Hackathon Project Phases Template for the **AutoSage App** project.

Hackathon Project Phases Template

Project Title:

Care wise

Team Name:

Bug fixers

Team Members:

- Syeda Suhana
- K Yalleshwari
- Vangeti Anitha Reddy
- Bejgam Divya

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI Symptom Checker and Treatment Advisor using Palm's chat-bison-001 is to provide an accessible, reliable, and personalized healthcare tool that empowers users to make informed decisions about their health.

Key Points:

1. **Problem Statement:**
    - AI-Powered Symptom Analysis – CareWise leverages Google's PaLM chat-bison-001 to provide users with accurate symptom assessments and potential health conditions based on their inputs.

○ **Personalized Treatment Guidance** – It offers tailored treatment recommendations, self-care tips, and when necessary, advice on seeking professional medical attention.

2. **Proposed Solution:**

   ○ **Intelligent Symptom Assessment** – CareWise uses PaLM's chat-bison-001 to analyze user-reported symptoms, compare them with medical data, and suggest possible conditions.

   ○ **Personalized Treatment & Guidance** – It provides tailored treatment recommendations, self-care tips, and guidance on when to seek medical attention, enhancing accessibility to healthcare information.

3. **Target Users:**

   ○ **General Public** – Individuals seeking quick, AI-driven health assessments and self-care advice for common symptoms.

   ○ **Healthcare Seekers** – Patients looking for preliminary guidance before consulting a doctor, helping them make informed decisions about their health.

4. **Expected Outcome:**

   ○ **Accurate Symptom Insights** – Users receive reliable AI-driven symptom analysis and possible condition predictions.

   — **Enhanced Healthcare Accessibility** – Provides quick self-care tips and guidance on when to seek medical attention, improving health awareness and decision-making.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Carewise Website.

Key Points:

1. **Technical Requirements:**

   ○ Programming Language: **Python**

   ○ Backend: **Google Gemini Flash API**

   ○ Frontend: **Streamlit Web Framework ,HTML,CSS**

   ○ Database: **Not required initially (API-based queries)**

2. **Functional Requirements:**

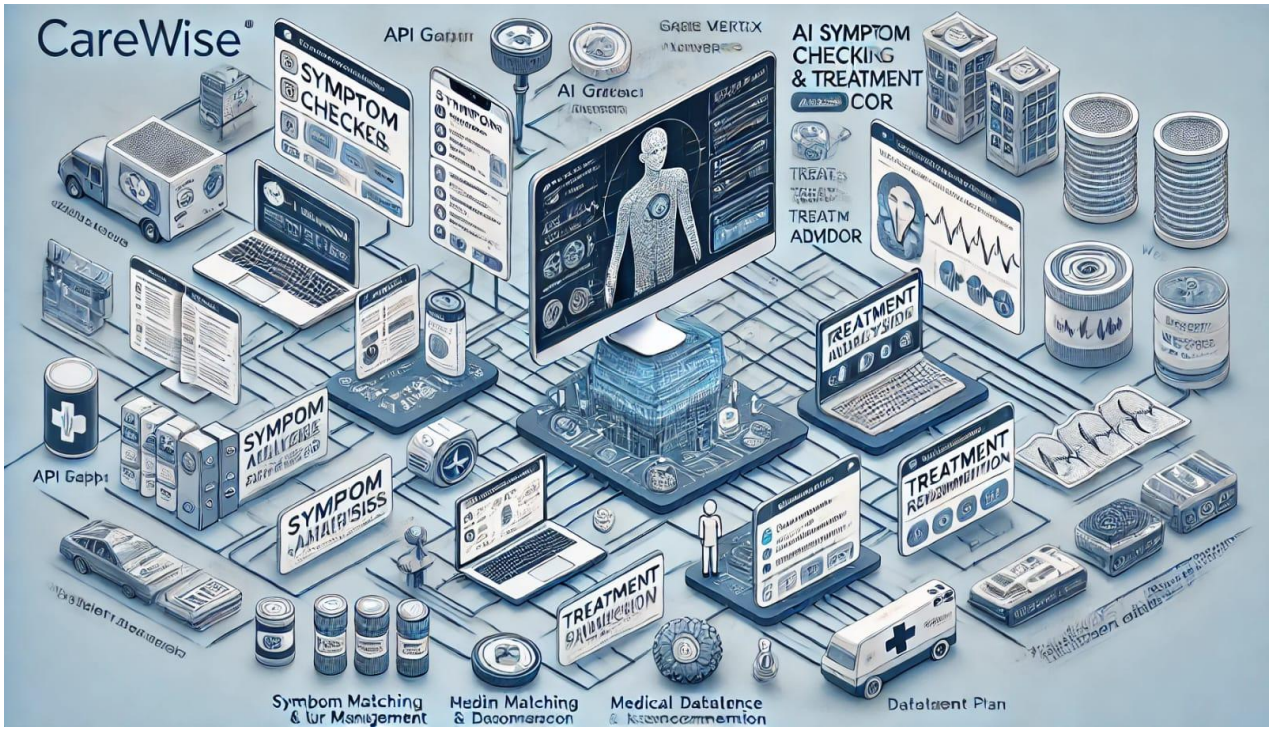
- **Symptom Analysis and Treatment Recommendation**: The system should allow users to input symptoms and provide AI-generated possible conditions, causes, and personalized treatment advice using chat-bison-001..
- **Emergency Detection and User Feedback**: The system must identify high-risk symptoms, recommend immediate action in emergencies, and collect user feedback to improve future advice accuracy.

3. **Constraints & Challenges:**
   - **Data Privacy and Compliance**: Ensuring user health data is securely handled and complies with privacy regulations like HIPAA or GDPR..
   - **Model Limitations and Accuracy**: Ensuring the AI provides accurate, reliable health recommendations and avoids incorrect diagnoses or advice.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   - **Client-Side (Frontend)**: The user interface (Web/Mobile) collects symptom data and sends it to the backend via API calls. The frontend, built with **React.js** or **Flutter**, displays AI-generated treatment advice and allows user feedback.

   - **Server-Side (Backend)**: The backend, implemented in **Python (Flask/FastAPI)**, processes user inputs, queries **Palm's chat-bison-001** API for symptom analysis, integrates with external databases (e.g., medication info), and returns personalized treatment recommendations.

2. **User Flow:**

- Step 1: User creates an account or logs in for personalized features, such as symptom tracking or history.
- Step 2: User inputs their symptoms (via text or voice). They may also enter optional demographic information (e.g., age, gender, medical history).
- Step 3: The backend processes the input and sends it to the **Palm's chat-bison-001** API for symptom analysis and treatment recommendations.
- Step 4: The backend processes the input and sends it to the **Palm's chat-bison-001** API for symptom analysis and treatment recommendations.
- Step 5: User is prompted to provide feedback on the AI's advice for accuracy.

3. **UI/UX Considerations:**

   - **User-Friendly Interaction**: Implement a conversational interface with easy symptom input, voice recognition, and accessible design features like high contrast and text resizing for all users.
   - **Personalized, Empathetic Feedback**: Offer clear, supportive responses tailored to the user's profile (age, gender, medical history) with immediate, attention-grabbin alerts for emergencies.

   .

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Symptom Analysis& Preprocessing	High	6 hours (Day 1)	End of Day 1	Member 1	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	AI-Powered Diagnosis Prediction	Medium	2 hours (Day 1)	End of Day 1	Member 2	API response format finalized	Basic UI with input fields
Sprint 2	Treatment Recommendations	High	3 hours (Day 2)	Mid-Day 2	Member 1& 2	API response, UI elements ready	Search functionality with filters
Sprint 2	Integration with Medical Databases	High	1.5 hours (Day 2)	Mid-Day 2	Member 1&4	API logs, UI inputs	Improved API stability
Sprint 3	User Feedback & AI Optimization	Medium	1.5 hours (Day 2)	Mid-Day 2	Member 2& 3	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Demo-ready project

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(**High Priority)** Set up the **environment** & install dependencies.
(**High Priority)** Integrate **Google Gemini API**.
(**Medium Priority)** Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(**High Priority)** Implement **search & comparison functionalities**.

(**High Priority)** Debug API issues & handle **errors in queries**.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

(Medium Priority) Test API responses, refine UI, & fix UI bugs. **(Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the CareWise Website

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Google Gemini Flash API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Gemini API integration**.
    - Develop **vehicle comparison and maintenance tips logic**.
    - Optimize **search queries for performance and relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** Delayed API response times.
      Fix: Implement **caching** to store frequently queried results.
    - **Challenge:** Limited API calls per minute.
      Fix: Optimize queries to fetch **only necessary data**.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Carewise website works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Test AI symptom diagnosis with common flu symptoms	AI correctly identifies flu symptoms and suggests treatment.	✅ Pass	Tester 1
TC-002	Functional Testing	Test AI symptom diagnosis with headache and dizziness	AI identifies possible causes and recommends advice.	✅ Pass	Tester 2
TC-003	Usability Testing	Test user interface interaction for symptom input	. User should be able to input symptoms easily and clearly	✅ Pass	Tester 3
TC-004	Performance Testing	Check AI response time for 5 different symptoms	Response time should be less than 3 seconds per query.	✅ pass	Developer
TC-005	Edge Case Testing	Test diagnosis for rare and less-known symptoms	AI should suggest a set of potential diagnoses or advise further consultation.	✅ pass	Tester 2
TC-006	Compatibility Testing	Test system performance with different device types	AI system should perform without issues on mobile and desktop devices.	✅ pass	DevOps

---

Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**
3. **Presentation**