

**Hackathon Project Phases Template** that ensures students can complete it efficiently while covering all six phases. The template is structured to capture essential information without being time-consuming.

---

# Hackathon Project Phases Template

## Project Title:

Auto Sage

## Team Name:

AutoAnalyzer

## Team Members:

- Kavya Supe
  - Krishnaveni Srirangam
  - Indu Namani
- 

## Phase-1: Brainstorming & Ideation

### Objective:

**Comprehensive Vehicle Information** – Provide detailed specifications, features, and performance insights for new two-wheeler and four-wheeler vehicles.

**Vehicle Comparisons** – Enable side-by-side comparisons of multiple vehicles based on specifications, price, and features for better decision-making.

**User-Friendly Interface** – Ensure a seamless and intuitive experience for users to easily navigate and find relevant vehicle information.

**Interactive Tools & Insights** – Offer interactive tools like EMI calculators, fuel efficiency estimators, and maintenance cost analysis to support smart purchasing decisions.

### Key Points:

## 1. Problem Statement:

- **Information Gap** – Lack of centralized vehicle details.
- **Comparison Challenge** – Difficulty in comparing multiple vehicles.
- **Outdated Data** – Delayed updates on pricing and trends.

## 2. Proposed Solution:

- AI-powered chatbot for real-time vehicle suggestions.
- Comprehensive comparison tables with specifications, features, and expert insights.
- A user-friendly interface that adapts to preferences for a tailored experience.

**Target Users:** Car & Bike Buyers, Automobile Enthusiasts, Dealerships & Showrooms, Auto Reviewers & Bloggers.

## Expected Outcome:

- Accurate, AI-generated vehicle recommendations.
- Easy-to-understand comparisons of multiple automobiles.
- Personalized results based on user preferences.
- Improved decision-making through data-driven insights.

---

## Phase-2: Requirement Analysis

### Objective:

- Define technical and functional requirements.

### Key Points:

#### 1. Technical Requirements:

- Integration with Google Generative AI (Gemini API) for intelligent responses.
- A React-based frontend for a seamless user experience.
- Secure API handling for fetching automobile data.
- Scalability for increasing users and expanding vehicle databases.

#### 2. Functional Requirements:

- User-friendly search bar for easy vehicle lookup.
- AI-generated insights based on real-time automobile trends.
- Vehicle comparison functionality with structured tables.
- Error handling for invalid queries.

#### 3. Constraints & Challenges:

- Real-time automobile data availability.

- Ensuring AI-generated results are up-to-date and reliable.
  - Handling API rate limits and performance optimization.
- 

### **Phase-3: Project Design**

#### **Objective:**

- Create the architecture and user flow.

#### **Key Points:**

##### **1. System Architecture Diagram:**

- Frontend: React.js for UI
- Backend: Google Generative AI API for recommendations
- Data Handling: Secure API calls & local state management

##### **2. User Flow:**

- User enters a search query (e.g., "Best electric cars under \$30,000").
- AI processes the query and fetches relevant vehicle insights.
- Suggestions are displayed with comparisons, reviews, and pricing.

##### **3. UI/UX Considerations:**

- Minimalist UI for easy navigation.
  - Responsive design for mobile and desktop users.
  - Dark mode/light mode for better readability.
- 

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

#### **Objective:**

- Break down the tasks using Agile methodologies.

#### **Key Points:**

##### **1.Sprint Planning:**

**Sprint Goal:** Develop a working prototype of AutoSage with search and recommendation features.

##### **2.Task Allocation:**

- Frontend Development: UI components, search bar, and results display.
- Backend Integration: API calls and AI model connection.
- Testing: Functional testing and bug fixes.

### 3. Timeline & Milestones:

- **Week 1: UI design & API integration**
- **Week 2: AI response handling & error handling**
- **Week 3: Testing & refinements**
- **Week 4: Finalization & deployment**

---

## Phase-5: Project Development

### Objective:

- Code the project and integrate components.

### Key Points:

#### 1. Technology Stack Used:

- **Frontend: React.js**
- **Backend: Google Generative AI API**
- **Styling: CSS/Tailwind**
- **Deployment: Vercel/Netlify**

#### 2. Development Process:

- **Create UI components → Connect AI API → Process user input → Display recommendations.**

#### 3. Challenges & Fixes:

- **Issue: API response formatting was inconsistent.**
- **Fix: Implemented text formatting for structured output.**

---

## Phase-6: Functional & Performance Testing

### Objective:

- Ensure the project works as expected.

### Key Points:

#### 1. Test Cases Executed:

- **Search with valid inputs (e.g., "Best SUVs 2024").**
- **Search with invalid inputs (e.g., "gibberish text").**
- **API failure handling and fallback suggestions.**

**2. Bug Fixes & Improvements:**

- Improved API request handling for faster results.
- Enhanced error messages for better user experience.

**3. Final Validation:**

- Does AutoSage provide accurate vehicle insights? ☒
- Is the UI intuitive and user-friendly? ☒
- Can users compare vehicles effectively?

**4. Deployment:**

- Hosting on Vercel/Netlify
  - GitHub repo for open-source collaboration
- 

**Final Submission**

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