

Car Rental Database

Section 9 Group 22
CPS510 November 27, 2023



Arhum Shah - 500 957 220
Harbaksh Singh - 500 886 288
Vraj Patel - 501 040 316

Project Description	2
Tables	2
ER Diagram	4
Queries In RA	5
Installation Instructions	8
Setting Up XAMPP and the Database	8
Turning on the Servers	11
Car Database UI Screenshots	12
Homepage	12
Booking Page	12
Customers Page	13
Car Page	13
Employee Page	14
Billing Page	14
Insurance Page	15
Maintenance Page	15

Project Description

The car rental application will be a database for a car rental company. It will be used to store customer information, vehicle information, etc. (the entities are listed below). This information will be how the business keeps track of its fleet. The relationships outlined below are vital to the database and how the different entities interact with one another.

Tables

Customer:

Customer_ID	First_name	Last_name	Email	Phone_number	Address	Date_of_birth
-------------	------------	-----------	-------	--------------	---------	---------------

Functionality: Customer will store customer information and maintain a history of customer rentals.

Car:

VIN_number	Make	Model	Year	Colour	Number_of_seats	Cost_per_day	Currently_Available
------------	------	-------	------	--------	-----------------	--------------	---------------------

Functionality: Car will hold information about all the cars in the system. Tracks car availability and cost to rent.

Booking:

Booking_ID	Customer_ID	VIN_number	Insurance_ID	Pick_up_day	Number_of_Days	Total_amount
------------	-------------	------------	--------------	-------------	----------------	--------------

Functionality: Booking will store rental information regarding the booking and calculate the total rental cost.

Employee:

Employee_ID	First_name	Last_name	Email	Phone_number	Position
-------------	------------	-----------	-------	--------------	----------

Functionality: Employee will contain information on every employee working at the car rental service.

Billing:

Billing_ID	Booking_ID	Bill_date	Status	Discount_amount	Late_fees	Taxed_amount	Total_amount
------------	------------	-----------	--------	-----------------	-----------	--------------	--------------

Functionality: Billing will contain information for each billing after the booking has been processed so the customer can be billed.

Insurance:

Insurance_ID	Coverage_type	Cost_per_day	Insurance_provider
--------------	---------------	--------------	--------------------

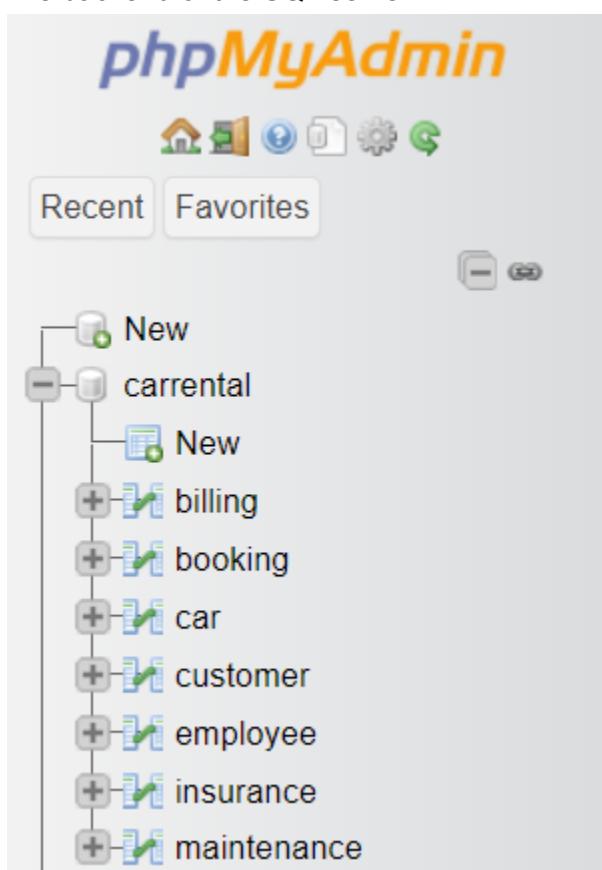
Functionality: Insurance will hold the customer's insurance information required to rent a car.

Maintenance:

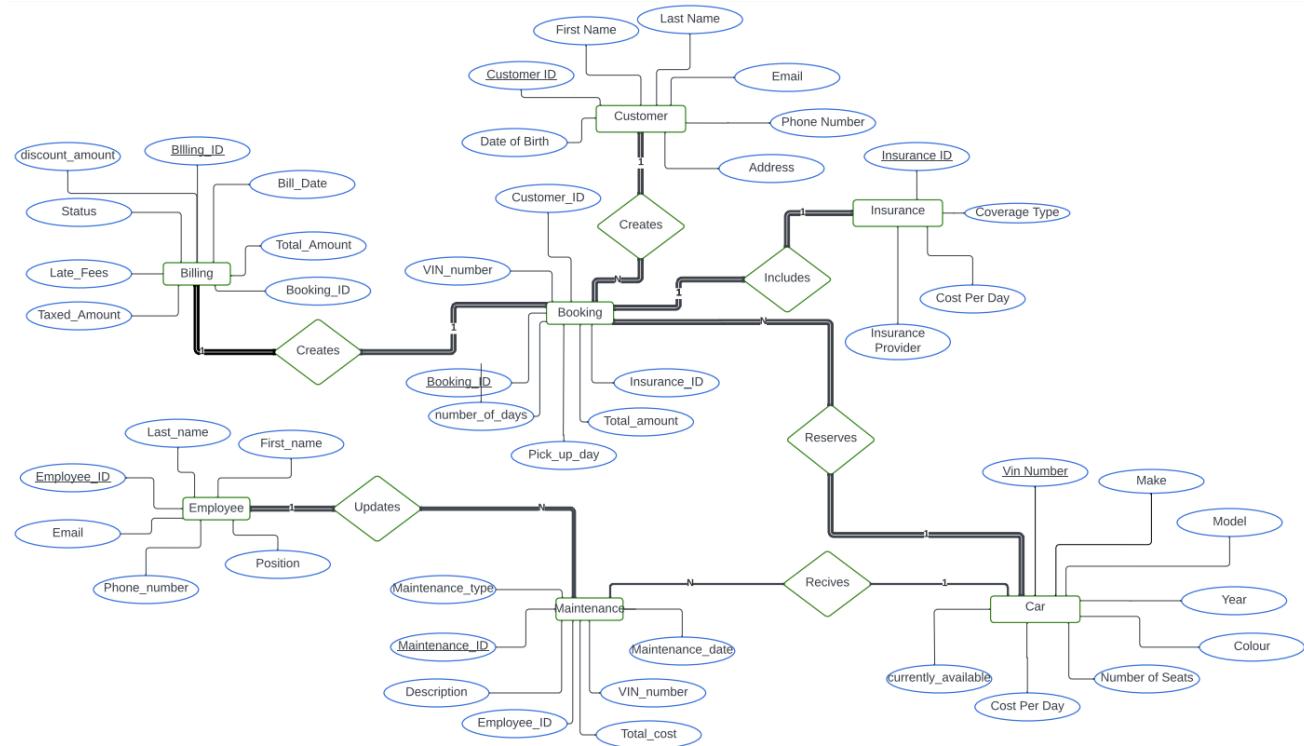
Maintenance_ID	VIN_number	Maintenance_type	Maintenance_Date	Description	Employee_ID	Total_cost
----------------	------------	------------------	------------------	-------------	-------------	------------

Functionality: Maintenance will record the history of all the maintenance performed on each car.

The backend of the SQL server:



ER Diagram



Queries In RA

-- this query retrieves the first name, last name and phone number of any bookings made.

```
SELECT c.first_name, c.last_name, c.phone_number  
FROM customer c  
WHERE EXISTS (  
    SELECT 1  
    FROM booking b  
    WHERE b.customer_id = c.customer_id  
);
```

RA:

```
 $\pi(first\_name, last\_name, phone\_number(\sigma$   
 $EXISTS(\pi_1(\sigma customer\_id=customer\_id(booking) \bowtie customer)(customer)));$ 
```

-- This query calculates the total number of bookings made by each customer and includes only customers who have made at least one booking.

```
SELECT c.customer_id, COUNT(b.booking_ID) AS total_bookings  
FROM customer c  
LEFT JOIN booking b ON c.customer_id = b.customer_id  
GROUP BY c.customer_id  
HAVING COUNT(b.booking_ID) > 0;
```

RA:

```
 $\pi customer\_id, total\_bookings(\gamma customer\_id, COUNT(booking\_ID) AS$   
 $total\_bookings(\rho c(customer) \bowtie c.customer\_id = b.customer\_id \rho b(booking)))$ 
```

-- This query lists the available car makes and models along with the count of available cars for each make and model.

```
SELECT make, model, COUNT(*) AS available_count  
FROM car  
WHERE currently_available = 'Y'  
GROUP BY make, model  
HAVING COUNT(*) > 0;
```

RA:

```
 $\pi make, model, available\_count(\gamma make, model, COUNT(*) AS available\_count(\sigma currently\_available = 'Y' (car)))$ 
```

```
-- This query finds customers who have rented cars and also had insurance for their bookings.
SELECT c.customer_id, c.first_name, c.last_name
FROM customer c
WHERE EXISTS (
    SELECT 1
    FROM booking b
    WHERE b.customer_id = c.customer_id
    AND EXISTS (
        SELECT 1
        FROM insurance i
        WHERE i.insurance_id = b.insurance_ID
    )
);

```

RA:

$$\pi \text{customer_id, first_name, last_name}(\sigma \text{EXISTS}(\pi 1(\sigma \text{customer_id} = \text{c.customer_id}(\text{booking}) \bowtie \sigma \text{insurance_id} = \text{b.insurance_ID}(\text{insurance})))$$

```
-- Select all unique maintenance and insurance types
SELECT DISTINCT m.maintenance_type AS type, 'Maintenance' AS category
FROM maintenance m
UNION ALL
SELECT DISTINCT i.coverage_type AS type, 'Insurance' AS category
FROM insurance i;
```

RA:

$$\pi \text{type, category}(\sigma \text{category} = \text{'Maintenance'}(\text{Maintenance})) \cup \pi \text{type, category}(\sigma \text{category} = \text{'Insurance'}(\text{Insurance}))$$

```
--This query allows you to find customers whose average booking amount exceeds
SELECT c.customer_id, AVG(b.total_amount) AS average_total_amount
FROM customer c
LEFT JOIN booking b ON c.customer_id = b.customer_id
GROUP BY c.customer_id
HAVING AVG(b.total_amount) > 200;
```

RA:

$$\pi \text{customer_id, average_total_amount}(\gamma \text{customer_id, AVG(total_amount)} \text{ AS average_total_amount}(\rho c(\text{customer}) \bowtie c.\text{customer_id} = b.\text{customer_id} \rho b(\text{booking})))$$

```

-- Show all bookings excluding ones without an insurance coverage
SELECT
    b.booking_ID,
    b.customer_id,
    i.coverage_type
FROM
    booking b
JOIN
    insurance i ON b.insurance_ID = i.insurance_id
MINUS
SELECT
    b.booking_ID,
    b.customer_id,
    i.coverage_type
FROM
    booking b
JOIN
    insurance i ON b.insurance_ID = i.insurance_id
WHERE
    i.coverage_type = 'NA';

```

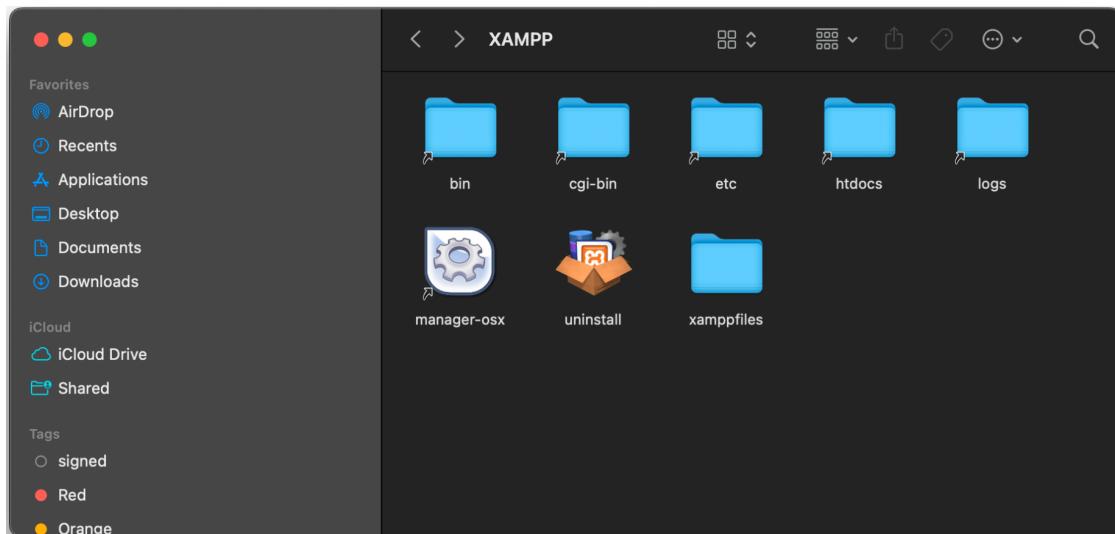
RA:

$\pi_{booking_ID, customer_id, coverage_type}(\sigma_{coverage_type = 'NA'}(insurance) - \pi_{booking_ID, customer_id, coverage_type}(insurance))$

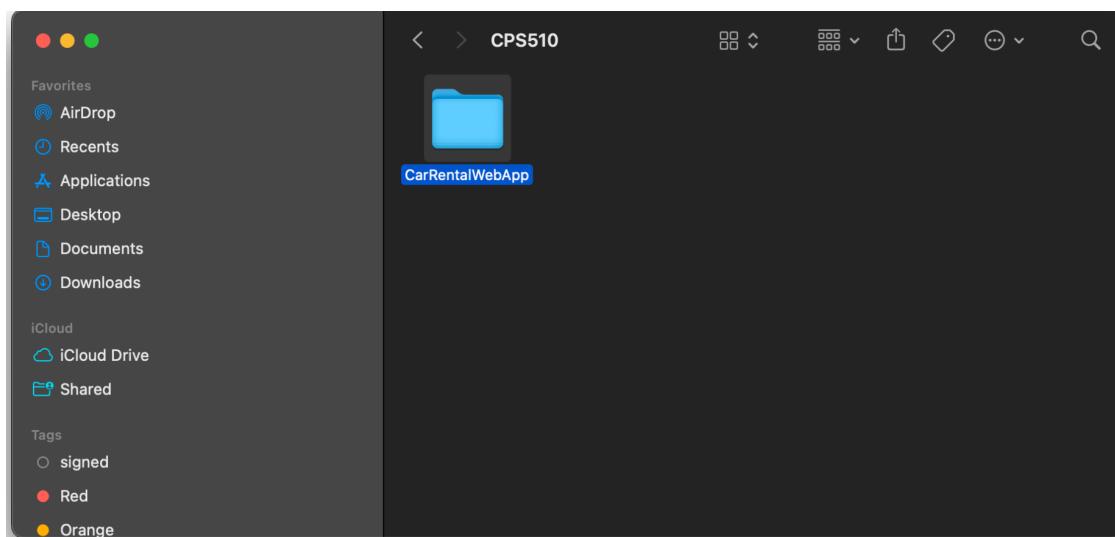
Installation Instructions

Setting Up XAMPP and the Database

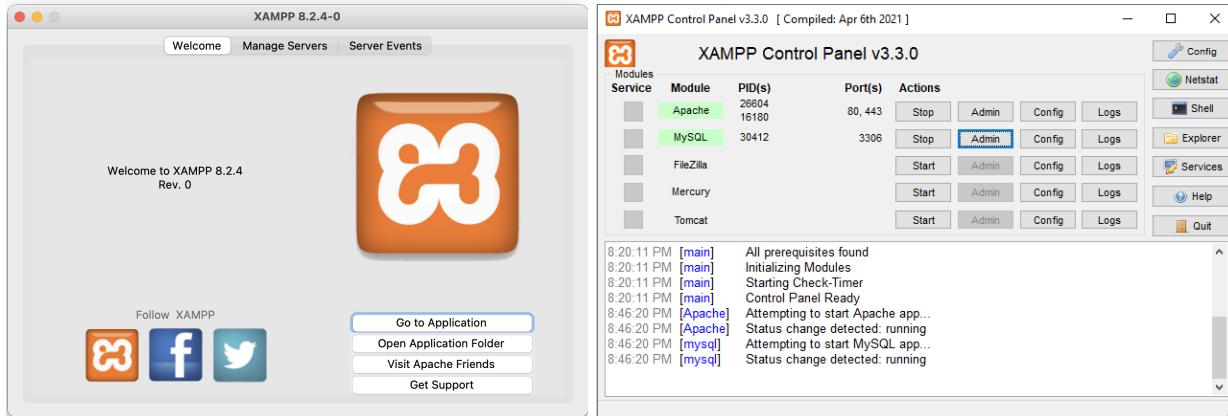
1. Follow the link below to find the downloads for XAMPP and select the version 8.0.20 download for your machine. <https://www.apachefriends.org/download.html>
You can also use this video guide to setup XAMPP for the first time.
<https://www.youtube.com/watch?v=-f8N4FEQWY>
2. Open the downloaded file and follow along with the installation setup prompts.
3. Once downloaded open your XAMPP application location and locate a folder called “htdocs” and open it. Create a folder called “CPS510”



4. Extract the CarRentalWebApp.zip file and drag the whole folder into the newly created “CPS510” file inside “htdocs.”



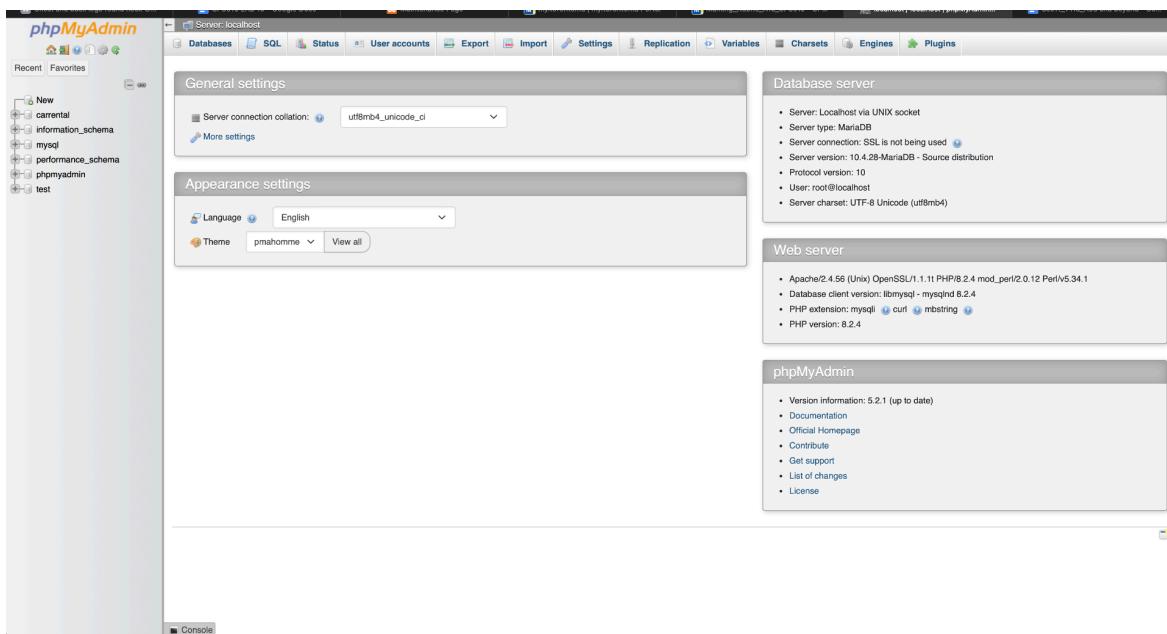
- Open up XAMPP application and click “Go to Application” on the main screen. This will open up a tab on your browser with the XAMPP web application. If on Windows, you will have to click on “Start” for the “MySQL” service first, then click on “Admin” to reach the XAMPP web application.



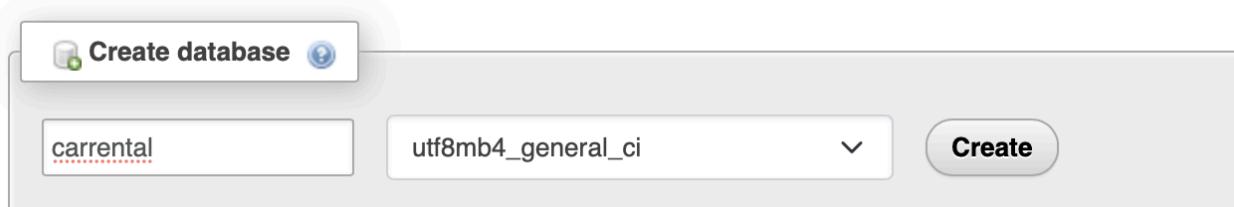
- Click on “phpMyAdmin” in the navigation bar to open up the database interface. Which will allow you to create the database for iFitness.



- Click on “New” in the navbar on the left to create a database for Car Rena



Databases



- Once the database is created, click on the “SQL” tab in the navbar at the top and paste all the MySQL code in the file “commands.sql” (“commands.sql” is located within CarRentalWebApp.zip) from line 2 onwards and then click “Go.”

The screenshot shows the MySQL Workbench interface with the 'Database: carrental' selected. The top navigation bar has tabs for 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', and 'Operations'. The 'SQL' tab is currently active. Below the tabs is a 'Run SQL query/queries on database carrental:' input field. The input field contains the number '1'. The main area shows the contents of the 'commands.sql' file, starting with:

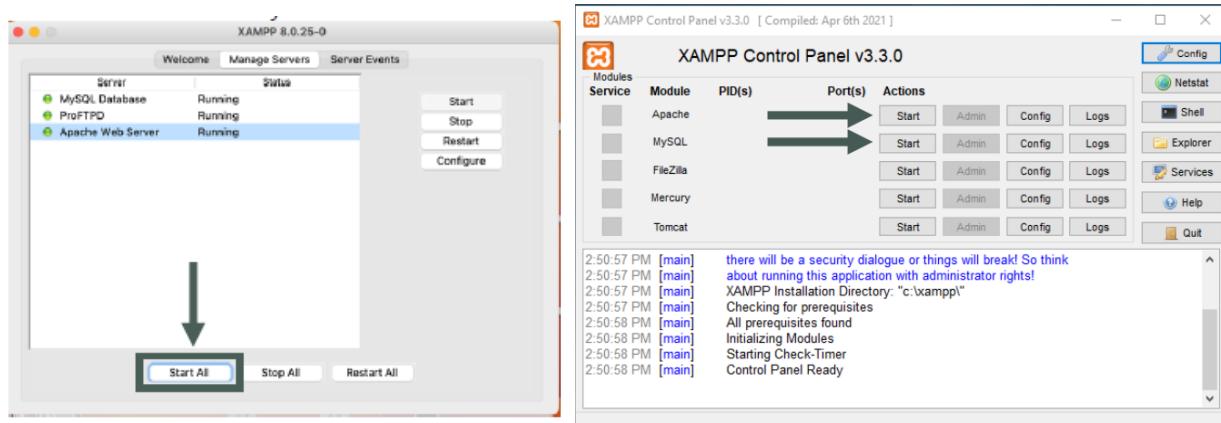
```
1 CREATE TABLE customer (
2     customer_id VARCHAR(20) PRIMARY KEY,
3     first_name VARCHAR(20) NOT NULL,
4     last_name VARCHAR(20) NOT NULL,
5     email VARCHAR(50),
6     phone_number VARCHAR(15),
7     address VARCHAR(100),
8     date_of_birth DATE
9 );
10 CREATE TABLE car (
11     VIN_number VARCHAR(10) PRIMARY KEY,
12     make VARCHAR(20) NOT NULL,
13     model VARCHAR(20) NOT NULL,
14     year VARCHAR(4),
15     colour VARCHAR(10),
16     number_of_seats INT,
17     cost_per_day DECIMAL(5, 2) NOT NULL,
```

Below the SQL code, there are several buttons: 'Clear', 'Format', 'Get auto-saved query', 'Bind parameters', and 'Bookmark this SQL query:'. At the bottom, there are options for 'Delimiter', 'Show this query here again', 'Retain query box', 'Rollback when finished', 'Enable foreign key checks', and a 'Go' button.

- Now your XAMPP database is fully setup!

Turning on the Servers

1. Go to the XAMPP application and click on “Manage Servers” in the top navbar. Then click on “Start All.” If on a Windows, you will have to click “Start” on all the Services individually.



2. Now using the following link (<http://localhost/CPS510/CarRentalWebApp/>) you should be able to open the CarRentalWebApp application.
3. You are all done! Enjoy using CarRentalWebApp !

Car Database UI Screenshots

Homepage



Booking Page

A screenshot of the booking page. At the top, there's a dark header bar with the text "CityZoom Rentals" on the left and navigation links for "Booking", "Customer", "Car", "Employee", "Billing", "Insurance", and "Maintenance" on the right. Below the header is a table titled "Bookings" with the following data:

Booking ID	Customer ID	VIN Number	Insurance ID	Pick Up Day	Number Of Days	Total Amount	Edit	Delete
BOOK1234	CUST002	VIN001	INS004	2023-11-21	5	500.00	<button>Edit</button>	<button>Delete</button>
BOOK4563	CUST004	VIN002	INS003	2023-11-21	3	210.00	<button>Edit</button>	<button>Delete</button>

Booking ID

Customer ID
 Select Customer

VIN Number
 Select Vehicle

Insurance ID
 Select Insurance

Pick Up Day
 2023-11-27

Number Of Days
 0

Total Amount
 0

Customers Page

CityZoom Rentals

Booking Customer Car Employee Billing Insurance Maintenance

Customers								
Customer ID	First Name	Last Name	Email	Phone Number	Address	Date of Birth	Edit	Delete
CUST002	Jane	Smith	janesmith@email.com	555-987-6543	456 Elm St, Oakville, Canada	1995-09-22	<button>Edit</button>	<button>Delete</button>
CUST004	Emily	Johnson	emily@email.com	555-333-2222	101 Pine St, Brampton, Canada	1991-03-28	<button>Edit</button>	<button>Delete</button>
CUST753	Joe	Adams	joe@test.com	123-467-1233	23 robern	2023-11-02	<button>Edit</button>	<button>Delete</button>

Customer ID

First Name

Last Name

Email

Phone number

Address

Date of Birth

Car Page

CityZoom Rentals

Booking Customer Car Employee Billing Insurance Maintenance

Cars									
VIN Nnumber	Make	Model	Year	Number of Seats	Colour	Cost Per Day	Currently Available	Edit	Delete
VIN001	Toyota	Camry	2022	Blue	5	50.00	N	<button>Edit</button>	<button>Delete</button>
VIN002	Honda	Civic	2024	White	7	40.00	Y	<button>Edit</button>	<button>Delete</button>
VIN003	Ford	F150	2020	Silver	2	70.00	Y	<button>Edit</button>	<button>Delete</button>
VIN004	BMW	X5M	2023	Black	4	150.00	N	<button>Edit</button>	<button>Delete</button>
VIN005	Mercedes Benz	S63	2022	Black	5	180.00	Y	<button>Edit</button>	<button>Delete</button>

VIN Number

Make

Model

Year

Colour

Employee Page

CityZoom Rentals

Booking Customer Car Employee Billing Insurance Maintenance

Employees							
Employee ID	First Name	Last Name	Email	Phone Number	Position	Edit	Delete
EMP001	John	Smith	john.smith@email.com	555-111-2222	Manager	Edit	Delete
EMP002	Emily	Johnson	emily.johnson@email.com	555-333-4444	Manager	Edit	Delete
EMP003	Michael	Brown	michael.brown@email.com	555-555-6666	Technician	Edit	Delete
EMP004	Jessica	Doe	jessica.doe@email.com	555-777-8448	Technician	Edit	Delete
EMP005	David	Williams	david.williams@email.com	555-999-0000	Technician	Edit	Delete

Employee ID

First Name

Last Name

Email

Phone number

Billing Page

CityZoom Rentals

Booking Customer Car Employee Billing Insurance Maintenance

Employees							
Employee ID	First Name	Last Name	Email	Phone Number	Position	Edit	Delete
EMP001	John	Smith	john.smith@email.com	555-111-2222	Manager	Edit	Delete
EMP002	Emily	Johnson	emily.johnson@email.com	555-333-4444	Manager	Edit	Delete
EMP003	Michael	Brown	michael.brown@email.com	555-555-6666	Technician	Edit	Delete
EMP004	Jessica	Doe	jessica.doe@email.com	555-777-8448	Technician	Edit	Delete
EMP005	David	Williams	david.williams@email.com	555-999-0000	Technician	Edit	Delete

Employee ID

First Name

Last Name

Email

Phone number

Insurance Page

CityZoom Rentals

Booking Insurance Car Employee Billing Insurance Maintenance

Explore Our Insurance Coverage Types

Choose the perfect coverage for your journey with CityZoom Rentals.



Comprehensive Coverage

Get peace of mind with our comprehensive coverage that includes a wide range of protections.



Prepare For The Unexpected

Minimize your liability for damage to the rental car in case of an accident. Opt in for insurance to ensure that you are protected from unexpected charges.

Insurance					
Insurance ID	Coverage Type	Cost Per Day (\$)	Insurance Provider	Edit	Delete
INS001	NA	0.00	NA	<button>Edit</button>	<button>Delete</button>
INS002	Basic	10.00	State Farm	<button>Edit</button>	<button>Delete</button>
INS003	Premium	30.00	AllState	<button>Edit</button>	<button>Delete</button>
INS004	Extended	50.00	CAA	<button>Edit</button>	<button>Delete</button>
INS005	Extended PLUS	100.00	CAA	<button>Edit</button>	<button>Delete</button>

Maintenance Page

CityZoom Rentals

Booking Customer Car Employee Billing Insurance Maintenance

Maintenance

Maintenance ID	VIN Number	Maintenance Type	Maintenance Date	Description	Employee ID	Total Cost	Edit	Delete
1	VIN001	Oil Change	2023-10-05	Regular maintenance	EMP002	75.00	<button>Edit</button>	<button>Delete</button>
2	VIN002	Brake Replacement	2023-10-10	Brake pads and rotors	EMP003	200.00	<button>Edit</button>	<button>Delete</button>
3	VIN003	Tire Rotation	2023-10-15	Rotate and balance tires	EMP002	50.00	<button>Edit</button>	<button>Delete</button>
4	VIN004	Engine Tune-Up	2023-10-18	Engine diagnostics and tune-up	EMP002	300.00	<button>Edit</button>	<button>Delete</button>
5	VIN005	Detailed	2023-10-20	Full car detailing	EMP001	100.00	<button>Edit</button>	<button>Delete</button>
6	VIN005	Detailed	2023-10-21	Engine diagnostics and tune-up	EMP002	300.00	<button>Edit</button>	<button>Delete</button>

Maintenance ID

VIN Number

Employee ID

Maintenance Type

Maintenance Date

ER Diagram

