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

Hackathon Project Phases Template

Project Title:

CoutureAI: Clothing Image Generator Using Stable Diffusion Pipeline.

Team Name:

InnoQuads

Team Members:

- Nithya Jogannagari
- Sania
- Sowmya Medipelly
- Vaishnavi P

Phase-1: Brainstorming & Ideation

Objective:

CoutureAI generates realistic clothing images from structured text descriptions using Generative AI. It helps users visualize and refine their fashion ideas before purchase or tailoring. This tool enhances personalization and bridges the gap between imagination and reality in fashion.

Key Points:

1. **Problem Statement:**
    - Finding custom clothing that matches personal style can be challenging, as traditional shopping lacks visualization
    - CoutureAI solves this by generating realistic outfit images from text descriptions, helping users see their designs before purchase or tailoring.
2. **Proposed Solution:**

- CoutureAI generates realistic clothing images from text descriptions using Generative AI. This helps users visualize and refine their fashion ideas before purchase or tailoring.

3. **Target users:**
    - **Fashion Enthusiasts** – Individuals who want to visualize and refine their custom clothing ideas.
    - **Designers & Tailors** – Professionals seeking AI-generated references for bespoke creations.
    - **E-commerce & Retailers** – Online stores looking to offer personalized fashion previews.
    - **Fashion Startups** – Businesses aiming to integrate AI-driven customization into their services.
4. **Expected Outcome:**
    - CoutureAI generates realistic clothing images based on structured text descriptions provided in the prompt. This enables users to visualize, customize, and refine their fashion ideas before purchase or tailoring.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the CoutureAI App.

Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: Google Colab, Diffusers' Stable Diffusion Pipeline, Torch & Transformers
    - Frontend: IPython Display
    - Database: No traditional database required since images are generated on demand.
2. **Functional Requirements:**
    - **Generate realistic clothing images** from text descriptions using Generative AI.

- Allow users to prompt detailed information to refine and customize designs.
- Allow **Support multiple fashion styles and customization options** (e.g., color, fabric, fit)
- Enable users to save, compare, and download generated outfits.

3. **Constraints & Challenges:**
   - **Ensuring high-quality and realistic image generation** within a reasonable time frame.
   - **Optimizing computational resources** for Stable Diffusion to run efficiently on Google Colab.
   - **Handling user input variations** (e.g., vague descriptions, conflicting style requests).
   - **Providing a streamlined interaction experience** using Google Colab's IPython Display for seamless output presentation.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:

1. **System Architecture:**
   - User enter a clothing description.
   - The system extracts key attributes from the text for AI inference.
   - The AI model generates a realistic clothing image based on input prompts.
   - The generated clothing image is displayed for user interaction, modifications, or downloads.
   - Generated images and queries are stored in a cloud database for future access.
2. **User Flow:**
   - Step 1: **User Input** – The user enters a clothing description (e.g., "A blue denim jacket with embroidered patterns") .

- Step 2: **AI Processing** – The backend processes the input and sends it to the Stable Diffusion model for image generation.
- Step 3: **Image Generation & Display** – The AI model generates a realistic clothing image, which is displayed on the UI for user review and refinement.

3. **UI/UX Considerations:**
   - **User-Friendly Interface –** A clean, intuitive design for seamless navigation and effortless interaction.
   - Users can **filter** generated images by selecting fabric, colour, style, and patterns.
   - **Dark & light mode** for better user experience.

Phase-4: Project Planning

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & Model Integration	🔴 High	8 hours (Day 1)	End of Day 1	Sania &Nithya Jogannagari	Python, Diffusers, Stable Diffusion setup	AI model integrated & ready to process inputs
Sprint 1	Frontend IPython display Development	🟡 Medium	3 hours (Day 2)	Mid-Day 2	Nithya Jogannagari	Image generation API response format finalized	Basic UI with input fields & upload options
Sprint 2	Text-to-Image Generation	🔴 High	4 hours (Day 1)	Mid-Day 2	Vaishnavi P	AI model setup, UI components ready	Generate realistic outfit images

							from descriptions
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Sowmya Medipelly	API logs, UI inputs	Improved stability, reduced model errors
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Sowmya Medipelly	AI model responses, UI layout completed	Responsive UI, smoother user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	Mid-Day 2	Nithya Jogannagari	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)** Model Integration
- (🟡 **Medium Priority)** Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

- (🔴 **High Priority)** Implement Text-to-Image Generation.
- (🔴 **High Priority)** Debug API issues & handle **errors in queries**.

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

- (🟡 **Medium Priority)** Test API responses, refine,debug.
- (🟢 **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:** IPython Display (used in Google Colab to show images)
    - **Backend:** Google Colab (Jupyter Notebook environment)

      Diffusers' Stable Diffusion Pipeline (for AI-based image generation)

      Torch & Transformers (for deep learning & model execution)
    - **Programming Language:** Python
2. **Development Process:**
    - Implement AI model integration using Stable Diffusion via Diffusers.
    - Develop text-to-image generation logic for realistic fashion visualization.
    - Enable customization options (e.g., modify fabric, color, fit) in generated designs.
    - Optimize AI processing to ensure quick and high-quality image generation.
    - Enhance IPython Display for seamless user interaction with input fields and result display.
3. **Challenges & Fixes:**
    - **Challenge:** Slow image generation time.
      Fix: Optimize model parameters and use Google Colab's GPU acceleration.
    - **Challenge:** Inconsistent or inaccurate clothing designs.
      Fix: Fine-tune input prompts and enhance text preprocessing for better AI interpretation.
    - **Challenge:** High computational load during multiple requests.
      Fix: Implement request queuing and optimize Stable Diffusion model execution.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the CoutureAI App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Input: "Generate a red evening gown with floral patterns.""	A realistic red evening gown with floral patterns is generated.	✅ Passed	Sowmya Medipelly
TC-002	Functional Testing	Upload a reference image and modify color to "blue."	The AI generates a similar outfit with a blue color.	✅ Passed	Vaishnavi P
TC-003	Performance Testing	Image generation time under 10 seconds.	AI should generate images within the expected time.	✅ Passed	VaishnaviP
TC-004	Bug Fixes & Improvements	Fix inaccurate outputs for vague descriptions..	AI should generate more relevant images..	✅ Fixed	Nithya Jogannagari & Sania
TC-005	Final Validation	Ensure UI works across devices (mobile & desktop)..	IPython Display should be fully responsive.	✅ Passed	Nithya Jogannagari & Sania
TC-006	Deployment Testing	Host the app using Google Colab	The app should be accessible online.	Working on it.	Sowmya Medipelly

---

Final Submission

1. Project Report Based on the templates
2. Demo Video (3-5 Minutes)
3. GitHub/Code Repository Link
4. Presentation