Hackathon Project Phases Template for the Couture AI: Clothing image generator project.

Hackathon Project Phases Template

Project Title:

Couture AI: Clothing image generator using stable diffusion pipeline

Team Name:

Team NARUTO

Team Members:

- Varshitha
- Laharika
- Sneha

Phase-1: Brainstorming & Ideation

Objective:

Generate a range of stylish, high-resolution images of clothing items such as dresses, jackets, and pants in various styles (e.g., casual, formal) to be used in an online clothing store's product catalog.

Key Points:

1. **Problem Statement:**
    - In the fashion industry, creating high-quality, diverse, and visually appealing clothing images is crucial for e-commerce platforms, marketing, and design prototyping. However, generating these images traditionally requires physical photography, fashion design expertise, and significant time and effort.

- Additionally, there is a growing demand for creating virtual clothing images that can be customized to various styles, seasons, and cultural contexts, catering to abroad and diverse audience.

2. **Proposed Solution:**

   - An image generator using Gemini Flash to provide real-time Cloth designs and sizes based on the user text.
   - The image generator offers eco-friendly designs insights based on user preferences.

3. **Target Users:**

   Fashion Designers
   Marketing and Advertising Agencies
   Personal Shoppers and Stylist Services
   AI-Driven Clothing Brands.

4. **Expected Outcome:**

   Faster, more cost-effective clothing design and marketing processes. Expansion into new market like virtual clothing, gaming, and personalized fashion. Improved user satisfaction and feedback-driven Ai mode improvements. Creation of sustainable, innovative brand identity.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Couture AI: clothing image generator

Key Points:

1. **Technical Requirements:**

   - **Programming Language**: Python
   - **Backend**: Node JS, Flash API**.**
   - **Frontend**: HTML
   - **Database**: Not required initially (API-based queries)

2. **Functional Requirements:**

   - Ability to fetch outfit details using Gemini Flash API.
   - Customizable image attributes.
   - Provides types of clothes designs based on seasons.

- Allow users to search eco-friendly cloth designs based on emissions and user text.

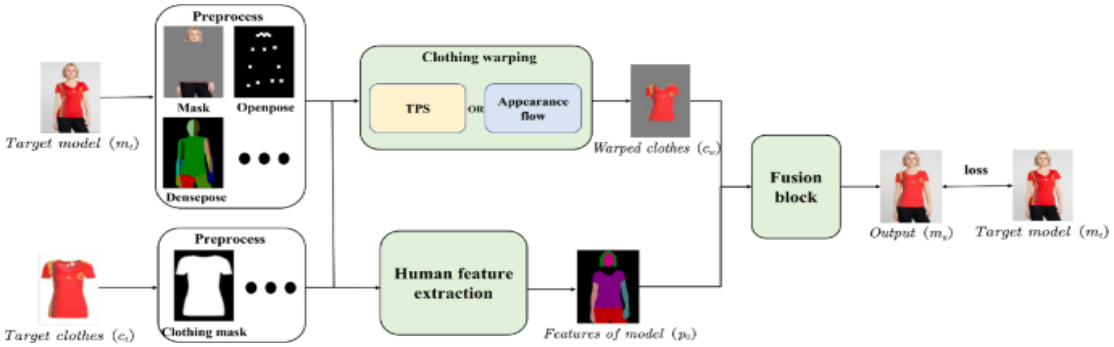
3. **Constraints & Challenges:**

   - Integration with E-commerce and Design platforms.
   - Handling API rate limits and cost control.
   - Providing a smooth scalability and performance**.**

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   - User enters the text design they want to.
   - image is processed using Google collab.
   - AI model fetches and processes the image.
   - The frontend displays the image containing a boy or a girl wearing a dress.

2. **User Flow:**

   - Step 1: User enters a text (e.g., "A girl wearing a denim skirt and a top").
   - Step 2: The backend calls the Gemini Flash API to collect data from users.
   - Step 3: The app processes the data and displays results in an easy-to-image format.

3. **UI/UX Considerations:**

   - Minimalist, user-friendly interface for seamless navigation.
   - Filters for Price, Size, color, and Fabrics.
   - Dark & light mode for better user experience.

- Color contrast and readable fonts.

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	Varshitha	Google API Key, Python, Stream lit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Member 2	API response format finalized	Basic UI with input fields
Sprint 2	Design Search & Comparison	🔴 High	3 hours (Day 2)	Mid-Day 2	Sneha	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Member 1&3	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Laharika	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 Google Gemini API.
(🟡 **Medium Priority)** Build a basic UI with input 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 Couture AI: Clothing image generator

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Stream lit
    - **Backend:** Google Gemini Flash API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement API key authentication and Gemini API integration.
    - Developing a high-quality, scalable AI clothing generator.
    - Optimize search text for designs and outfits.
3. **Challenges & Fixes:**
    - **Realism and texture Quality**
      Challenge: Generating realistic fabric textures, and materials.
      Fix: Use high-resolution training data with diverse fabric types and lighting conditions.
    - **Challenge:** Capturing the intricate details of different fabric types can be difficult.

      Fix: Use high-resolution texture datasets and tarin the model on fabric-specific datasets with photorealistic detail.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Couture AI: clothing image generator works as expected.

Test	Category	Test Scenario	Expected Outcome	Status	Tester

Case ID					
TC-001	Functional Testing	Query "Stylish red jacket with gold embroidery "	Relevant budget cars should be displayed.	✅ Passed	Varshitha
TC-002	Functional Testing	Query "outfits based on the user text"	Seasonal tips should be provided.	✅ Passed	Sneha
TC-003	Performance Testing	API response time under 500ms	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.	❌ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app using Stream lit 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.