portfolio website-SDLC

My Software Development life cycle i's using SCRUM which is a type of iterative Model

1. Planning

Objective: Define project scope, goals, and requirements.

Requirements

- **home page**: contains an overview about me and buttons leads to sub pages like (blog-posts,projects,contat me).
- **users page**: this for admin/login or visitor
- **Blog Posts page**: List of articles (titles, summaries, "Read More" links)
- **Projects page**: Cards with project names, images, tech stack, GitHub links
- **contact me page**: Form (name, email, message), social media links, email button

Analysis

- **Target Audience**: Determine if the site is for recruiters, clients, or peers.
- **Tools**: Choose technologies (e.g., HTML/CSS/JS, React, or a static site generator like Gatsby).
- **Timeline**: Set deadlines for design , development, and launch.

2. Design

Objective: Create a visual UI for the vebsite

UI/UX Design:

- Choose color schemes, typography, and branding.
- Ensure intuitive navigation (e.g., sticky header, clear CTAs).
- Prototyping: Interactive mockups to test user flow.
- Feedback Loop: Review with peers or mentors for improvements.

3. Development

Objective: Build the website based on design specs. Implementation:

Frontend:

- Code responsive layouts (HTML/CSS, Flexbox/Grid).
- Add interactivity (JavaScript/React animations).

Backend

- connect fortend to servers that contains database for users and other website components

4. Testing

Objective: Ensure functionality, usability, and performance. Implementation:

- Unit Testing: Check individual components (e.g., form validation).
- Cross-Browser/Device Testing: Verify compatibility (Chrome, Safari, mobile).
- Performance Testing: Optimize images, minimize CSS/JS (Lighthouse audit).
- User Testing: Collect feedback on navigation and load speed.
- Bug Fixing: Resolve issues (e.g., broken links, 404 errors).

5. Maintenance

Objective: Keep the site updated and secure. Implementation:

- Content Updates: Add new projects or skills periodically.
- Performance Monitoring: Use tools like Google Analytics.
- Security Patches: Update dependencies (e.g., npm packages).
- Feedback Integration: Improve based on user suggestions.