# **Hackathon Project Phases Template**

## **Project Title:**

## ***Al-Powered Ethical Decision Making Simulations with Mistral***

**Team Name: Grash**

**Team Members:**

1. **Vemula Harshavardhan Reddy**
2. **K. Ganesh**
3. **J. Hemanth Naga Sai Kumar**
4. **T. Srikar Reddy**
5. **N. Rushikesh**

**Phase-1: Brainstorming & Ideation**

**Objective**

Develop an AI-powered ethical decision-making tool that presents moral dilemmas and provides AI-driven insights based on user choices.

🔹 **Problem Statement**

Many individuals and organizations struggle with ethical decision-making in complex scenarios.

There is a need for an AI-powered tool that presents moral dilemmas and evaluates responses to promote responsible decision-making.

1. **🔹 Proposed Solution**

**Backend:** Flask-based server storing moral dilemma questions in a JSON file.

**Frontend:** Streamlit interface displaying questions with images and descriptions, allowing users to select ethical choices.

**AI Analysis:** Mistral AI (or Gemini AI) analyzes user choices and provides insights into their ethical implications.

**Target Users:**

Policymakers making critical ethical decisions.

Business leaders analyzing corporate ethics.

Educators and students learning about ethics.

Psychologists studying ethical decision-making.

**Expected Outcome:**

A functional AI-driven tool that fosters ethical discussions and helps users understand the impact of different choices in real-world scenarios.

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

### **Objective:**

### Define the technical and functional requirements for the Ethical Decision-Making Simulations project.

### **Key Points:**

1. **Technical Requirements:**
   * **Programming Language:** Python
   * **Backend:** Flask
   * **Frontend:** Streamlit Web Framework
   * **Database:** JSON-based storage for dilemmas
   * **AI Integration:** Mistral AI for ethical analysis
2. **Functional Requirements:**
   * Serve pre-defined ethical dilemmas from JSON.
   * Display two contrasting ethical choices with images.
   * Allow user responses and forward them to Mistral AI for evaluation.
   * Provide AI-generated insights on ethical implications.
3. **Constraints & Challenges:**
   * Ensuring relevant and unbiased ethical analysis.
   * Handling large sets of dilemmas efficiently.
   * Designing an intuitive and engaging UI.

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

### **Objective:**

Develop the architecture and user flow of the application.

### **Key Points:**

1. **System Architecture:**
   * **User Interaction:** User selects an ethical scenario on the UI.
   * **Data Processing:** The backend fetches the relevant scenario from JSON.
   * **AI Analysis:** The user’s choice is sent to Mistral AI for evaluation.
   * **Result Display:** The frontend presents the AI-generated ethical insights.
2. **User Flow:**
   * **Step 1:** User selects a moral dilemma.
   * **Step 2:** Backend retrieves the dilemma and presents it with two contrasting choices.
   * **Step 3:** User selects a choice, which is sent to Mistral AI.
   * **Step 4:** AI provides an analysis of the ethical implications of the choice.
   * **Step 5:** User sees the result and gains insights.
3. **UI/UX Considerations:**
   * **Minimalist and user-friendly interface.**
   * **Engaging visuals to support ethical dilemmas.**
   * **Dark & light mode for accessibility.**

## 

## **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 | Backend Setup & API Integration | 🔴 High | 6 hours (Day 1) | End of Day 1 | Developer 1 | Flask  setup, JSON dilemmas | API connection working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 3 hours (Day 1) | End of Day 1 | Developer 2 | API response format finalized | Basic UI with input fields |
| Sprint 2 | Ethical Scenario Handling | 🔴 High | 4 hours (Day 2) | Mid-Day 2 | Developer 3 | Backend and UI elements ready | Scenarios displayed correctly |
| Sprint 2 | Mistral AI Integration | 🔴 High | 5 hours (Day 2) | Mid-Day 2 | Developer 4 | API connectivity with Mistral AI | AI-generated ethical analysis |
| Sprint 3 | UI Enhancements & Debugging | 🟡 Medium | 2 hours (Day 2) | Mid-Day 2 | Developer 2 | API response, UI layout completed | Improved user experience |
| Sprint 3 | Final Testing & Deployment | 🟢 Low | 1 hour (Day 2) | End of Day 2 | Entire Team | Working prototype | Demo-ready project |

### 

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

### **Objective:**

### Implement core features of the Ethical AI project.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Flask
   * **AI Integration:** Mistral AI
   * **Programming Language:** Python
2. **Development Process:**
   * Implement JSON-based ethical dilemmas.
   * Integrate Flask with Streamlit.
   * Develop AI interaction logic with Mistral AI.
3. **Challenges & Fixes:**
   * **Challenge:** AI biases in ethical analysis.  
     **Fix:** Implement a diverse dataset of ethical dilemmas.
   * **Challenge:** Slow API response times.  
     **Fix:** Optimize API queries and caching.

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

### **Objective:**

Ensure that the Ethical AI application functions as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | User selects an ethical scenario | The correct dilemma and choices appear | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | User submits a response | Mistral AI provides analysis | ✅ Passed | Tester 2 |
| TC-003 | Performance Testing | API response time under 500ms | Quick response from AI | ✅ Passed | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fix UI glitches | Improved UI experience | ✅ Fixed | Developer |
| TC-005 | Final Validation | Ensure responsiveness across devices | Works on mobile & desktop | ✅ Passed | Tester 2 |
| TC-006 | Deployment Testing | Host app using Streamlit Sharing | App accessible online | ✅ Passed | DevOps |

## **Final Submission**

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