



***AUTO HEAVEN***

**By:**

Swapnil Kamate

Avadh Soni

Evan Mangone

**Table of Contents:**

<b>3</b>	<b>..... Summary</b>
<b>4</b>	<b>..... Introduction</b>
<b>4</b>	<b>..... Comparative Analysis/SWOT Analysis</b>
<b>5</b>	<b>..... Sitemap</b>
<b>6</b>	<b>..... Business Rules and Requirements</b>
<b>7</b>	<b>..... Attributes, Entities, ER Diagram</b>
<b>9</b>	<b>..... Queries to Create Tables Data</b>
<b>12</b>	<b>..... Queries to Retrieve Data</b>
<b>15</b>	<b>..... Database Application</b>
<b>19</b>	<b>..... References</b>
<b>20</b>	<b>..... Appendix</b>

## **Summary**

Welcome to AutoHeaven! We started with a simple, but powerful idea: that buying a car should be easy and straightforward. To this end, we have constructed a website which offers luxury electric and hybrid vehicles. Our target demographic to market to are consumers who care about the environment and believe it is worth spending the extra money to protect it.

Our website is still in development as we originally planned on having three main sections of the website as you can see from our sitemap below. However, we have developed an application to interact with our database. Here, users can type in a query to browse our brands, models, and accessories and find the right car. We also designed a homepage with pictures and a search bar (see page 15 - Application for more details).

The database is centered around three tables: brands, models, and accessories. The brand to models and models to accessories relationships are one-to-many (see ERD on page 7). However, we do not have pricing by models or brand as this attribute can vary within the brand (depending on what model you select) and within the model itself (depending on what accessory you select). We still have pricing for the accessories and we are working on evaluating the base pricing of our models (since base-prices would not change within the model table). The current data in the brand and model tables are data that would not change within the brand or model table. So, in the brand table we have brand\_id and brand\_name. In the model table we have model\_id, model\_name, model\_type. For attributes that can change with accessories, like horsepower and the charge time of the battery, we used base\_horsepower, and base\_chargetime as this would be consistent within the model table.

On page 9 you can see how we inserted the data into the tables. Notice that for the brand table we chose to only have luxury cars on our website. This is a strategic business decision to focus on a niche market. Our competitors have a lot of variety on their websites so we felt that someone who knew they wanted a luxury car would be able to come to our website and not waste time searching through cars that do not fit what they are looking for. Right now, we only have a few models for each brand because we do not want to overwhelm the customer, but we will be adding more variety in the future!

Finally, our application is the crowned jewel of our project. There are more details on page 15, but for a quick overview: we used flask, a web framework and python module, for creating database applications.

Now you can find the right luxury car with ease! We will be setting up customer support soon so if you have any questions or feedback let us know. Until then, enjoy being in Autoheaven when you visit our website!

## **Introduction:**

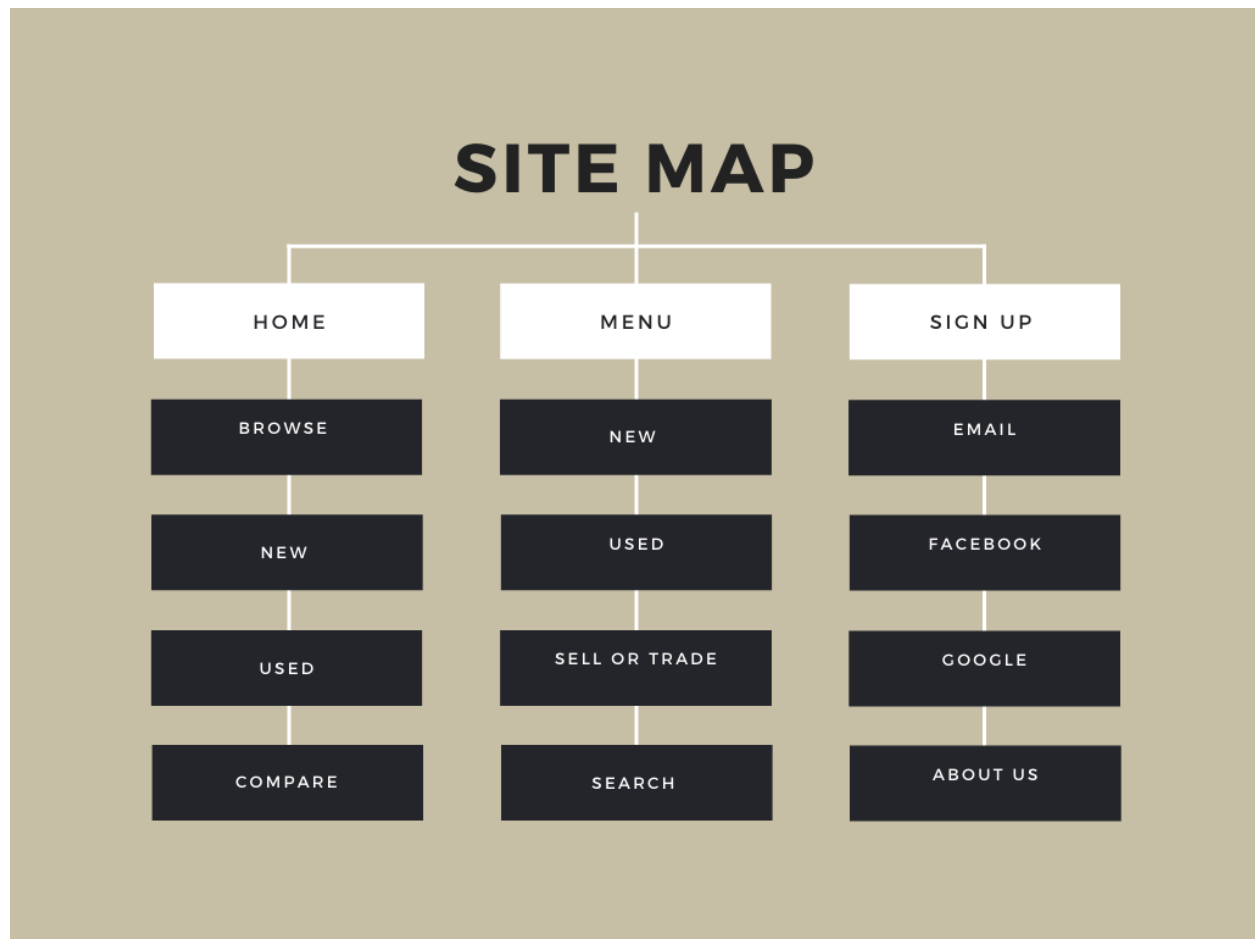
AutoHeaven provides a user friendly online experience for customers to find the right car and spread awareness through reviews and social media. Below is a comparative analysis and the business rules of how the website would operate. Our digital website helps to connect car buyers with a wide range of models and prices. We have a complete database of used cars which can be purchased or traded in. We offer customizations on style, color, interior, exterior and more! There's also flexibility in the payment method such as EMI, cash benefits etc.

## **Comparative Analysis**

### **SWOT Analysis:**

<p><b>Strengths:</b></p> <ul style="list-style-type: none"> <li>-Simple and easy to use (Select your preferred make, model, trim, and options)</li> <li>-Very good customer service</li> <li>-They have many outlets in a lot of different regions and a good distribution chain to support it</li> <li>-Skilled labor force due to high investment in training programs</li> <li>-Brand recognition demonstrated by customer retention</li> </ul>	<p><b>Weaknesses:</b></p> <ul style="list-style-type: none"> <li>- Limited to partnership deals</li> <li>-High employee turnover rate, forcing them to spend more than their competitors on training</li> <li>-Profitability ratio below industry average</li> <li>-Their product is in the mature stage of its life cycle which makes innovation necessary</li> <li>-R&amp;D costs are higher than average which cuts into their bottom line</li> <li>-They are not innovating as much as their R&amp;D investment would indicate, their technology is good but not cutting-edge</li> </ul>
<p><b>Opportunities:</b></p> <ul style="list-style-type: none"> <li>-interest rates are low meaning customers are more incentivized to take out loans</li> <li>-aggressive marketing campaigns via influencers</li> <li>-marketing their positive impact on the environment</li> <li>-using big data to better know their customer and potential customer behavior</li> <li>-take advantage of niche markets rather than losing market share to firms operating in them</li> <li>-less trade barriers will reduce cost of inputs</li> <li>-globalization allows them to operate internationally</li> <li>-the auto industry is projected to do better as vaccines become more available</li> </ul>	<p><b>Threats:</b></p> <ul style="list-style-type: none"> <li>-A poor job market and an economy in a recession makes big purchases like a car more difficult</li> <li>-Online security makes potential consumers afraid to make big purchases online</li> <li>-heavy competition and many new entrants</li> <li>-fluctuations in overseas governments and economies make represent a risk to their international operations</li> <li>-Uber, Lyft and taxis are substitute good which put the whole industry at risk</li> <li>-competitors are investing heavily in new technologies and trucare will have to do the same to keep up</li> </ul>

## Sitemap:



Being able to sign up through facebook and other social media platforms is a great way to get potential customers and spread brand awareness.

The site is user friendly as our cars are organized into new and used and further categorized by make, model, year, color and more.

Finally, the “compare” feature allows you to compare multiple offers so you always get the best one!

## **Business Rules:**

### Make to Model (1:M)

A car company (make) will have many car models.

A particular model belongs to a specific car company (make).

### Seller to used cars (1:M)

A seller can have multiple used cars (optional).

A used car should have only one seller.

\*a seller is existence dependent on having a used car to sell

### Car to Engine (1:1)

A car will have only one unique engine.

An engine belongs to only one specific car

\*a car is existent dependent on an engine

### Car to Specification Settings (M:M)

A car can be customized with many different specification settings (optional).

All the specifications can be used for many different cars (optional).

### Car to License plate (1:1)

A car will have only one license plate.

A license plate will belong to one car.

### Customer to Invoice (1:M)

A Customer can have many billing invoices (optional).

But an invoice will belong to a particular customer.

### Vehicle to Location (1:M)

A vehicle will have one location.

A location will have many vehicles.

\*a vehicle is existence dependent on a location

### Customer to Pages Visited on Website (M:M)

A customer can visit many pages on the website (optional).

A website page will have many customers visit it (optional).

## **Requirements:**

A customer should be of minimum 18 years of old to make a purchase.

A customer should have a SSN number.

## **ATTRIBUTES, ENTITIES & ENTITY RELATIONSHIP DIAGRAM:**

### **BRAND**

#### Attributes:

Brand\_ID (PK), Brand\_Name, Brand\_HQ

#### Entities:

Tesla  
Jaguar  
Audi  
Porsche  
Mercedes Benz  
BMW  
Volvo  
Polestar

### **MODEL**

#### Attributes:

Model\_ID (PK), Model\_Name, Model\_Type (SUV, Sedan, sports car),  
Model\_BaseHorsePower\*(280-500), Model\_BaseBatteryRange\*(miles on charge)(250-500),  
Model\_Length (175-200), Model\_Width(70-80), Model\_Height(55-67),  
Model\_BaseChgTime\*(CHARGING TIME - 8-10), Model\_BaseDriveWheels\*(Front, Rear, All),  
Brand\_ID (FK)

\*Attributes with "Base" means that it can change with accessories/add-ons

#### Entities:

(Tesla)  
Model S  
Model S Plaid  
Model S Plaid+  
Model X  
Model Y  
Model 3  
(Audi)  
e-tron  
(Jaguar)  
I-Pace  
(Polestar)  
Polestar-2  
Taycan  
(Volvo)  
XC40

## ACCESSORY

### Attributes:

Accessory\_ID, Accessory\_Name, Accessory\_Price, Model\_ID

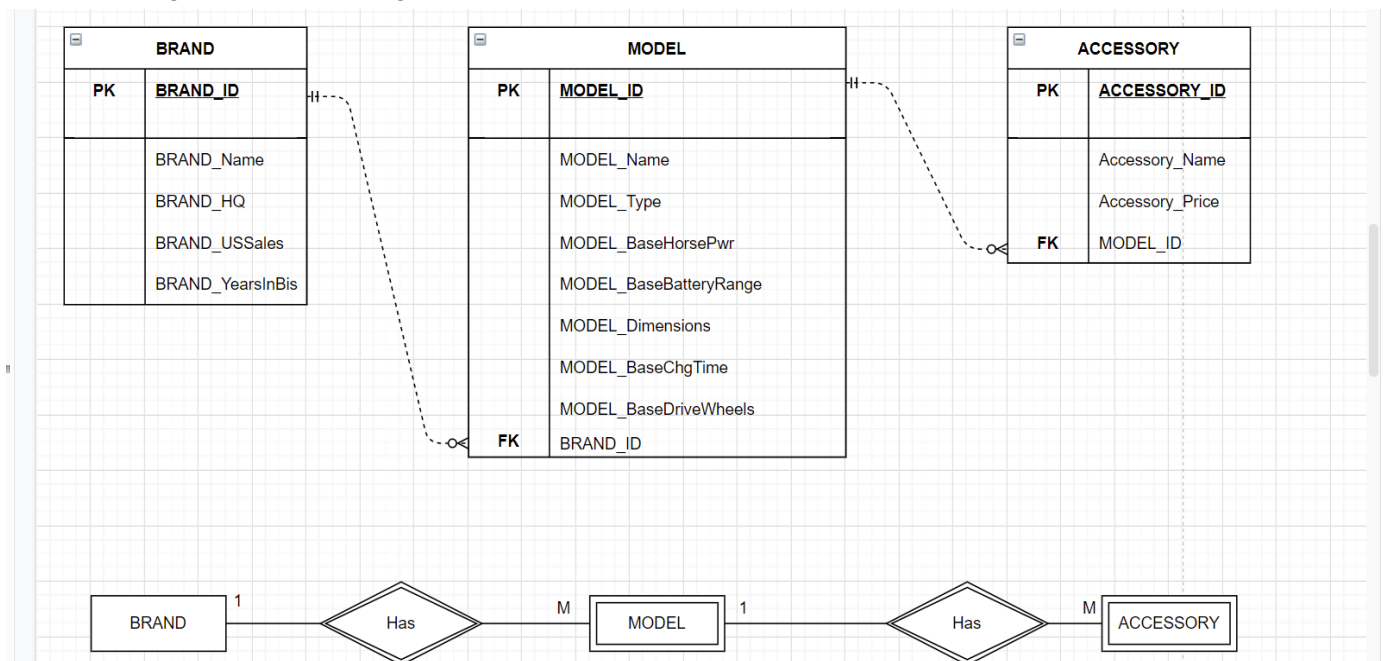
### Entities:

Heated Seats  
Larger Battery  
Faster Charge  
All Wheel Drive  
Panoramic Moonroof  
Pedestrian Detection  
Power Trunk/Liftgate  
Adaptive Cruise Control  
Remote Engine Start  
Parking Sensors  
Hands Free Doors  
Cooled Front Seats

## CAR (Not in ERD, just to help us visualize the DB)

Model\_ID, CAR\_Miles, CAR\_Price, Car\_Color, Car\_Year, Car\_NumAccidents,  
Car\_NumPriorOwners, CAR\_Year

**ERD** (No BRAND\_USSales or BRAND\_YearsInBis. MODEL\_Dimensions was split into MODEL\_Height, MODEL\_Length, and MODEL\_Width):





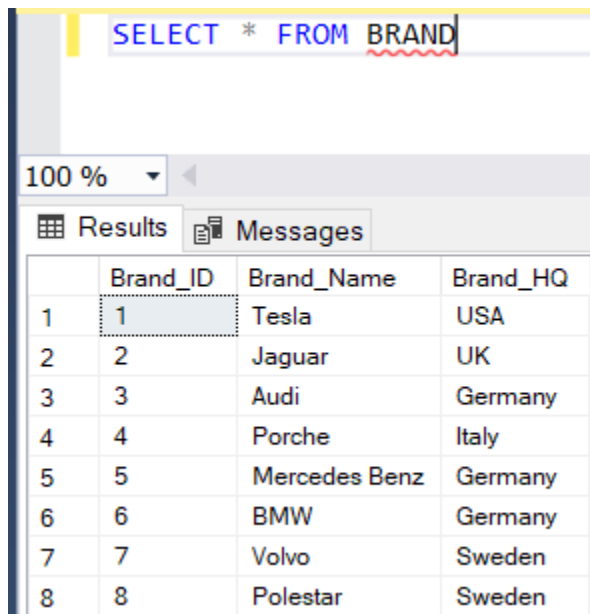
## Tables With Data:

### **Brand**

#### Queries:

```
CREATE TABLE BRAND(
Brand_ID INT Primary Key,
Brand_Name varchar(25),
Brand_HQ varchar(25)
);
INSERT INTO BRAND VALUES (1,'Tesla','USA');
INSERT INTO BRAND VALUES (2,'Jaguar','UK');
INSERT INTO BRAND VALUES (3,'Audi','Germany');
INSERT INTO BRAND VALUES (4,'Porsche','Italy');
INSERT INTO BRAND VALUES (5,'Mercedes Benz','Germany');
INSERT INTO BRAND VALUES (6,'BMW','Germany');
INSERT INTO BRAND VALUES (7,'Volvo','Sweden');
INSERT INTO BRAND VALUES (8,'Polestar','Sweden');
```

#### Screenshot:



The screenshot shows a SQL query editor with the query `SELECT * FROM BRAND` and its results displayed in a table. The table has four columns: Brand\_ID, Brand\_Name, and Brand\_HQ. The results are as follows:

	Brand_ID	Brand_Name	Brand_HQ
1	1	Tesla	USA
2	2	Jaguar	UK
3	3	Audi	Germany
4	4	Porche	Italy
5	5	Mercedes Benz	Germany
6	6	BMW	Germany
7	7	Volvo	Sweden
8	8	Polestar	Sweden

## Model

### Queries:

```
CREATE TABLE MODEL (
Model_ID INT Primary Key,
Model_Name varchar(25),
Model_Type varchar(25),
Model_BaseHorsePwr Number(7,2),
Model_BaseBatteryRange Number(7,2),
Model_length Number(7,2),
Model_width Number(7,2),
Model_height Number(7,2),
Model_BaseChgTime DateTime,
Model_BaseDriveWheels varchar(25),
Brand_ID INT
Constraint FK Foreign Key (Brand_ID)
References Brand (Brand_ID)
);
```

```
INSERT INTO MODEL VALUES (101,'MODEL S','SUV',400,520,180,75,57,6,'All',1)
INSERT INTO MODEL VALUES (102,'Jaguar
I-Pace','Sedan',394,234,184,75,61,8,'Front',2)
INSERT INTO MODEL VALUES (103,'Audi etron','SUV',355,222,193,76,66,7,'Rear',3)
INSERT INTO MODEL VALUES (104,'Porche
Taycan','Sports',616,201,177,70,48,9,'Front',4)
INSERT INTO MODEL VALUES (105,'Mercedes
EQ','Sedan',500,480,178,68,55,8,'All',5)
INSERT INTO MODEL VALUES (106,'BMW i3','Compact',181,153,160,65,58,5,'Front',6)
INSERT INTO MODEL VALUES (107,'XC90','Sedan',380,475,200,78,66,10,'Rear',7)
INSERT INTO MODEL VALUES (108,'Polestar 1','Sports',619,550,175,80,62,8,'All',8)
```

### Screenshot:

SELECT \* FROM MODEL

100 %

Results Messages

	Model_ID	Model_Name	Model_Type	Model_BaseHorsePwr	Model_BaseBatteryRange	Model_length	Model_width	Model_height	Model_BaseChgTime	Model_BaseDriveWheels	Brand_ID
1	101	MODEL S	SUV	400.00	520.00	180.00	75.00	57.00	1900-01-07 00:00:00.000	All	1
2	102	Jaguar I Pace	Sedan	394.00	234.00	184.00	75.00	61.00	1900-01-09 00:00:00.000	Front	2
3	103	Audi etron	SUV	355.00	222.00	193.00	76.00	66.00	1900-01-08 00:00:00.000	Rear	3
4	104	Porche Taycan	Sports	616.00	201.00	177.00	70.00	48.00	1900-01-10 00:00:00.000	Front	4
5	105	Mercedez EQ	Sedan	500.00	480.00	178.00	68.00	55.00	1900-01-09 00:00:00.000	All	5
6	106	BMW i3	Compact	181.00	153.00	160.00	65.00	58.00	1900-01-06 00:00:00.000	Front	6
7	107	XC90	Sedan	380.00	475.00	200.00	78.00	66.00	1900-01-11 00:00:00.000	Rear	7
8	108	Polestar 1	Sports	619.00	550.00	175.00	80.00	62.00	1900-01-09 00:00:00.000	All	8

## Accessory

### Queries:

```
CREATE TABLE ACCESSORY(
Accessory_ID int primary key,
Accessory_name varchar(25),
Accessory_Price number(7,2),
Model_ID INT
Constraint FK2 Foreign Key (Model_ID)
References MODEL (Model_ID)
);
```

```
INSERT INTO ACCESSORY VALUES (1001,'Heated Seats',140,101)
INSERT INTO ACCESSORY VALUES (1002,'Larger Battery',166,102)
INSERT INTO ACCESSORY VALUES (1003,'Faster Charge',500,103)
INSERT INTO ACCESSORY VALUES (1004,'All Wheel Drive',250,104)
INSERT INTO ACCESSORY VALUES (1005,'Hands Free Doors',350,105)
INSERT INTO ACCESSORY VALUES (1006,'Adaptive Cruise Control',550,106)
INSERT INTO ACCESSORY VALUES (1007,'Pedestrian Detection',400,107)
INSERT INTO ACCESSORY VALUES (1008,'Panoramic Moonroof',280,108)
```

### Screenshot:

`SELECT * FROM ACCESSORY`

100 %

Results Messages

	Accessory_ID	Accessory_name	Accessory_Price	Model_ID
1	1001	Heated Seats	140.00	101
2	1002	Larger Battery	166.00	102
3	1003	Faster Charge	500.00	103
4	1004	All Wheel Drive	250.00	104
5	1005	Hands Free Doors	350.00	105
6	1006	Adaptive Cruise Control	550.00	106
7	1007	Pedestrian Detection	400.00	107
8	1008	Panoramic Moonroof	280.00	108

## SQL queries to retrieve data

1) The below query gives model details for a particular Brand. If a user clicks on Brand say Tesla, all the details about Tesla will be displayed along with Brand details

/\*When the user clicks on a certain brand (let's say Tesla) then the models for that specific brand is displayed \*/

```
SELECT * FROM MODEL AS M
JOIN BRAND AS B
ON B.Brand_ID = M.Brand_ID
WHERE Brand_Name = 'Tesla'
```

`SELECT * FROM MODEL AS M  
JOIN BRAND AS B  
ON B.Brand_ID = M.Brand_ID  
WHERE Brand_Name = 'Tesla'`

100 %

Results Messages

	Model_ID	Model_Name	Model_Type	Model_BaseHorsePwr	Model_BaseBatteryRange	Model_length	Model_width	Model_height	Model_BaseChgTime	Model_BaseDriveWheels	Brand_ID	Brand_ID	Brand_Name	Brand_HQ
1	101	MODEL S	SUV	400.00	520.00	180.00	75.00	57.00	1900-01-07 00:00:00.000	All	1	1	Tesla	USA

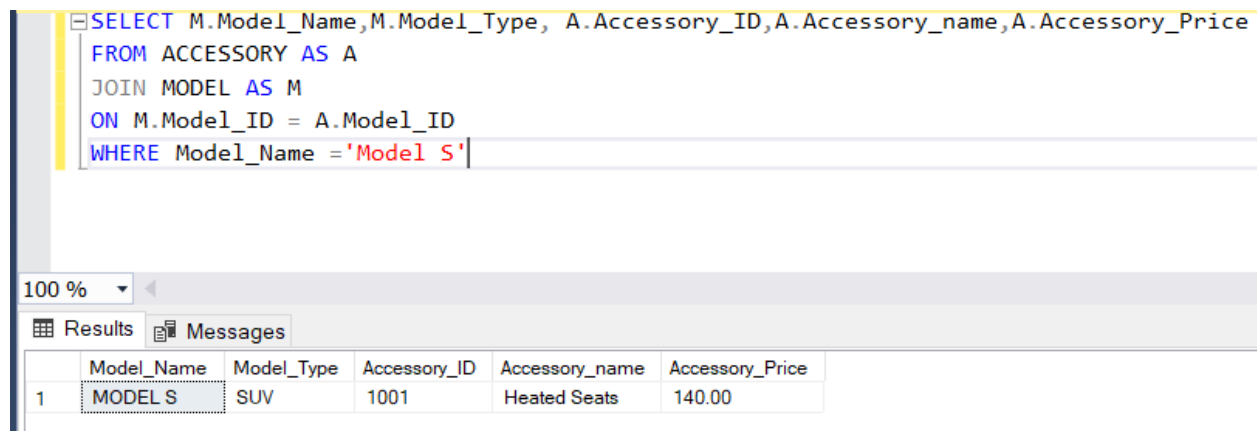
2) The below query gives details of the model along with it's accessory. Query can be modified to display details about any Model. Once a user clicks on a certain model it displays accessory details.

/\*When the user clicks on a certain model (Let's say Model S), then the specific information including the price of that model is displayed\*/

```

SELECT M.Model_Name,M.Model_Type,
A.Accessory_ID,A.Accessory_name,A.Accessory_Price
FROM ACCESSORY AS A
JOIN MODEL AS M
ON M.Model_ID = A.Model_ID
WHERE Model_Name ='Model S'

```



The screenshot shows a SQL query editor with the following query:

```

SELECT M.Model_Name,M.Model_Type, A.Accessory_ID,A.Accessory_name,A.Accessory_Price
FROM ACCESSORY AS A
JOIN MODEL AS M
ON M.Model_ID = A.Model_ID
WHERE Model_Name ='Model S'

```

Below the query editor, the 'Results' tab is active, displaying a single row of data:

	Model_Name	Model_Type	Accessory_ID	Accessory_name	Accessory_Price
1	MODEL S	SUV	1001	Heated Seats	140.00

3) The below query was written to fetch the details about the vehicle dimensions. When a user enters a particular brand, say Audi, it will give vehicle dimensions such length,width and height.

**\*We did not create a color table since color can vary within model, we only made three tables in our database: Brand, Model, Accessory- instead we figured vehicle dimensions do not vary within Model so here we fetched data pertaining to that**

*/\*When user enters a model it gives dimensions of the vehicle\*/*

```

SELECT B.Brand_Name,M.Model_length,M.Model_width,M.Model_height
FROM MODEL AS M
JOIN BRAND AS B
ON B.Brand_ID = M.Brand_ID
WHERE Brand_Name ='Audi'

```

```

SELECT B.Brand_Name,M.Model_length,M.Model_width,M.Model_height
FROM MODEL AS M
JOIN BRAND AS B
ON B.Brand_ID = M.Brand_ID
WHERE Brand_Name = 'Audi'

```

100 %

Results Messages

	Brand_Name	Model_length	Model_width	Model_height
1	Audi	193.00	76.00	66.00

4) This query was written by combining all the 3 tables. It gives quick details about a car's brand,model and it's accessories. Below is an example of it.

/\*When the user clicks on a certain brand it gives quick details about it's model and accessories\*/

```

SELECT
B.Brand_Name,B.Brand_HQ,M.Model_Name,M.Model_Type,M.Model_BaseDriveWheels,A.Accessory_name,A.Accessory_Price
FROM BRAND AS B
JOIN MODEL AS M
ON B.Brand_ID = M.Brand_ID
JOIN ACCESSORY AS A
ON M.Model_ID = A.Model_ID
WHERE B.Brand_Name = 'BMW'

```

```

SELECT B.Brand_Name,B.Brand_HQ,M.Model_Name,M.Model_Type,M.Model_BaseDriveWheels,A.Accessory_name,A.Accessory_Price
FROM BRAND AS B
JOIN MODEL AS M
ON B.Brand_ID = M.Brand_ID
JOIN ACCESSORY AS A
ON M.Model_ID = A.Model_ID
WHERE B.Brand_Name = 'BMW'

```

100 %

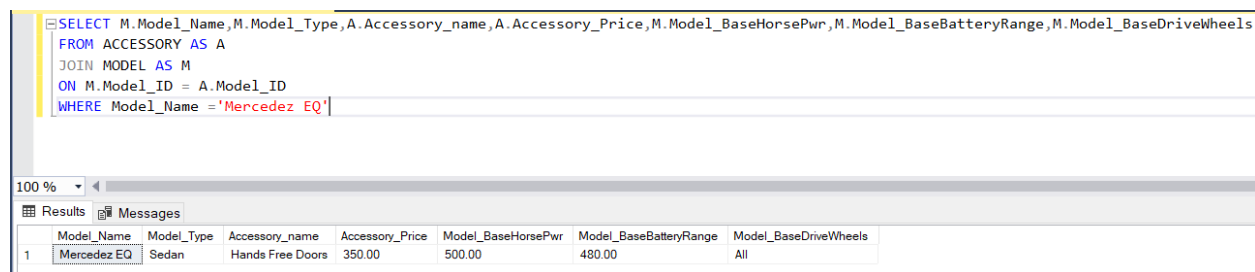
Results Messages

	Brand_Name	Brand_HQ	Model_Name	Model_Type	Model_BaseDriveWheels	Accessory_name	Accessory_Price
1	BMW	Germany	BMW i3	Compact	Front	Adaptive Cruise Control	550.00

5) Below query gives details about the vehicle engine. When a user clicks on a certain model, it displays the model battery range, horse power etc.

/\*When the user clicks on a certain model, it gives model details and accessory details specific to the vehicle engine.\*/

```
SELECT
M.Model_Name,M.Model_Type,A.Accessory_name,A.Accessory_Price,M.Model_Base
HorsePwr,M.Model_BaseBatteryRange,M.Model_BaseDriveWheels
FROM ACCESSORY AS A
JOIN MODEL AS M
ON M.Model_ID = A.Model_ID
WHERE Model_Name ='Mercedes EQ'
```



The screenshot shows a database query editor with the following SQL query:

```
SELECT M.Model_Name,M.Model_Type,A.Accessory_name,A.Accessory_Price,M.Model_BaseHorsePwr,M.Model_BaseBatteryRange,M.Model_BaseDriveWheels
FROM ACCESSORY AS A
JOIN MODEL AS M
ON M.Model_ID = A.Model_ID
WHERE Model_Name ='Mercedes EQ'
```

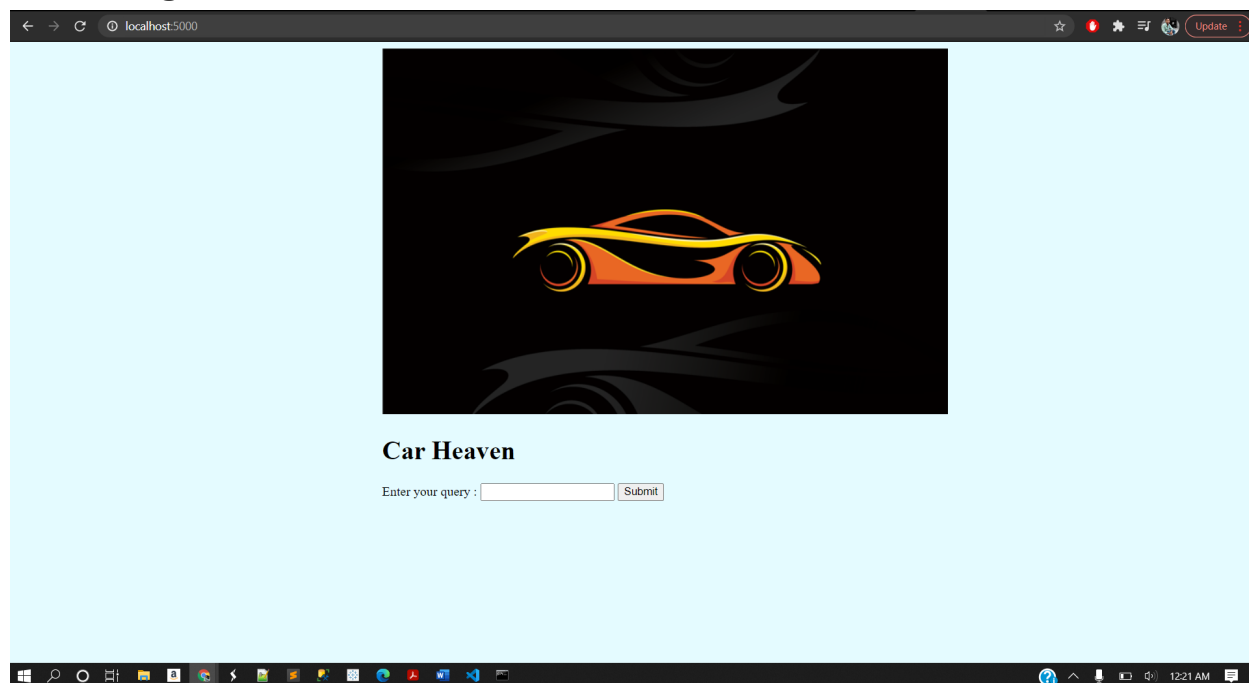
Below the query editor, the results are displayed in a table with 7 columns: Model\_Name, Model\_Type, Accessory\_name, Accessory\_Price, Model\_BaseHorsePwr, Model\_BaseBatteryRange, and Model\_BaseDriveWheels. The results show one row for the Mercedes EQ model.

	Model_Name	Model_Type	Accessory_name	Accessory_Price	Model_BaseHorsePwr	Model_BaseBatteryRange	Model_BaseDriveWheels
1	Mercedes EQ	Sedan	Hands Free Doors	350.00	500.00	480.00	All

## Database Application

We have built our Database application using Flask. Flask is a web framework, it's a Python module that lets you develop web applications easily. Flask depends upon a jinja template engine. We have connected Flask python with our database by giving connection details and using HTML/CSS we are displaying our results to users. Our Database application is a query based application where users can enter any type of select queries to get details of the car. Application has been built considering all the test cases. A person can not modify database models by entering DML commands. He can just view different car details such as BRAND, MODEL and ACCESSORIES. Although users have all the rights to run any select join queries and get data. Total 3 files have been used to build this application. Flask.py , display.html and main.html. Upon entering the query in the database application it will take us to the main page and there we can see our result. A button has been added in the main page so that users can go back and enter new query to get details. Below are some screenshots of our Database application

## Main Page:



## Query Results:

model_id	model_name	model_type	model_basehorsepwr	model_basebatteryrange	model_length	model_width	model_height	model_basechgtme	model_basedrivewheels	brand_id
101	MODEL S	SUV	400.00	520.00	180.00	75.00	57.00	1900-01-07 00:00:00	All	1
102	Jaguar I Pace	Sedan	394.00	234.00	184.00	75.00	61.00	1900-01-09 00:00:00	Front	2
103	Audi etron	SUV	355.00	222.00	193.00	76.00	66.00	1900-01-08 00:00:00	Rear	3
104	Porche Taycan	Sports	616.00	201.00	177.00	70.00	48.00	1900-01-10 00:00:00	Front	4
105	Mercedes EQ	Sedan	500.00	480.00	178.00	68.00	55.00	1900-01-09 00:00:00	All	5
106	BMW i3	Compact	181.00	153.00	160.00	65.00	58.00	1900-01-06 00:00:00	Front	6
107	XC90	Sedan	380.00	475.00	200.00	78.00	66.00	1900-01-11 00:00:00	Rear	7
108	Polestar 1	Sports	619.00	550.00	175.00	80.00	62.00	1900-01-09 00:00:00	All	8

[Return to query](#)

**Code:**  
**Flask.py**



```

flask_app.py
1  from datetime import datetime
2  from flask import render_template, flash, request, redirect, Flask
3  import pypodbc
4  from datetime import datetime
5
6  app = Flask(__name__)
7  app.secret_key = 'secret key'
8
9  # creating connection Object which will contain SQL Server Connection
10 connection = pypodbc.connect('Driver={ODBC Driver 17 for SQL Server};Server=SWAPNIL;Database=Fact_db;uid=skamate;pwd=skamate')# Creating
11 cursor = connection.cursor()
12
13 @app.route('/')
14 def main_page():
15     return render_template("main.html")
16
17 @app.route("/form_submit", methods=['POST'])
18 def back_call():
19     query = request.form.getlist('text')[0]
20     query = query.lower()
21     x = query.split(" ")[0]
22
23     if x == "select":
24         cursor.execute(query)
25         records = [tuple(i[0] for i in cursor.description)]
26
27         for row in cursor:
28             records.append(row)
29         return render_template("display.html", values = records)
30     return redirect("/")
31

```

## Display.html

```

templates > <> display.html > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Test1</title>
5  </head>
6  <body>
7  |   <form method="GET" action="/">
8  |   |   <table style='border:1px solid black'>
9  |   |   |   {%for each in values%}
10 |   |   |   <tr style='border:1px solid black'>{%for value in each%}
11 |   |   |   |   <td style='border:1px solid black'>{{value}}</td>
12 |   |   |   |   {%endfor%}
13 |   |   |   </tr>
14 |   |   |   {%endfor%}
15 |   |   </table>
16 |   |   <input type="submit" value="Return to query">
17 |   </form>
18
19
20 </body>
21 </html>

```

## Main.html

```
templates > main.html > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Home Page</title>
5  </head>
6  <body style="background-color: #e4fbff;">
7      <div style="align-items: center;padding-left: 450px;">
8
9          
10         <br>
11         <h1> Car Heaven</h1>
12
13
14         <form method ="POST" action="/form_submit">
15             Enter your query : <input type="text" name="text">
16             <input type="submit" name= "submit">
17         </form>
18
19     </div>
20
21 </div>
22 </body>
23 </html>
```

## References

<https://www.truecar.com/>

<https://plugstar.com/>

<https://www.macrotrends.net/stocks/charts/TRUE/truecar/profit-margins>

<https://www.washingtonpost.com/business/2021/01/28/us-economy-recession/>

<https://finance.yahoo.com/quote/TRUE/financials/>

<https://www.cnbc.com/2021/01/14/with-a-lot-of-optimism-and-vaccine-hopes-us-auto-sales-could-increase-as-much-as-10percent-in-2021.html>

<https://www.consumeraffairs.com/automotive/truecar.html>

## Appendix

Tables we did not use (to help us better envision the ones we did use):

Employee: EMP\_NAME,EMP\_ADDRESS,EMP\_PHONE,EMP\_EMAIL

Customer: C\_NAME,C\_ADDRESS,C\_PHONE,C\_EMAIL,C\_SSN,C\_INVOICE

Car: CAR\_MODEL,CAR\_MILES,CAR\_MILEAGE,CAR\_PRICE