Kanban Board Phase 1 for OPT Volunteers Management Timeline.
(Aug 4, 2024 to Sept 13, 2024)

Week 1 (August 4 - August 9)

Stage 1: Planning and Design
Objective: Lay the groundwork for the application by understanding requirements and creating initial designs.

August 5-6: Requirements Document
Tasks:

- Identify user roles (Admin, Project Manager, Volunteer) and their permissions.
- Define functional requirements (e.g., project creation, task assignment, progress tracking).
- Define non-functional requirements (e.g., performance, security, scalability).
- Detail user stories and use cases.

August 7: Database Schema
Tasks:

- Design MongoDB schema with collections: Users, Projects, Tasks, Comments, Notifications.
- Define relationships between collections.
- Draft initial schema models.

August 8: API Specifications
Tasks:

- Outline RESTful API endpoints for CRUD operations.
- Define request and response formats for each endpoint.
- Specify authentication and authorization mechanisms.

August 9: Prototype Designs
Tasks:

- Create wireframes/mockups for each user role's interface using Figma.
- Design Admin Dashboard for user and project management.
- Design Project Manager Dashboard for project and task management.

- Design Volunteer Dashboard for task list and progress updates.
- Gather feedback and iterate on designs.

Week 2 (August 12 - August 16)

Stage 2: Setup and Basic Implementation
Objective: Establish the development environment and start basic backend and frontend development.

August 12: Development Environment Setup
Tasks:

- Set up local development environment (Node.js, React).
- Configure cloud environment on Google Cloud Platform (GCP) (Compute Engine, Cloud Storage, Cloud SQL).
- Install necessary tools and dependencies.

August 13: Version Control Setup
Tasks:

- Initialize Git repositories for backend and frontend.
- Define branch strategies (e.g., main, develop, feature branches).
- Set up CI/CD pipelines using Google Cloud Build.

August 14: Backend Framework Setup
Tasks:

- Set up initial Node.js project structure.
- Create initial API endpoints for user authentication and project creation.
- Test endpoints using Postman or similar tools.

August 15-16: Frontend Framework Setup
Tasks:

- Set up initial React project structure with routing and state management.
- Implement initial components: Login, Dashboard, Project list.
- Connect frontend with backend APIs for authentication and project data.

Week 3 (August 19 - August 23)

Stage 2: Setup and Basic Implementation (continued)
Objective: Continue backend and frontend development, focusing on database integration and initial components.

August 19: Database Integration
Tasks:

- Connect the application with MongoDB (hosted on GCP).
- Test basic CRUD operations for Users and Projects collections.
- Ensure database connection stability and security.

August 20-21: Initial Backend Endpoints
Tasks:

- Develop API endpoints for user authentication (registration, login).
- Develop API endpoints for project creation and management.
- Implement basic business logic and error handling.

August 22-23: Initial Frontend Components
Tasks:

- Implement Login component with authentication flow.
- Implement Dashboard component to display project list.
- Fetch and display data from backend APIs.
- Ensure responsive design and user-friendly UI.

Week 4 (August 26 - August 30)

Stage 3: Core Development and Testing
Objective: Develop core functionalities, integrate frontend with backend, and perform thorough testing.

August 26-27: Core API Development
Tasks:

- Develop API endpoints for task management (create, update, delete tasks).
- Develop API endpoints for comments and notifications.
- Implement advanced business logic (task assignment, status updates).

August 28-29: Frontend Components Development
Tasks:

- Implement Task board component to display tasks.
- Implement Task details component for viewing and editing tasks.
- Implement Comments component for task discussions.
- Implement Notifications component for updates and alerts.

August 30: Authentication Implementation
Tasks:

- Set up user authentication and authorization using JWT.
- Define role-based access control (RBAC) for different user roles.
- Secure API endpoints with authentication middleware.

Week 5 (September 2 - September 6)

Stage 3: Core Development and Testing (continued)
Objective: Continue core development, testing, and prepare for deployment.

September 2-3: Initial Deployment
Tasks:

- Deploy the application on GCP for internal testing.
- Set up necessary GCP services (Compute Engine, Cloud Storage, Cloud SQL, Cloud CDN).
- Configure environment variables and deployment scripts.

September 4-5: Testing
Tasks:

- Conduct unit testing for backend and frontend components.
- Conduct integration testing to ensure seamless interaction between components.
- Conduct load testing to evaluate application performance under stress.
- Identify and fix any issues or bugs discovered during testing.

September 6: Security Enhancements Preparation
Tasks:

- Prepare for implementing security measures (secure communication, data encryption).
- Review security best practices and standards.
- Plan for secure API practices (HTTP to HTTPS).

Week 6 (September 9 - September 13)

Stage 4: Finalization and Launch
Objective: Finalize the application with security enhancements, user feedback integration, and prepare for launch.

September 9-10: Security Enhancements
Tasks:

- Implement secure communication using HTTPS.
- Implement data encryption for sensitive information.
- Secure API endpoints with authentication and authorization.

September 11: User Acceptance Testing (UAT)
Tasks:

- Conduct testing with actual users (Admin, Project Managers, Volunteers).
- Gather feedback on usability, functionality, and performance.
- Address any issues or improvements suggested by users.

September 12: Performance Optimization
Tasks:

- Optimize application performance for speed and efficiency.
- Implement caching strategies for frequently accessed data.
- Optimize database queries and API responses.

September 13: Final Deployment and Launch
Tasks:

- Deploy the optimized application on GCP with scalability options enabled.
- Provide comprehensive documentation and training materials for all user types.
- Officially launch the application and monitor its performance.
- Collect user feedback for further improvements.