 2 HTML - navbar**

<header>

      <!-- Navbar with brand logo and toggle button -->

      <nav class="navbar navbar-expand-lg navbar-light bg-light">

        <a class="navbar-brand" href="#">

          <!-- Brand logo with Font Awesome icon -->

          <i class="fas fa-palette mr-2"></i>Color Palette

        </a>

        <!-- Navbar toggle button for responsive navigation -->

        <button

          class="navbar-toggler"

          type="button"

          data-toggle="collapse"

          data-target="#navbarNav"

          aria-controls="navbarNav"

          aria-expanded="false"

          aria-label="Toggle navigation"

        >

          <span class="navbar-toggler-icon"></span>

        </button>

        <!-- Navbar content with menu items -->

        <div

          class="collapse navbar-collapse justify-content-end"

          id="navbarNav"

        >

          <ul class="navbar-nav">

**Figure 3 HTML - plain button**

            <!-- Menu item for plain color selection -->

            <li class="nav-item">

              <a class="nav-link" href="#" id="plainBtn">

                <i class="fas fa-paint-brush"></i>

              </a>

***Figure 5 HTML - gradient button***

            <!-- Menu item for gradient color selection -->

            <li class="nav-item">

              <a class="nav-link" href="#" id="gradientBtn">

                <i class="fas fa-fill-drip"></i>

              </a>

            </li>

***Figure 6 HTML - dark-light mode***

            <!-- Menu item for dark mode toggle -->

            <li class="nav-item">

              <a class="nav-link" href="#" id="darkModeBtn">

                <i class="fas fa-moon"></i>

              </a>

            </li>

          </ul>

        </div>

      </nav>

    </header>

**Figure 7 HTML - main content**

    <div class="container mt-5">

      <div class="row justify-content-center">

        <div class="col-md-6">

          <!-- Title for color selector -->

          <h2 class="text-center mb-4">Background Color Selector</h2>

          <div class="color-picker-container" id="colorPickerContainer">

            <!-- Color input fields -->

            <div class="form-group">

              <input

                type="color"

                id="color1"

                class="form-control"

                value="#00ff00"

              />

            </div>

            <div class="form-group">

              <input

                type="color"

                id="color2"

                class="form-control"

                value="#0000ff"

              />

            </div>

            <!-- Color display text -->

            <div class="color-display mt-3" id="colorDisplay">

              #00ff00 to #0000ff

            </div>

            <!-- Options container for gradient selection -->

            <div class="options mt-3" id="optionsContainer">

              <select

                id="gradientType"

                class="form-control"

                style="display: none"

              >

                <option value="linear">Linear Gradient</option>

              </select>

              <!-- Button for generating random gradient -->

              <button id="randomGradientBtn" class="btn btn-primary mt-2">

                Random Gradient

              </button>

            </div>

            <!-- Options container for plain color selection -->

            <div class="options mt-3" id="plainOptionsContainer">

              <!-- Button for generating random color -->

              <button id="randomColorBtn" class="btn btn-primary mt-2">

                Random Color

              </button>

            </div>

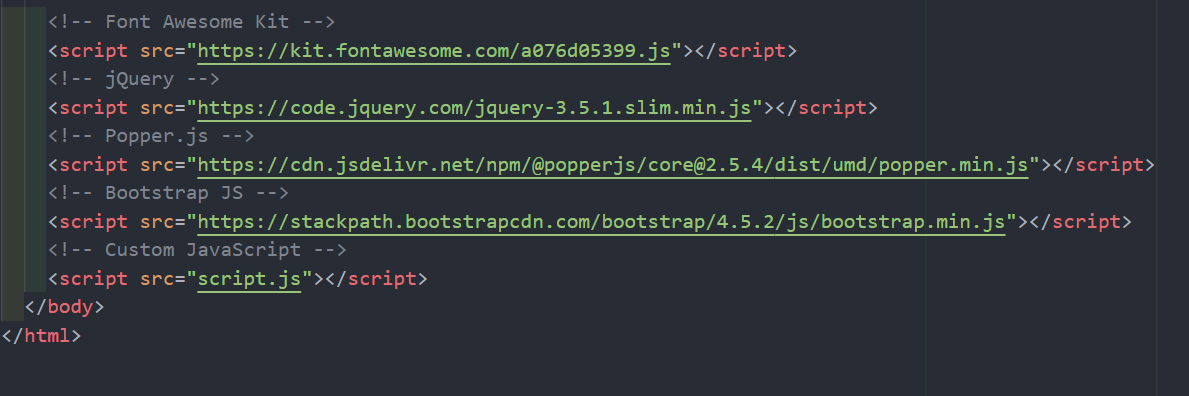
          </div>

        </div>

      </div>

    </div>

***Figure 8 HTML - CDN links***

******

***Figure 9 CSS - body***

body {

  background-color: #f8f9fa;

  transition: background-color 0.3s ease;

}

***Figure 10 CSS - navbar***

.navbar {

  background-color: #f8f9fa **!important**;

}

.navbar-nav .nav-link {

  color: #000 **!important**;

}

***Figure 11 CSS - dark mode***

.dark-mode body {

  background-color: #1f1f1f;

  transition: background-color 0.3s ease;

}

.dark-mode .navbar {

  background-color: #343a40 **!important**;

}

.dark-mode .navbar-nav .nav-link {

  color: #fff **!important**;

}

.dark-mode .color-picker-container {

  background-color: #343a40;

  border: 1px solid #ced4da;

}

.dark-mode .options {

  background-color: #343a40;

  border: 1px solid #ced4da;

}

.dark-mode .navbar-brand,

.dark-mode .color-display {

  color: #fff;

}

***Figure 12 JS - DOM***

document.**addEventListener**('DOMContentLoaded', function () {

  const colorPickerContainer = document.**getElementById**('colorPickerContainer');

  const optionsContainer = document.**getElementById**('optionsContainer');

  const plainOptionsContainer = document.**getElementById**('plainOptionsContainer');

  const color1 = document.**getElementById**('color1');

  const color2 = document.**getElementById**('color2');

  const colorDisplay = document.**getElementById**('colorDisplay');

  const plainBtn = document.**getElementById**('plainBtn');

  const gradientBtn = document.**getElementById**('gradientBtn');

  const darkModeBtn = document.**getElementById**('darkModeBtn');

  const gradientType = document.**getElementById**('gradientType');

  const randomGradientBtn = document.**getElementById**('randomGradientBtn');

  const randomColorBtn = document.**getElementById**('randomColorBtn');

***Figure 13 JS - Plainbackground***

  function **setPlainBackground**() {

      const colorValue = color1.value;

      document.body.style.background = colorValue;

      colorDisplay.innerText = `${colorValue}`;

  }

***Figure 14 - Gradient Background***

  function **setGradientBackground**() {

    const gradientColor1 = color1.value;

    const gradientColor2 = color2.value;

    // Applying linear gradient that smoothly transitions between two colors

    const gradient = `linear-gradient(to right, ${gradientColor1}, ${gradientColor2})`;

    document.body.style.background = gradient;

    colorDisplay.innerText = `${gradientColor1} to ${gradientColor2}`;

}

***Figure 15 - Generate Random Color***

  function **generateRandomColor**() {

      const randomColor = '#' + Math.**floor**(Math.**random**() \* 16777215).**toString**(16);

      color1.value = randomColor;

**setPlainBackground**();

  }

***Figure 16 - Generate Random Gradient***

  function **generateRandomGradient**() {

      const randomColor1 = '#' + Math.**floor**(Math.**random**() \* 16777215).**toString**(16);

      const randomColor2 = '#' + Math.**floor**(Math.**random**() \* 16777215).**toString**(16);

      color1.value = randomColor1;

      color2.value = randomColor2;

**setGradientBackground**();randomColorBtn.**addEventListener**('click', **generateRandomColor**);

  randomGradientBtn.**addEventListener**('click', **generateRandomGradient**);

  }

***Figure 17 - Event Listener Plain Button***

  plainBtn.**addEventListener**('click', function () {

      colorPickerContainer.style.display = 'block';

      optionsContainer.style.display = 'none';

      plainOptionsContainer.style.display = 'block';

  });

***Figure 18 - Event Listener Gradient Button***

gradientBtn.**addEventListener**('click', function () {

      colorPickerContainer.style.display = 'block';

      optionsContainer.style.display = 'block';

      plainOptionsContainer.style.display = 'none';

  });

***Figure 19 JS - Dark Mode Light Mode***

  darkModeBtn.**addEventListener**('click', function () {

      document.body.classList.**toggle**('dark-mode');

      const isDarkMode = document.body.classList.**contains**('dark-mode');

      darkModeBtn.innerHTML = isDarkMode ? '<i class="far fa-moon"></i>' : '<i class="far fa-sun"></i>';

  });

const isDarkMode = window.**matchMedia** && window.**matchMedia**('(prefers-color-scheme: dark)').matches;

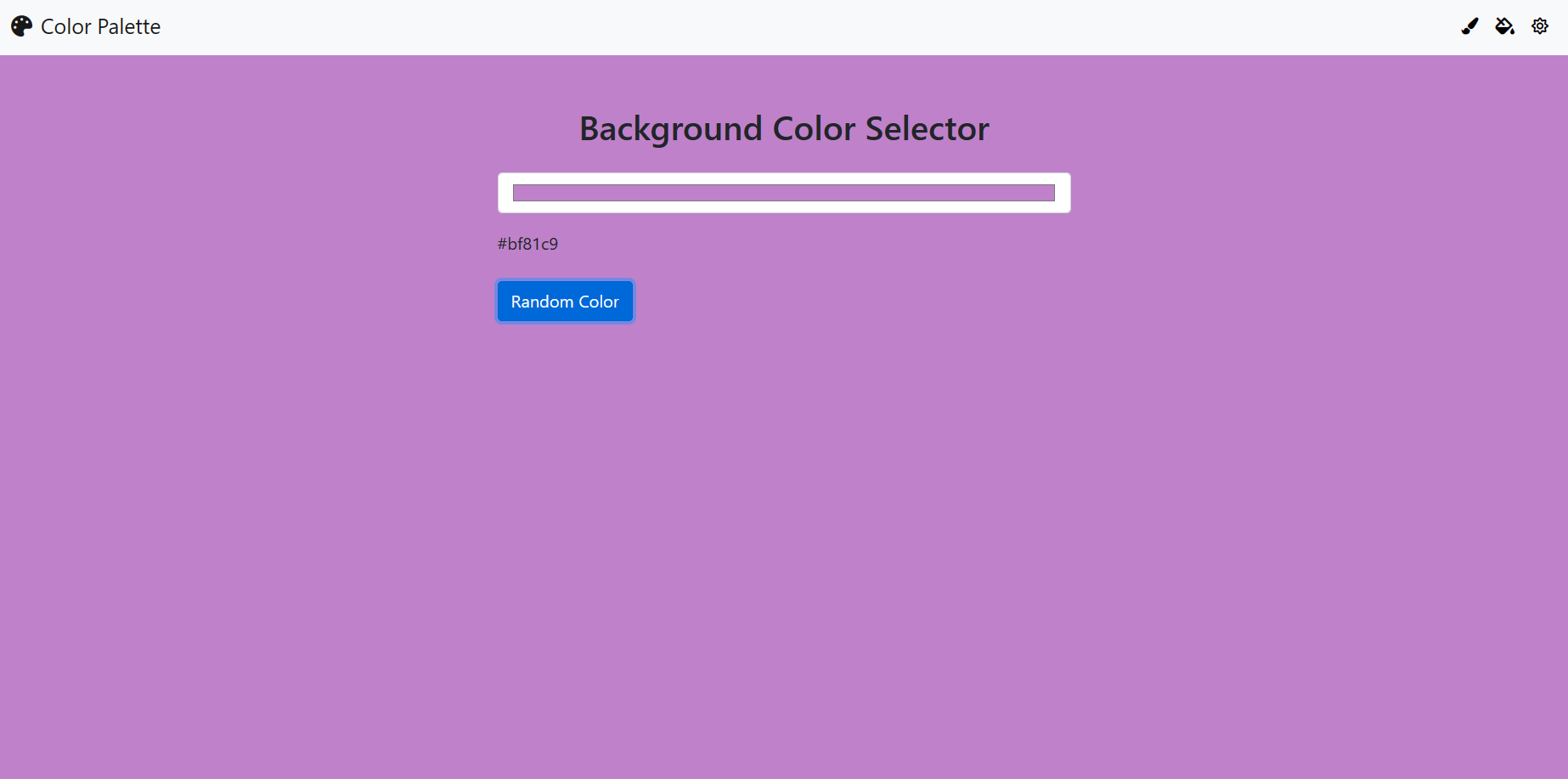
  if (isDarkMode) {

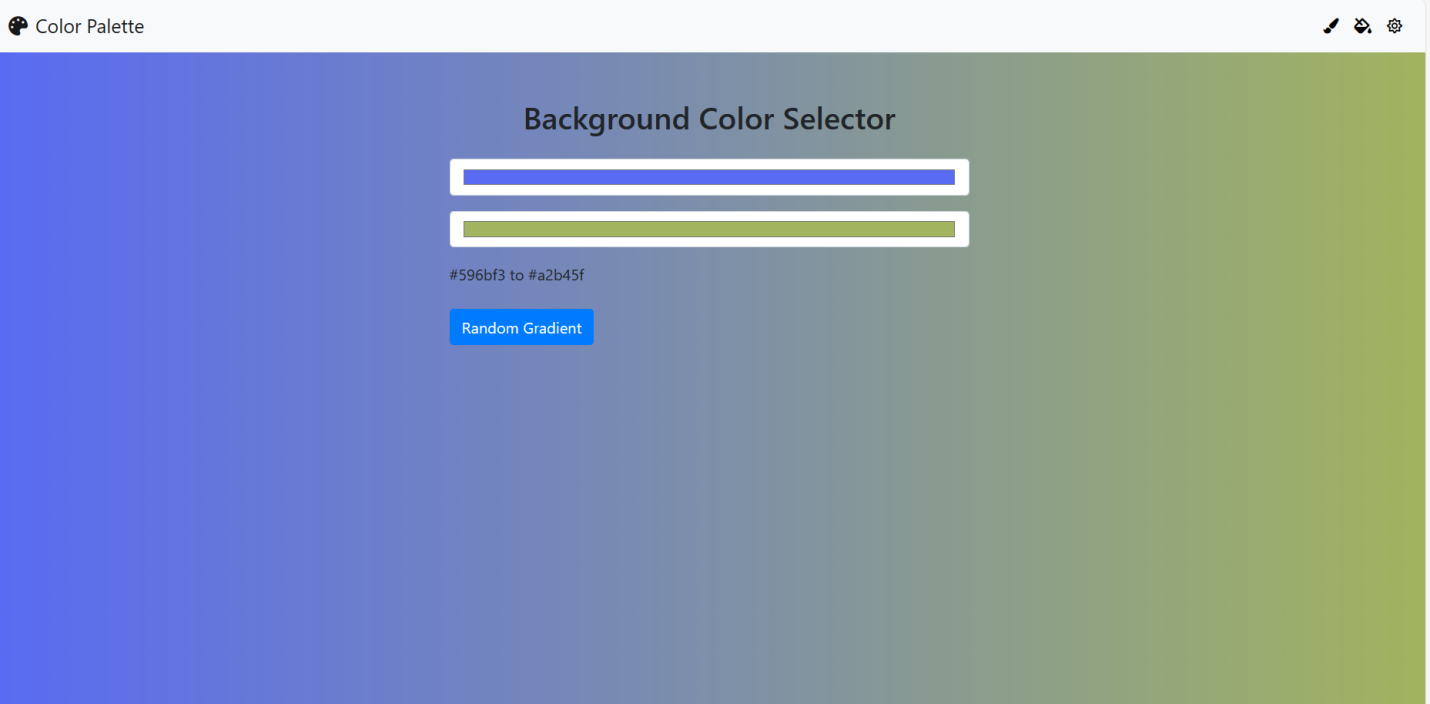
      document.body.classList.**add**('dark-mode');

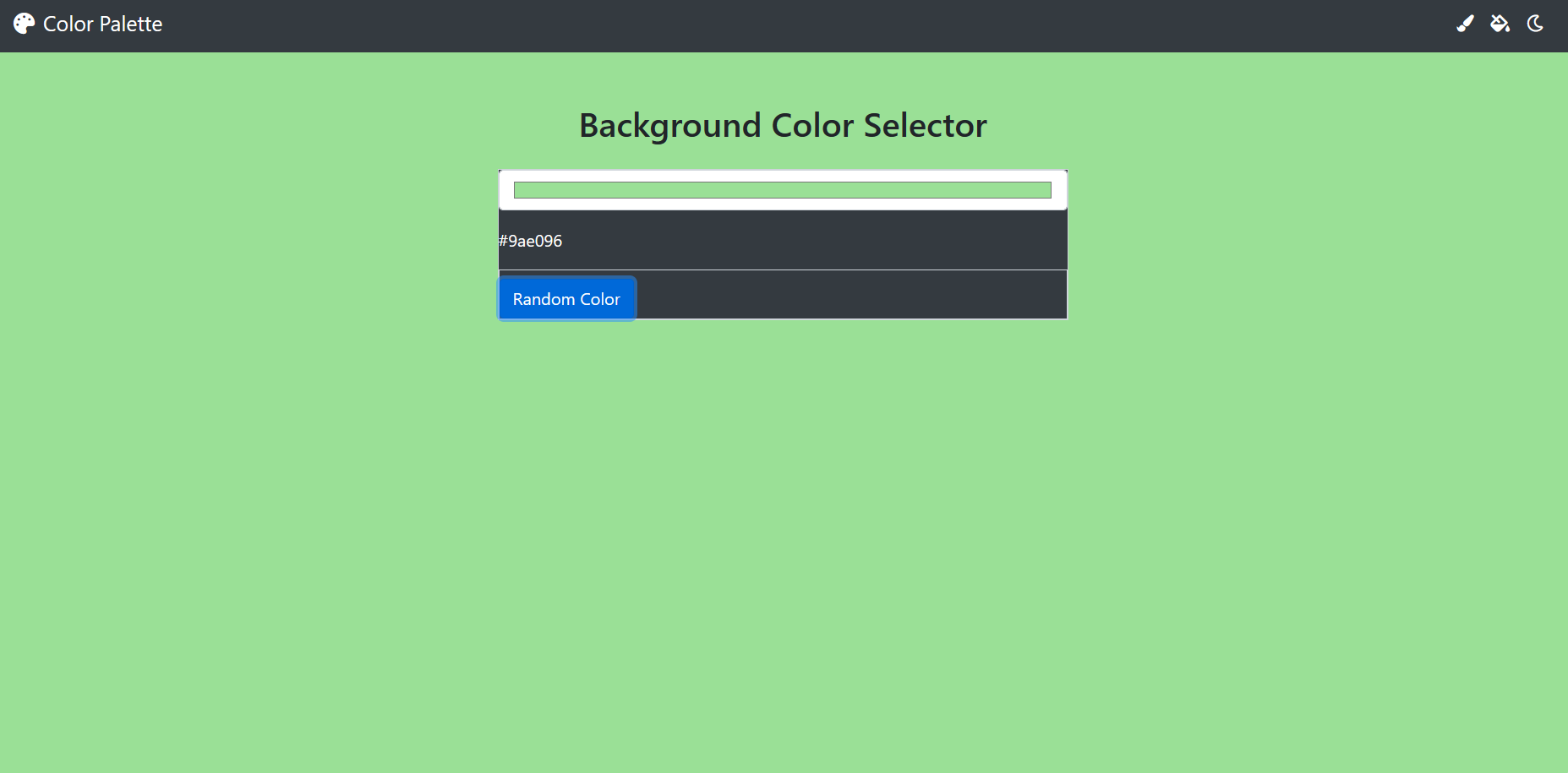
      darkModeBtn.innerHTML = '<i class="far fa-moon"></i>';

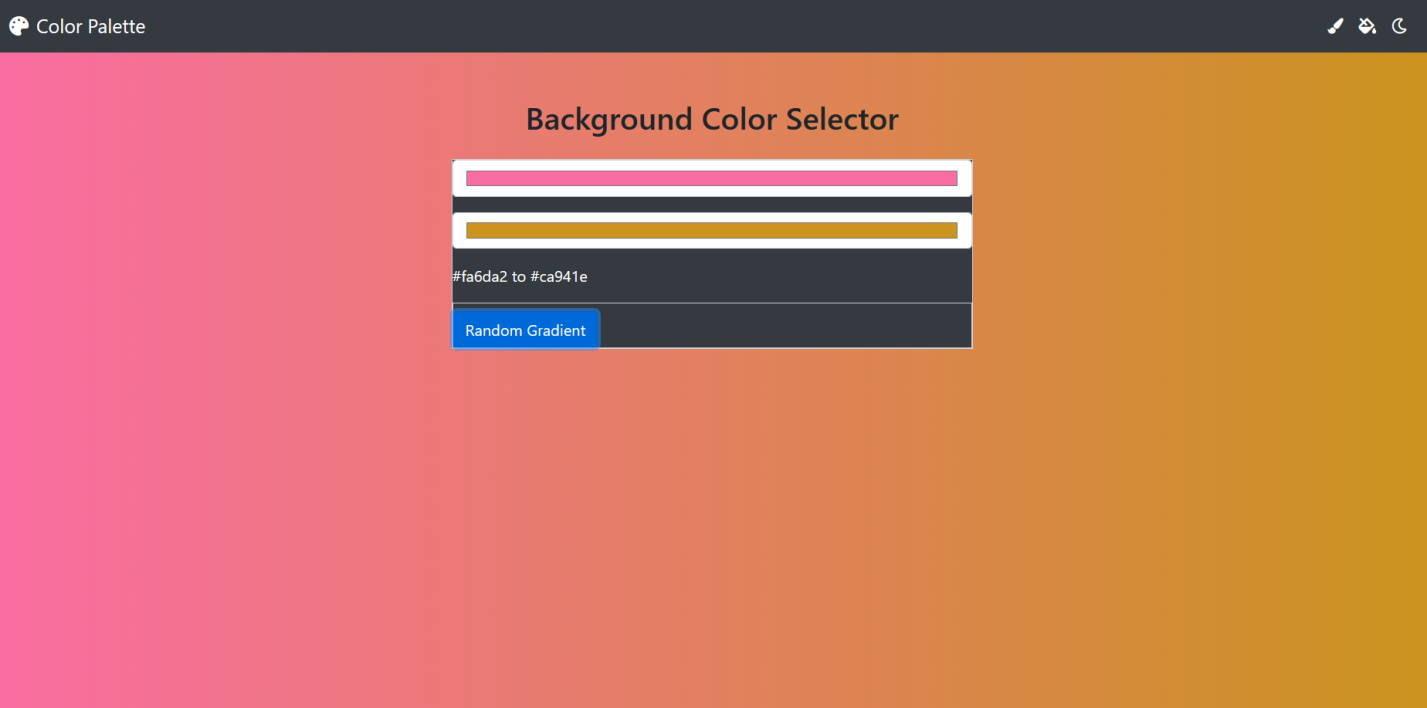
  }

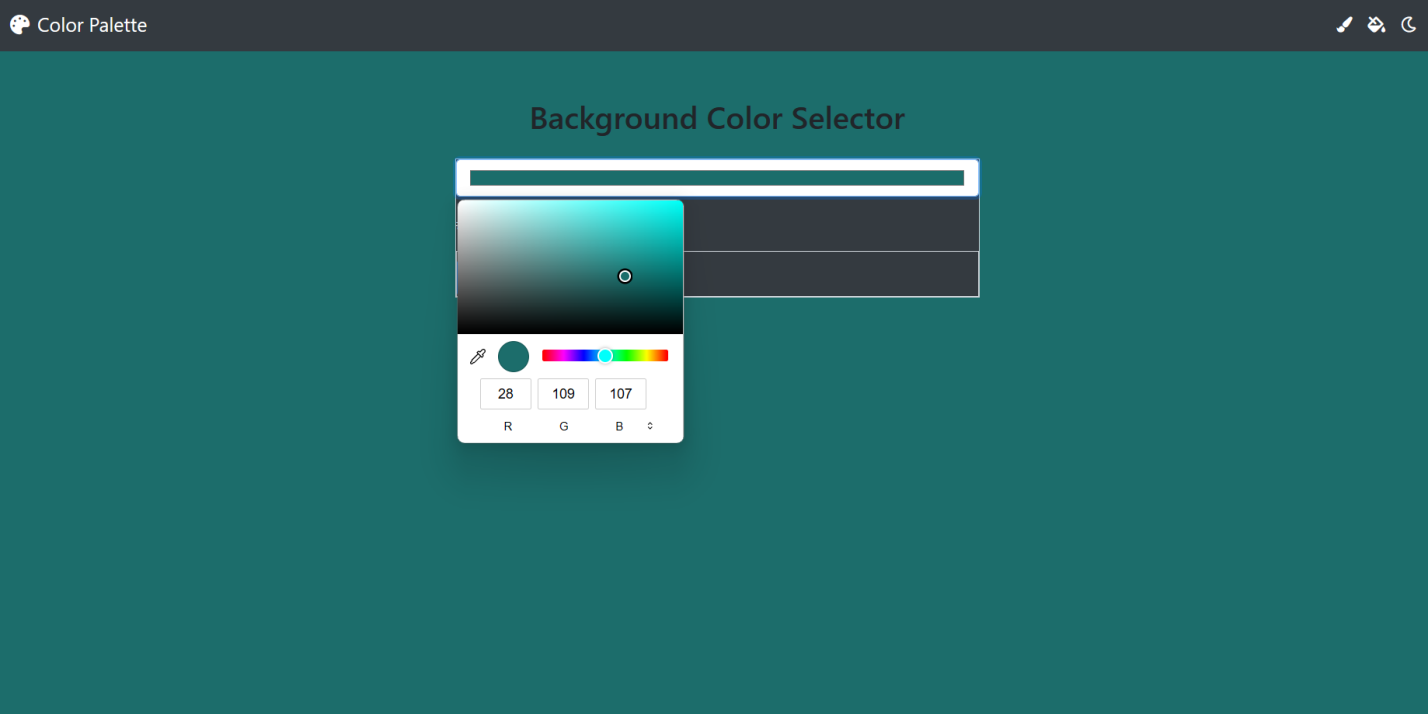
**Result -** Web-page Screenshots: -







****



***References -***

1. [www.w3schools.com](http://www.w3schools.com) - Used this website to use the concepts of JavaScript and Bootstrap
2. <https://getbootstrap.com/docs/5.3/getting-started/introduction/> - Used this website for understanding bootstrap classes
3. <https://fonts.google.com/> - provides the font used