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

# **Hackathon Project Phases Template**

## **Project Title:**

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

## **Team Name:**

Tech Wizards

## **Team Members:**

* S.Naveen
* N.Vishnu kumar
* G.Subhash
* G.Vinay

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

Develop an AI-powered application designed to provide users with immediate and accurate symptoms. Leveraging advanced AI technology

### **Key Points:**

1. **Problem Statement:**
   * Many users struggle with Lack of immediate access to medical consultation in certain situations.
   * Difficulty in selecting appropriate over-the-counter medications and home remedies.
   * Users also need guidance for a reliable and quick way to assess symptoms and determine potential treatment options.
2. **Proposed Solution:**
   * Development of **CareWise**, an AI-powered symptom checker and treatment advisor.
   * Integration of **Palm’s chat-bison-001** model for accurate and instant symptom analysis.
   * The app offers providing tailored recommendations for **medications, potential side effects, allergy cautions, and home remedies**.
3. **Target Users:**
   * Individuals experiencing **common ailments** like headaches, fevers, or colds.
   * People who need **quick medical advice** but may not have immediate access to healthcare professionals.
   * Users looking for a **convenient and AI-driven health assistant** for self-care decisions.
4. **Expected Outcome:**
   * A functional AI-powered **symptom checker and treatment advisor** that provides **accurate medical insights** based on user queries and real-time symptom analysis.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the CareWiseAI App.

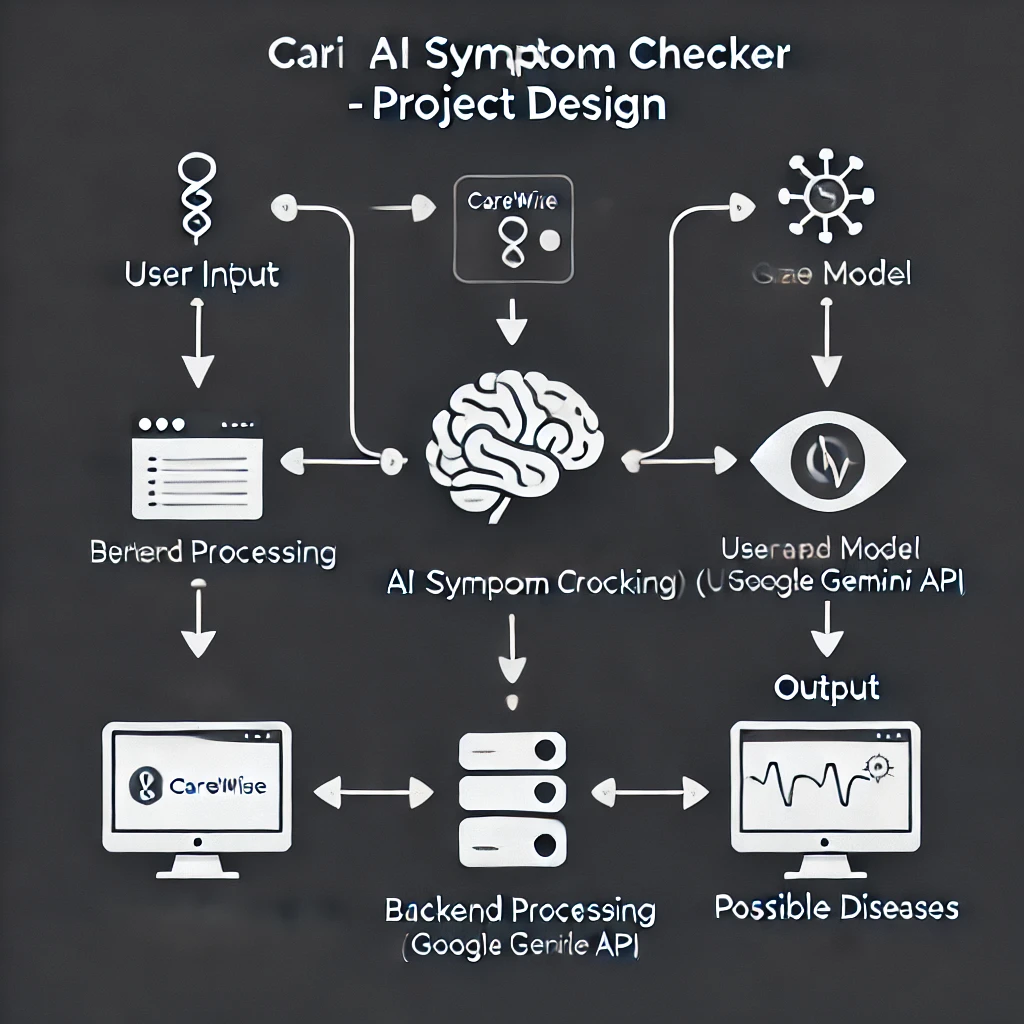
### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: scikit learn,numpy,joblib
   * Frontend: **Gradio,HTML/CSS**
   * **Data:** kaggle Dataset
2. **Functional Requirements:**
   * Ability to analyze user symptoms using Palm’s chat-bison-001 AI model.
   * **Provide tailored recommendations** for over-the-counter medications, including dosage and potential side effects.
   * **Display allergy warnings** based on user input and medical history.
   * Allow **users to track past symptom queries** and recommendations for future reference.
3. **Constraints & Challenges:**
   * Ensuring real-time updates from **Gemini API**.
   * Handling **API rate limits** and optimizing API calls.
   * Providing a **smooth UI experience** with Streamlit.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

**1. System Architecture:**

○ **User enters symptoms** via the app’s interface.  
 ○ **Query is processed** using Palm’s chat-bison-001 AI model.  
 ○ **AI model fetches and analyzes** symptom-related data.

**2. User Flow:**

○ **Step 1:** User inputs symptoms (e.g., "I have a fever and headache").  
 ○ **Step 2:** The AI processes the symptoms and queries relevant medical data.  
 ○ **Step 3:** The system provides a list of **recommended medications, side effects, and home remedies**.  
 ○ **Step 4:** If symptoms indicate a serious condition, the app suggests **seeking medical attention.**  
○ **Step 5:** User can save recommendations or ask for additional clarifications.

**3. UI/UX Considerations:**

○ **Clean, intuitive interface** with simple symptom input and response layout.  
 ○ **Categorized results** for medications, side effects, and home remedies.  
 ○ **Dark & light mode** for better readability.  
 ○ **Voice input support** for accessibility.  
 ○ **Interactive chatbot experience** for engaging and natural symptom discussions.  
 ○ **Quick-access emergency button** for severe symptom alerts.

## 

## **Phase-4: Project Planning (Agile Methodologies)**

### **Objective:**

**Break down development tasks for efficient completion using Agile methodologies.**

| **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** | **Developer 1** | **Palm’s API Key, Python, Streamlit setup** | **API connection established & working** |
| **Sprint 1** | **Frontend UI Development** | **🟡 Medium** | **2 hours (Day 1)** | **End of Day 1** | **Developer 2** | **API response format finalized** | **Basic UI with input fields** |
| **Sprint 2** | **Symptom Analysis & Treatment Suggestions** | **🔴 High** | **3 hours (Day 2)** | **Mid-Day 2** | **Developer 3** | **API response, UI elements ready** | **AI-driven symptom assessment & treatment recommendations** |
| **Sprint 2** | **Error Handling & Debugging** | **🔴 High** | **1.5 hours (Day 2)** | **Mid-Day 2** | **Developer 4** | **API logs, UI inputs** | **Improved API stability** |
| **Sprint 3** | **Testing & UI Enhancements** | **🟡 Medium** | **1.5 hours (Day 2)** | **Mid-Day 2** | **Developer 2 & 3** | **API response, UI layout completed** | **Responsive UI, improved 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 development environment & install dependencies.  
🔴 High Priority – Integrate Palm’s chat-bison-001 AI API for symptom analysis.  
🟡 Medium Priority – Build a basic UI with input fields for user symptom input.**

**Sprint 2 – Core Features & Debugging (Day 2)**

**🔴 High Priority – Implement AI-based symptom analysis and treatment suggestions.  
🔴 High Priority – Debug API issues, improve response accuracy, and handle user input errors.**

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

**🟡 Medium Priority – Test API responses, refine UI, and fix UI bugs for better UX.  
🟢 Low Priority – Prepare final demo & deploy the application.**

## **Phase-5: Project Development**

### **Objective:**

Implement core features of the CareWiseAI App.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit for a user-friendly interface.
   * **Backend:** Palm’s chat-bison-001 AI API for symptom analysis.
   * **Programming Language:** Python for API integration and data processing.
2. **Development Process:**
   * **Implement API key authentication** and integrate Palm’s chat-bison-001 AI model.
   * **Develop logic for symptom analysis** and personalized treatment recommendations.
   * **Optimize AI queries** to improve response accuracy and relevance.
3. **Challenges & Fixes:**

○ **Challenge:** Inconsistent AI-generated recommendations.  
 **Fix:** Fine-tune queries and implement validation checks for accuracy.  
 ○ **Challenge:** High API response time for symptom analysis.  
 **Fix:** Implement caching for frequently searched symptoms.  
 ○ **Challenge:** Handling ambiguous or incorrect user inputs.  
 **Fix:** Introduce input validation and guided question prompts for better accuracy.

## **Phase-6: Functional & Performance Testing**

### **Objective:**

Ensure that the CareWiseAI App works as expected.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | Query "Common cold symptoms" | Possible diseases should be displayed. | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | Query "Headache and nausea" | AI should suggest possible causes. | ✅ Passed | Tester 2 |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Programming Testing | Verify AI processes symptoms correctly | AI should not provide unrelated outputs. | ✅ Passed | Developer |
| TC-005 | Programming Testing | Handle incorrect or incomplete symptom input | App should show a friendly error message. | ❌ Failed - No error handling | Tester 2 |
| TC-006 | UI/UX Testing | Ensure UI works on mobile & desktop | App should be fully responsive. | ✅ Passed | Tester 1 |
| TC-007 | Deployment Testing | Host the app using Google Cloud Run | App should be accessible online. | 🚀 Deployed | DevOps |

## **Final Submission**

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**