Day 1: Setup and Initial Development

**Issue 1: Environment Setup**

- Description: Set up development environments for all team members, including necessary software and tools (IDEs, Docker, etc.).

- Checklist:

* Install Python, React.js, and all required libraries.
* Set up MongoDB accounts and databases.
* Ensure all developers have access to GitHub repository.

**Issue 2: Backend Initialization**

- Description: Initialize the Flask backend with basic routing and REST API functionalities.

- Checklist:

* Create Flask project structure.
* Set up basic routes (Health Queries, Provider Locator, Emergency Services).
* Initialize connection to MongoDB.

**Issue 3: Frontend Setup**

- Description: Scaffold the React.js application and set up routing and state management.

* Checklist:
* Create React project using Create React App.
* Set up routing with React Router.
* Implement global state management using Context API or Redux.

**Issue 4: API Integrations Setup**

- Description: Set up and test OpenAI API and Google Maps API integrations.

- Checklist:

* Implement OpenAI API integration for NLP functionalities.
* Implement Google Maps API for the healthcare provider locator.
* Ensure API keys and credentials are securely stored.

Day 2: Core Development and Integration

**Issue 5: Implement Personalized Health Queries**

- Description: Develop the functionality to process and respond to user health queries using NLP.

- Checklist:

* Create endpoint to receive health queries.
* Integrate OpenAI API to analyze and respond to queries.
* Implement voice input capabilities if time allows.

**Issue 6: Healthcare Provider Locator Feature**

- Description: Implement functionality to locate healthcare providers using Google Maps.

- Checklist:

* Develop backend logic to query MongoDB for provider data.
* Integrate Google Maps to display locations.
* Create frontend components to show map and provider details.

**Issue 7: Community Forum Module**

- Description: Develop features for the community forum where users can post, comment, and interact.

- Checklist:

* Design and implement database schema for forum posts and comments.
* Create backend APIs for posting, retrieving, and managing posts.
* Build frontend components for viewing and interacting with forum posts.

**Issue 8: Emergency Services Locator**

- Description: Implement a quick access feature to locate nearby emergency services.

- Checklist:

* Develop functionality to determine user's location.
* Integrate with Google Maps to find and display nearby emergency services.
* Create user interface for quick access on the frontend.

**Testing and Final Preparations**

**Issue 9: System Testing**

- Description: Perform comprehensive testing across all functionalities to ensure reliability and performance.

- Checklist:

* Conduct unit tests on backend functionalities.
* Execute integration tests for frontend and backend interactions.
* Perform end-to-end testing of the entire application.

**Issue 10: Presentation Preparation**

- Description: Prepare the final presentation and supporting materials to showcase the project.

- Checklist:

* Create a compelling presentation outlining project features and impact.
* Prepare a demonstration video of the working application.
* Compile and review all documentation for submission.