# AutoValuator: Car Price Prediction Model

## A Minor Project Synopsis Submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya

## Towards Partial Fulfillment for the Award of Bachelor of Technology in Computer Science and Engineering

**Submitted By**
**Priyanshi Goyal (0827CS221204)**
**Rahul Sharma (0827CS221217)**
**Rachit Shivhare(0827CS221213)**
**Pranay Jain (0827CS221195)**

**Guided by:**
**Prof. Krupi Saraf**

**Department of Computer Science and Engineering**
**Acropolis Institute of Technology & Research, Indore**
**Jan-June 2025**

## 1. Introduction

AutoValuator is a smart and easy-to-use website that helps users predict the price of a car based on its details like brand, model, year, fuel type, kilometres driven, and more. Many people face confusion when buying or selling a car because they don’t know its correct market value. Our project solves this problem using machine learning. By entering basic information about the car, users can get an estimated price within seconds. This saves time, removes guesswork, and helps users make better decisions. AutoValuator aims to make car buying and selling easier, faster, and more trustworthy for everyone.

## 2. Objective

- ❖ Build a smart platform that can predict the price of a car based on user input.
- ❖ Help users know the estimated market value of used cars easily.
- ❖ Use machine learning to make accurate and quick predictions.
- ❖ Allow users to input car details like brand, model, year, fuel type, and kilometers driven.
- ❖ Make car buying and selling decisions easier and more informed.

## 3. Scope

This project focuses on building a web-based platform where users can easily check the estimated price of a car by entering basic details. It is designed for anyone who wants to buy or sell a used car and needs to know its fair market value. In its initial version, AutoValuator will focus on predicting prices using important features like brand, model, year, fuel type, and kilometres driven. It will not include features like car loan estimation or insurance suggestions for now. The platform requires an internet connection and works best on modern web browsers. In the future, more features like comparison between multiple cars, saving search history, and user accounts can be added to improve the overall experience.

## 4. Study Of Existing System

1. **CarWale**

• **Problems Addressed:** Provides car prices, specifications, and user reviews.
• **Advantages:** Good UI, covers a wide range of car models.
• **Disadvantages:** Prices shown are general estimates, not customized based on specific car condition or usage.
• **Gaps Identified:** Does not provide price prediction based on user-input features like km driven, year, fuel type, etc.
• **Reference Link:** https://www.carwale.com/

2. **CarDekho**

• **Problems Addressed:** Helps users buy/sell cars with price details.
• **Advantages:** Large car database, EMI and loan options available.
• **Disadvantages:** No dynamic price prediction for used cars; depends on listings.
• **Gaps Identified:** Lacks machine learning–based price estimation for used cars.
• **Reference Link:** https://www.cardekho.com/

3. **OLX Autos**

• **Problems Addressed:** Platform to buy and sell used cars.
• **Advantages:** Easy to post and view listings.
• **Disadvantages:** No exact price prediction, only listing-based info.
• **Gaps Identified:** No tool to estimate a car's price based on actual condition and details.
• **Reference Link:** https://www.olx.in/cars/

4. **Cars24**

• **Problems Addressed:** Online car selling and buying service.
• **Advantages:** Simple selling process, inspection services available.
• **Disadvantages:** Estimated price often depends on internal inspection, not available directly to users.
• **Gaps Identified:** No feature to get a quick price prediction without going through full process.
• **Reference Link:** https://www.cars24.com/

## 5. Project Description

### Flowchart:

The flow of information in the project can be illustrated as follows:

1. **User Input**: The user enters car details like brand, model, manufacturing year, fuel type, transmission type, kilometres driven, etc.
2. **Data Preprocessing**: The input data is cleaned and converted into a format that the machine learning model can understand.
3. **Prediction Model**: A trained machine learning model (such as Linear Regression or Random Forest) processes the data.
4. **Price Prediction**: The model calculates and returns an estimated price of the car.
5. **Result Display**: The predicted car price is shown on the screen in a simple and understandable way.
6. **Optional Feedback**: Users may give feedback to improve prediction accuracy in future versions.

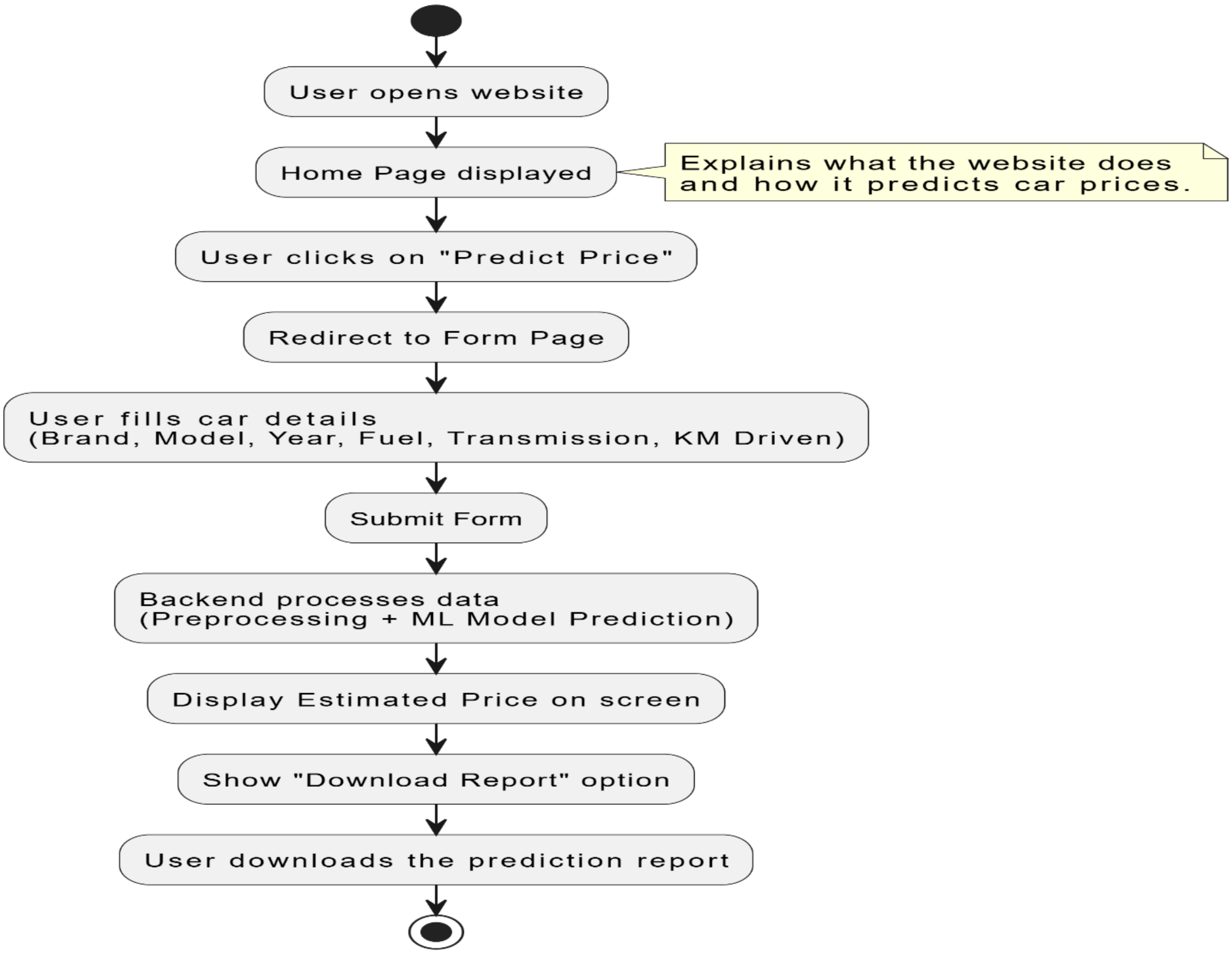

User opens website
Home Page displayed
Explains what the website does and how it predicts car prices.
User clicks on "Predict Price"
Redirect to Form Page
User fills car details (Brand, Model, Year, Fuel, Transmission, KM Driven)
Submit Form
Backend processes data (Preprocessing + ML Model Prediction)
Display Estimated Price on screen
Show "Download Report" option
User downloads the prediction report

**Figure: Flow Chart of System**

**ER Diagram:**
The ER (Entity-Relationship) diagram defines the core structure and interactions between various entities in the **AutoValuator – Car Price Prediction System**. It highlights how data flows between users, cars, evaluation tools, and the generated price reports. Below is a brief description of each entity and its role in the system:

**1. User:**
This entity represents the registered individuals who use the AutoValuator platform to predict their car's resale value.
**Attributes:**

- userID (unique identifier)
- name
- password

Each user owns one or more cars and can request price predictions through the evaluation tool. Users can also view the reports generated for their vehicles.

**2. Car:**
This entity contains the detailed specifications of the car owned by a user. These specifications are used by the evaluation tool to estimate the car's value.
**Attributes:**

- carID (unique identifier)
- brand
- model
- year
- fuelType
- transmissionMedium

Each car is linked to a user and is evaluated using the prediction tool.

**3. Admin:**
The Admin manages the platform and is responsible for approving users and overseeing the evaluation tools.
**Attributes:**

- username
- password

The admin approves new users and manages the tools used for evaluating the car data.

**4. Evaluation Tool:**
This entity represents the model or logic used to evaluate car details and predict their resale value.
**Attributes:**

- toolID (unique identifier)
- toolName
- version

It is managed by the Admin and is responsible for generating a price report based on the car's attributes.

**5. Price Report:**

This entity stores the final output of the evaluation – the estimated price of the car.

**Attributes:**

- reportID (unique identifier)
- estimatedPrice

Each report is linked to one car and is generated by the evaluation tool. Users can view this report to understand the value of their car.

**Relationships:**

- **Admin** approves **Users** and manages the **Evaluation Tool**.
- **Users** own one or more **Cars**.
- **Cars** are evaluated using the **Evaluation Tool**.
- The **Evaluation Tool** generates a **Price Report**.
- **Users** can view the generated **Price Report**.

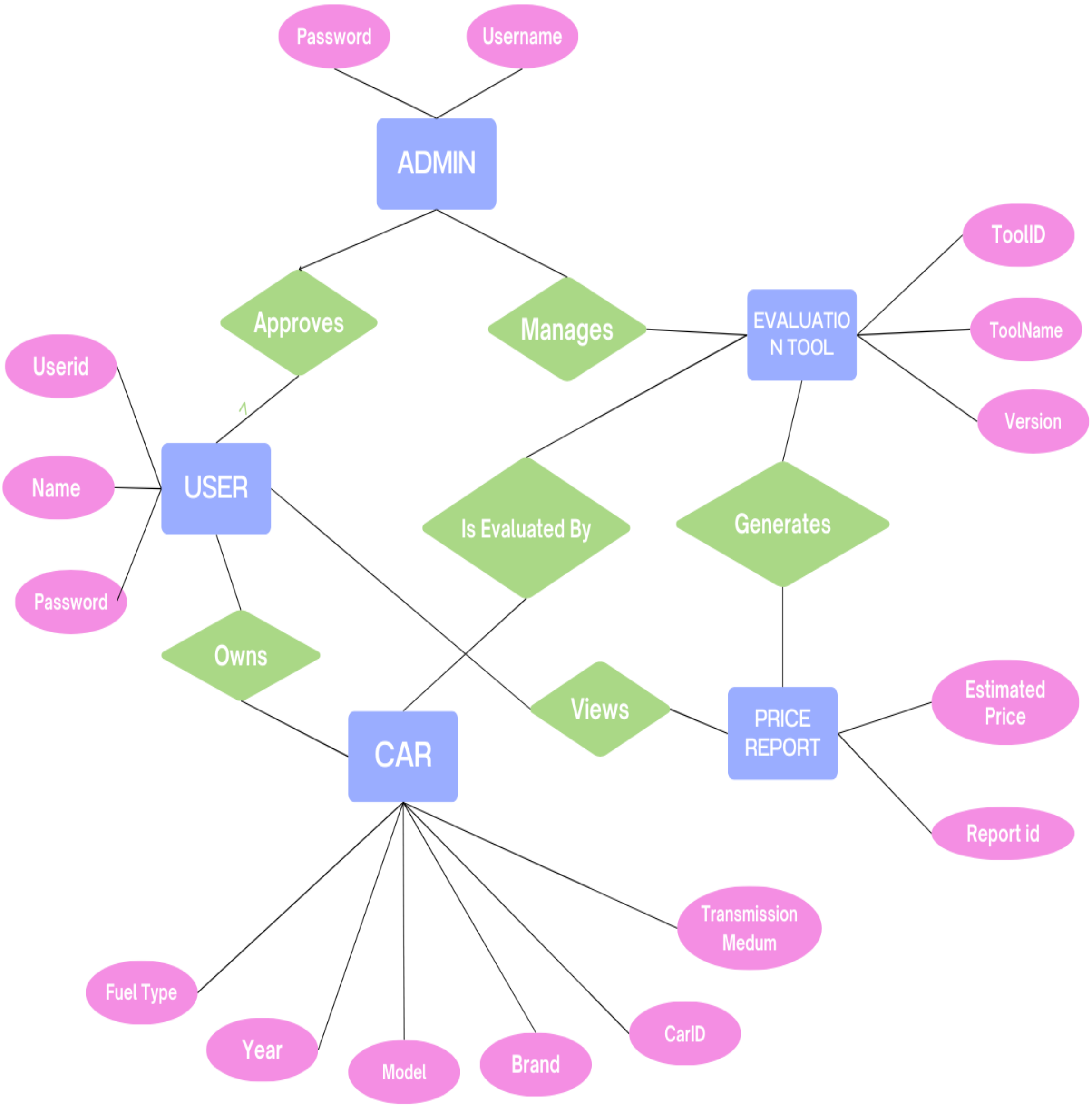


**Figure: ER Diagram**

## 6. Methodology

The methodology adopted in this project for predicting the price of used cars involves the following key steps:

**1. Data Collection**

The dataset used in this project was imported from Kaggle using the KaggleHub API. It contains various features of used cars like brand, model, manufacturing year, kilometers driven, fuel type, transmission, engine capacity, mileage, number of seats, and the target variable: selling price.

**2. Data Preprocessing**

Several preprocessing techniques were applied to clean and prepare the data for training:

- Unnecessary columns such as max_power, car_name, seller_type, and Unnamed: 0 were dropped.
- Column names were renamed for better readability.
- The Engine-Capacity values were converted from cc to liters.
- Missing or incorrect values (e.g., cars with 0 seats) were fixed by replacing them with appropriate values.
- Duplicate entries were removed to avoid data redundancy.
- One-hot encoding was applied to categorical features like brand, fuel type, and transmission to convert them into numeric format.

**3. Feature Selection**

To determine which features were most important in predicting the car price, an ExtraTreesRegressor model was used. This model provided feature importance scores, and a bar chart was plotted to visualize the top influencing features.

**4. Data Splitting**

The dataset was split into training (80%) and testing (20%) sets using the train_test_split function from scikit-learn. This ensures that the model is evaluated on unseen data to check its real performance.

**5. Model Building**

Multiple machine learning regression algorithms were considered, including:

- Linear Regression
- Support Vector Regression (SVR)
- Decision Tree Regressor
- Extra Trees Regressor
- Random Forest Regressor (Final choice)

After testing different models, the Random Forest Regressor was selected due to its better performance in terms of accuracy and lower error values.

6. **Model Evaluation**

The model was evaluated using the following metrics:

- Mean Squared Error (MSE)
- R-squared ($R^2$) Score

These metrics were used to assess how well the model predicted car prices on the test data. A higher $R^2$ score indicates better performance.

7. **Visualization**

Important features influencing the price were visualized using bar graphs. This helped in understanding which attributes (like mileage, engine capacity, etc.) have more impact on the final price.

| PHASES | TIMELINE | TASK COMPLETED |
|---|---|---|
| Requirements Gathering | Week 1 | Identified dataset, defined evaluation objectives, and functional specifications |
| Data Collection | Week 2 | Collected structured form data or sample entries for training |
| Data Preprocessing | Week 3 | Cleaned, normalized data, handled missing values, and feature selection |
| Model Selection & Design | Week 4 | Chose appropriate ML algorithm (e.g., Linear Regression, Random Forest, etc.) |
| Model Training | Week 5 | Trained model using historical data and tuned hyperparameters |
| Model Evaluation | Week 6 | Evaluated accuracy using metrics like MAE, RMSE, and cross-validation |

| PHASES | TIMELINE | TASK COMPLETED |
|---|---|---|
| Integration & Development | Week 7-8 | Integrated model with frontend/backend, created user form and dashboard interface |
| Testing | Week 9 | Performed end-to-end system testing and ML model prediction testing |
| Deployment | Week 10 | Deployed system on cloud/local server |
| Maintenance | Week 11 | Collected feedback, improved performance, and updated model periodically. |

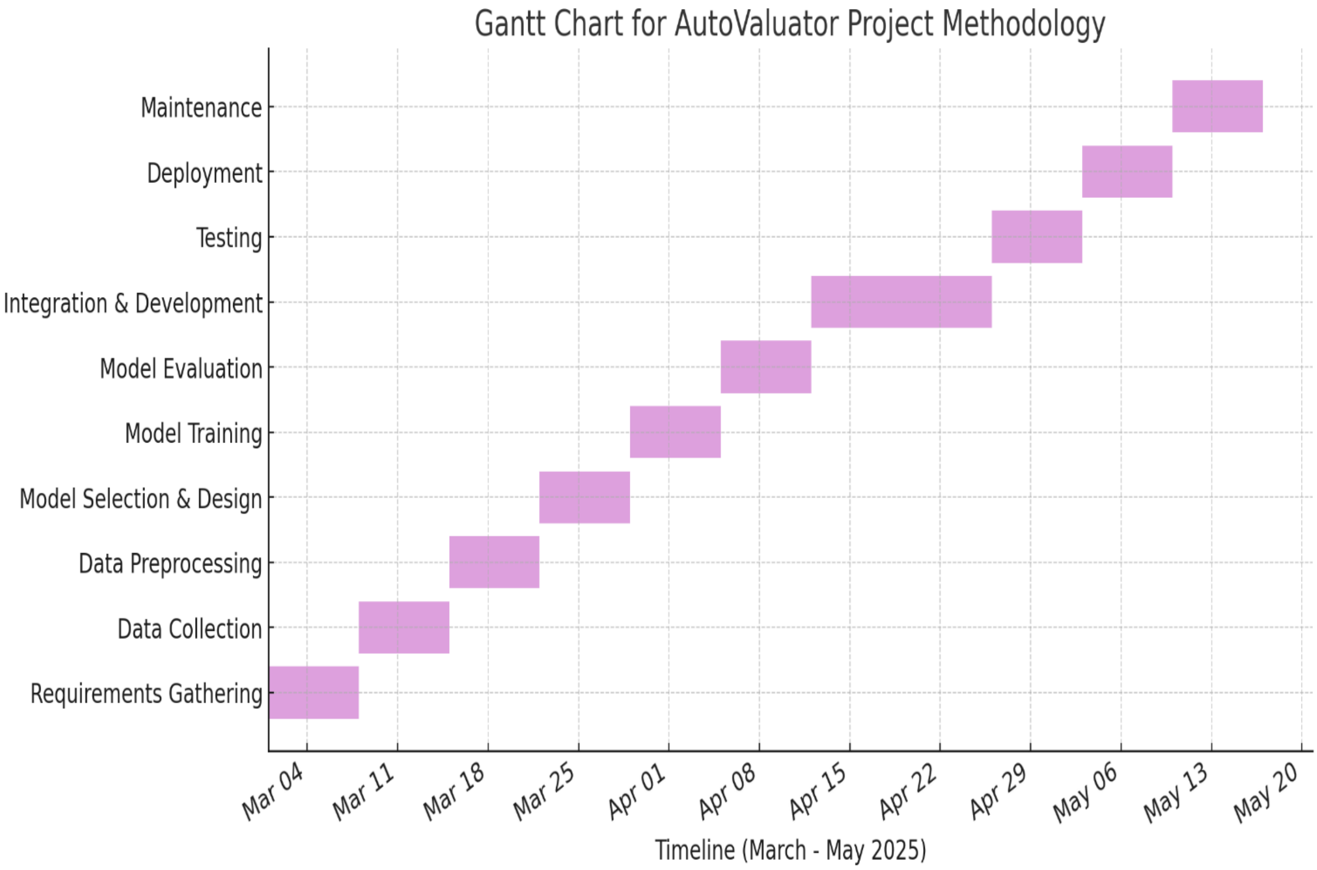


**Figure: Gantt Chart**

## 7. Features

Here are the main features that make your project stand out:

1. **Form-Based Car Data Entry:** Users can enter car details such as brand, model, year, fuel type, transmission, and kilometers driven through a simple and intuitive form.
2. **ML-Based Price Prediction:** The system uses trained Machine Learning models like Linear Regression or Random Forest to predict the estimated resale value of the car accurately.
3. **Instant Report Generation:** After submitting the car details, users receive a detailed evaluation report including estimated price, model info, and confidence range.
4. **Data-Driven Decision Support:** The platform helps users make informed decisions by evaluating current market trends, historical data, and similar car listings.

5. **Simple & User-Friendly Interface:** Designed with a clean and intuitive layout, users can smoothly navigate through the evaluation process without any technical hassle.

## 8. System Architecture

The system architecture of this project is made up of two main parts that work together to provide accurate car price predictions:

1. **Frontend (User Interface):**

- **What it does:**
  This is what users see and interact with. It allows them to enter car details such as brand, model, year, fuel type, transmission, and kilometres driven through a clean and simple form.

- **How it works:**
  Built using **React.js, CSS and JavaScript** the frontend collects user input and sends it to the backend for processing. It also displays the predicted selling price received from the backend.

2. **Backend (Processing Engine):**

   - **What it does:**
     This part takes the user input, runs it through a trained **Machine Learning model**, and returns the estimated price of the car.

   - **How it works:**
     Developed using **Python**, the backend receives car details, loads the trained ML model (like Decision Regression or Random Forest), performs the prediction, and sends the result back to the frontend.

## How They Work Together:

1. **User Input:**
   The user submits car details via the frontend form.

2. **Backend Prediction:**
   The backend processes this input using the ML model and calculates the car’s estimated selling price.

3. **Response:**
   The frontend displays the predicted price instantly to the user in a report format.

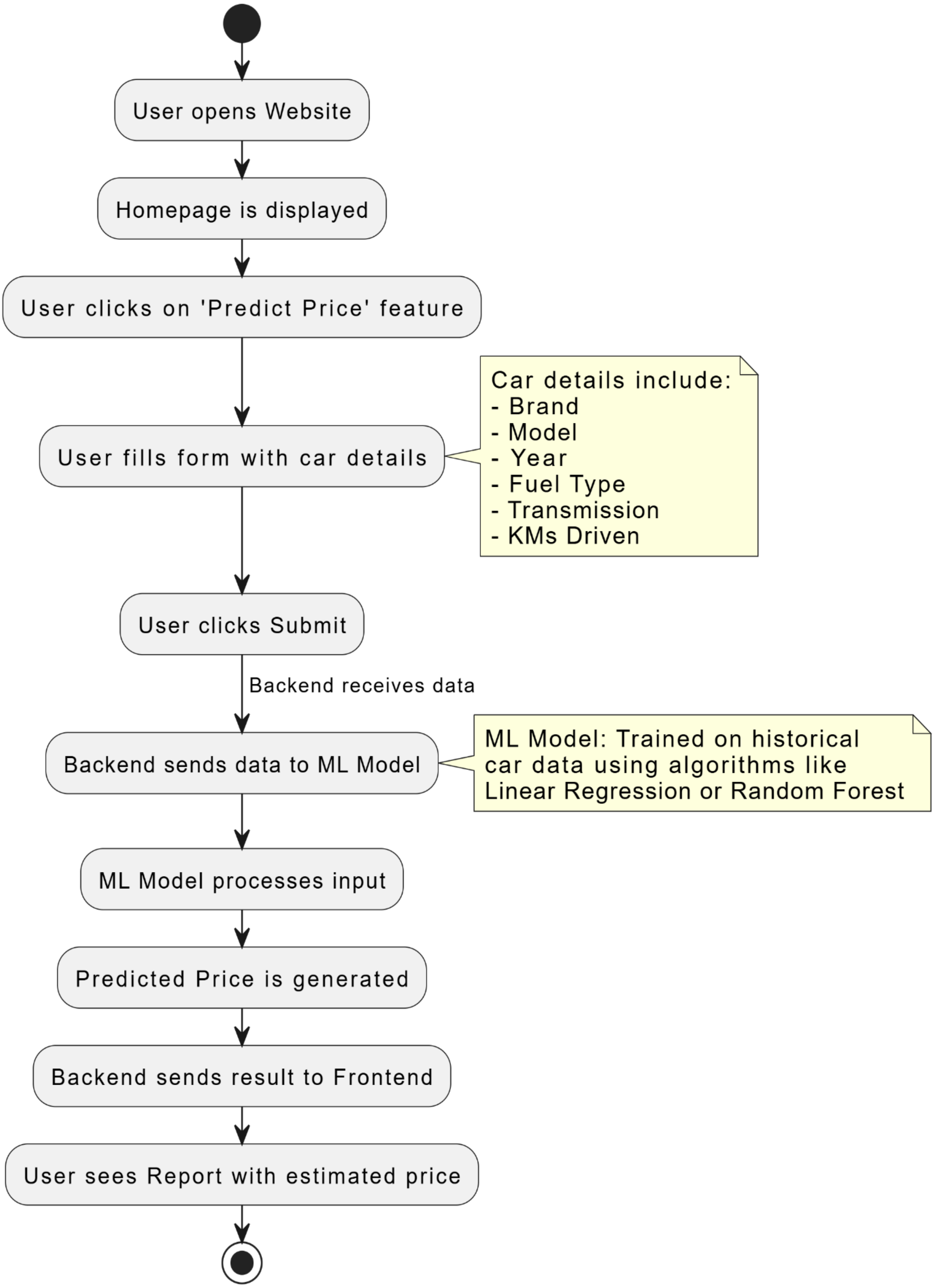

User opens Website
Homepage is displayed
User clicks on 'Predict Price' feature
User fills form with car details
Car details include:
- Brand
- Model
- Year
- Fuel Type
- Transmission
- KMs Driven
User clicks Submit
Backend receives data
Backend sends data to ML Model
ML Model: Trained on historical car data using algorithms like Linear Regression or Random Forest
ML Model processes input
Predicted Price is generated
Backend sends result to Frontend
User sees Report with estimated price

**Figure: System Architecture**

## 9. User Interface (UI)

The User Interface (UI) of the AutoValuator platform is designed to be intuitive and beginner-friendly, ensuring users can easily predict their car’s price without any confusion or complexity.

1. **Homepage**:
   - The homepage welcomes users with a clean, minimal layout.
   - A prominent **"Estimate Now"** button is placed at the center to grab attention and guide users directly to the main feature.
   - A simple top navigation bar allows users to navigate easily across the platform if needed.

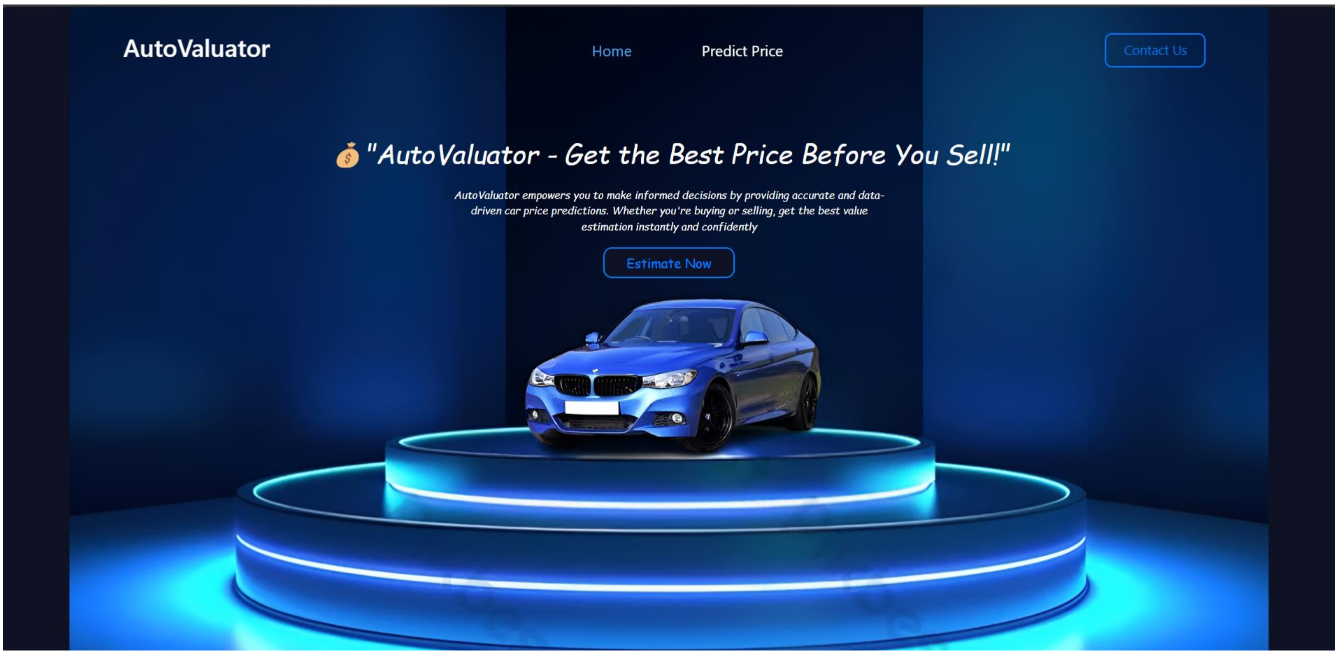
2. **Predict Price** :
   - This page includes a form-based interface where users can fill in essential car details like:
     - Brand, Model, Year
     - Fuel Type, Transmission
     - KMs Driven, Ownership
   - Each field has a dropdown or input box with hints, making it easier even for non-technical users.
   - A “Get Estimated Price” button is shown at the bottom to submit the form.

3. **Report Page (After Prediction):**
   - Once the price is predicted using the ML model, users are taken to a new page showing the estimated car price.
   - A neat, card-style layout displays:
     - User’s input summary
     - Predicted price
   - A "Download Your Report" button allows users to save a PDF version of their car valuation.

**4. Mobile-Friendly Design:**

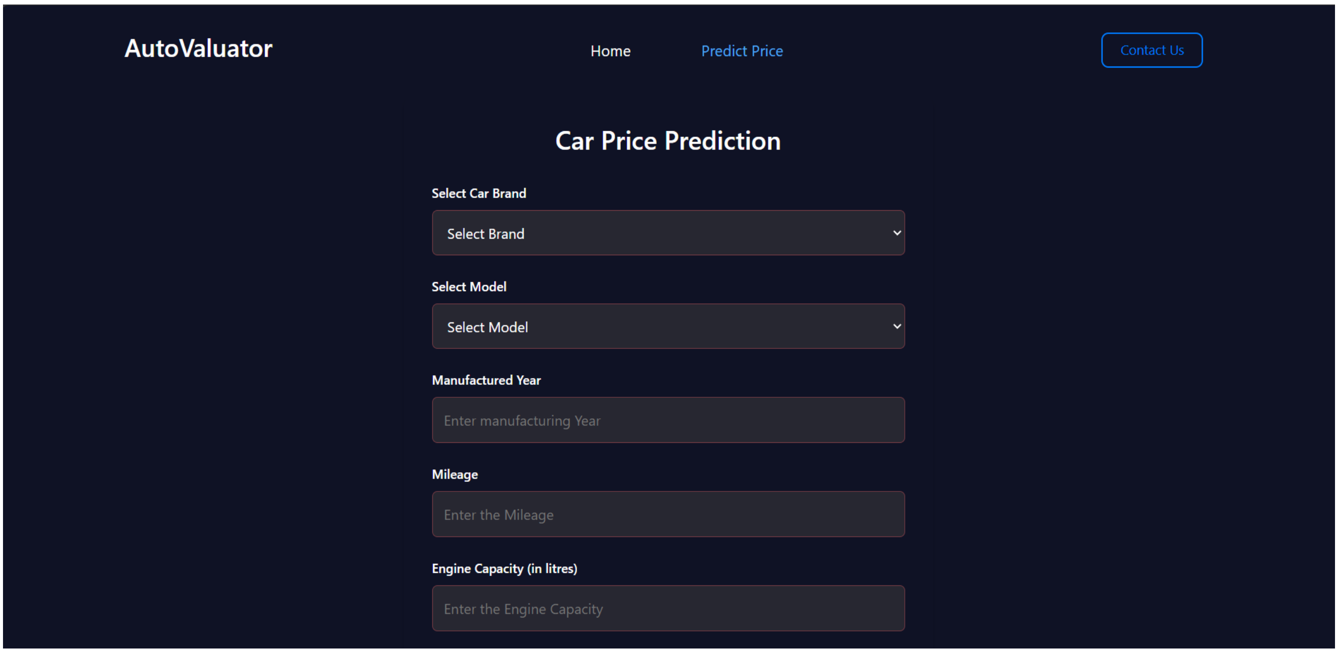
- The entire platform is responsive and works seamlessly on mobile devices.
- Buttons, forms, and text auto-adjust for smaller screens, ensuring a smooth experience on the go.
- All actions like form filling, prediction, and downloading reports can be done from a phone without hassle.

This simple, clean design makes sure users can find what they need without feeling overwhelmed. Every step is straightforward to ensure a smooth experience!



💰 "AutoValuator - Get the Best Price Before You Sell!"

*AutoValuator empowers you to make informed decisions by providing accurate and data-driven car price predictions. Whether you're buying or selling, get the best value estimation instantly and confidently*



Car Price Prediction

Select Car Brand

Select Brand

Select Model

Select Model

Manufactured Year

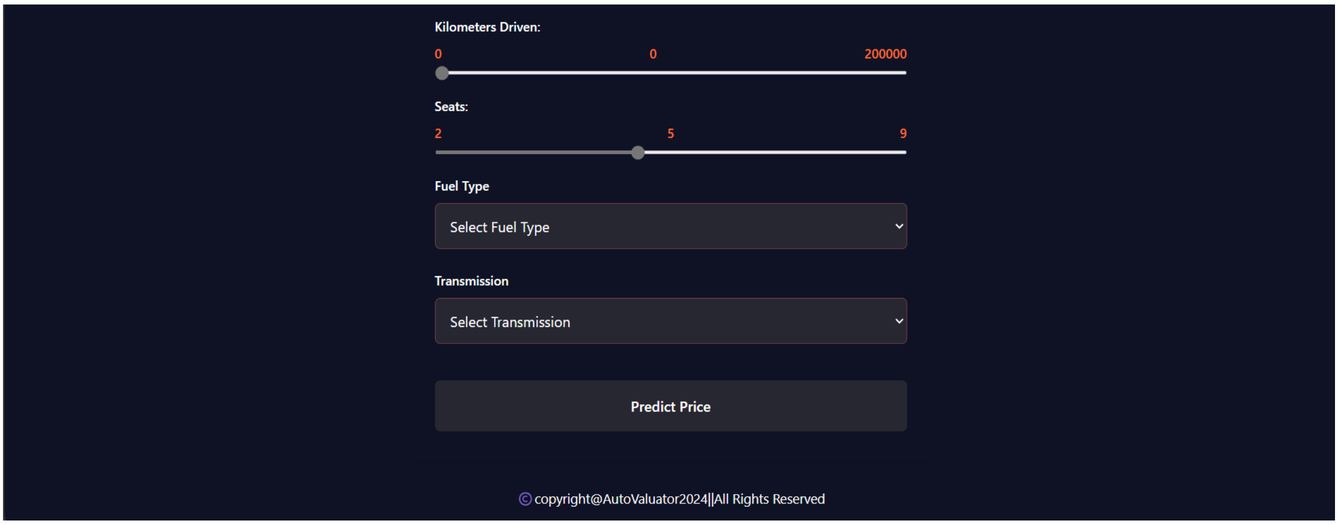
Enter manufacturing Year

Mileage

Enter the Mileage

Engine Capacity (in litres)

Enter the Engine Capacity



Kilometers Driven:

0 0 200000

Seats:

2 5 9

Fuel Type

Select Fuel Type

Transmission

Select Transmission

## 10. Technology Stack

Here's a simple breakdown of the technologies we'll use to build the project:

1. **Frontend (User Interface)**:

   - **React.js:** Used to create a dynamic and responsive user interface with component-based architecture.
   - **CSS:** For styling and designing a clean, modern, and visually appealing layout.
   - **JavaScript:** To handle interactivity, form validation, and dynamic UI behavior.

2. **Backend (Processing & Logic)**:

   - **Python:** Acts as the main backend processing language, managing logic and connecting the ML model with the frontend.
   - **Machine Learning Libraries:** Used to train, evaluate, and deploy the ML model for car price prediction.

     These are the list of libraries used in our project:

     - numpy
     - pandas
     - os
     - matplotlib.pyplot
     - sklearn.ensemble
     - sklearn.linear_model
     - sklearn.svm
     - sklearn.tree
     - sklearn.model_selection
     - sklearn.metrics
     - kagglehub
     - pickle

This tech stack ensures the platform runs smoothly, processes files quickly, and securely handles user data.

## 11. Testing Plan

To ensure that the AutoValuator platform performs accurately and smoothly for users, we'll follow a structured testing approach. Here's the plan:

1. **Unit Testing:**

- **What it is:** We will test individual components of the project separately — such as the form input fields, price prediction logic, and the ML model.
- **Why it's important:** This helps us identify any issues in small units before integrating them into the complete system.

2. **Integration Testing:**

- **What it is:** We'll check how different parts of the system work together — for example, whether user inputs from the frontend are correctly sent to the backend and whether the predicted price is properly shown on the frontend.
- **Why it's important:** Ensures seamless communication between UI and backend, and validates proper flow.

3. **User Testing:**

- **What it is:** Real users will test the platform by entering car details and predicting prices to see if the UI is intuitive and the results are understandable.
- **Why it's important:** Users give feedback from a practical perspective, helping to improve usability and fix hidden issues.

4. **Performance Testing:**

- **What it is:** We'll test the platform's speed and responsiveness when multiple users use it simultaneously or when complex data is entered.
- **Why it's important:** Ensures the platform performs well under load and provides results in real-time without delays.

5. **Security Testing:**

- **What it is:** We'll test for secure data handling — especially ensuring that the user inputs are protected from common threats like injection attacks or data breaches.
- **Why it's important:** Builds trust by ensuring that the website is safe and secure for public use.

6. **Bug Fixing and Retesting:**

- **What it is:** After discovering any bugs, we'll fix them and retest the full workflow — from user input to report download — to make sure everything works correctly.
- **Why it's important:** Fixes don't cause new issues, and the overall system remains stable.

## 12. Expected Outcome

1. **Smart Price Prediction Tool**
   • The project delivers a reliable and intelligent platform for users to predict the resale price of their cars using machine learning.
   • It provides instant, data-driven predictions based on various car attributes like model, year, fuel type, etc.

2. **Streamlined User Experience**
   • Users can easily input car details and get accurate price predictions within seconds.
   • Eliminates the need for manual price research or dealer dependence.

3. **Improved Decision-Making**
   • Helps users make informed decisions when selling their cars by offering fair market estimates.
   • Builds user confidence by providing a downloadable prediction report.

4. **User-Friendly Web Interface**
   • Built with React, CSS, and JavaScript, the UI is clean, modern, and easy to use for all age groups.
   • Supports both desktop and mobile viewing for accessibility on-the-go.

5. **Fast & Efficient Backend**
   • Powered by Python and a trained ML model, the backend handles predictions smoothly and responds quickly.
   • Ensures real-time output without delay, even with complex input data.

6. **Report Download Feature**
   • Users get a detailed car value report after prediction, which can be downloaded and shared.
   • Adds a professional touch and increases trust in the result.

7. **Data Privacy Focused**
   - No unnecessary data storage; user inputs are only used for real-time prediction.
   - Keeps user data secure and private throughout the session.

## 13. Resources And Limitation

To build the AutoValuator – Car Price Predictor, basic hardware like a laptop or PC is required for development and testing. Software tools include Visual Studio Code for coding, React, CSS, and JavaScript for frontend design, and Python with libraries like pandas and scikit-learn for backend and ML. Hosting platforms like Netlify and PythonAnywhere are used for deployment, and browsers/mobile devices for UI testing.

However, the project has some limitations. The prediction accuracy depends on the dataset used, and it doesn't support real-time market data. Rare or unique car models may produce less accurate results. Also, features like location-based pricing or image-based evaluation are not included in this version. Minor compatibility issues may occur on older devices or browsers.

## 14. Conclusion

In conclusion, this project aims to provide a simple and effective solution for users to predict accurate car prices based on key inputs like brand, model, fuel type, and more. With a clean, user-friendly interface and machine learning on the backend, the platform ensures that users can easily get an estimated price for their vehicle in just a few steps.

While there are some limitations, such as dependency on existing data and lack of real-time market updates, these can be improved with future enhancements. Overall, AutoValuator helps users make informed decisions, making the car-selling process faster, smarter, and more convenient.

## 15. Reference

1. S. Gupta, *"Machine Learning Models for Predictive Analysis in the Automotive Sector,"* Journal of Intelligent Systems, vol. 40, no. 2, pp. 88–94, Mar. 2022.
   https://doi.org/10.1016/j.jintsys.2022.03.002

2. L. Wang, *"Data-Driven Approaches to Used Car Valuation,"* International Journal of Data Science, vol. 27, no. 3, pp. 134–139, Jul. 2021.
   https://doi.org/10.1007/s41060-021-00272-3

3. A. Mehta, *"A Survey on Vehicle Price Prediction Using Machine Learning,"* IEEE Transactions on Artificial Intelligence, vol. 36, no. 5, pp. 210–215, Sept. 2020.
   https://doi.org/10.1109/TAI.2020.3034257

## Research Papers:

1. Car Price Prediction Using Data Science and Machine Learning https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2024/55247/final/fin_irjmets1714900105.pdf
2. Cars Price Prediction Using Machine Learning https://ijcrt.org/papers/IJCRT2404455.pdf
3. Predicting Used Car Prices with Deep Learning – Stanford University https://cs230.stanford.edu/projects_fall_2021/reports/102475818.pdf
4. ProbSAINT: Probabilistic Tabular Regression for Used Car Pricing https://arxiv.org/abs/2403.03812
5. Vehicle Price Prediction Using Machine Learning Algorithms https://www.ijert.org/research/vehicle-price-prediction-using-machine-learning-algorithms-IJERTCONV10IS02045.pdf
6. Car Price Prediction Using ML Techniques – IJIRT https://ijirt.org/published-paper/IJIRT151705_PAPER.pdf
7. Car Price Prediction Using ML – IJRPR Journal https://ijrpr.com/uploads/V5ISSUE1/IJRPR22107.pdf
8. Used Car Price Prediction Using Machine Learning https://www.ijraset.com/fileserve.php?FID=34209
9. Vehicle Price Prediction By Aggregating Decision Tree Model With Boosting Model
   https://arxiv.org/abs/2307.15982
10. Machine Learning for Predicting Automobile Prices – SSRN https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3852339