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

# Hackathon Project Phases Template

**Project Title:**

**Team Name:**

Outfit oracle

**Team Members:**

N. tejaswi

M.Hima bindu

## Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-powered vehicle expert tool using Gemini Flash to help users compare and analyze vehicle specifications, reviews, and eco-friendly options.

**Key Points:**

1. **Problem Statement:**

**In the dynamic fashion industry, the demand for personalized and visually appealing clothing items is ever-increasing. Traditional online shopping platforms often lack the ability to offer customized visual representations of user-defined clothing styles, limiting the shopper's experience and satisfaction. CoutureAI aims to solve this problem by integrating Diffusers' Stable Diffusion Pipeline with a user-friendly Streamlit interface, allowing users to generate realistic images of clothing based on their descriptions. This project empowers consumers to visualize their unique clothing ideas, enhancing the shopping experience and bridging the gap between imagination and reality in fashion design.**

1. **Proposed Solution:**

**Meet Emma, a fashion-forward individual who loves expressing her unique style through clothing. Emma often imagines specific clothing designs but struggles to find them in stores or online. With CoutureAI, she can easily input detailed descriptions of her envisioned outfits, such as "a red satin evening gown with a sweetheart neckline and lace sleeves." CoutureAI then generates a realistic image of the dress based on her description. This helps Emma visualize her custom designs, refine her ideas, and even share them with a tailor for bespoke creations. CoutureAI transforms Emma's fashion dreams into vivid images, making her shopping and design process more personalized and satisfying.**

1. **Target Users:**

* 1. Fashion Designers: Quick visualizations of clothing designs before prototypes.
  2. E-commerce Platforms: Generate product images without expensive photoshoots.
  3. Marketing Teams: Create promotional content and social media visuals.
  4. Clothing Brands: Generate multiple clothing designs for ads and catalogs.
  5. Personal Stylists and Fashion Influencers: Showcase style ideas and lookbooks.
  6. Virtual Try-On Platforms: Allow users to digitally try on clothes before buying.

AI/Tech Developers: Build fashion tech tools like virtual fitting rooms.

1. **Expected Outcome:**

**The clothing image generator will offer quick, cost-effective clothing visuals for design, marketing, and e-commerce. It will enhance user experience and reduce reliance on physical photoshoots**

## Phase-2: Requirement Analysis

**Objective:**

define the needs and expectations of users to ensure the clothing image generator meets its intended goals.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language:

○ Backend: **Google Gemini Flash API**

○ Frontend: **Streamlit Web Framework**

○ Database: **Not required initially (API-based queries)**

1. **Functional Requirements:**

* **Image Generation: Ability to generate high-quality clothing images based on user input (design, style, colors, etc.).**
* **Customization Options: Users should be able to modify clothing details such as size, fabric, patterns, and colors.**
* **Predefined Templates: Provide a library of clothing templates (e.g., shirts, dresses) for quick image generation.**
* **Real-Time Rendering: Generate and display clothing images in real-time with minimal delays.**

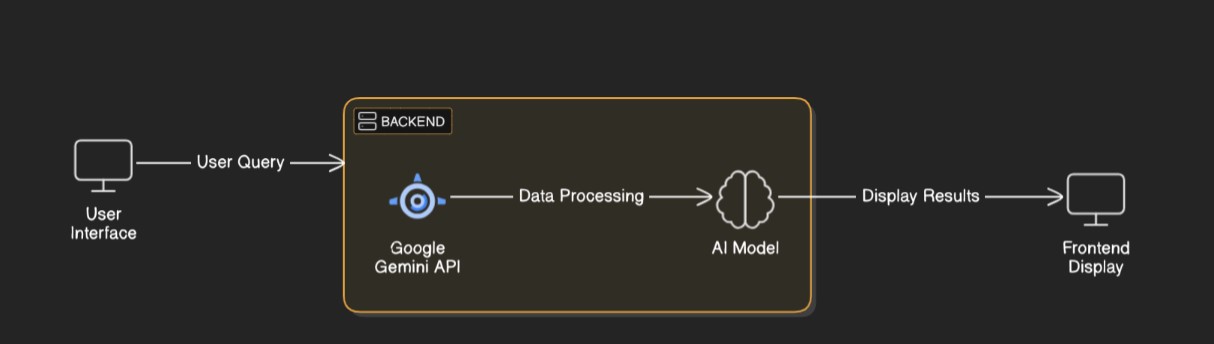
1. **Constraints & Challenges:**

* Image Quality and Realism: Achieving high-quality, realistic clothing visuals that meet user expectations for detail and texture.
* Processing Speed: Ensuring fast generation times, especially when handling complex designs or high-resolution images.
* Customization Complexity: Balancing a wide range of customization options without overwhelming users or creating a complicated

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* User Interaction: Users interact with a drag-and-drop interface or a form-based design tool for customizing clothing.
* Real-Time Preview: Users get instant previews of their designs.
* Profile Management: Users can create and manage their profiles, save designs, and make adjustments

1. **User Flow:**

* Landing Page → Registration/Login → Design Dashboard → Customization → AI
* Image Generation → Preview/Save/Download/Share
* Profile Management → View/Edit Saved Designs → Manage
* Preferences → Log Out

1. **UI/UX Considerations:**

**The UI/UX should focus on a clean, intuitive interface with easy navigation and real-time design previews. It should provide a seamless experience across devices, with responsive design and quick, interactive feedback.**

## 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 | Member 1 | Google API Key,  Python, Streamlit 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 | Vehicle Search &  Comparison | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1& 2 | 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&4 | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Member 2& 3 | 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)**

* Technologies: React.js, HTML, CSS, JavaScript
* Task: Set up the UI components (design dashboard, customization tools, preview section).
* Integration: Connect the frontend to backend APIs for user data, design input, and image rendering

**Sprint 2 – Core Features & Debugging (Day 2)**

* Core features include real-time clothing design customization, AI-generated image rendering, and seamless sharing to e-commerce and social media.
* Debugging involves testing for UI responsiveness, AI model accuracy, and smooth integration between frontend, backend, and third-party services.

## Phase-5: Project Development

**Objective:**

Implement core features of the AutoSage App.

**Key Points:**

1. **Technology Stack Used:**

Testing enhancements focus on improving test coverage, including unit, integration, and user acceptance tests to ensure all features work seamlessly. Submissions involve thorough quality assurance checks, bug fixes, and deployment to production after successful testing phases

1. **Development Process:**

**The development stack includes React.js for the frontend, Node.js and Express.js for the backend, and Python with TensorFlow or PyTorch for AI image generation. For storage, AWS S3 or Google Cloud Storage is used, along with MongoDB or PostgreSQL for database management.**

1. **Challenges & Fixes:**

**Challenges include achieving high-quality, realistic AI-generated images and ensuring fast processing times. Fixes involve optimizing AI models for efficiency and implementing caching or CDN services to reduce latency and improve performance.**

## 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 | Query "Best budget cars under ₹10 lakh" | Relevant budget cars should be displayed. | ✅ Passed | Tester 1 |
| TC-002 | Functional  Testing | Query "Motorcycle maintenance tips for  winter" | Seasonal tips should be provided. | ✅ Passed | Tester 2 |
| 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 | Develop er |
| 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  Streamlit Sharing | App should be accessible online. | 🚀 Deployed | DevOps |

## Final Submission

## The final submission includes a fully functional clothing image generator with AI-driven customization, real-time rendering, and integration with e-commerce and social media. It also contains complete documentation, user guides, and testing results for deployment.