NASSCOM GEN AI HACKATHON

**AutoSage**

**Team Name:**Infinity Drive

**Team Members:**

G. Yogeshwar

G. Vishnu

J. Venkatesh

P. Rakesh

**Phase-1: Brainstorming & Ideation**

**Objective:**

Develop an AI-powered vehicle comparison and recommendation tool using Google Gemini AI to help users analyze vehicle specifications, reviews, financing options, and resale value.

**Key Points:**

* **Problem Statement:**
* Users struggle to find reliable, up-to-date information on vehicles before purchasing.
* Lack of guidance on financing, resale value, maintenance, and eco-friendly vehicle options.
* **Proposed Solution:**
* An AI-powered application that fetches real-time vehicle details, comparisons, EMI calculations, and expert recommendations.
* Integration of a multilingual chatbot for diverse user accessibility.
* **Target Users:**
* Vehicle buyers looking for comparisons and financing options.
* Current owners needing resale value estimation and maintenance tips.
* Eco-conscious consumers searching for hybrid and electric vehicles.
* **Expected Outcome:**
* A functional AI-powered vehicle research platform with interactive tools like EMI calculator, resale estimator, and AI-powered suggestions.

**Phase-2: Requirement Analysis**

**Objective:**

Define the technical and functional requirements for the AutoSage App.

* **Technology Stack:**
* Frontend: React.js, React Bootstrap, React Router, Chart.js
* Backend: Node.js, Express.js
* Database: MongoDB (via Mongoose)
* APIs: Google Gemini AI (Chatbot), Google Translate API (Multilingual Support)
* **Core Functionalities:**
* Vehicle search by price, brand, fuel type, mileage, and body type.
* Side-by-side vehicle comparison with key specs and expert reviews.
* EMI & Resale value estimators to assist users in financial planning.
* AI-powered chatbot for real-time queries about vehicle performance, features, and recommendations.
* **Challenges & Solutions:**

|  |  |
| --- | --- |
| **Challenge** | **Solution** |
| **API Rate Limits** | Optimize API requests and use caching mechanisms |
| **Handling Large Data Sets** | Implement pagination and database indexing |
| **Real-time Updates** | Ensure seamless synchronization with API data |

**Phase-3: Project Design**

**Objective:**

Develop the system architecture and user flow to ensure an intuitive and seamless experience.

* **System Architecture:**

1. **User Input (React.js Frontend):**

Users enter search queries or filter vehicles based on their preferences.

1. **Backend Processing (Node.js & Express.js):**

Processes API requests and retrieves vehicle specifications, reviews, and comparisons.

1. **AI Chatbot Integration (Google Gemini AI):**

Provides vehicle suggestions and financing options based on user queries.

1. **Data Storage (MongoDB):**

Stores frequently accessed vehicle data, user preferences, and reviews.

* **User Flow:**

1. User searches for a vehicle or compares multiple vehicles.
2. Backend retrieves real-time data from APIs.
3. Results are displayed with vehicle specifications, EMI & resale value, and user ratings.
4. Users interact with the AI chatbot for personalized recommendations.

* **UI/UX Considerations:**

1. Simple navigation with well-structured search filters.
2. Responsive design optimized for mobile and desktop usage.
3. Interactive comparison tables with real-time data visualization.

**Phase-4: Project Planning (3-Day Agile Sprint)**

**Objective:**

Complete the entire development within 3 days using rapid sprints.

**Sprint Planning & Priorities:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Dependencies** |
| Day 1 | Environment Setup (React, Node, MongoDB) | 🔴 High | 3 Hours | Day 1 | Installation of dependencies |
| Day 1 | API Integration (Google Gemini AI) | 🔴 High | 2 Hours | Day 1 | API key setup |
| Day 1 | UI Development (Search & Comparison Page) | 🟡 Medium | 3 Hours | Day 1 | API data format finalized |
| Day 2 | EMI & Resale Value Calculator | 🔴 High | 2 Hours | Day 2 | Formula implementation |
| Day 2 | AI Chatbot | 🔴 High | 3 Hours | Day 2 | Google Gemini AI setup |
| Day 2 | Testing & Debugging | 🟡 Medium | 3 Hours | Day 2 | UI & Backend functionalities completed |
| Day 3 | Final Enhancements & Bug Fixing | 🔴 High | 2 Hours | Day 3 | Debugging logs |
| Day 3 | Deployment & Documentation | 🟢 Low | 3 Hours | Day 3 | App fully tested |

**Phase-5: Project Development**

**Objective:**

Implement the core features of the AutoSage App using the defined tech stack.

* **Technology Stack Used:**
* Frontend: React.js, React Bootstrap, React Router, Chart.js
* Backend: Node.js, Express.js
* Database: MongoDB with Mongoose
* APIs: Google Gemini AI (Chatbot), Google Translate API
* **Development Process:**
* Set up the development environment.
* Build UI components for search, filters, and vehicle comparison.
* Implement backend API integrations.
* Add AI chatbot functionality.
* Conduct rigorous testing and optimizations.
* Deploy and document the project.

**Phase-6: Functional & Performance Testing**

**Objective:**

Ensure high performance and usability of the AutoSage App.

**Testing Strategy:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** |
| TC-001 | Functional Testing | Search "Best SUVs under ₹15 lakh" | Relevant SUVs should be displayed | ✅ Passed |
| TC-002 | Functional Testing | Compare two vehicles | Comparison table should be generated | ✅ Passed |
| TC-003 | Performance Testing | API response under 500ms | Should return data quickly | ⚠ Needs Optimization |
| TC-004 | UI/UX Testing | Ensure responsive design | UI should work on mobile & desktop | ✅ Passed |
| TC-005 | Bug Fixing | Fix chatbot response issues | AI should return correct answers | ✅ Fixed |