**Capstone Project**

SUMMATIVE CAPSTONE PROJECT

Project Title

**Car Rental Management**

**Start Date**: 9th February, 2021

**End Date**: 17th May, 2021

Submitted By

Matlapudi Lakshmi Tulasi(530F)

Introduction

1.1 What the Problem is

A new up and coming car rental service is wishing to have a user interface that will allow their

Customers to view the models, descriptions and prices of different cars available.

The Web App will be responsive, allowing for the customer to view it on any device, from tablets to

Mobile phones and desktop computers. The administrator will also be able to login through the

Same form but have the ability to add/remove new car rentals, change prices, and so on. Potential

Customers should be able to view all the cars available to rent even without logging in as well as

Rent without having an account, though the option is provided upon checkout.

### 1.2 Goals for the Project

This software, called ‘CAR RENTAL MANAGEMENT, will allow for the company to access their

database securely and safely in a user-friendly online environment. Allowing for them to change car

Information with ease. The software will be in sync with the both the Web App, allowing for

real-time up-to-date services for their customers.

Both registered and non-registered users will be able to search car rentals by price, model,

seating and any other potential searches. They will also be able to select and pay for the

service.

The company would instantly get that service demand through their 24/7 car rental support built

directly into the software and either reserve the car for pickup or send out a pickup truck carrying

the new rental car to the desired location upon time request.

There should be a Web App version for the software to connect with for those who wish for a

quick car rental servicer. The Web App will be responsive to any device using it. This allows for

consumers to access the service from any sort of hardware device: tablet, computer, mobile

devices, and so on. The software itself will be available on all computer platforms that are

running any aspect of Linux, Windows 7, Windows 8, and Mac operating systems.

The software will also be easily available to Windows 10 when it comes out later in the year.

Besides, computers, the software has a minimal version for tablets for those working for the

company to easily navigate through customer orders when they are on the road or simply away

from the computer. Tablets are not required for full use the software however, in case the

company has a budget that does not allow for them.

1.3 Motivation for the Project

With this company in need of a better system, we felt it was our obligation to help them in their

time of need. To develop such a system that would not only ease the burden on the company’s

customers, but the company itself. Our team has an immense amount of knowledge when it

comes to problem solving, programming, and communication. Not only would we strive to give

the car rental service everything they desired, but we will continue to make sure the software is

at its very best and beyond. Each one of us will always and will continue to give 100% and more

to making the transition a breeze for the car rental service.

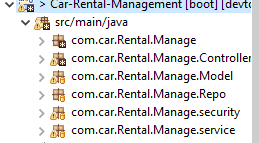
**Technologies**

The goal of this application is to get an in-depth feeling for the usage of the Spring Framework. The following subjects were applied inside this project:

* Spring Boot
* Spring Security
* Spring MVC
  + Thymeleaf
  + [Thymeleaf Layout Dialect](https://ultraq.github.io/thymeleaf-layout-dialect/)
  + Bootstrap 4
  + CSS
  + HTML5
  + JavaScript
* Oracle SQL(12c)

# project structure

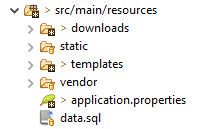
### src/main/java/...



* Manage: Contains Main method with @@SpringBootApplication
* security: Contains all configuration classes. In particular messages and security.
* controller:
  + view: Classes annotated with @Controller, which are used to return dynamically rendered HTML views as Endpoints.
  + File Upload Util class that helps to upload files and reduce code repeating.
* Model: Contains all POJO entity classes.
* Repo: Contains all repositories to access the database.
* service: Contains all service classes which perform further business logic.

### src/main/resources/...

Contains templates, insert scripts and configuration files.



* data.sql: Used for initial test data.
* application.Properties: Contains the server configuration.
* templates: This package consists of the root layout file index.html (using the Thymeleaf Layout Dialect). This root layout consists of fragments, which are seperated in static content that should be visible on every page and dynamic content that should only be rendered depending on the current request.

# Domain model (JPA entity classes)

The main entity class is Booking.java. This class contains several unidirectional @ManyToOne mappings to the other entities. It was decided to do unidirectional mapping because any sort of dependency to other classes is done through this class only. Thus, all other classes do not need an extra reference to this class.

# Authentication

## - Roles

There are three access types (user roles) in this application:

### USER

By logging in as a user that has the role of ROLE\_User(Employee of company) the user can issue new rentals. Also, the employee has the possibility to finish those. Employees can access all paths starting with /booking/\*\* and add customers for car booking purpose.

### Admin

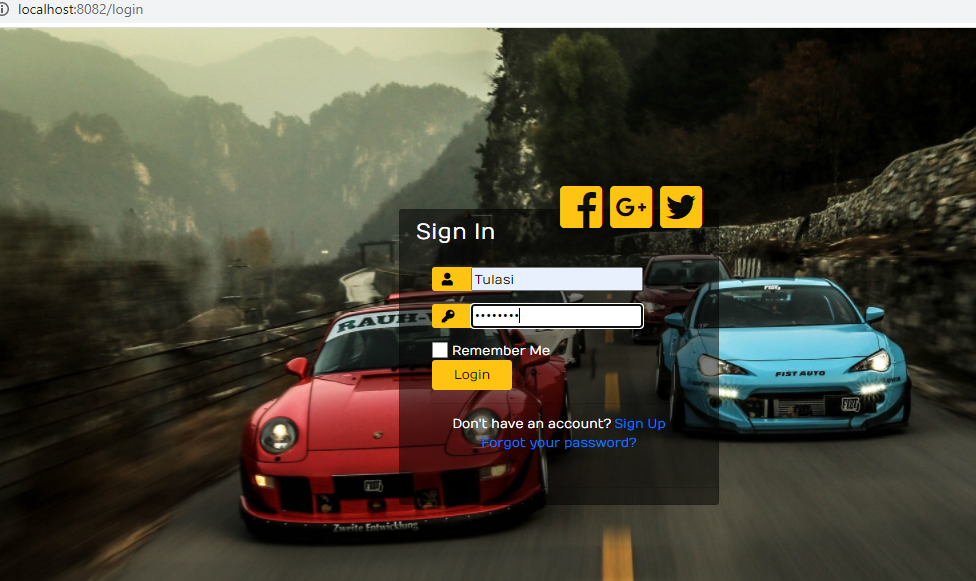
Users with the role of ROLE\_ADMIN have access to several REST endpoints. Admins can access all paths starting with / \*\*. Admins can have access to add new employees, modify and delete employees and have access to other REST endpoints.

### Manager

Manager with the Role \_Manager have access to add new cars to the list, add customers for car booking service.

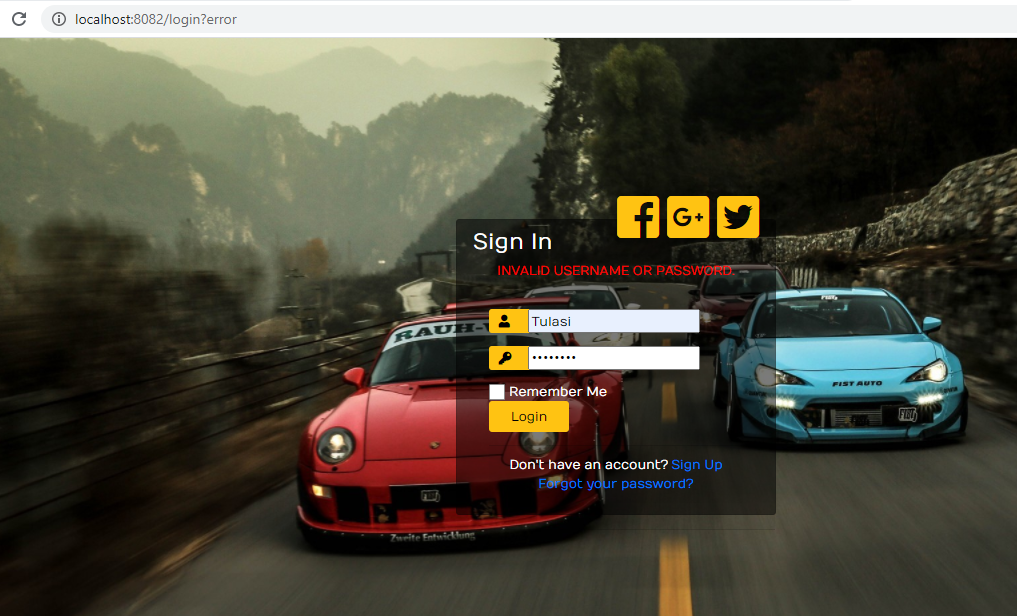
## - In memory users

To keep things simple, two users were created with Spring Security HTTP basic auth. (username, password)

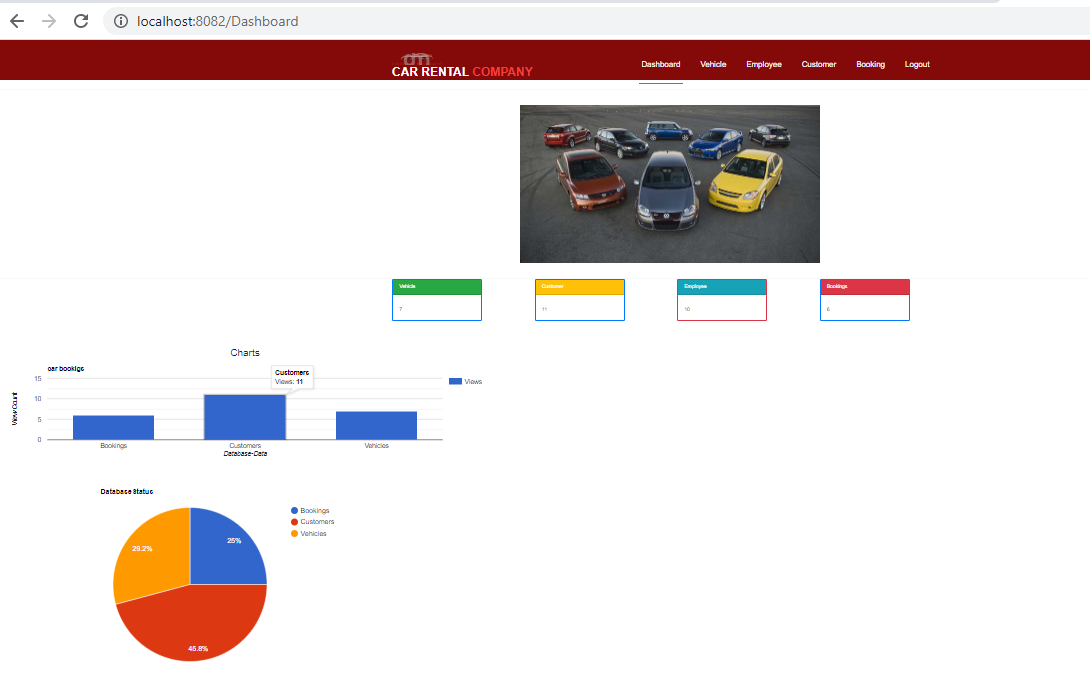
* ROLE\_User: This user has (user, user) as credentials.
* 
* ROLE\_ADMIN: This user has (admin, admin) as credentials.

### Redirection

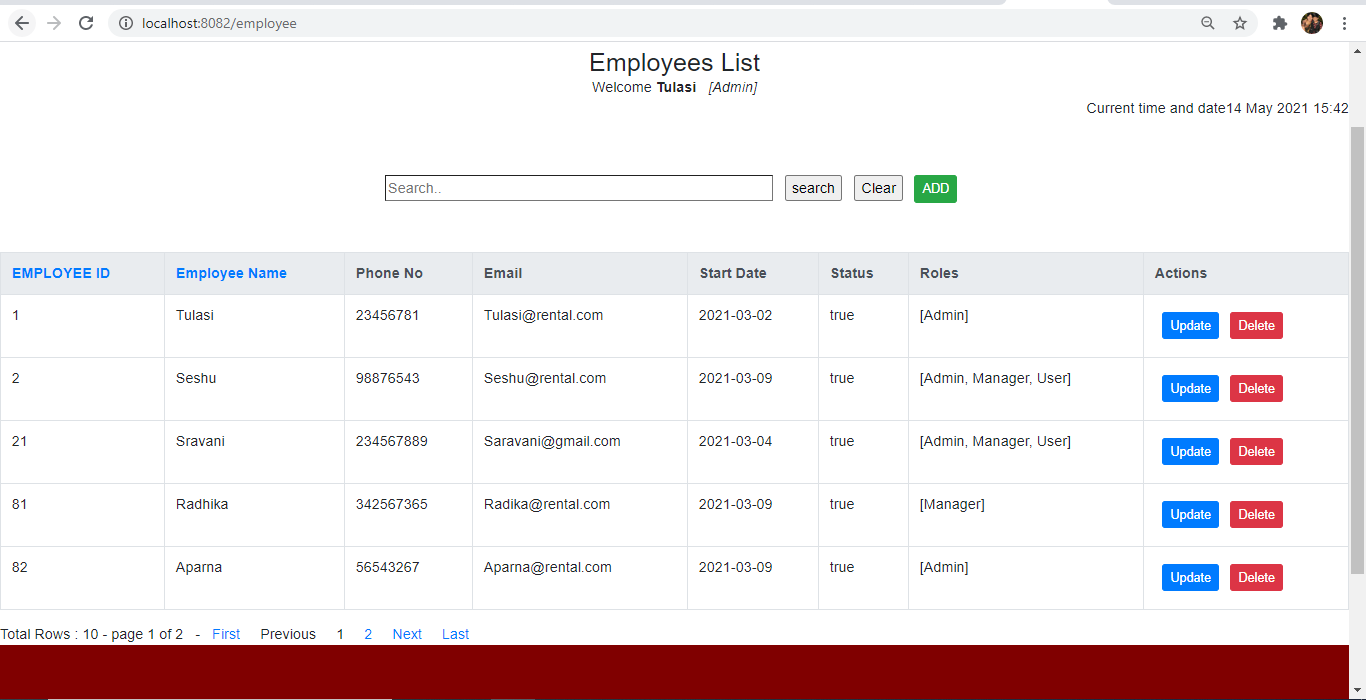
If unauthenticated users try to access paths of a specific role, which they are not part of, they will be redirected to the login screen.



After Login As Admin you can see Dashboard Screen. In this You can view Database data with piechart and bargraph.



When sign in as Admin ..Admin can able to add New Employee, Update Employee details and delete Employee



When login as Employee(User) you can view only Employee list.. unable to do crudOperations.

