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

Hackathon Project Phases Template

Project Title:

CareWise: AI Symptom Checker and Treatment Advisor using Palm's chat-bison-001\

Team Name: CUREINTEL

Team Members:

- Penmethsa Shanmukhi
- Tekula Sreevidya
- Siddamsetty Sai Deepanvitha
- Reddeddi Poojitha
- Kota Sneha

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered health assistant using **Palm's chat-bison-001** to analyze symptoms, suggest conditions, and provide treatment guidance.

Key Points:

1. **Problem Statement:**
    - Users lack immediate, reliable medical advice for symptoms.
    - Self-diagnosis often leads to misinformation and anxiety.
    - Provides real-time, accurate medical insights and preventive care tips.

2. **Proposed Solution:**

   - AI-driven symptom analysis, treatment recommendations, and urgency assessment.
   - Provides real-time, accurate medical insights and preventive care tips.

3. **Target Users:**

   - Individuals seeking quick symptom assessment.
   - Caregivers & remote users needing accessible medical advice.
   - Health-conscious users looking for preventive guidance.

4. **Expected Outcome:**

   - A functional AI health advisory app for smarter self-care and informed medical decisions.
   - Reduced hospital visits through AI-driven guidance.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the CareWise App.

Key Points:

1. **Technical Requirements:**

   - Programming Language: Python
   - Backend: Google PaLM's chat-bison-001 API.
   - Frontend: Flask
   - Database: Not required initially (API-based queries)

2. **Functional Requirements:**

   - Allow users to input symptoms and receive AI-powered health insights.
   - Fetch possible conditions, urgency levels, and treatment suggestions from PaLM's chat-bison-001.
   - Provide seasonal health tips for preventive care.
   - Offer insights on common illnesses, home remedies, and emergency indicators.

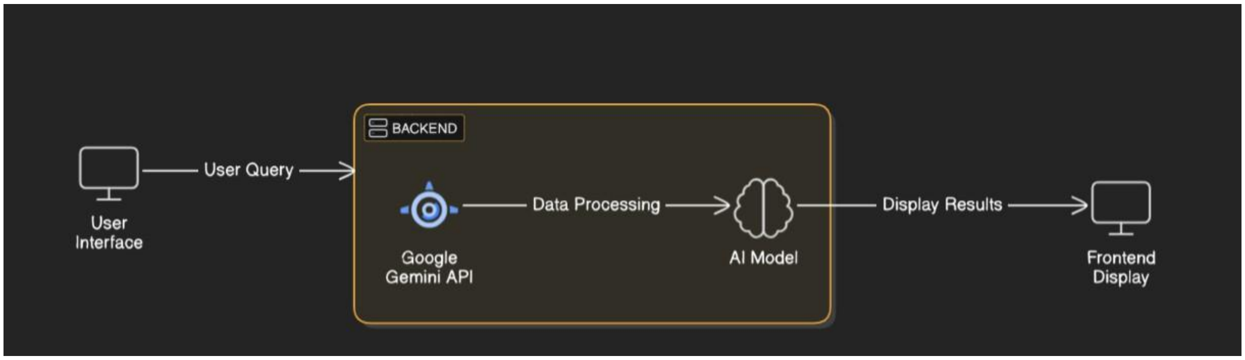
3. **Constraints & Challenges:**

   - Ensuring real-time medical insights with PaLM API.
   - Managing API rate limits and optimizing queries for efficiency.
   - Providing a user-friendly, intuitive UI with Flask for seamless interactions.

Phase-3: Project Design

Objective:

Develop the system architecture and user flow for the CareWise AI-powered Symptom Checker & Treatment Advisor using PaLM’s chat-bison-001 API.



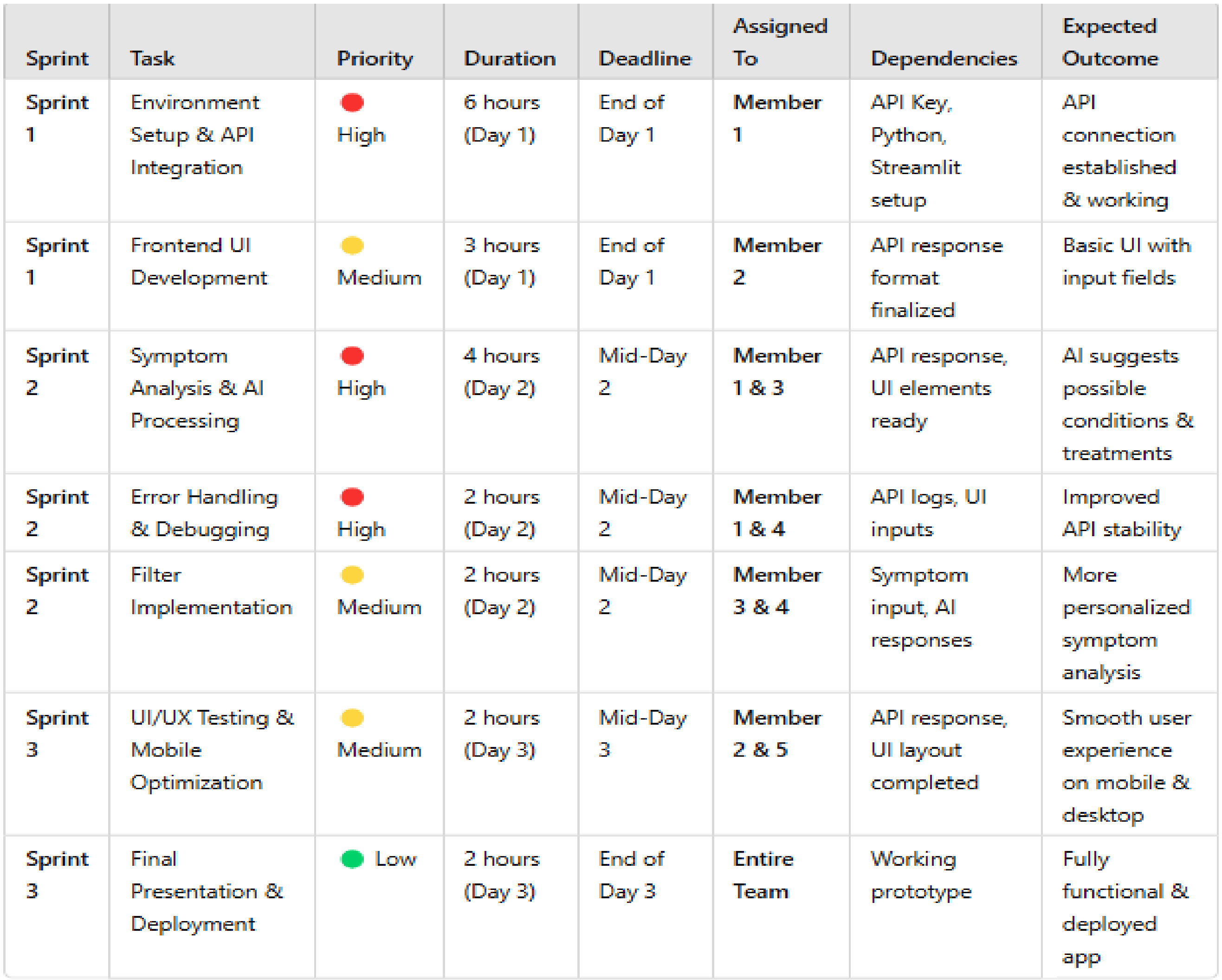
Key Points:

1. **System Architecture:**
   - User enters health-related query via UI (e.g., "I have a fever and sore throat").
   - Query is processed using Google PaLM’s chat-bison-001 API.
   - AI model analyzes symptoms and provides possible conditions & treatment advice.
   - Front-end displays AI-generated diagnosis, urgency level, and recommendations.
2. **User Flow:**
   - Step 1: User enters symptoms into the input field.
   - Step 2: Back-end calls PaLM chat-bison-001 API to process the query.
   - Step 3: AI returns:
     - Possible medical conditions.
     - Urgency level (low, moderate, high).
     - Recommended home remedies/treatments.
   - Step 4: The app displays results in an easy-to-read format.
   - Step 5: User decides whether to self-care or consult a doctor.
3. **UI/UX Considerations:**
   - **Minimalist, user-friendly interface for seamless interaction.**
   - **Filters for age, preexisting conditions, and symptom duration.**
   - **Dark & Light mode for accessibility & better user experience.**

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	Environment Setup & API Integration	🔴 High	6 hours (Day 1)	End of Day 1	**Member 1**	API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	3 hours (Day 1)	End of Day 1	**Member 2**	API response format finalized	Basic UI with input fields
Sprint 2	Symptom Analysis & AI Processing	🔴 High	4 hours (Day 2)	Mid-Day 2	**Member 1 & 3**	API response, UI elements ready	AI suggests possible conditions & treatments
Sprint 2	Error Handling & Debugging	🔴 High	2 hours (Day 2)	Mid-Day 2	**Member 1 & 4**	API logs, UI inputs	Improved API stability
Sprint 2	Filter Implementation	🟡 Medium	2 hours (Day 2)	Mid-Day 2	**Member 3 & 4**	Symptom input, AI responses	More personalized symptom analysis
Sprint 3	UI/UX Testing & Mobile Optimization	🟡 Medium	2 hours (Day 3)	Mid-Day 3	**Member 2 & 5**	API response, UI layout completed	Smooth user experience on mobile & desktop
Sprint 3	Final Presentation & Deployment	🟢 Low	2 hours (Day 3)	End of Day 3	**Entire Team**	Working prototype	Fully functional & deployed app

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(□ **High Priority)** Set up the **environment** & install dependencies.
(□ **High Priority)** Integrate Google PaLM chat-bison-001 API.
(□ **Medium Priority)** 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.

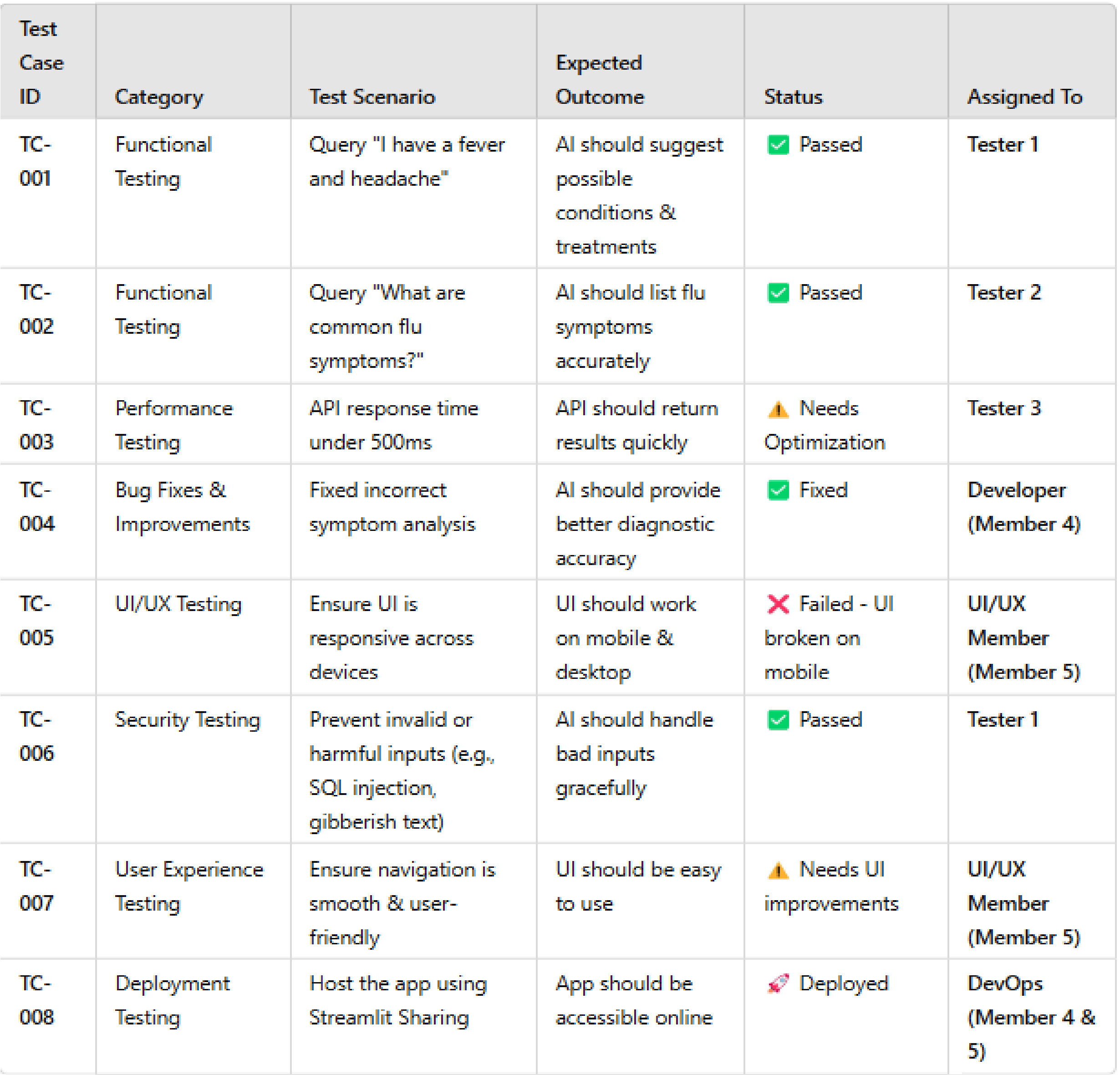
Key Points:

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 App works as expected.



Test Case ID	Category	Test Scenario	Expected Outcome	Status	Assigned To
TC-001	Functional Testing	Query "I have a fever and headache"	AI should suggest possible conditions & treatments	✅ Passed	**Tester 1**
TC-002	Functional Testing	Query "What are common flu symptoms?"	AI should list flu symptoms accurately	✅ Passed	**Tester 2**
TC-003	Performance Testing	API response time under 500ms	API should return results quickly	⚠️ Needs Optimization	**Tester 3**
TC-004	Bug Fixes & Improvements	Fixed incorrect symptom analysis	AI should provide better diagnostic accuracy	✅ Fixed	**Developer (Member 4)**
TC-005	UI/UX Testing	Ensure UI is responsive across devices	UI should work on mobile & desktop	❌ Failed - UI broken on mobile	**UI/UX Member (Member 5)**
TC-006	Security Testing	Prevent invalid or harmful inputs (e.g., SQL injection, gibberish text)	AI should handle bad inputs gracefully	✅ Passed	**Tester 1**
TC-007	User Experience Testing	Ensure navigation is smooth & user-friendly	UI should be easy to use	⚠️ Needs UI improvements	**UI/UX Member (Member 5)**
TC-008	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online	🚀 Deployed	**DevOps (Member 4 & 5)**