**1. Planning**

**Objective:** Determine the project scope, purpose, and requirements.

* **Scope:** The website will allow users to browse blog posts, view project details, and submit a contact form.
* **Requirements:** Based on the diagrams:
  + Database with users, blog\_posts, and contact\_forms tables.
  + Functionalities like blog post listing, project viewing, and contact form submission.
  + Integration with email services to handle contact forms.
* **Stakeholders:** You (developer/owner) and website users.
* **Tools/Technologies:** Decide to use HTML/CSS, a backend framework (e.g., Django/Flask), and a database (e.g., PostgreSQL/MySQL).

**2. Design**

**Objective:** Design the system architecture, user interface, and database schema.

* **ERD Design:** The Entity-Relationship Diagram (ERD) shows tables (users, blog\_posts, and contact\_forms):
  + blog\_posts references users via a foreign key (user\_id).
  + contact\_forms captures user-submitted messages.
* **Website Interaction Diagram:** Defines system processes:
  + Fetching blog posts from the database.
  + Displaying project details.
  + Sending contact form emails using an Email Service.
* **UI/UX Mockups:** Create wireframes for the website with sections like Home, Blog, Projects, and Contact.

**3. Development**

**Objective:** Implement the planned design using appropriate technologies.

* **Backend:**
  + Set up database models for users, blog\_posts, and contact\_forms.
  + Create API endpoints to fetch blog posts, project details, and handle contact form submissions.
* **Frontend:**
  + Develop HTML/CSS templates for the blog list, project details, and contact form.
  + Use JavaScript (or frameworks like React/Vue) for dynamic rendering.
* **Integration:**
  + Connect the contact form to the email service to send emails.
  + Link the blog system to fetch blog data.

**4. Testing**

**Objective:** Ensure the website is functional, secure, and bug-free.

* **Unit Testing:**
  + Test backend routes for fetching blog posts and handling contact form submissions.
* **Integration Testing:**
  + Validate integration between the website and the email service.
* **UI Testing:**
  + Ensure blog posts and project details display correctly.
  + Test the form for proper validation and submission.
* **User Acceptance Testing (UAT):**
  + Verify user workflows, such as browsing blogs, viewing projects, and submitting forms.

**5. Maintenance**

**Objective:** Ensure long-term stability and scalability of the website.

* **Monitoring:** Implement logging for errors (e.g., failed form submissions).
* **Updates:**
  + Update the blog system to allow adding/editing posts.
  + Add user authentication if needed for administrators.
* **Optimization:**
  + Optimize database queries and improve page load performance.
* **Support:** Collect user feedback to make further improvements.