**Hackathon Project Phases Template** for the **Logo Craft** project.

## **Project Title:**

### Innovative Logo Generation with Diffusion Technology

## **Team Name:**

GenLogo Innovators

## **Team Members:**

* T Nagasri
* G Ashwini
* M Manasa
* P Maithri

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

### **Objective:**

Develop an AI-powered logo generation tool using **Stable Diffusion** to help businesses create unique, high-quality logos based on user descriptions.

### **Key Points:**

1. **Problem Statement:**
   * Many businesses struggle to create professional, memorable logos due to a lack of design expertise and high costs. Traditional logo design services are expensive and time-consuming.
2. **Proposed Solution:**
   * An AI-powered web application using **Stable Diffusion** to generate logos based on brand identity, industry, and style preferences provided by users.
3. **Target Users:**
   * Startups and small businesses needing branding solutions.
   * Entrepreneurs and freelancers who require cost-effective logo designs.
   * Non-designers looking for AI-generated logo ideas.
4. **Expected Outcome:**
   * A functional AI-powered logo generator that creates unique, downloadable logos based on user input.

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

### **Objective:**

Define the technical and functional requirements for the Logo Craft.

### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * **Backend:** Flask (Handles user input and AI model requests)
   * Frontend: **Streamlit Web Framework**
   * **Storage:** Firebase/AWS S3 (for saving generated logos)
2. **Functional Requirements:**
   * Accept user input: **brand name, industry, style preferences, and color scheme**.
   * Generate multiple logo variations using Stable Diffusion.
   * Provide a **downloadable PNG file** of the generated logo.
   * Enable **refinements** (adjust colors, fonts, or add icons).
3. **Constraints & Challenges:**
   * Optimizing model performance for **fast logo generation**.

Ensuring **high-quality** output with appropriate image resolution.

* + Hosting model **efficiently** to handle multiple requests.

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

### **Objective:**

Develop the system architecture and user flow.



### **Key Points:**

1. **System Architecture:**
   * User enters **brand details** on the website
   * Flask backend processes input and **calls Stable Diffusion**.
   * AI model generates **multiple logo variations**.
   * User selects a logo and **downloads the final image**.
2. **User Flow:**
   * **Step 1:** User enters brand name, industry, style, and colors.
   * **Step 2:** Backend calls Stable Diffusion to generate logos.
   * **Step 3:** User previews generated logos and selects the best one.
   * **Step 4:** User downloads the logo in **PNG/SVG format**
3. **UI/UX Considerations:**
   * **Simple & interactive interface** for easy input and selection.
   * **Live preview** of generated logos.
   * **Mobile-friendly design** for easy access.

## 

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

### **Objective :**Break down tasks for efficient development.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| Sprint 1 | Environment Setup & AI Model Integration | 🔴 High | 6 hours (Day 1) | End of Day 1 | Shanawaz | Google API Key, Python, Streamlit setup | Flask and AI model are set up and working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Member 2 | API response format finalized | Basic HTML/CSS form for user input |
| Sprint 2 | AI Processing & Logo Generation | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | anwar | API response, UI elements ready | Stable Diffusion generates logos based on user input |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Member 1&4 | API logs, UI inputs | Ensure smooth functionality & handle invalid inputs |
| Sprint 3 | UI/UX Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | mohammad | API response, UI layout completed | Improve interface & add download button |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour (Day 2) | End of Day 2 | Entire Team | Working prototype | Fully functional & deployable 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:**

Develop and implement core features of LogoCraft.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** HTML, CSS, JavaScript
   * **Backend:** Flask (Python)
   * **Programming Language:** Python
2. **Development Process:**
   * **Set up Flask backend** to process user input.
   * **Integrate Stable Diffusion** for AI-powered logo generation..
   * **Develop UI** for users to enter brand details.
   * **Optimize image quality** for professional results.
   * **Enable downloads** for generated logos.
3. **Challenges & Fixes:**
   * **Challenge:** Slow AI response times  
      **Fix:** Optimize prompt processing & reduce model load

|  |
| --- |
|  |

* + **Challenge:** Handling invalid inputs  
     **Fix:** Implement input validation

## **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 Testing | Enter brand details & generate logo | Logo is generated successfully | ✅ Passed | shanwaz |
| TC-002 | UI Testing | Check if user input form works correctly | Form accepts input properly | ✅ Passed | anwar |
| TC-003 | Performance Testing | AI response time under 5 seconds | Logos align with input description | ⚠ 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 | Host app on Flask server | App is 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**