The Auto Shop

R1

The Auto Shop is going to be a web-based service which allows users to both post their cars for sale and browse through cars for purchase. The sample dataset for this project will be created by hand using common knowledge and a few sample descriptions and images. The real dataset will be created through a program (will be accessible through the Git) using either a generative text API or some form of random but controlled generation. The administrators will be the Technology Operations team at The Auto Shop (just Araad Shams in this case), and the users will be any individual who has a car to sell or who needs to buy a car.

In specifics, this application will provide a user-to-user car selling service. Before making the purchase, the buyer will be able to see a description about the car, the company which made the car, the model name of the car, the make year of the car, the price of the car, contact information for the seller, and the odometer reading of the car. Additionally, the user will be able to see other cars made by the same company, other cars made in the same year, other cars near the same price range, and ratings/reviews of the seller. Additionally, the buyer will be able to see cars within a certain radius of his location. The seller will also be able to edit and delete their posts. The buyer will be able to review the seller provided that they have purchased from the seller.

R2

The Auto Shop will be built a text-based interface. This is to make it so that **other people** who would like to make online ecommerce platforms can use our service, almost as an API, that they can query to see cars and perform various operations. A graphical interface was becoming a little too overwhelming for this project, so a text-based version with all the same features and components will be made. Created locally using Python. There will also be a back-end server management system with NodeJS running locally, which maintains the queries and allows us to use simple GET and POST methods.

R3

For the sample data, it will be manually hand entered and the plan of the attack for the real data that will populate the database is to create a program that, either through some Generative Text processing or through some randomization algorithm created, gives us appropriate entities and tuples that can work together and populate the database. They will be created according to the schema shown in the next section.

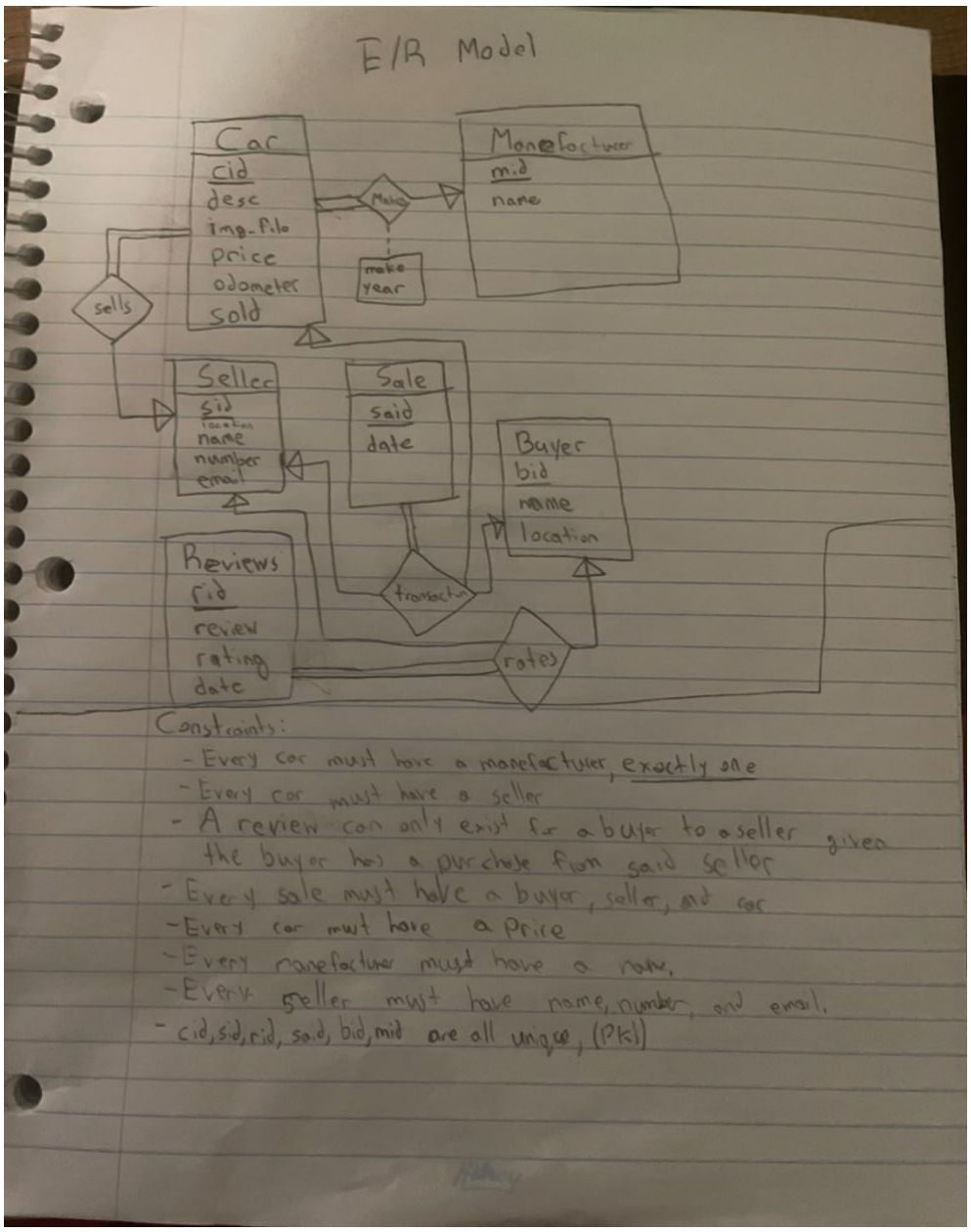
R4

For the production database, I will be using the following recommended dataset (https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data). The CSV file has been firstly, significantly reduced as the original file size would have slowed everything down. Next, we have kept the following rows.

1	cid	price	year	name	odometer	seller	sold	manufacturer
2	1	33590	2014	gmc sierra 1500 crew cab slt	57923	4	yes	SuperCar
3	2	22590	2010	chevrolet silverado 1500	71229	11	no	SuperCar
4	3	39590	2020	chevrolet silverado 1500 crew	19160	6	yes	SuperCar
5	4	30990	2017	toyota tundra double cab sr	41124	8	yes	SuperCar
6	5	15000	2013	ford f-150 xlt	128000	10	no	SuperCar
7	6	27990	2012	gmc sierra 2500 hd extended cab	68696	5	yes	SuperCar
8	7	34590	2016	chevrolet silverado 1500 double	29499	8	no	SuperCar
9	8	35000	2019	toyota tacoma	43000	14	no	SuperCar
10	9	29990	2016	chevrolet colorado extended cab	17302	12	yes	SuperCar
11	10	38590	2011	chevrolet corvette grand sport	30237	14	no	SuperCar
12	11	4500	1992	jeep cherokee	192000	6	yes	SuperCar
13	12	32990	2017	jeep wrangler unlimited sport	30041	15	no	SuperCar
14	13	24590	2017	chevrolet silverado 1500 regular	40784	12	no	SuperCar
15	14	30990	2016	chevrolet colorado crew cab z71	34940	6	yes	SuperCar
16	15	27990	2014	toyota tacoma access cab pickup	17805	9	yes	SuperCar
17	16	37990	2016	chevrolet camaro ss coupe 2d	9704	15	yes	SuperCar
18	17	33590	2014	toyota tundra crewmax sr5 pickup	55251	12	no	SuperCar
19	18	30990	2019	ford ranger supercrew xl pickup	1834	1	yes	SuperCar
20	19	27990	2018	nissan frontier crew cab pro-4x	37332	2	yes	SuperCar
21	20	0	2011	jeep compass	99615	9	no	SuperCar
22	21	34590	2018	ford f150 super cab xl pickup 4d	20856	9	yes	SuperCar
23	22	30590	2016	toyota tacoma double cab sr5	30176	4	no	SuperCar
24	23	32990	2020	jeep wrangler sport suv 2d	20581	9	no	SuperCar
25	24	38990	2020	ford f150 supercrew cab xlt	12231	14	yes	SuperCar
26	25	22590	2017	ram 1500 regular cab tradesman	39508	7	no	SuperCar
27	26	31590	2020	mazda mx-5 miata club	2195	10	yes	SuperCar
28	27	27990	2020	ford ranger supercab xl pickup	10688	10	no	SuperCar
29	28	31590	2019	cadillac xt4 sport suv 4d	12102	12	no	SuperCar
30	29	19900	2004	ford f250 super duty	88000	14	yes	SuperCar
31	30	16590	2016	jeep renegade sport suv 4d	35835	15	yes	SuperCar
32	31	26990	2016	ford f150 regular cab xl pickup	14230	8	yes	SuperCar
33	32	25590	2015	gmc sierra 1500 regular cab	35290	8	no	SuperCar
34	33	14000	2012	honda odyssey	95000	2	yes	SuperCar
35	34	28590	2018	ram 1500 quad cab express pickup	30047	6	yes	SuperCar
36	35	24590	2013	gmc sierra 1500 extended cab slt	80318	5	no	SuperCar
37	36	25990	2019	ram 1500 classic regular cab	12302	15	yes	SuperCar
38	37	34990	2018	ford mustang gt premium	18650	1	no	SuperCar
39	38	27990	2017	chevrolet colorado extended cab	22120	5	no	SuperCar
40	39	22500	2001	ford f450	144700	8	yes	SuperCar
41	40	32990	2019	chevrolet silverado 1500 ld	6897	8	no	SuperCar
42	41	31990	2013	toyota tundra double cab pickup	55068	1	yes	SuperCar
43	42	29990	2014	chevrolet silverado 1500 double	26129	4	no	SuperCar
44	43	23990	2016	chevrolet silverado 1500 regular	41568	3	no	SuperCar
45	44	22990	2012	toyota tacoma access cab pickup	37725	11	yes	SuperCar
46	45	26990	2014	chevrolet silverado 1500 crew	63129	6	yes	SuperCar
47	46	33990	2017	jeep wrangler unlimited sahara	34152	13	yes	SuperCar
48	47	15000	2017	dodge charger rt 4dr sedan	90000	6	no	SuperCar
49	48	35590	2020	honda civic si coupe 2d	9054	6	yes	SuperCar

This is for the cars table. Around 1700 entries from the given dataset are loaded in. For the other tables, some random values were generated which **correlate** properly to the cars table. They can all be found in the excel file attached to this submission. At the end, all of these lines were mapped to INSERT statements. (All visible in the excel file)

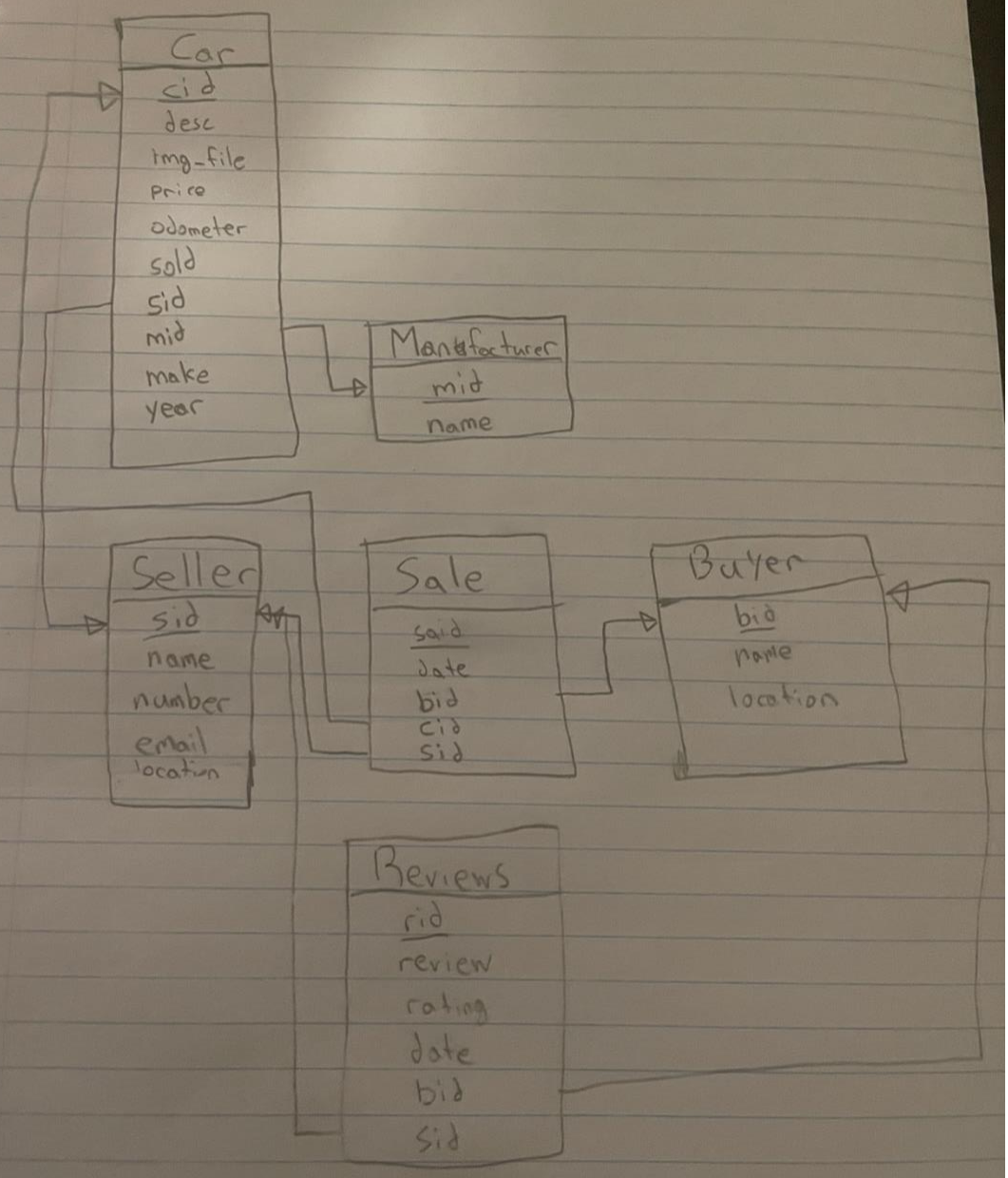
R5



Additional Constraints:

- Every review with a bid + sid pair must have a corresponding bid + sid pair in Sale

Relational Data Model



R6a – Interface

This feature will be a basic search filter. For instance, if a person who is looking for a car has certain requirements on the type of car he is looking for where it may be made in a certain year, have less than a certain number on the odometer, be made by a certain company, etc. In the main page, the user will have certain check boxes and text inputs below the search bar that let them choose the filters, and then upon hitting an apply button, the query will run and the page will update to show cars that meet their requirements.

R6b – SQL Template

CALL AutoShopDB.get_filtered_cars(2012, 10000000, "Atlas");

This uses the stored procedure defined as below:

```
PROCEDURE `get_filtered_cars`(IN make_year int, IN odometer_max int, IN make_wanted varchar(45))
BEGIN
   SELECT * from AutoShopDB.Car WHERE (make_year = 0 OR year >= make_year) AND (odometer_max = 0 OR odometer <=
odometer_max) AND (make_wanted = "" OR make = make_wanted);
END
```

R6b – Generated Output

cid	desc	img_file	price	odometer	sold	sid	mid	make	year
2	slow :(	img.png	$15.99	100	yes	1	2	Atlas	2012

R6b – Generated Output Text

```
# cid  desc  img_file      price  odometer   sold   sid     mid    make year
2      slow :(      img.png     $15.99100     yes    1       2      Atlas  2012
```

R6d – Testing with Production Database

```
Here is the result of your query:

ID: 100434
Make: 2021 SuperCar
Price: $99950
Sold: yes
Description: chevrolet corvette stingray spor


Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home
```

```
Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home

1
Please enter the filters for your search, or enter 0/Nothing for no specific filter
What make year are you looking for newer than: 2021
What is the maximum odometer reading you want: 1000
What is the name of the make you want:


Here is the result of your query:

ID: 100434
Make: 2021 SuperCar
Price: $99950
Sold: yes
Description: chevrolet corvette stingray spor


Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home
```

R6c – Implementation and Query Testing

```
call get_filtered_cars(2021, 1500, "")
```

34:1

ult Grid Filter Rows: Search Export:

cid	desc	img_file	price	odometer	sold	sid	mid	make	year
100175	toyota tacoma	nothing.png	$32186	1443	yes	15	1	SuperCar	2021
100228	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	9	1	SuperCar	2021
100318	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	10	1	SuperCar	2021
100359	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	9	1	SuperCar	2021
100421	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	no	9	1	SuperCar	2021
100434	chevrolet corvette stingray spor	nothing.png	$99950	510	yes	5	1	SuperCar	2021
100465	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	8	1	SuperCar	2021
100553	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	no	4	1	SuperCar	2021
100606	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	15	1	SuperCar	2021
100702	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	no	3	1	SuperCar	2021
100842	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	no	11	1	SuperCar	2021
100870	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	6	1	SuperCar	2021
101068	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	9	1	SuperCar	2021
101150	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	no	2	1	SuperCar	2021
101196	SPECIAL FINANCE PROGRAM 2020	nothing.png	$500	1400	yes	6	1	SuperCar	2021
101304	SPECIAL FINANCE PROGRAM 2021	nothing.png	$1000	1400	yes	10	1	SuperCar	2021
101388	SPECIAL FINANCE PROGRAM 2021	nothing.png	$1000	1400	no	8	1	SuperCar	2021
101487	SPECIAL FINANCE PROGRAM 2021	nothing.png	$1000	1400	no	9	1	SuperCar	2021
101524	SPECIAL FINANCE PROGRAM 2021	nothing.png	$1000	1400	yes	11	1	SuperCar	2021
101605	SPECIAL FINANCE PROGRAM 2021	nothing.png	$1000	1400	yes	15	1	SuperCar	2021

```
v1040-wn-rt-a-115-12:UserInterface araadshams$ python3 main.py
Welcome to The Auto Shop!
Please log in with your ID number: 3

Thank you! Please choose one of the below options (enter 1,2,3):

1. See cars available for purchase
2. View your cars for sale
3. View your order history
4. Close App

1


Please enter the filters for your search, or enter 0/Nothing for no specific filter
What make year are you looking for newer than: 2022
What is the maximum odometer reading you want:
What is the name of the make you want: Sienna


Here is the result of your query:

{'cid': 1, 'desc': 'A FAST CAR', 'img_file': 'img.png', 'price': '$15.99', 'odometer': 100, 'sold': 'not
', 'sid': 1, 'mid': 1, 'make': 'Sienna', 'year': 2022}


Would you like to:
1. Search again
2. Go back home
```

```
app.get("/carsFiltered", (req, res) => {
  console.log(req.query);
  const q =
    "call get_filtered_cars(" +
    req.query["make_year"] +
    ", " +
    req.query["odometer_max"] +
    ', "' +
    req.query["make_wanted"] +
    '");';
  db.query(q, (err, data) => {
    if (err) return res.json(err);
    return res.json(data);
  });
});
```

The implementation in the UI can be seen in the *src* folder, in the main python file!

R7a – Interface

This feature will be a basic reviews selection on the seller. Based on the car that the person is looking at, on the same page, reviews about the seller will be queried and shown with date and timestamps at the bottom of the page. The user's rating will also be shown next to their profile name. This will allow the buyer to make a more informed purchase and they will know how trustworthy the person they are dealing with is

R7b – SQL Template

CALL AutoShopDB.get_reviews(1);

This uses the stored procedure defined as below:

```
CREATE PROCEDURE `get_reviews` (IN car_id int)
BEGIN
SELECT * FROM Reviews WHERE sid = (SELECT sid FROM Car WHERE Car.cid = car_id);
END
```

R7b – Generated Output

rid	review	rating	date	bid	sid
1	My favourite person in the world!	5	2015-12-10 00:00:00	1	1
2	What an amazing selllerrrr	5	2023-02-28 03:19:09	1	1

R7b – Generated Output Text

rid review rating date bid sid
1 My favourite person in the world! 5 2015-12-10 00:00:00 1 1
2 What an amazing selllerrrr 5 2023-02-28 03:19:091 1

R7c - Implementation and Query Testing

```
1 •  call get_reviews(101633)
```

0% 25:1

Result Grid Filter Rows: Search Export:

rid	review	rating	date	bid	sid
13	The customer service is helpful and responsive.	5	2019-01-01 00:00:00	7	5

```
1 •  call get_reviews(100101)
```

00% 24:1

Result Grid Filter Rows: Search Export:

rid	review	rating	date	bid	sid
16	The site offers a secure checkout process.	4	1992-01-01 00:00:00	4	8
25	The site loads quickly and runs smoothly.	3	2011-01-01 00:00:00	5	8

R7d – Implementation and Testing

```
Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home

2


Please enter the ID of the car for the seller you'd like to see reviews for: 1


Here are the reviews for the seller of this car:

Rating: 5 stars
Date: 2015-12-10
Review: My favourite person in the world!

Rating: 5 stars
Date: 2023-02-28
Review: What an amazing selllerrrr

Rating: 5 stars
Date: 2023-03-26
Review: amazingggg



Would you like to:
```

```
app.get("/reviews", (req, res) => {
  console.log(req.query);
  const q = "call get_reviews(" + req.query["car_id"] + ");";
  db.query(q, (err, data) => {
    console.log(q);
    if (err) return res.json(err);
    return res.json(data);
  });
});
```

```
ID: 101633
Make: 2020 SuperCar
Price: $24958
Sold: no
Description: toyota camry

ID: 101652
Make: 2020 SuperCar
Price: $33590
Sold: no
Description: chevrolet silverado 1500 double

ID: 101654
Make: 2021 SuperCar
Price: $40900
Sold: yes
Description: toyota tacoma 4wd



Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home

2


Please enter the ID of the car for the seller you'd like to see reviews for: 101633


Here are the reviews for the seller of this car:

Rating: 5 stars
Date: 2019-01-01
Review: The customer service is helpful and responsive.



Would you like to:
1. Search again
2. See reviews on Sellers
3. Go back home
```

The implementation in the UI can be seen in the *src* folder, in the main python file!

R8a – Interface

This feature will allow a buyer to buy a car from a seller. Once the car has been bought, the buyer will then be allowed to comment and write a review about the seller by navigating to the seller's profile page and, once recognized that a valid transaction has been made between this buyer and seller, the buyer will be able to leave a review about the seller.

R8b – SQL Template

CALL AutoShopDB.add_comment("What an amazing selllerrrr", 4.9, 1, 1);

This uses the stored procedure defined as below:

```
CREATE PROCEDURE `add_comment` (IN review varchar(3000), IN rating decimal, IN buyer_id int, IN seller_id INT)
BEGIN
  INSERT INTO AutoShopDB.Reviews VALUES (NULL, review, rating, NOW(), buyer_id, seller_id);
END
```

R8b – Generated Output
N/A, Insert Statement (will show some confirmation on the website UI though)

R8b – Generated Output Text
N/A, Insert Statement (will show some confirmation on the website UI though)

```
call add_comment("this guy is so goood!!!!!!!", 4, 1, 1);
call get_reviews(2)
```

20:2

sult Grid Filter Rows: Search Export:

rid	review	rating	date	bid	sid
2	What an amazing selllerrrr	5	2023-02-28 03:19:09	1	1
5	this guy is so goood!!!!!!!	4	2023-03-28 12:35:25	1	1

R9a – Interface

This feature will allow users to only show sellers who are in a certain radius of their location. Using Long, Lat coordinates (determined at time of sign up), the buyer will be able to, on the main page, define a radius in which they want to search in, and then the SQL Query will remove all results that are outside of said radius.

R9b – SQL Template

CALL AutoShopDB.show_sellers_within(100, 1);

This uses the stored procedure defined as below:

```
CREATE PROCEDURE `show_sellers_within` (IN distance int, IN buyer_id int)
BEGIN
   SELECT * FROM Car WHERE ABS((SELECT location FROM Buyer WHERE buyer_id = Buyer.bid) - (SELECT
location FROM Seller WHERE Seller.sid = Car.sid)) <= distance;
END
```

R9b – Generated Output

1	A FAST CAR	img.png	$15.99	100	not	1	1	Sienna	2022
2	slow :(	img.png	$15.99	100	yes	1	2	Atlas	2012

R9b – Generated Output Text

cid desc img_file price odometer sold sid mid make year
1 A FAST CARimg.png $15.99100 not 1 1 Sienna 2022
2 slow :(img.png $15.99100 yes 1 2 Atlas 2012

R10a – Interface

This feature will allow users to modify comments that they have written for cars that they have purchased. Comments that they have written can be modified at any time and the comment will be flagged as edited.

R10b – SQL Template

```
CREATE PROCEDURE `modify_comment`(IN commentId int, IN commentStr varchar(3000))
BEGIN
```

```
    UPDATE AutoShopDB.Reviews
    SET review = commentStr
    WHERE rid = commentId;
END
```

R10b – Generated Output

```
3 •   CALL AutoShopDB.modify_comment(4, "YOOOOOO");
4 •   SELECT * FROM AutoShopDB.Reviews;
5
```

1:1

esult Grid Filter Rows: Search Edit: Export

rid	review	rating	date	bid	sid
1	My favourite person in the world!	5	2015-12-10 00:00:00	1	1
2	What an amazing selllerrrr	5	2023-02-28 03:19:09	1	1
3	amazingggg	5	2023-03-26 10:03:45	1	1
4	YOOOOOO	5	2023-03-26 10:13:46	1	1
NULL	NULL	NULL	NULL	NULL	NULL

R11a – Interface

This feature will allow users delete comments that they have written in the past. These comments will simply be deleted and no record of them will be kept anywhere in the database. A user can see their comments for a specific car and delete them when they would like to.

R11b – SQL Template

```
CREATE PROCEDURE `delete_comment` (IN commentId int)
BEGIN
   DELETE FROM AutoShopDB.Reviews
   WHERE rid = commentId;
END
```

R11b – Generated Output

```
CALL AutoShopDB.delete_comment(4);
SELECT * FROM AutoShopDB.Reviews;
```

100% 33:3

Result Grid Filter Rows: Search Edit: Export

rid	review	rating	date	bid	sid
1	My favourite person in the world!	5	2015-12-10 00:00:00	1	1
2	What an amazing selllerrrr	5	2023-02-28 03:19:09	1	1
3	amazingggg	5	2023-03-26 10:03:45	1	1
NULL	NULL	NULL	NULL	NULL	NULL

R11c - Implementation and Query Testing

rid	review	rating	date	bid	sid
16	The site offers a secure checkout process.	4	1992-01-01 00:00:00	4	8
25	The site loads quickly and runs smoothly.	3	2011-01-01 00:00:00	5	8

```
call delete_comment(16);

call get_reviews(100101);
```

00% 1:1

Result Grid Filter Rows: Search Export:

rid	review	rating	date	bid	sid
25	The site loads quickly and runs smoothly.	3	2011-01-01 00:00:00	5	8

R11d – Implementation and Testing

```
ID: 101578
Make: 2016 SuperCar
Price: $24900
Sold: yes
Description: ford econoline

ID: 101643
Make: 2013 SuperCar
Price: $0
Sold: no
Description: gmc sierra 2500 hd denali 4x4

ID: 101664
Make: 2009 SuperCar
Price: $3550
Sold: yes
Description: kia rio

ID: 101668
Make: 2013 SuperCar
Price: $24800
Sold: yes
Description: chevrolet avalance


Would you like to:
1. See, add, and modify reviews on Sellers
2. Go back home

1


Enter the ID of the car of the seller you want reviews on: 101664

Review ID: 28
Rating: 5 stars
Date: 2020-01-01
Review: The website is difficult to navigate.



Would you like to:
1. Add a review
2. Modify a review
3. Delete a review
4. Go Home
3


Enter the ID of the review you want to delete: 28
Deleted!
Would you like to:
1. Modify another review for the same car?
2. Modify another review for a different car?
3. Go home
```

The implementation in the UI can be seen in the *src* folder, in the main python file!

R17

So for this milestone, I kind of reworked the user interface side and switched to a text based python program with a backend NodeJS server which can essentially deal with the database and manage calls to the database.

Whenever it is accessed, the database is queried, and the appropriate JSON object is returned. Additionally, many new features were added and as per the requirements, 3 were implemented. The production database was also generated and tested. Those are the major changes for this milestone