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

# Hackathon Project Phases Template

**Project Title:**

**Logo Craft**: Innovative Logo Generation with Diffusion Technology

**Team Name:**

Crafters..

**Team Members:**

* Vijaya sree Mallikanti
* Dharaneni Rohith Rao
* Yashilaa M
* Jonnalagadda Vedha Sree
* BERUNADH NAGA PAVAN KORUMILLI

## Phase-1: Brainstorming & Ideation

**Objective:**

To develop Logo Craft, an AI-powered logo generation tool utilizing Diffusion Technology, enabling users to create bespoke, visually compelling logos through descriptive input.

**Key Points:**

1. **Problem Statement:**

 Many individuals and businesses encounter challenges in acquiring high-quality, unique logos due to the limitations of template-based generators and the expense of professional design services.

 There is a need for an accessible tool that produces original, customized logos tailored to individual brand identities.

1. **Proposed Solution:**

 An application, Logo Craft, leveraging AI and Diffusion Technology, will generate distinct, non-template-based logos from user-provided descriptions.

 The application will feature interactive customization, allowing users to refine and enhance AI-generated designs with real-time suggestions based on branding principles.

1. **Target Users:**

 Entrepreneurs, small business owners, and individuals requiring professional-grade logos.

 Users seeking cost-effective and efficient logo creation solutions.

 Those wanting a unique, and non-template based logo.

1. **Expected Outcome:**

 A functional, user-friendly application that delivers original, high-quality logos.

 The application will democratize logo design, making it accessible to a wide range of users.

 Users will be able to easily customize, and refine their logo

## Phase-2: Requirement Analysis

**Objective:**

To figure out exactly what Logo Craft needs to be built, both the techy stuff and what it will do for the user.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python.**

○ Backend: **FastAPI, Pydantic, Uvicorn, Requests, Transformers, Diffusers, Torch, Accelerate.**

○ Frontend: **Streamlit, Pillow.**

○ Database: **Not required initially (API-based queries).**

1. **Functional Requirements:**

 **Logo Creation:** Let users type in descriptions of what they want their logo to look like and use Diffusion Technology to make it.

 **Show the Logos:** Display the logos in a way that's easy to see and understand.

 **Let Users Change Things:** Allow users to tweak and change the logos until they're perfect. Give them suggestions on how to make them even better.

 **Easy to Use:** Make sure anyone can use Logo Craft, even if they don't know a lot about design.

1. **Constraints & Challenges:**

 **Making Sure the AI Works Well:** We need to make sure the Diffusion Technology is fast and makes good logos.

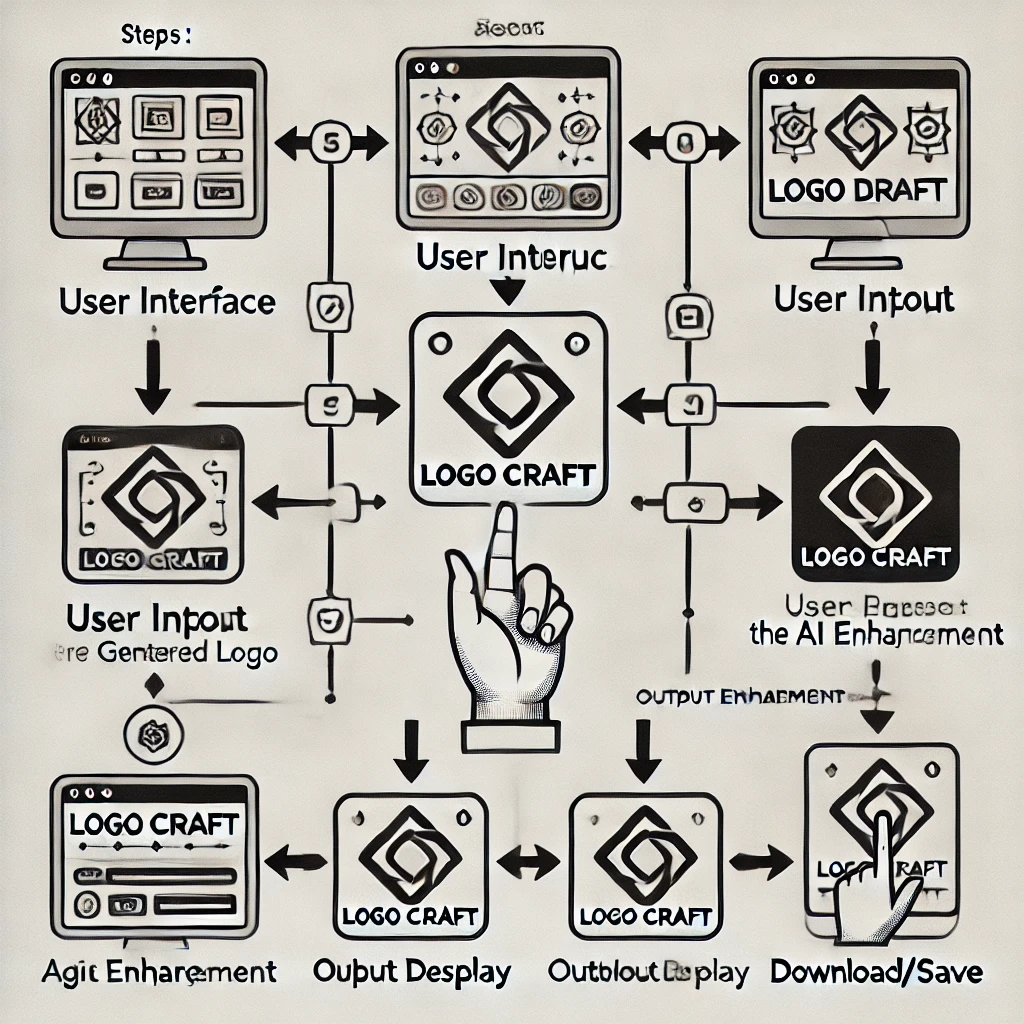
 **Keeping the Website Fast:** We want to make sure the website doesn't get slow when a lot of people are using it.

 **Making it User Friendly:** We have to make sure that the website is easy for anyone to understand and use.

## Phase-3: Project Design

**Objective:**

To plan how Logo Craft will be built and how people will use it.



**Key Points:**

1. **System Architecture:**

 **User’s Idea:** You type in what kind of logo you want (like "a playful robot logo").

 **AI Artist:** Logo Craft’s AI (using Diffusion Technology) gets your idea.

 **Magic Happens:** The AI creates a logo based on your description.

 **Showtime:** The logo shows up on the screen for you to see.

1. **User Flow:**

 **Step 1: Tell Logo Craft:** You type in what you want your logo to look like (e.g., "a cool dragon logo for my gaming channel").

 **Step 2: AI Does Its Thing:** Logo Craft’s AI gets to work and creates a logo.

 **Step 3: See Your Logo:** The logo shows up on the screen, and you can see it!

1. **UI/UX Considerations:**

 **Simple and Clean:** We'll make the app easy to use, like a clean and organized clubhouse.

 **Easy to Change Things:** You'll be able to easily change colors, shapes, and other things on your logo.

 **Light or Dark:** You can choose if you want the app to be bright or dark, depending on what you like.

## 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 | Preparing Template for the project | 🔴 Medium | 4 hours  (Day 1,2) | Mid-Day 2 | Rohith | ChatGPT  And  Gemini | Idea of Template |
| Sprint 2 | Prerequisites  of the  project | 🟡  Medium | 4 hours  (Day 1) | Day 1 | Vedha | PyTorch | Enabling training and deploying diffusion models |
| Sprint 3 | Prerequisites  of the  project | 🔴 Medium | 4 hours  (Day 1) | Day 1 | Vijaya | Diffusers  And  matplotlib | Visualizing logo variations and transformations |
| Sprint 4 | Building the solution  (frontend) | 🔴 High | 7 hours  (Day 1,2) | Mid-Day 2 | Yashilaa | Streamlit  And  safetensors | User can enter a prompt |
| Sprint 5 | Building the solution  (backend) | 🟡  High | 7 hours  (Day 1,2) | Mid-Day 2 | Pavan | Huggingface hub  And  accelerate | Accepting text prompts and customizing parameters |

### **Sprint Planning with Priorities**

#### **Sprint 1 – Template Preparation (Day 1 & 2)**

🟡 **Medium Priority** - Preparing Template for the project (Rohith)  
➡ Dependencies: ChatGPT and Gemini  
➡ Expected Outcome: Idea of Template

#### **Sprint 2 – Prerequisites (Day 1)**

🟡 **Medium Priority** - Setting up prerequisites for the project (Vedha)  
➡ Dependencies: PyTorch  
➡ Expected Outcome: Enabling training and deploying diffusion models

#### **Sprint 3 – Prerequisites Continued (Day 1)**

🟡 **Medium Priority** - Visualizing logo variations and transformations (Vijaya)  
➡ Dependencies: Diffusers and Matplotlib  
➡ Expected Outcome: Visualizing logo variations and transformations

#### **Sprint 4 – Frontend Development (Day 1 & 2)**

🔴 **High Priority** - Building the solution (Frontend) (Yashilaa)  
➡ Dependencies: Streamlit and Safetensors  
➡ Expected Outcome: User can enter a prompt

#### **Sprint 5 – Backend Development (Day 1 & 2)**

🔴 **High Priority** - Building the solution (Backend) (Pavan)  
➡ Dependencies: Huggingface Hub and Accelerate  
➡ Expected Outcome: Accepting text prompts and customizing parameter.

## Phase-5: Project Development

**Objective:**

To make Logo Craft work! We're building the main parts of the app.

**Key Points:**

1. **Technology Stack Used:** 
   1. Programming Language: **Python.**

○ Backend: **FastAPI, Pydantic, Uvicorn, Requests, Transformers, Diffusers, Torch, Accelerate.**

○ Frontend: **Streamlit, Pillow.**

○ Database: **Not required initially (API-based queries).**

1. **Development Process:**

 **Connecting the AI:** We'll make sure the AI (Diffusion Technology) is connected to the website so it can make logos.

 **Making Logos:** We'll program Logo Craft to create logos from the words you type.

 **Helping You Change Things:** We'll make it so you can easily change the colors, shapes, and other parts of your logo.

 **Making it Fast:** We will make sure that the app works quickly.

1. **Challenges & Fixes:**

 **Problem:** Sometimes the AI takes too long to create a logo.

* **Fix:** We'll save the logos that people make a lot, so Logo Craft can show them faster next time. (Like remembering your favorite toys!)

 **Problem:** The AI can only make a certain number of logos at once.

* **Fix:** We'll make Logo Craft ask the AI for only the important parts of the logo, so it doesn't waste time. (Like only asking for the toys you really want.)

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the logo Craft program works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Examining the path of torch.classes raised | Importing Touch class in VS code | ✅ Passed | Yashilaa |
| TC-002 | Functional  Testing | Query "Motorcycle maintenance tips for  winter" | Seasonal tips should be provided. | ✅ Passed | Pavan |
| TC-003 | Performance  Testing | API response time under  500ms | API should return results quickly. | ⚠ Needs Optimization | Vijaya |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | ✅ Fixed | Rohith |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI | Vedha |

`

## Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**

