

## **Documentation**

CSC 436 - Star Rental Project

Group 3: Taisa Yakimovich, Michael Martisius, Evan Ung, Nadia Sutyrina

### **1) Project idea**

**Car rental platform. Users can rent a wide selection of cars from multiple locations.**

### **2) User requirements (Basic operations and questions)**

- 1. User registration and login**
- 2. Browse cars**
- 3. Renting a vehicle**
- 4. Payment processing**

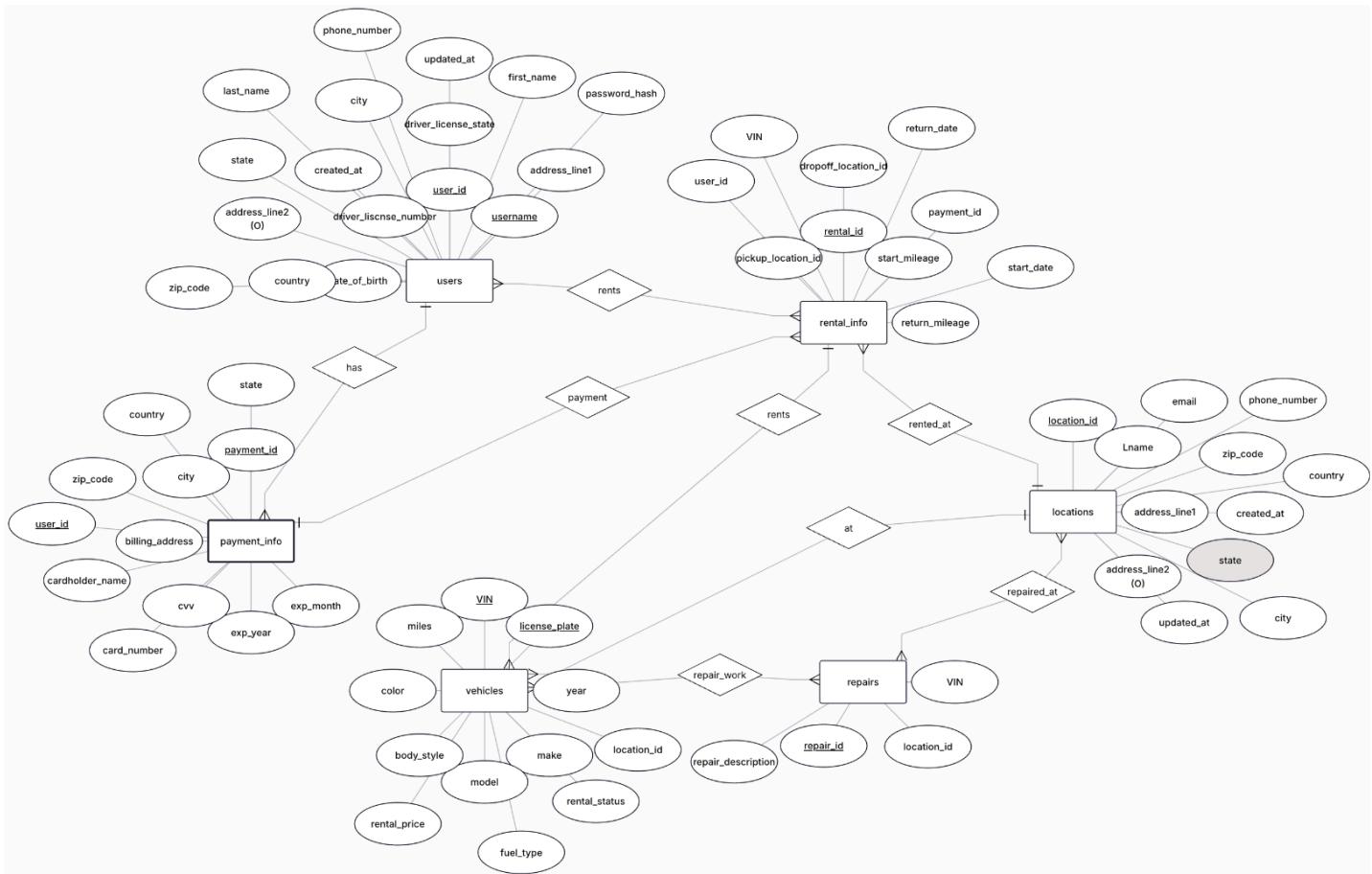
**Example questions that can be answered through the database:**

- How many cars are currently in inventory?**
- What cars are available for rent?**
- How much do cars cost to rent?**
- What cars have been rented?**
- What types of cars can be rented?**
- What years are available?**

### 3) ERD with Database tables, relationships, constraints etc

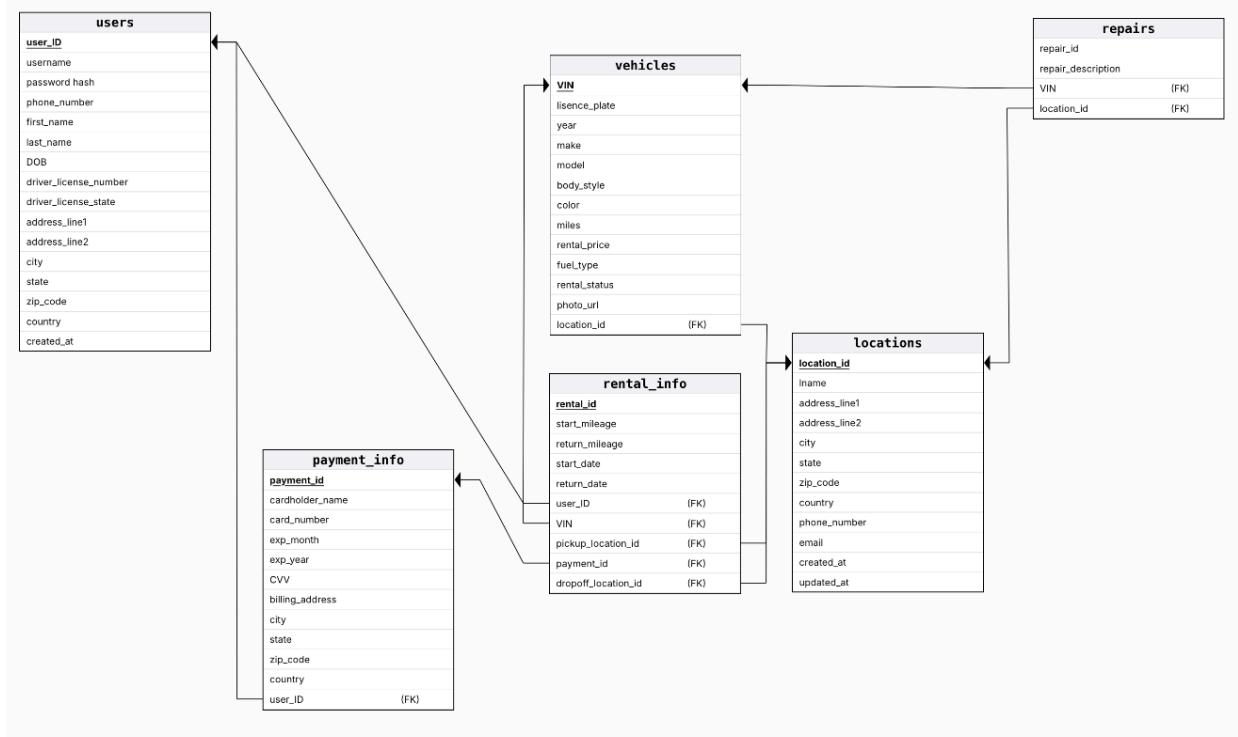
ERD Image Download:

[https://drive.google.com/file/d/1DqjVGisfn0WhC6G2swN\\_8KJI4p8u6ejQ/view?usp=sharing](https://drive.google.com/file/d/1DqjVGisfn0WhC6G2swN_8KJI4p8u6ejQ/view?usp=sharing)



## Relational Schema Download:

[https://drive.google.com/file/d/15SvwOvHw52h1cNDq7j89JD52jmL1VoCV/view?usp=drive\\_link](https://drive.google.com/file/d/15SvwOvHw52h1cNDq7j89JD52jmL1VoCV/view?usp=drive_link)



## 4) Data sources, if exists

**Data was AI generated using ChatGPT. Images sourced online using google images for copyright free images.**

## 5) SQL queries and sample results for each, (cover different types of queries DDL, DML, DCL,)

### DDL:

Create sql queries we used to create our database and tables in it.

```
CREATE SCHEMA Star_Rentals;
USE Star_Rentals;

-- =====
-- TABLE DECLARATIONS
-- =====
-- 1. LOCATIONS
CREATE TABLE locations (
    location_id INT PRIMARY KEY,
    Lname CHAR(100),
    address_line1 VARCHAR(100),
    address_line2 VARCHAR(100),
    city VARCHAR(50),
    state VARCHAR(30),
    zip_code INT,
    country VARCHAR(20),
    phone_number VARCHAR(15),
    email VARCHAR(100),
    created_at TIMESTAMP,
    updated_at TIMESTAMP
);

2 14:42:30 CREATE SCHEMA Star_Rentals          1 row(s) affected
3 14:42:30 USE Star_Rentals                   0 row(s) affected
4 14:43:07 CREATE TABLE locations ( location_id INT PRIMARY KEY, Lname CHAR(100), address_line1 V... 0 row(s) affected
5 14:43:07 CREATE TABLE users ( user_id INT PRIMARY KEY, username VARCHAR(100), password_hash... 0 row(s) affected
6 14:43:07 CREATE TABLE payment_info ( payment_id INT PRIMARY KEY, cardholder_name VARCHAR(100... 0 row(s) affected
```

Alter queries we used to assign Foreign Keys to the corresponding tables.

```
-- VEHICLES + LOCATIONS
ALTER TABLE vehicles
ADD CONSTRAINT fk_vehicle_location
FOREIGN KEY (location_id)
REFERENCES locations(location_id) ON DELETE RESTRICT ON UPDATE CASCADE;

-- PAYMENT_INFO + USERS
ALTER TABLE payment_info
ADD CONSTRAINT fk_payment_user
FOREIGN KEY (user_id)
REFERENCES users(user_id) ON DELETE RESTRICT ON UPDATE CASCADE;

-- RENTAL_INFO + VEHICLES
ALTER TABLE rental_info
ADD CONSTRAINT fk_rental_vehicle
FOREIGN KEY (VIN)
REFERENCES vehicles(VIN) ON DELETE RESTRICT ON UPDATE CASCADE;
```

35	14:50:03	ALTER TABLE vehicles ADD CONSTRAINT fk_vehicle_location FOREIGN KEY (location_id) REFEREN...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
36	14:50:03	ALTER TABLE payment_info ADD CONSTRAINT fk_payment_user FOREIGN KEY (user_id) REFEREN...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
37	14:50:03	ALTER TABLE rental_info ADD CONSTRAINT fk_rental_vehicle FOREIGN KEY (VIN) REFERENCES v...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
38	14:50:03	ALTER TABLE rental_info ADD CONSTRAINT fk_rental_user FOREIGN KEY (user_id) REFERENCES u...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
39	14:50:03	ALTER TABLE rental_info ADD CONSTRAINT fk_rental_pickup_location FOREIGN KEY (pickup_locati...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
40	14:50:03	ALTER TABLE rental_info ADD CONSTRAINT fk_rental_dropoff_location FOREIGN KEY (dropoff_locati...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0
41	14:50:03	ALTER TABLE rental_info ADD CONSTRAINT fk_rental_payment FOREIGN KEY (payment_id) REFEREN...	0 row(s) affected	Records: 0	Duplicates: 0	Warnings: 0

Drop query:

Drop query:		
1	•	DROP DATABASE IF EXISTS Star_Rentals;
2		
3	•	CREATE SCHEMA Star_Rentals;
4	•	USE Star_Rentals;
5		
26	14:50:03	DROP DATABASE IF EXISTS Star_Rentals
27	14:50:03	CREATE SCHEMA Star_Rentals
28	14:50:03	USE Star_Rentals

DML:

We used Insert query to insert generated data into all our tables.

```
2 -- 1. LOCATIONS
3 • INSERT INTO locations VALUES
4 (1, 'Downtown Branch', '123 Main St', NULL, 'Providence', 'RI', 02903, 'USA', '401-555-1000', 'downtown@rentalco.com', NOW(), NOW(),
5 (2, 'Airport Branch', '1 Airport Rd', 'Terminal A', 'Warwick', 'RI', 02886, 'USA', '401-555-2000', 'airport@rentalco.com', NOW(), NOW(),
5 (3, 'East Side Branch', '88 Thayer St', NULL, 'Providence', 'RI', 02906, 'USA', '401-555-3000', 'eastside@rentalco.com', NOW(), NOW(),
7 (4, 'Newport Branch', '45 Ocean Ave', NULL, 'Newport', 'RI', 02840, 'USA', '401-555-4000', 'newport@rentalco.com', NOW(), NOW()),
3 (5, 'Boston Branch', '500 Beacon St', NULL, 'Boston', 'MA', 02115, 'USA', '617-555-5000', 'boston@rentalco.com', NOW(), NOW());
```

## Update

```
220 -- 1) Vehicle returned: mark the Ford F-150 as Available and sync mileage from rental #2
221 • SELECT * from VEHICLES;
222 • UPDATE vehicles
223     SET rental_status = 'Available',
224         miles = 22550
225     WHERE VIN = '1FTFW1EF1EFA23456';
226 • SELECT * from VEHICLES;
```

```
51 -- 5) Refresh card expiration for payment #3 (Bob Wilson)
52 • SELECT * from payment_info;
53 • UPDATE payment_info
54 SET exp_month = '10',
55     exp_year  = '2026'
56 WHERE payment_id = 3;
57 • SELECT * from payment_info;
```

## Delete

```
2      -- 1) Remove a stored payment method
3 •  DELETE FROM payment_info
4 WHERE payment_id = 2;
5
6 •  Select * from payment_info;
```

Result Grid												
payment_id	cardholder_name	card_number	exp_month	exp_year	cvv	billing_address	city	state	zip_code	country	user_id	
1	John Doe	4111111111111111	05	2027	123	12 Elm St	Providence	RI	02903	USA	1	
3	Bob Wilson	378282246310005	10	2026	789	44 Oak Ave	Newport	RI	02840	USA	3	
4	Carol Johnson	6011000990139424	03	2028	234	777 Hope St	Providence	RI	02906	USA	4	
HULL	HULL	HULL	NULL	NULL	NULL	HULL	HULL	NULL	NULL	HULL	NULL	

## DQL:

### Right Join

```
416  /* RIGHT JOIN
417      Return every rental and the matching customer name when available.
418 */
419 •  SELECT
420     r.rental_id,
421     r.user_id,
422     CONCAT(u.first_name, ' ', u.last_name) AS customer_name,
423     r.start_date,
424     r.return_date
425   FROM users u
426   RIGHT JOIN rental_info r ON u.user_id = r.user_id
427   ORDER BY r.rental_id;
```

Result Grid					
	rental_id	user_id	customer_name	start_date	return_date
▶	1	1	John Doe	2025-07-01	2025-07-05
	2	2	Alice Smith	2025-06-10	2025-06-15
	3	3	Bob Wilson	2025-08-20	2025-08-23
	4	4	Carol Johnson	2025-05-10	2025-05-12
	5	5	David Lee	2025-09-02	2025-09-07

## Average

```
290      -- Average miles between all cars
291 •  Select AVG(miles) as Average_miles
292     From vehicles;
```

Result Grid		Filter Rows:	Export:
Average_miles			
▶ 23510.0000			

## Join + count

```
294      -- How many cars are currently in inventory per location?
295 •  SELECT
296        l.Lname AS location_name,
297        COUNT(v.VIN) AS vehicle_count
298      FROM
299        vehicles v
300      JOIN
301        locations l ON v.location_id = l.location_id
302      GROUP BY
303        l.location_id, l.Lname;
304
305      -- Find which vehicles are located in which branch
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
location_name	vehicle_count			
Downtown Branch	1			
Airport Branch	1			
East Side Branch	1			
Newport Branch	1			
Boston Branch	1			

```

305      -- Find which vehicles are located in which branch.
306 •   SELECT
307      v.make,
308      v.model,
309      v.license_plate,
310      v.color,
311      l.Lname AS location_name
312  FROM vehicles v
313  JOIN locations l ON v.location_id = l.location_id;
314
315  -- TO DO
316  -- 1.Index on rentals table
317  -- ...

```

Result Grid | Filter Rows: Export: Wrap Cell Content

	make	model	license_plate	color	location_name
▶	Jeep	Grand Cherokee	RI9012	Black	East Side Branch
	Ford	F-150	RI5678	Red	Airport Branch
	Toyota	Camry	RI1234	Blue	Downtown Branch
	Tesla	Model 3	MA7890	Silver	Boston Branch
	Hyundai	Sonata	RI3456	White	Newport Branch

## Natural Join

```

370  /* NATURAL JOIN:
371      Repairs and vehicles both share columns VIN and location_id.
372      Which vehicles are currently linked to repair records, with price and description?
373 */
374 •   SELECT
375      r.repair_id,
376      v.VIN,
377      v.make,
378      v.model,
379      v.rental_price,
380      r.repair_description
381  FROM repairs r NATURAL JOIN vehicles v
382  ORDER BY v.make, v.model;

```

Result Grid | Filter Rows: Export: Wrap Cell Content

	repair_id	VIN	make	model	rental_price	repair_description
▶	4	1FTFW1EF1EFA23456	Ford	F-150	59.99	Windshield replacement
	1	5NPE24AF7FH123456	Hyundai	Sonata	44.99	Oil change and tire rotation
	5	1C4RJFAG1FC123456	Jeep	Grand Cherokee	69.99	Engine diagnostic
	3	3FA6P0LU0HR123456	Tesla	Model 3	94.99	Battery replacement
	2	1HGCM82633A123456	Toyota	Camry	49.99	Brake pad replacement

## Inner Join

```
384  /* INNER JOIN:  
385      List every rental with the customer name and vehicle details.  
386  */  
387 •  SELECT  
388      r.rental_id,  
389      u.first_name,  
390      u.last_name,  
391      v.make,  
392      v.model,  
393      r.start_date,  
394      r.return_date  
395  FROM rental_info r  
396  INNER JOIN users u ON u.user_id = r.user_id  
397  INNER JOIN vehicles v ON v.VIN      = r.VIN  
398  ORDER BY r.start_date DESC;  
399
```

	rental_id	first_name	last_name	make	model	start_date	return_date
▶	5	David	Lee	Tesla	Model 3	2025-09-02	2025-09-07
	3	Bob	Wilson	Jeep	Grand Cherokee	2025-08-20	2025-08-23
	1	John	Doe	Toyota	Camry	2025-07-01	2025-07-05
	2	Alice	Smith	Ford	F-150	2025-06-10	2025-06-15
	4	Carol	Johnson	Hyundai	Sonata	2025-05-10	2025-05-12

## Left Join

```
400  /* LEFT JOIN:  
401      What's the latest rental (if any) for each vehicle?  
402  */  
403 •  SELECT  
404      v.VIN,  
405      v.make,  
406      v.model,  
407      lr.last_start AS last_rental_start  
408  FROM vehicles v  
409  LEFT JOIN (|  
410      SELECT VIN, MAX(start_date) AS last_start  
411      FROM rental_info  
412      GROUP BY VIN  
413  ) AS lr ON lr.VIN = v.VIN  
414  ORDER BY (lr.last_start IS NULL), lr.last_start DESC;  
415
```

	VIN	make	model	last_rental_start
▶	3FA6P0LU0HR123456	Tesla	Model 3	2025-09-02
	1C4RJFAG1FC123456	Jeep	Grand Cherokee	2025-08-20
	1HGCM82633A123456	Toyota	Camry	2025-07-01
	1FTFW1EF1EFA23456	Ford	F-150	2025-06-10
	5NPE24AF7FH123456	Hyundai	Sonata	2025-05-10

## Right Join

```
416  /* RIGHT JOIN
417      Return every rental and the matching customer name when available.
418 */
419 • SELECT
420     r.rental_id,
421     r.user_id,
422     CONCAT(u.first_name, ' ', u.last_name) AS customer_name,
423     r.start_date,
424     r.return_date
425 FROM users u
426 RIGHT JOIN rental_info r ON u.user_id = r.user_id
427 ORDER BY r.rental_id;
428 |
```

Result Grid					
	rental_id	user_id	customer_name	start_date	return_date
▶	1	1	John Doe	2025-07-01	2025-07-05
	2	2	Alice Smith	2025-06-10	2025-06-15
	3	3	Bob Wilson	2025-08-20	2025-08-23
	4	4	Carol Johnson	2025-05-10	2025-05-12
	5	5	David Lee	2025-09-02	2025-09-07

## Sum, Joins, Group By, Having, Order By

```
446  /* Which pickup locations and (year, month) pairs generated > $100 in estimated revenue? */
447 • SELECT
448     l.Lname AS location_name,
449     YEAR(r.start_date) AS yr,
450     MONTH(r.start_date) AS mo,
451     SUM(DATEDIFF(COALESCE(r.return_date, CURDATE()), r.start_date) * v.rental_price) AS revenue_est
452 FROM rental_info r
453 JOIN vehicles v ON v.VIN = r.VIN
454 JOIN locations l ON l.location_id = r.pickup_location_id
455 GROUP BY
456     l.location_id, l.Lname,
457     YEAR(r.start_date), MONTH(r.start_date)
458 HAVING
459     SUM(DATEDIFF(COALESCE(r.return_date, CURDATE()), r.start_date) * v.rental_price) > 100
460 ORDER BY
461     yr, mo, revenue_est DESC;
```

Result Grid				
	location_name	yr	mo	revenue_est
▶	Airport Branch	2025	6	299.950008392334
	Downtown Branch	2025	7	199.9600067138672
	East Side Branch	2025	8	209.9699935913086
	Boston Branch	2025	9	474.94998931884766

## Range expression

```
477  *  
478   Average distance per rental, by vehicle, filtered  
479     Which vehicles average more than 200 miles per rental?  
480 */  
481 • SELECT  
482   v.VIN,  
483   v.make,  
484   v.model,  
485   AVG(GREATEST(r.return_mileage - r.start_mileage, 0)) AS avg_miles_per_rental  
486 FROM vehicles v  
487 JOIN rental_info r ON r.VIN = v.VIN  
488 GROUP BY v.VIN, v.make, v.model  
489 HAVING avg_miles_per_rental > 200  
490 ORDER BY avg_miles_per_rental DESC;
```

Result Grid			
VIN	make	model	avg_miles_per_rental
1FTFW1EF1EFA23456	Ford	F-150	550.0000
1HGCM82633A123456	Toyota	Camry	500.0000
5NPE24AF7FH123456	Hyundai	Sonata	320.0000
3FA6P0LU0HR123456	Tesla	Model 3	280.0000
1C4RJFAG1FC123456	Jeep	Grand Cherokee	250.0000

## Count

```
492  *  
493   How many vehicles of each status are at every location?  
494 */  
495 • SELECT  
496   l.Lname AS location_name,  
497   v.rental_status,  
498   COUNT(*) AS vehicle_count  
499 FROM vehicles v  
500 JOIN locations l ON l.location_id = v.location_id  
501 GROUP BY l.Lname, v.rental_status  
502 ORDER BY l.Lname, vehicle_count DESC;  
503
```

Result Grid		
location_name	rental_status	vehicle_count
Airport Branch	Available	1
Boston Branch	Available	1
Downtown Branch	Available	1
East Side Branch	Available	1
Newport Branch	Maintenance	1

DCL:

Grant All Privileges to a SuperUser

```
504  *  
505      Grant all privileges to a super user  
506  */  
507 • CREATE USER 'star_admin'@'localhost' IDENTIFIED BY 'StrongAdminPassword!';  
508  
509      -- Give full privileges on this schema  
510 • GRANT ALL PRIVILEGES  
511      ON Star_Rentals.*  
512      TO 'star_admin'@'localhost'  
513      WITH GRANT OPTION;  
  
✓ 149 18:21:36 CREATE USER 'star_admin'@localhost' IDENTIFIED BY 'StrongAdminPassword!' 0  
✓ 150 18:21:36 GRANT ALL PRIVILEGES ON Star_Rentals.* TO 'star_admin'@localhost' WITH GRANT OPTION 0
```

Create a new user and grant different types of privileges to them

```
515  *  
516      Create another user with privileges.  
517      Grant, Revoke privileges  
518      Drop the user  
519  */  
520  
521 • CREATE USER 'star_student'@'localhost' IDENTIFIED BY 'StudentPass2025!';  
522      -- Allow reading data from ALL tables in the DB  
523 • GRANT SELECT ON Star_Rentals.* TO 'star_student'@'localhost';  
524  
525      -- Allow modifying only the rental and repair related data  
526 • GRANT INSERT, UPDATE ON Star_Rentals.rental_info TO 'star_student'@'localhost';  
527 • GRANT INSERT, UPDATE, DELETE ON Star_Rentals.repairs TO 'star_student'@'localhost';  
528  
529      -- Allow to create views, but NOT tables or structure changes  
530 • GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@'localhost';
```

Output

Action Output

#	Time	Action	Message
✓ 148	18:15:16	SELECT r.rental_id, r.user_id, CONCAT(u.first_name, " ", u.last_name) AS customer_name, r.sta...	5 row(s) returned
✓ 149	18:21:36	CREATE USER 'star_admin'@localhost' IDENTIFIED BY 'StrongAdminPassword!'	0 row(s) affected
✓ 150	18:21:36	GRANT ALL PRIVILEGES ON Star_Rentals.* TO 'star_admin'@localhost' WITH GRANT OPTION	0 row(s) affected
✓ 151	18:26:50	CREATE USER 'star_student'@localhost' IDENTIFIED BY 'StudentPass2025!'	0 row(s) affected
✓ 152	18:27:11	GRANT SELECT ON Star_Rentals.* TO 'star_student'@localhost'	0 row(s) affected
✓ 153	18:27:11	GRANT INSERT, UPDATE ON Star_Rentals.rental_info TO 'star_student'@localhost'	0 row(s) affected
✓ 154	18:27:11	GRANT INSERT, UPDATE, DELETE ON Star_Rentals.repairs TO 'star_student'@localhost'	0 row(s) affected
✓ 155	18:27:11	GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@localhost'	0 row(s) affected

## Revoke

```
532      -- Revoke delete permission on repairs, but keep insert/update
533 •  REVOKE DELETE ON Star_Rentals.repairs FROM 'star_student'@'localhost';
534
535
```

Output

#	Time	Action
163	18:31:28	GRANT INSERT, UPDATE ON Star_Rentals.rental_info TO 'star_student'@'localhost'
164	18:31:28	GRANT INSERT, UPDATE, DELETE ON Star_Rentals.repairs TO 'star_student'@'localhost'
165	18:31:28	GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@'localhost'
166	18:31:28	REVOKE DELETE ON Star_Rentals.repairs FROM 'star_student'@'localhost'
167	18:31:32	GRANT INSERT, UPDATE ON Star_Rentals.rental_info TO 'star_student'@'localhost'
168	18:31:32	GRANT INSERT, UPDATE, DELETE ON Star_Rentals.repairs TO 'star_student'@'localhost'
169	18:31:32	GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@'localhost'
170	18:31:32	REVOKE DELETE ON Star_Rentals.repairs FROM 'star_student'@'localhost'

## Drop User

```
535 •  REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'star_student'@'localhost';
536
537 •  DROP USER 'star_student'@'localhost';
```

Output

#	Time	Action
165	18:31:28	GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@'localhost'
166	18:31:28	REVOKE DELETE ON Star_Rentals.repairs FROM 'star_student'@'localhost'
167	18:31:32	GRANT INSERT, UPDATE ON Star_Rentals.rental_info TO 'star_student'@'localhost'
168	18:31:32	GRANT INSERT, UPDATE, DELETE ON Star_Rentals.repairs TO 'star_student'@'localhost'
169	18:31:32	GRANT CREATE VIEW, SHOW VIEW ON Star_Rentals.* TO 'star_student'@'localhost'
170	18:31:32	REVOKE DELETE ON Star_Rentals.repairs FROM 'star_student'@'localhost'
171	18:32:59	REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'star_student'@'localhost'
172	18:33:16	DROP USER 'star_student'@'localhost'

## 6) Indexes

```
payment_id
  ▼ Indexes
    □ PRIMARY
    □ fk_rental_dropoff_location
    □ idx_rentalinfo_vin
    □ idx_rentalinfo_user
    □ idx_rentalinfo_dates
    □ idx_rentalinfo_locations
    □ idx_rentalinfo_payment
  319      -----
  320 •  CREATE INDEX idx_rentalinfo_vin
  321      ON rental_info (VIN);
  322 •  CREATE INDEX idx_rentalinfo_user
  323 •  CREATE INDEX idx_rentalinfo_dates
  324 •  CREATE INDEX idx_rentalinfo_locations
  325 •  CREATE INDEX idx_rentalinfo_payment */
```

## 7) Views

```
354     How heavily is each vehicle used (rental count, miles driven, average rental length)?  
355  */  
356 • CREATE OR REPLACE VIEW v_vehicle_utilization AS  
357   SELECT  
358     v.VIN,  
359     v.make,  
360     v.model,  
361     COUNT(r.rental_id) AS rental_count,  
362     SUM(GREATEST(r.return_mileage - r.start_mileage, 0)) AS miles_driven,  
363     AVG(DATEDIFF(r.return_date, r.start_date)) AS avg_days_per_rental  
364   FROM vehicles v  
365   LEFT JOIN rental_info r ON r.VIN = v.VIN  
366   GROUP BY v.VIN, v.make, v.model;  
367  
368 • Select * from v_vehicle_utilization;  
369  
370 /* -----
```

Result Grid					
VIN	make	model	rental_count	miles_driven	avg_days_per_rental
1C4RJFAG1FC123456	Jeep	Grand Cherokee	1	250	3.0000
1FTFW1EF1EFA23456	Ford	F-150	1	550	5.0000
1HGM82633A123456	Toyota	Camry	1	500	4.0000
3FA6P0LU0HR123456	Tesla	Model 3	1	280	5.0000
SNPE24AF7FH123456	Hyundai	Sonata	1	320	2.0000

## 8) Description of the Technology used for your interface

Implementation.

**For our interface, we used Angular and Figma to design and create our frontend interface. Requests are made to the backend using HTTP.**

**For our backend, we used Python with fastapi and unicorn to create and host the API, and databases package to send and receive queries to the MySql database.**

## 9) Screenshots of the GUI to explain different scenarios.

### 1. User login

After opening the website, you will find a button on the top right to sign in:

The screenshot shows a web-based car rental search interface. At the top, there are dropdown menus for 'Pick-up' and 'Drop-off' locations, a date range selector, and a 'Search' button. Below these are five search results for different vehicles:

- 2022 Jeep Grand Cherokee**: \$69.99 / Day. Type: SUV, Fuel: Gasoline, Location: East Side Branch, Providence, RI. Includes a 'Reserve' button.
- 2021 Ford F-150**: \$59.99 / Day. Type: Truck, Fuel: Gasoline, Location: Airport Branch, Warwick, RI. Includes a 'Reserve' button.
- 2020 Chevrolet Malibu**: \$45.99 / Day. Type: Sedan, Fuel: Gasoline, Location: Cranston Branch, Cranston, RI. Includes a 'Reserve' button.
- 2020 Toyota Camry**: \$49.99 / Day. Type: Sedan, Fuel: Gasoline, Location: Dovetailed Branch, Providence, RI. Includes a 'Reserve' button.
- 2022 Nissan Altima**: \$47.99 / Day. Type: Sedan, Fuel: Gasoline, Location: Airport Branch, Warwick, RI. Includes a 'Reserve' button.

On the right side of the search results, there are filters for 'Car Type' (checkboxes for SUV, Truck, Sedan), 'Price Range' (dropdown), 'Location' (dropdown), and 'Transmission' (dropdown). A 'Apply Filters' button is also present.

After clicking the button where you will be prompted to enter your existing account credentials or to create a new account:

The screenshot shows a modal window titled 'Star Rentals' containing a sign-in form. The form includes fields for 'Username' (with 'jdoe' entered) and 'Password' (with '\*\*\*\*\*' entered). Below the password field is a visibility toggle icon. A 'Sign In' button is located at the bottom of the form. At the very bottom of the modal, there is a link 'Don't have an account? Sign Up'.

After a successful sing in, the top right corner will display a message “Welcome, [username]”:

Screenshot of the Star Rentals Car Rental Store Page showing a list of available cars. The top right corner displays "Welcome, jdoe!".

Car Model	Type	Location	Rental Price	Action
2022 Jeep Grand Cherokee	SUV	East Side Branch, Providence, RI	\$69.99 / Day	Reserve
2021 Ford F-150	Truck	Airport Branch, Warwick, RI	\$59.99 / Day	Reserve
2020 Chevrolet Malibu	Sedan	Cranston Branch, Cranston, RI	\$45.99 / Day	Reserve
2020 Toyota Camry	Sedan	Downtown Branch, Providence, RI	\$49.99 / Day	Reserve
2022 Nissan Altima	Sedan	Airport Branch, Warwick, RI	\$47.99 / Day	Reserve
2023 Tesla Model 3			\$94.99 / Day	

Filter sidebar on the right:

- Car Type:  SUV,  Truck,  Sedan
- Price Range
- Location
- Transmission
- Apply filters**

## 2. Browse cars

Screenshot of the Star Rentals Car Rental Store Page showing a list of available cars. The top right corner displays "Sign In".

Car Model	Type	Location	Rental Price	Action
2022 Jeep Grand Cherokee	SUV	East Side Branch, Providence, RI	\$69.99 / Day	Reserve
2021 Ford F-150	Truck	Airport Branch, Warwick, RI	\$59.99 / Day	Reserve
2020 Chevrolet Malibu	Sedan	Cranston Branch, Cranston, RI	\$45.99 / Day	Reserve
2020 Toyota Camry	Sedan	Downtown Branch, Providence, RI	\$49.99 / Day	Reserve
2022 Nissan Altima	Sedan	Airport Branch, Warwick, RI	\$47.99 / Day	Reserve
2023 Tesla Model 3			\$94.99 / Day	

Filter sidebar on the right:

- Car Type:  SUV,  Truck,  Sedan
- Price Range
- Location
- Transmission
- Apply filters**

To browse cars, start at the home page. On the right you will find all the filters, including:

Car Type

- SUV
- Truck
- Sedan

Price Range

Location

Transmission

**Apply filters**

Car Type

Price Range

Location ↴

Transmission

**Apply filters**

Car Type

Price Range

Location ↴

- Downtown Branch
- Airport Branch
- East Side Branch
- Newport Branch
- Boston Branch
- Cranston Branch

Transmission

**Apply filters**

Car Type

Price Range

Location

Transmission

Gasoline
  Electric

Apply filters

### 3. Renting a vehicle

To rent a vehicle, on the top of the home page you can select the desired pick up and drop off location and date range, then apply filters:

Pick up
Downtown Branch
Drop off
Airport Branch
Date Range
2025.12.01 - 2025.12.06
Search

2022 Jeep Grand Cherokee

SUV  
 Gasoline  
 East Side Branch, Providence, RI

\$69.99 / Day
Reserve

2021 Ford F-150

Truck  
 Gasoline  
 Airport Branch, Warwick, RI

\$59.99 / Day
Reserve

2020 Chevrolet Malibu

Sedan  
 Gasoline  
 Cranston Branch, Cranston, RI

\$45.99 / Day
Reserve

2020 Toyota Camry

Sedan  
 Gasoline  
 Downtown Branch, Providence, RI

\$49.99 / Day
Reserve

2022 Nissan Altima

Sedan  
 Gasoline  
 Airport Branch, Warwick, RI

\$47.99 / Day
Reserve

2023 Tesla Model 3

\$94.99 / Day

Car Type

Price Range

Location

Transmission

Gasoline
  Electric

Apply filters

After picking everything you need, click “Search” and you will see all the cars that fit you parameters:

The screenshot shows the Star Rentals website interface. At the top, there are navigation links: "Star Rentals", "Car Rental Store Page", "My Rentals", "Register Info", and "Car Database". On the right, it says "Sign Out" and "Welcome, jdoe!". Below the header, there are search filters: "Pick-up" (set to "Downtown Branch"), "Drop-off" (set to "Airport Branch"), "Date Range" (set to "2025.12.01 - 2025.12.06"), and a "Search" button. To the right of the search bar is a sidebar with dropdown menus for "Car Type", "Price Range", "Location", and "Transmission", along with a "Apply filters" button. The main content area displays a single car listing: "2022 Mazda CX-5" with a small image of the car, "SLV", "Gasoline", and "Downtown Branch, Providence, RI". The price is listed as "\$68.99 / Day" with a "Reserve" button.

In this example, the user is searching for an SUV at Downtown Beach pick up, drop off at the Airport Branch for dates between 2025.12.01 - 2025.12.06, and was able to find an available car. To proceed with renting, click “Reserve” and follow the prompts to process your payment (scenario 4):

This screenshot shows the same website interface as above, but with a modal dialog box overlaid. The dialog is titled "Reserve a car" and contains "Car Details": Make: Mazda, Model: CX 5, Year: 2022, and Cost Per Day: 68.99. Below this, there is a message: "Pay using card ending in 1111" with a "Cancel" button and a "Reserve" button. The background of the website is dimmed to indicate the modal is active.

After successful payment, you will be able to see your reservation under “My Rentals”:

The screenshot shows the 'Rental History' section of the Star Rentals website. It displays two completed car rentals:

- 2020 Toyota Camry**  
Rental Date: 2025-07-01 - 2025-07-05  
Starting Miles: 30000  
Pick up From: 1  
Return car to: 2
- 2022 Mazda CX-5**  
Rental Date: 2025-12-01 - 2025-12-06  
Starting Miles: 14000  
Pick up From: 1  
Return car to: 2

Each listing includes a small thumbnail image of the car, the car's model and year, the rental dates, starting miles, and pickup and return locations.

When trying to look up a car for with the same dates and filters, no car will appear, as no car is now available for new reservation:

The screenshot shows the search interface of the Star Rentals website. The search parameters are set as follows:

- Pick-up: Downtown Branch
- Drop-off: Airport Branch
- Date Range: 2025.12.01 - 2025.12.06
- Search button

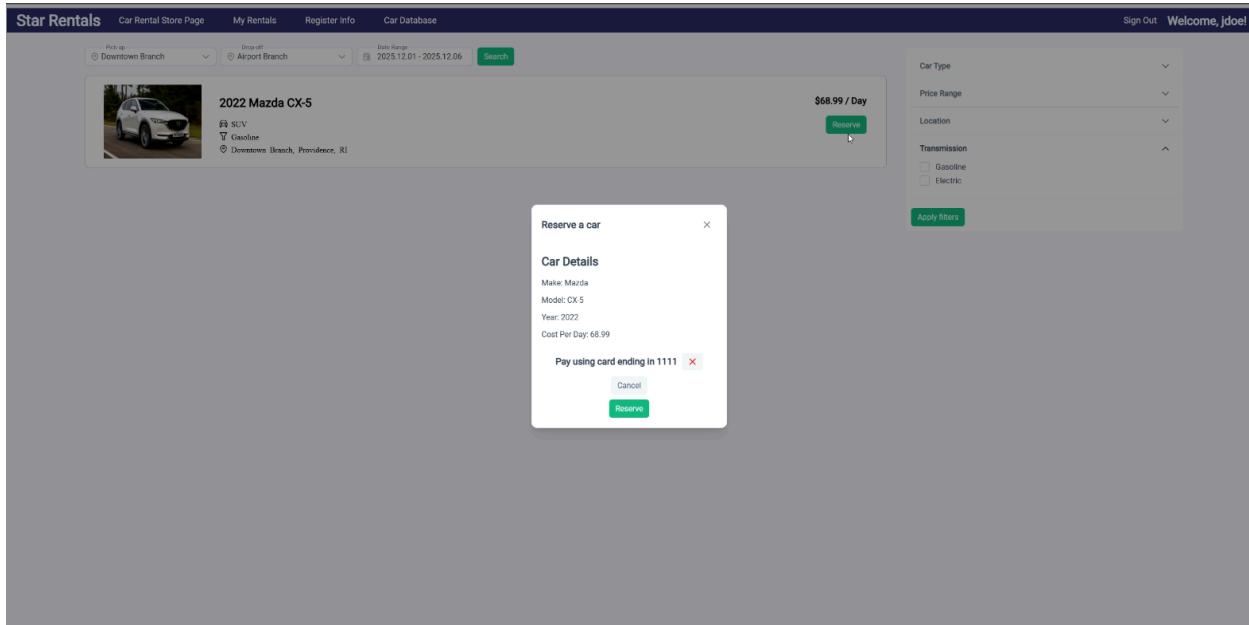
On the right side, there are several dropdown menus for applying filters:

- Car Type
- Price Range
- Location
- Transmission
  - Gasoline
  - Electric

A green "Apply filters" button is located at the bottom of the filter sidebar.

#### 4. Payment processing

A payment ID will be connected to the user if they have rented before. If there is no payment linked yet, you will be asked to enter



After confirming the payment information, click “Reserve” and you will get a notification for successful reservation

