School of Information Studies
Syracuse University

IST 615 – Cloud Management

Final Project Report

Car Retail Website

Due Date: 12/05/2022
Submission Date: 12/05/2022
Number of Pages: 14

Team members:
Mikhail Pravin Pinto (mpinto01@syr.edu)
Kirat Saran (ksaran@syr.edu)

Contents

Introduction:

We are building a car retail website using cloud where buyers can buy cars based on manufacturers, mileage, distance traveled, etc since the demand to buy vehicles has increased post pandemic.
The dataset is from the site Craigslist which was uploaded to Kaggle.com.
Link to the dataset:
https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data?datasetId=62920
We have used Microsoft Azure Cloud Services to implement the solution.

Implementation:

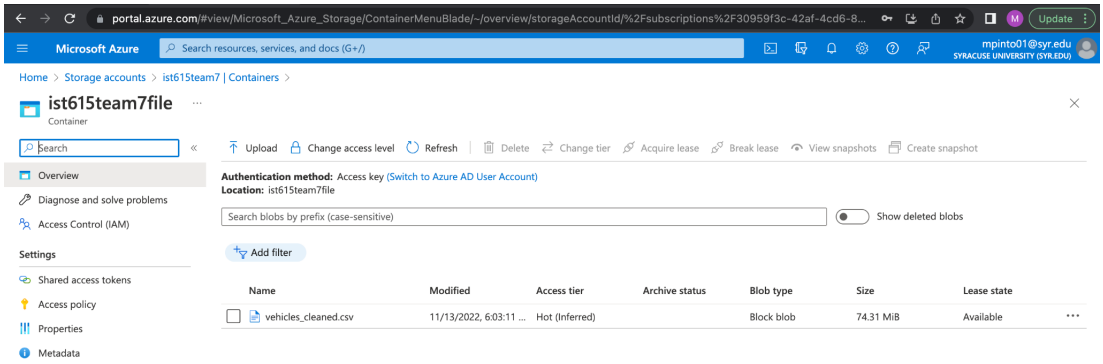
1. *Data Cleaning*

The dataset was downloaded in the form of a csv file. We cleaned the data by removing rows with null values in columns like condition, manufacturer, model, fuel, odometer, and transmission.
Column with the County name was also dropped as it had more than 50% null values. Similarly, the size and region URL was also removed due to a lot of missing values. The description was also dropped as the column contained identical descriptions for more than 50% of the records.

2. *Azure Blob Storage*

Our cleaned dataset file ‘vehicles_cleaned.csv’ was uploaded to Azure blob service and stored in a container.

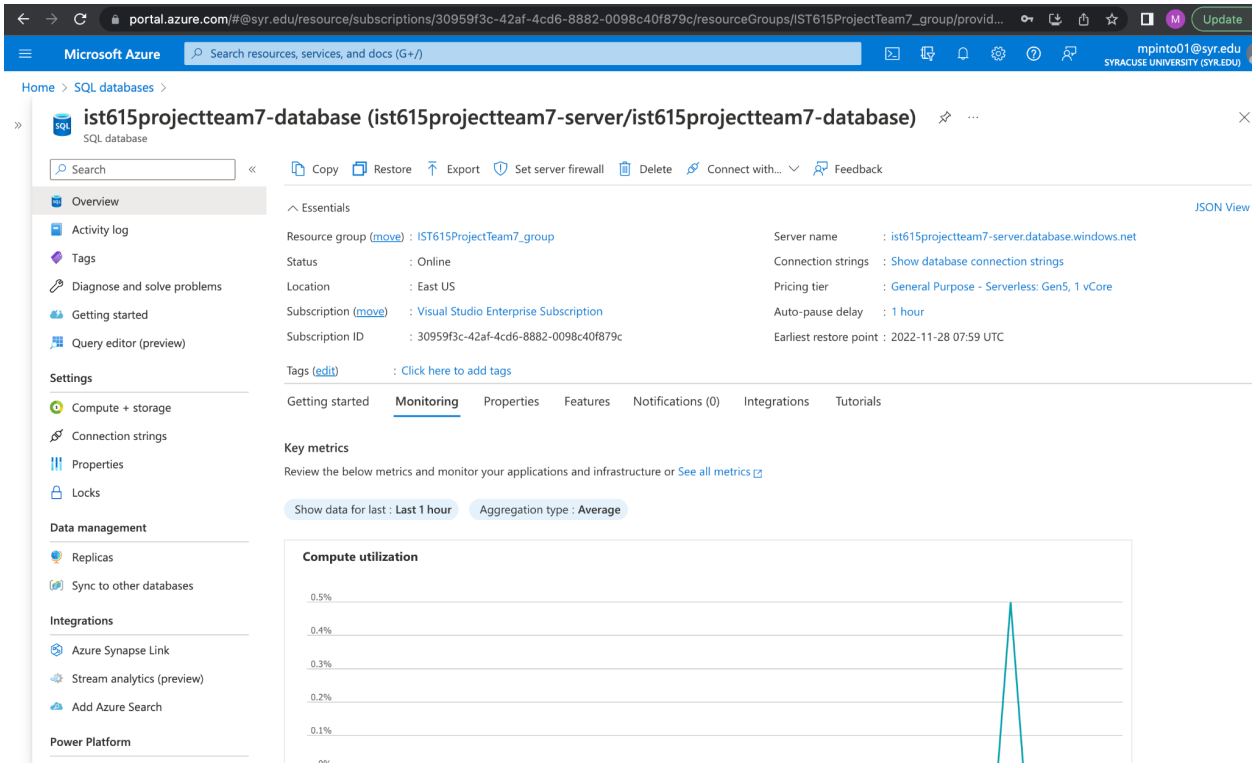
Screenshot 1: A container created in Azure blob storage to upload and store our cleaned csv data file.

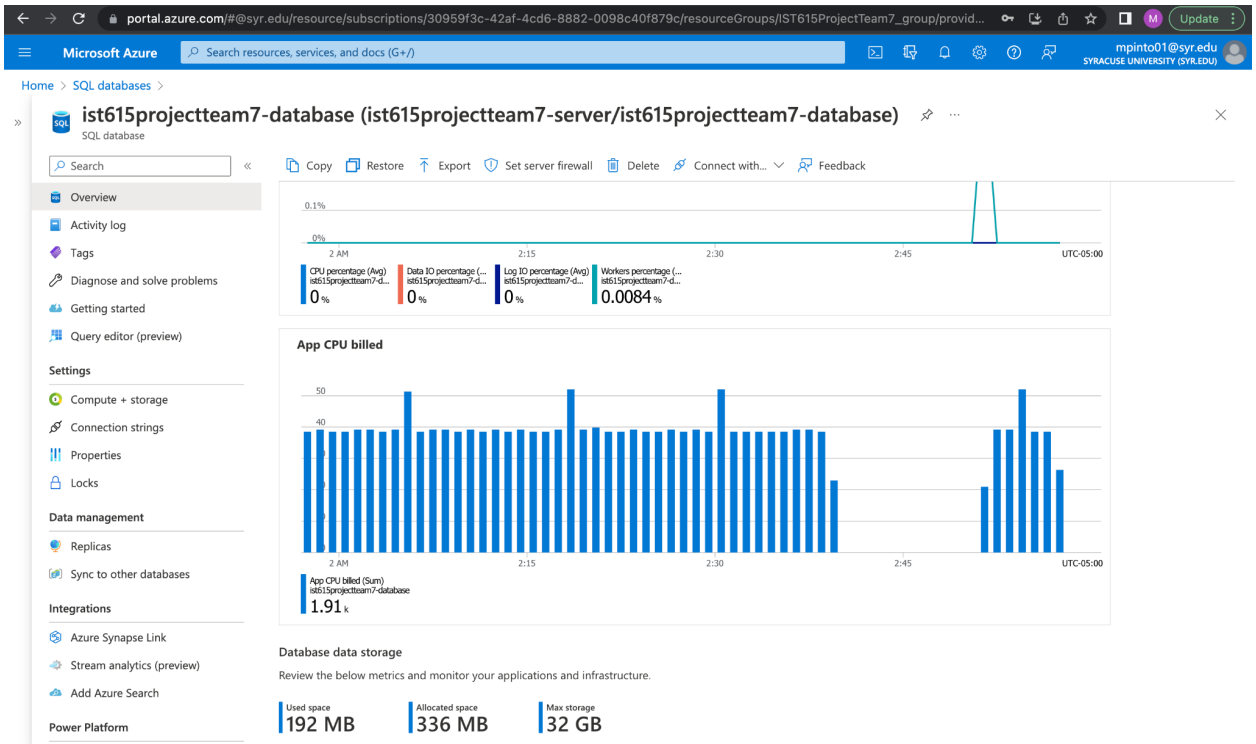


4. *Azure SQL Database*

The SQL Database was set up so that the data from the blob storage can be integrated with the Database which will finally communicate with the website.

Screenshot 2 and 3: Azure SQL Database setup created for the purpose of data integration.

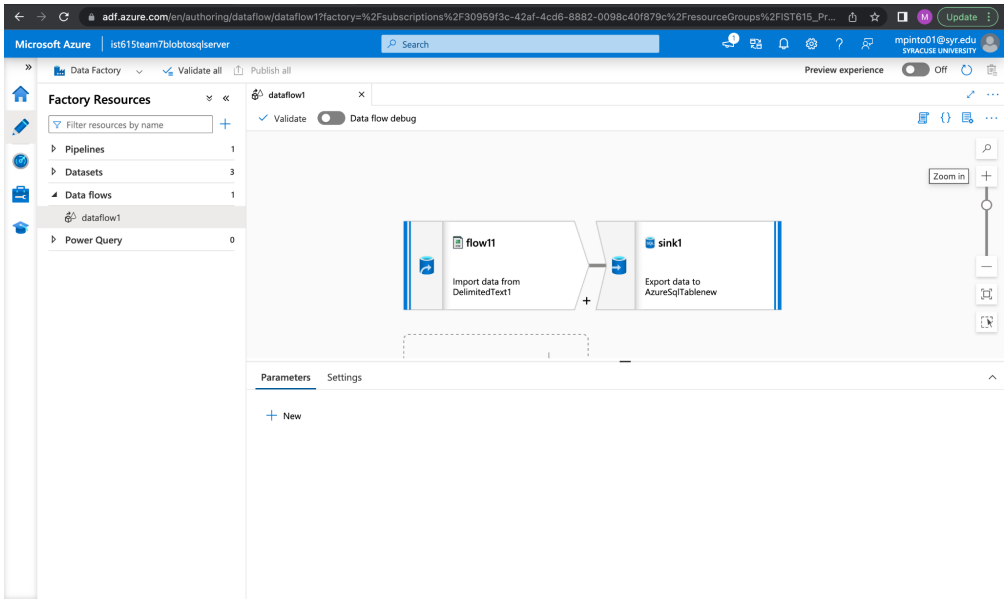


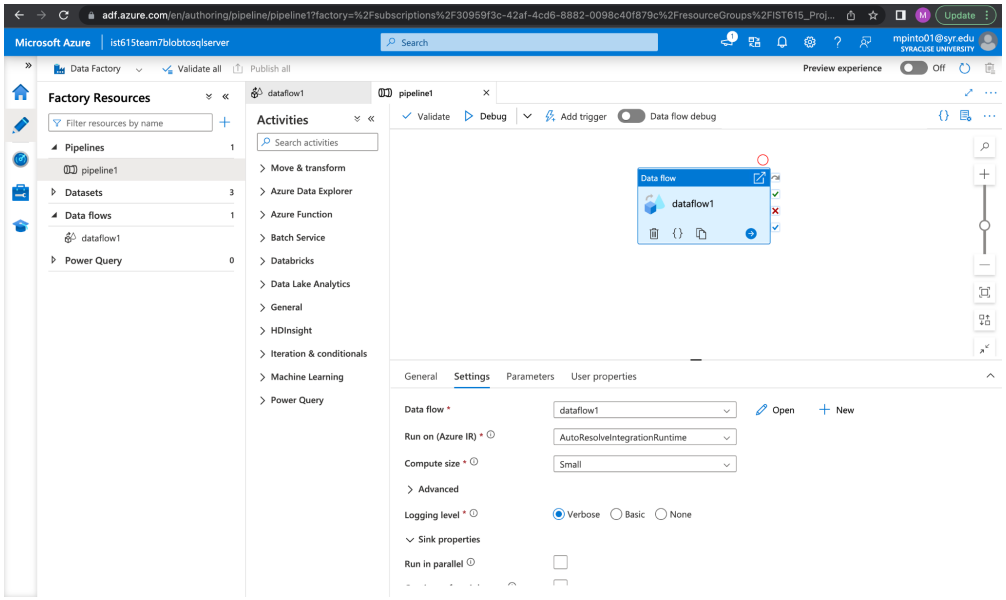


3. *Azure Data Factory*

Azure Data Factory has been used to facilitate data engineering and data integration. A data flow was created to get data from the Azure Blob storage container to Azure SQL DB. This was executed using a pipeline to get data from Azure blob storage to Azure SQL DB. To achieve this we had to set up the source i.e Azure Blob storage in a right way so that the column headers and the values are accurately analyzed. Finally using the Data flow each field in the csv file from the Azure Blob storage was mapped to the column headers. Using this would make it easy for the destination i.e the Azure SQL Database.

Screenshot 4 and 5: Azure Data Factory setup to create a data workflow from blob storage to SQL DB.

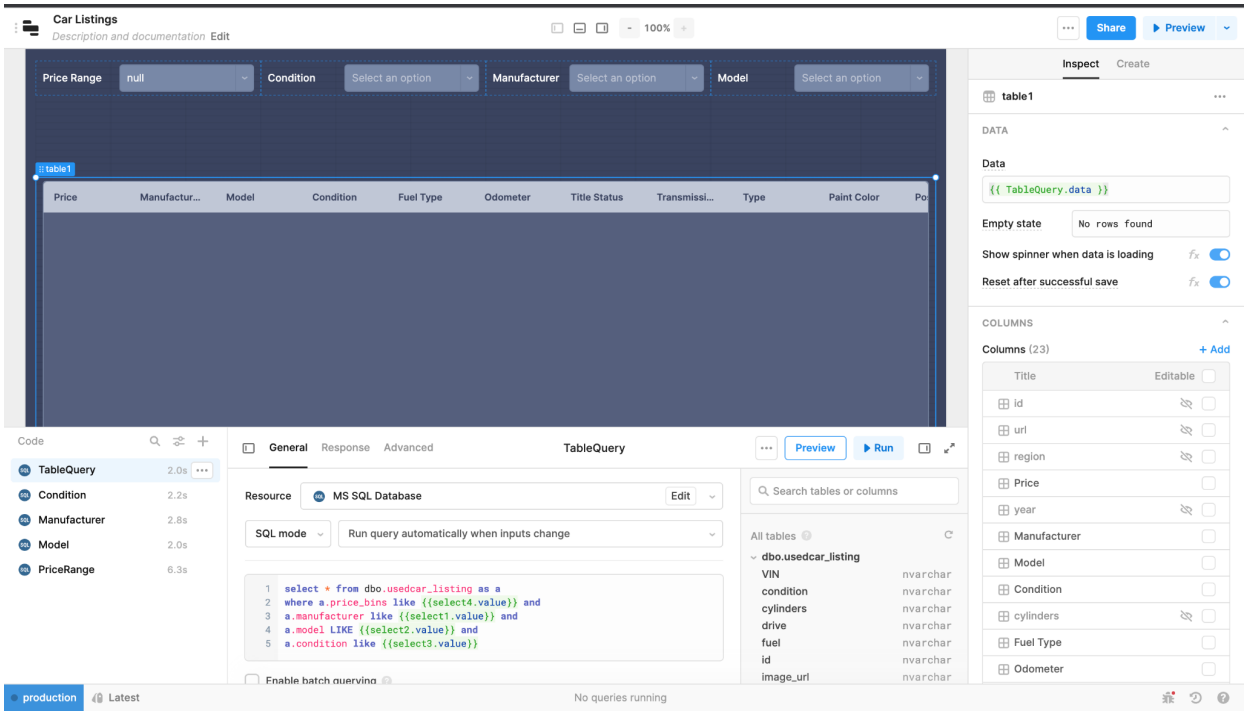




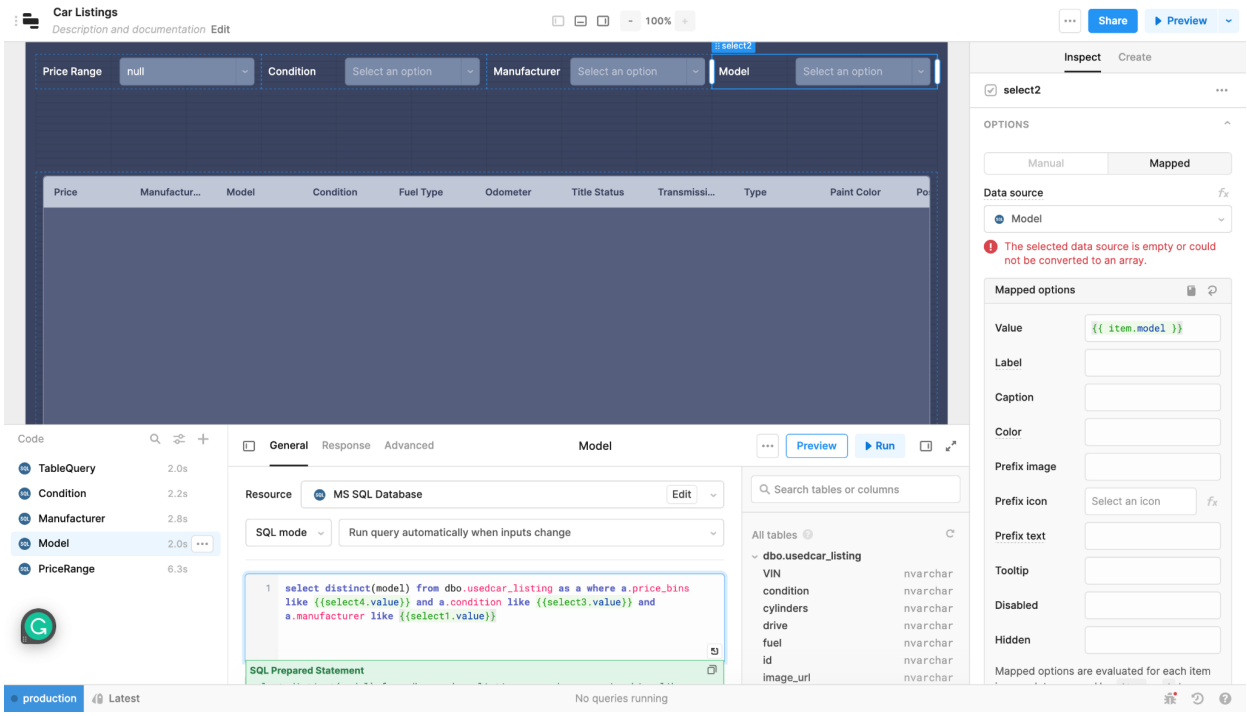
4. Retool Service

Retool is a platform where they allow their user to connect to databases and create an application which acts like a CRUD application. To create our website we have created three different pages one in which we can look for the cars listed on the website. To do that basic SQL + JavaScript is used as shown in the below screenshot. For retool to communicate with the database some ip's were added to the firewall rules of the database.

Screenshot 6 and 7: SQL queries used for the development of webpage to display cars.

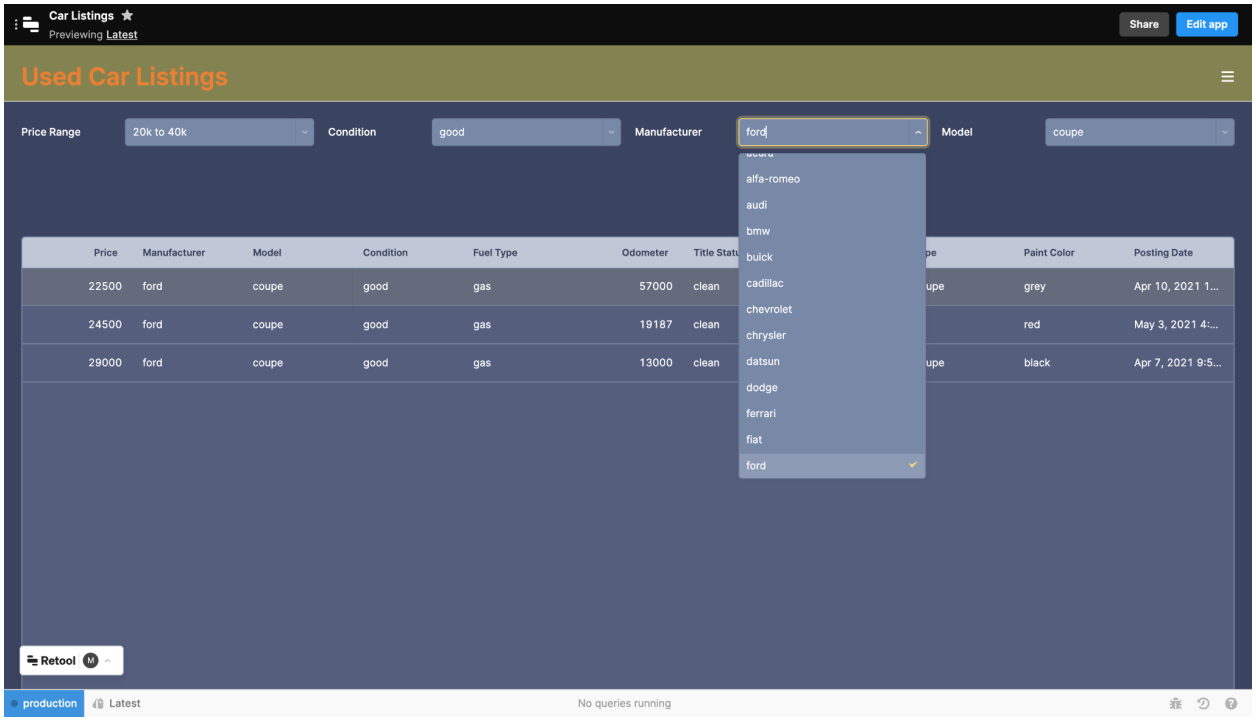


Each of the cells is attached to a query which gets values from the database based on the selections made by the user. This can be seen in the screenshot below.



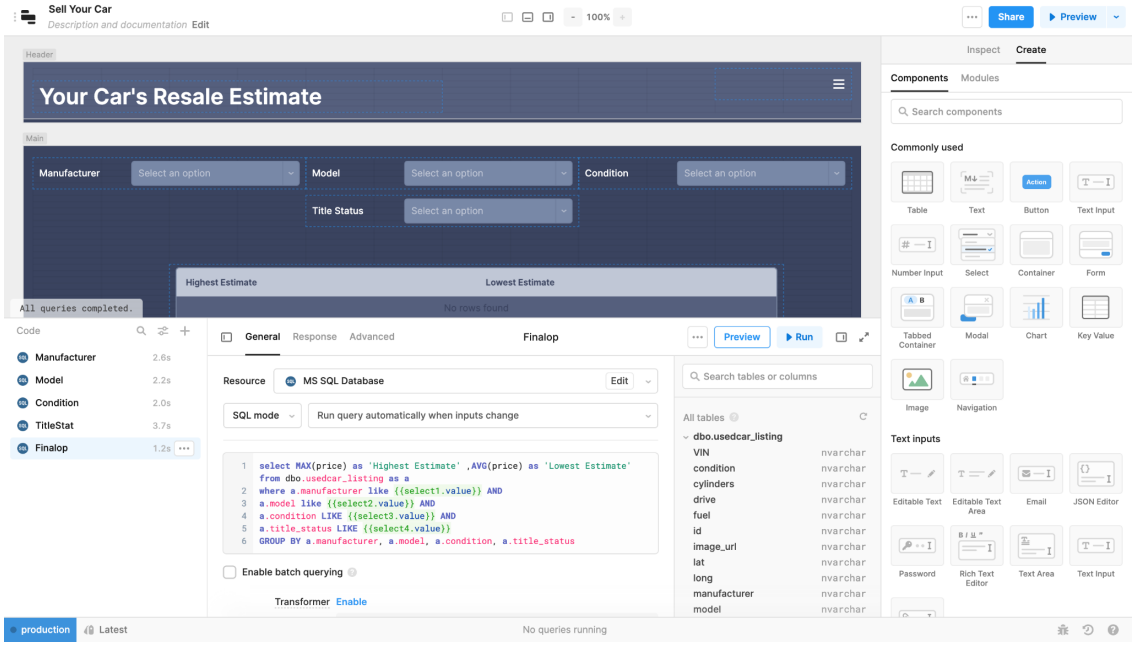
Once the user enters all the fields they will be able to the listings that are available on the site. It can be seen in the screenshot below. Paint type was not included as we did not have enough information for each category of cars.

Screenshot 8: Webpage for users to list used cars

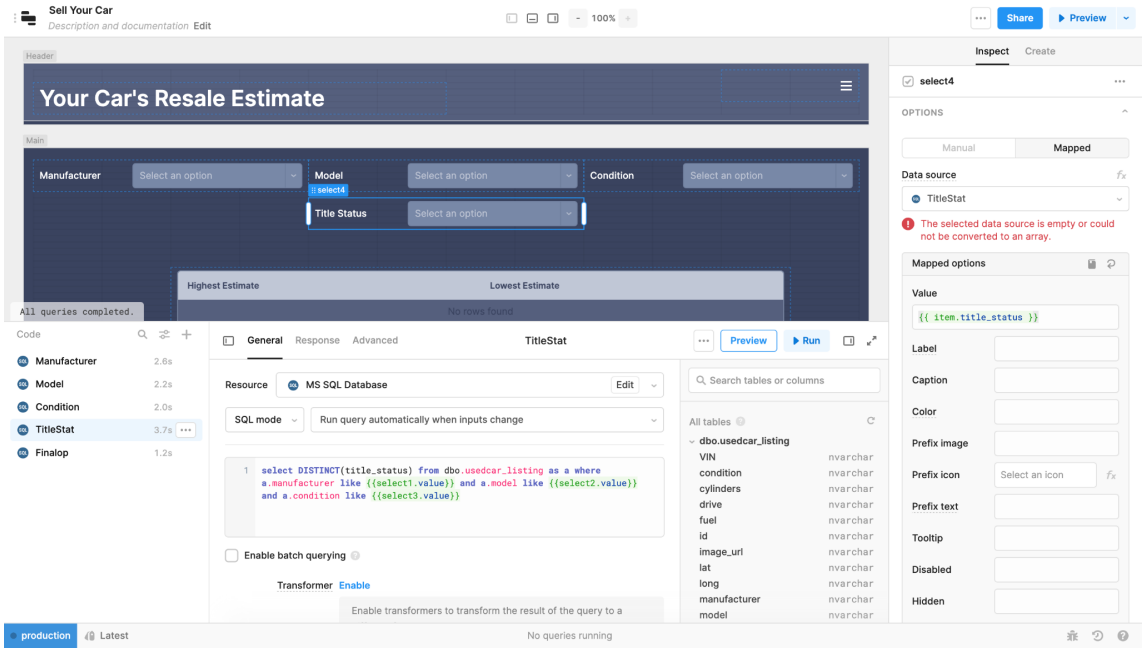


If the user wants to check the value of their car, we have created a new page where the user can check the highest and the lowest estimated resale value according to the data that we have in our database. This page can be accessed by clicking the navigation button on the top and selecting know your car's worth. Similar to the first page the user can select options from the select lists that are on the page. Once the user selects all four then they get an estimate of their car. The query used to trigger this action is mentioned in the screenshot below. In that query the highest estimate is the highest value that the car has been listed in our system with i.e by the manufacturer, model, condition and the title status (It tells us the category for times the car has been crashed and the parts have been changed for the same).

Screenshot 9 and 10: SQL queries used for the development of webpage to display price estimates for any user's car

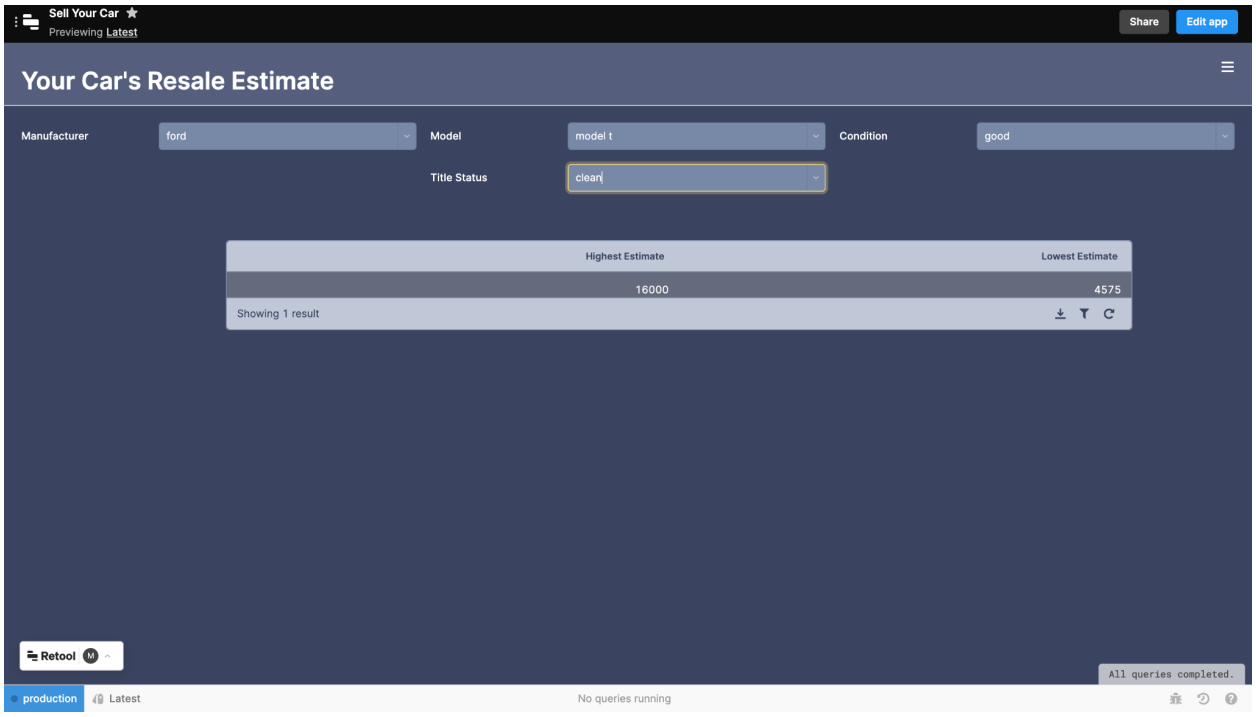


Similar to the previous page each selection list is connected to a query which retrieves data from the database. The example for this is in the screenshot below.



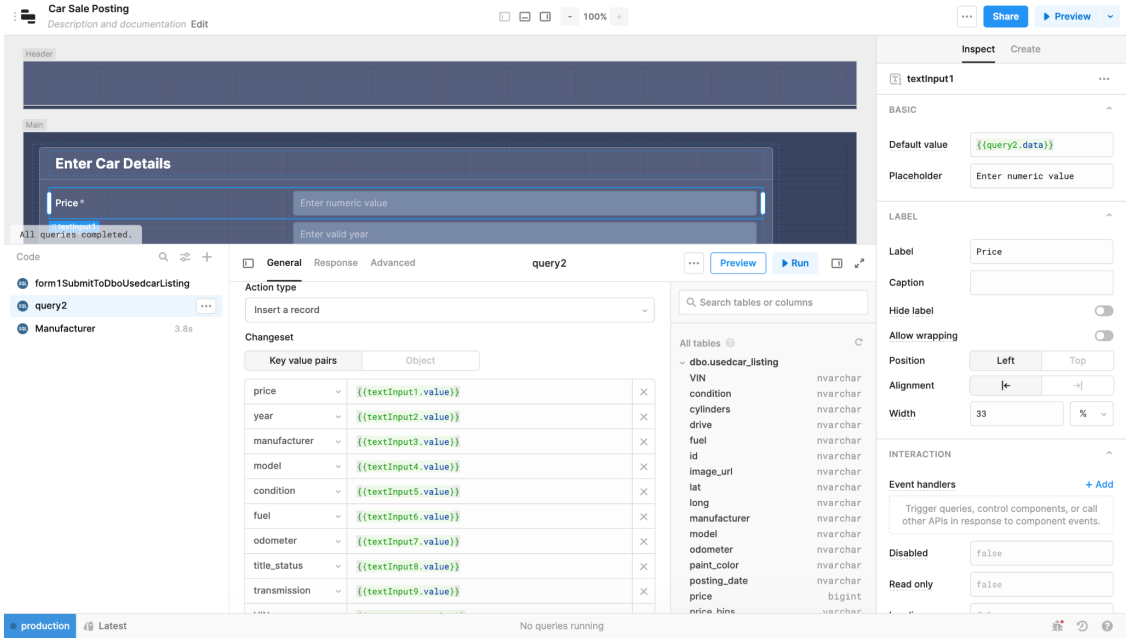
A full working example of this can be seen in the visual below. The highest estimate is the highest ever posted estimate for the selections made by the user and the lowest is the average of all the listings for the selections for that car type. Paint type was not included as we did not have enough information for each category of cars.

Screenshot 11: Webpage for users to list price estimates for their car

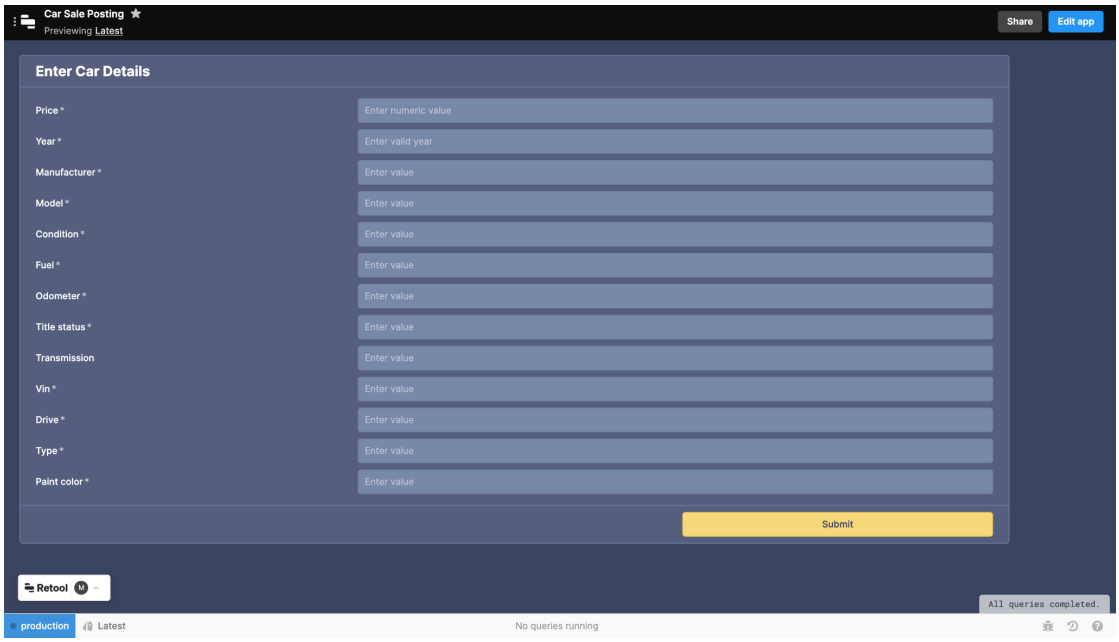


Lastly a third page was created so that the users will be able to post their own car for sale. For this the following options were selected. Each of the fields have to be filled in order to post the car for sale. To post the query an Insert action has been used which works exactly like the SQL insert into database command. Each of the fields has been given some constraints to enter the information.

Screenshot 12: Setup to insert the user's car's data that is put for sale into our database.



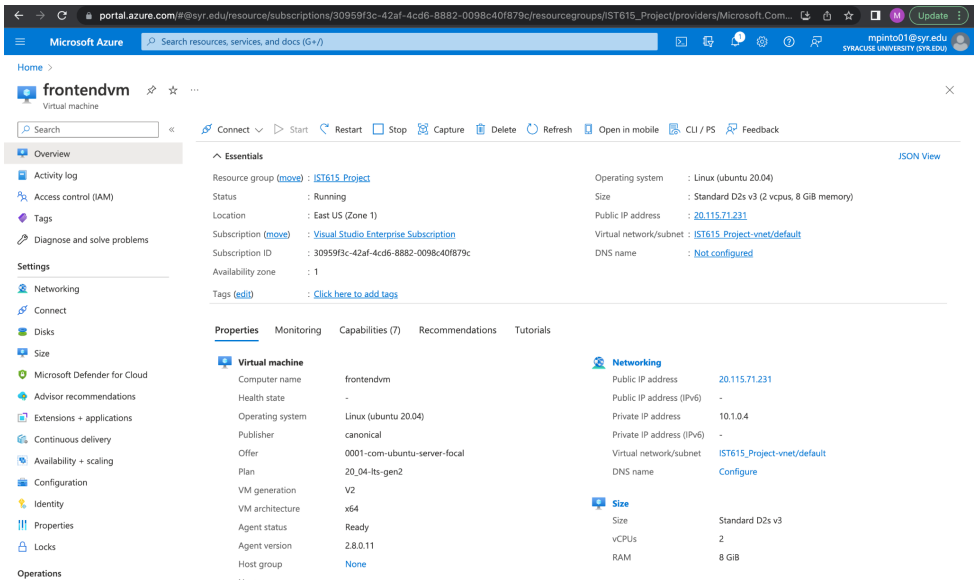
Screenshot 13: Webpage for users to enter their car details when putting it up for sale.

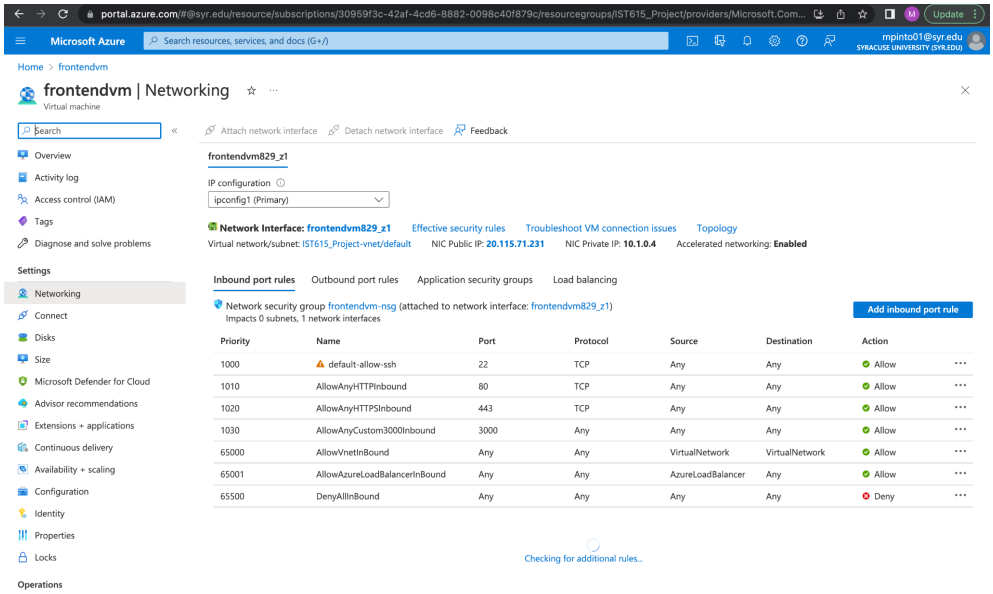


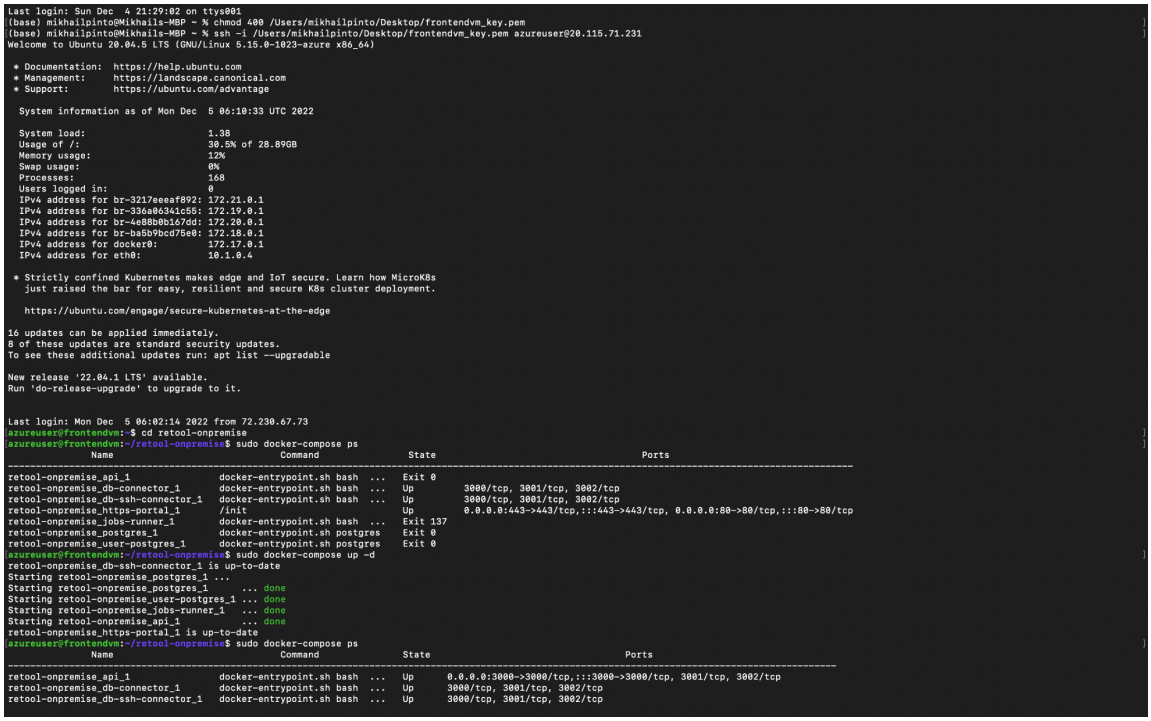
5. *Azure Virtual Machine*

In order to host this application/website a virtual machine was created in which we cloned a git docker environment which is responsible for hosting our website. This environment is created by retool themselves. A reference to this is me.ntioned in this link. As mentioned in the documentation we created the virtual machine and added the necessary security and network rules to it so that retool can communicate with Azure. Below screenshots show that we have connected to the virtual machine and check if the docker state is UP. The retool service is hosted on port 3000. To access it we need to put in our public ip address {{pulic_ip}}/3000.

Screenshot 14, 15 and 16: Azure Virtual Machine setup to host our website.





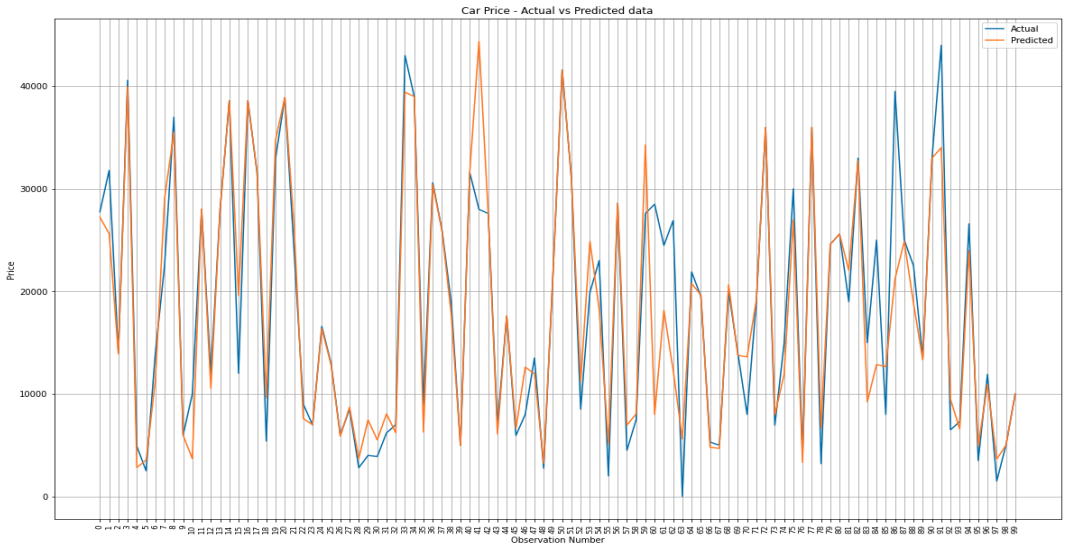


Once this is done we can go ahead and see the hosted website on the public_ip address and port 3000. The website can also be managed using the same VM.

6. Azure Databricks

This would be used for the purpose of python scripting, analysis and model creation in our project. We wrote a script to integrate Azure Blob storage and Azure Databricks. Using this script, we can read data from our container in blob storage to do further analysis on our dataset. Some additional data cleansing was done and new columns were added to add on to the features to build models .Three models - Linear Regression, Random Forest and XGBoost were built to predict estimated prices for users' cars that they need to post online for sale. Our best predicting model was Random Forest and it is being used to predict prices for these used cars. Predicted prices data is stored in csv format in another blob storage named ‘predictions.’

Screenshot 17: Graph for comparison of Random Forest Model price predictions and actual prices of cars.



Tasks Completed:

We were successfully able to implement the integration between Azure Blob Storage and SQL Database and create and host a website on a Virtual Machine. Finally we were also able to do some data analysis for our data and predict prices of cars using Azure Databricks.

Issues encountered:

- The biggest challenge that we faced was creating a front-end for our website as it is not in our skillset to do it. The initial plan to do it using App Services (Web App + Database) integration was dropped as it would take deep knowledge of .NET and some HTML and JavaScript. In order to tackle this issue we used an external service Retool https://retool.com/use-case/sql-front-end which was finally hosted on an Azure Virtual Machine.
- The second challenge that we faced was connecting Azure Databricks to Power BI to showcase our findings from our data and display our predicted prices. Due to unavailability of proper power bi licensing, we were unable to go around this issue or come up with a solution in time.
- Managing the cost for each service was extremely difficult due to the dataset being large and we had a lot of computes to handle.

Future Scope:

- After proper testing of our model built, include our predicted prices in the price estimator webpage of our website.
- Establish connection to PowerBI and build dashboards to display our data analysis, predicted prices and our model performance over time.