

# 1. Problem Statement & Objectives

| Category          | Details   |
|-------------------|---|
| Problem Statement | Developers need a professional portfolio website to showcase skills, projects, and contact details.   |
| Objectives        | <ul style="list-style-type: none"><li>- Build a responsive and modern portfolio website.</li><li>- Ensure intuitive UI/UX.</li><li>- Include projects, skills, and a contact form.</li><li>- Use Angular, Bootstrap, HTML, CSS, JavaScript.</li></ul> |

# 2. Use Case Diagram & Descriptions

| Actors                            | Description  |
|-----------------------------------|--|
| Visitor (User)                    | Views portfolio, projects, and fills contact form.     |
| Developer (Admin) (if applicable) | Updates portfolio details (if CMS-like feature added). |

| Key Interactions        | Description   |
|-------------------------|---|
| View Portfolio          | User browses skills, projects, and profile.                     |
| Project Interaction     | User clicks project → Views details → Redirects to GitHub/demo. |
| Contact Form Submission | User fills out form → System validates & sends email.           |

### 3. Functional & Non-Functional Requirements

| Category       | Requirements  |
|----------------|---|
| Functional     | <div><div>✔ Display skills, projects, and bio.</div><div>✔ Implement contact form.</div><div>✔ Smooth navigation between sections.</div></div>  |
| Non-Functional | <div><div>⚡ Responsive Design (mobile/tablet/desktop).</div><div>⚡ Fast Performance (optimized assets).</div><div>⚡ SEO-friendly (search visibility).</div><div>⚡ Accessible UI (clear fonts &amp; contrast).</div></div> |

### 4. Database Design & Data Modeling *(Only if needed)*

| Entity                  | Attributes  |
|-------------------------|---|
| Projects                | ID, Title, Description, Technologies, Image, Link |
| Skills                  | ID, Skill Name, Proficiency Level                 |
| Messages (Contact Form) | ID, Name, Email, Message, Date                    |

### 5. Data Flow & System Behavior

| Diagram Type                           | Description  |
|--|--|
| DFD (Data Flow Diagram)                | Shows data flow between components (portfolio content retrieval & contact form submission).  |
| Sequence Diagram                       | Represents interactions like user loading portfolio and submitting a contact form.   |
| Activity Diagram                       | <p>📌 <b>Browsing Portfolio:</b> User clicks project → Details appear → Link opens GitHub/demo.</p> <p>📌 <b>Contact Form Submission:</b> User enters details → Clicks submit → System validates → Displays success message.</p> |
| State Diagram                          | <p>- Portfolio Page States: Initial → Data Loading → Display Content.</p> <p>- Contact Form States: Empty → User Input → Validation → Success/Error.</p>   |
| Class Diagram (For Angular Components) | <p>- AppComponent → Main container.</p> <p>- HeaderComponent → Navigation.</p> <p>- ProjectComponent → Project showcase.</p> <p>- ContactComponent → Form handling.</p>  |

## 6. UI/UX Design & Prototyping

| Category             | Details   |
|----------------------|---|
| Wireframes & Mockups | <ul style="list-style-type: none"><li>📌 Header: Developer name, navigation menu.</li><li>📌 Hero Section: Profile pic, short bio, CTA button.</li><li>📌 Projects Section: Grid of projects with images &amp; descriptions.</li><li>📌 Skills Section: List of technologies.</li><li>📌 Contact Form: Name, email, message.</li></ul> |
| UI/UX Guidelines     | <ul style="list-style-type: none"><li>🎨 Color Scheme: Modern and professional.</li><li>🔤 Typography: Clear fonts (e.g., Montserrat).</li><li>📱 Responsive Design: Bootstrap grid layout.</li><li>🖱️ User Interaction: Smooth animations.</li></ul>  |

## 7. System Deployment & Integration

| Category           | Details   |
|--------------------|---|
| Technology Stack   | <p><b>Frontend:</b> Angular, HTML, CSS, JavaScript, Bootstrap.</p> <p><b>Backend (if needed):</b> Node.js/Express or Firebase.</p> <p><b>Hosting:</b> GitHub Pages, Netlify, Vercel, Firebase Hosting.</p>        |
| Deployment Diagram | 🔗 User requests website → Server delivers Angular static files → Browser renders content.   |
| Component Diagram  | <ul style="list-style-type: none"><li>- HomeComponent (Landing Page)</li><li>- ProjectsComponent (Displays projects)</li><li>- SkillsComponent (Lists skills)</li><li>- ContactComponent (Handles form)</li></ul> |

8. Additional Deliverables (If Applicable)

| Category                       | Details  |
|--------------------------------|--|
| API Documentation (If dynamic) | <ul style="list-style-type: none"><li>📌 GET /projects → Fetch all projects.</li><li>📌 POST /contact → Submit contact form.</li></ul>   |
| Testing & Validation           | <ul style="list-style-type: none"><li>✅ Unit Tests: Verify Angular components.</li><li>✅ Integration Tests: Ensure form submission works.</li><li>✅ User Testing: Test responsiveness &amp; accessibility.</li></ul> |
| Deployment Strategy            | <ul style="list-style-type: none"><li>🚀 Hosting on Netlify/Vercel for auto-deployment.</li><li>🚀 CI/CD Pipelines for auto-updates from GitHub.</li></ul>   |