**Car Dealer Database and Web Application Project**

**1mUHA**

**201531014 - Muhammed Safa Demirel**

1. Abstract

With this application, car dealers will have the chance to show their cars to everyone on the internet, so they can reach more potentials buyers.

1. Table of contents

i. Abstract 1

ii. Table of contest 1

iii.List of figures 2

iv. List of tables 2

1. Introduction 3

2. Used Technologies 3

3. Requirements 3

3.a. Data Requirements 3

3.b. Functional Requirements 4

4. Database Modeling Steps 5

4.a. Er Diagram & Relational Schema 5

4.b. Normalization 5

5. Database Implementation Steps 6

6. Queries from the Database 10

7. Application Modeling Steps 12

7.a. Use Case Diagram 12

8. Application Implementation 13

8.a. Connection code and example code 13

8.b. Application User Interfaces 15

9. References 18

1. List of Figures

Figure 1. ER Diagram & Relational Schema 1

Figure 2. Brand table 1

Figure 3. Type table 1

Figure 4. Model table 1

Figure 5. Color table 1

Figure 6. Fuel table 1

Figure 7. Gear table 1

Figure 8. Image table 1

Figure 9. Comment table 1

Figure 10. Maintenance table 1

Figure 11. Option table 1

Figure 12. Car\_option table 1

Figure 13. Car table 1

Figure 14. Selecting first 10 cars with all details 1

Figure 15. Selecting maintenance details of the cars 1

Figure 16. Selecting selected options on the cars 1

Figure 17. Selecting color,gear,fuel names on the cars 1

Figure 18. Selecting model,brand, type names on the cars 1

Figure 19. Selecting comment of the cars 1

Figure 20. Use case diagram 1

Figure 21. Connection to database 1

Figure 22. Listing cars on the first page 1

Figure 23. Index page 1

Figure 24. Detail page 1

Figure 25. Home of the admin section 1

Figure 26. Car model page on the admin section 1

Figure 27. Option page on the admin section 1

1. List of Tables

Table 1. Data requirements 1

Table 2. Functional requirements 1

Table 3. Normalization forms 1

1. Introduction

I wanted to make a site where the car dealers can sell the cars on the internet. They could reach more people with internet. User interface is pretty simple for all users. Also with this web site car dealers doesnt have to pay high fees to other car sales sites.

1. Used Technologies

I used MySQL on the database side and PHP on the server side. I made database connections with PDO class on php. PDO is more secure than mysql\_query. Yet there are difficulties with PDO’s usage. PDO’s syntax is a little bit complicated therefore i used a builder class called Pdox. Pdox is a fast, efficient and useful Query Builder and PDO Class. Check out the Figure.20 to see usage of Pdox.

1. Requirements
2. Data Requirements

|  |  |
| --- | --- |
| Table Name | Data Requirements |
| brand | id, name |
| color | id, name |
| fuel | id, name |
| gear | id, name |
| option | id, type, name |
| car\_option | id, car\_id, option\_id |
| type | id, name |
| model | id, brand\_id, name, type\_id |
| car | id, title,description, cover\_image\_url, model\_id, color\_id, gear\_id, fuel\_id, year, mileage, seat, engine\_power, number\_owner, price, status |
| comment | id, car\_id, message, name, date |
| maintenance | id, car\_id, last\_time, description |
| image | id, car\_id, url |

Table 1 – Data Requirements

1. Functional Requirements

|  |  |
| --- | --- |
| Table Name | Functional Requirements |
| brand | All brands must have a name and must be unique. |
| color | All colors must have a name and must be unique. |
| fuel | All fuels must have a name and must be unique. |
| gear | All gears must have a name and must be unique. |
| option | All options must have a name and a type and must be unique on that type and name combination. |
| car\_option | All car options must have a car id and a option id and must be unique on that car id and option id combination. |
| type | All types must have a name and must be unique. |
| model | All models must have a brand id, a name and a type id and must be unique on that brand id, name, type id combination. |
| car | All cars must have a title, a description, a cover image url, a url, a model id, a color id, a gear id, a fuel id, a year, a mileage, a seat, a engine power, a number of owners, a price. |
| comment | All comments must have a car id, a message, a name and a date. |
| maintenance | All maintenances must have a car id, a last time and a description. |
| image | All images must have a car\_id and a url and must be unique on that car id and url. |

Table 2 – Functional Requirements

1. Database Modeling Steps
2. Er Diagram & Relational Schema

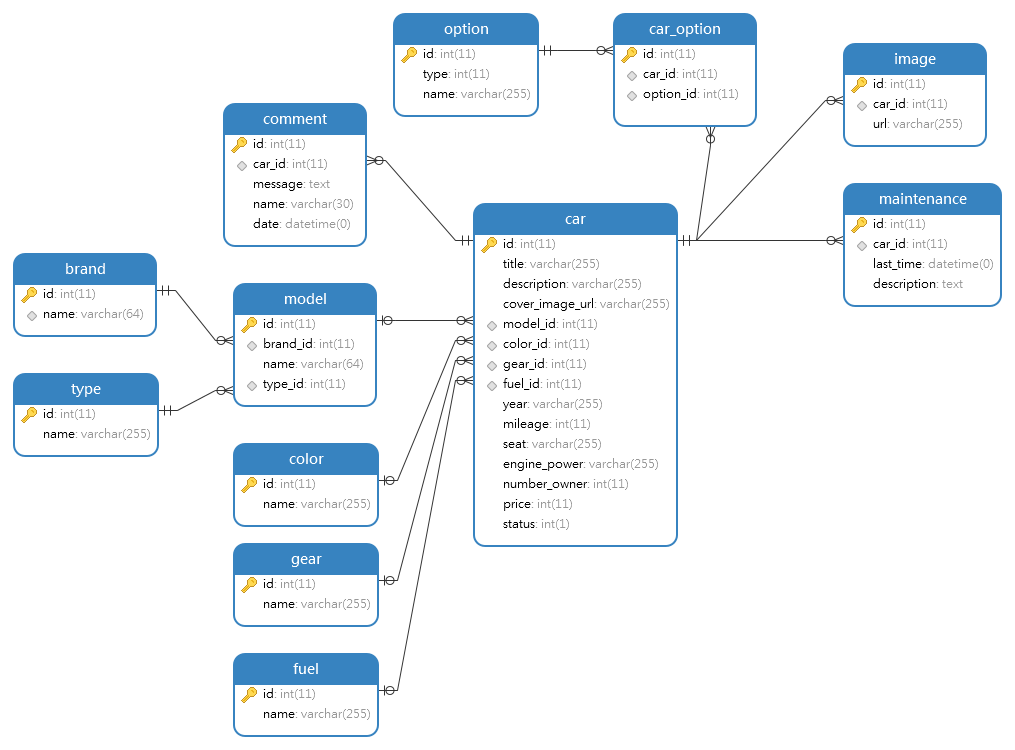


Figure 1 – ER Diagram & Relational Schema

1. Normalization

|  |  |
| --- | --- |
| Table Name | Normalization Form |
| brand | 2NF |
| color | 2NF |
| fuel | 2NF |
| gear | 2NF |
| option | 2NF |
| car\_option | 2NF |
| type | 2NF |
| model | 2NF |
| car | No normalization. |
| comment | No normalization. |
| maintenance | No normalization. |
| image | No normalization. |

Table 3 – Normalization forms

1. Database Implementation Steps

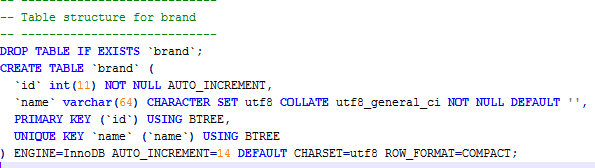


Figure 2 - Brand table

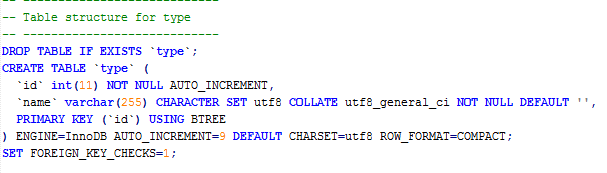


Figure 3 - Type table

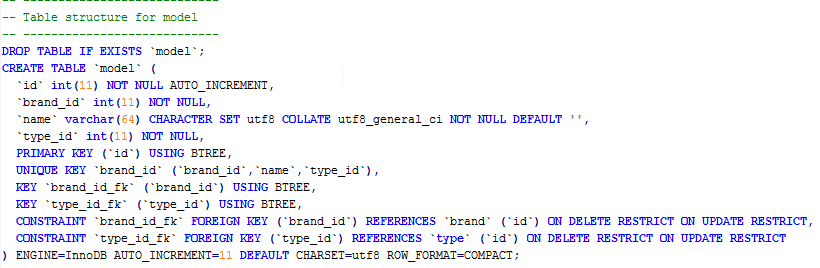


Figure 4 - Model table

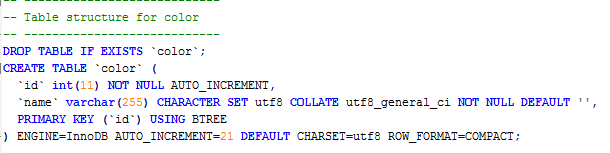


Figure 5 - Color table

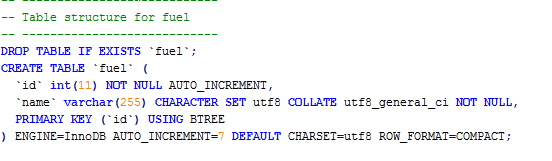


Figure 6 - Fuel table

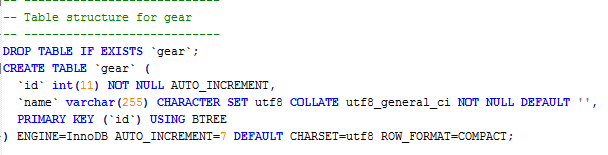
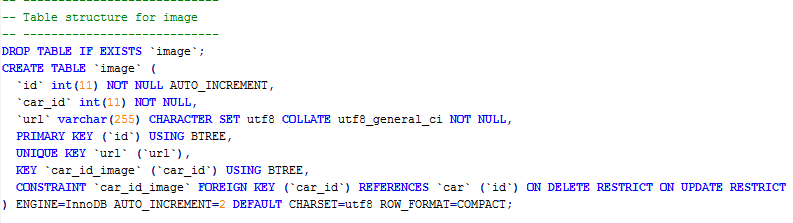


Figure 7 - gear table



[Figure 8 - Image table](http://localhost/)

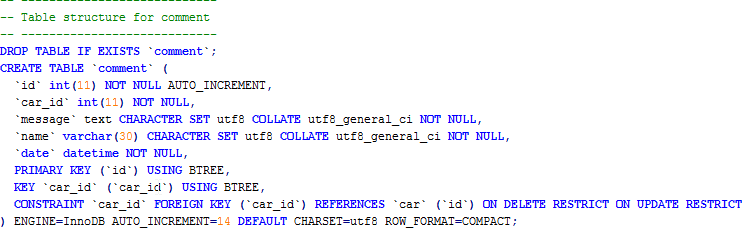


Figure 9 - Comment table

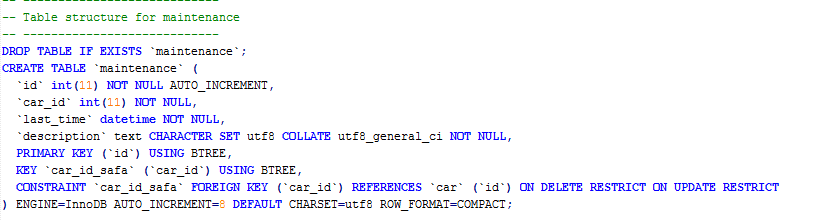


Figure 10 - Maintenance table

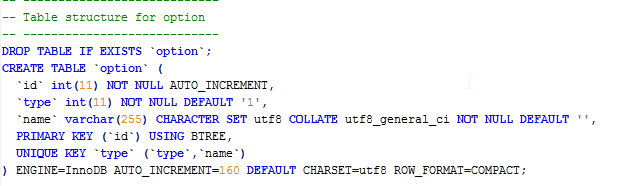


Figure 11 - Option table

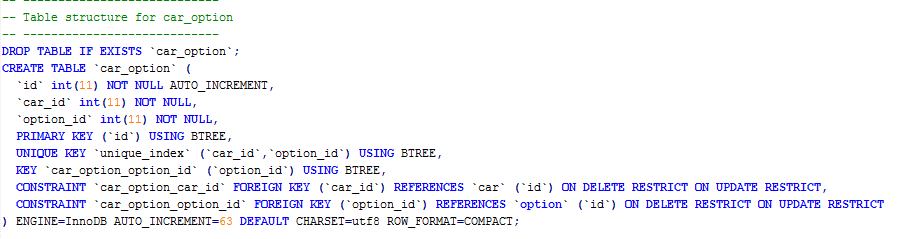


Figure 12 - Car\_option table

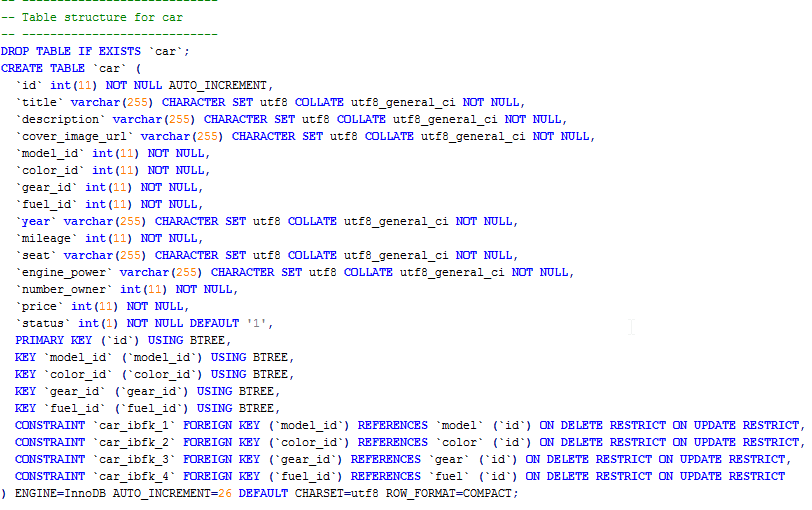


Figure 13 - Car table

1. Queries from the Database

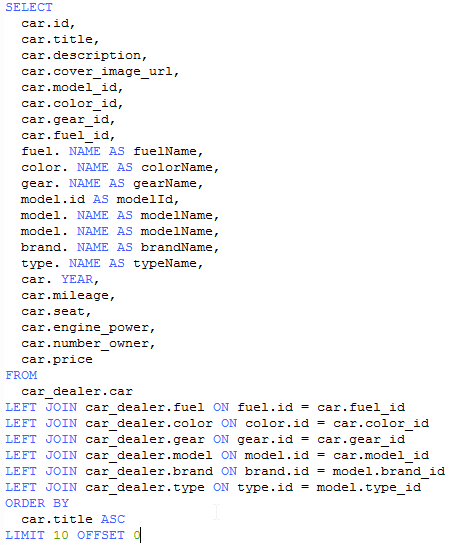


Figure 14 – Selecting first 10 cars with all details

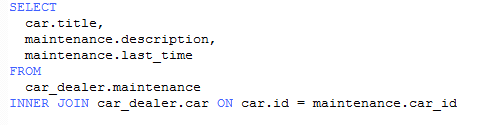


Figure 15 - Selecting maintenance details of the cars

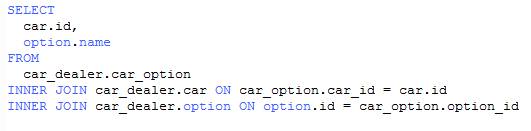


Figure 16 - Selecting selected options on the cars.

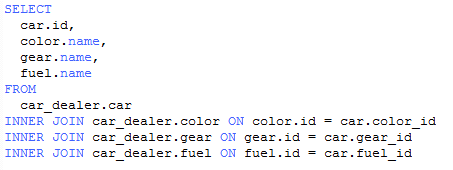


Figure 17 - Selecting color, gear, fuel names on the cars.

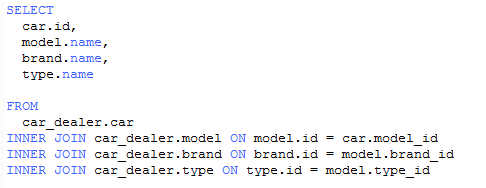


Figure 18 - Selecting model,brand,type names on the cars.

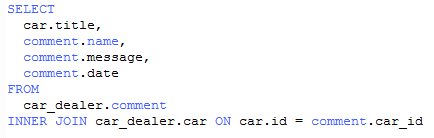


Figure 19 - Selecting comments of the cars.

1. Application Modeling Steps
   1. Use Case Diagram

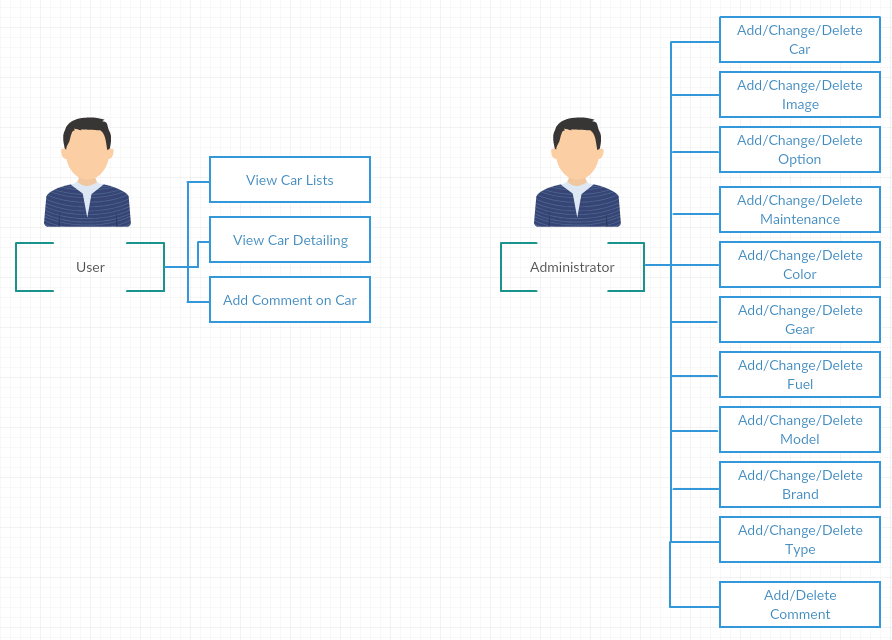


Figure 20 – Use Case Diagram

1. Application Implementation
   1. Connection code and example codes

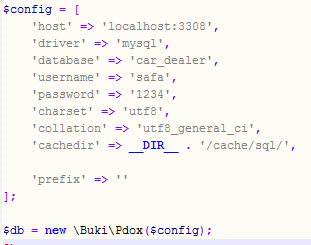


Figure 21 – Connection to database.

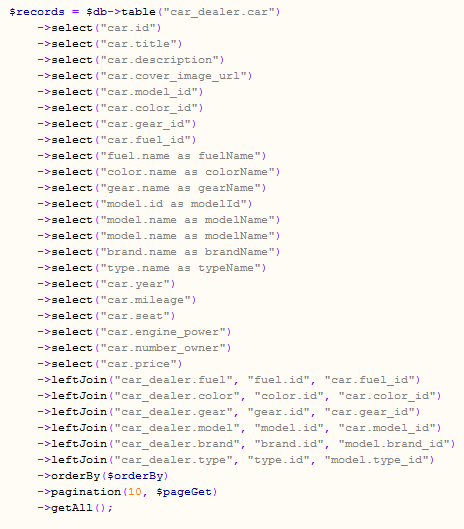


Figure 22 – Listing the cars on the first page.

1. Application User Interfaces

.

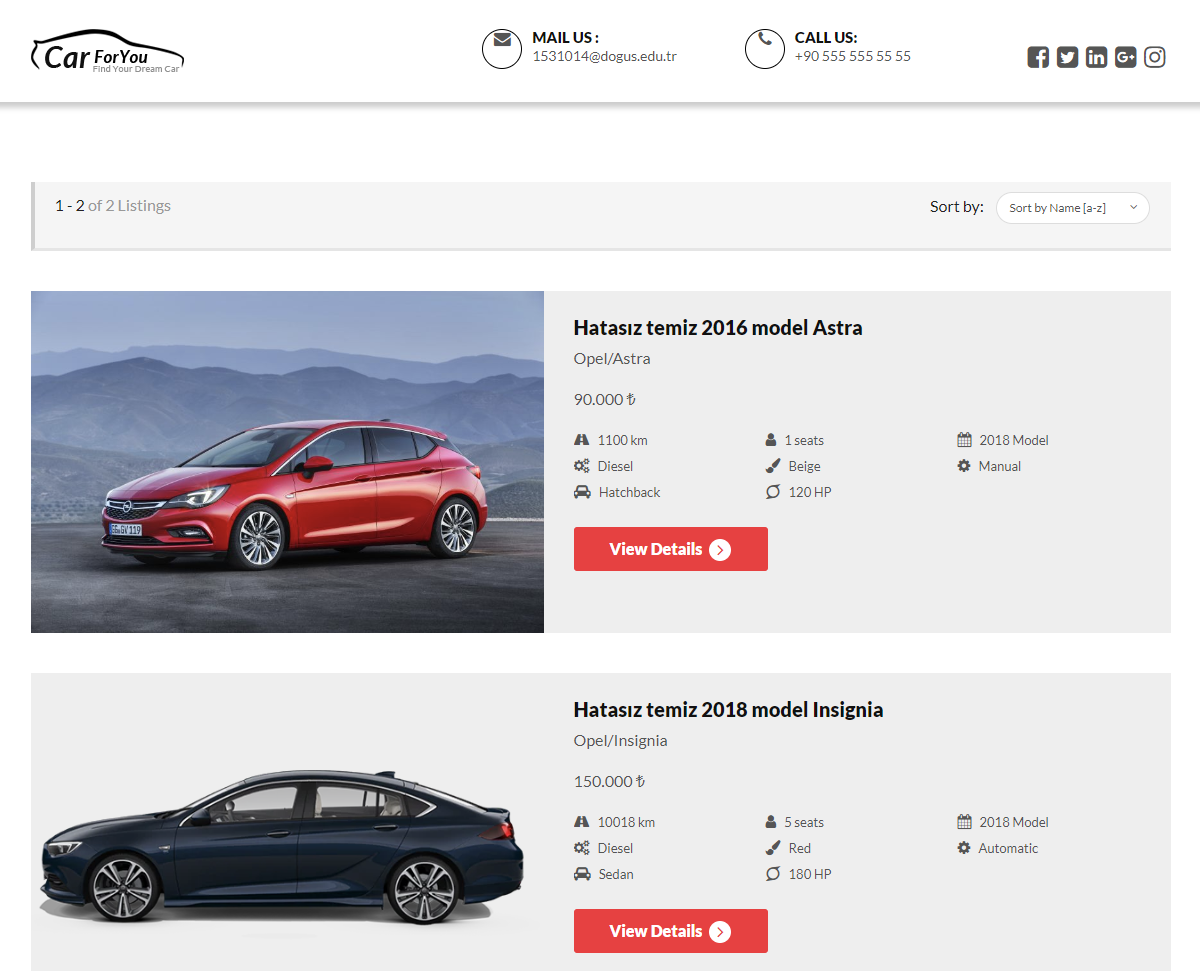


Figure 23 – Index page. Listing all cars.

In this page user can view all cars that listed. They can see some of the basic informations of the cars. They can sort the list by name, price, mileage.

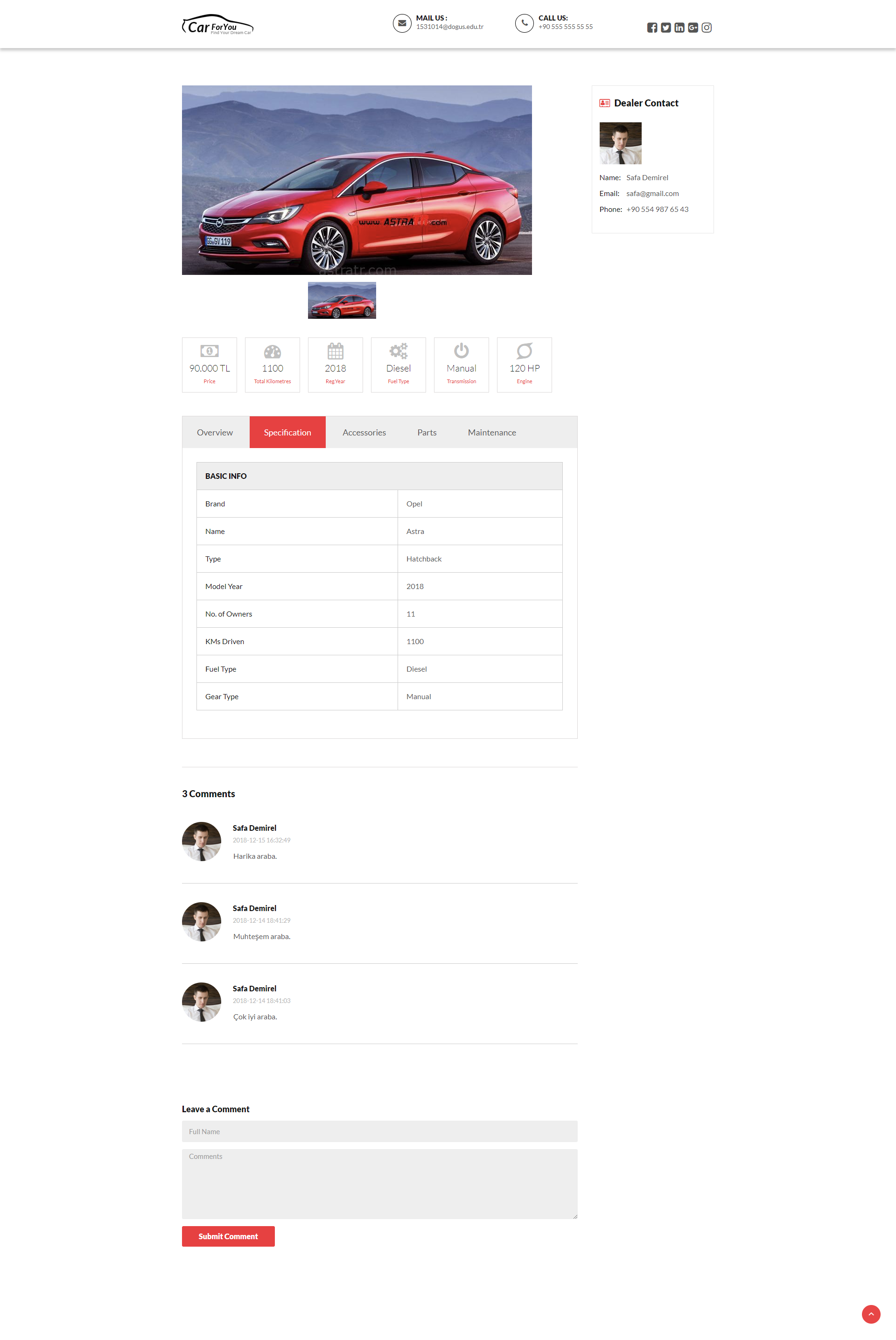


Figure 24 – Detail page. User can see detals, options and comments of the car.

In this page user can see all of the details of the car. User can see which options this car have, which parts of this car painted or replaced. User can see comments of the other people about this car. User can see the maintenance logs of the car.

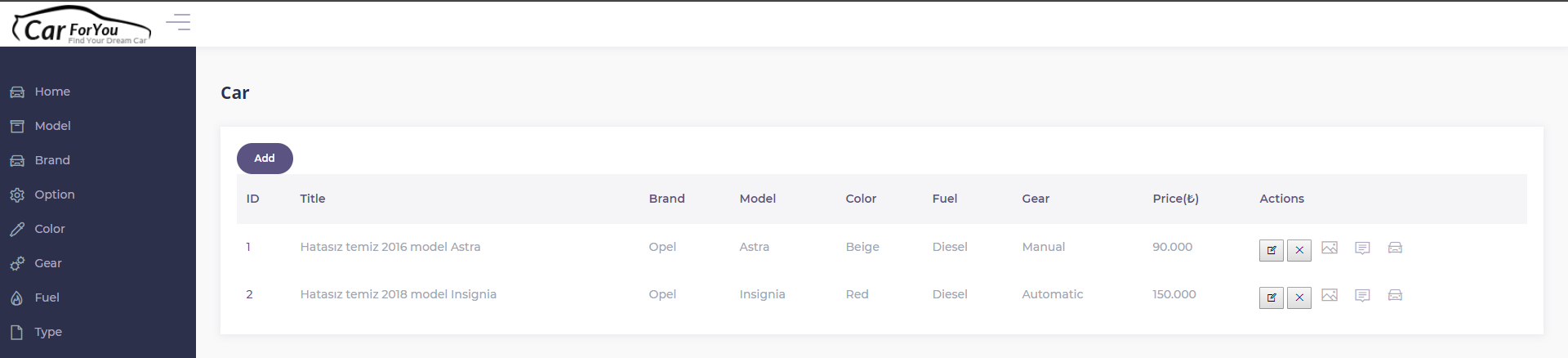


Figure 25 –Home of the admin page.

In the admin page we have Home, Model, Brand, Option, Color, Gear, Fuel, Type sections.

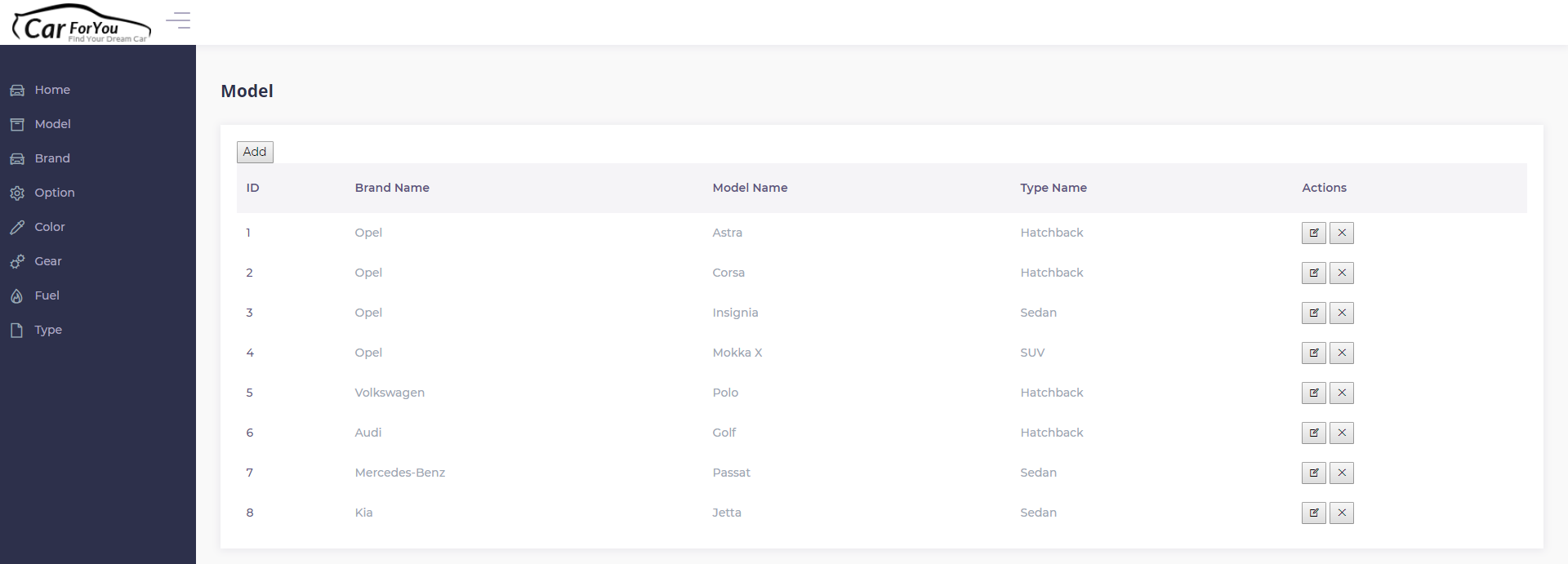


Figure 26 – Car model page.

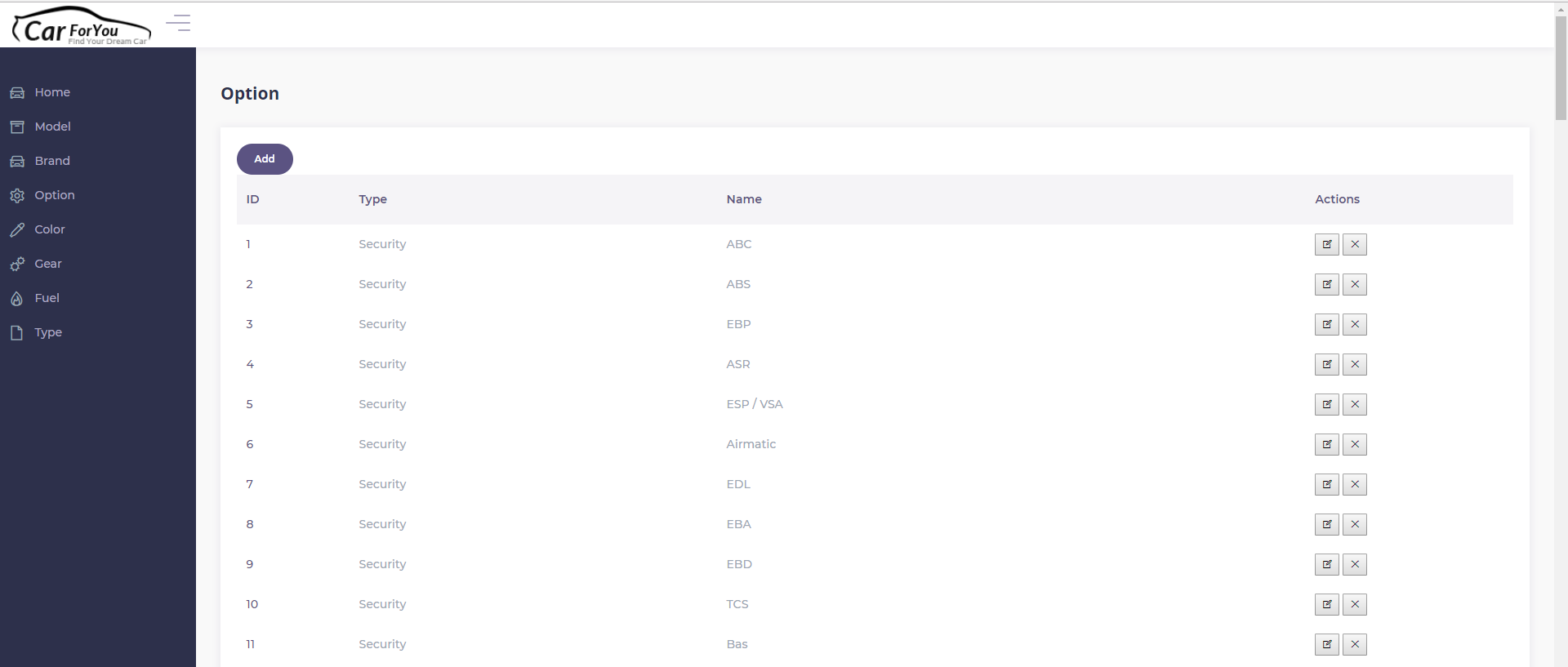


Figure 27 – Option page.

1. References

Fast, efficient and useful Query Builder and PDO Class for PHP.

<https://github.com/izniburak/pdox>