

Week 3 Task 3 Project Documentation

Week 3 – Interactive Portfolio Website

Project Documentation (Task 3)

1. Project Overview and Objectives

1.1 Project Description

This project extends the Week 2 styled portfolio by adding **JavaScript interactivity**. It was created as **Week 3 – Task 3** for The Developers Arena Web Development Internship program.

The portfolio now includes **fully functional interactive features**: dark/light mode toggle, profile views counter, and a complete **Interactive Task Manager** (add, delete, mark complete, clear all).

1.2 Objectives

The main objectives of this Week 3 project are:

1. Implement **JavaScript fundamentals**: variables, functions, DOM manipulation, event handling
2. Create **interactive UI components**:
 - o Dark/Light mode toggle with localStorage persistence
 - o Profile views counter (demo with increment/decrement)
 - o **Interactive Task Manager** (core Week 3 project requirement)
3. Demonstrate **event handling**: click events, keyboard events (Enter key)
4. Maintain **Week 1 HTML + Week 2 CSS** structure while adding functionality
5. Prepare for **responsive design** (Week 4)

2. Setup and Installation Instructions

2.1 Prerequisites

- A text editor (VS Code recommended for JS debugging)
- Modern web browser with Developer Console (F12)
- **Profile image** (Rittik.jpg) in images/ folder

2.2 Step-by-Step Installation

Local Setup

text

week3-portfolio/

```
|--- index.html      # HTML + CSS + JavaScript (all-in-one)  
|--- README.md  
└--- Rittik.jpg
```

1. Save the complete code as index.html
2. Add Rittik.jpg to images/ folder
3. Open in browser → **Test all interactions**

GitHub Pages

1. Push to GitHub repository
2. Enable GitHub Pages (main branch)
3. Live URL: <https://username.github.io/repo/>

2.3 Testing Checklist

- Dark/Light mode toggle works + persists
- Profile counter increments/decrements
- Task Manager: Add, Delete, Complete, Clear All
- Enter key adds tasks
- Console shows no JavaScript errors (F12)

3. Code Structure Explanation

3.1 File Structure

Single file (index.html with embedded <style> + <script>):

text

index.html ← HTML (Week 1) + CSS (Week 2) + JS (Week 3)

```
└--- Rittik.jpg
```

3.2 JavaScript Organization (Inside <script>)

1. Dark/Light Mode Toggle:

javascript

```
const modeToggle = document.getElementById('modeToggle');
```

```
const isDarkPref = localStorage.getItem('theme') === 'dark';
```

```
modeToggle.addEventListener('click', () => {
```

```
    document.body.classList.toggle('dark');
```

```
    localStorage.setItem('theme', darkEnabled ? 'dark' : 'light');
```

```
});
```

2. Profile Views Counter:

javascript

```
let counter = 0;
```

```
function updateCounter() { counterSpan.textContent = counter; }
```

```
incrementBtn.addEventListener('click', () => { counter++; updateCounter(); });
```

3. Interactive Task Manager (Main Project):

javascript

```
function createTaskItem(text) {
```

```
    // DOM creation: li, span, buttons
```

```
}
```

```
function addTask() {
```

```
    // Input validation + task creation
```

```
}
```

```
addTaskBtn.addEventListener('click', addTask);
```

```
taskInput.addEventListener('keydown', (e) => {
```

```
    if (e.key === 'Enter') addTask();
```

```
});
```

3.3 Key JavaScript Concepts Demonstrated

Concept	Implementation	Example
Variables	let, const	let counter = 0;
Functions	Reusable code	function addTask() {}
DOM Selection	getElementById, querySelector	document.getElementById('taskInput')
Event Handling	addEventListener	'click', 'keydown'
DOM Manipulation	createElement, appendChild	Dynamic task list items
localStorage	Persistence	Theme preference saving
Conditional Logic if/else		Input validation, counter limits

4. Screenshots of Working Application

4.1 Interactive Features Demonstrated

A. Dark/Light Mode Toggle:

text

Light Mode → [Click] → Dark Mode (persists on reload)

B. Profile Views Counter:

text

0 → [+] → 1 → 2 → [+] → 5 → [-] → 4

C. Interactive Task Manager (Core Project):

text

Input: "Learn JavaScript" → [Add Task] → ✓ Task appears

[Done] → Task strikethrough ✓

[Delete] → Task removed ✓

[Clear Completed] → Removes done tasks ✓

[Clear All] → Empty list ✓

4.2 How to Capture Interactive Screenshots

1. Chrome DevTools (F12):

- Toggle Device Toolbar for mobile/desktop views
- Record GIF: Ctrl+Shift+P → "Record screencast"

2. Screen Recording (Recommended):

- Record task creation → completion → deletion
- Show dark mode toggle + counter

3. Static Screenshots:

text

week3-dark-mode.png

week3-task-manager.png

week3-counter.png

week3-mobile.png

4.3 Expected Behavior

Action	Expected Result
--------	-----------------

Click 🌙 Dark Mode Entire site → dark theme

Reload page Dark mode persists

+ button Counter increases

Enter key (task input) Adds new task

"Done" button Task gets strikethrough

"Delete" button Task removed instantly

"Clear All" Task list empty

5. Explanation of How Technical Requirements Were Met

5.1 Requirement 1: JavaScript Interactivity Added

- **Three interactive components** fully functional
- **Pure vanilla JavaScript** (no libraries)
- **Event-driven architecture** demonstrated

5.2 Requirement 2: Interactive Task Manager (Main Project)

Features Implemented:

text

- Add tasks (click + Enter key)
- Delete individual tasks
- Mark tasks complete (strikethrough)
- Clear all completed tasks
- Clear all tasks
- Real-time empty state ("No tasks yet...")
- Input validation (no empty tasks)
- Responsive task list (scrollable)

5.3 Requirement 3: DOM Manipulation Mastery

javascript

```
// Dynamic element creation  
  
const li = document.createElement('li');  
  
const span = document.createElement('span');  
  
li.appendChild(span);  
  
tasksList.appendChild(li);
```

- **300+ lines** of production-ready JavaScript
- **Event delegation** patterns
- **Dynamic UI updates**

5.4 Requirement 4: Event Handling Implemented

Event Type Elements Actions

click Buttons (6+) Toggle, add, delete, clear

keydown Task input Enter → add task

Page load localStorage Restore dark mode

5.5 Requirement 5: Week 1-2 Foundation Preserved

- **Same HTML structure** (Week 1)
- **Same CSS styling** (Week 2)
- **CSS variables respected** by JavaScript (dark mode)
- **Progressive enhancement** achieved

5.6 Bonus JavaScript Features

text

- localStorage persistence (dark mode)
 - Input validation + error messages
 - Keyboard accessibility (Enter key)
 - Real-time UI updates (no page reload)
 - Mobile-friendly touch targets
 - Console debugging ready (no errors)
 - Cross-browser compatible
-

6. GitHub Repository Structure

text

week3-portfolio/

```
|--- index.html      # Complete interactive portfolio  
|--- README.md      # This documentation  
    └ Rittik.jpg     # Profile image
```

File size: ~35KB (includes all HTML/CSS/JS)

7. Browser Console Debugging

Open **F12 → Console** to verify:

text

- No JavaScript errors
- All event listeners attached

localStorage working

Responsive breakpoints active

Week 3 Success: Fully interactive portfolio with **professional Task Manager**, dark mode, counter, and smooth UX. Ready for responsive design (Week 4)! 
