**Mental Health Support App**

1. **Introduction**

This report outlines a basic development plan for the **Mental Health Support Application**, which is a ***web-based platform*** designed to offer accessible and culturally relevant mental health resources to users in Africa. The application will leverage *React for the frontend*, *Flask for the backend*, and SQLite *(comes with flask)* for data storage.

1. **Target Audience**

The primary target audience ***includes young adults and professionals in Africa who seek*** accessible mental health resources. This group includes individuals experiencing stress, anxiety, or depression who prefer self-help tools and those interested in enhancing their mental well-being.

1. **Key Features and Their Importance**

* **Mood Tracking:** Allow users to ***monitor their emotional states and identify patterns*** over time.
* **Guided Meditation:** Provide ***structured meditation sessions*** to help reduce stress and improve mental well-being.
* **Journaling Features:** Encourage users to ***express their thoughts and feelings***, facilitating emotional processing and clarity.
* **AI Chatbot:** Deliver ***instant support and interaction***, offering companionship and guidance.
* **Personalized Suggestions:** Ensure users receive ***relevant resources*** and tips based on their data.

1. **Proposed Name for the Application**

* **AfyaMind** (**Meaning:** **Healthy Mind in Swahili)**

1. ***Additional Feature Suggestions***

* ***Community Support:*** *A forum or group feature for users to share experiences and support each other.*
* ***Daily Affirmations:*** *Daily positive affirmations to uplift users and enhance their mental health.*
* ***Emergency Contacts:*** *A feature for* ***users to quickly contact a trusted person*** *or* ***mental health professional*** *in times of crisis.*

1. **Development Guide**
   1. ***Setting Up the Development Environment***

* **Frontend:**
  + Install Node.js and create a React project using create-react-app.
  + Use a CSS framework like Bootstrap or Material-UI for styling.
* **Backend:**
  + Set up Python and Flask. Create a virtual environment and install necessary packages (Flask, Flask-SQLAlchemy, etc.).
* **Database:**
  + Use SQLite for storing user data.
  1. ***Designing the Database Schema***
* **User Table:** Stores user information (username, email, password).
* **Mood Logs Table:** Records daily mood tracking data.
* **Journal Entries Table:** Stores user journal entries.
* **Personalization Table:** Contains user preferences and suggestion data.
  1. ***Building the Backend (Flask)***
* **User Authentication:** Implement registration and login functionality using Flask-Login.
* **API Endpoints:** Create RESTful endpoints for submitting mood logs, saving journal entries, fetching suggestions, and interacting with the AI chatbot.
* **Data Processing:** Implement logic for mood data processing and generating suggestions.
* **AI Integration:** Consider using a pre-trained AI model or a cloud-based AI service for the chatbot feature, ***like Hugging Face Transformers for the chatbot feature.***
  1. ***Building the Frontend (React)***
* **UI Components:** Develop components for mood tracking, journaling, meditation guides, and the chatbot interface.
* **State Management:** Use React's built-in state management (useState, useContext) for handling user data.
* **API Integration:** Use axios or fetch to connect with the Flask backend endpoints.
  1. ***Testing and Debugging***
* **Unit Testing:** Write tests for individual components and backend endpoints using a testing framework like Jest.
* **Integration Testing:** Ensure proper communication between frontend and backend.
* **User Testing:** Collect feedback from potential users to refine the UI/UX.
  1. ***Deployment***
* **Frontend:** Deploy the React app using platforms like Vercel or Netlify.
* **Backend:** Deploy the Flask app using services like Heroku or AWS.
* **Database:** Ensure proper configuration and security for the SQLite database.

1. **Additional Considerations**

* **Accessibility:** Ensure the application is accessible to users with disabilities by following accessibility guidelines like **WCAG.**
* **Privacy and Security:** Implement measures to protect user data, such as ***encryption*** and secure ***authentication.***
* **Scalability:** Consider using a scalable database like ***PostgreSQL or MongoDB*** if the application is expected to grow significantly.
* **Content Creation:** Develop ***high-quality content*** for guided meditation sessions, journaling prompts, and affirmations.
* ***Community Building:*** *Adopt a supportive and inclusive community through features like forums and group chats.*

**Technology Stack:**

* **Frontend:** React, JavaScript, CSS (Bootstrap or Material-UI)
* **Backend:** Flask, Python
* **Database:** SQLite
* **AI:** Pre-trained AI models or cloud-based AI services (e.g., Google Cloud AI, AWS SageMaker)
* **Testing:** Jest
* **Deployment (Hosting):** Vercel, Netlify, Heroku, AWS