# **Hackathon Project Phases Template**

## **Project Title:**

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

## **Team Name:**

Bug Busters

## **Team Members:**

* Muluktla Jahnavi
* Kemidi Manish
* Morla Manish
* Sahu Amritha Kumari

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

### **Objective:**

LogoCraft aims to create professional logo designs by using diffusion technology to craft custom logos that reflect end-user descriptions. This project helps achieve the creation of unique logos that represent their brand values and vision.

### **Key Points:**

1. **Problem Statement:**
   * In today's competitive market, establishing a memorable brand identity through a captivating logo is crucial for businesses.
   * However, many companies face challenges in creating unique and compelling logos that accurately represent their brand values and vision.
2. **Proposed Solution:**
   * Addressing the issue by leveraging cutting-edge Diffusion technology to generate custom logos based on user-provided descriptions.
   * Empowering businesses of all sizes to effortlessly create professional-grade logos that resonate with their target audience.
3. **Target Users:**
   * **Entrepreneurs and Startups**: Individuals starting new businesses who need a unique and professional logo to establish their brand identity.
   * **Freelance Designers**: Freelancers who want to leverage AI technology to deliver high-quality logos to their clients.
   * **Charities and NGOs**: Non-profits that need professional logos to effectively communicate their mission and attract donors, volunteers, and supporters.
4. **Expected Outcome:**
   * An **AI-powered logo generation app** that generates innovative logos based on the user description.

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

### **Objective:**

Define the technical and functional requirements for the LogoCraft .

### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **PyTorch in Google Colab**
   * Frontend: **React.js**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * **User Authentication**: Ensure the user is logged in via the Hugging Face CLI to access the model.
   * **Prompt Handling**: Allow users to input text prompts for generating images.
   * **Image Generation**: Utilize the Stable Diffusion model to generate images based on user prompts.
   * **Image Saving**: Provide the capability to save generated images locally.
3. **Constraints & Challenges:** **Resource Limitations**: Generating images with the Stable Diffusion model requires significant computational resources, such as a powerful GPU.

 **Prompt Sensitivity**: The quality and relevance of the generated images heavily depend on the clarity and specificity of the input prompts.

 **Authentication Dependencies**: Access to the Stable Diffusion model requires a valid authentication token from Hugging Face, which can be an obstacle if not managed properly.

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

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * User enters a text prompt.
   * The Stable Diffusion model is loaded from the Hugging Face repository using the StableDiffusionPipeline class.
   * This model is then moved to the GPU for efficient processing.
   * The input prompt is processed by the Stable Diffusion model to generate an image. The model performs inference steps to produce the final image based on the prompt.
   * The generated image is either saved locally using the PIL library or displayed directly if the code is run in an environment like Google Colab.
2. **User Flow:**
   * Step 1: U**ser Authentication**: The user logs in via the Hugging Face CLI.
   * Step 2: P**rompt Submission**: The user inputs a text prompt to generate an image.
   * Step 3: **Image Generation**: The Stable Diffusion model processes the prompt and generates an image.
   * Step 4: **Image Handling**: The generated image is either saved locally or displayed directly.
3. **UI/UX Considerations:**
   * **Simplicity: Ensure the user interface is clean and intuitive, allowing users to easily input prompts and view results.**
   * **Feedback: Provide clear feedback on the status of image generation, such as progress indicators or success/failure notifications.**
   * **Accessibility**: **Design the interface to be accessible to all users, including those with disabilities, by following best practices for inclusive design.**

## 

## **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 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Setup project  environment | High | 1 week | 07-Feb | Team A | None | Development environment established | |  |  |  |  |  |  |  | |  | | 🔴 High | 4 hours (Day 1) | End of Day 1 | K. Manish and M. Jahnavi | none | |  | | --- | | Development  environment  established | |  | |
| Sprint 1 | |  |  | | --- | --- | | User authentication setup |  | | 🔴 High | 2 hours (Day 1) | End of Day 1 | M. Jahnavi and S.Amritha | |  |  | | --- | --- | | Project  environment  setup |  | | |  | | --- | | User login  Functionality  implemented | |  | |
| Sprint 2 | |  |  | | --- | --- | | Model integration |  | | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | M. Manish and K. Manish | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | User  Authentication  setup |  | |  | | |  | | --- | | Stable Diffusion  model integrated | |  | |
| Sprint 2 | Prompt Handling logic | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | K. Manish and  M. Manish | |  |  | | --- | --- | | Model integration |  | | |  | | --- | | Text prompts can  be processed | |  | |
| Sprint 3 | |  |  | | --- | --- | | Image generation  optimization |  | | 🔴 High | 1 hour (Day 2) | Mid-Day 2 | K. Manish and S. Amritha | |  |  | | --- | --- | | Prompt  handling logic |  | | |  | | --- | | Efficient image  Generation  implemented | |  | |
| Sprint 3 | |  |  | | --- | --- | | Image saving  functionality |  | | 🔴 High | 1 hour (Day 2) | Mid- Day 2 | Entire Team | |  |  | | --- | --- | | Image  generation  optimization |  | | |  |  | | --- | --- | | |  | | --- | | Generated  images can be  saved locally | | |

### 

### **Sprint Planning with Priorities**

### **Sprint 1 – Setup & Integration (Day 1)**

**(🔴 High Priority)** **Establish the development environment, including necessary tools and dependencies.**  
 **(🔴 High Priority) Implement user login functionality via the Hugging Face CLI.**

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

**(🔴 High Priority)** **Integrate the Stable Diffusion model from Hugging Face into the project**.  
 **(🟡 Medium Priority)** **Develop the logic for handling user text prompts to ensure they are processed correctly by the model.**

### **Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(🔴 High Priority)** **Optimize the image generation process to ensure efficient performance.**  
 **(🔴 High Priority)** **Implement the capability to save generated images locally for user access.**

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

### **Objective:**

Implement core features of the LogoCraft .

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** React.js
   * **Backend:** PyTorch
   * **Programming Language:** Python
2. **Development Process:**
   * **Sprint Planning**: Define tasks and priorities for each sprint.
   * **Implementation**: Develop and optimize features iteratively.
   * **Testing and Feedback**: Continuously test and refine based on feedback.
3. **Challenges & Fixes:**
   * **Resource Limitations: Ensure efficient usage of GPU resources and optimize performance.**

* + **Prompt Sensitivity: Provide clear guidelines for effective prompt creation to improve image quality.**
  + **Authentication Dependencies: Manage and securely store authentication tokens to ensure consistent access to the model.**
  + **Fix:** **Optimize queries to fetch** **only necessary data**.

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

### **Objective:**

Ensure that the LogoCraft App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | |  |  | | --- | --- | | User logs in via  Hugging Face CLI |  | | |  |  | | --- | --- | | User successfully  logs in |  | | ✅ Passed | K. Manish |
| TC-002 | Functional Testing | |  |  | | --- | --- | | User inputs a prompt |  | | |  |  | | --- | --- | | Prompt is correctly  received and  processed |  |   . | ✅ Passed | M. Manish |
| TC-003 | Functional Testing | |  |  | | --- | --- | | Generate image using  Stable Diffusion model |  | | |  |  | | --- | --- | | Image is generated  without errors |  | | ✅ Passed | M. Jahnavi |
| TC-004 | Performance testing | |  |  | | --- | --- | | Image generation with  15 inference steps |  |   . | Data accuracy should be improved. | ✅ Fixed | K. Manish |
| TC-005 | Performance testing | |  |  | | --- | --- | | Image generation with  200 inference steps |  | | |  |  | | --- | --- | | Image quality is high,  and generation time  is reasonable |  | | ✅ Passed | S. Amritha |
| TC-006 | Deployment Testing | |  |  | | --- | --- | | Model handling  multiple simultaneous  prompts |  | | |  |  | | --- | --- | | System handles load  without crashing |  |   . | ✅ Passed | M. Manish |