
Hackathon Project Phases Template

Project Title:

CoutureAI: Clothing Image Generator Using Stable Diffusion Pipeline

Team Name:

GENGUILD

Team Members:

- Uppara Uday Sagar
- N.Dinesh
- U.Sai Charan
- S.Ajay
- Vuthuri Sai Sharan

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered tool that automates the creation of clothing designs, reducing the time and effort required by fashion designers.

Key Points:

1. Problem Statement:

- The fashion industry is rapidly evolving, with increasing demand for personalized, innovative, and sustainable clothing designs. Designers and brands often face challenges in creating unique, high-quality designs efficiently.
- Traditional design processes are time-consuming, resource-intensive, and may lack the creativity needed to stand out in a competitive market. Additionally, small businesses and independent designers may not have access to advanced tools or resources to generate professional designs.

2. Proposed Solution:

- The proposed system not only streamlines the design process but also promotes sustainability by reducing material waste associated with traditional.
- The Stable Diffusion model, known for its ability to produce detailed and coherent images from textual descriptions, is fine-tuned specifically for the fashion domain, ensuring that the generated clothing images are both visually appealing and contextually relevant.

3. Target Users:

- Fashion Designers
- E-Commerce platforms
- Custom Clothing Brands

4. Expected Outcome:

- The primary outcome is the generation of photorealistic or stylized images of clothing items, including dresses, shirts, pants, accessories, and more.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the **CoutureAI**

Key Points:

1. Technical Requirements:

- Stable Diffusion Pipeline Integration
- Hardware Requirements
- Software Stack
- Dataset ○ Model

2. Functional Requirements:

- Prompt-Based Generation
- Customization Options
- Search and Filter:
- Feedback Mechanism

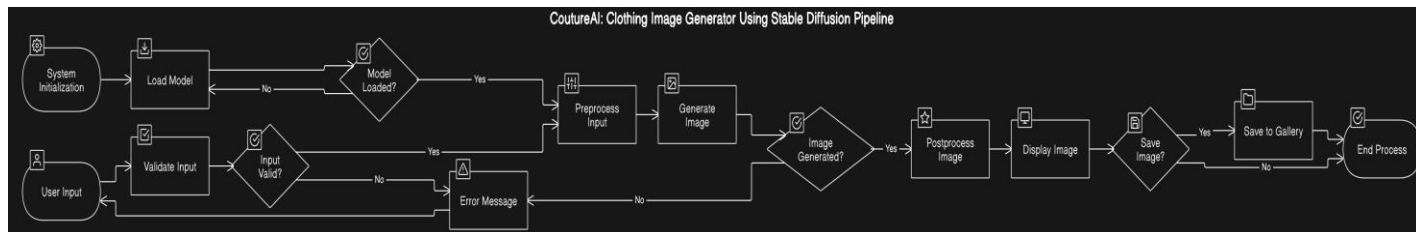
3. Constraints & Challenges:

- Generating realistic clothing images requires a diverse, high-quality dataset of fashion items. Curating such a dataset can be time-consuming and expensive.
 - Adapting Stable Diffusion for fashion-specific tasks requires fine-tuning, which demands significant computational resources and expertise.
 - Users may provide vague or abstract prompts.
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

Core Functionality:

- Image Generation: The primary purpose is to generate clothing images using the Stable Diffusion pipeline.
- User Input: The system accepts user input (likely text prompts or image references) to guide the generation process.
- Model Loading: It requires a pre-trained model to be loaded for the Stable Diffusion process.
- Preprocessing: Input data undergoes preprocessing before being fed to the model.
- Postprocessing: The generated image might be further processed (e.g., upscaling, refinement) after the initial generation.
- Display and Saving: The final image is displayed to the user and can be saved to a gallery. **Workflow and Components:**

- System Initialization: The system starts by initializing its components.
- Load Model: The system attempts to load the necessary Stable Diffusion model.
- Model Loaded? (Decision Point): If the model fails to load, the process likely stops or displays an error.
- User Input: The user provides input to guide the image generation.
- Validate Input: The system checks if the user input is valid.
- Input Valid? (Decision Point): If the input is invalid, an error message is displayed.
- Error Message: Displays errors to the user, potentially for model loading or invalid input.
- Preprocess Input: The input is prepared for the Stable Diffusion model.
- Generate Image: The Stable Diffusion pipeline generates the clothing image.
- Image Generated? (Decision Point): If the image generation fails, the process likely stops or displays an error.
- Postprocess Image: The generated image is refined or enhanced.
- Display Image: The final image is shown to the user.
- Save Image? (Decision Point): The user is given the option to save the image.
- Save to Gallery: The image is saved if the user chooses to.
- End Process: The system finishes its operation.

Key Design Considerations:

- Stable Diffusion Pipeline: The system leverages the power of Stable Diffusion, a state-of-the-art text-to-image generation model.
- User Interface: The design implies a user interface with buttons and feedback mechanisms.
- Error Handling: The system includes error checks and messages to guide the user.
- Efficiency: The design aims to streamline the process from input to output.

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	Uday sagar	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	Dinesh	API response format finalized	Basic UI with input fields
Sprint 2	Backend Development	High	3 hours (Day 2)	Mid-Day 2	Sai Charan	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	Ajay	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	Sharan	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 AutoSage App.

Key Points:

- 1. **Technology Stack Used:**
 - **Frontend:** Streamlit
 - **Backend:** Google Gemini Flash API
 - **Programming Language:** Python
- 2. **Development Process:**
 - Implement **API key authentication** and **Gemini API integration**.
 - Develop **vehicle comparison and maintenance tips logic**.
 - Optimize **search queries for performance and relevance**.
- 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	Generate a dress design based on the prompt "Elegant evening gown with floral patterns"	The AI should generate a high-quality dress image matching the description	Pending	Tester 1
TC-002	Functional Testing	Generate an outfit based on a vague prompt like "casual wear"	The AI should generate a reasonable casual outfit suggestio	Pending	Tester 2
TC-003	Performance Testing	Measure image generation time for different prompts	Imageshould begenerated in under 5 seconds	Needs Optimization	Tester 3

TC-004	Bug Fixes & Improvements	Validate that AI-generated images are fashion-relevant	The generated clothing images should be realistic and wearable	Pending	Developer
TC-005	Security Testing	Test unauthorized API access attempts	Unauthorized users should not be able to access the AI model	Passed	Tester 2
TC-006	Deployment Testing	Host the AI model on the cloud and ensure accessibility	The AI tool should be accessible online without errors	Deployed	DevOps

Final Submission

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