# **Hackathon Project Phases Template**

## **Project Title:**

**Open AI: Clothing Image Generator Using DALL.E**

## **Team Name:**

**Syntax Squadd**

## **Team Members:**

* G. Sai Ram
* Bilal Rafiq
* Mohammad safiullah Anas
* Mir Roshan Ali

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

### **Objective:**

Develop an AI-powered which generates realistic clothing images using AI. It helps in fashion design, virtual try-ons, and e-commerce. It creates new outfits based on text or image inputs.

### **Key Points:**

1. **Problem Statement:**

The fashion industry needs more personalized and visually appealing clothing options. Traditional online shopping lacks customized visual representations, limiting user experience. Open AI solves this by using DALL.E and a Streamlit interface to generate realistic clothing images from user descriptions, helping shoppers visualize their ideas.

1. **Proposed Solution:**

* People often struggle with choosing their dressing style and how it will look in real life.
* They have an idea in mind but they don’t get proper final design and it requires lots of time.
* Our project, **Open AI**, helps bridge the gap between imagination and reality in fashion design.
* It achieves this by integrating **DALL.E** with a **user-friendly Streamlit interface**.
* This allows users to generate realistic clothing images based on their descriptions, making fashion decisions easier.

1. **Target Users:**
   * **Fashion Designers** which helps to explore concepts of new designs.
   * **E-Commerce Platforms** : They Require high-quality product images and virtual try-ons.
   * **Marketing & Advertising Agencies** – Create fashion-related promotional content which helps them to increase their sales and customer reptations.
2. **Expected Outcome:**
   * Where it takes description from users and generates visual image, If users aren't satisfied with the generated design, they can further customize it to match their vision. We also recommend designs based on their preferences and current market trends.

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

### **Objective:**

Taking description from user and creating image according to user till they satisfied.

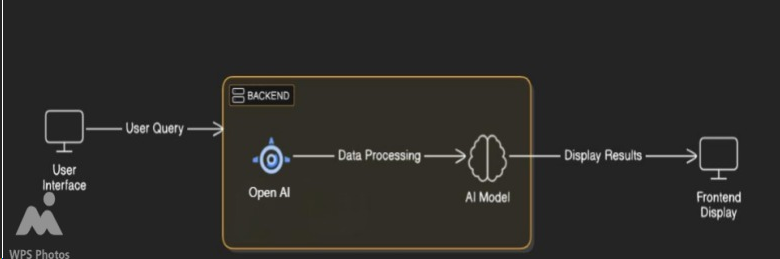
### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: Open AI
   * Frontend: **Streamlit Web Framework**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Ability to **Generate images**  using open AI API.
   * Display **clothing images** in an intuitive UI.
   * Provide **real-time Fashion Trends** based on Popular fashion.
3. **Constraints & Challenges:**
   * Ensuring real-time updates from **Open AI**.
   * Handling **API rate limits** and optimizing API calls.
   * Providing a **smooth UI experience** with Streamlit.

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

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * User enters Description of clothing via UI.
   * Query is processed using **Open API**.
   * AI model fetches and processes the data.
   * The frontend displays - image is generated according to description.
2. **User Flow:**
   * Step 1: User enters a Description of Cloths.
   * Step 2: The backend **calls the Open AI API** to retrieve description data.
   * Step 3: The Web application processes the data and **displays results** in an image format.

## **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 | Safiullah,  Roshan | Open API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Sai Ram | API response format finalized | Basic UI with input fields |
| Sprint 2 | User interference | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Bilal | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Safiuallah  Roshan | API logs, UI inputs | Improved API stability |
| Sprint 3 | User interface | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | safiullah | API 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 **Open API**.  
 **(🟡 Medium Priority)** Build a **basic UI with input fields**.

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

**(🔴 High Priority)** User Interface.  
 **(🔴 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:**

The core feature of it is image generation using user Description.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** open API
   * **Programming Language:** Python
2. **Development Process:**
   * Implement **API key authentication** and **Open API integration.**
   * Optimize **user feedback and recommendations**.
3. **Challenges & Fixes:**
   * **Challenge:** Delayed API response times.  
      **Fix:** Implement **caching** to store frequently queried results.
   * **Challenge:** Limited API calls per minute.  
      **Fix:** Optimize queries to fetch **only necessary data**.

## **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 | Query ‘Give me red shirt with lion design on right corner’ | Red shirt with lion design . | ✅ Passed | Safiullah |
| TC-002 | Functional Testing | Query "peach color shirt with HI design in center’ | What kind style should be provided. | ✅ Passed | Safiullah |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Sai ram |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | ✅ Fixed | Roshan |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Bilal |
| 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**