**Hackathon Project Phases Template**

**Project Title:** CoutureAI: Clothing Image Generator Using AI

**Team Name:** Code Slingers

**Team Members:**

● Malraju Abhiram  
● Koyyala Nagendra Kishor  
● Mareedu Srinivas  
● Pabboju Uday Chary

**Phase-1: Requirement Analysis**

**Objective:**  
Develop an AI-powered fashion design tool using CoutureAI to help users visualize and create personalized clothing items based on detailed descriptions.

**Key Points:**

1. **Problem Statement:**
   * Many users struggle to find clothing that matches their specific style preferences and ideas in stores or online.
   * Traditional online shopping platforms lack the ability to provide realistic, customized visualizations of clothing based on user descriptions.
2. **Proposed Solution:**
   * An AI-powered application that uses the Stable Diffusion Pipeline to generate realistic images of clothing based on user inputs.
   * The app allows users to describe their desired clothing (e.g., "a red satin evening gown with lace sleeves") and instantly see a visual representation.
   * It also offers users the ability to refine designs and share them with tailors for custom creations.
3. **Target Users:**
   * Fashion-conscious individuals looking for personalized clothing designs.
   * Consumers who struggle to find specific clothing items in stores and want to visualize their custom ideas.
   * People interested in bespoke fashion, wanting to communicate their design ideas with tailors.
4. **Expected Outcome:**
   * A functional AI-powered fashion design app that provides users with accurate, realistic images of custom clothing based on their detailed descriptions.
   * Enhanced user experience in online fashion shopping and design, leading to more personalized, satisfying, and creative fashion purchases.

**Phase 2: Requirement Analysis**

1. **Technical Requirements:**

**1. Frontend:** HTML, CSS and JS

**2. Backend:** Node.js, Express

**3 . Database:** Local Data Storage

1. **Functional Requirements:**
   * **User Input:** A text box where users can describe their custom clothing ideas.
   * **Image Generation:** Backend (AI model) processes the description to generate a realistic image of the outfit.
   * **Real-Time Output:** Once the user submits the description, the generated image should be displayed in real-time.
   * **Sharing and Saving Options:** Allow users to save the image or share it via social media or email.
   * **User Feedback:** Users can provide feedback on the generated image to refine AI model outputs.
2. **Constraints & Challenges:**
   * **Ensuring Real-Time Updates:** API calls to Stable Diffusion or similar models must be optimized for speed and responsiveness.
   * **API Rate Limits:** Since image generation could be resource-heavy, handle API rate limits by implementing request queues or caching.
   * **Smooth UI Experience:** A simple, intuitive UI that allows users to input their descriptions and view results quickly. Ensure accessibility and fast page load times.

**Phase-3: Project Design**

**Objective:**  
Develop the architecture and user flow of the CoutureAI application for personalized clothing design.

**Key Points:**

1. **System Architecture:**
   * User inputs a clothing description or style idea through the UI.
   * The description is processed using the Stable Diffusion AI model integrated with Gemini Flash.
   * The AI model generates and processes realistic clothing images based on user input.
   * The frontend displays generated clothing visuals along with customization options, such as fabric choice, color variations, and design refinements.
2. **User Flow:**
   * **Step 1:** User inputs a detailed description (e.g., "a red satin evening gown with lace sleeves") via the interface.
   * **Step 2:** The backend sends the input to the Stable Diffusion model for image generation.
   * **Step 3:** The app processes the generated image and displays a realistic visualization of the clothing item to the user.
   * **Step 4:** The user can refine the design, adjust details, or save/share the image for future reference or tailoring.
   * **Step 5:** User can explore further options or create additional designs.
3. **UI/UX Considerations:**
   * **Minimalist, intuitive interface** with easy navigation to enhance user experience.
   * **Real-time preview** of clothing designs to allow users to see their ideas evolve.
   * **Customizable features**, such as color and fabric selection, with sliders and drop-down menus.
   * **Dark and light mode** for flexible viewing preferences.
   * **Mobile and desktop optimization** for seamless use across devices.

**Phase-4: Project Planning (Agile Methodologies)**

**Objective:**

| **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 | Koyyala Nagendra Kishor | Hugging face API Key, JS | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Mareedu Srinivas | API response format finalized | Basic UI with input fields |
| Sprint 2 | Clothing Design & Image Generation | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Malraju Abhiram | API response, UI elements ready | Design image generation functionality ready |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Pabboju Uday Chary | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Entire Team | 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 Stable Diffusion AI API.  
(🟡 Medium Priority) Build a Responsive UI with input fields.

**Sprint 2 – Core Features & Debugging (Day 1)**  
(🔴 High Priority) Implement clothing design & image generation functionality.  
(🔴 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 CoutureAI App.

**Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** HTML
   * **Backend:** Stable Diffusion AI Model
   * **Programming Language:** JS
2. **Development Process:**
   * Implement API key authentication and Stable Diffusion API integration.
   * Develop clothing design generation and customization logic.
   * Optimize queries to ensure quick and accurate image generation.
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:**

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | Query "A red satin evening gown with lace sleeves" | Generate image of the gown with lace sleeves. | ✅ Passed | Malraju Abhiram |
| TC-002 | Functional Testing | Query "A blue denim jacket with floral embroidery" | Generate image of the jacket with floral details. | ✅ Passed | Mareedu Srinivas |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Pabboju Uday Chary |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect image generation. | Correct image based on input description. | ✅ Fixed | Koyyala Nagendra Kishor |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ✅ Fixed | Mareedu Srinivas |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | App should be accessible online. | 🚀 Deployed | Entire Team |

**Final Submission:**

1. **Project Report** Based on the templates.
2. **Demo Video (3-5 Minutes)** showcasing app functionality.
3. **GitHub/Code Repository Link** for the source code.
4. **Presentation** to demonstrate the project in action.