

**Student Name 1**: Abhishek Dutta

**Student Name 2**: Priyangshu Das

**Program**: Bachelor of Computer Application

**Semester**: 5th Semester

**Enrollment ID 1**: ADTU/2022-25/BCASP/040

**Enrollment ID 2**: ADTU/2022-25/BCASP/031

**Subject**: Techno Professional Skills

**Submitted to:** Dr. Prashanta Pratim Bairagi**Project Report**

**Title: Web Application for Insurance Deals**

**1. Abstract**

This project report details the design, development, and implementation of a web application for finding the best car insurance deals. The application allows users to input their car details and insurance preferences, displaying tailored results fetched from mock datasets or external sources. Built using modern technologies like React, Flask, and Tailwind CSS, the project emphasizes responsive design, performance optimization, and scalability.

**2. Acknowledgements**

We would like to thank [Your Mentor/Instructor/Organization] for their guidance and support throughout this project. Their insights have been instrumental in achieving the project's objectives.

**3. Introduction**

* **Purpose:**  
  The project aims to provide an efficient and user-friendly platform for comparing car insurance deals tailored to user inputs.
* **Scope:**  
  The web application enables users to search and sort insurance options based on various criteria, including premium, coverage type, and zero-depreciation options.
* **Objectives:**
  1. Develop a responsive and interactive frontend for seamless user experience.
  2. Implement a robust backend to process user inputs and manage data.
  3. Optimize the system for scalability and performance.

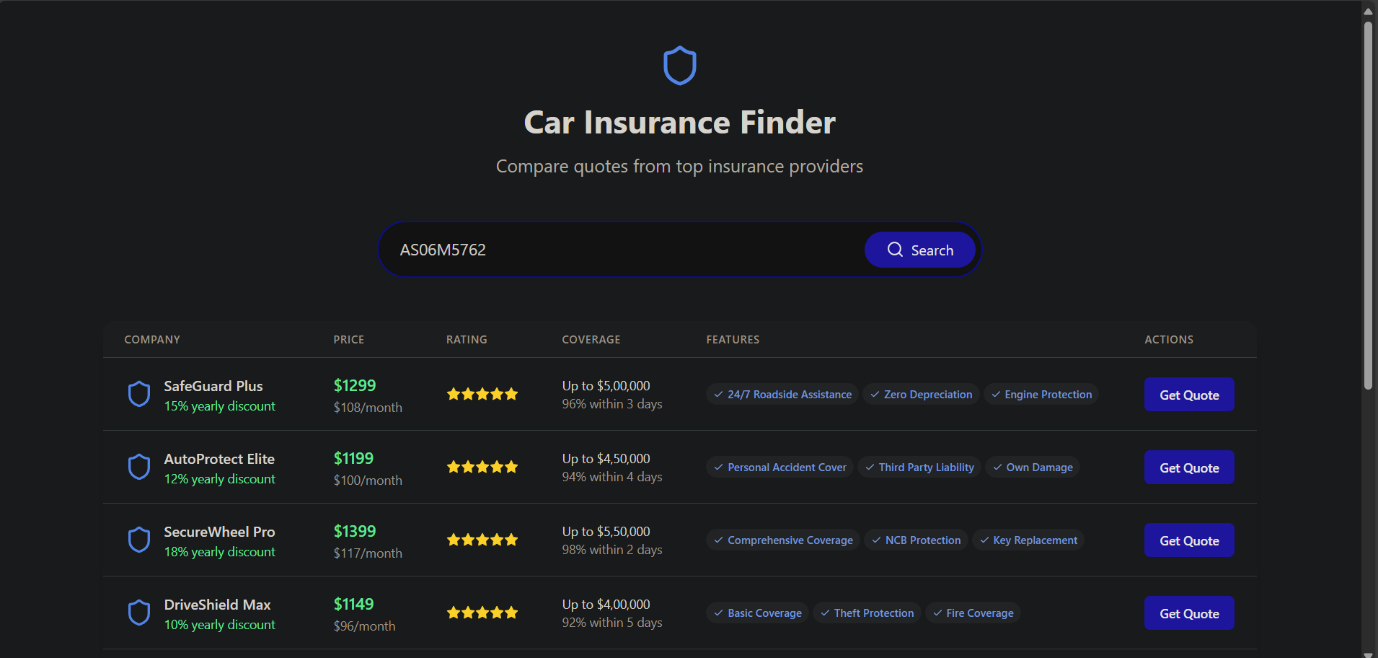
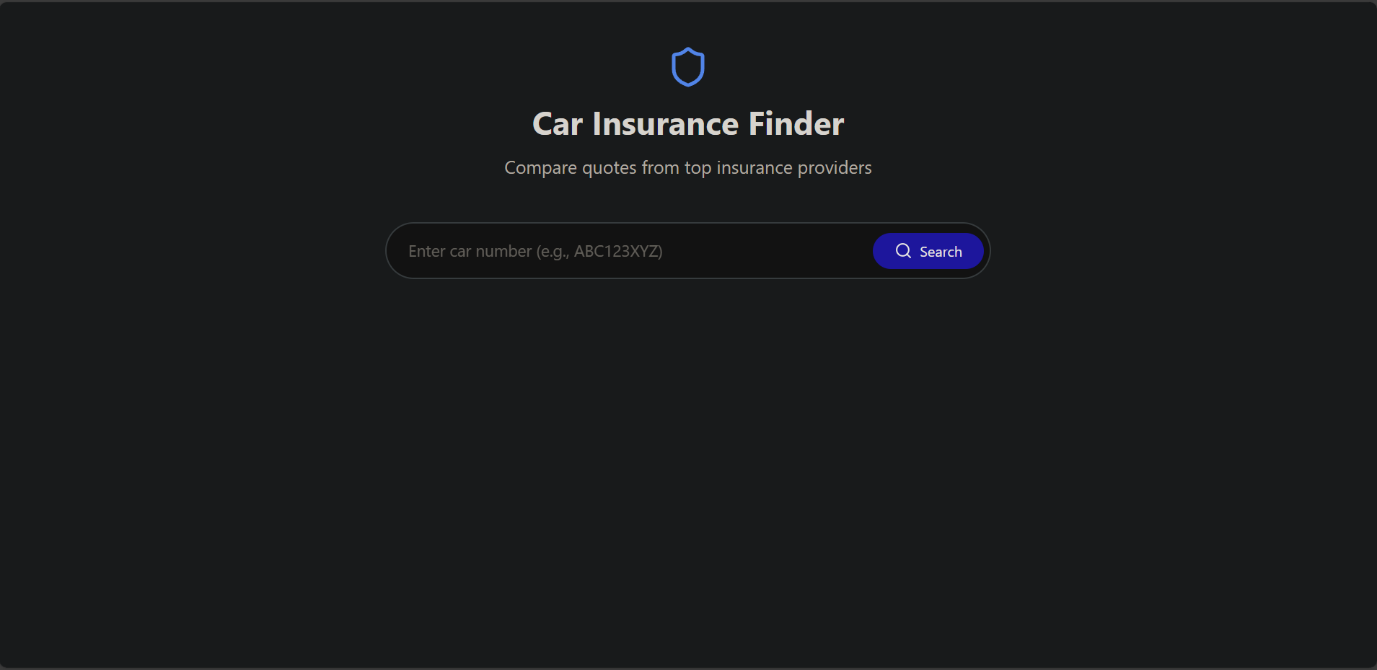
**4. System Analysis**

* **Existing System:**  
  Manual comparison of insurance options through different websites is time-consuming and inefficient.
* **Proposed System:**  
  The proposed system automates the process, providing a centralized platform for comparing insurance deals based on user preferences.
* **System Requirements:**
  + **Frontend:** React (v18.2.0), Tailwind CSS (v3.4.1).
  + **Backend:** Flask.
  + **Development Tools:** Vite for builds, Node.js for the development environment.
  + **Database:** Mock JSON data (optional integration with Firebase/SQLite).

**5. System Design**

* **Architecture:**  
  The application follows a modular architecture with separation of concerns between frontend, backend, and data layers.

**Screenshots:**



**6. Minimum System Requirements**

To use the application effectively, users must meet the following minimum system requirements:

**Hardware Requirements:**

* **Operating System:** Windows 10 or higher, macOS 10.14 or higher, or modern Linux distributions.
* **Processor:** Intel Core i3 (Quad Core recommended) or equivalent.
* **RAM:** 4 GB (8 GB recommended for optimal performance).
* **Storage:** 256 GB of available space (SSD recommended).

**Software Requirements:**

* **Web Browser:** Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari.
* **Node.js (v16 or higher):** For running the frontend during development.
* **Python 3.8 or higher:** Required for running the Flask backend.
* **Internet Connectivity:** Stable connection for data fetching and live updates.

**Development Environment (Optional for Developers):**

* Vite and npm for building and testing the frontend.
* Text Editor or IDE: Visual Studio Code, PyCharm, or similar tools.

**7. Implementation**

* **Frontend:**  
  React components for modular UI, styled with Tailwind CSS.
  + Key Components: Header, SearchForm, Table.
  + Responsive design using Tailwind’s grid system.
* **Backend:**  
  Flask APIs for managing user inputs and retrieving data.
* **State Management:**  
  React’s useState for handling local component states.

**8. Features**

1. **User Input:**
   * Fields for car number and insurance preferences.
   * Dropdown for selecting zero-depreciation coverage.
2. **Data Display:**
   * Results displayed in a responsive, sortable table.
   * Columns for provider name, premium, coverage type, and zero-depreciation notes.
3. **Interactivity:**
   * Gradient backgrounds, animated buttons, and hover effects.
   * Loading states for enhanced user experience.

**9. Testing and Results**

* **Testing:**
  + Unit tests for React components using Jest.
  + API testing with Postman for Flask endpoints.
* **Results:**
  + Successful rendering of responsive UI across devices.
  + Data fetched and displayed accurately based on user inputs.

**10. Conclusion and Future Enhancements**

* **Conclusion:**  
  The project successfully provides an interactive platform for comparing car insurance deals. Its modular architecture and modern tech stack ensure scalability and maintainability.
* **Future Enhancements:**
  1. Integrate live APIs from insurance providers.
  2. Implement user authentication for personalized experiences.
  3. Enhance the database with real-time storage and updates.

**11. References**

* React Documentation: [https://reactjs.org](https://reactjs.org/)
* Flask Documentation: [https://flask.palletsprojects.com](https://flask.palletsprojects.com/)
* Tailwind CSS: [https://tailwindcss.com](https://tailwindcss.com/)

**12. Appendices**

**A. Glossary:**

* **Third-Party Insurance:** Liability-only coverage.
* **Comprehensive Insurance:** Includes third-party liabilities and vehicle damage.
* **Zero-Depreciation Insurance:** Covers the full replacement value without depreciation.

**B. List of Tables:**

* Insurance results table with columns for provider name, premium, coverage type, and notes.