

Established – 1961

Subject: Web Designing

SEVA SADAN'S
R. K. TALREJA COLLEGE
OF
ARTS, SCIENCE & COMMERCE
ULHASNAGAR – 421 003



CERTIFICATE

This is to certify that Ms. Prajakta Mahendra Sangulkar of F.Y. Information Technology (FYIT) Roll No. 2541039 has satisfactorily completed the Web Designing Mini Project entitled “Portfolio with Responsive Layout & Media Integration” during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for submission.

PROF. INCHARGE

HEAD OF DEPT

INDEX

SR. NO.	CHAPTERS	PAGE NO.
1.	INTRODUCTION	1
2.	REQUIREMENT SPECIFICATION	2
3.	SYSTEM DESIGN	3
4.	SYSTEM IMPLEMENTATION	5
5.	SYSTEM TESTING AND RESULTS	7
6.	FUTURE SCOPE AND CONCLUSION	8
7.	REFERENCES	10
8.	GLOSSARY	11

1.INTRODUCTION

The Academic Web Portfolio is a responsive, single-page web application designed to showcase a student's professional identity and technical growth. By replacing static resumes with an interactive platform, it integrates project previews, skill sets, and industry certifications into a unified digital hub.

The project emphasizes modern UI/UX principles, including a custom dark/light theme engine and a mobile-first responsive layout. Built using HTML5, CSS3, JavaScript, and jQuery, it demonstrates a practical application of frontend technologies to create a functional and accessible professional presence.

2.REQUIREMENT SPECIFICATION

Functional Requirements

The system is designed to perform the following core functions:

- **Dynamic Theme Switching:** The application provides a user-controlled toggle to switch between Light Mode and Dark Mode. This function dynamically updates CSS variables to change the entire color palette of the site.
- **State Persistence:** The system uses the Web Storage API (localStorage) to remember the user's theme preference. If a user selects Dark Mode and refreshes the page, the function ensures the site remains in Dark Mode.
- **Interactive Modal Overlay:** The "Quick Details" function allows users to view extra information about projects (like the Ticket Reservation System) via a popup window. This uses jQuery to overlay content without reloading the page.
- **Smooth Scroll Navigation:** The navigation menu functions by linking to section IDs. It uses the scroll-behavior: smooth function to transition the user seamlessly between Home, About, Skills, and Contact.
- **Mobile-First Responsiveness:** The layout functions as a multi-column grid on desktop but automatically collapses into a single-column list on mobile devices using CSS media queries.
- **Media & Document Integration:** The project functions as a gateway to external resources, linking directly to certification PDFs and GitHub repositories.

3.SYSTEM DESIGN

The system is designed using a Modular Frontend Architecture, separating the content, style, and logic into three distinct layers to ensure easy maintenance and scalability.

3.1 Architecture Overview

- **Structural Layer (HTML):** Defines the semantic skeleton of the portfolio, organizing data into logical sections like About, Skills, and Contact.
- **Presentation Layer (CSS):** Manages the visual aesthetic. It uses a Variable-driven Design System to allow for real-time theme changes.
- **Logical Layer (JavaScript/jQuery):** Handles user interactions, such as theme toggling and modal animations.

3.2 Theme Engine Logic

The design utilizes CSS Custom Properties (Variables) defined at the :root level.

- **Mechanism:** When the user clicks the toggle button, a JavaScript function switches the global theme attribute.
- **Storage:** The selected theme state is saved in the browser's LocalStorage, ensuring that the user's preference (Dark or Light) persists even after the page is refreshed.

3.3 Responsive Grid Layout

To ensure the portfolio is accessible on all devices, the system employs a Card-based Grid System.

- Fluidity: Elements for "Skills" and "Certificates" are placed in flexible grid containers.
- Adaptability: The design uses CSS Media Queries to automatically restack these cards from a multi-column desktop view to a single-column mobile view, ensuring readability without manual resizing.

3.4 Interface Interaction (Modals)

The design uses Modals to display detailed project information. This prevents the user from being redirected to a different page, keeping the experience fast and integrated. The modal's visibility is controlled by jQuery's fade effects for a smooth transition.

4.SYSTEM IMPLEMENTATION

System implementation involves the actual coding and integration of technologies to transform the design into a functional web application. The project was built using a standard frontend stack, focusing on clean, semantic code and modern CSS techniques.

4.1 Technology Stack

- **HTML5:** Used for the semantic structure of the website, ensuring that search engines and screen readers can interpret the content correctly.
- **CSS3:** Utilized for advanced styling, including Flexbox for the navigation bar and CSS Grid for the responsive card layouts.
- **JavaScript:** Handled the logic for theme detection and the storage of user preferences.
- **jQuery:** Used to simplify DOM manipulation, specifically for handling click events and smooth modal animations.

4.2 Implementation of Key Features

1.Dark/Light Theme Engine:

Implemented using CSS variables (--bg-color, --text-color). JavaScript was used to toggle a .dark-theme class on the body tag, which overrides these variables.

2.State Persistence:

The localStorage.setItem() and localStorage.getItem()

methods were implemented to ensure the browser remembers if a user preferred Dark or Light mode after the session ends.

1. Responsive Media Queries:

A "Mobile-First" approach was implemented using @media rules. The grid columns automatically adjust from 3 columns (desktop) to 1 column (mobile) based on screen width.

2. Interactive Modals:

jQuery's .click() and .fadeIn() functions were used to create the "Quick Details" view for project showcases, ensuring an interactive and smooth user experience.

4.3 Coding Standards

- **Modularity:** CSS and JavaScript files are kept separate from the HTML to ensure a clean workspace.
- **Comments:** Descriptive comments were added throughout the code to explain the purpose of various functions for future updates.

5.SYSTEM TESTING AND RESULTS

System testing ensures that the portfolio functions correctly across different devices and browsers. This phase verifies the theme logic, responsiveness, and interactive elements.

5.1 Test Plan and Execution

Testing was conducted manually using browser developer tools to verify that visual changes matched the underlying code logic.

5.2 Test Cases and Results Table

5.3 Result Analysis

The testing confirms the application is stable and ready for deployment:

- **UI/UX:** CSS variables update instantly without flickering.
- **Logic:** jQuery and localStorage manage theme states and modals without errors.
- **Compatibility:** The site maintains a 100% layout pass rate on Chrome, Edge, and mobile browsers.

6. FUTURE SCOPE AND CONCLUSION

6.1 Future Scope

While the current version of the Academic Web Portfolio is fully functional, it serves as a foundation for more advanced features. Future enhancements could include:

- **Backend Integration:** Implementing a database (like MongoDB or Firebase) to allow for dynamic content updates through an Admin Dashboard without editing the source code.
- **API Integration:** Connecting a functional Contact Form using services like EmailJS or a custom Node.js backend to receive inquiries directly.
- **Blog/Journal Module:** Adding a section for technical writing and daily learning logs to improve SEO and professional engagement.
- **Language Localization:** Adding a multi-language toggle (e.g., English, Hindi, Marathi) to reach a broader audience.

6.2 Conclusion

The development of the Academic Web Portfolio successfully bridges the gap between theoretical knowledge and practical application. By utilizing HTML5, CSS3, and JavaScript, the project effectively demonstrates:

- **Responsive Design:** The importance of maintaining a high-quality user experience in a mobile-centric world.

- **State Management:** The utility of using LocalStorage for enhanced User Experience (UX) by remembering user preferences.
- **Interactive Elements:** The power of jQuery in creating smooth, interactive interface elements like modals and theme toggles.

In conclusion, this project fulfills all the academic requirements of the FYIT curriculum. It stands as a professional digital identity that not only showcases technical skills and certifications but also reflects a commitment to modern web standards and clean coding practices.

7. REFERENCES

The development of the Academic Web Portfolio was supported by official documentation and online learning platforms. The following resources were used for technical guidance:

7.1 Official Documentation

- **MDN Web Docs (Mozilla Developer Network):** Used for in-depth documentation on HTML5 semantic elements, CSS Grid, and Flexbox layouts.
- **jQuery API Documentation:** Referenced for implementing event handlers, `.fadeIn()` animations, and DOM manipulation logic.
- **W3Schools:** Utilized for quick references on CSS media queries and responsive web design best practices.

7.2 Tools & Libraries

- **Google Fonts API:** Used for integrating high-quality typography to enhance the UI/UX.
- **FontAwesome:** Provided the scalable vector icons for the social media links and the theme-toggle (Sun/Moon) switch.

7.3 Certification Sources

- **The Forage:** Providing the simulation platforms for Deloitte and JPMorganChase virtual experience programs referenced in the portfolio.
- **Coursera / Udemy:** Sources for technical certifications in web development fundamentals.

8. GLOSSARY

- **API (Application Programming Interface):** A set of rules that allows different software entities to communicate with each other.
- **CSS Grid:** A two-dimensional layout system for the web that allows developers to create complex layouts more easily.
- **DOM (Document Object Model):** A programming interface for web documents that represents the page so that programs can change the document structure, style, and content.
- **Flexbox:** A one-dimensional layout method for arranging items in rows or columns, making it easier to design flexible responsive layout structures.
- **HTML5 (Hypertext Markup Language):** The standard markup language used to create the structure of web pages.
- **jQuery:** A fast, small, and feature-rich JavaScript library designed to simplify HTML DOM tree traversal and manipulation.
- **LocalStorage:** A property that allows JavaScript sites and apps to save key-value pairs in a web browser with no expiration date.
- **Media Queries:** A feature of CSS that allows content to adapt to different screen resolutions (e.g., mobile vs. desktop).
- **Modal:** A graphical control element in the form of a small window that appears on top of the main page content.

- **Responsive Web Design (RWD):** An approach to web design that makes web pages render well on a variety of devices and window or screen sizes.
- **UI (User Interface):** The visual part of the website that the user interacts with (buttons, colors, fonts).
- **UX (User Experience):** The overall experience a person has as they interact with the website, focusing on how easy and pleasant it is to use.