Hackathon Project Phases Template for the AutoSage App project.

Hackathon Project Phases Template

# Project Title:

# Logo craft: innovative logo generation with diffusion technology

# Team Name:

(logo creation team)

# Team Members:

* G.Kavya sri
* A.Uma
* B.Vyshnavi
* B.Sri tejeshwini

# Phase-1: Brainstorming & Ideation

## Objective:

“ To leverage diffusion -bas\*ed generative AI models for creating unique , high quality and brand -aligned logos that incorporate creativity, adaptability and modern aesthetics.

## Key Points:

1. Problem Statement:

Traditional logo design methods are time-intensive and lack adaptability,while existing AI generated logos *often* fail to align with brand-specific requirements. There is a need for an innovative logo generation system that leverages diffusion technology to create high quality, brand- aligned and customizable logos efficiently.

1. Proposed Solution:
   * The solution leverages disffusion based generative AI models create high-quality customizable,and brand -aligned logos.
   * These models,known for there ability to generate intricate and high resolution imags, ensure that the logos are innovative unique,and adapetable to various branding needs.
2. Target Users:
   * Startups and small businesses
   * Graphic designers and creative professionals
   * Marketing and branding agencies
   * Entrepreneurs and solopreneurs
3. Expected Outcome:
   * Faster and more cost -effective logo design process.
   * Highly creative, diverse and brand consistent logo options.

# Phase-2: Requirement Analysis

## Requirement Analysis for Innovative Logo Generation with Diffusion Technology

## 1. Functional Requirements

## These define the core capabilities the system must have.

## 1.1. Logo Generation

## Generate high-quality logos using diffusion models.

## Support various styles (minimalist, abstract, 3D, vintage, etc.).

## Ability to include text and symbols in logos.

## Allow users to input preferences (colors, shapes, typography).

## Provide multiple variations for user selection.

## 1.2. User Input & Customization

## Upload existing logos for enhancement or rework.

## Allow manual adjustments to AI-generated designs.

## Support real-time preview and refinement.

## Enable fine-tuning for brand identity alignment.

## 1.3. Output & Exporting

## Export logos in various formats (PNG, JPG, SVG, EPS).

## Ensure vector-based output for scalability.

## Provide high-resolution outputs suitable for print and digital use.

## 1.4. AI Model Interaction

## Use pre-trained diffusion models with fine-tuning for logos.

## Allow users to influence the generation process through prompts.

## Optimize diffusion process for faster generation.

## ---

## 2. Non-Functional Requirements

## These focus on system quality, performance, and usability.

## 2.1. Performance & Scalability

## Ensure fast logo generation (<10 sec per logo).

## Optimize model to run on standard GPUs efficiently.

## Support batch processing for multiple logo generations.

## 2.2. Usability & User Experience

## Intuitive UI for users with

# Phase-3: Project Design

## Key Points:

Key Points for Project Design – Innovative Logo Generation with Diffusion Technology

1. System Architecture

Frontend: Web-based UI for user interaction (React, Vue.js)

Backend: Python-based API (FastAPI, Flask) to process requests

AI Model: Fine-tuned diffusion model for logo generation

Database: PostgreSQL or MongoDB for storing user preferences & designs

Deployment: Cloud-based (AWS, GCP) or on-premise GPU setup

---

2. AI Model Design

Base Model: Stable Diffusion / Custom-trained diffusion model

Fine-Tuning: Train on diverse logo datasets (vectorized logos, typography)

Input Conditioning: Accepts text prompts, color palettes, and brand guidelines

Output Optimization: Ensures high-resolution, vectorized (SVG, PNG) formats

---

3. User Interaction & Features

Customization Panel: Users select styles, fonts, colors, and elements

Live Preview: Real-time rendering of AI-generated logos

Editable Layers: Users can tweak AI-generated elements (shapes, text)

Batch Generation: Create multiple variations in one session

---

4. Performance Optimization

Efficient Model Execution: Use FP16 precision, optimized GPU inference

Caching Mechanism: Store frequently used logo templates for quick access

Parallel Processing: Multi-GPU support for faster generation

---

5. Output & Exporting

Formats Supported: PNG, JPG, SVG, EPS

Resolution Handling: Dynamic resizing for different use cases (web, print)

Brand Kit Integration: Generate complementary brand assets (favicon, social media banners)

---

6. Security & Compliance

User Data Privacy: Secure storage of generated logos & user preferences

Plagiarism Detection: AI-based originality check to prevent logo duplication

Watermarking: Temporary watermark on drafts before final purchase/export

---

7. Future Scalability & Enhancements

AI Refinement: Continual learning from user feedback

Integration with Design Tools: Plugins for Adobe Illustrator, Figma

Mobile & Web Support: Adaptive UI for multiple devices

Would you like a detailed workflow diagram or technical implementation breakdown?

# 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 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

## Project Development Plan – Innovative Logo Generation with Diffusion Technology

## ---

## 1. Project Planning & Research

## Define Objectives: AI-powered logo generation with customization features.

## Research Diffusion Models: Compare Stable Diffusion, DALL·E, and custom-trained models.

## Analyze Market Needs: Identify target users (designers, businesses, startups).

## ---

## 2. Technology Stack Selection

## Programming Language: Python (backend & AI model), JavaScript (frontend).

## AI Framework: PyTorch, Hugging Face Diffusers, OpenCV for image processing.

## Frontend: React.js or Vue.js for an interactive UI.

## Backend: FastAPI or Flask for handling API requests.

## Database: PostgreSQL or MongoDB for user data & preferences.

## Deployment: AWS, GCP, or local GPU-based server.

## ---

## 3. AI Model Development

## Dataset Collection: Curate logo datasets with diverse styles & vector formats.

## Model Training & Fine-Tuning: Adapt a pre-trained diffusion model for logo generation.

## Feature Engineering: Add text, symbols, and branding elements to the generation process.

## Optimization: Reduce inference time and enhance output resolution (FP16 precision, multi-GPU processing).

## ---

## 4. Application Development

## Frontend Development

## User Input Interface: Allows users to define style, colors, and typography.

## Real-time Preview: Dynamic rendering of generated logos.

## Editable Layers: Users can tweak elements of the AI-generated logos.

## Backend Development

## API for Logo Generation: Connects frontend with the AI model.

## User Authentication: Secure login & session management.

## Storage & Retrieval: Save user-generated logos for future access.

## ---

## 5. Testing & Optimization

## Model Evaluation: Ensure high-quality outputs with minimal artifacts.

## User Experience Testing: Check UI responsiveness and ease of use.

## Performance Testing: Optimize GPU usage and reduce generation time.

## ---

## 6. Deployment & Maintenance

## Cloud/On-Premise Deployment: Deploy the AI model & backend services.

## User Feedback Integration: Continuously improve based on user input.

## Feature Expansion: Add AI-guided customization and vector export enhancements.

## ---

# 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

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