Hackathon Project Phases Template

Project Title:

AI Symptom checker and treatment advisor

Team Name:

MediGuide

Team Members:

- P Greeshma
- S Roopa Vishal
- Sk Cherishma
- T Sushma

Phase-1: Brainstorming & Ideation

Objective:

The AI Symptom Checker and Treatment Advisor includes six modules for diagnosis, treatment recommendations, telehealth integration, and security. TensorFlow, and PostgreSQL, it ensures scalable, secure, and AI-driven healthcare.

Key Points:

1. **Problem Statement:**
    - Lack of timely, accurate, and accessible medical guidance leads to delayed diagnoses, unnecessary hospital visits, and high healthcare costs. An AI-powered solution is needed to provide real-time symptom analysis, treatment recommendations, and telehealth integration for better healthcare access.
2. **Proposed Solution:**
    - AI-Powered Symptom Analysis & Treatment Advisor – Uses PaLMs Chat Bison 001 and machine learning to provide real-time symptom assessment, possible diagnoses, and personalized treatment recommendations.
    - Seamless Telehealth – ensuring timely medical assistance, affordability, and

accessibility while maintaining data security and compliance.

3. **Target Users:**
    - **Individuals & Patients – People seeking quick, AI-driven symptom analysis and treatment recommendations, especially in remote areas.**
    - **Healthcare Providers & Telemedicine Platforms – Doctors, clinics, and hospitals looking to enhance patient pre-screening, reduce workload, and improve efficiency.**
    - **Pharmacies & Insurance Companies – Businesses benefiting from AI-driven prescriptions, medication recommendations, and health data insights for better service delivery.**

4. **Expected Outcome:**
    - Improved Healthcare Accessibility & Efficiency – Enables faster, AI-driven symptom analysis, accurate treatment recommendations, and seamless telehealth integration, reducing hospital congestion, consultation costs, and healthcare disparities.

Phase-2: Requirement Analysis

Objective:

Develop an AI-powered Symptom Checker and Treatment Advisor to provide accurate, real-time medical insights and seamless telehealth integration for better healthcare accessibility.

Key Points:

1. **Technical Requirements:**
    - PaLMs Chat Bison 001, TensorFlow, and NLP models for AI processing.
    - Secure database (PostgreSQL/MongoDB) with API integration for real-time updates.

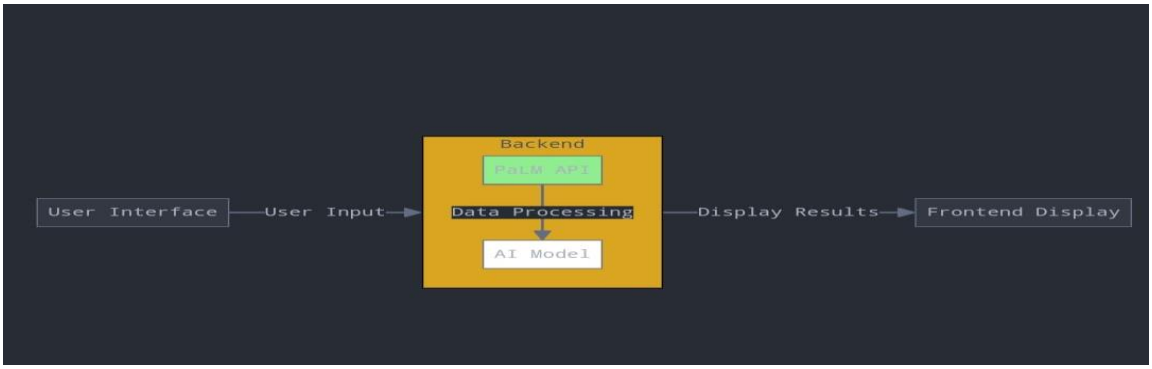
2. **Functional Requirements:**
    - User-friendly interface for symptom input via text.
    - Real-time diagnosis, treatment suggestions, and doctor connectivity.

3. **Constraints & Challenges:**
    - Ensuring medical accuracy and compliance with regulations (HIPAA/GDPR).
    - Handling multilingual support and scalability for global users.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - A modular AI-driven framework using PaLMs Chat Bison 001, TensorFlow, and cloud-based databases, integrating real-time APIs for telehealth.
2. **User Flow:**
    - Users enter symptoms via text/voice, AI analyzes inputs, provides possible diagnoses and treatment recommendations, and offers doctor connectivity for further assistance.
3. **UI/UX Considerations:**
    - A clean, intuitive interface with multilingual support, voice input, and responsive design, ensuring easy navigation and accessibility for all users

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	API Integration	🔴 High	6 hours (Day 1)	End of Day 1	Sk Cherishma	Google API Key, Python setup	API connection established
Sprint 1	Basic UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	S Roopa Vishal	API response format	Basic UI with input fields
Sprint 2	Symptom Analysis Implementation	🔴 High	3 hours (Day 2)	Mid-Day 2	T Sushma	API response ready	AI-driven symptom identification
Sprint 2	Treatment Suggestion Module	🔴 High	1.5 hours (Day 2)	Mid-Day 2	P Greeshma & S Roopa Vishal	AI Module Integration	Treatment advice displayed
Sprint 3	UI Enhancements & Testing	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Sk Cherishma	UI completed	Improved user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	Entire Team	Fully functional prototype	Demo-ready project

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(🔴 **High Priority)** Set up the **environment** & install dependencies.
(🟡 **Medium Priority)** Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** Implement **symptom analysis functionality.**
(🔴 **High Priority)** Develop treatment recommendation module.

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

(🟡 **Medium Priority)** Improve response time & optimize performance.
(🟢 **Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the AI Symptom Checker.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Google palm API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API authentication and palm integration.**
    - Develop **symptom-matching and analysis logic**.
    - Optimize **response time and improve model accuracy.**
3. **Challenges & Fixes:**
    - **Challenge:** Slow API response times.
      Fix: Implement **caching** for common queries.
    - **Challenge:** Handling medical accuracy concerns.
      Fix: Cross-reference with medical database and sources.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional	Query "Fever and headache"	Suggested possible conditions.	✅ Passed	Greeshma
TC-002	Functional	Query "Cold and sore throat"	Treatment recommendations provided	✅ Passed	T Sushma
TC-003	Performance	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Sk Cherishma
TC-004	Bug Fixes	Fixed incorrect AI suggestions	Improved data accuracy	✅ Fixed	Developer
TC-005	UI Testing	Ensure mobile compatibility	UI should work on mobile & desktop	❌ Failed - UI broken on mobile	S Roopa Vishal
TC-006	Deployment Testing	Deploy app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	DevOps

---

Final Submission

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