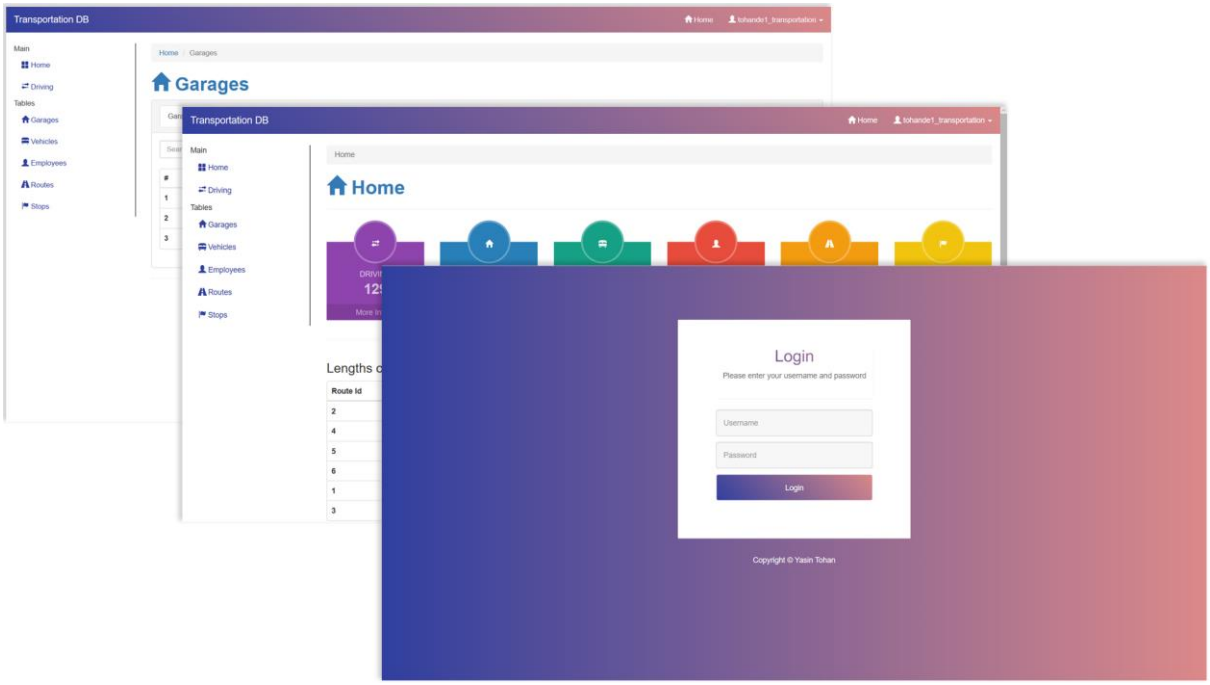
Public Transportation Database System

Yasin Tohan – 11945863940

The public transportation database system stores driving, employee, stop, route and garage information of all kinds of transportation vehicles in the city. Transportation managers can access many personal information of the person using a transportation vehicle using this system. They can also learn the type of vehicle, the number of seats in the vehicle, and the route the vehicle will go.



You can go to http://datahw.tohandesign.com/ to test the project without installing it.

(All permissions)

Username: transportation

Password: transportationpass

(View permission only)

Username: user

Password: testuserpass

Requirements

- Each garage has its own unique id, capacity and addres.
- Every garage has vehicles and employs employees.
- Each vehicle has its own unique id, number of seats, type, model and route.
- The SSN number, name, gender, age, date of birth, phone numbers and address of each employee are recorded.
- Each employee can use different vehicles on different dates.
- There are routes with unique id, source stop, departure stop and departure times.
- Each vehicle can follow different routes.
- There are stops with unique id, name and location.
- Each route goes through multiple stops.

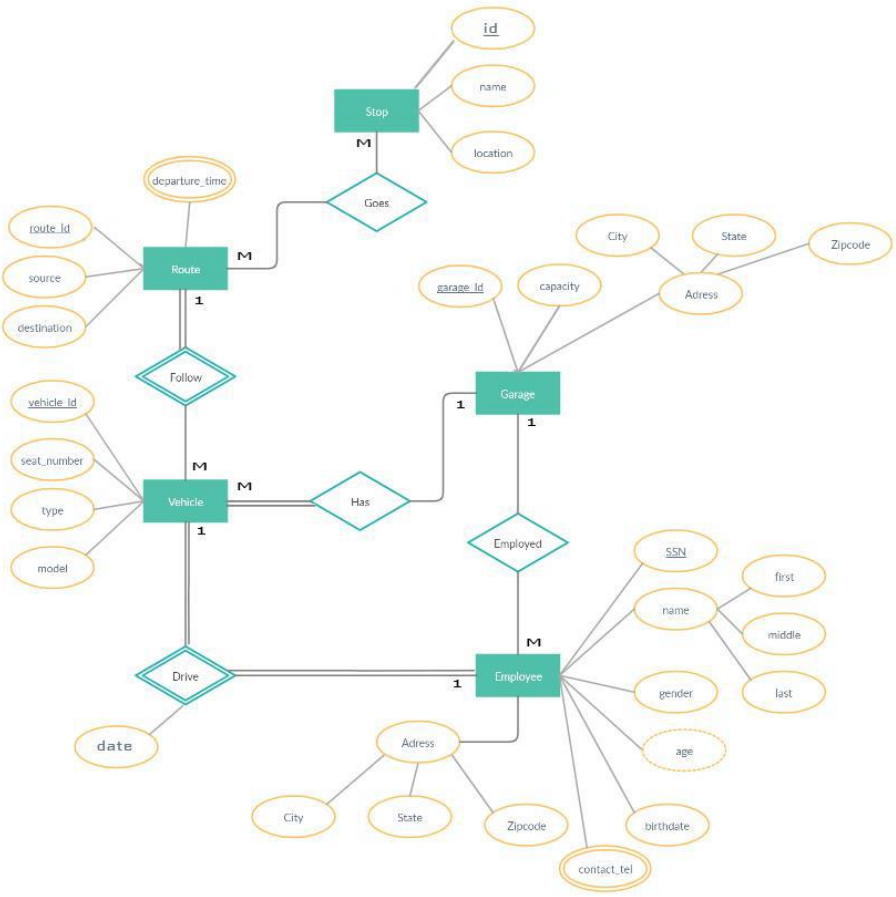
Queries

1. What is the most used vehicle type in May?
2. Which is the longest route?
3. What is the average age of employees?
4. What is the most departing vehicle?
5. Which garage has the most vehicles?

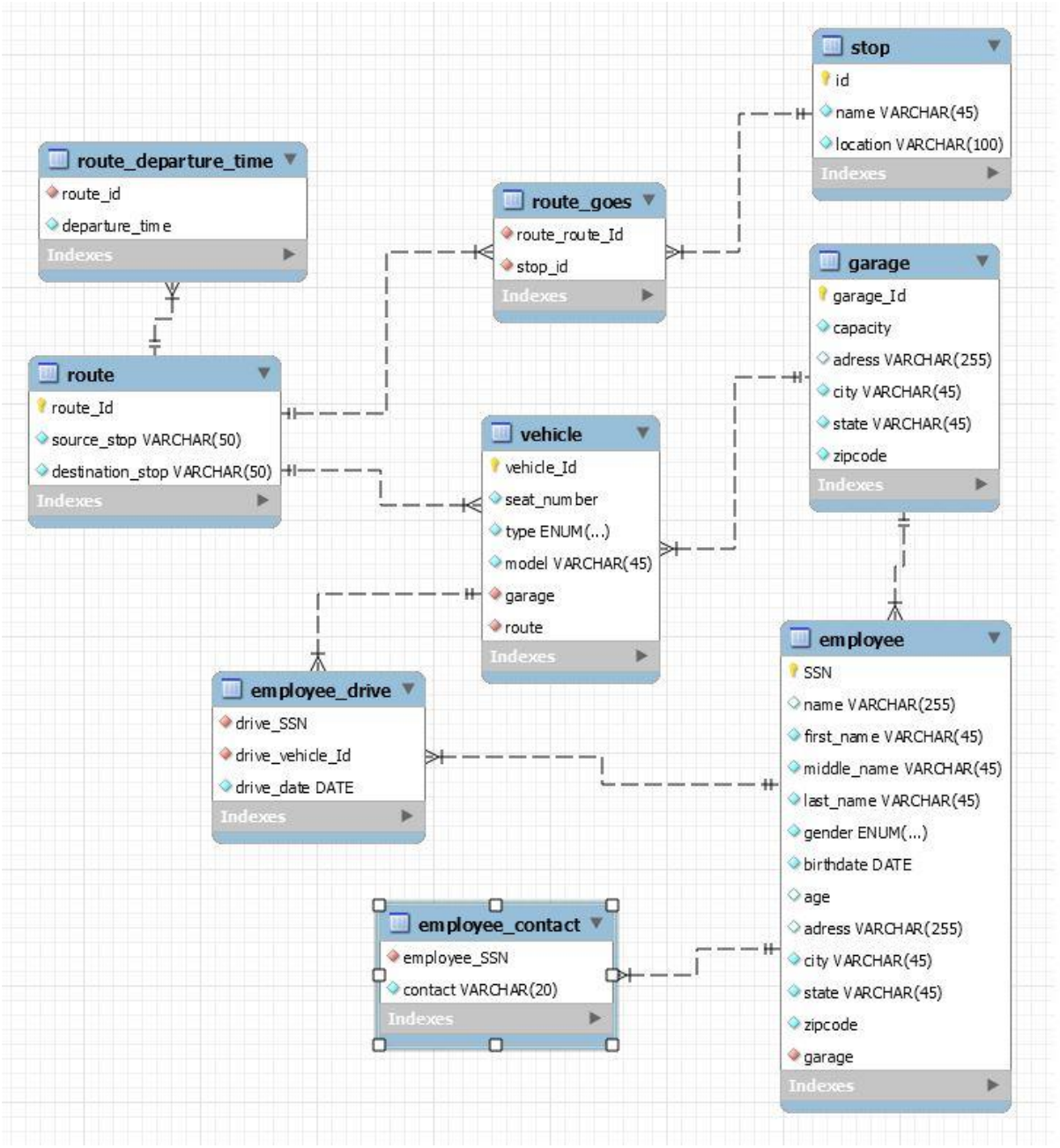
E/R Design

- **Garage** is a strong entity, with an identifier, **garage_Id**, created to be the primary key used to distinguish between garages.
- **Employee** is a strong entity, with an identifier, **SSN**, created to be the primary key used to distinguish between employees.
- **Vehicle** is a strong entity, with an identifier, **vehicle_Id**, created to be the primary key used to distinguish between vehicles.
- **Route** is a strong entity, with an identifier, **route_Id**, created to be the primary key used to distinguish between routes.
- **Stop** is a strong entity, with an identifier, **id**, created to be the primary key used to distinguish between stops.
- Every vehicle has to belong to a garage.
- Every vehicle should be driven by an employee, and vehicles and employees should not be left idle.
- Garage and Vehicle are related through the one-to-many Attempts relationships. A vehicle can not exist without a garage.
- Employee and Vehicle are related through the one-to-one Attempts relationships. the employee and the vehicle cannot exist without each other.
- Vehicle and Route are related through the many-to-one Attempts relationships. A vehicle can not exist without a route.
- Route and Stops are related through the many-to-many Attempts relationships.
- When a employee drive a vehicle, there is attribute to capture the date.

E/R Model



SQL Diagram


stop
id
name VARCHAR(45)
location VARCHAR(100)
Indexes
route_departure_time
route_id
departure_time
Indexes
route_goes
route_route_Id
stop_id
Indexes
garage
garage_Id
capacity
adress VARCHAR(255)
city VARCHAR(45)
state VARCHAR(45)
zipcode
Indexes
route
route_Id
source_stop VARCHAR(50)
destination_stop VARCHAR(50)
Indexes
vehicle
vehicle_Id
seat_number
type ENUM(...)
model VARCHAR(45)
garage
route
Indexes
employee
SSN
name VARCHAR(255)
first_name VARCHAR(45)
middle_name VARCHAR(45)
last_name VARCHAR(45)
gender ENUM(...)
birthdate DATE
age
adress VARCHAR(255)
city VARCHAR(45)
state VARCHAR(45)
zipcode
garage
Indexes
employee_drive
drive_SSN
drive_vehicle_Id
drive_date DATE
Indexes
employee_contact
employee_SSN
contact VARCHAR(20)
Indexes

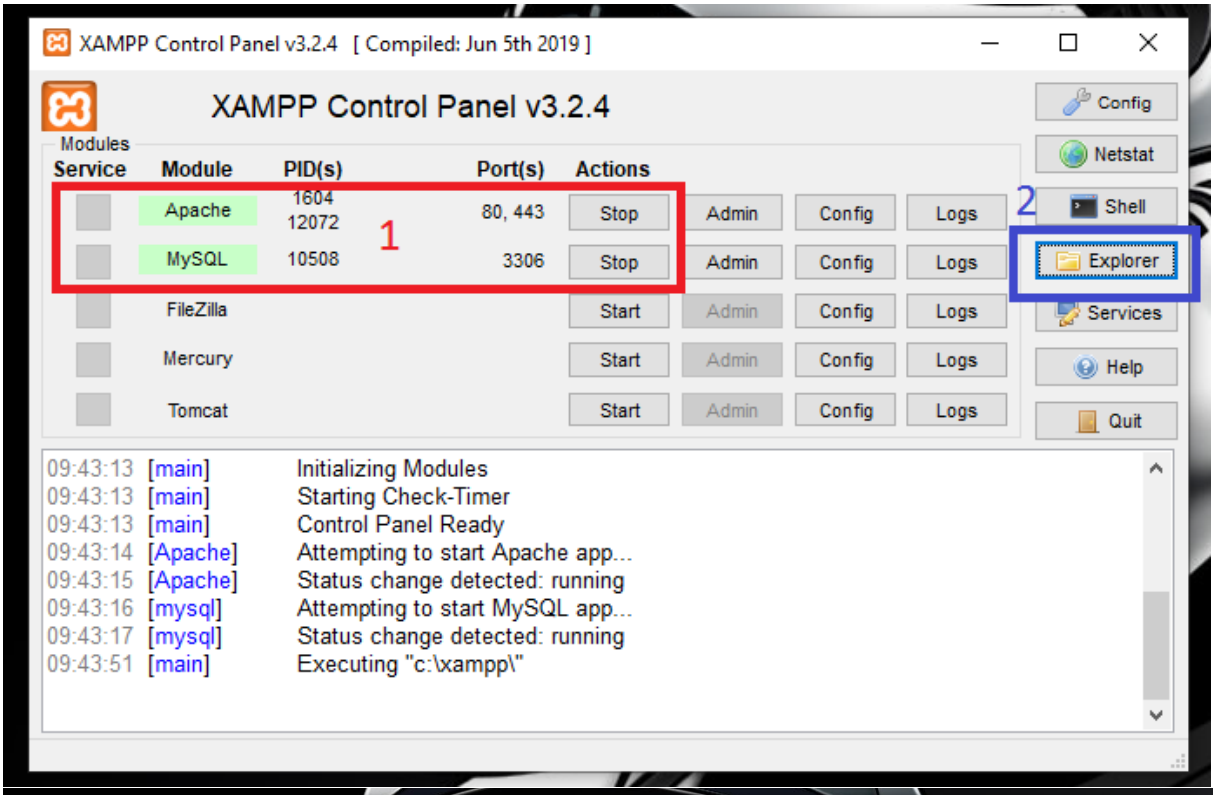
Local Installation

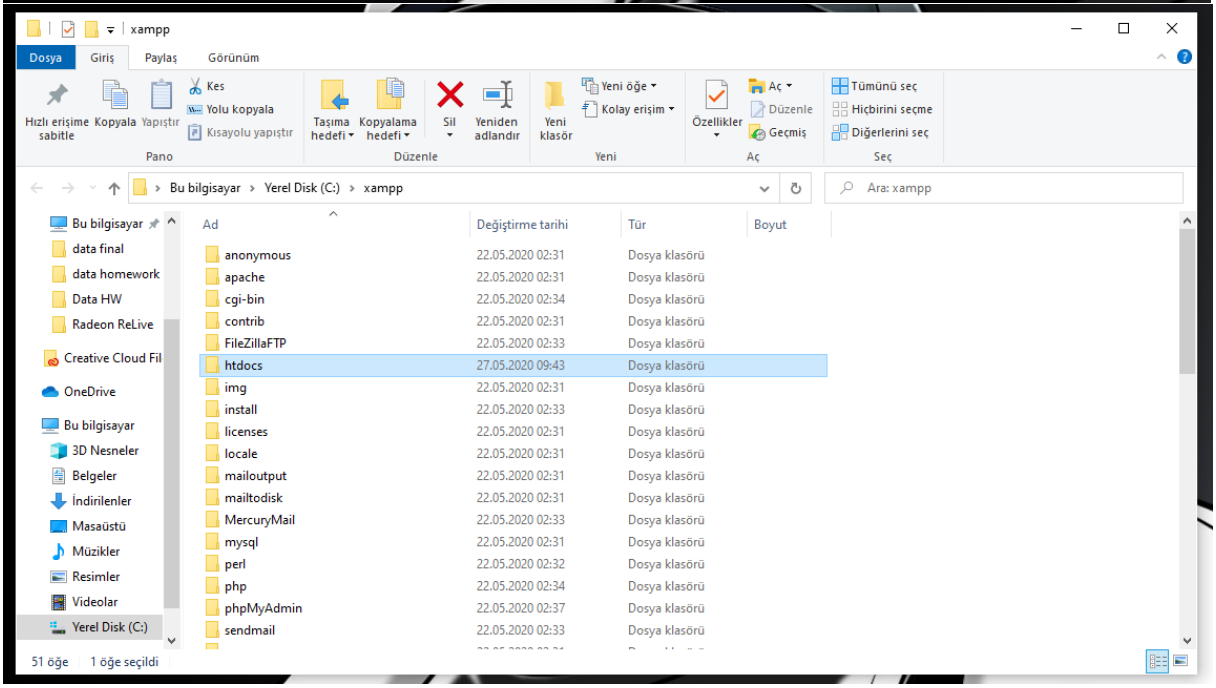
Step-1

We open the xampp application and run the **Apache** and **MySQL** local server.

Step-2

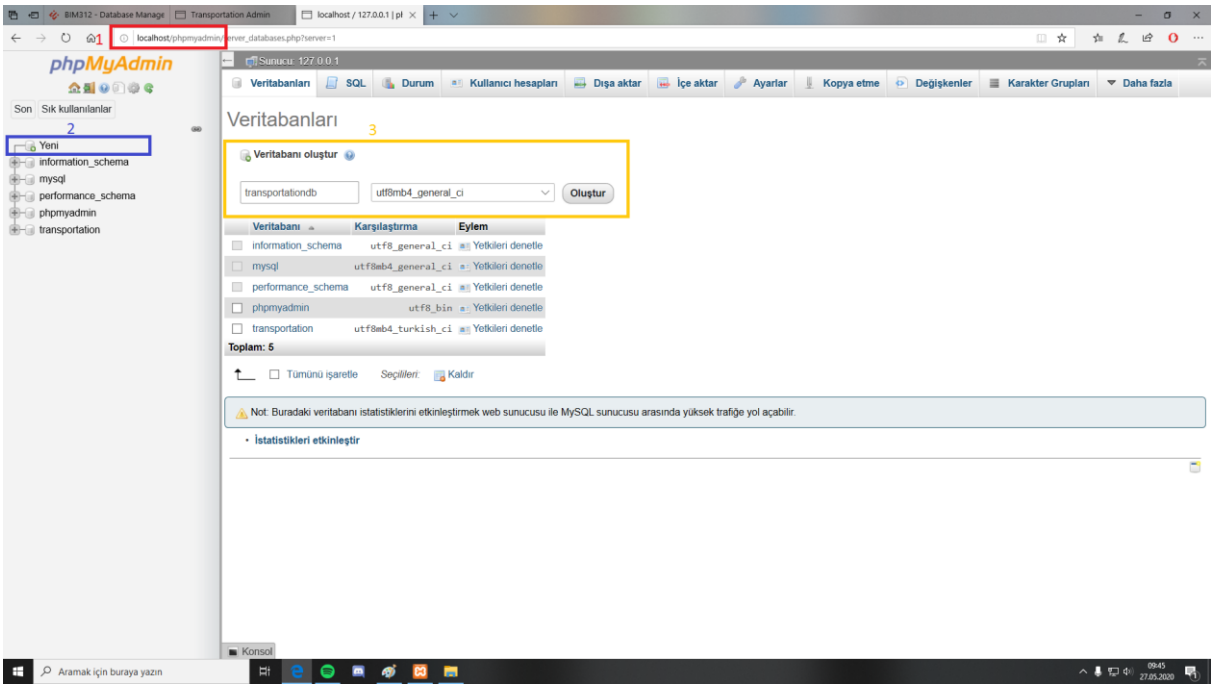
We open the file location of the xampp application and put our project in the **htdocs** folder.





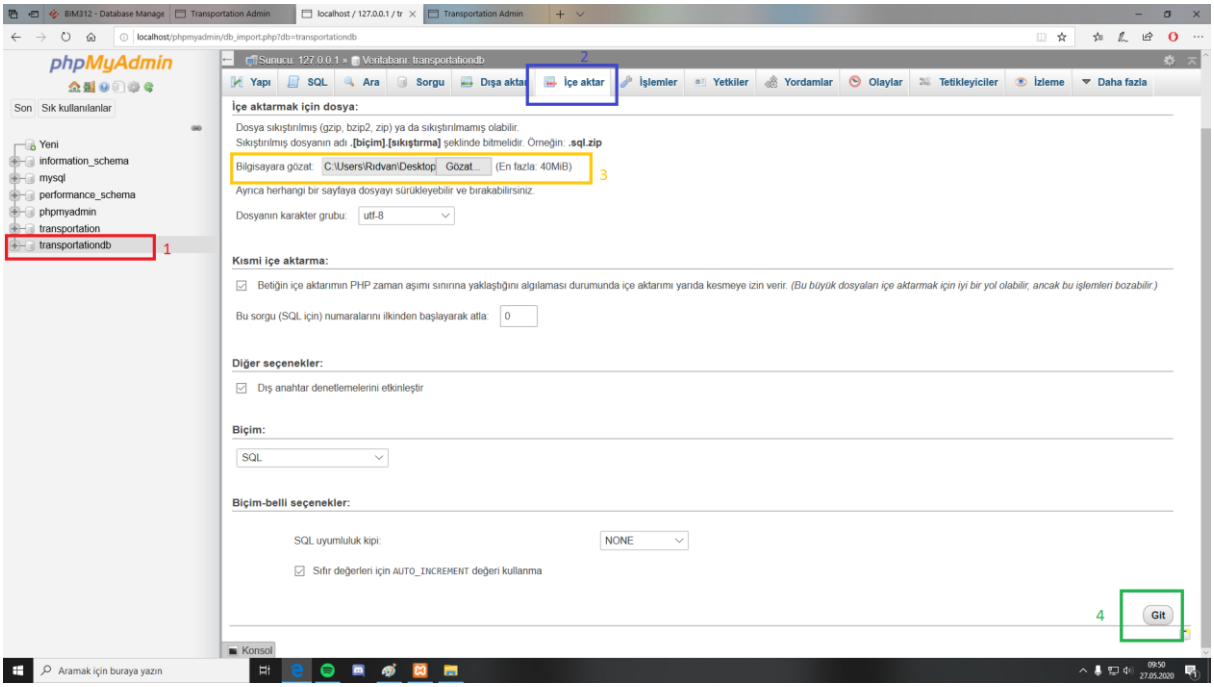
Step-3

We create our database by entering the **“localhost/phpmyadmin”** page from our browser, pressing the new button and entering our database name.



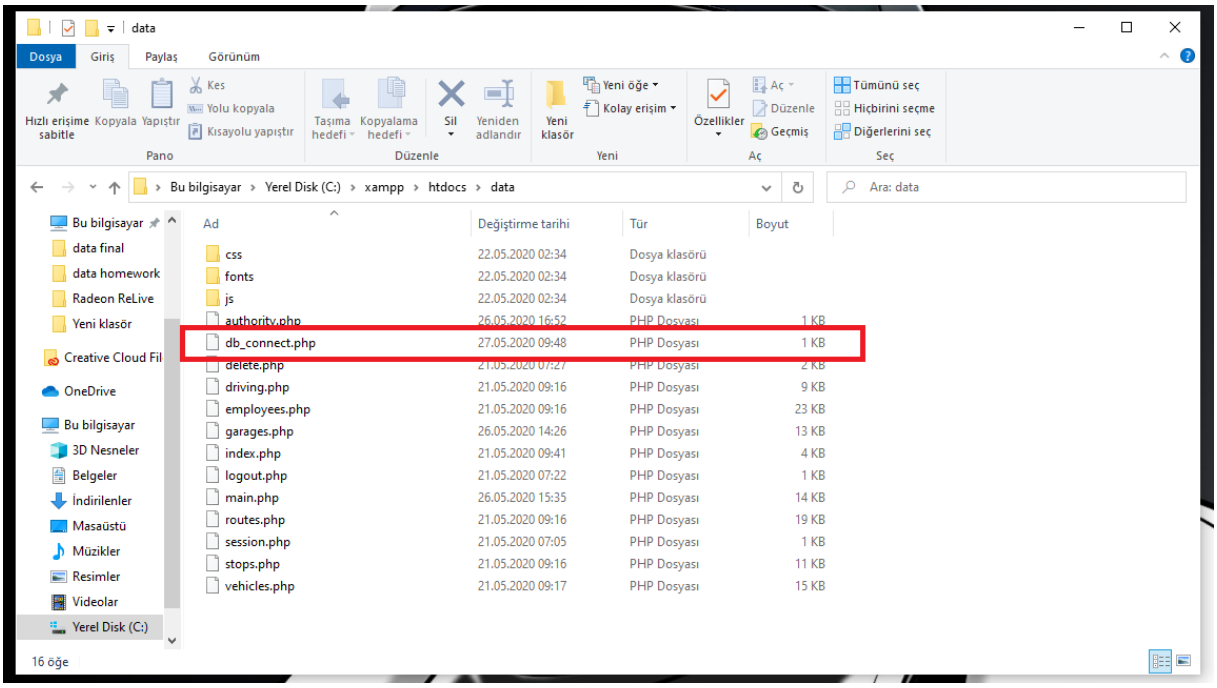
Step-4

After entering the database we created, we enter the **"import"** tab. After selecting the sql file of our project from our computer, we complete the import.



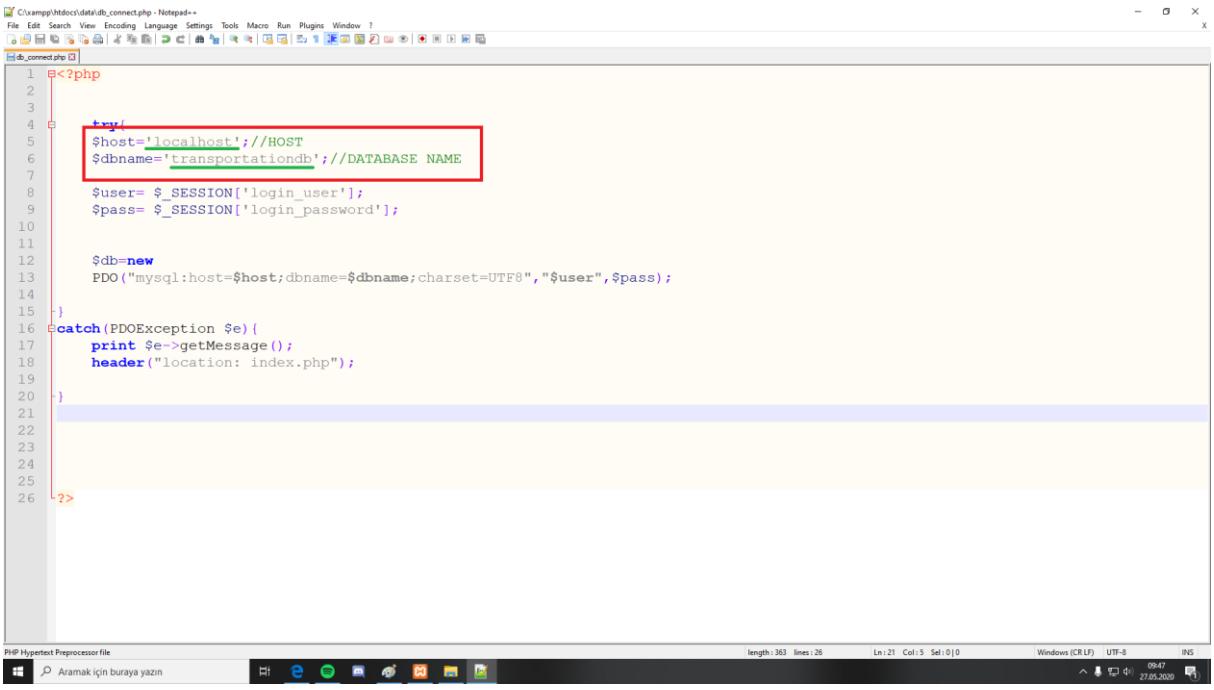
Step-5

We enter our project file, which we put in the htdocs folder. We open the file **"db_connect.php"** in any editor.



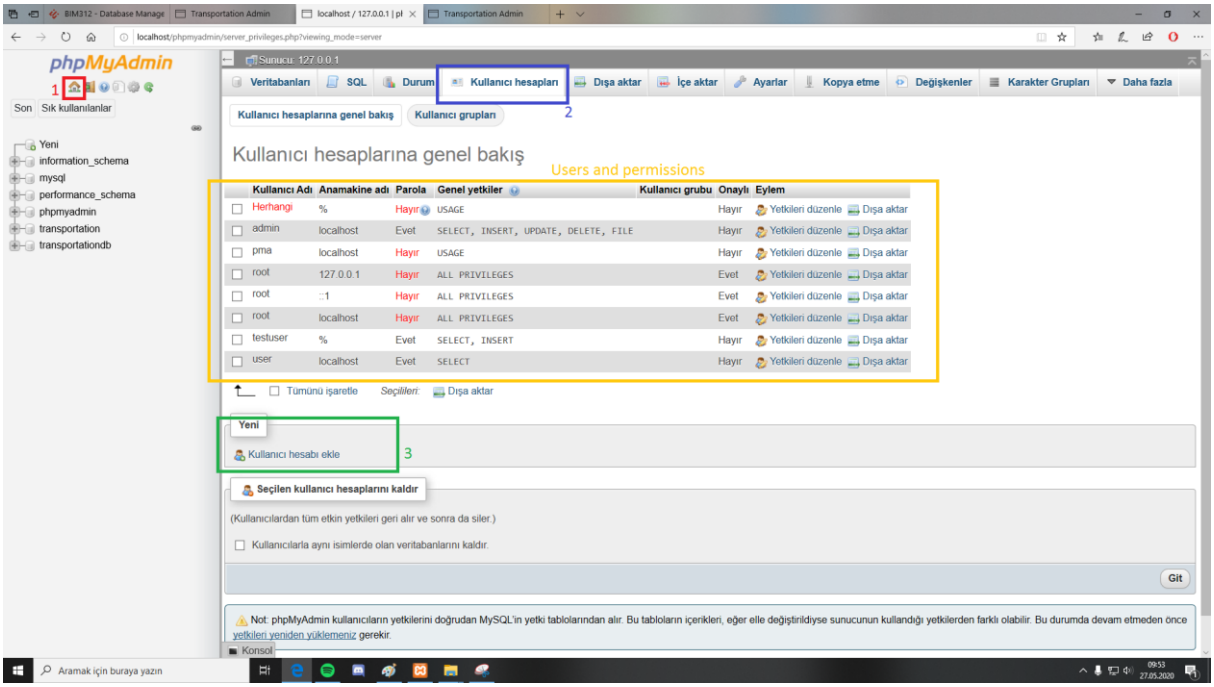
Step-6

We write the name of the database to the **“$dbname”** variable in the file we open.



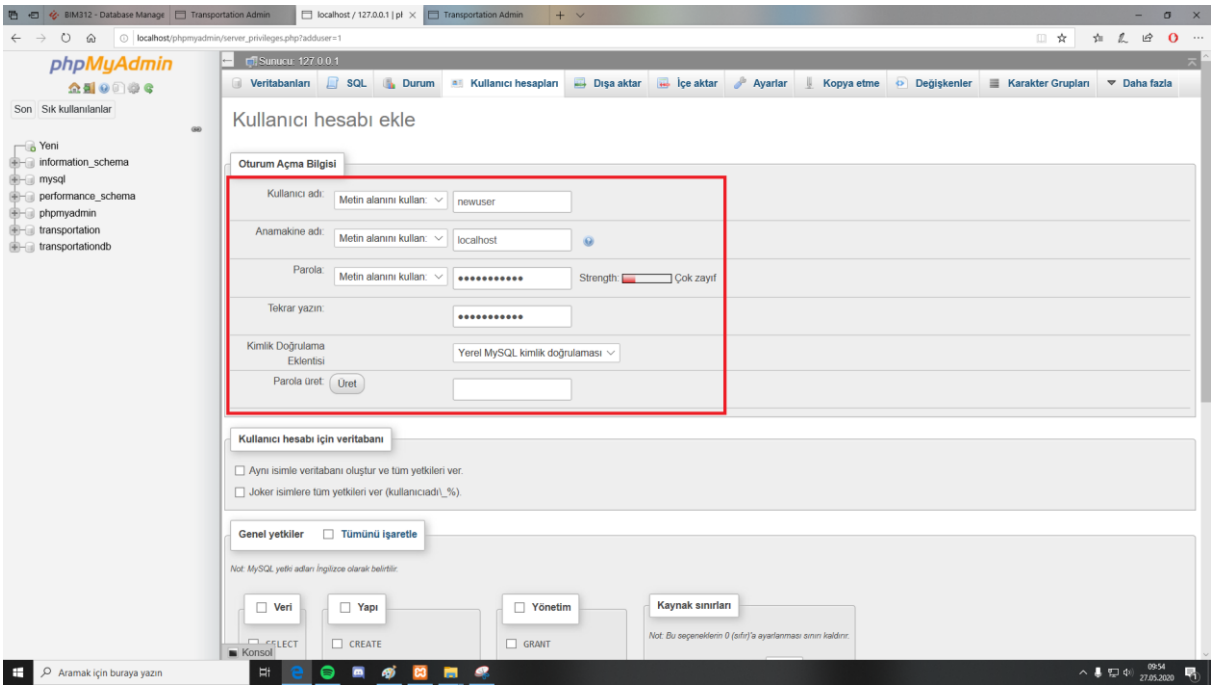
Step-7 (Optional)

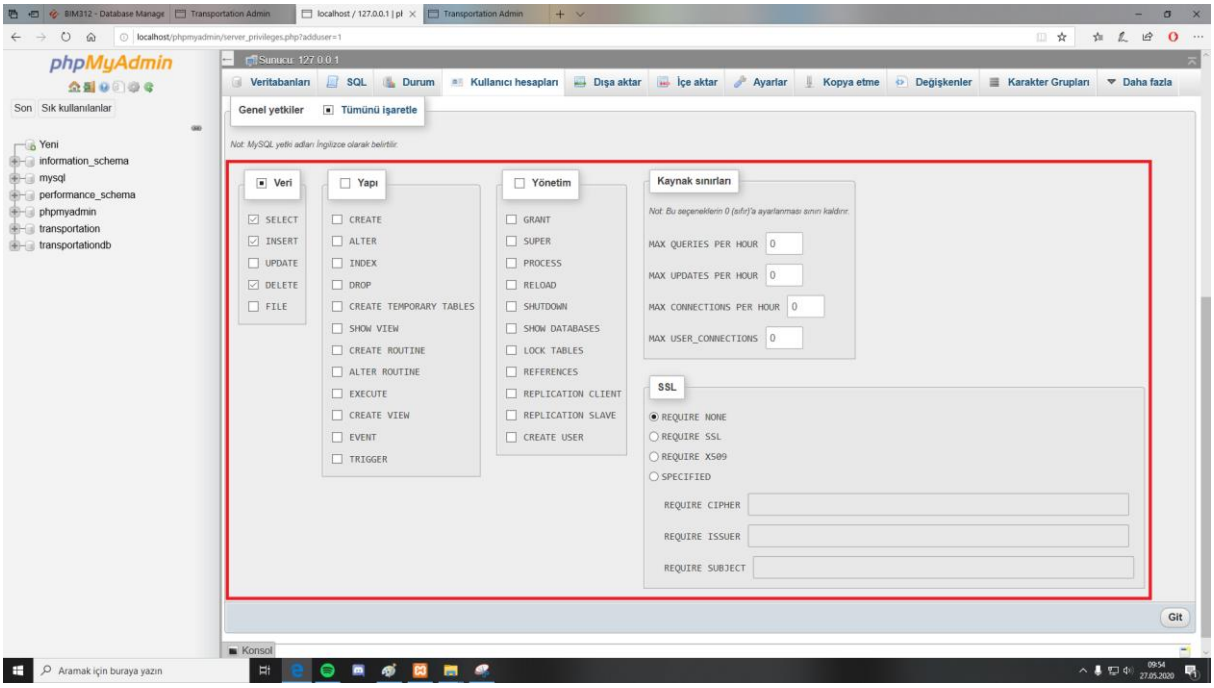
You can see the existing users and their privileges in the yellow section.



Adding new users

To add new users, you can add new users by entering the user accounts section on our phpmyadmin page. You can add a new user by selecting the user name, password, server address and permissions from the window that opens after pressing the add user button.



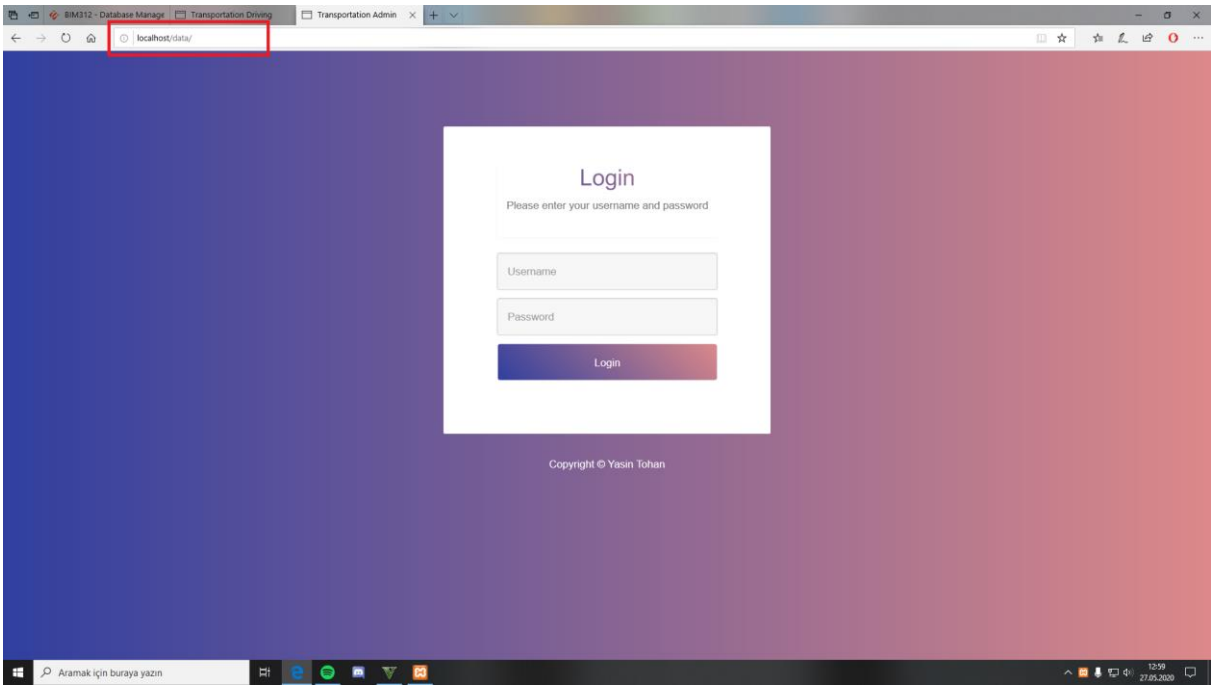


Step-8

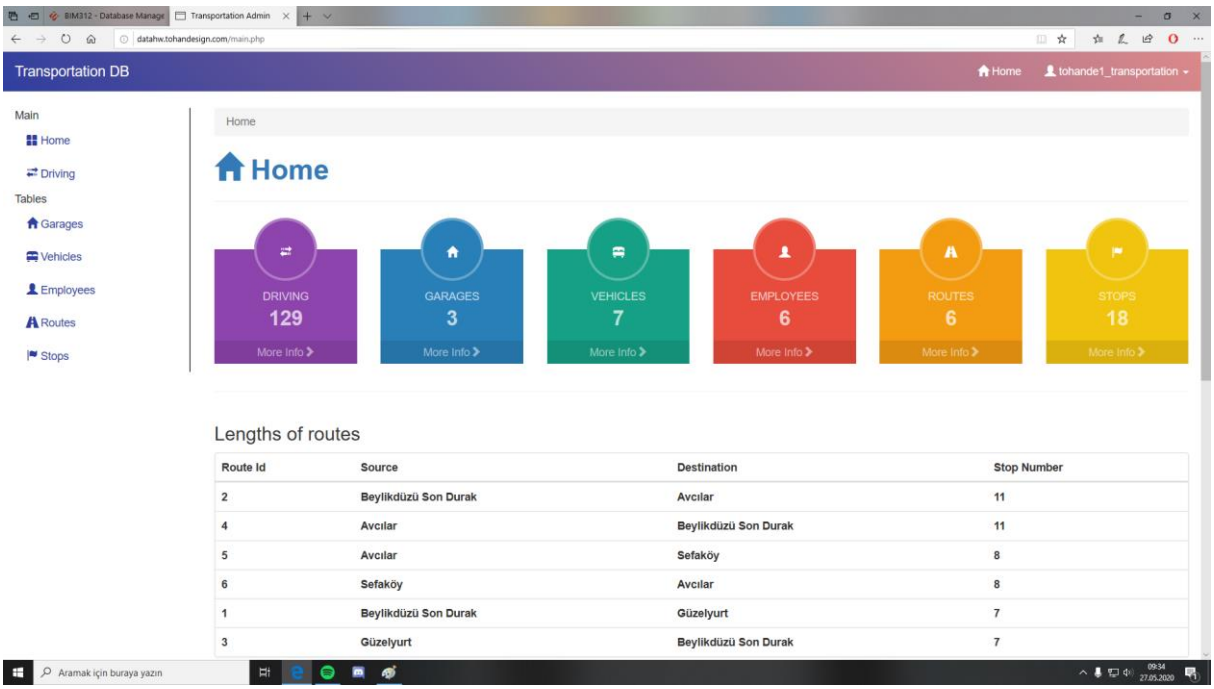
We open our project by typing **“localhost/[folder name]”** from our browser.

We log in by entering the username and password of one of the users we have added to our phpmyadmin page.

Login page

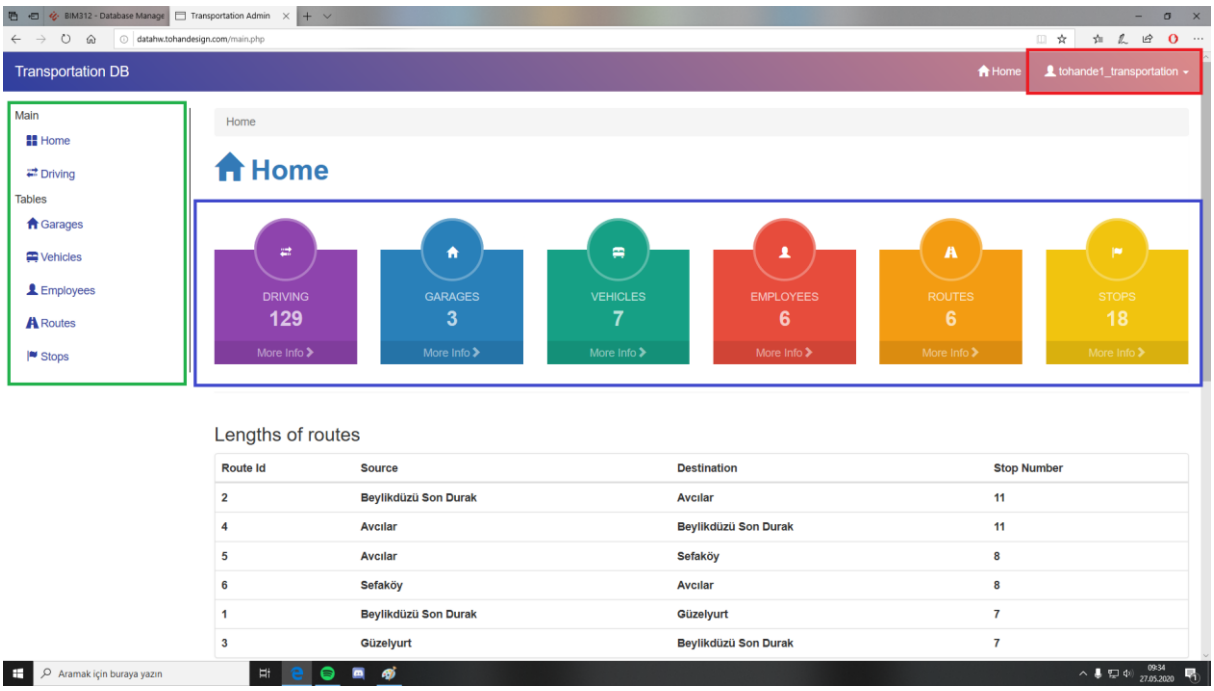


Main page

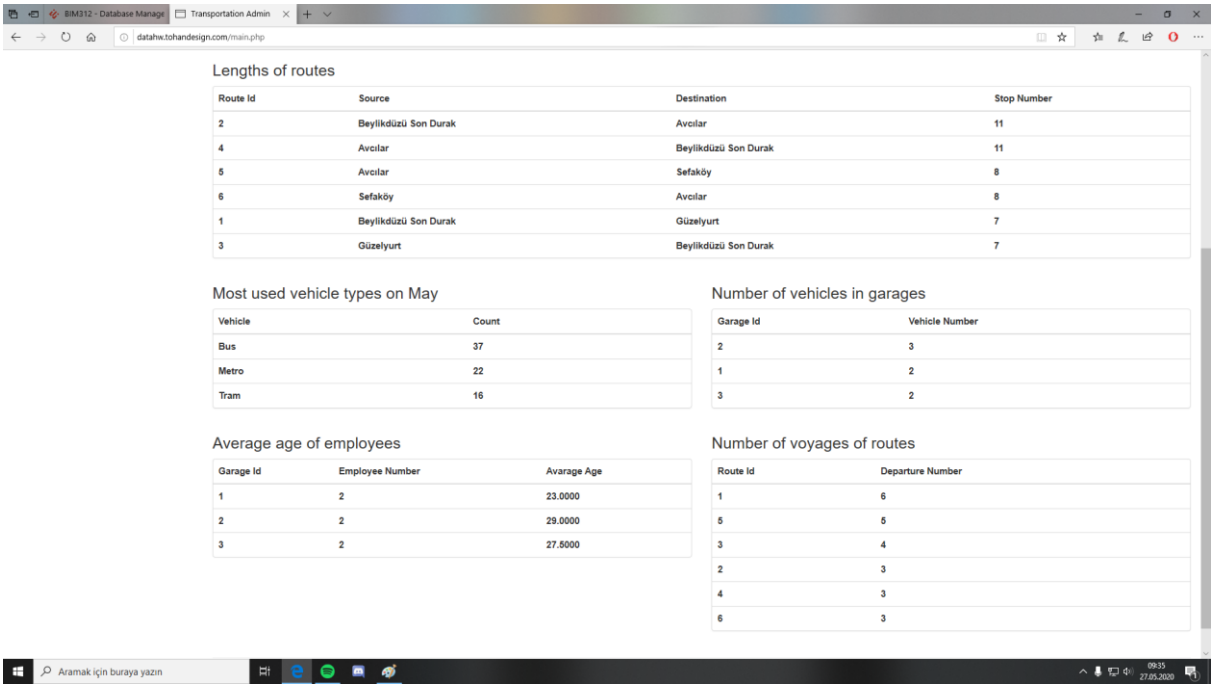


System Features

1- On the main page, you can see the user name you entered in the **red** section, and the number of data registered in the system in the **blue** section. you can navigate between the pages in the **green** section on each page.

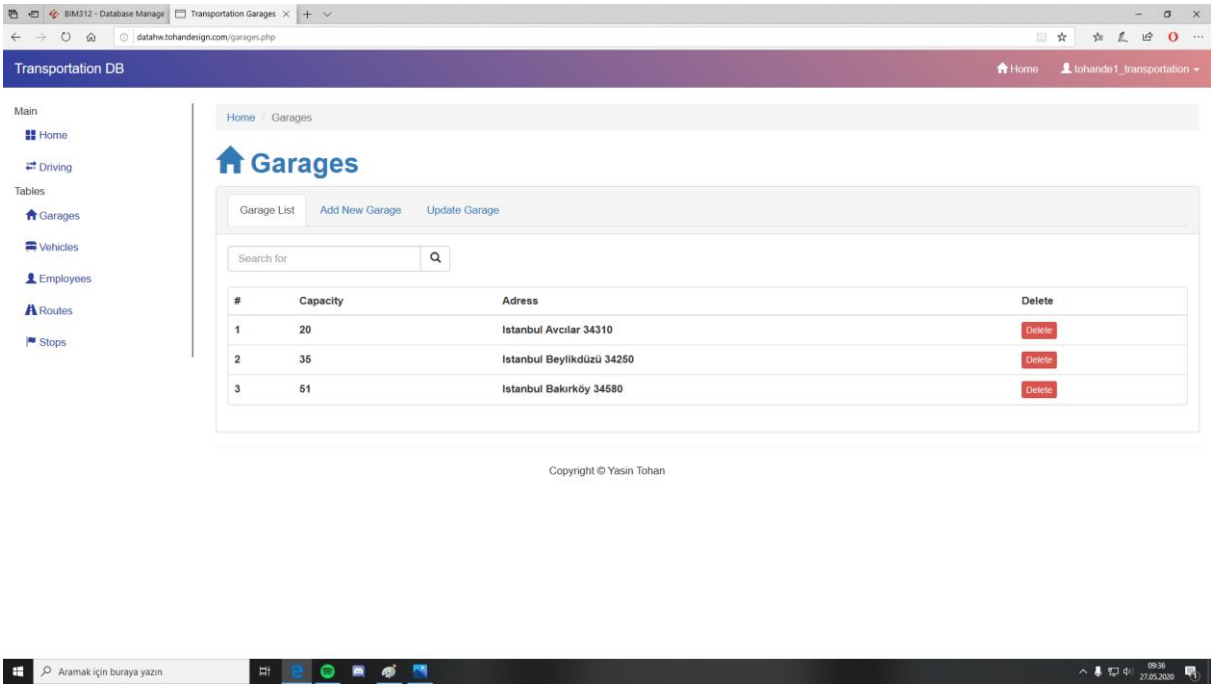


2- At the bottom of the main page, you can see the lengths of routes, the number of vehicles types used in a given month, the number of vehicles in garages, the average age of employees in garages and number of voyages of routes.

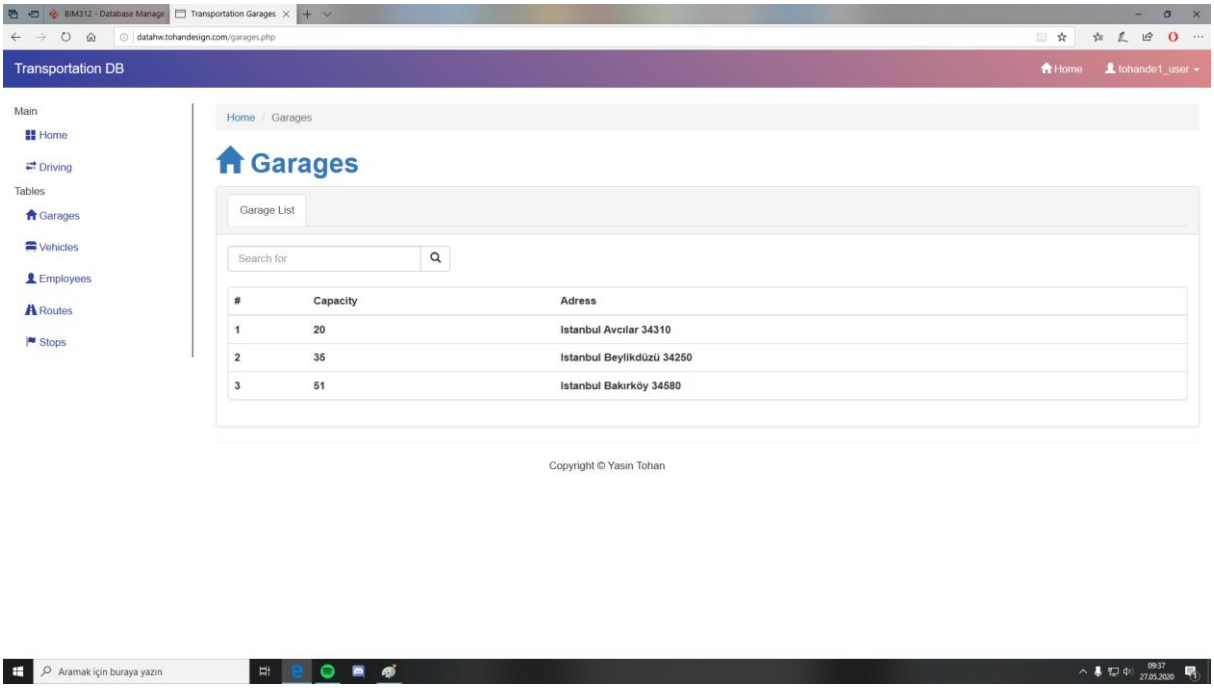


3- When different users log in, each user can only see the sections that are authorized.

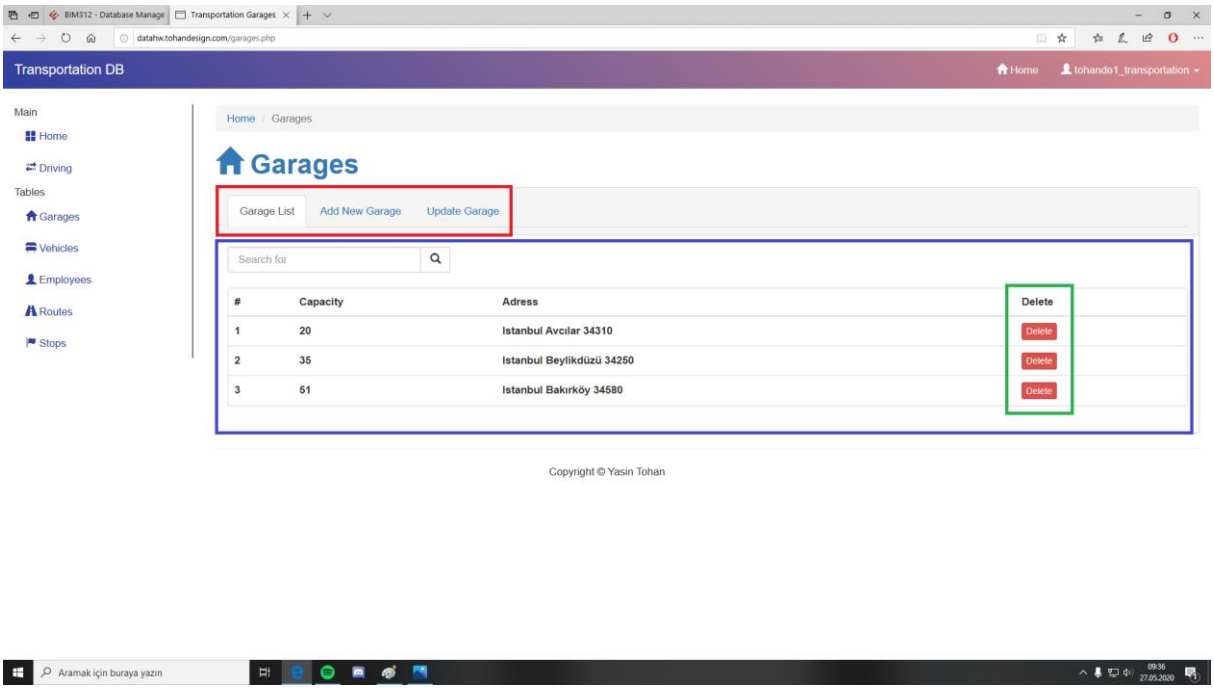
The user named **“transportation”** has every permission.



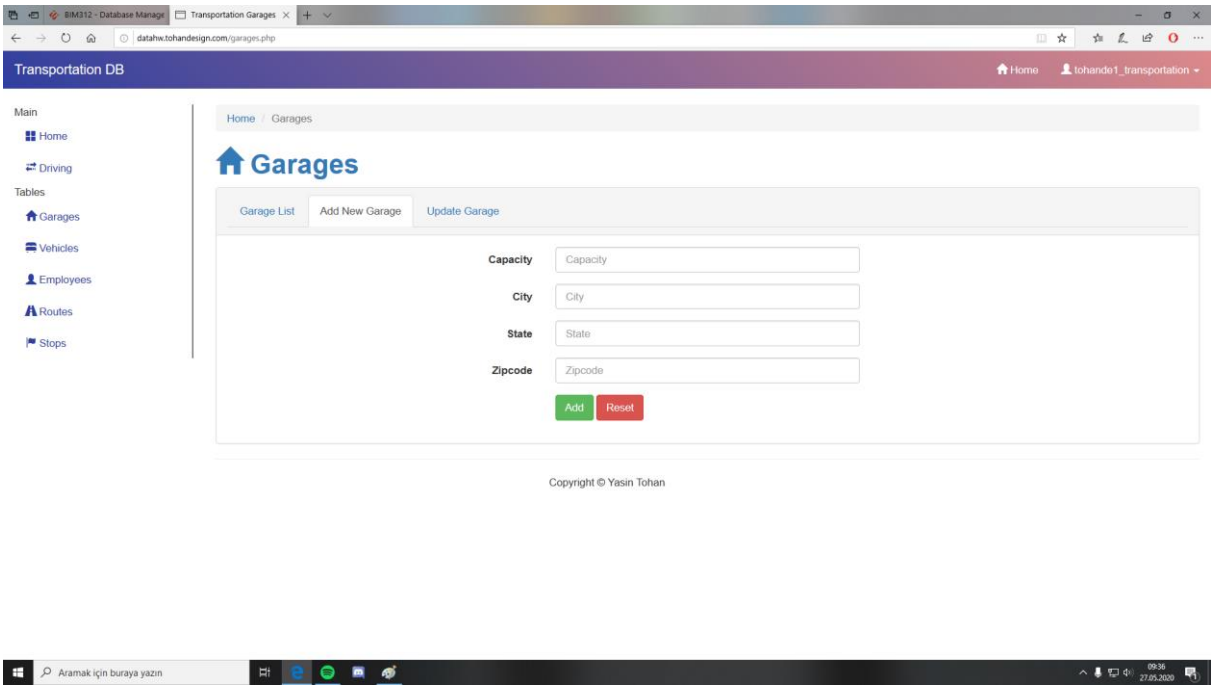
The user named **“user”** only has permission to view.



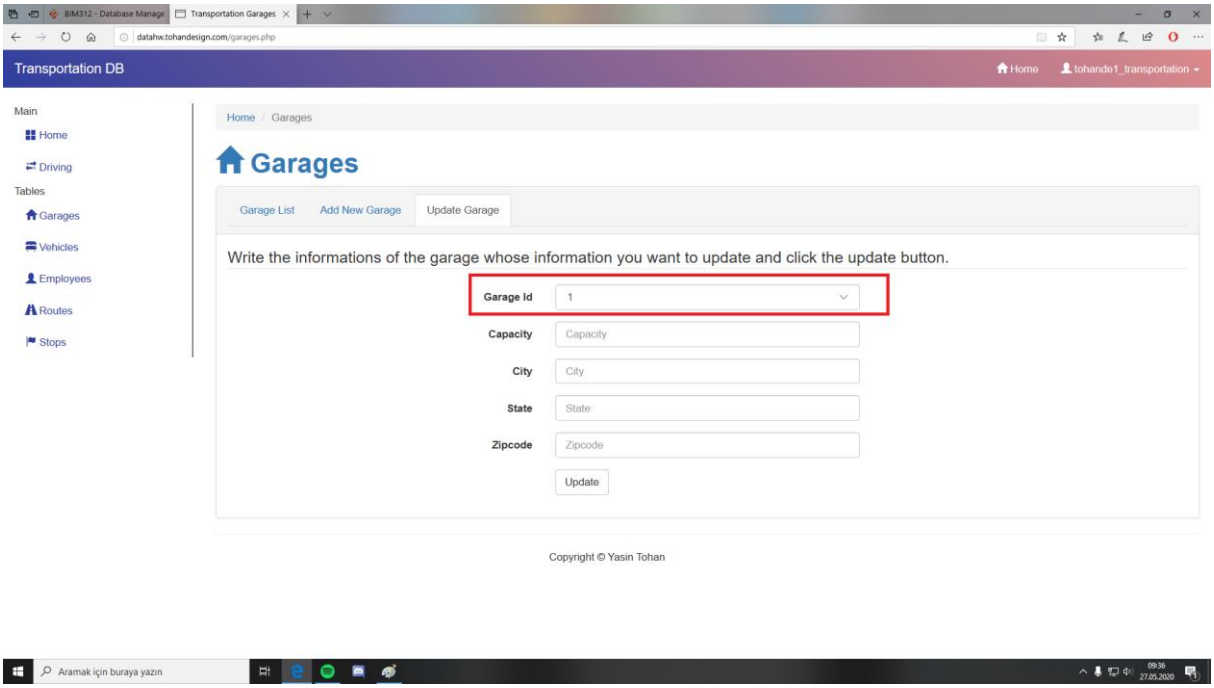
4- When you enter any page, you can change the tabs by clicking the buttons in the red section. You can see the data in the database as a table in the blue section. You can also search in the blue section. You can delete the desired data by clicking the buttons in the green section.



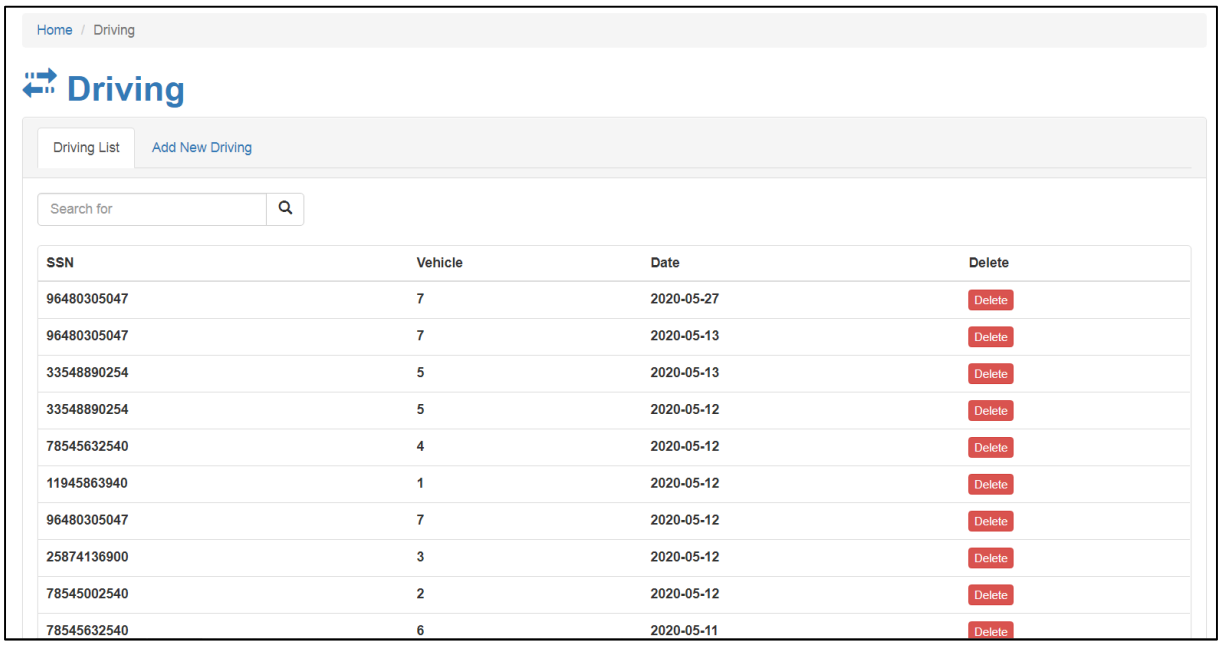
5- To add a new data, you can add it by entering the add new tab and entering the data.



6- To update an existing data, you can update it by selecting the primary key of the data you want to change from the red section and entering new information.



7- From the **driving** page you can see which vehicle was driven on which date. Also you can add new drivings.



SSN	Vehicle	Date	Delete
96480305047	7	2020-05-27	Delete
96480305047	7	2020-05-13	Delete
33548890254	5	2020-05-13	Delete
33548890254	5	2020-05-12	Delete
78545632540	4	2020-05-12	Delete
11945863940	1	2020-05-12	Delete
96480305047	7	2020-05-12	Delete
25874136900	3	2020-05-12	Delete
78545002540	2	2020-05-12	Delete
78545632540	6	2020-05-11	Delete