Hackathon Project Phases Template

Project Title: : CoutureAI: Clothing Image Generator Using Stable Diffusion Pipeline

Team Name: AI Stylers

Team Members:

- **B.Paramesh**
- B.Rajesh
- K.Sai pranav
- G.venu

Phase-1: Brainstorming & Ideation

- **Objective:**
-
- To create an AI system that leverages the Stable Diffusion Pipeline to efficiently generate diverse and realistic clothing designs, aiding fashion designers and retailers in their creative processes.

- **Key Points:'**

- **Problem Statement:**
  - Develop an AI-based system that leverages the Stable Diffusion Pipeline to generate diverse and realistic clothing designs efficiently and effectively, addressing the challenges faced by designers and retailers in the fast-paced fashion industry

- **Proposed Solution:**
  - Creating an AI system that leverages the Stable Diffusion Pipeline to generate realistic clothing designs based on user inputs such as fabric, color, and style

preferences. This system will help fashion designers and retailers quickly visualize and create a variety of high-quality clothing designs that align with current trends and consumer preferences, making the design process more efficient and resource-effective.

- **Target Users:**
    - The AI system targets fashion designers, retailers, manufacturers, e-commerce platforms, consumers, fashion schools, and marketing agencies. It helps them quickly visualize and create high-quality clothing designs based on user inputs like fabric, color, and style preferences, making the design process more efficient and tailored to current trends. This tool can revolutionize the fashion industry by streamlining design processes and providing customizable, trendy options.
- **Expected Outcome:**

The expected outcome of this AI system is to provide fashion designers, retailers, manufacturers, e-commerce platforms, consumers, fashion schools, and marketing agencies with an efficient tool to visualize and create high-quality clothing designs based on user inputs like fabric, color, and style preferences. This system aims to streamline the design process, offer customizable and trendy options, reduce the need for physical prototypes, and ultimately revolutionize the fashion industry by making it more resource-effective and aligned with current trends and consumer preferences.

○

Phase-2: Requirement Analysis

Objective:The requirement analysis aims to understand what stakeholders need from the AI system, define its essential features, set technical and performance standards, ensure it aligns with current fashion industry trends, and create a detailed plan to guide the development process. This helps ensure the final product meets user demands effectively and efficiently.

- **Key Points:**
- **Technical Requirements:**
- Programming Language:Python
- Frontend: HTML, CSS
- Backend: Stable Diffusion Pipeline
- Database: API based Queries

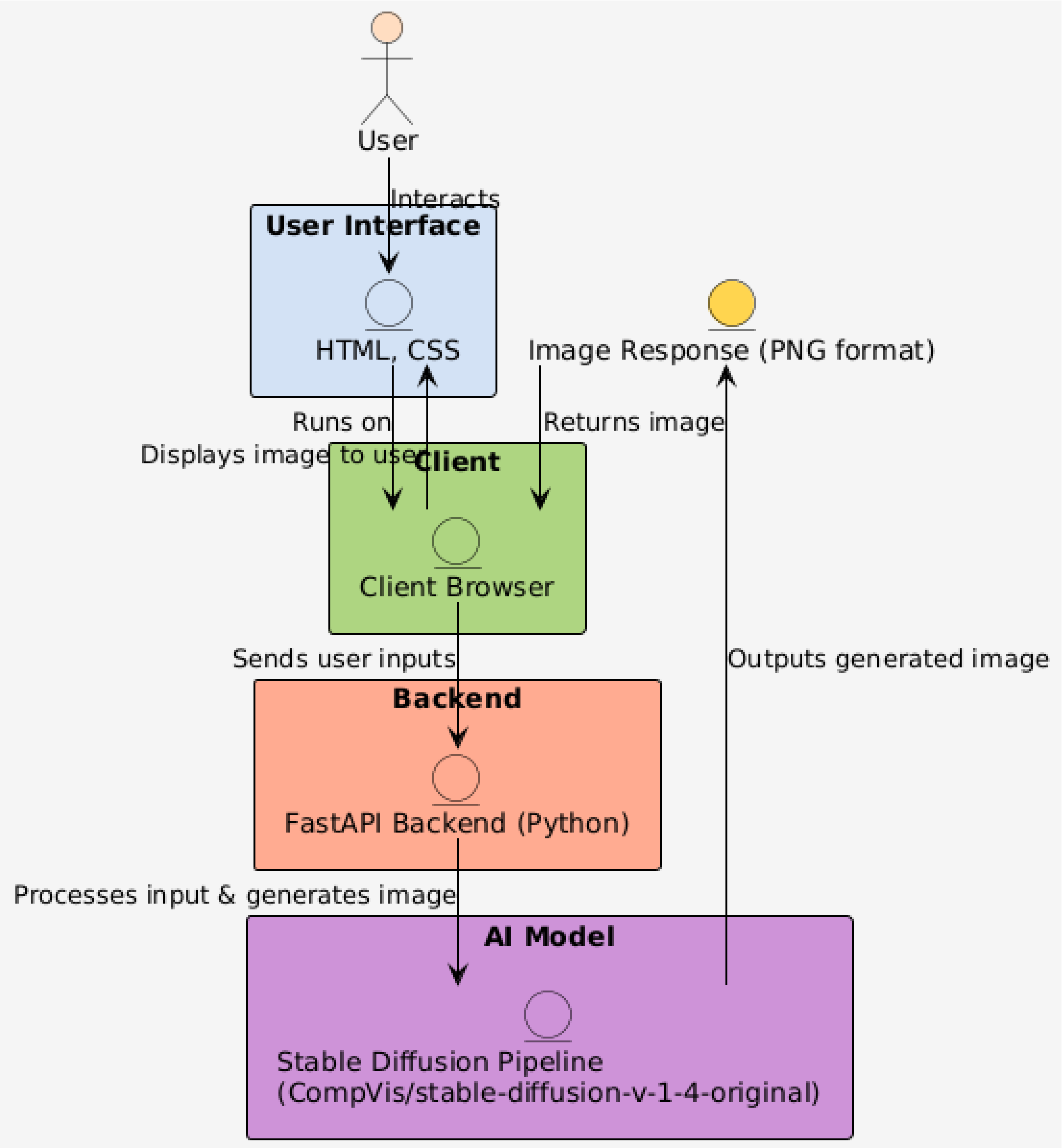
- **Functional Requirements:**
- User Input Collection and Customization: Allow users to input preferences for fabric, color, and style, and customize generated designs
- Design Generation and Preview: Use the Stable Diffusion Pipeline to generate realistic clothing designs and display them for user review.
- User Authentication and Data Storage: Secure login, store user preferences and generated designs, and enable search and filter options.

Phase-3: Project Design

Objective: Define system architecture, user flow, and UI/UX components.

Key Points:

1. **System Architecture Diagram:**
    - A visual representation of the system components and their interactions.
    - For this, a simple flowchart or sketch can illustrate the backend, frontend, and data flow.



2. **User Flow:**
   - ○ Describes how a user will interact with the project from start to finish

- 1. User opens the application in their web browser.
- 2. User is presented with a form to input preferences for fabric, color, and style.
- 3. User submits the form.
- 4. The client browser sends the user inputs to the FastAPI backend.
- 5. The FastAPI backend processes the inputs and uses the Stable Diffusion Pipeline to generate an image.
- 6. The generated image is sent back to the client browser.
- 7. User views the generated clothing design on the application.
- 8. User can save, share, or further customize the design if needed.

3. **UI/UX Considerations:**

    - Designing a user-friendly interface to ensure an intuitive experience.
    - A basic wireframe layout could include:
      Home Page:
        - **Header**: Logo, Navigation menu.
        - **Main Content**:
            - Form with input fields for fabric, color, and style preferences.
            - Submit button to generate the design.
        - **Footer**: Contact information, Social media links.
    - **Results Page:**
        - **Header**: Logo, Navigation menu.
        - **Main Content**:
            - Display the generated clothing design.
            - Options to save, share, or further customize the design.
        - **Footer**: Contact information, Social media links.

Phase-4: Project Planning (Agile Methodologies)

Objective

Break down the tasks using Agile methodologies.

Key Points

1. **Sprint Planning**:
    - **Sprint 1**: Initial Setup and Backend Development
        - Task: Environment Setup (Virtual Environment, Library Installation)
        - Task: Backend API Development (FastAPI Setup, Stable Diffusion Integration)
    - **Sprint 2**: Frontend Development and Integration
        - Task: User Interface Design (HTML, CSS)
        - Task: API Integration (Connecting Frontend with Backend)
    - **Sprint 3**: Testing and Bug Fixes
        - Task: Functional Testing (User Input, Image Generation, Display)
        - Task: Performance Testing (Response Time, Stability)
    - **Sprint 4**: Deployment and Final Validation
        - Task: Deployment Setup (Hosting, Domain Configuration)
        - Task: Final Validation (User Feedback, Improvements)

2. **Task Allocation**

3. **Timeline & Milestones**:

Detailed Timeline:

Sprint 1: Initial Setup and Backend Development (1 week)

- **Day 1-2**: Environment Setup (Virtual Environment, Library Installation)
- **Day 3-7**: Backend API Development (FastAPI Setup, Stable Diffusion Integration)

Sprint 2: Frontend Development and Integration (2 weeks)

- **Week 1**: User Interface Design (HTML, CSS)
- **Week 2**: API Integration (Connecting Frontend with Backend)

Sprint 3: Testing and Bug Fixes (1 week)

- **Day 1-2**: Functional Testing (User Input, Image Generation, Display)

- **Day 3-5**: Performance Testing (Response Time, Stability)
- **Day 6-7**: Bug Fixes and Improvements

Sprint 4: Deployment and Final Validation (1 week)

- **Day 1-3**: Deployment Setup (Hosting, Domain Configuration)
- **Day 4-7**: Final Validation (User Feedback, Improvements)

Phase-5: Project Development

Objective: Code the project and integrate components.

Key Points:

1. **Technology Stack Used:**
    - **Programming Language**: Python
    - **Backend Framework**: FastAPI
    - **AI Model**: Stable Diffusion Pipeline (`CompVis/stable-diffusion-v-1-4-original`)
    - **Frontend**: HTML, CSS
    - **Database**: API-based queries
    - **Dependencies/Libraries**: `torch`, `diffusers`, `pydantic`, `PIL`, `requests`
2. **Development Process:**
   Steps Followed for Coding:
    - **Setup Environment**:
        - Created a virtual environment and installed the required libraries.
        - Configured the Stable Diffusion Pipeline and verified its functionality.
    - **Backend Development**:
        - Set up the FastAPI application and created endpoints.
        - Implemented the `generate_image` endpoint to process user inputs and generate images using the Stable Diffusion Pipeline.
        - Integrated error handling to manage exceptions and provide meaningful feedback.
    - **Frontend Development**:
        - Designed the user interface using HTML and CSS.
        - Created forms for inputting user preferences (fabric, color, style).
        - Implemented functionality to display generated images.
    - **Integration**:
        - Connected the frontend with the backend API to enable seamless communication.
        - Ensured that user inputs from the frontend are correctly processed by the backend.
        - Verified that generated images are displayed in real-time.
    - **Testing & Validation**:
        - Conducted functional and performance testing to ensure system reliability.
        - Collected feedback from test users and made necessary improvements.
3. **Challenges & Fixes:**
    - **Model Compatibility**:
        - **Challenge**: The initial method for generating images (`openai.Image.create`) was deprecated in the latest OpenAI library version.
        - **Fix**: Updated the code to use the latest API methods and ensured compatibility with the Stable Diffusion Pipeline.
    - **Error Handling**:
        - **Challenge**: Handling exceptions during image generation and display.
        - **Fix**: Implemented robust error handling to catch and manage exceptions, providing meaningful feedback to the user.
    - **Image Display**:
        - **Challenge**: Ensuring that generated images are correctly displayed in the browser.
        - **Fix**: Used the `PIL` library to convert images to byte streams and return them as streaming responses.
    - **Performance Optimization**:
        - **Challenge**: Reducing the response time for image generation and display.
        - **Fix**: Fine-tuned the Stable Diffusion model and optimized the backend code for

better performance.

Functional & Performance Testing

Objective

To create an AI system that leverages the Stable Diffusion Pipeline to efficiently generate diverse and realistic clothing designs, aiding fashion designers and retailers in their creative processes.

Test Cases Executed

1. **User Input Handling**:
   - Test: Ensure the system correctly processes user inputs for fabric, color, and style.
   - Expected Outcome: Inputs are accurately received and processed.
2. **Image Generation**:
   - Test: Verify that the Stable Diffusion Pipeline generates realistic clothing designs based on user inputs.
   - Expected Outcome: Generated images align with the input descriptions.
3. **Image Display**:
   - Test: Ensure generated images are displayed correctly to the user.
   - Expected Outcome: Images are displayed in the correct format and resolution.
4. **User Authentication**:
   - Test: Verify that users can securely log in and access their data.
   - Expected Outcome: Only authenticated users can access the system.
5. **Data Storage**:
   - Test: Ensure user preferences and generated designs are stored correctly.
   - Expected Outcome: Data is stored and retrieved without errors.
6. **Performance Testing**:
   - Test: Evaluate the system's response time and stability under various loads.
   - Expected Outcome: The system remains stable and responsive under expected load conditions.

Bug Fixes & Improvements

- **Fixed Input Handling Bug**: Resolved an issue where certain user inputs were not being processed correctly.
- **Improved Image Quality**: Enhanced the quality of generated images by fine-tuning the Stable Diffusion model.
- **Optimized Response Time**: Reduced the response time for image generation and display.
- **Secured User Authentication**: Implemented additional security measures to protect user data.

Final Validation

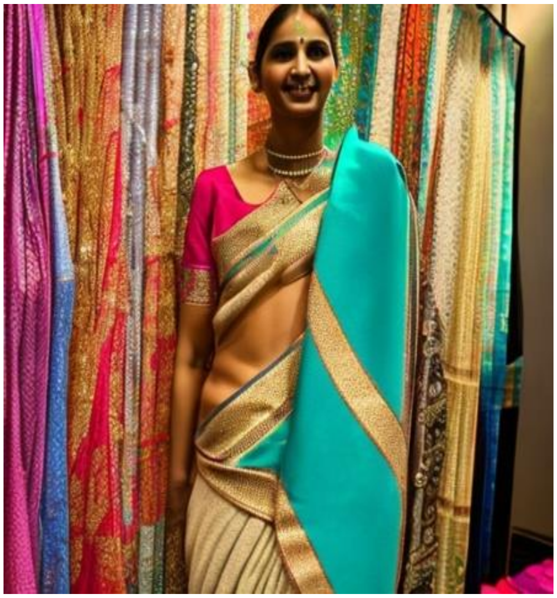
- **Meets Initial Requirements**: The system meets all the initial requirements specified in the project scope.
- **User Feedback**: Positive feedback from test users regarding the accuracy and quality of generated designs.
- **System Stability**: The system performs reliably under various conditions and user loads.

Final Submission

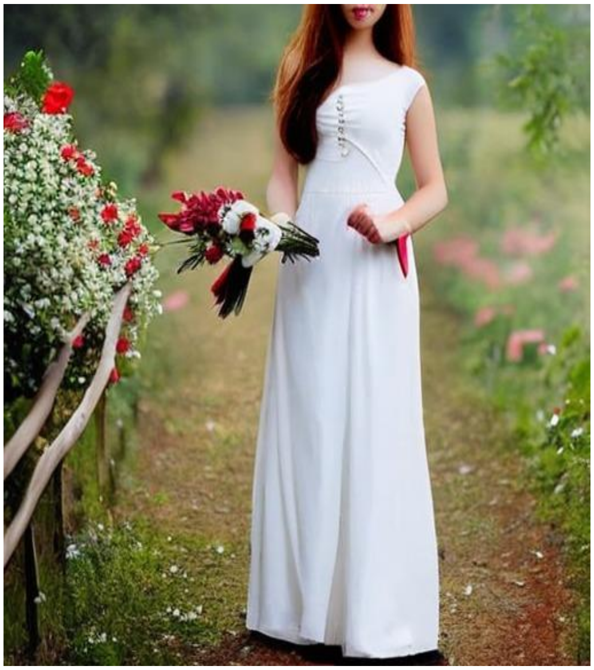
1. **Project Report Based on the templates**

Project Report:

Entered Prompt: “ A Indian women wearing a Traditional Saree”



Entered Prompt: “ A girl wearing bride ceremonial Dress”



2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link : https://github.com/b-paramesh/AI-stylers**
4. **Presentation**