Ahmed Wael Mohamed & Mostafa Mohamed Salah

6704-6894

Abstract

This is a web project demanded by FOE SSP department as a DB project By Dr. Yousry Taha

WEB PROJECT

Car Rental System

**Introduction**

# This project included working on three stages :

# Identifying the idea of the project and its necessities.

# Translating those necessities to an actual form of ERD and converging it into tables using normalization methods.

# Implementing the idea and design of the site (Backend and frontend development).

**1st Stage**

# 1.Defining the problem:

# For starters, In order that the interface system to be accessible from multiple places at the same time, It should be implemented as a web project.

# In this project we have three basic pages(Login page, Admin portal, User portal).

# For that type of a system (which is closer to be a company) , We need some basic entities which are :

# Person (for holding any of the people involved with the company information)

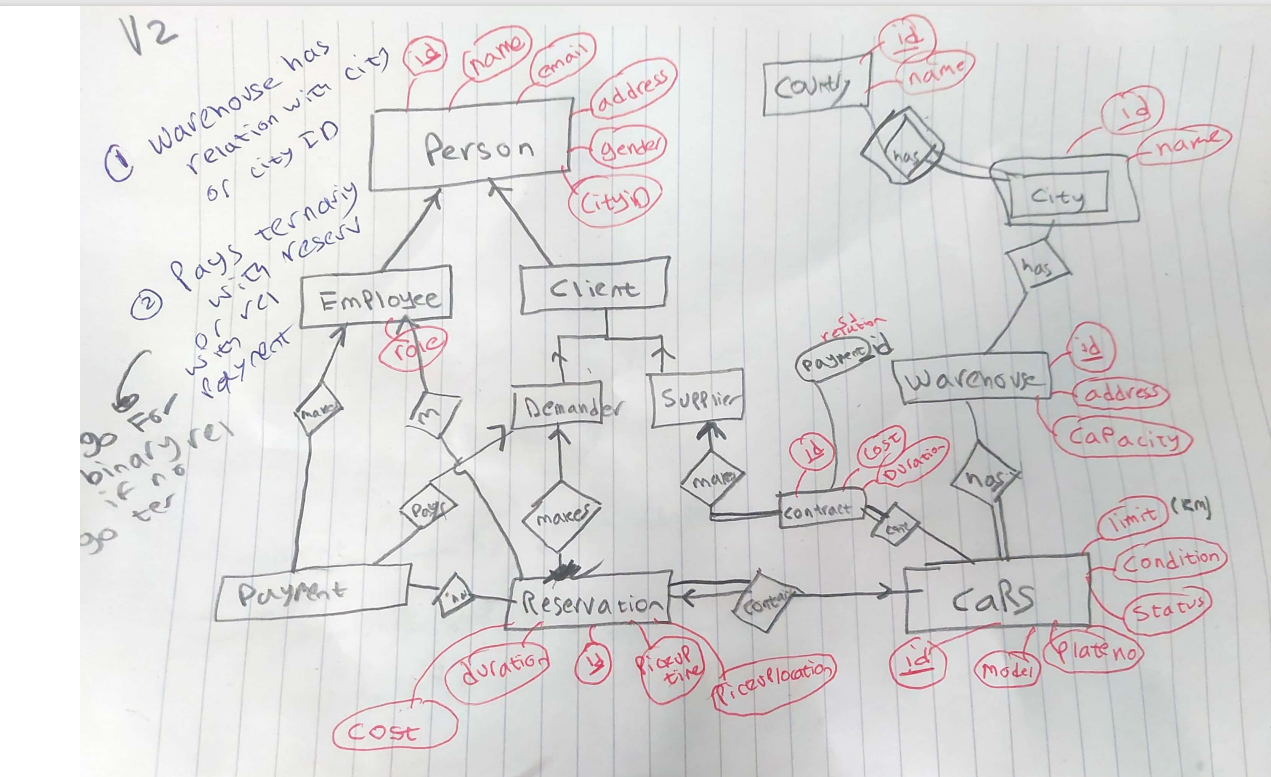
# Employee (To hold the basic info of the employee other than his personal info)

# Demander (To hold the info of our customers)

# Warehouse (Office)

**2nd Stage**

# ERD and DDL :



# **DDL:**

CREATE DATABASE carrentalsystem;

use carrentalsystem;

CREATE TABLE country(

 countryid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY ,

 countryname VARCHAR(40) NOT NULL,

 UNIQUE (countryname)

);

CREATE TABLE city(

 cid INT NOT NULL  AUTO\_INCREMENT PRIMARY KEY,

 cname VARCHAR(40) NOT NULL,

 countryid INT NOT NULL ,

 FOREIGN KEY(countryid) REFERENCES country(countryid)

);

CREATE TABLE person(

    pid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY ,

    fname VARCHAR(40) NOT NULL,

    mname VARCHAR(40),

    lname VARCHAR(40) NOT NULL,

    email VARCHAR(225) UNIQUE,

    gender ENUM('male', 'female'),

    role ENUM('SuperAdmin', 'Admin', 'user'),

    cid INT,

    mobileno VARCHAR(15),

    PASSWORD VARCHAR(255),

    BankNo VARCHAR(15),

    FOREIGN KEY(cid) REFERENCES city(cid)

);

CREATE TABLE employee(

    eid INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

    salary FLOAT NOT NULL,

    enrolled\_date TIMESTAMP,

    POSITION VARCHAR(30),

    In\_Work enum('Full Load','Free'),

    pid INT NOT NULL UNIQUE,

    FOREIGN KEY(pid) REFERENCES person(pid)

); CREATE TABLE demander(

    Did INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

    pid INT NOT NULL UNIQUE,

    FOREIGN KEY(pid) REFERENCES person(pid)

); CREATE TABLE supplier(

    sid INT NOT NULL AUTO\_INCREMENT,

    car\_count INT,

    pid INT NOT NULL UNIQUE,

    FOREIGN KEY(pid) REFERENCES person(pid),

    PRIMARY KEY(sid)

); CREATE TABLE payment(

    pay\_id INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

    amount INT NOT NULL,

    pid INT NOT NULL,

    FOREIGN KEY(pid) REFERENCES person(pid)

);CREATE TABLE warehouse(

    warehouse\_id INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

    Aval\_capacity INT NOT NULL,

    location VARCHAR(225),

    city\_id INT NOT NULL,

    FOREIGN KEY(city\_id) REFERENCES city(cid)

); CREATE TABLE car(

    car\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY ,

    NAME VARCHAR(225) NOT NULL,

    model VARCHAR(6),

    plate\_no VARCHAR(10),

    Status enum ('reserved','rented','active','broken'),

        cond VARCHAR(220),

        warehouse\_id INT NOT NULL,

        FOREIGN KEY(warehouse\_id) REFERENCES warehouse(warehouse\_id)

);

CREATE TABLE contract(

    contra\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

    sid INT NOT NULL,

    car\_id INT NOT NULL,

    cost FLOAT,

    status enum ('Rejected','Accepted','Pending'),

    start\_date DATE,

    end\_date DATE,

    FOREIGN KEY(car\_id) REFERENCES Car(car\_id),

    FOREIGN KEY(sid) REFERENCES supplier(sid)

);CREATE TABLE employee\_pay\_cont(

    eid INT NOT NULL,

    pid INT NOT NULL,

    contra\_id INT NOT NULL,

    PRIMARY KEY(eid, pid, contra\_id),

    FOREIGN KEY(eid) REFERENCES employee(eid),

    FOREIGN KEY(pid) REFERENCES payment(pay\_id),

    FOREIGN KEY(contra\_id) REFERENCES contract(contra\_id)

); CREATE TABLE reservation(

    reserve\_id INT NOT NULL AUTO\_INCREMENT,

    Did INT NOT NULL,

    startDate DATE NOT NULL,

    EndDate DATE NOT NULL,

    car\_id INT NOT NULL,

    Eid INT NOT NULL,

    PRIMARY KEY(reserve\_id),

    FOREIGN KEY(car\_id) REFERENCES Car(car\_id),

    FOREIGN KEY(Eid) REFERENCES Employee(eid)

);CREATE TABLE demander\_pay(

    Did INT NOT NULL,

    reserve\_id INT NOT NULL,

    pid INT NOT NULL,

    eid INT NOT NULL,

    PRIMARY KEY(eid, pid),

    FOREIGN KEY(eid) REFERENCES employee(eid),

    FOREIGN KEY(pid) REFERENCES payment(pay\_id),

    FOREIGN KEY(reserve\_id) REFERENCES reservation(reserve\_id)

);

ALTER TABLE reservation ADD pay\_status enum ("paid","Not") NOT NULL;

DROP TABLE demander\_pay;

CREATE TABLE demander\_pay(

Did INT NOT NULL,

reserve\_id INT NOT NULL,

pid INT NOT NULL,

PRIMARY KEY(Did, pid),

FOREIGN KEY(Did) REFERENCES demander(Did),

FOREIGN KEY(pid) REFERENCES payment(pay\_id),

FOREIGN KEY(reserve\_id) REFERENCES reservation(reserve\_id)

);

ALTER TABLE reservation ADD reserve\_status enum("On","Terminated") NOT NULL;

ALTER TABLE reservation ADD FOREIGN KEY (Did) REFERENCES demander(Did);

ALTER TABLE payment ADD pay\_dir enum ("In","Out") NOT NULL;

ALTER TABLE reservation ADD reserve\_date date NOT NULL;

ALTER TABLE payment ADD pay\_date date NOT NULL;

ALTER TABLE demander\_pay ADD UNIQUE(reserve\_id);

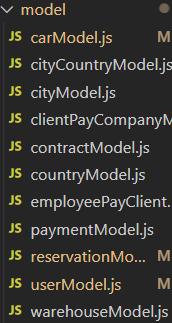
ALTER TABLE person ADD UNIQUE(BankNo, PASSWORD);

ALTER TABLE car ADD price INT NOT NULL;

**3rd Stage**

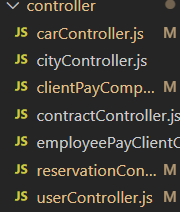
# **Backend :**

# We implemented the models needed for all the tables used in that project which are :



# This was done to include all the facilities of each part of the project in separated modules to be able to use them in multiple functions.

# For each model we defined their controllers to be able to handle the incoming request which are delivered to the frontend and routed to perform their functions.



# **Frontend:**

1) Index.html:

This is the starting page for the application and it is the login page.

The visitor choses if he is an admin or a user and it sends the data to the router to check the data.

The user can also click on register if he does not have an account and this sends him to register.html.

2) register.html:

This page asks the user for his information and registers a new account.

It takes the data, checks on it using JS such as checking if the password and the password confirmation are equal, checks on the email using regex, and checks using HTML if the username already exists.

3) checkLogin.html:

This page checks if the username and the password entered are correct or not using HTML and it keeps the user id for further use in next pages.

It checks if the visitor is an admin or not and if he is it sends him to admin.html and if he is not then it sends him to user.html.

4) user.html:

This page allows the user to search for a car to reserve and the search function is implemented using AJAX.

The user can reserve a car using its information.

The JS checks if the pickup time is after the return time and the HTML checks if the reservation is possible.

The user can pay for reserved cars by clicking pay, which sends him to payment.html.

5) payment.html:

This page shows all reservations made by the user and not paid yet.

It allows him to pay by clicking the button and updates the cars reservations as being paid.

6) admin.html:

This page is the admin portal where the admin can choose between different options:

1. Advanced search

b- Add car

c- Edit car

d- Reservations

e- Reports

7) advancedSearch.html and advancedSearchPage.html:

These pages uses Ajax to search for values needed by the admin.

8) addCar.html and addCarPage.html:

These pages allows allow the admin to add new cars to the system.

9) editCar.html and editCarPage.html:

These pages allows allow the admin to edit cars in the system.

10) reservations.html:

This page shows all reservations made in the system.

11) reports.html:

This page allows the admin to select from 5 different reports:

1. Reservations made during specific periods for a specific customer
2. All reservations made during specific periods
3. Get the status of a specific car
4. All reservations for a specific customer
5. Payments made between a specific period

12) specificPeriodCust.html:

This page takes a customer id and a time period to show all reservations made by that customer in that time period.

13) specificPeriodNoCust.html:

This page takes a time period to show all reservations made by all customer in that time period.

14) getCarStatus:

This page shows the status of a car at any given day.

15) customerRes:

This page takes a user id and shows all reservations made by that user.

16) dailyPayments:

This page takes a start and end date and show payments made in that time period.