45-Day Plan for Developing Angular Web App

Phase 1: Landing Page (Days 1-15)

Day 1-2: Project Setup

- Set up the Angular project using the Angular CLI.
- Configure necessary dependencies (e.g., Angular Material, Bootstrap, etc.).

Day 3-4: Landing Page Design

- Review the existing mockups for the landing page.
- Implement the landing page layout and design.

Day 5-8: Landing Page Development - Part 1

- Create the basic structure of the landing page.
- Add sections like header, hero section, about us, and contact.

Day 9-11: Landing Page Development - Part 2

- Add features such as call-to-action buttons, subscription forms, etc.
- Incorporate animations or transitions to enhance user experience.

Day 12-15: Landing Page Finalisation

- Review and optimise the landing page for responsiveness and performance.
- Gather feedback from stakeholders and make necessary adjustments.

Day 16-17: Projects and Events

- Create a page to showcase projects and events together with blog posts.
- Implement features to list projects, events, and blog posts based on categories and tags.

Phase 2: Blog Site and CRUD Functionality (Days 16-35)

Day 18-20: Blog Listing and Detail Page

- Create a page to display a list of blog posts.
- Implement a blog post detail page that shows the full content of a blog post.

Day 21-26: CRUD Functionality for Blog Posts

- Implement Create, Read, Update, and Delete (CRUD) operations for blog posts.

- Design a user-friendly interface for managing blog posts (e.g., a dashboard for admins).

Day 27-28: Blog Categories and Tags

- Add categories and tags functionality to classify blog posts.
- Allow users to filter blog posts based on categories and tags.

Day 29-36: Payment Gateway, Donations, and Admin Dashboard

- Implement the donation feature, allowing users to make donations securely.
- Create an admin dashboard to track and manage donation transactions [if third party API lets us].

Phase 3: Final Touches, Testing, and Deployment (Days 36-45)

Day 36-40: Testing and Bug Fixing

- Conduct thorough testing of the entire web app.
- Identify and fix any bugs or issues that arise during testing.

Day 41-43: Deployment to Staging Environment

- Prepare the app for deployment to a staging environment.
- Conduct final checks and ensure everything is ready for production.

Day 44: Deployment to Production

- Deploy the web app to the production environment.
- Make any final adjustments and improvements based on feedback.

Day 45: Final Review and Documentation

- Conduct a final review of the web app's functionality and design.
- Document the project, including code documentation and user guides.

Backend development plan

Day 1 -2:

Project Setup:

- Set up a new Node.js project using a package manager like npm or yarn.
- Initialise a version control system (e.g., Git) to track changes.

Database Configuration:

- Install the necessary database driver and libraries for Node.js.

Day 3-4 :

User Authentication:

- Implement user authentication and authorization using popular strategies like JWT (JSON Web Tokens).
- Set up routes for user registration, login, and logout.

Day 5-6

User CRUD Operations:

- Create API endpoints for user management, including creating, reading, updating, and deleting users.
- Implement password encryption for user security.

Day 8

Blog Management:

- Design the database schema for blogs and their related entities (e.g., categories, tags).
- Create API endpoints for creating, reading, updating, and deleting blog posts.
- Implement pagination and sorting for listing blogs.

Day 9

Project Management:

- Define the database schema for projects and any associated data.
- Set up API endpoints to handle project CRUD operations.

Day 10

Event Handling:

- Design the database schema for events and related entities.
- Create API endpoints to manage events, such as creating, updating, and deleting events.

Day 11

Product Management:

- Design the database schema for products and any related data.
- Set up API endpoints for managing products (e.g., create, read, update, delete).

Day 12-13

Validation and Error Handling:

- Implement validation for incoming requests to ensure data integrity.
- Set up error handling middleware to provide meaningful error responses.

Day 14 File Uploads:

- If required, implement file upload functionality for images or other media related to blogs, projects, or products.

Day 15-17 Middleware and Security:

Add middleware for logging, security headers, and other cross-cutting concerns.

Secure the API using best practices to prevent common security vulnerabilities.

Day 18-19 Testing:

Write unit tests and integration tests to ensure the reliability of the backend.

Use testing frameworks like Mocha or Jest.

Day 20:

Documentation:

- Generate API documentation using tools like Swagger or Postman.
- Document code where necessary for better maintainability.

Day 21-25

Deployment:

- Deploy the Node.js backend and set up necessary environment configurations.

Monitoring and Scaling:

Implement monitoring and logging to track the application's performance.