**Hackathon Project Phases Template** for the **Playful AI** project.

Hackathon Project Phases Template

# Project Title:

**Playful AI: Intelligent Board Game Opponents and Advisors**

# Team Name:

Impact Players

# Team Members:

* G. Rishi Kumar
* Rahul Kichad
* G. Rohan Krishna
* S. Sai Sampreeth

# 

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an AI-powered board game tool using **GenAI** to provide intelligent opponents and strategic advice, enhancing the gameplay experience for casual players, enthusiasts, and professionals.

## Key Points:

1. **Problem Statement:**

**Challenges for Players:**

* + Casual players struggle to find opponents that match their skill level, leading to either overly easy or frustratingly difficult games.
  + Players often lack access to **strategic advice** or tools to improve their gameplay and decision-making.
  + Board game enthusiasts and professionals need advanced tools to simulate **intelligent opponents** and receive **strategic guidance** for skill development.

**Current Gaps:**

* + Limited availability of adaptive AI opponents that can adjust to different skill levels.
  + Lack of tools that provide real-time strategic insights and personalized gameplay recommendations.

1. **Proposed Solution:**

**AI-Powered Board Game Tool** **using GenAI to offer**:

* **Intelligent Opponents:** Adaptive AI for balanced, skill-matched challenges.
* **Strategic Advice:** Real-time insights to improve gameplay and decision-making.
* **Dynamic Gameplay:** Opponents adjust strategies based on player actions.

**Key Features:**

* **Adaptive Difficulty:** Customizable game and difficulty levels.
* **Strategic Insights:** Tips and analysis for skill improvement.
* **Multi-Game Support:** Compatible with various popular board games.

1. **Target Users:**

* **Casual Players:** Individuals looking for fun, engaging, and balanced gameplay experiences.
* **Board Game Enthusiasts:** Players seeking to improve their skills and challenge themselves with intelligent opponents.
* **Professionals:** Advanced players and competitors who need tools for strategic analysis and skill development.

1. **Expected Outcome:**

**A** **functional AI-powered board game tool** **that delivers:**

* + **Adaptive Opponents:** AI opponents that provide a balanced challenge for players of all skill levels.
  + **Strategic Guidance:** Real-time advice and insights to help players improve their gameplay.
  + **Enhanced Gameplay Experience:** A more engaging, immersive, and enjoyable board game experience for all users.

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the Playful AI website

## Key Points:

1. **Technical Requirements:**

* **Programming Language:** JavaScript
* **Frontend Framework:** Next.js
* **Backend:** Gemini (GenAI) API
* **Database:** MongoDB (for storing user preferences, game history, and AI opponent configurations)

1. **Functional Requirements:**

**Intelligent Opponents:**

* Use **Gemini (GenAI)** to create adaptive AI opponents that match the player’s skill level.
* Allow players to select their preferred game and difficulty level.

**Strategic Advice:**

* + Provide real-time guidance and insights to help players improve their strategies.
  + Display tips, recommendations, and analysis in an intuitive UI.

**Dynamic Gameplay:**

* + Ensure AI opponents adjust their strategies based on the player’s actions.
  + Support multiple popular board games for diverse player preferences.

**User Profiles:**

* + Store user preferences, game history, and AI opponent configurations in **MongoDB**.

1. **Constraints & Challenges:**

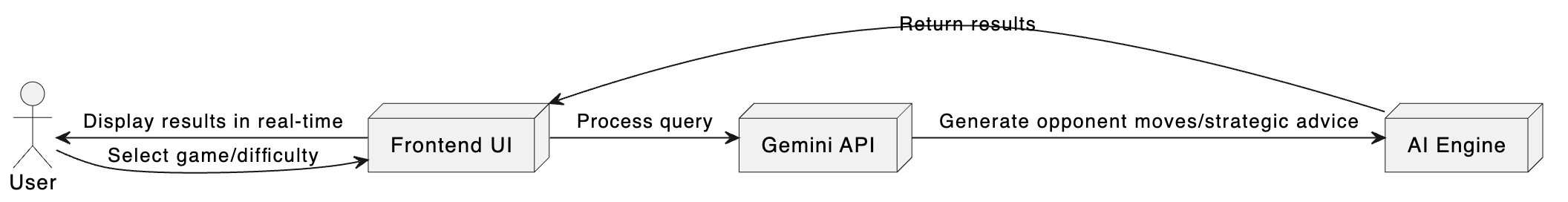
* Ensuring **real-time updates** and responsiveness from the **Gemini (GenAI)** API.
* Handling **API rate limits** and optimizing API calls for smooth performance.
* Providing a **user-friendly UI experience** with **Next.js** for seamless gameplay.
* Managing **MongoDB** efficiently to store and retrieve user data without performance

Bottlenecks.

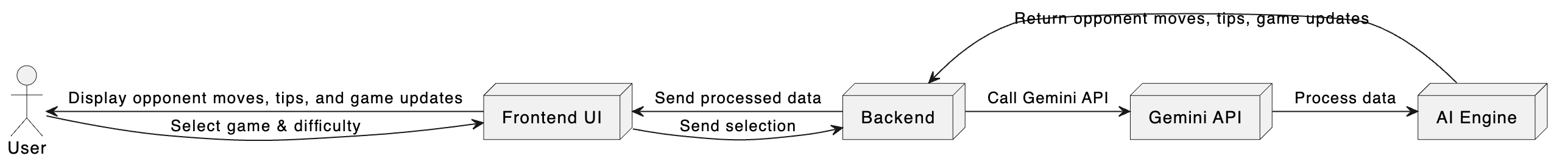
# Phase-3: Project Design

## Key Points:

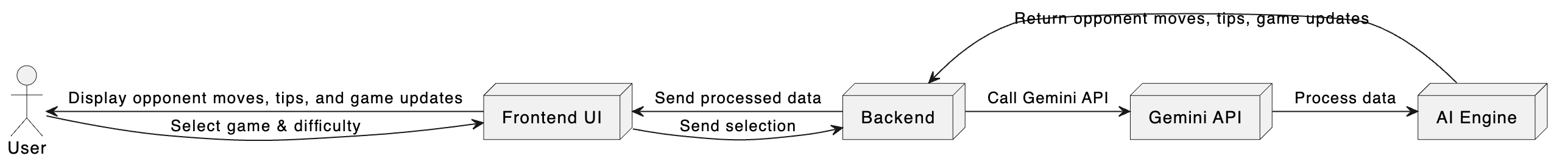
1. **System Architecture:**



1. **User Flow:**



1. **UI/UX Considerations:**



# Phase-4: Project Planning (Agile Methodologies)

# Objective:

| **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 | G. Rishi Kumar | Gemini API Key, Node.js, MongoDB setup | API connection established & working |
| **Sprint 1** | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Rahul Kichad | API response format finalized | Basic UI with game selection and difficulty input |
| **Sprint 2** | Game Selection & Opponent Setup | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | G. Rishi Kumar, Rahul Kichad | API response, UI elements ready | Game selection and adaptive opponent functionality |
| **Sprint 2** | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | G. Rishi Kumar, S. Sai Sampreeth | API logs, UI inputs | Improved API stability and error-free gameplay |
| **Sprint 3** | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Rahul Kichad, G. Rohan Krishna | 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 (Node.js, MongoDB).
* 🔴 **High Priority**: Integrate **Gemini API** and ensure connectivity.
* 🟡 **Medium Priority**: Build a **basic UI** for game selection and difficulty input.

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

* 🔴 **High Priority**: Implement **game selection & adaptive opponent setup**.
* 🔴 **High Priority**: Debug **API issues** & handle errors in game logic.

#### **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 **Playful AI**.

#### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** React.js
   * **Backend:** Google Gemini API
   * **Database:** MongoDB
   * **Programming Language:** Node.js, JavaScript
2. **Development Process:**
   * Implement **API key authentication** and Gemini API integration.
   * Develop **game selection & adaptive opponent** setup.
   * Optimize **API requests** for efficient response handling.
3. **Challenges & Fixes:**
   * **Challenge:** API response delays.
     + **Fix:** Implement caching to reduce redundant API calls.
   * **Challenge:** Limited API request quota.
     + **Fix:** Optimize requests to fetch only essential data.

# Phase-6: Functional & Performance Testing

## Objective:

Ensure that the **Playful AI** website works as expected.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| **TC-001** | Functional Testing | Query "Select a game with medium difficulty" | Relevant games should be displayed. | ✅ Passed | Tester 1 |
| **TC-002** | Functional Testing | Query "Find an opponent for chess" | Opponent should be assigned based on skill. | ✅ 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 API responses. | Data accuracy should be improved. | ✅ Fixed | Developer |
| **TC-005** | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Tester 2 |
| **TC-006** | Deployment Testing | Deploy Playful AI on a hosting platform. | App should be accessible online. | 🚀 Deployed | DevOps |