1. Project Overview

1.1. Objectives

Develop an open-source web application using Laravel where users can create accounts, manage consultation and medication plans, and view insights and advertisements.

1.2. Scope

- User registration and authentication
- User dashboard with consultation and medication management
- Admin panel for managing hospitals, departments, advertisements, and users
- Notification system for next visit reminders
- Responsive design

2. Technical Stack

- Backend Framework: Laravel
- Frontend Framework: Vue.js and Laravel Blade engine
- Database: MySQL
- Authentication: Laravel Passport / Sanctum
- Notifications: Laravel Notifications
- File Storage: Laravel File storage / Local Storage
- Deployment: Docker, AWS / DigitalOcean
- Version Control: Git
- Version Control tool: Bitbucket

3. System Architecture

3.1. Structure

Frontend->API requests->Backend-> Database

3.2. Components

- Frontend: Handles the user interface and interactions.
- Backend: Manages application logic, data processing, and API endpoints.
- Database: Stores user data, consultation plans, medication plans, etc.
- Notifications: Manages notifications for user reminders.
- File Storage: Stores uploaded files (prescriptions, test results).

4. User Stories and Requirements

4.1. User Registration and Authentication

- User Story: As a user, I want to register and log in so that I can access the application.
- Requirements:
  - Registration form with email, password, and validation
  - Login form with email and password
  - Password reset functionality
  - Email verification

4.2. User Dashboard

- User Story: As a user, I want to view my dashboard where I can manage my consultation and medication plans.
- Requirements:
  - Navbar with navigation options
  - Main section with options to create Consultation Plan and Medication Plan
  - Shortcuts for running medicines, running consultations, next visit

    - Advertisement section
    - Insights and data display section
    - Google ad integration at the bottom

4.3. Consultation Plan Management

- User Story: As a user, I want to create and manage my consultation plans.
- Requirements:
    - Form to create a consultation plan with hospital selection, department, date, file upload, next visit date, and reminder settings
    - Validation to prevent duplicate hospital entries

4.4. Medication Plan Management

- User Story: As a user, I want to create and manage my medication plans.
- Requirements:
    - Form to create a medication plan with medication name, dosage, timing (morning/day/night), and quantity

4.5. Notifications

- User Story: As a user, I want to receive reminders for my next visit.
- Requirements:
    - Notification system to remind users before their next visit date

5. Database Schema

5.1. Tables

- Users
    - id
    - name
    - email
    - password
    - email_verified_at
    - created_at
    - updated_at
- Hospitals
    - id
    - name
    - location
    - created_at
    - updated_at
- Departments
    - id
    - name
    - created_at
    - updated_at
- ConsultationPlans
    - id
    - user_id
    - hospital_id
    - department_id
    - consultation_date
    - prescription_file
    - test_files
    - next_visit_date
    - reminder_time
    - created_at
    - updated_at
- MedicationPlans
    - id
    - user_id

    - plan_name
    - created_at
    - updated_at
- Medications
    - id
    - medication_plan_id
    - medication_name
    - dosage
    - timing (enum: morning, day, night)
    - quantity
    - created_at
    - updated_at
- Advertisements
    - id
    - sponsor_name
    - start_date
    - end_date
    - status
    - redirect_link
    - sorting
    - timestamp
    - image
    - created_at
    - updated_at

6. API Endpoints Format

6.1. User Authentication

- POST `/api/register`
- POST `/api/login`
- POST `/api/password-reset`
- POST `/api/email-verify`

6.2. Consultation Plan Management

- GET `/api/consultation-plans`
- POST `/api/consultation-plans`
- GET `/api/consultation-plans/{id}`
- PUT `/api/consultation-plans/{id}`
- DELETE `/api/consultation-plans/{id}`

6.3. Medication Plan Management

- GET `/api/medication-plans`
- POST `/api/medication-plans`
- GET `/api/medication-plans/{id}`
- PUT `/api/medication-plans/{id}`
- DELETE `/api/medication-plans/{id}`

6.4. Advertisement Management (Admin)

- GET `/api/advertisements`
- POST `/api/advertisements`
- GET `/api/advertisements/{id}`
- PUT `/api/advertisements/{id}`
- DELETE `/api/advertisements/{id}`

7. User Interface

- Figma will be developed by UI/UX designer
- Template will be developed by web designer with all required input and action functionalities

8. Admin Panel

8.1. Dashboard

- **Overview of site statistics and user activities.**

8.2. Hospital Management

- Add, edit, and delete hospitals.

8.3. Department Management

- Add, edit, and delete departments.

8.4. Advertisement Management

- Create and manage advertisements.

8.5. User Management

- View list of users, block/unblock users.

9. Security

9.1. Data Protection

- Use encryption for sensitive data (e.g., passwords, user information).

9.2. Authentication and Authorization

- Implement OAuth2 or JWT for secure authentication.
- Role-based access control for admin functionalities.

9.3. Input Validation

- Validate all user inputs to prevent SQL injection and XSS attacks.

10. Performance Considerations

10.1. Caching

- **Implement caching strategies for frequently accessed data.**

10.2. Load Testing

- **Perform load testing to ensure the application can handle expected traffic.**

12. Deployment

12.1. Environment Setup

- **Set up development, staging, and production environments.**

12.2. Continuous Integration/Continuous Deployment (CI/CD)

- **Implement CI/CD pipeline for automated testing and deployment in bitbucket.**

13.1 Conclusion

This project aims to develop an open-source web application using Laravel, providing users with a comprehensive dashboard to manage consultation and medication plans. The application will include user registration, hospital and department management, notifications for next visits, and advertisement integration. The admin panel will offer functionalities for managing hospitals, departments, advertisements, and users.