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

Hackathon Project Phases Template

Project Title:

CoutureAI: Clothing Image Generator Using Stable Diffusion Pipeline

Team Name:

SYNTH

Team Members:

- D.Amulya
- D.Meghana
- L.Kavyanjali
- G.Harshini

Phase-1: Brainstorming & Ideation

Objective:

The objective of **CoutureAI** is to empower users with AI-driven, realistic clothing visualization from text descriptions, enhancing personalization in fashion design and shopping.

Key Points:

1. **Problem Statement:**
    - **Lack of AI-Driven Clothing Previews** – E-commerce platforms don't offer personalized outfit visualization.
    - **Disconnect Between Ideas & Reality** – Consumers struggle to design and convey unique fashion concepts.

2. **Proposed Solution:**

    - CoutureAI uses **Stable Diffusion** to generate realistic clothing images from text descriptions, enabling users to visualize custom outfit ideas.
    - It enhances the shopping and design experience by bridging the gap between imagination and reality, empowering consumers, designers, and tailors while promoting sustainability.
    - Integrate a **personalized recommendation system** for clothing designs based on user preferences, linked to **Stable Diffusion** for custom image generation.
3. **Target Users:**

    - **Consumers –** Fashion enthusiasts seeking personalized outfit visualization.
    - **Designers & Tailors –** Professionals using AI for custom clothing creation.
    - **Fashion Brands & Retailers –** Businesses enhancing customization and engagement.
    - **E-Commerce Platforms** – Online stores integrating AI-driven fashion previews.
4. **Expected Outcome:**

    - AI-powered platform that generates realistic clothing images from text descriptions, enhancing customization, visualization, and sustainability in fashion.

    - **Recommendation system** suggesting clothing designs and where similar models are available, integrated with **Stable Diffusion** for custom image generation.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the CoutureAI.

Key Points:

1. **Technical Requirements:**

    - Programming Language: **Python**
    - Backend: **FastAPI**
    - Frontend: **Streamlit Web Framework**
    - Database: **Google Drive, Hugging Face Model Hub**
2. **Functional Requirements:**

    - Generate realistic clothing images from text descriptions using Stable Diffusion.

- Provide a seamless user experience through a Streamlit-based platform for input and visualization.
- Ensure efficient image generation and data management with PyTorch, CUDA, and cloud storage.

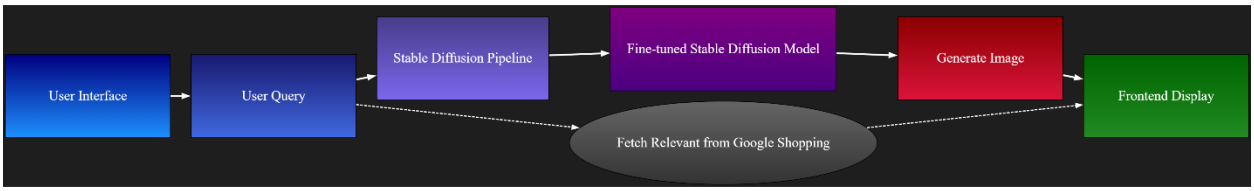
3. **Constraints & Challenges:**

   - High computational resources are required for efficient image generation, limiting scalability.
   - Accurately interpreting diverse text descriptions is challenging and requires constant refinement.
   - Real-time image generation may cause delays, especially for complex designs.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   - **User Input:** Users enter text descriptions of clothing via the UI.
   - Query is sent to **Stable Diffusion v1.5** for text-to-image generation.
   - AI model is **fine-tuned** for high-quality textures and styles.
     Frontend Display: Generated images are displayed in real-time.
     Infrastructure: Runs on **Google Colab GPUs** with **Ngrok** for online access.
2. **User Flow:**

   - **Step 1:** User enters a text prompt (e.g., *"A black leather jacket with silver zippers"*).
   - **Step 2:** AI processes the request via the Stable Diffusion pipeline.
   - **Step 3:** Fine-tuned AI model generates a **realistic** clothing image.
   - **Step 4:** The image is displayed for **preview, editing, or downloading**.

3. **UI/UX Considerations:**

    - **Minimalist & Intuitive UI** for smooth interaction.
    - **Filters for fabric, color, and style** for better personalization.
    - **Real-time generation** ensures quick and seamless fashion visualization.

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	Amulya	Huggingface API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	□ Medium	2 hours (Day 1)	End of Day 1	Meghana	API response format finalized	Basic UI with input fields
Sprint 2	Finetune the model with the clothes dataset	□ High	3 hours (Day 2)	Mid-Day 2	Kavyanjali	Image Generation and product recommendations	Image generation functionality with recommendations
Sprint 2	Error Handling & Debugging	□ High	1.5 hours (Day 2)	Mid-Day 2	Harshini	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	□ Medium	1.5 hours (Day 2)	Mid-Day 2	Amulya & Meghana	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) Finetune Stable Diffusion model with the **DeepClothes Datset**
(□ Medium Priority) Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(High Priority) Implement **Image generation and product recommendations**
(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 CoutureAI App.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Fined Tuned Stable Diffusion Model, SerpAPI for Google Shopping API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **Huggingface access key authentication** and **SerpAPI integration**.
    - Develop **finetuned Stable Diffusion model for image generation**.
    - Optimize **model for performance and relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** Delayed image generation response times.
      Fix: Implement **caching** to store frequently queried results.
    - **Challenge:** Limited API calls to serpAPI.
      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 "A red Satin dress with bell sleeves"	An image of the expected red satin dress	✅Passed	Amulya
TC-002	Functional Testing	Query "A blue and black denim coord set"	Expected image is generated	✅Passed	Meghana
TC-003	Performance Testing	Image Generation under 5seconds	API should return results quickly.	⚠ 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.	Validated.	Tester 2
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**