# A picture containing logo Description automatically generated

# Ren a Car Web (MERN stack) - USER GUIDE

Hammada Ballali Ebbi – 20203543

CSCCORE1H001AZ2022/3 -- Coursework – Advanced Web Development

## Description:

The Rent a Car Website is a web application built with the purpose of serving as a platform to expose and manage a Rent a Car business (Khalilocation). The application fits the User requirements:

# User Stories for Khalillocation Website

1. Implement a Website to manage a rent a car business

2. Add a public facing page with basic contact info

3. Add an employee login to the notes app

4. Provide a welcome page after login

5. Provide easy navigation

6. Display current user and assigned role

7. Provide a logout option

8. Require users to login at least once per week

9. Provide a way to remove user access asap if needed

10. Notes are assigned to specific users

11. Notes have a ticket #, title, note body, created & updated dates

12. Notes are either OPEN or COMPLETED

13. Users can be Employees, Managers, or Admins

14. Notes can only be deleted by Managers or Admins

15. Anyone can create a note or a new car.

16. Employees can only view and edit their assigned notes, but they can edit any car.

17. Managers and Admins can view, edit, and delete all notes and cars.

18. Only Managers and Admins can access User Settings

19. Only Managers and Admins can create new users

20. Desktop mode is most important but should be available in mobile

31. Car description includes Model, Brand, Type, Year and picture.

## Tools:

This application has been built using MERN stack tools:

* Node.js
* Express.js
* MongoDB
* React.js
* Redux.js

## Table Description automatically generated

## Implementation:

The code consists in a Backend and a Frontend:

* Backend: It creates a server to store and collect data from a database built in MongoDB. It consists in:
* Config folder: It contains the files that allow to set cors and allowed origins configuration.
* Controllers folder: The controllers files are used to define the way in which the components are built.
* Logs folder: The logs files allow to track the errors and requests.
* Middleware folder: Stores the files that handle incoming HTTP requests and prepare outgoing HTTP responses.
* Models folder: Specifies the structure of the components.
* Node\_modules folder: All node libraries.
* Public folder: Store the css file.
* Routes folder: Store the route files.
* Views folder: Stores the html files.
* .env file : Stores the global variables.
* gitIgnore file
* package-lock.json : Dependecies.
* package.json : Dependencies.
* server.js file: Implements the server.
* Frontend: It gets the data from the server and allows to visualize it. It consist in:
* Node\_modules folder : Node Libraries
* Public folder :
* Src folder
* gitIgnore file
* package-lock.json file

## 