**IT-314**
**Software Engineering** **Prof. Saurabh Tiwari**

# IRIS - Vehicle Recognition Website


**Dhirubhai Ambani Institute of Information and Communication Technology**

| STUDENT ID | STUDENT NAME |
|---|---|
| Nisarg Patel | 201801013 |
| Khyati Bhuva | 201801119 |
| Palak Verma | 201801161 |
| Swapnil Dhola | 201801163 |
| Deep Malani | 201801184 |
| Siddharth Chauhan | 201801196 |
| Ajay Vala | 201801414 |
| Mohil Khimani | 201801416 |
| Shabbir Murtaza | 201801428 |
| Vishal Patel | 201801436 |

# INDEX

## 1. Problem Statement:

The main aim of the project is to develop Vehicle Optical Recognition Technology. As input, users can snap a photo and get the result within just a few seconds. It will provide the details about the vehicle make, model, year and starting retail price. In addition, it will provide this same information for the two closest competitors so a user can do some quick comparison among the vehicles.

## 2. Functional Requirements.

1. **Image upload:** Users can upload an image from a device or snap a photo from a camera.
2. **Identification:** Identify the vehicle from the image uploaded by the user.
3. **Recommendation of similar vehicles:** After identifying the vehicle from the image, the user will be shown recommendations of similar vehicles based on price and type of vehicle.
4. **Additional Filters:** Users will have an option to find possible vehicles with the applied filters like on-road price, engine power and gear transmission type, fuel and performance etc.
5. **Comparison between vehicles:** User can select model and variant of vehicle which it wants to compare. As a result, the specifications can be compared in form of table.
6. **Invalid input:** When the user enters an image which doesn't contain a vehicle or if the image is blurred, then the user should be asked to enter a valid image. When a user uploads something that isn't in image format(.txt, .bin file), then the product asks them to upload images/files of specific extensions only.

## 3. Non Functional Requirements:

a. <u>Product Requirements:</u>
   i. **Performance:** The output should be given within a certain amount of time.
   ii. **Accessibility:** The product should be available to users at all times.
   iii. **Security:** The product code and dataset should be secured and protected from any malicious activity.
   iv. **Maintenance:** The maintenance should be quick and timely updates should be made. Also, the build and maintenance cost should be low.
   v. **Efficiency:** The model should give accurate output.
   vi. **Usability:** UI/UX should be easy to use and be understandable by any new user.
   vii. **Flexibility:** For Dataset for any type of vehicle, it should easily use this model.
   viii. **Reusability**: It can be reused for making other projects like price comparison from different sites.

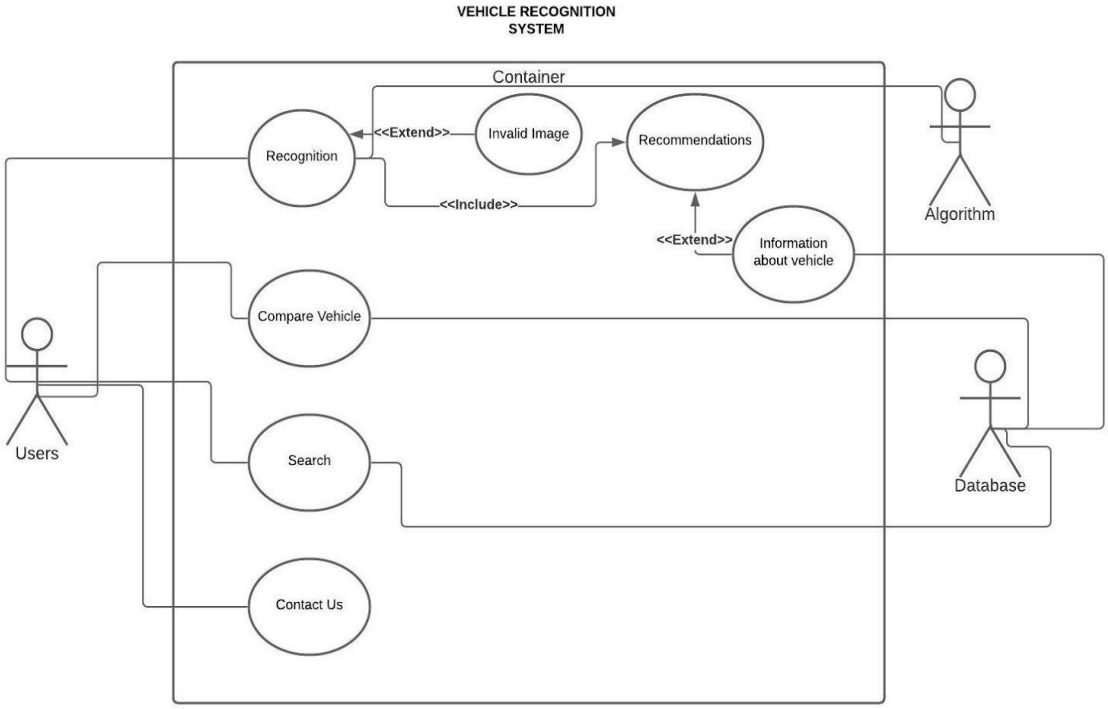
b. External requirements:
   i. **Standard constraints:** The prediction rate of our model.
   ii. **Stability requirements:** System should be able to handle a specific amount of users at once and should not be crashed frequently.
   iii. **Robustness requirements:** Incase the system crashes, it should ensure data is safe and data is recovered quickly.
   iv. **Portability:** It should work on or above certain versions of browsers(mobile and PCs).
   v. **Scalability:** If the user scale increases to state/national level, the project should be able to incorporate changes accordingly.

c. Domain requirements:
   - As our app is based on a machine learning recognition system, we would require our app to give accuracy of recognizing vehicles to be more than certain percentage.
   - Backend server must be operational for the entirety of the time

## 4. Elicitation Techniques:

- Stakeholder analysis- We identified the stakeholders and assigned roles.
- Analysis of existing systems or documentation - We analyzed the features of Google Lens for Optical Recognition part and analyzed CarDekho.com for the features which can be added in our scenario.
- Background reading - We read many articles and research papers on Object Classification and recognition and also its implementation.
- Task observation - We checked different websites for observation
- Questionnaires - We circulated a google form among peers
- Interviewing - We Interviewed people who submitted the features to be added in the google forms. We tried to understand their point of view and will try to add those features in our website.
- Brainstorming - We discussed what could be done to realise the project
- Use cases and scenarios - The user might upload invalid pictures. The system might give wrong answers as recognition. The recommendations that are shown to the user might be not related to the car in the picture.

## 5. Use Case Diagram




## 6. User Stories

1. **Must have:**

- As a user, I want to be able to compare two cars so that I can find differences between them.
- As a user, I want to be able to filter out cars so that I can see cars that fit my requirements.
- As a user, I want to be able to search for elaborate details regarding a specific car.
- As a user, I want to be able to see recommended cars so that I can see similar cars.
- As a developer, I want to be able to access the image database so that I can recognise the car in the picture.
- As a developer, I want to be able to access the model so that I can recognise the car in the picture.

2. **Should Have:**

- As a user, I want to be able to see details of the developers so that I can contact them
- As a user, I want to be able to get information about the car by uploading pictures of the car.
- As a user, I want to be able to upload images from any device easily so that I can use the product anywhere.

- As a user, I need the database to stay updated so that I get the results as per the latest models available.
- As an owner, I want the product to handle a large number of requests so that there is no down time.
- As a developer, I want to be able to access the csv which contains the details about the cars so that I can display the car details.
- As a developer I want the system to alert the user if only one of the cars is selected and the user has pressed the submit button.
- As a developer I want the system to acknowledge that user's feedback has been recorded.
- As a developer, I want to be able to access the url which will be generated after the submit button is clicked so that I can process it.

3. **Could Have:**

- As a user, I want to be able to contact the concerned persons so that they can solve my issues.
- As a user I want the system to give me correct information about the car with at least 90% accuracy.
- As an owner, I want the system to be user friendly and easy to use so that more people can be benefited from using this product.

## 7. Implementation strategy of your project with a detailed classification of Sprints

[1 Feb - 7 Feb] Sprint1: Within the first week, brainstorming was performed with all the members. All the ideas about the problem statement were collected. Various meetings were conducted to realise what we actually wanted to see and implement in our product. We tried to search various already existing such products and websites like cardekho, cars24, and also some other projects on github.

[8 Feb - 14 Feb] Sprint 2: For the second week, all the ideas collected were sorted and prioritized according to necessity. We applied some techniques to come to the conclusion in which order we should start our implementation. Basic roadmap for completion of the tasks was decided along with time milestones.

[15 Feb - 21 Feb] Sprint 3: The third week consisted of planning the functionalities and design of the system and prioritizing them. We had decided to implement four basic functionalities and we conducted many meetings to decide the overall look of the design. We drew the rough sketch while considering everyone's opinion.

[22 Feb - 26 Feb] Sprint 4: In the fourth week, members were divided into four teams: Front End, Model, Database, Back end Teams. We were asked to volunteer and given the option to join any of the teams. We decided that these teams will function independently at an independent stage and will ask for help or suggestions when needed or at weekly meetings.

[8 March - 14 March] Sprint 5: For the fifth week, Model team had explored image datasets, Front End team explored suitable technologies, Database team explored database of vehicles and Back end team explored how to combine Front End with Model and Database. Everyone was exploring to find the best tech stack that can be used in our product. We thought thoroughly exploring will lead to less issues like incompatibility in our project. Since then we had our mid sem exams from 27th Feb till 6th March, we did not include this in our sprint since we could not really do anything.

[15 March - 28 March] Sprint 6: For the next two weeks, teams started the initial stage of implementation. For this we finalised what tech stack and languages will be used to implement the functionalities. At the weekly meeting, we discussed problems we faced, tried finding solutions together.

[29 March - 4 April] Sprint 7: Teams are working on pending development. All team members together reviewed the overall project and planned individual activities. Suggestions were given by each member regarding the changes that could be done or if there was a better way to do the things that were implemented. This is also when our mid evaluation was conducted.

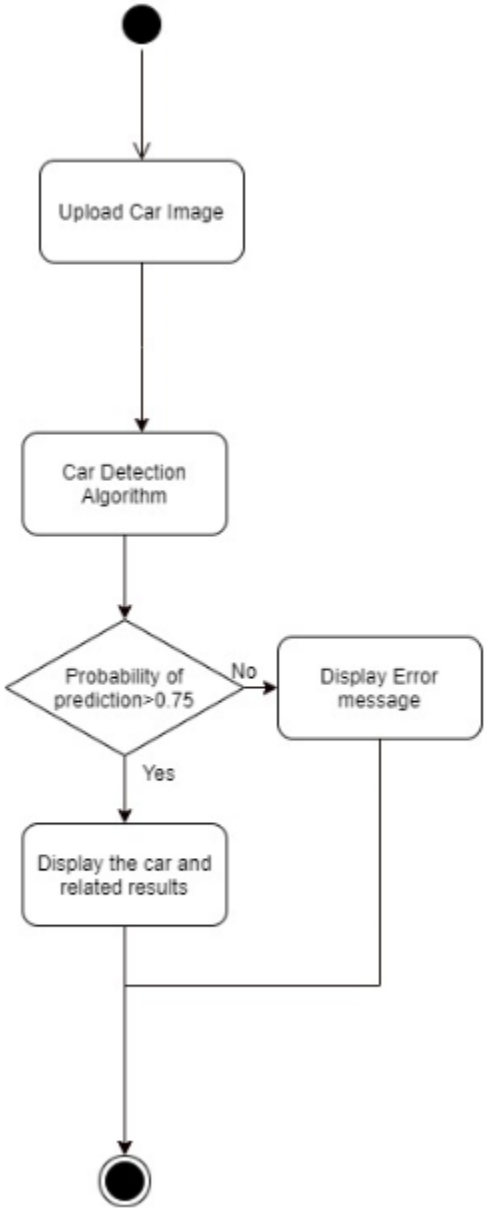
[5 April - 11 April] Sprint 8: We have completed above 70 percent of work. Front end had completed two web pages. Model had achieved 85 percent accuracy. Set up for integration was completed. Half of the dataset for the details was completed. We discussed progress and reviewed our next plans.

[12 April - 18 April] Sprint 9: Front End completed all the web pages. Minor touch ups were pending. The model accuracy improved to 92 percent. The database team completed their collection of all details of cars and others members helped them to collect images for the image dataset. In the integration part, the backend team had finished functionalities for home, compare and contact us page. They started working on implementing the filter page functionality.

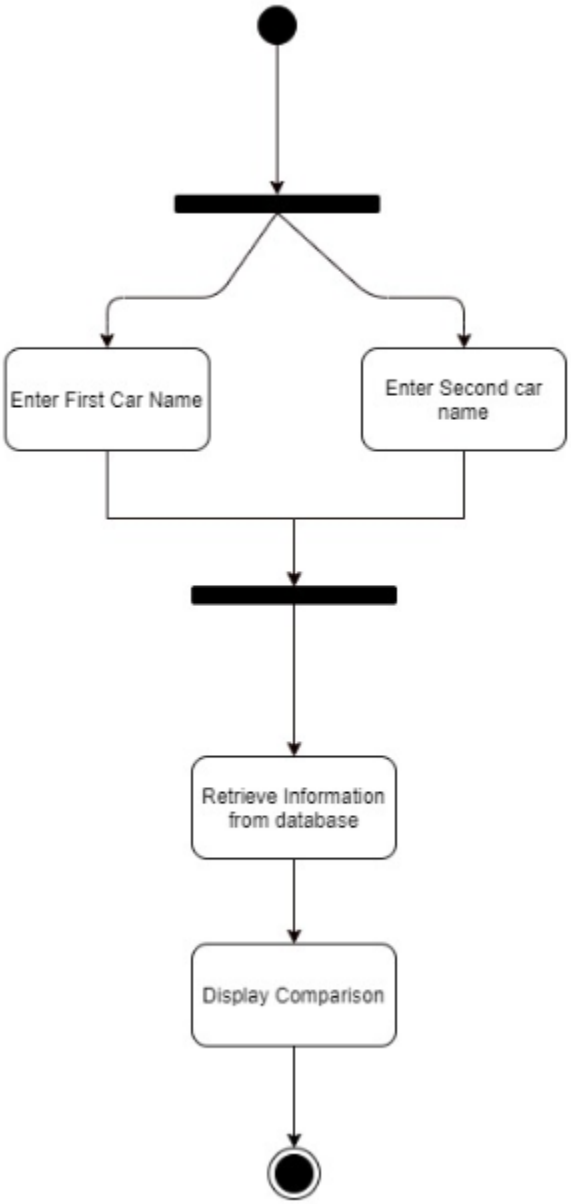
[19 April - 25 April] Sprint 10: Backend team completed the filter page. Front end team did some of the minor touch ups after integration. The model accuracy improved to 96 percent. The integration team reviewed every integration and performed integration testing after each team had completed the unit testing. Then in a final meeting, we met to recheck the overall product and performed system testing. Since then we had our final exams from 26th April till 1st May, we did not include this in our sprint since we could not really do anything.

[1 April - 9 May] Sprint 11: The Members started working on documentation and presentation for the submission along with making the video. We met regularly to decide on our content progressively. In the end, we completed our preparation of materials for submission and also had a run through.

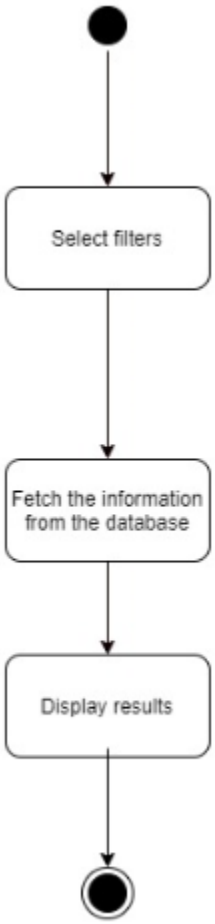
## 9. Activity Diagram



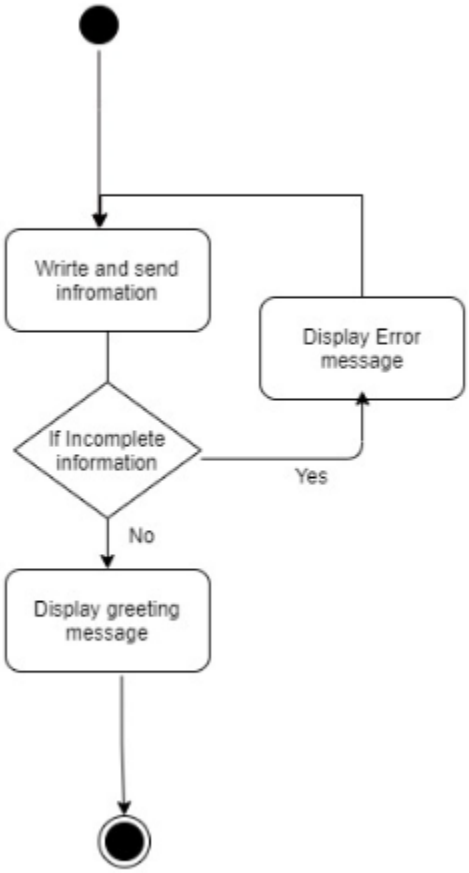
## COMPARE PAGE



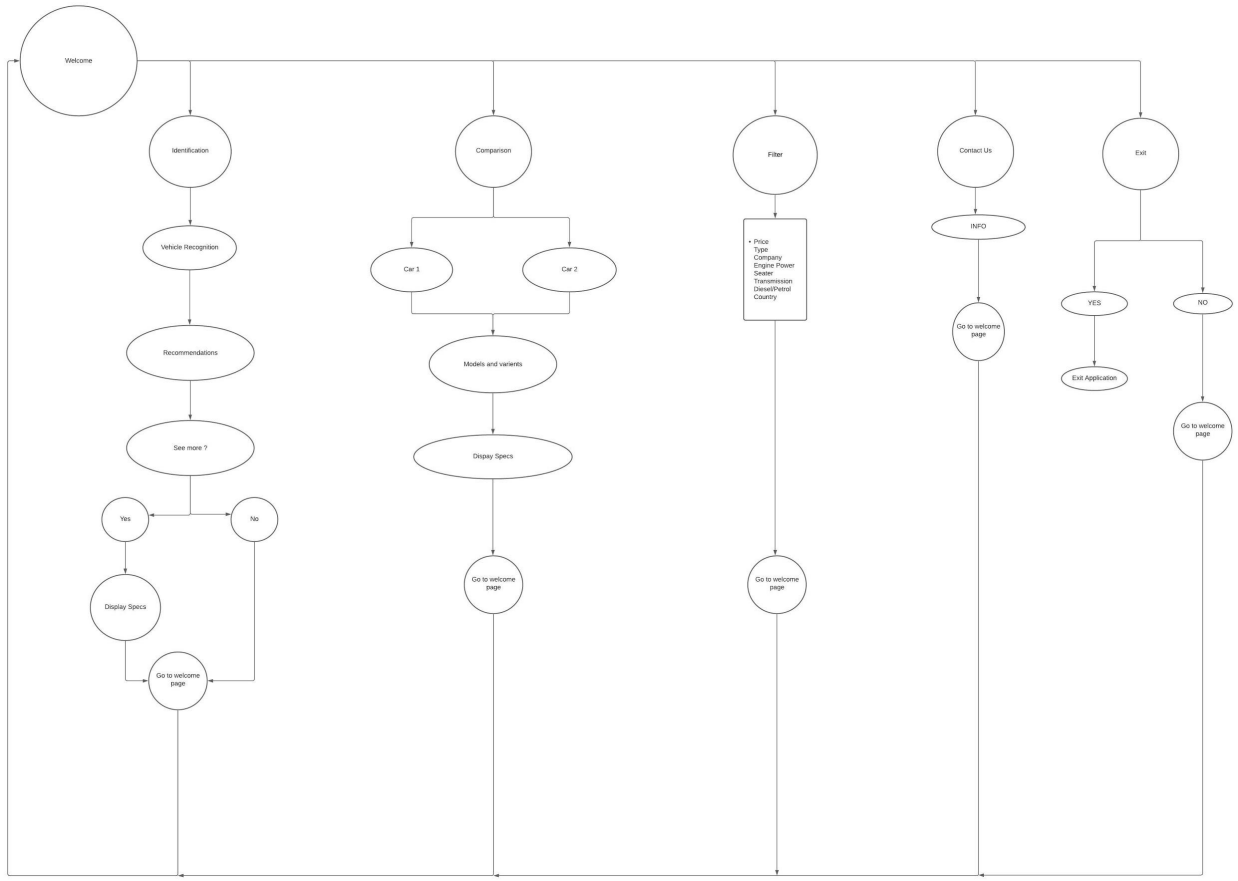
## FILTER PAGE


Select filters
Fetch the information from the database
Display results

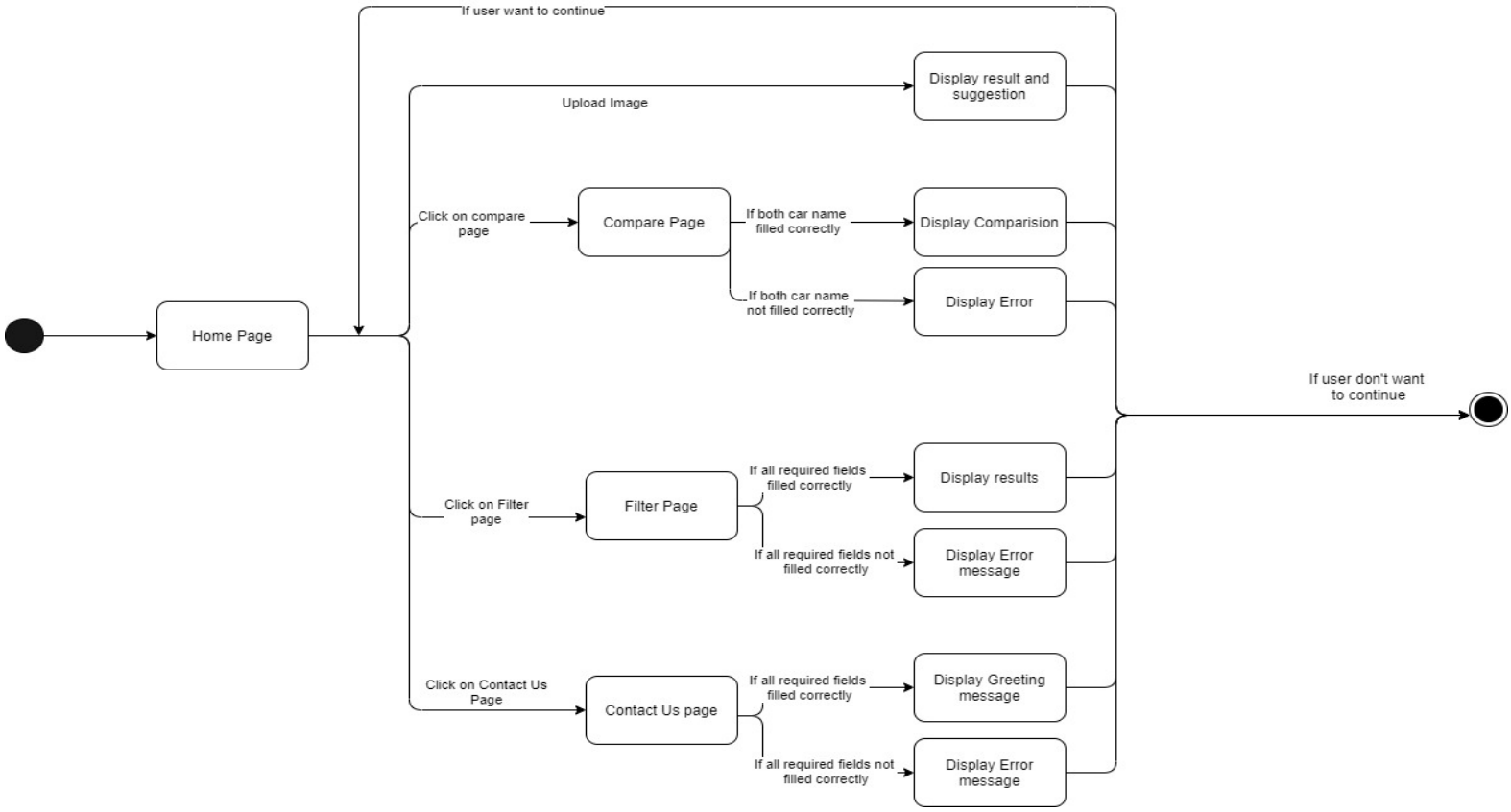
## CONTACT US PAGE



## 9. Concept Map

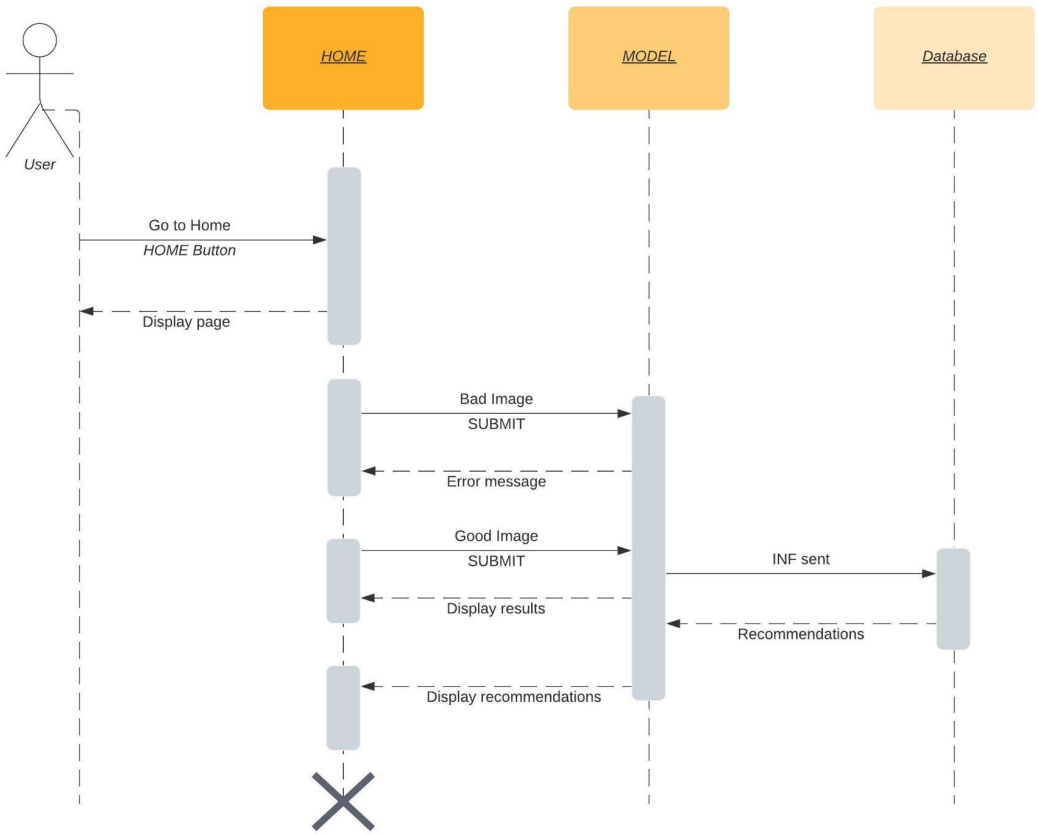


## 10. State Diagram

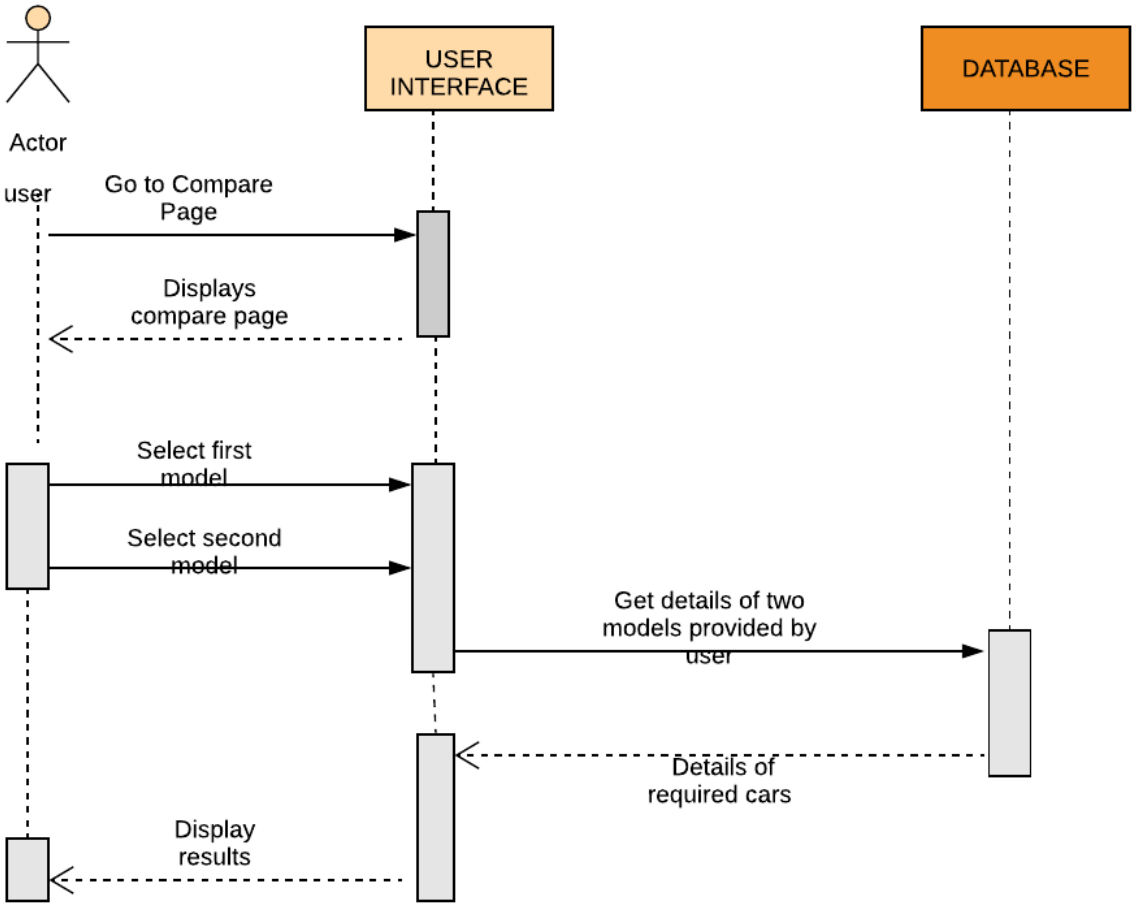


## 11. Sequence Diagrams:

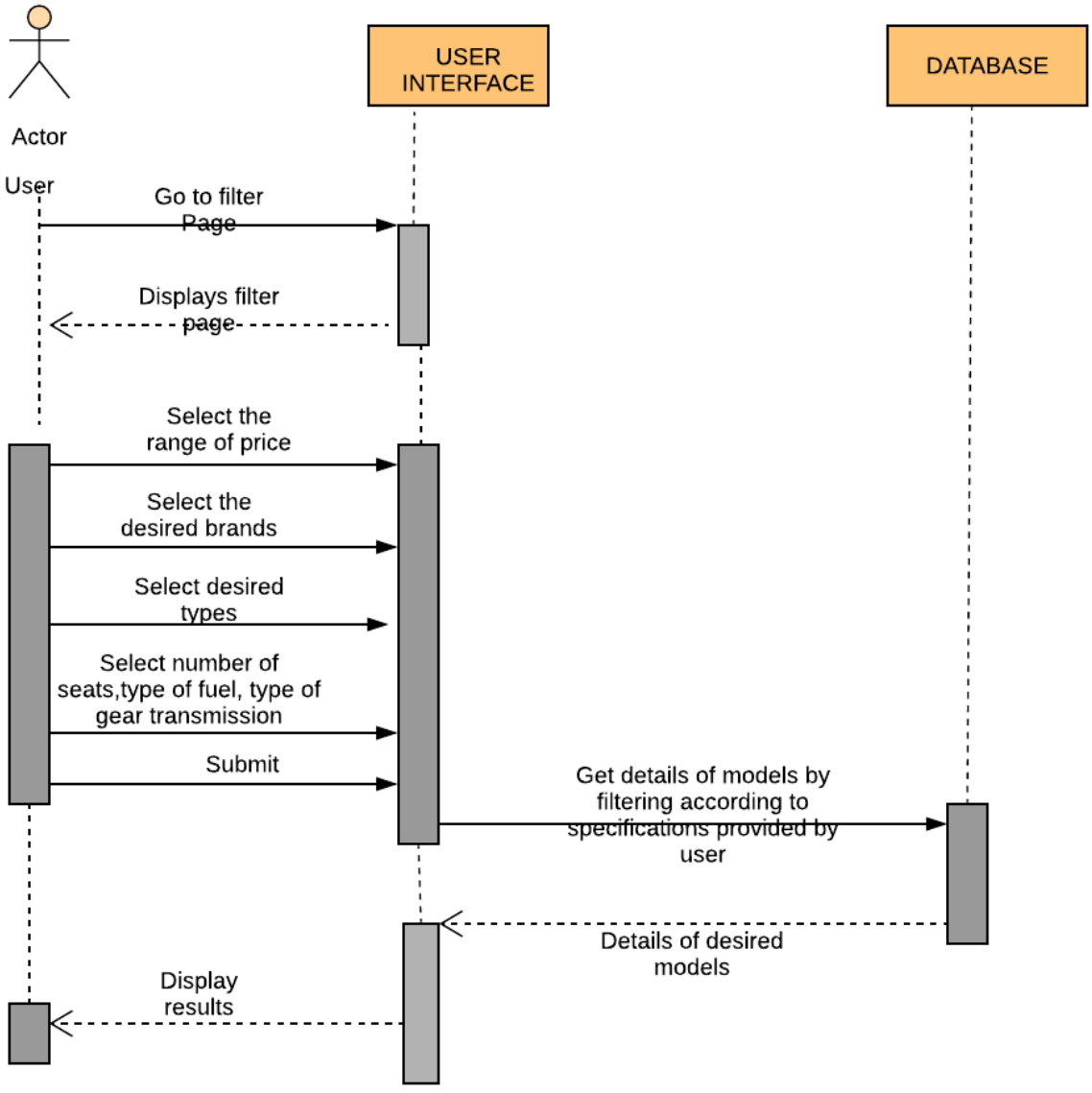
### HOME PAGE



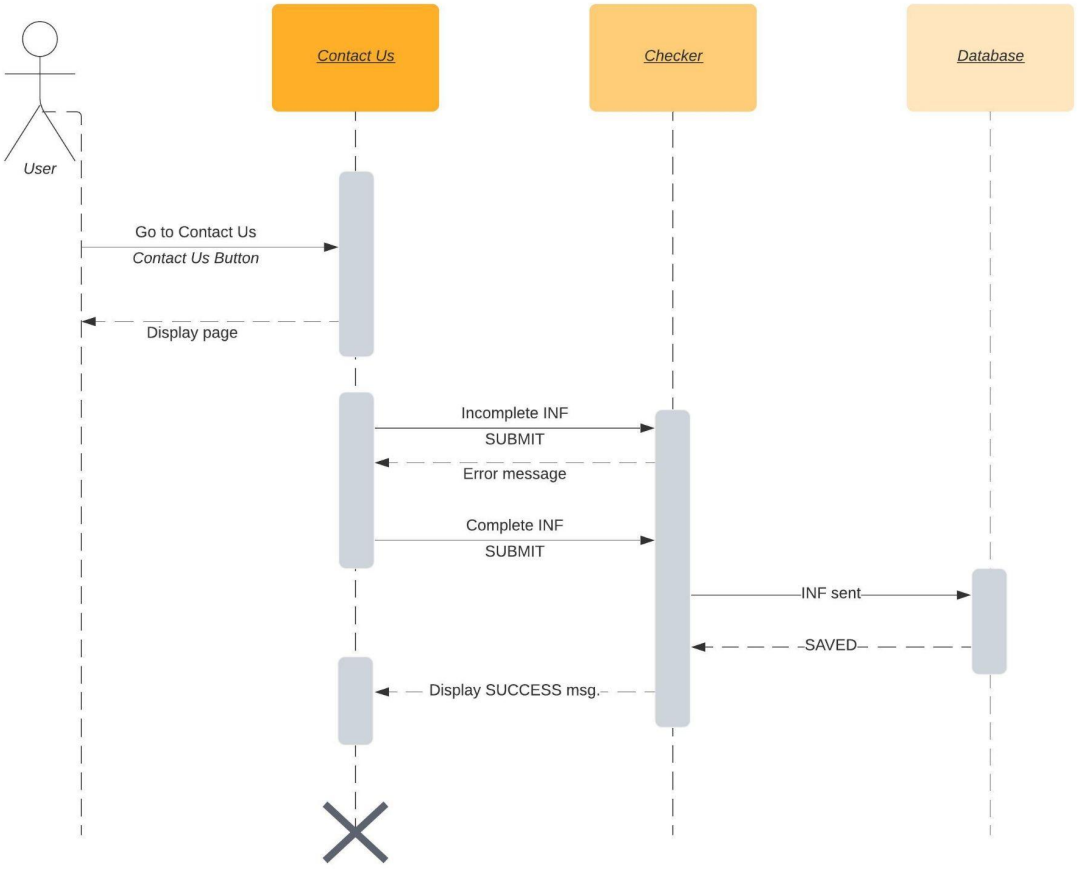
# COMPARE PAGE



# FILTER PAGE


Actor
User
USER INTERFACE
DATABASE
Go to filter Page
Displays filter page
Select the range of price
Select the desired brands
Select desired types
Select number of seats,type of fuel, type of gear transmission
Submit
Get details of models by filtering according to specifications provided by user
Details of desired models
Display results

# CONTACT US PAGE



## 12. Testing:

### A.) Model Testing

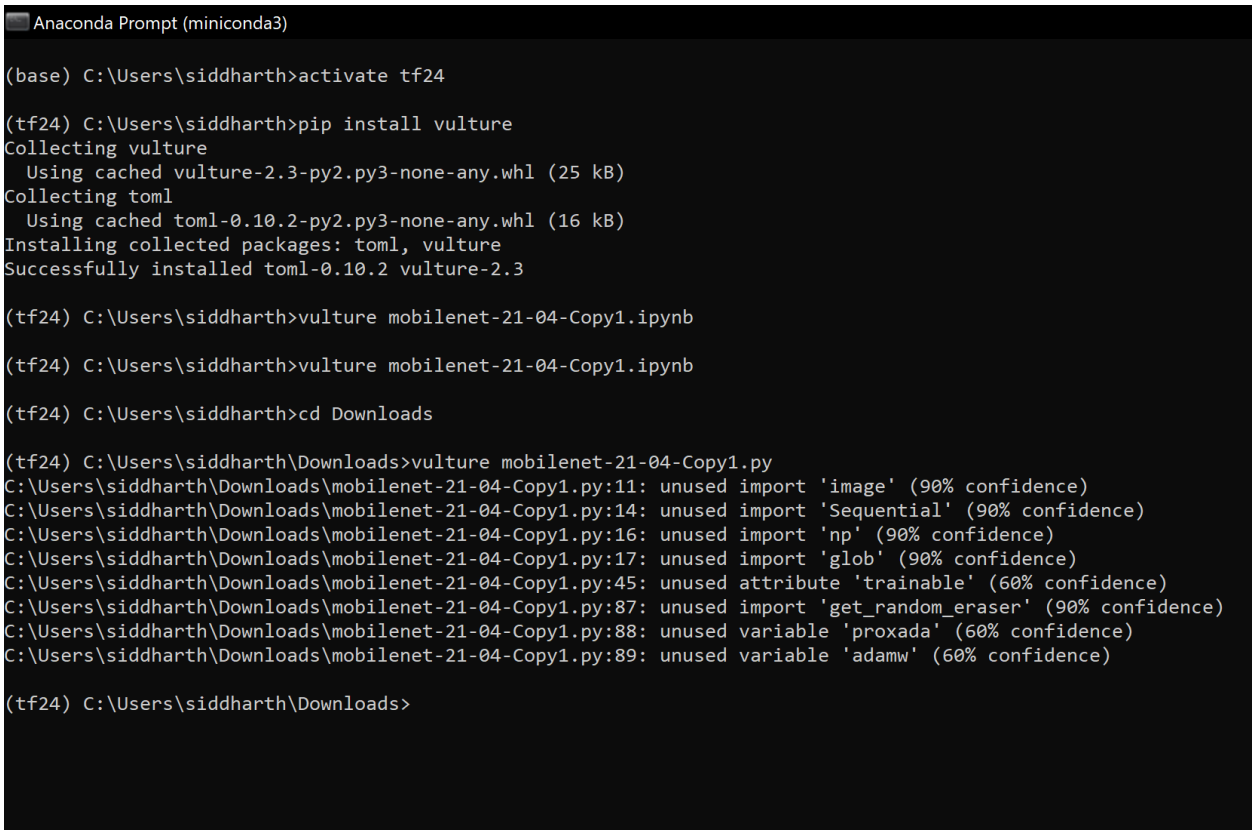
**Accuracy of different models for 20 classes.**

- ➔ VGG16 - 80%
- ➔ VGG19 - 80%
- ➔ Xception - 75%
- ➔ Resnet50 - 35%
- ➔ MobileNet - 80-85%
- ➔ InceptionResNetV2 - 75-80%
- ➔ InceptionV3 - 70-75%

### Static Analysis

For Static analysis we have used the Vulture static analysis tool. Vulture finds unused code in large python code. This tool is used for finding and removing errors and unused code in large python code.
Here results given below:

**Run 1**



During the run of vulture static analyzer, it gave the information about which function or library was not getting used .

**Run 2:**

```
Anaconda Prompt (miniconda3)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy1.py:17: unused import 'glob' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy1.py:45: unused attribute 'trainable' (60% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy1.py:87: unused import 'get_random_eraser' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy1.py:88: unused variable 'proxada' (60% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy1.py:89: unused variable 'adamw' (60% confidence)

(tf24) C:\Users\siddharth\Downloads>vulture mobilenet-21-04-Copy1(1).py
Error: mobilenet-21-04-Copy1(1).py could not be found.

(tf24) C:\Users\siddharth\Downloads>vulture mobilenet-21-04-Copy.py
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:10: unused import 'Model' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:11: unused import 'image' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:13: unused import 'MobileNet' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:17: unused import 'tfa' (90% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:47: unused attribute 'trainable' (60% confidence)
C:\Users\siddharth\Downloads\mobilenet-21-04-Copy.py:49: unused attribute 'trainable' (60% confidence)

(tf24) C:\Users\siddharth\Downloads>vulture mobilenet-21-04-Copy0.py

(tf24) C:\Users\siddharth\Downloads>vulture mobilenet-21-04-Copy0.py

(tf24) C:\Users\siddharth\Downloads>
```

After removing unwanted libraries and functions the analyzer was not giving any warnings. After refining code all errors and warnings got resolved.

## Class wise Prediction:

Here the aim of this classwise testing to check whether any class is giving very less accuracy or not. I.e whether a model is predicting wrong results for each picture of any particular image class or not.

**Total Prediction rate = 96%**

| | Car | Accuracy |
|---|---|---|
| 0 | testing\AM General Hummer SUV 2000 | 1 |
| 1 | testing\Acura RL Sedan 2012 | 0.8 |

| | | |
|---|---|---|
| 2 | testing\Acura TL Sedan 2012 | 1 |
| 3 | testing\Acura ZDX Hatchback 2012 | 0.85 |
| 4 | testing\Aston Martin V8 Vantage Convertible 2012 | 0.85 |
| 5 | testing\Aston Martin Virage Convertible 2012 | 0.95 |
| 6 | testing\Audi A5 Coupe 2012 | 1 |
| 7 | testing\Audi R8 Coupe 2012 | 1 |
| 8 | testing\Audi S4 Sedan 2012 | 0.9 |
| 9 | testing\Audi S5 Convertible 2012 | 0.8 |
| 10 | testing\Audi S6 Sedan 2011 | 1 |
| 11 | testing\Audi TT RS Coupe 2012 | 0.9 |
| 12 | testing\BMW 1 Series Convertible 2012 | 0.95 |
| 13 | testing\BMW 3 Series Sedan 2012 | 0.85 |
| 14 | testing\BMW ActiveHybrid 5 Sedan 2012 | 1 |

| | | |
|---|---|---|
| 15 | testing\BMW M3 Coupe 2012 | 1 |
| 16 | testing\BMW M5 Sedan 2010 | 0.95 |
| 17 | testing\BMW M6 Convertible 2010 | 1 |
| 18 | testing\BMW X3 SUV 2012 | 0.95 |
| 19 | testing\BMW X5 SUV 2007 | 0.95 |
| 20 | testing\BMW X6 SUV 2012 | 0.9 |
| 21 | testing\BMW Z4 Convertible 2012 | 0.95 |
| 22 | testing\Bentley Arnage Sedan 2009 | 0.9 |
| 23 | testing\Bentley Continental GT Coupe 2012 | 1 |
| 24 | testing\Bentley Mulsanne Sedan 2011 | 1 |
| 25 | testing\Bugatti Veyron 16.4 Convertible 2009 | 1 |
| 26 | testing\Buick Enclave SUV 2012 | 1 |
| 27 | testing\Cadillac CTS-V Sedan 2012 | 1 |

| | | |
|---|---|---|
| 28 | testing\Cadillac SRX SUV 2012 | 1 |
| 29 | testing\Chevrolet Avalanche Crew Cab 2012 | 1 |
| 30 | testing\Chevrolet Camaro Convertible 2012 | 0.85 |
| 31 | testing\Chevrolet Corvette Convertible 2012 | 1 |
| 32 | testing\Chevrolet Corvette ZR1 2012 | 0.85 |
| 33 | testing\Chevrolet Impala Sedan 2007 | 1 |
| 34 | testing\Chevrolet Malibu Sedan 2007 | 0.85 |
| 35 | testing\Chevrolet Monte Carlo Coupe 2007 | 0.95 |
| 36 | testing\Chevrolet Silverado 1500 Classic Extended Cab 2007 | 1 |
| 37 | testing\Chevrolet Sonic Sedan 2012 | 1 |
| 38 | testing\Chevrolet Tahoe Hybrid SUV 2012 | 0.8 |
| 39 | testing\Chevrolet Traverse SUV 2012 | 0.95 |
| 40 | testing\Chrysler Aspen SUV 2009 | 1 |

| | | |
|---|---|---|
| 41 | testing\Chrysler PT Cruiser Convertible 2008 | 1 |
| 42 | testing\Chrysler Sebring Convertible 2010 | 1 |
| 43 | testing\Dodge Caliber Wagon 2012 | 1 |
| 44 | testing\Dodge Challenger SRT8 2011 | 1 |
| 45 | testing\Dodge Charger Sedan 2012 | 1 |
| 46 | testing\Dodge Durango SUV 2012 | 1 |
| 47 | testing\Dodge Journey SUV 2012 | 1 |
| 48 | testing\Dodge Magnum Wagon 2008 | 1 |
| 49 | testing\Dodge Ram Pickup 3500 Crew Cab 2010 | 0.95 |
| 50 | testing\FIAT 500 Abarth 2012 | 1 |
| 51 | testing\FIAT 500 Convertible 2012 | 1 |
| 52 | testing\Ferrari 458 Italia Convertible 2012 | 0.9 |
| 53 | testing\Ferrari FF Coupe 2012 | 1 |

| | | |
|---|---|---|
| 54 | testing\Ford Edge SUV 2012 | 1 |
| 55 | testing\Ford Fiesta Sedan 2012 | 1 |
| 56 | testing\Ford Focus Sedan 2007 | 0.9 |
| 57 | testing\Ford Freestar Minivan 2007 | 1 |
| 58 | testing\Ford Mustang Convertible 2007 | 0.95 |
| 59 | testing\GMC Acadia SUV 2012 | 1 |
| 60 | testing\GMC Canyon Extended Cab 2012 | 0.9 |
| 61 | testing\GMC Savana Van 2012 | 1 |
| 62 | testing\GMC Terrain SUV 2012 | 1 |
| 63 | testing\Honda Accord Coupe 2012 | 1 |
| 64 | testing\Hyundai Accent Sedan 2012 | 0.85 |
| 65 | testing\Hyundai Elantra Sedan 2007 | 0.95 |
| 66 | testing\Hyundai Elantra Touring Hatchback 2012 | 1 |

| | | |
|---|---|---|
| 67 | testing\Hyundai Santa Fe SUV 2012 | 1 |
| 68 | testing\Hyundai Sonata Hybrid Sedan 2012 | 0.95 |
| 69 | testing\Hyundai Sonata Sedan 2012 | 0.9 |
| 70 | testing\Hyundai Tucson SUV 2012 | 1 |
| 71 | testing\Hyundai Veracruz SUV 2012 | 0.9 |
| 72 | testing\Jaguar XK XKR 2012 | 0.9 |
| 73 | testing\Jeep Compass SUV 2012 | 1 |
| 74 | testing\Lamborghini Aventador Coupe 2012 | 0.95 |
| 75 | testing\Lamborghini Reventon Coupe 2008 | 1 |
| 76 | testing\Mercedes-Benz C-Class Sedan 2012 | 0.95 |
| 77 | testing\Mercedes-Benz E-Class Sedan 2012 | 1 |
| 78 | testing\Mercedes-Benz S-Class Sedan 2012 | 1 |
| 79 | testing\Mercedes-Benz SL-Class Coupe 2009 | 0.95 |

| | | |
|---|---|---|
| 80 | testing\Mitsubishi Lancer Sedan 2012 | 0.95 |
| 81 | testing\Nissan 240SX Coupe 1998 | 1 |
| 82 | testing\Nissan Juke Hatchback 2012 | 1 |
| 83 | testing\Nissan Leaf Hatchback 2012 | 1 |
| 84 | testing\Porsche Panamera Sedan 2012 | 0.95 |
| 85 | testing\Rolls-Royce Ghost Sedan 2012 | 0.95 |
| 86 | testing\Rolls-Royce Phantom Sedan 2012 | 0.9 |
| 87 | testing\Spyker C8 Convertible 2009 | 1 |
| 88 | testing\Suzuki Aerio Sedan 2007 | 0.95 |
| 89 | testing\Suzuki Kizashi Sedan 2012 | 1 |
| 90 | testing\Suzuki SX4 Hatchback 2012 | 1 |
| 91 | testing\Tesla Model S Sedan 2012 | 0.95 |
| 92 | testing\Toyota 4Runner SUV 2012 | 1 |

| | | |
|---|---|---|
| 93 | testing\Toyota Camry Sedan 2012 | 0.95 |
| 94 | testing\Toyota Corolla Sedan 2012 | 0.95 |
| 95 | testing\Toyota Sequoia SUV 2012 | 0.95 |
| 96 | testing\Volkswagen Beetle Hatchback 2012 | 1 |
| 97 | testing\Volkswagen Golf Hatchback 1991 | 1 |
| 98 | testing\Volvo C30 Hatchback 2012 | 0.9 |
| 99 | testing\Volvo XC90 SUV 2007 | 1 |

Here the model is giving good accuracy for each image class. The least accuracy is 0.8 and maximum accuracy in predicting image is 1. The overall accuracy of the model is 96%.

### BlackBox testing

Here for blackbox testing we have divided inputs into two classes. Here input is in the form of an image.

**Valid Class:**

Expected: Valid

Verdict: Valid
Test: Pass

Expected: Valid
Verdict: Valid
Test: Pass

**Invalid class**

Expected: Invalid
Verdict: Invalid
Test: Pass

Expected: Invalid
Verdict: Invalid
Test: Pass

Expected: Invalid

Verdict: Invalid
Test: Pass

Expected: Invalid
Verdict: Invalid
Test: Pass

**Boundary value analysis**

Expected: Invalid
Verdict: Invalid
Test: Pass

Expected: Invalid
Verdict: Invalid
Test: Pass

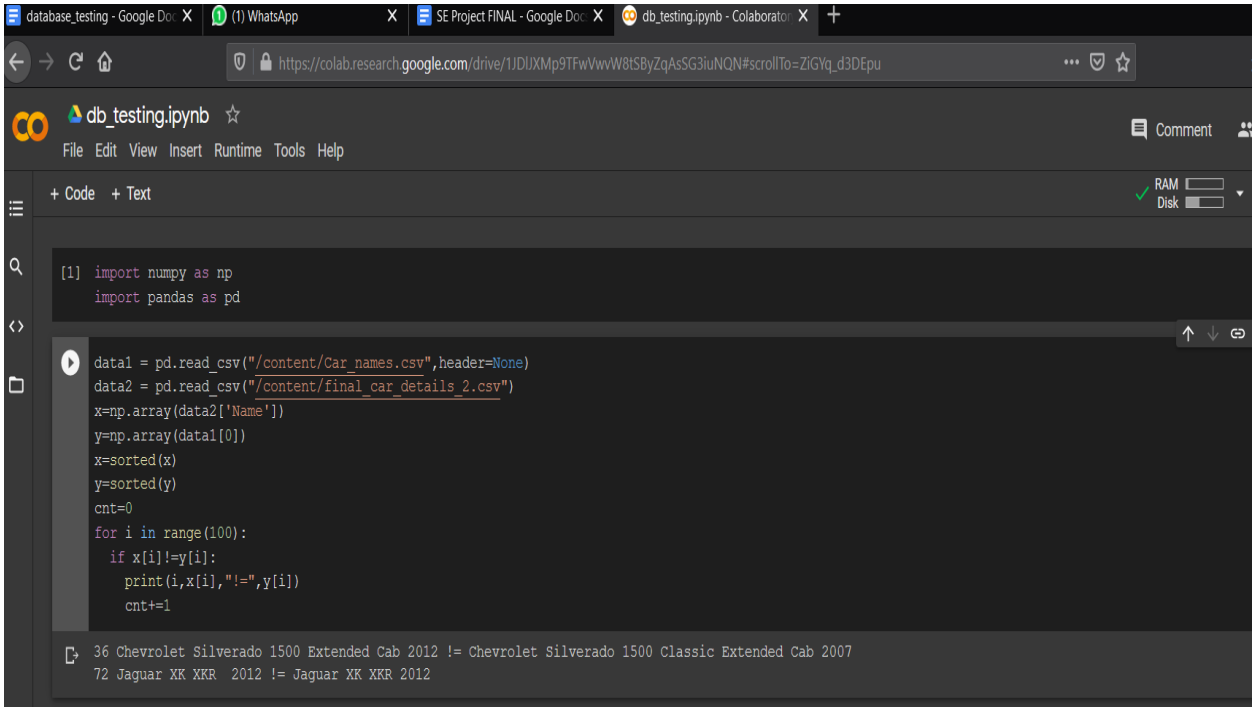
Expected: Valid

Verdict: Valid
Test: Pass


Expected: Valid
Verdict: Valid
Test: Pass

**B.) Database Testing:**

In our product, the database was created in postgresql but before creating it we tested the dataset we created by manually searching the models with the models included in the Machine Learning model part. The testing was done to ensure that the model names used in the ML part of our product matches with the dataset we created in order to ensure that it does not cause any problem with the subsequent steps in the development of our product.

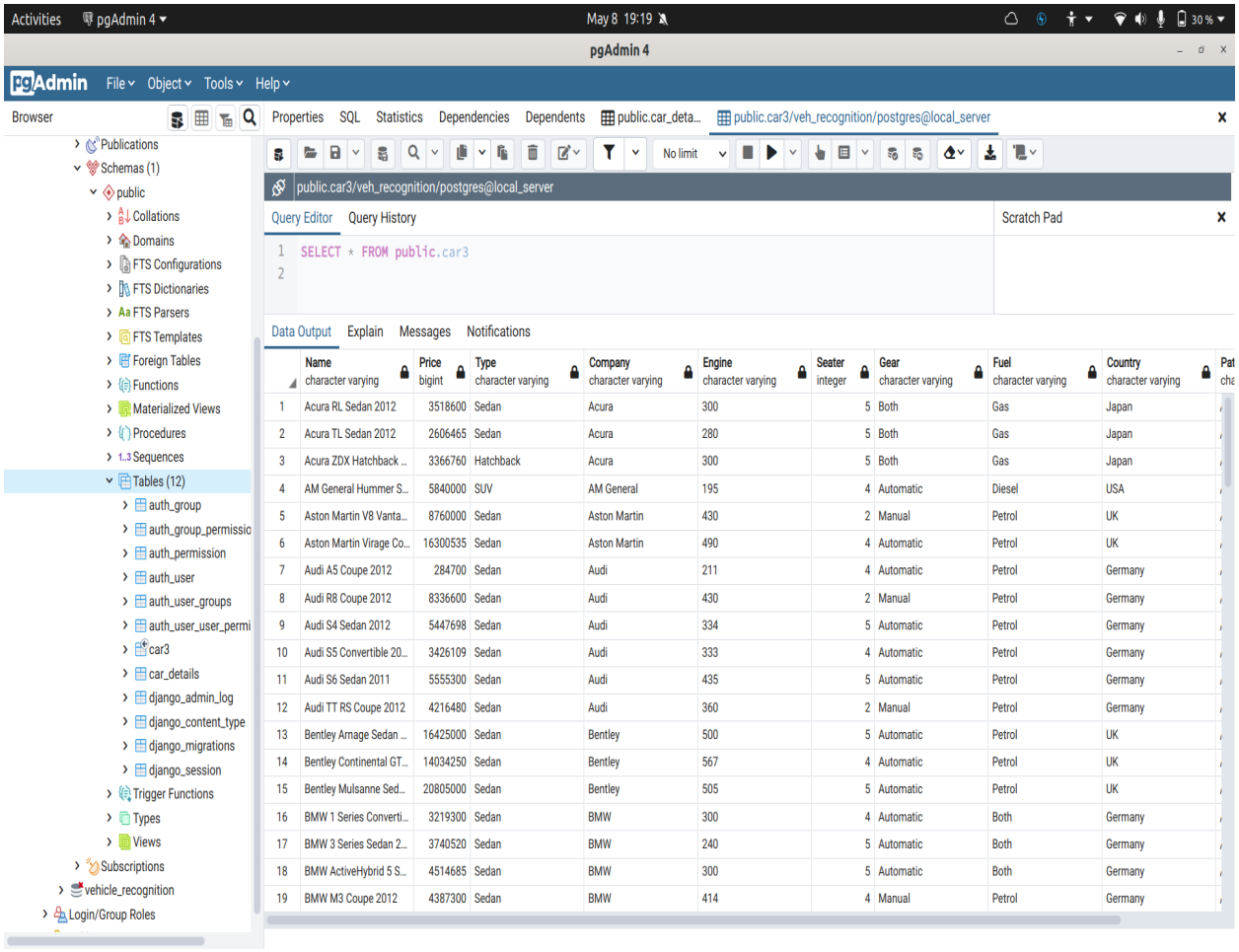



The testing phase pointed out the following errors:

```
36 Chevrolet Silverado 1500 Extended Cab 2012 != Chevrolet Silverado 1500 Classic Extended Cab 2007
72 Jaguar XK XKR  2012 != Jaguar XK XKR 2012
```

The error shows that the two names of the models are not matching. So, we made some changes accordingly and finalized our dataset file in the csv format.

Then after, the database is created in postgresql and to test the same we tried to display the content by running a simple sql query.
The results are as follows:

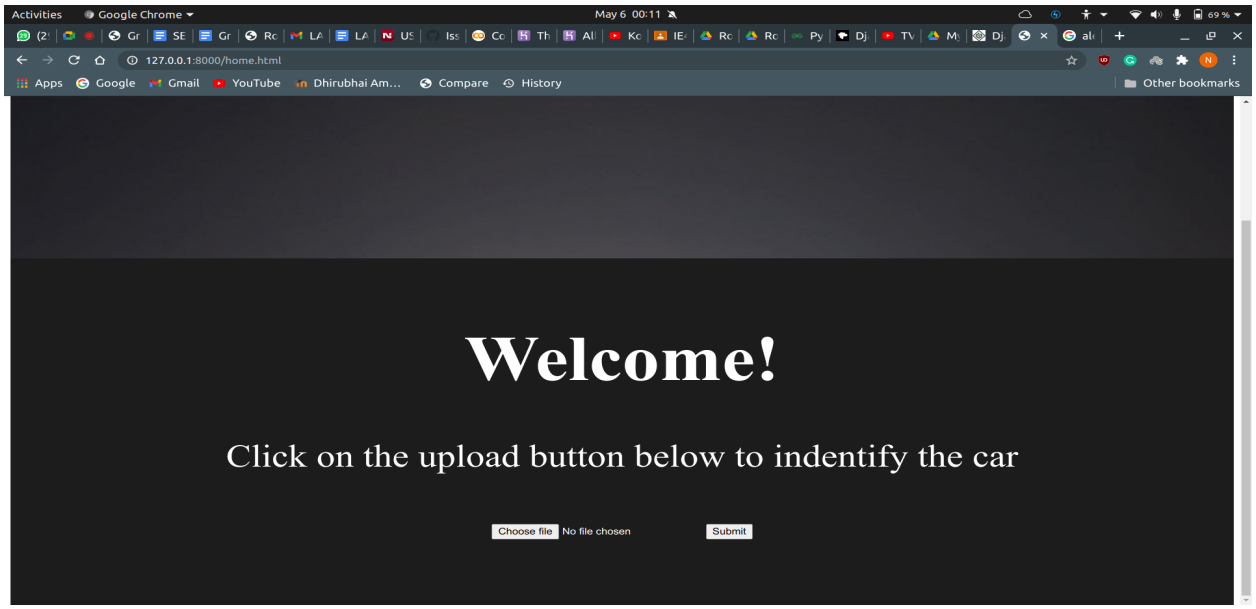



| | Name character varying | Price bigint | Type character varying | Company character varying | Engine character varying | Seater integer | Gear character varying | Fuel character varying | Country character varying | Pat cha |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Acura RL Sedan 2012 | 3518600 | Sedan | Acura | 300 | 5 | Both | Gas | Japan | |
| 2 | Acura TL Sedan 2012 | 2606465 | Sedan | Acura | 280 | 5 | Both | Gas | Japan | |
| 3 | Acura ZDX Hatchback ... | 3366760 | Hatchback | Acura | 300 | 5 | Both | Gas | Japan | |
| 4 | AM General Hummer S... | 5840000 | SUV | AM General | 195 | 4 | Automatic | Diesel | USA | |
| 5 | Aston Martin V8 Vanta... | 8760000 | Sedan | Aston Martin | 430 | 2 | Manual | Petrol | UK | |
| 6 | Aston Martin Virage Co... | 16300535 | Sedan | Aston Martin | 490 | 4 | Automatic | Petrol | UK | |
| 7 | Audi A5 Coupe 2012 | 284700 | Sedan | Audi | 211 | 4 | Automatic | Petrol | Germany | |
| 8 | Audi R8 Coupe 2012 | 8336600 | Sedan | Audi | 430 | 2 | Manual | Petrol | Germany | |
| 9 | Audi S4 Sedan 2012 | 5447698 | Sedan | Audi | 334 | 5 | Automatic | Petrol | Germany | |
| 10 | Audi S5 Convertible 20... | 3426109 | Sedan | Audi | 333 | 4 | Automatic | Petrol | Germany | |
| 11 | Audi S6 Sedan 2011 | 5555300 | Sedan | Audi | 435 | 5 | Automatic | Petrol | Germany | |
| 12 | Audi TT RS Coupe 2012 | 4216480 | Sedan | Audi | 360 | 2 | Manual | Petrol | Germany | |
| 13 | Bentley Arnage Sedan ... | 16425000 | Sedan | Bentley | 500 | 5 | Automatic | Petrol | UK | |
| 14 | Bentley Continental GT... | 14034250 | Sedan | Bentley | 567 | 4 | Automatic | Petrol | UK | |
| 15 | Bentley Mulsanne Sed... | 20805000 | Sedan | Bentley | 505 | 5 | Automatic | Petrol | UK | |
| 16 | BMW 1 Series Converti... | 3219300 | Sedan | BMW | 300 | 4 | Automatic | Both | Germany | |
| 17 | BMW 3 Series Sedan 2... | 3740520 | Sedan | BMW | 240 | 5 | Automatic | Both | Germany | |
| 18 | BMW ActiveHybrid 5 S... | 4514685 | Sedan | BMW | 300 | 5 | Automatic | Both | Germany | |
| 19 | BMW M3 Coupe 2012 | 4387300 | Sedan | BMW | 414 | 4 | Manual | Petrol | Germany | |

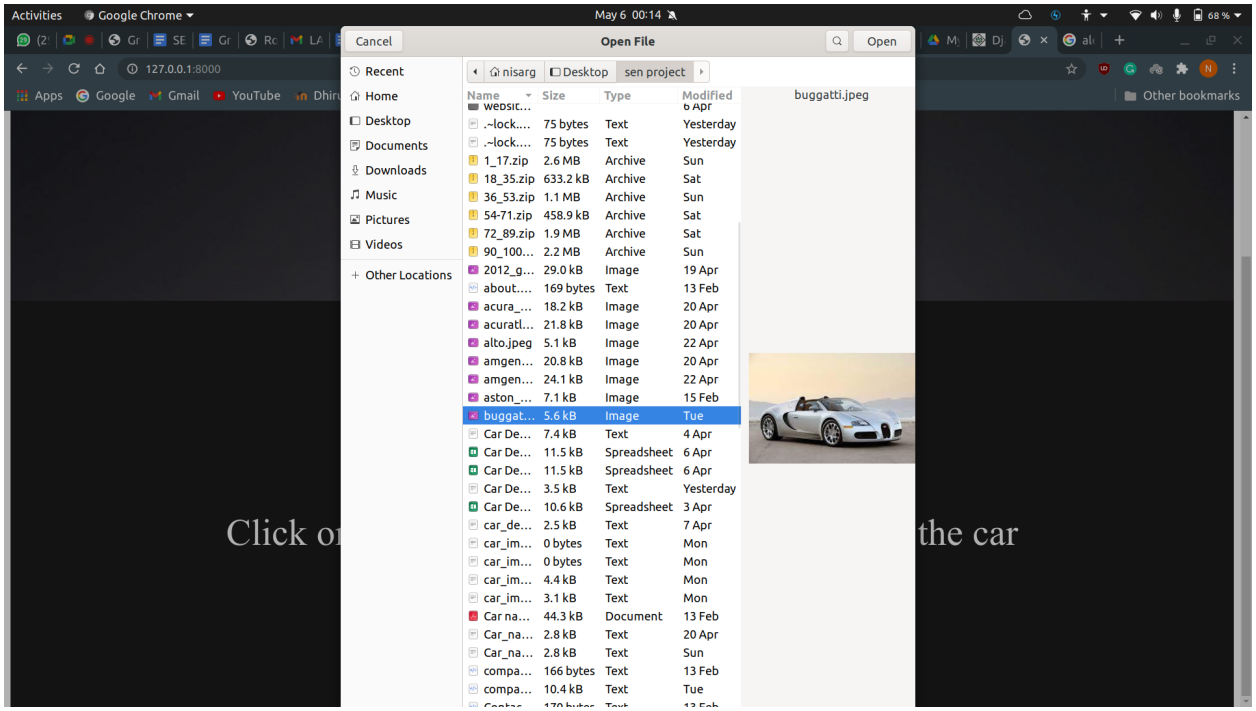
## C.) GUI Testing:

**Home Page**

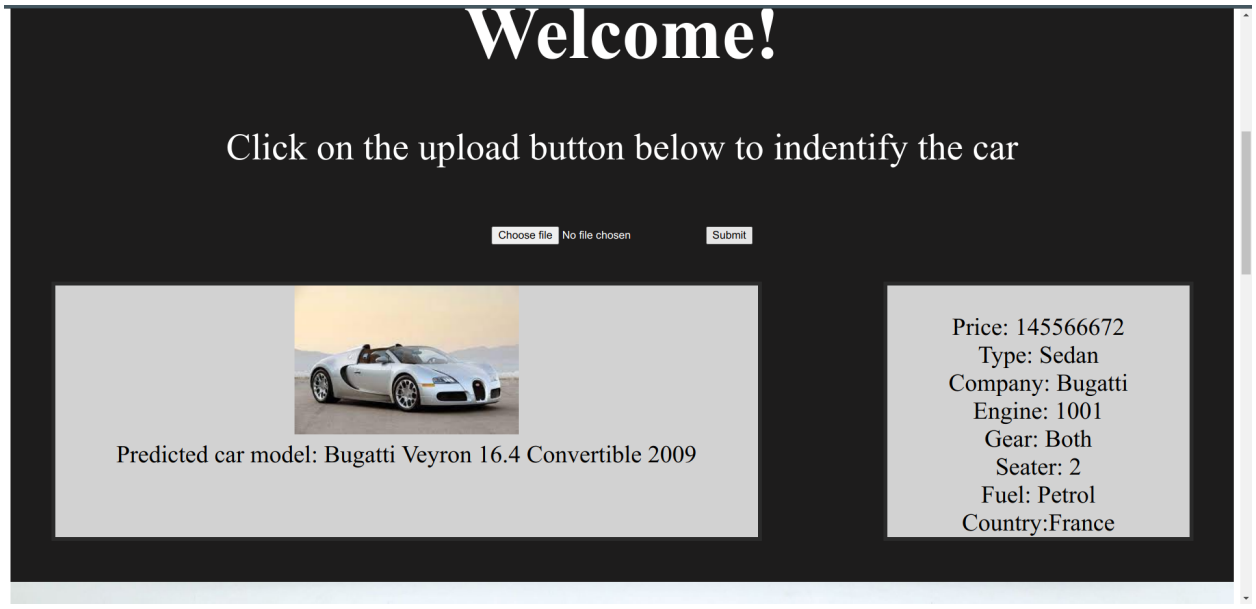

Below picture shows the choose file box where we can upload image that we want to test for recognition.



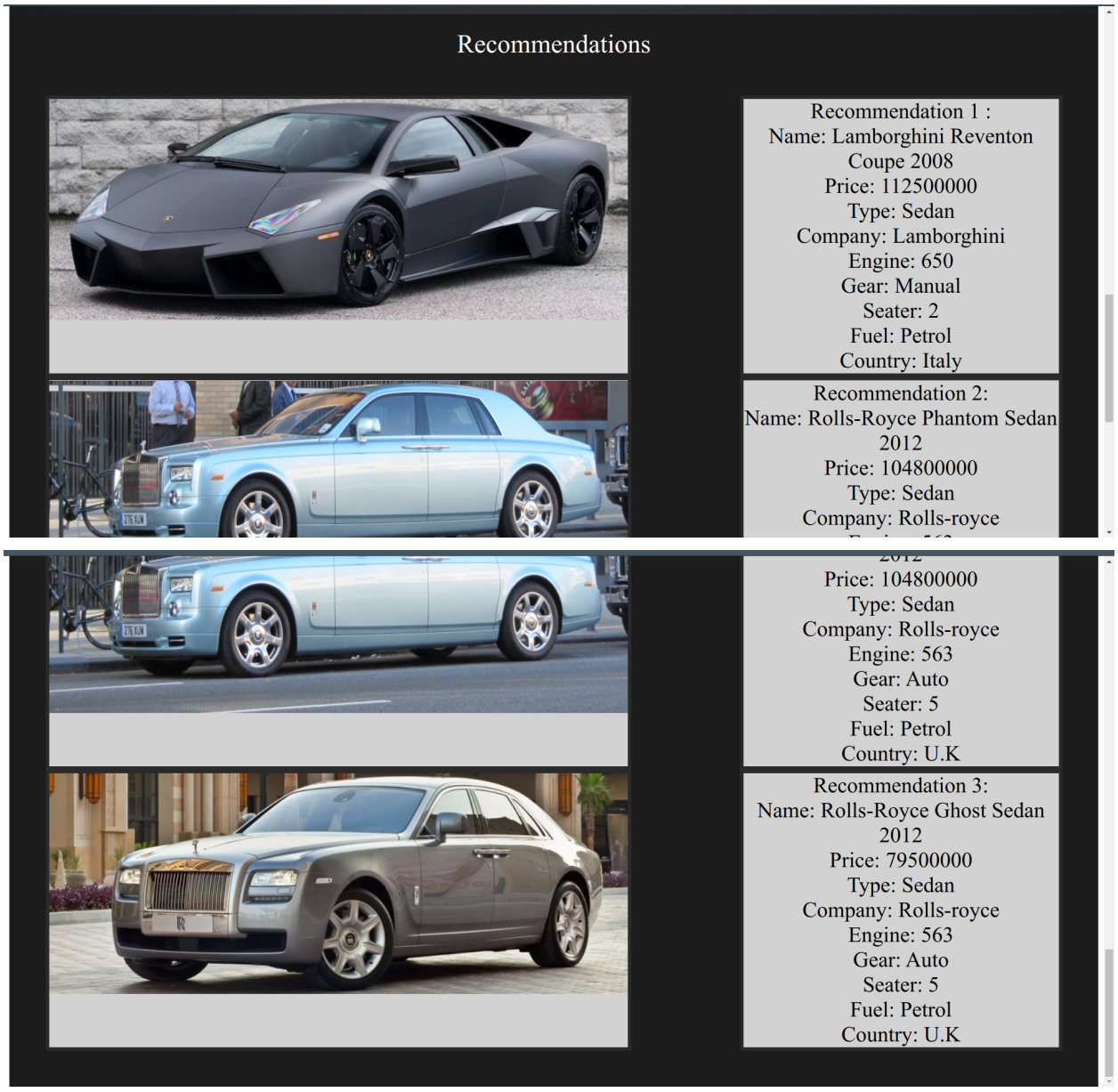
When a user clicks on the “choose file” button then the directory opens from where the user has to select an image. In the following picture car selected is Bugatti



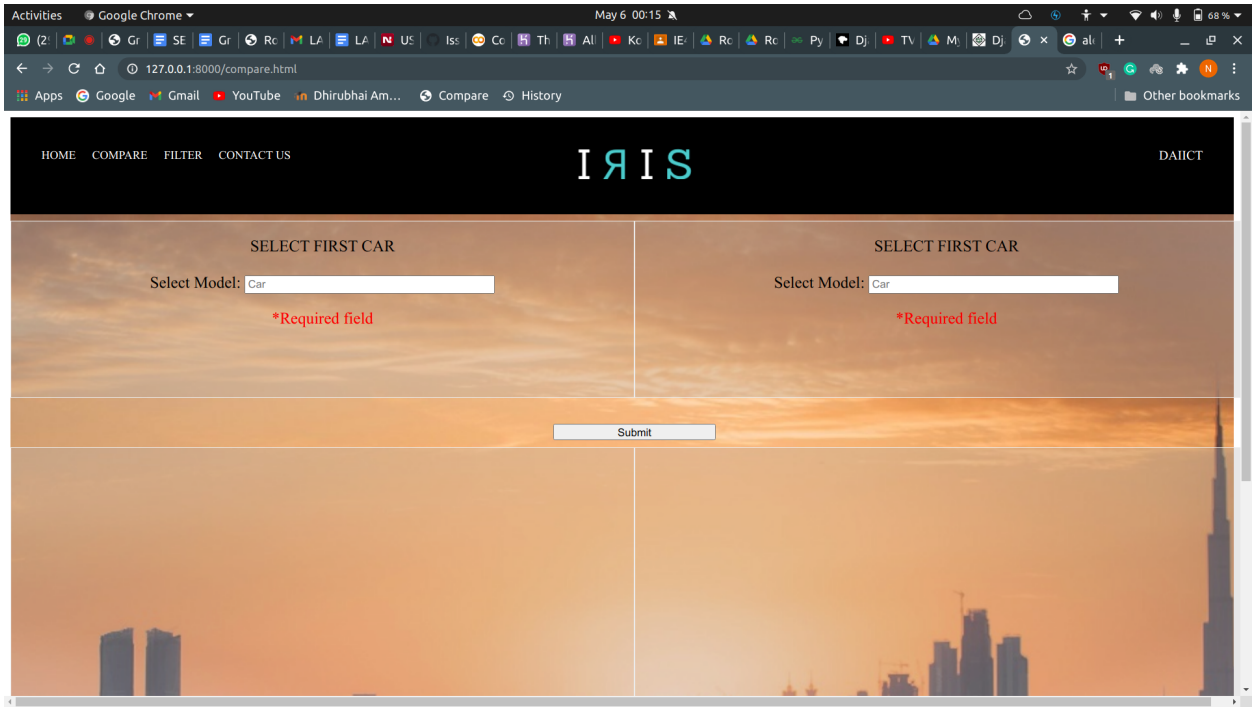

The following picture shows the output from the database giving the details of the car that was present in the image uploaded.



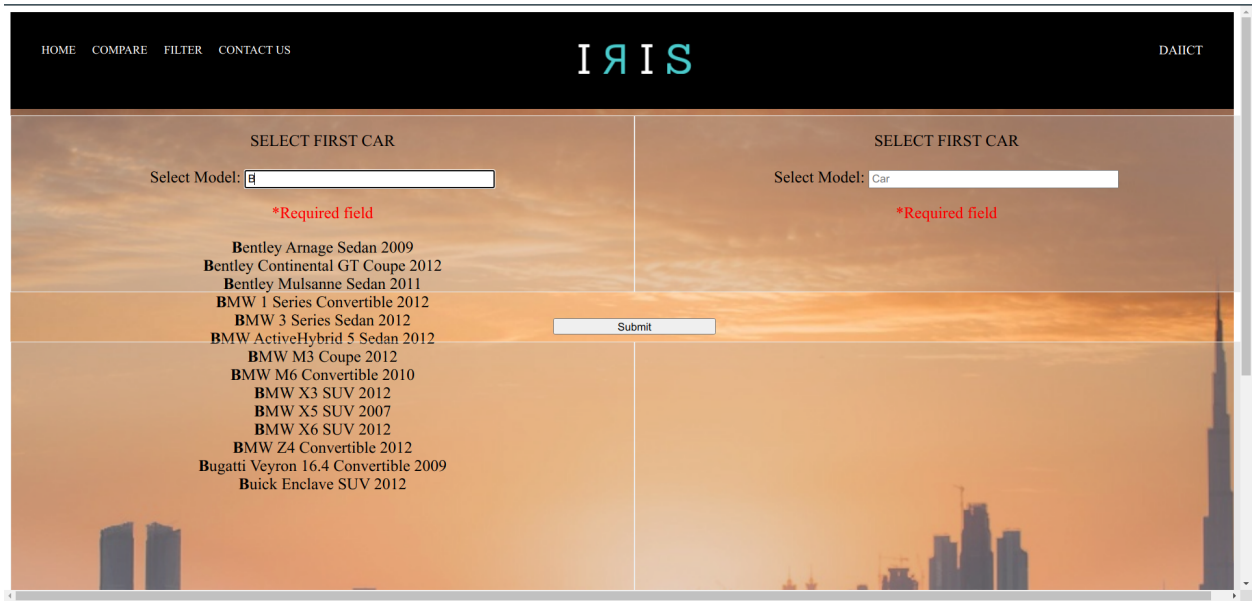
The following figure shows the recommendations of cars that were similar to the price of the car that was present in the image.

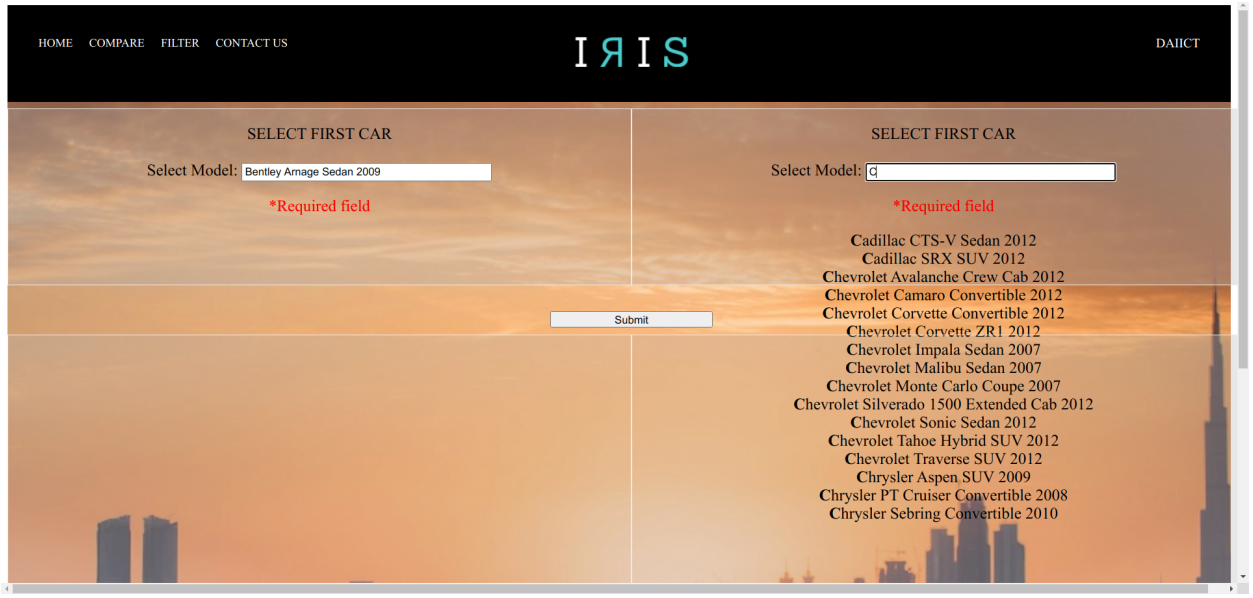


## Compare Page

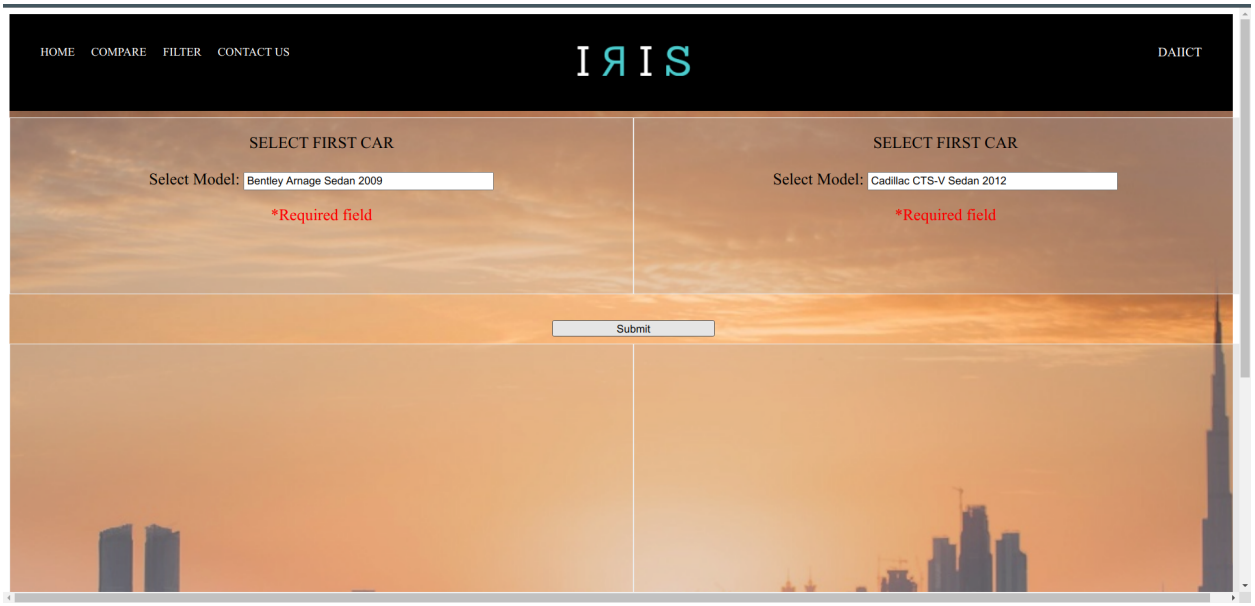



The following picture shows an autocomplete feature which will show options to select the car from starting with the letter that was inputted by the user.

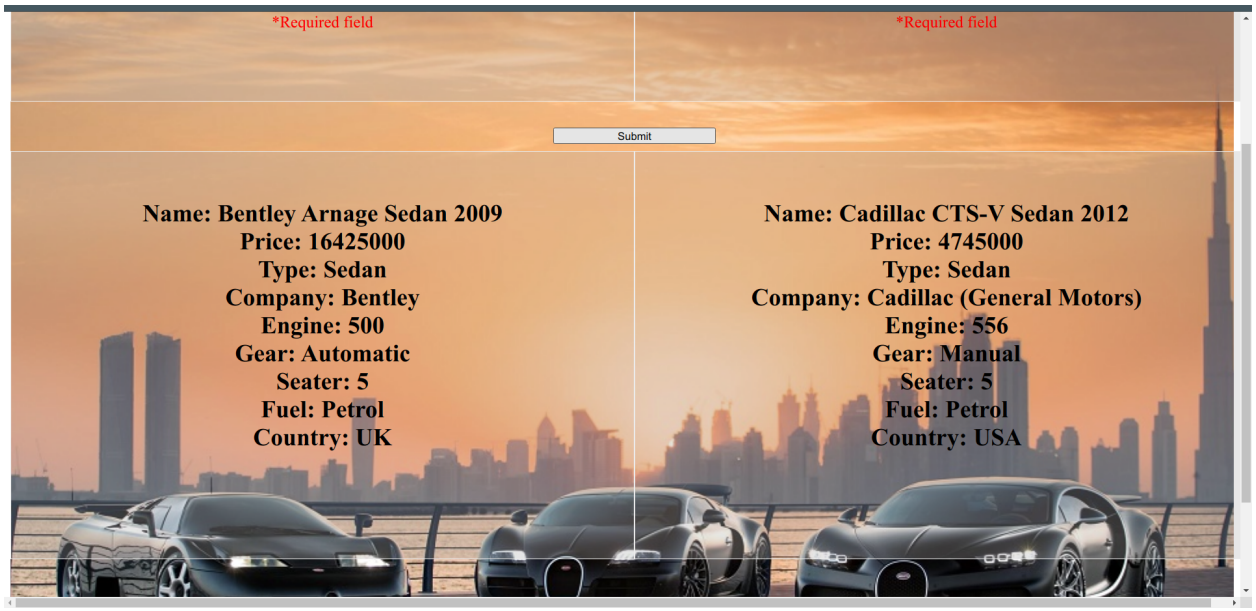




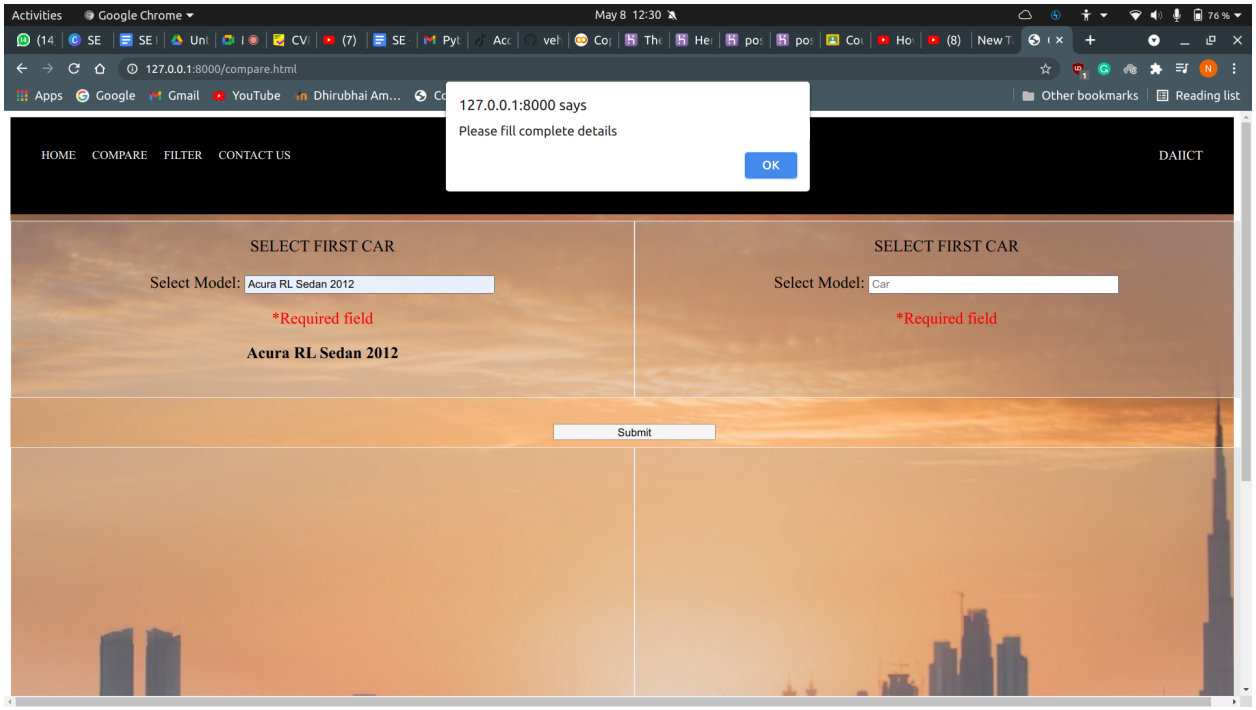

Both the cars are selected as shown in the following screen



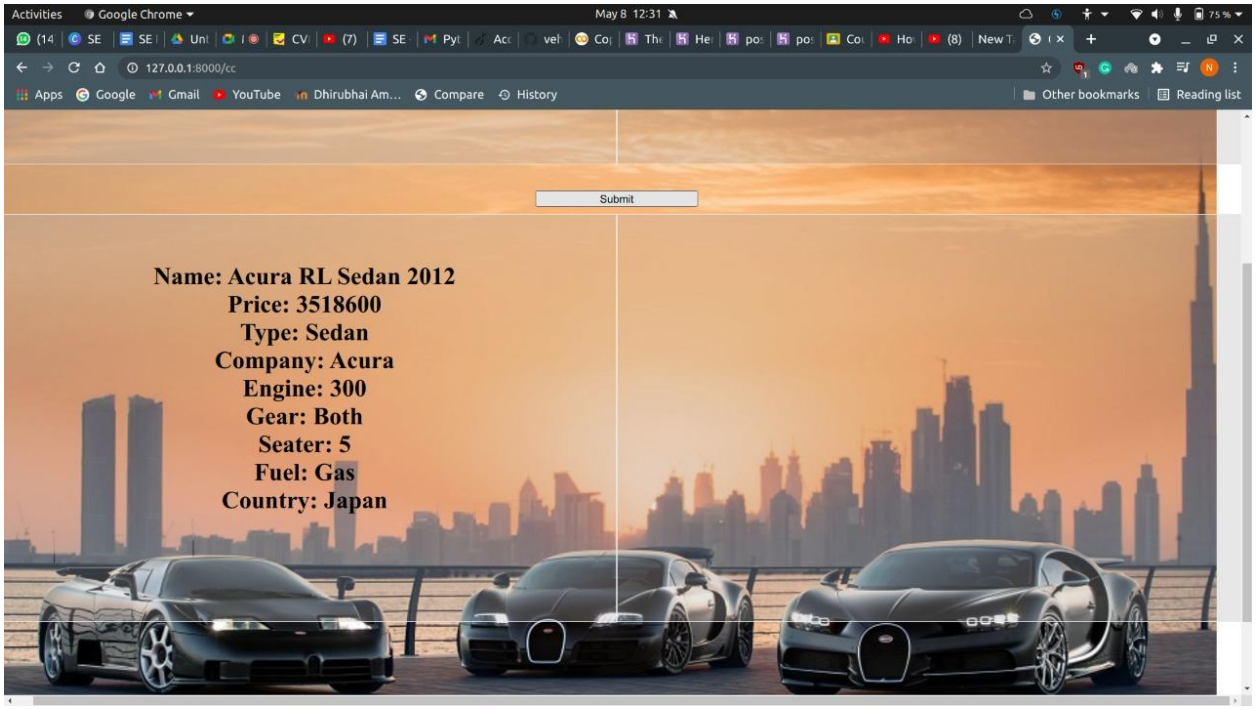
**The Following screen shows the data about the car that was selected by the user**




Shows alert when one or both  the fields are not selected

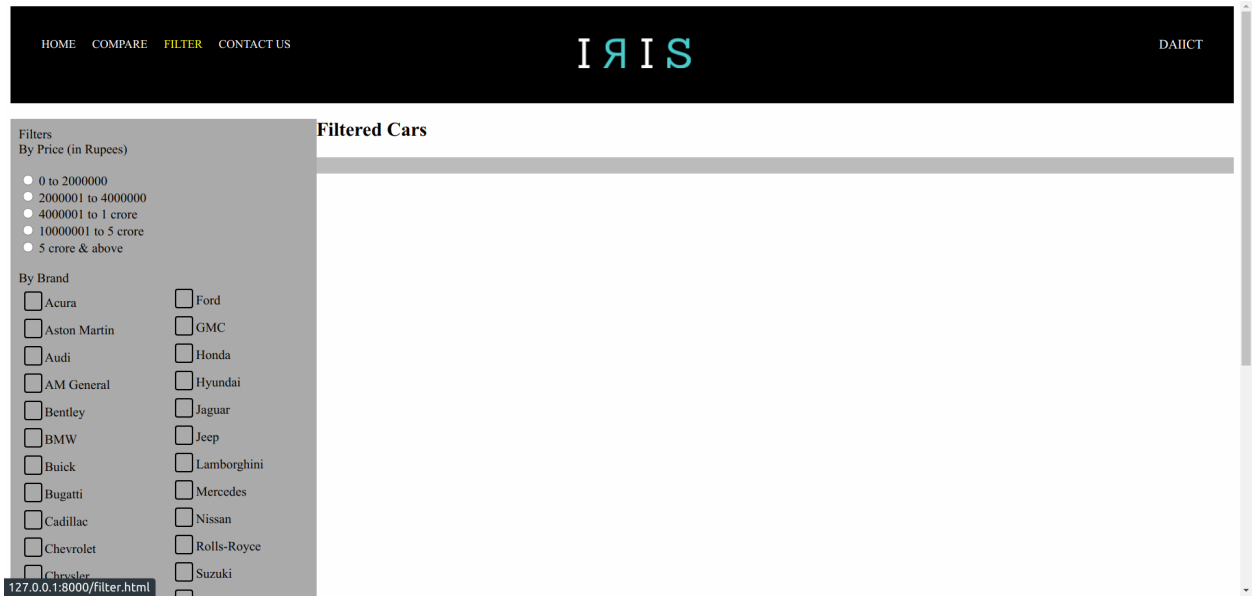



When the user clicks on Ok then details of the car that was selected will be displayed.



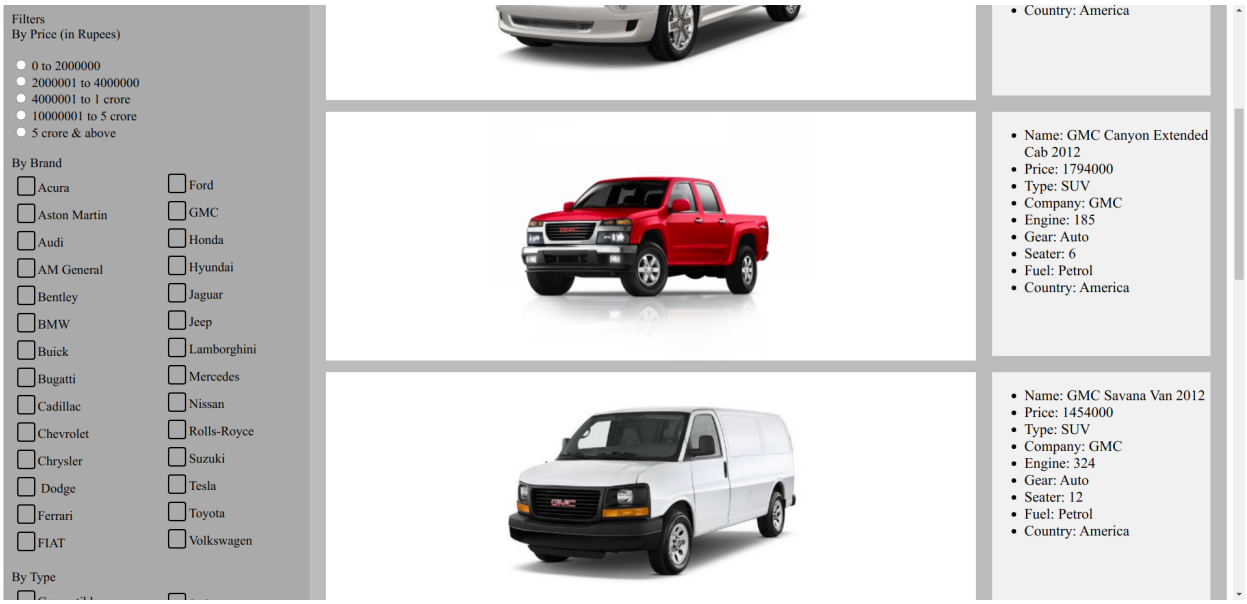

## Filter Page

Below is the image of filter page and here are some different requirements which the user has to select in order to get the specifications of the car he/she imagined



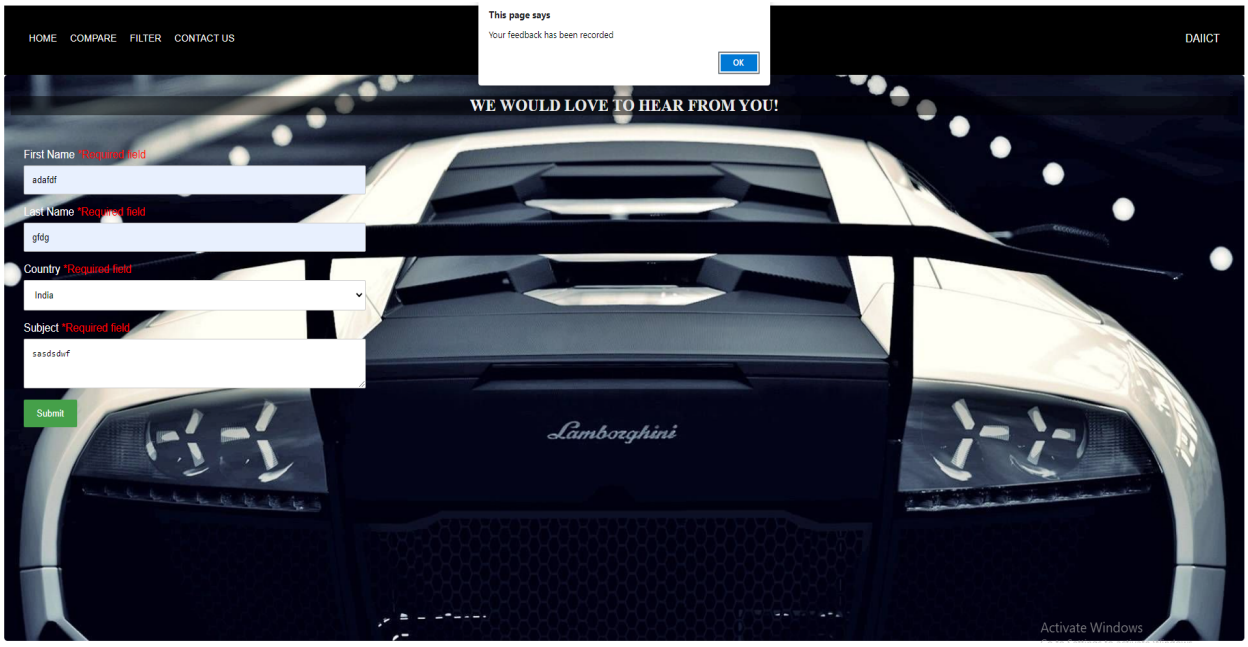

Output for the filters that were selected in above picture s shown in the following figure:

Below figure shows the sticky feature where the left panel can be seen on the screen even when the user scrolls down upto the end of the page.

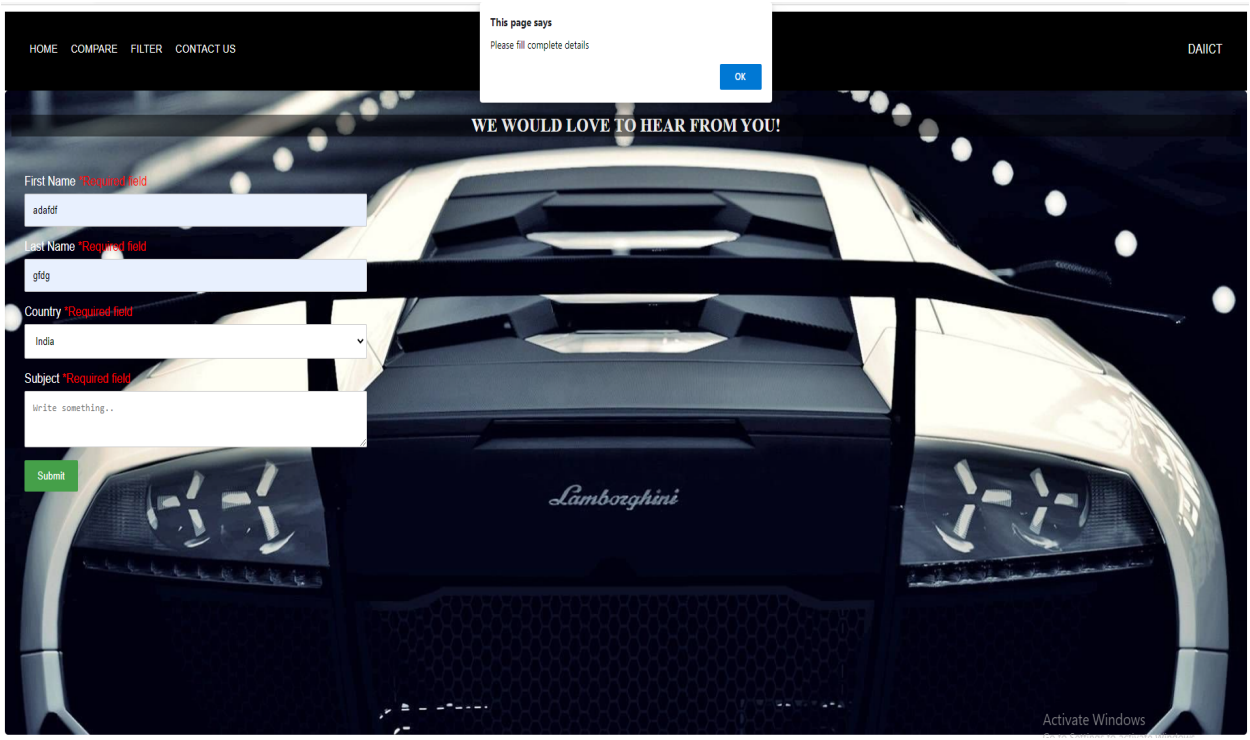



## Contact us

Shows a message when the feedback is successfully recorded:

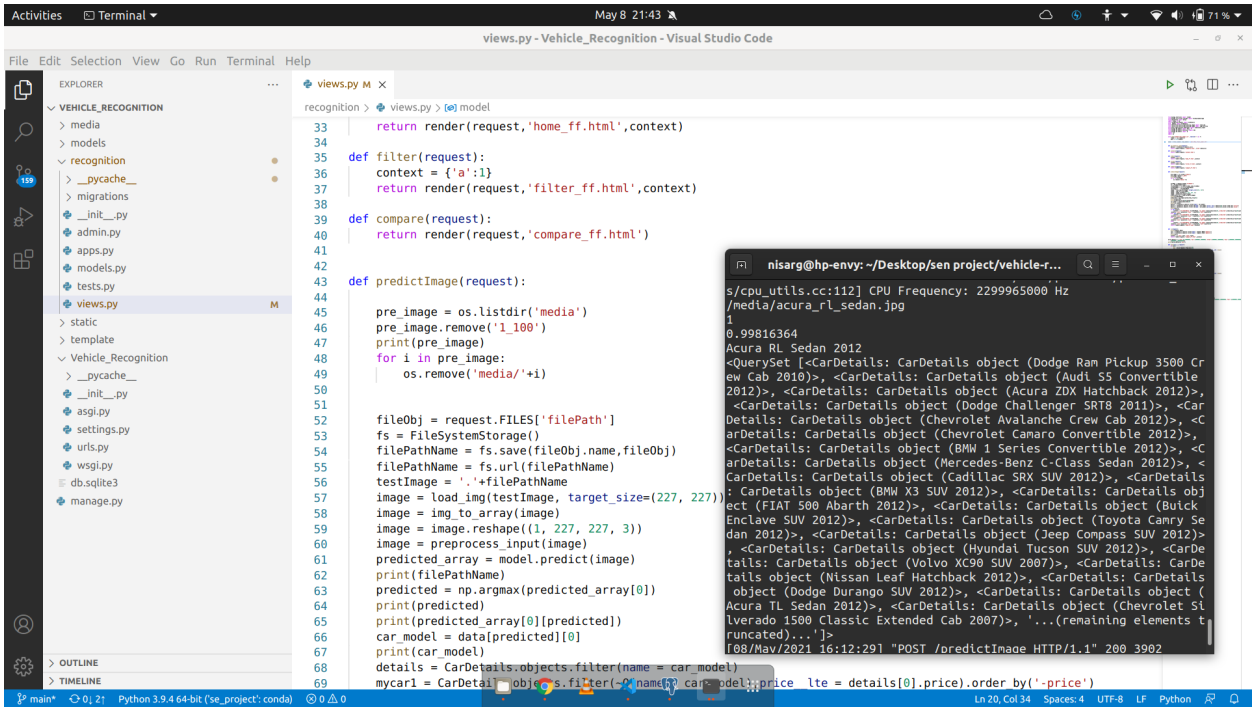



Shows alert message when one or more fields are not entered



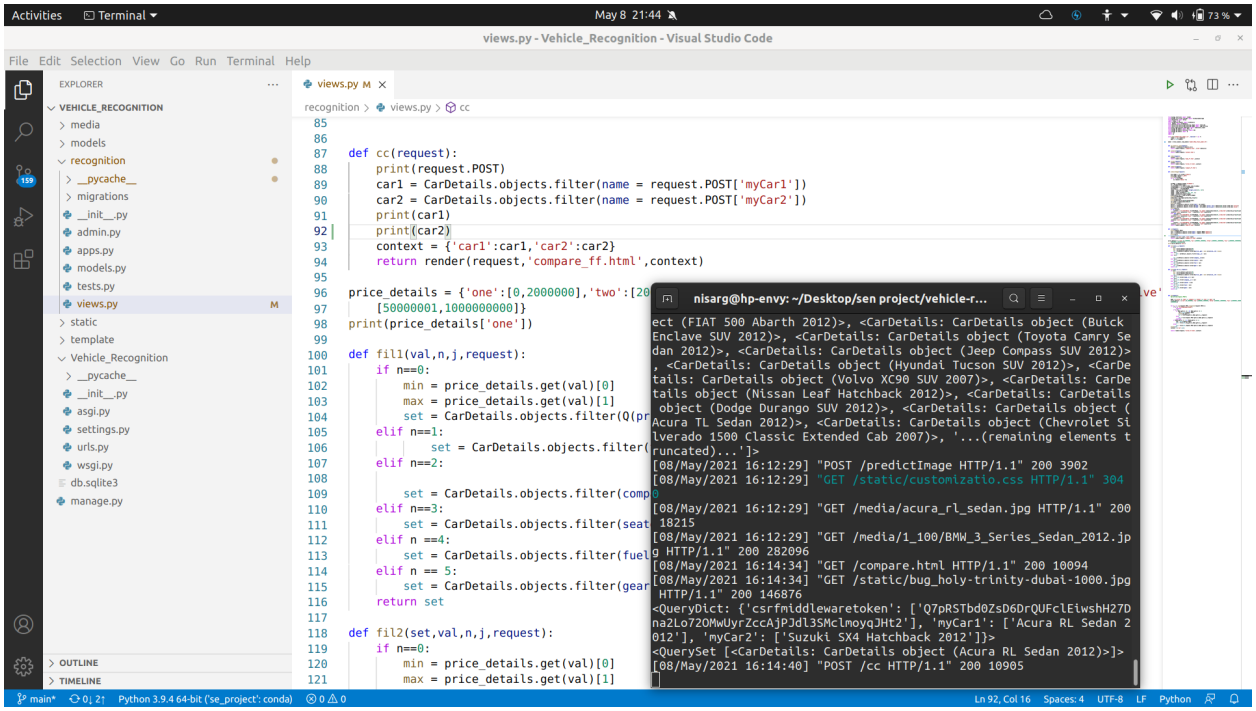

## D) Backend and Integration Testing

Home Page function Testing

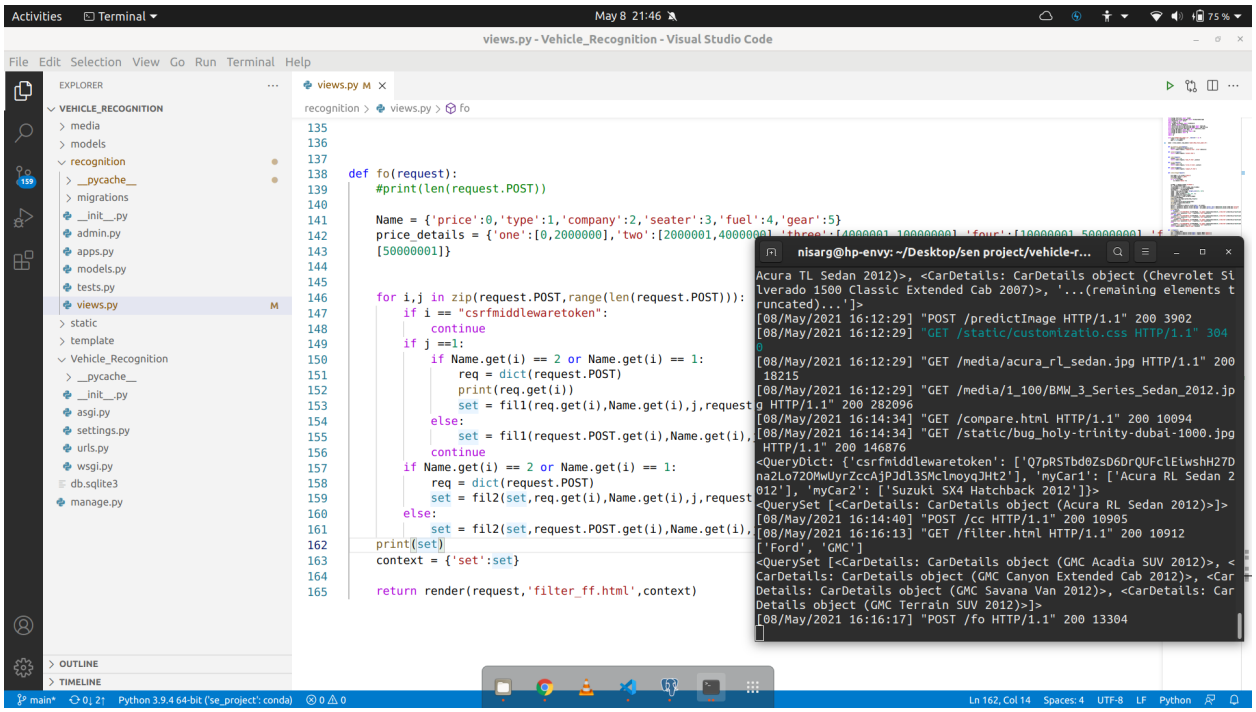



Code and its backend output.

## Compare Page function testing




## Filter Function testing



## 13. Planning for further development:

Our top most priority is Integration of backend and frontend and integration of backend with model. We also plan to increase Model's accuracy while adding more cars in the database. With the current dataset available, the scope of our project is limited. We also wished to add other features in our model like uploading photos of the license plate of the car and showing whether it's real or fake but since the data is confidential, the feature can't be included. Similarly, if the dataset of car owners is available, we can display car and owner information in real time which can be a lot of help to police and government agencies.

## 14. Open Issues:

1. Currently we are only doing car recognition and not for all vehicles. This is because of a lack of proper image dataset for other vehicles. We can expand our scope of this project if required datasets are available.
2. The current image dataset we are using is the best available dataset but consists of models that were manufactured upto 2012 which is old but we can modify our dataset once the working model is ready.
3. In the contact us page, we are not storing any data entered by the user. We could easily create a database for this when the system is actually deployed for public use.
4. As our model is giving 95% accuracy which is quite good but there is always a scope for improvement.
5. We provided few of the specifications of the models for comparison but more detailed information about the models can be included for better comparison.
6. The page works well when some filters are selected but when there are no filters selected, the car display section remains empty.

**15. Contributions: Everyone had fairly and equally contributed in documentation of mid evaluation, final evaluation, presentation, testing and video. Although the contribution was important, we are not mentioning specially for any member since all had worked hard.**

1. **Nisarg Patel (201801013)**

    - Backend development & Integration :
      We have used Django for backend and frontend integration. It provides a systematic framework for different modules and allows easy routing which helps locating errors. It also allows us to use modules for faster development. Configured various modules to integrate the pages, model, image database. Implemented functionalities of predicting image for home page,filter options of cars for filter page and contact us page.

    - Designing Schema & creating database :
      The database is created in Postgres and is then integrated with our project.

    - Image dataset integration :
      Backend for image dataset processing and its integration with the product where the file path of images were stored in postgres.

    - Website Hosting:
      Hosted the final version of the website on heroku with model and database on AWS.

2. **Khyati Bhuva (201801119):** As a group leader, I had responsibility to arrange meetings, decide agendas of meetings, be in constant contact and act as a point of contact with all teams and also note their problems and solve their problems, review all documents before submission etc. Most important responsibility was to create a productive and positive working environment with maintaining teamspirit and unity and to make sure that everyone is on the same page. As a member, I had contributed the following:

- Compare page:
  Frontend development of Compare Page was done , this page compares two cars so that a user can differentiate between cars and make a correct choice.

- Contact us page :
  Worked on the frontend development of CONTACT US page ,in this page the user can provide the feedback about the website.

- Image database :
  Typically we found out suitable image for the assigned cars and resized them accordingly

**3. Palak Verma (201801161)**

- Filter page :
  Frontend development of FILTER PAGE was done ,this page basically aims to find a suitable car for a user according to his/her requirements.This contains filters by “price”,”brand”,”type”, ”seater”, ”fuel”, ”gear”. When the user states the requirements we give the suitable cars and their specifications.

- Image database :
  Typically we found a suitable image for the assigned cars and resized them accordingly.

- Front end development :
  Navigation bar where four tabs to go on respective pages is shown along with college name and project title.

**4. Swapnil Dhola (201801163)**

- Model designing and development :
  We have done designing and development of models using transfer learning techniques. We have used mobilenet architecture and VGG architecture to develop image detection models for car detection.

- Data Augmentation and data preparation :
  Preparation of image dataset for training and testing. Since the data is not enough for testing and training I have done data augmentation to get more data for the image dataset.

- Evaluation :
  Final evaluation of model. Testing of How much accuracy model is giving and how the model is working for different classes. I have also done blackbox testing and static analysis for codes.

**5. Deep Malani (201801184)**

- Home Page :
  Frontend development of HOME PAGE was done, it basically consists of a “upload image” button which inputs the image of car and recognizes the car and gives the specifications of the same also gives the recommendations of cars which are related to the car uploaded .To add elegance to the page i have used the parallax scroll feature where the background content (i.e. an image) is moved at a different speed than the foreground content while scrolling.

- Image database :
  Typically we found out suitable images for the assigned cars and resized them accordingly.

- Logo:
  As we also wanted to incorporate the logo in our navigation bar, we had to resize while maintaining the aspect ratio. The image pixels also have to match with the padding of other elements and height of the bar.

**6. Siddharth Chauhan (201801196)**

- Model designing and development :
  Designing the model and developing the model using transfer learning techniques. Developed model using mobilenet and VGG architecture.

- Model Improvement :
  Improving the model using different hyperparameters and different learning methods. Using proper hyperparameters to tune the model and improve the accuracy of the model.

- Model validation :
  Validating the model with techniques like feature extraction. Whether or not the model we are using is learning.

- Image database:

Typically we found out suitable image for the assigned cars and resized them accordingly

7. **Ajay Vala (201801414)**

- Backend development & Integration:
  We have used Django for backend and frontend integration. Integration of the backend with the database for the COMPARE page functionality.
  I have tried to incorporate search as well as compare features into the same page.

- Data set refinement:
  Data aggregation and formatting car data in the manner desired by the integration team.

- Database validation:
  Checking and ensuring accuracy and the consistency of the database.

8. **Mohil Khimani (201801416)**

- Car database :
  Since the car dataset was not available, me and Vishal did the task of manually making the dataset by searching details for the individual cars like features, pricing, manufacturing details, etc and integrating it into a dataset. The dataset had to be formatted through various stages of changes according to the requirements of the backend team for the query running process.

- Image database :
  For the filter page, we required photos of the cars for showing results after filtering according to the user's needs. Thus, for that purpose, we collected relevant images of the cars and resized them according to requirements from the frontend team.

9. **Shabbir Murtaza (201801428)**

- Recommendation Functionality:
  Worked in the integration part of frontend with the database for creating recommendations giving the closest possible car to the predicted car. The prediction is based on price as we thought this was the best parameter for this. I implemented the algorithm using django query sets.

- Routing the frontend pages to the expected functionality.

- Final integration: Connected the static frontend pages using jinja to the backend python code. The frontend pages can then display the results returned from python code making it dynamic.

**10. Vishal Patel(201801436)**

- Since the car dataset was not available, me and Mohil manually searched for individual cars and collected the details like Price, Engine Power, Type, Number of seats, Fuel type for the required dataset in the product. The datasets we created had gone through various stages of formatting and corrections according to the requirements of the backend team.

- Firestore Database:
  I have created a firestore database in the firebase which stores the database in the json tree format and tried to run a few queries for the same. The database I created was a NoSql database but the backend team found some difficulties in integrating it with our product so the backend team has created the Sql database.

- Image Database:
  We found some suitable images and resized them according to required specifications for the models included in our product for the image database.

---