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

Hackathon Project Phases Template

Project Title:

Clothing image generator using stable diffusion pipeline

Team Name:

Fashion intelligence

Team Members:

- Varshini Jagirapu
- Keerthana Bathula
- Venkata Kalyani

Phase-1: Brainstorming & Ideation

Objective:

The objective of a clothing image generator using the Stable Diffusion pipeline is to create high-quality, realistic, and diverse images of clothing items. This can be applied in various domains, such as fashion design, e-commerce, virtual try-ons, and marketing.

Key Points:

1. **Problem Statement:**
    - Difficulty in generating realistic textures, folds, lighting and fabric details.
    - Limited flexibility to adapt to rapidly changing fashion trends.
2. **Proposed Solution:**
    - Generate high-quality, realistic, and diverse clothing images that match the input descriptions.

- By solving this, the proposed system will empower fashion designers, e-commerce platforms, and marketers to create visually appealing.

3. **Target Users:**

   - **Reduce costs** associated with traditional photoshoots and sample production.
   - **Create unique and trendy clothing visuals** for social media content.
   - **Provide students in fashion design and related fields** with a tool for learning and experimentation.

4. **Expected Outcome:**

   - The **"Clothing image generator using stable diffusion pipeline"** gives High resolution images, Cost and Time efficiency, Enhanced creativity and innovation.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the AutoSage App.

Key Points:

1. **Technical Requirements:**

   - Programming Language: **Python**
   - Backend: **Google colab**
   - Database: **Stores prompts and Generated images**

2. **Functional Requirements:**

   - Input handling.
   - Image generation.
   - Model processing, Output and Exporting.
   - Integration and Extensibility.

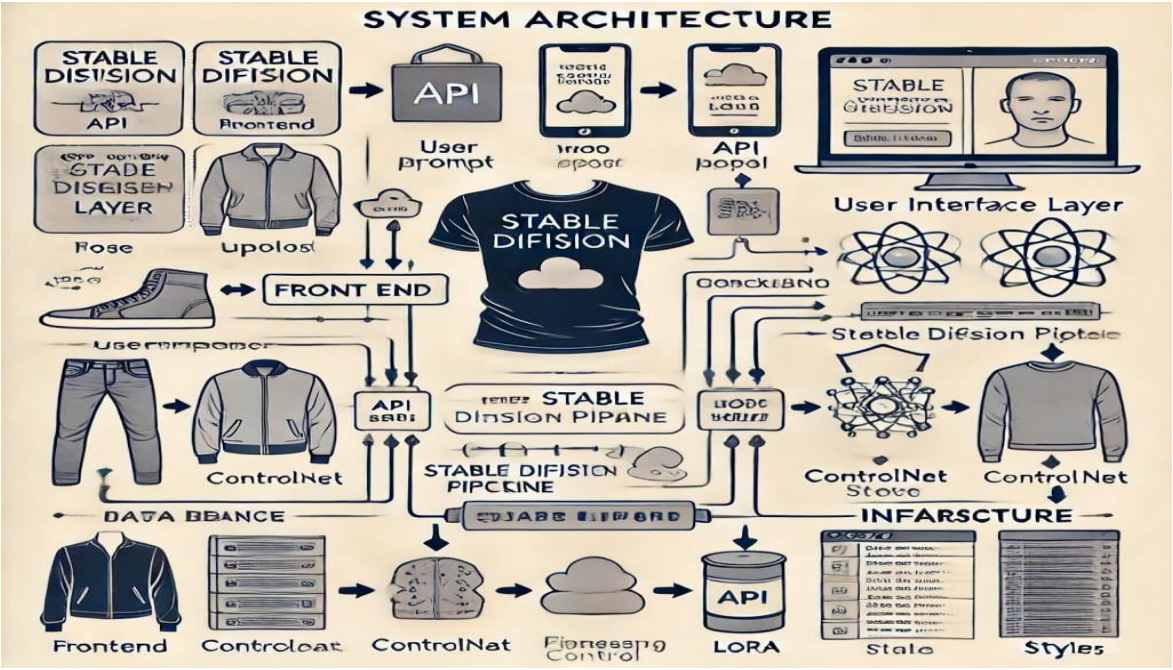
3. **Constraints & Challenges:**

   - The model could unintentionally generate images resembling copyrighted designs.
   - Real-world Integration.
   - Edge Artifacts and blurriness can occur in complex designs.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - User enters stable diffusion text prompt .
    - Query is processed using **Google colab API**.
    - AI model fetches and generates the image.
    - The backend compresses and formats the output.
2. **User Flow:**
    - Step 1: User enters text prompt. (e.g., "A red silk evening gown with gold embroidery").
    - Step 2: The backend **validates input, formats it, and send it to the AI model.**
    - Step 3: Users can refine prompts, regenerate, or download the image.
3. **UI/UX Considerations:**
    - **Provide a quick tutorial or guided onboarding for first-time users**

○ **Provide a real-time progress indicator while stable diffusion processes the image.**
○ **Dark & light mode** for better user experience.

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	varshini	Google API Key, Python, google colab setup.	API connection established & working
Sprint 1	Backend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Venkata Kalyani	API response format finalized	Basic UI with prompt fields
Sprint 2	prompt Search & Comparison	🟢 High	3 hours (Day 2)	Mid-Day 2	varshini	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🟢 High	1.5 hours (Day 2)	Mid-Day 2	Varshini& Keerthana.	API logs, prompts.	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Keerthana& Venkata Kalyani.	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	J. varshini, B.keerthana, ch. Venkata kalyani	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 **Google API**.

(🟡 Medium Priority) Build a **basic UI with prompt fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(🟢 High Priority) Implement **search & comparison functionalities**.

(🟢 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 Clothing Image Generator.

Key Points:

1. **Technology Stack Used:**
    - **Backend:** Google colab API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Google API integration**.
    - Enhance Model with Control Mechanisms.
    - Optimize **search queries for performance and relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** AI may struggle to generate sharp, detailed fabric textures.
      Fix: use Real- ESRGAN for Image upscaling and Enhancement.
    - **Challenge:** Hard to maintain brand consistency across designs.
      Fix: Use control net (pose guidance) to maintain structure across multiple views.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the clothing image generator works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "A fashion designer wants to visualise new outfit ideas"	The designer gets a realistic clothing concept within minutes.	☑ Passed	Tester 1
TC-002	Functional Testing	Query "a movie studio needs quick costume design mockups before finalising"	Saves time in costume brainstorming sessions.	☑ Passed	Tester 2
TC-003	Performance Testing	API response time under 1 minute.	API should return results quickly.	☑ Passed	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.	☑ Fixed	Tester 2
TC-006	Deployment Testing	By testing, ensures clothing image generator.	Image generator should be accessible online.	☑ Fixed	DevOps

---

Final Submission

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