Week 3 Documentation – JavaScript Interactive Portfolio Website

1. Project Overview

This project is an Interactive Portfolio Website developed as part of Week 3 of the internship. The main objective is to use JavaScript to convert a static website into a dynamic and interactive application.

2. Project Goals and Objectives

1. Understand JavaScript fundamentals
2. Implement DOM manipulation and event handling
3. Build interactive features for better user experience
4. Validate user input using JavaScript
5. Store user preferences using Local Storage

3. Setup Instructions

1. Download or clone the project repository
2. Ensure all files are present (index.html, style.css, script.js, images folder)
3. Open index.html in any modern web browser
4. No additional installation or server setup is required

4. Code Structure and File Organization

The project follows a clean and modular file structure to maintain readability and scalability.

1. index.html – Contains the structure and layout of the website
2. style.css – Handles styling and dark/light mode appearance
3. script.js – Contains all JavaScript logic and interactivity
4. images/ – Stores image assets used in the slider

5. JavaScript Features Implemented

1. Dark/Light Mode Toggle using classList and Local Storage
2. Show/Hide content functionality using DOM manipulation
3. Image slider using arrays and index-based navigation
4. Dynamic To-Do List with add and remove operations
5. Form validation with real-time error and success messages

6. Form Validation Logic

JavaScript is used to validate the contact form before submission. The email field is checked for valid format, and the message field is validated for minimum length. If validation fails, appropriate error messages are displayed. Successful validation shows a confirmation message without reloading the page.

7. Interactive Elements Explanation

1. Buttons trigger JavaScript functions using event listeners
2. DOM elements are dynamically created and removed
3. Theme preference is stored and retrieved using Local Storage

8. Technical Details

Arrays are used to store image paths for the slider. Functions are designed to be reusable and modular. Conditional logic ensures proper validation and user feedback. The architecture follows a simple client-side scripting model.

9. Testing and Validation

1. Tested dark/light mode toggle functionality
2. Verified form validation with valid and invalid inputs
3. Checked image slider navigation
4. Tested to-do list add and delete operations

10. Visual Documentation

Screenshots of the website showing dark mode, image slider, to-do list, and form validation can be added in the repository or appended to this document if required.

11. Quality Standards Checklist

1. Clear project overview and objectives
2. Well-organized code structure
3. Complete documentation and explanations
4. Fully functional interactive features
5. Testing evidence included