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

# Hackathon Project Phases Template

**Project Title:**

Logo Craft: Innovation Logo Generation with Diffusion Technology

**Team Name:**

AWESOME 5

**Team Members:**

* Charan
* trishal
* mounika
* nityashree
* meghana

## Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-powered logo generation tool using diffusion technology to create unique, innovative, and customizable logos for businesses and individuals.

**Key Points:**

1. **Problem Statement:**

- Many businesses and individuals struggle to create unique and professional logos due to lack of design skills or resources.

- Existing logo generation tools often produce generic or low-quality designs.

1. **Proposed Solution:**

- An AI-powered application using diffusion technology to generate high-quality, customizable logos.

- The app will allow users to input their preferences (e.g., colors, styles, themes) and generate logos in real-time.

1. **Target Users:**

Small businesses and startups looking for affordable logo design solutions.

- Freelancers and individuals who need quick and unique logos.

- Designers seeking inspiration or a starting point for their projects.

1. **Expected Outcome:**

- A functional AI-powered logo generation tool that produces high-quality, customizable logos based on user inputs.

.

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the Logo Craft application**.**

**Key Points:**

1. **Technical Requirements:**

-Programming Language: Python

- AI Model: Diffusion-based generative model (e.g., Stable Diffusion)

- Frontend: Streamlit Web Framework

- Backend: hugging face inference api /python-flask

- Database:MongoDB for storing user preferences and generated logos

1. **Functional Requirements:**

- Ability to generate logos based on user inputs (e.g., colors, themes, styles).

- Provide customization options for generated logos (e.g., text, icons, layout).

- Allow users to download logos in multiple formats (e.g., PNG, SVG).

- Offer a preview feature to visualize logos before finalizing.

1. **Constraints & Challenges:**

- Ensuring high-quality logo generation with diffusion technology.

- Handling user customization requests in real-time.

- Optimizing the AI model for fast and efficient logo generation**.**

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the Logo Craft application.

logo-craft/

│

├── huggingface\_api.py # Contains API integration code

├── logo\_generator.py # Main Streamlit app

├── requirements.txt # List of required dependencies

└── assets/ # (Optional) Store any static assets like logos, icons, etc.

**Key Points:**

1. **System Architecture:**

-System Architecture

- User inputs logo preferences via the UI.

- The backend processes the inputs and calls the diffusion model.

- The AI model generates the logo and sends it back to the frontend.

- The frontend displays the generated logo with customization options.

1. **User Flow:**

-Step 1: User enters logo preferences (e.g., colors, themes, styles).

- Step 2: The backend calls the diffusion model to generate the logo.

- Step 3: The app displays the generated logo with customization options.

- Step 4: User customizes the logo and downloads it.

1. **UI/UX Considerations:**

Intuitive and user-friendly interface for easy navigation.

Customization options for colors, fonts, and icons.

Preview feature to visualize the logo before downloading.

## 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 | charan | AI Model, Python, Streamlit setup | AI model integration & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | nithyashree | AI model response format finalized | Basic UI with input fields |
| Sprint 2 | Vehicle Search &  Comparison | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Mounika & meghana | AI model response, UI elements ready | Logo generation with customization options |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Charan & trishal | AI model logs, UI inputs | Improved AI model stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Nihyashree & mounika | AI model 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.

**(**🔴 **High Priority)** Integrate **Google Gemini API**.

**(**🟡 **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 Logo Craft application.

**Key Points:**

1. **Technology Stack Used:**

- Frontend: Streamlit

- Backend: hugging face inference api,image handling

- AI Model:Diffusion-based generative model (e.g., Stable Diffusion)

- Programming Language: Python

1. **Development Process:**

- Implement AI model integration and API handling.

- Develop logo generation and customization logic.

- Optimize the AI model for fast and efficient logo generation.

1. **Challenges & Fixes:**

- Challenge: High computational requirements for diffusion models.

- Fix:implement model optimization techniques to reduce resource usage.

- Challenge: Real-time customization requests.

- Fix: Use caching to store frequently generated logos.

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the Logo Craft application works as expected.

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Generate a logo with specific colors and themes | Logo should match the specified colors and themes. | ✅ Passed | nithyashree |
| TC-002 | Functional  Testing | Customize a generated logo with text and icons | Logo should reflect the customizations.  . | ✅ Passed | mounika |
| TC-003 | Performance  Testing | Logo generation time under 5 seconds | | Logo should be generated quickly. | ⚠ Needs Optimization | trishal |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect logo generation | Logo accuracy should be improved. | ✅ Fixed | charan |
| TC-005 | Final Validation | Ensure UI is responsive across devices | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | meghana |
| TC-006 | Deployment  Testing | Host the app using Streamlit Sharing | |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**