Project Design Documentation

Project Name: AVA

GitHub Repository: AVA GitHub

---

1. Introduction

1.1 Purpose

AVA is a web-based platform aimed at addressing women's health concerns through machine learning-powered tracking and diagnosis tools. Initially developed for PCOS/PCOD tracking, AVA is being expanded to cover other issues such as thyroid disorders. . By integrating frontend, backend, and machine learning components, AVA delivers a seamless and interactive healthcare experience.

1.2 Scope

The AVA platform comprises multiple health-focused functionalities:

- PCOS/PCOD Tracker *(Implemented)*
- Thyroid Health Tracker *(In Progress)*
- Menstrual Cycle Prediction
- Symptom Checker for Women's Health
- Community Support Forum
- Secure User Authentication & Data Storage
- AI-Powered Health Consultation & Insights *(Future Development)*

1.3 Target Audience

- Women seeking health insights and condition tracking
- Medical professionals and researchers studying women's health
- Healthcare startups and telemedicine platforms

---

2. Project Goals

- Empower Women with Personalized Health Insights
- Build a Secure & Scalable Health Platform
- Enhance User Engagement with an Intuitive UI/UX
- Leverage AI & Machine Learning for Accurate Predictions
- Foster a Supportive Community for Women's Health
- Ensure Future Scalability & Innovation

---

3. System Overview

3.1 Architecture

AVA follows a **modular architecture** with the following components:

- **Frontend:** React.js for dynamic UI components and Tailwind CSS for styling
- **Backend:** FastAPI for API handling and Flask for ML model integration
- **Database:** PostgreSQL for structured user data storage
- **Machine Learning:** TensorFlow and Scikit-learn for predictive models
- **Hosting & Deployment:** GitHub Pages for frontend, AWS/Heroku for backend and ML services

3.2 System Flow:

1. Users interact with the frontend to input symptoms or track health.
2. The backend processes requests, retrieving or updating data.
3. Machine learning models analyze input data and return insights.
4. Users receive visualized reports, insights, or recommendations.

---

4. Functional Requirements

4.1 User Roles

1. **Guest Users:** Access general health resources and articles.
2. **Registered Users:** Input health data, track progress, and receive personalized insights.
3. **Admin/Healthcare Professionals:** Monitor anonymized user trends and contribute expert insights.

4.2 Core Functionalities

PCOS/PCOD Tracker *(Implemented)*

- Allows users to input symptoms and medical history.
- Uses an ML model to assess PCOS/PCOD likelihood.
- Provides lifestyle and dietary recommendations.

Thyroid Health Tracker *(Upcoming)*

- Similar workflow to the PCOS tracker.
- Monitors symptoms like weight fluctuations, fatigue, and hormonal changes.

Menstrual Cycle Predictor

- Uses past cycle data to predict upcoming periods.
- Sends reminders and detects irregularities.

User Dashboard

- Provides personalized health trends.
- Features interactive charts and downloadable reports.

User Interface

- Clean, eye-catching design with intuitive navigation.
- A refreshing color palette that stands out from traditional pink/red themes.
- Engaging visuals—charts, graphs, and progress trackers for easy understanding.

Community Forum *(Planned Feature)*

- Secure platform for discussions and peer support.
- Moderated by healthcare professionals.

---

5. Non-Functional Requirements

- **Scalability:** Support for growing user base with cloud deployment.
- **Security:** OAuth 2.0 for authentication and AES encryption for data.
- **Performance:** Backend API response time under 500ms.
- **Usability:** Accessible and mobile-friendly UI.
- **Maintainability:** Modular code with automated testing and documentation.
- **Reliability:** Regular backups and failover systems in case of outages.

6. Technology Stack

- **Frontend:** React.js, Tailwind CSS
- **Backend:** FastAPI, Flask (for ML integration)
- **Database:** PostgreSQL
- **ML Model Training:** TensorFlow, Scikit-learn
- **Testing:** PyTest, Selenium
- **Deployment:** AWS, Docker (Containerization)

---

7. Development Workflow

7.1 Team Structure

- **Frontend Developer:** UI/UX design and frontend implementation
- **Backend Developer:** API development and database management
- **AI/ML Specialist:** Model training, testing, and integration

7.2 Agile Development Process

- **Sprints:** 2-week iterations
- **Task Management:** GitHub Issues and Kanban boards
- **Version Control:** Git & GitHub
- **Continuous Integration:** Automated builds and tests via GitHub Actions

---

8. Deployment Strategy

8.1 CI/CD Pipeline

1. **Code Push:** Developers commit to GitHub
2. **Automated Build:** GitHub Actions trigger build and test pipeline
3. **Deployment:**
    - Frontend → GitHub Pages
    - Backend & ML Services → AWS (Dockerized containers)

---

9. Testing Plan

9.1 Unit Testing

- **Backend API:** PyTest-based testing for core API endpoints.
- **Frontend Components:** Jest for React components.

9.2 Integration Testing

- Testing the interaction between UI, backend, and ML models.

9.3 Performance Testing

- Load testing backend API with simulated users.

9.4 Security Testing

- Conducting penetration tests to detect vulnerabilities.

---

10. Future Enhancements

- AI Chatbot for Women's Health Consultations
- Multi-Language Support for Broader Accessibility
- Telehealth Integration for Virtual Doctor Consultations

- Wearable Device Compatibility for Real-Time Health Monitoring

---

11. Project Limitations & Challenges

- Data Privacy & Security Regulations (Compliance with HIPAA/GDPR)
- Limited Computing Resources for ML Model Training
- Scalability Challenges with Growing User Base
- Integration Complexity Across Multiple Components
- Accuracy & Reliability of Health Predictions
- Development Timeline & Resource Constraints
- Cross-Platform Compatibility (Web & Mobile)
- Ensuring User Trust & Ethical AI Usage

---

12. Underlying Assumptions

- Users will have internet access and good connectivity to use the web platform.
- Users will provide accurate health data for meaningful insights.
- Users will have basic digital literacy to navigate the platform effectively.
- The ML models will improve over time with more user data.

---

13. Project Outputs & Deliverables

 Fully Functional Web Platform:
A responsive and user-friendly web application with integrated PCOS/PCOD tracking and expanding functionalities for thyroid health and menstrual cycle prediction.

 Machine Learning Models:
Trained and integrated ML models for PCOS/PCOD assessment, with ongoing development for thyroid disorder detection and menstrual cycle prediction.

 User Dashboard:
Personalized dashboards displaying health trends, insights, and downloadable reports.

 Secure Authentication System:
OAuth 2.0-based secure login and user data encryption ensuring privacy and protection.

 Community Support Forum (Planned):
A moderated space for peer support and discussions around women's health topics.

 API Documentation:
Comprehensive documentation for backend APIs and ML model integrations for future scalability.

 Testing Reports:
Detailed results from unit, integration, performance, and security tests, ensuring reliability and robustness.

 Deployment Pipeline:
A CI/CD pipeline enabling automated builds, testing, and deployments for both frontend and backend services.

 User Guide & Documentation:
Complete user manuals and technical documentation for end-users and developers.

 Future Enhancement Plan:
Roadmap for upcoming features like AI-powered consultations, multi-language support, and telehealth integrations.

14. Conclusion

AVA is designed as a comprehensive AI-powered women's health tracking system, offering personalized insights through machine learning. With an agile development approach and a scalable cloud-based architecture, it aims to enhance women's healthcare accessibility and reliability. The team is continuously working on refining existing features while integrating new functionalities to improve the platform's impact and usability.