# CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

A Project Report On

**Car Retailers Management System**

(**Web application platform for selling and buying used cars.**)

### SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF

PG DIPLOMA IN ADVANCED COMPUTING

FROM C-DAC Knowledge Park Bangalore

**Dated Submitted By-**

August 2019 Abhijeet Srivastav (DAC 01)

Anuraag Patil (DAC 17)

Ashish Kumar Manjhi(DAC 19)

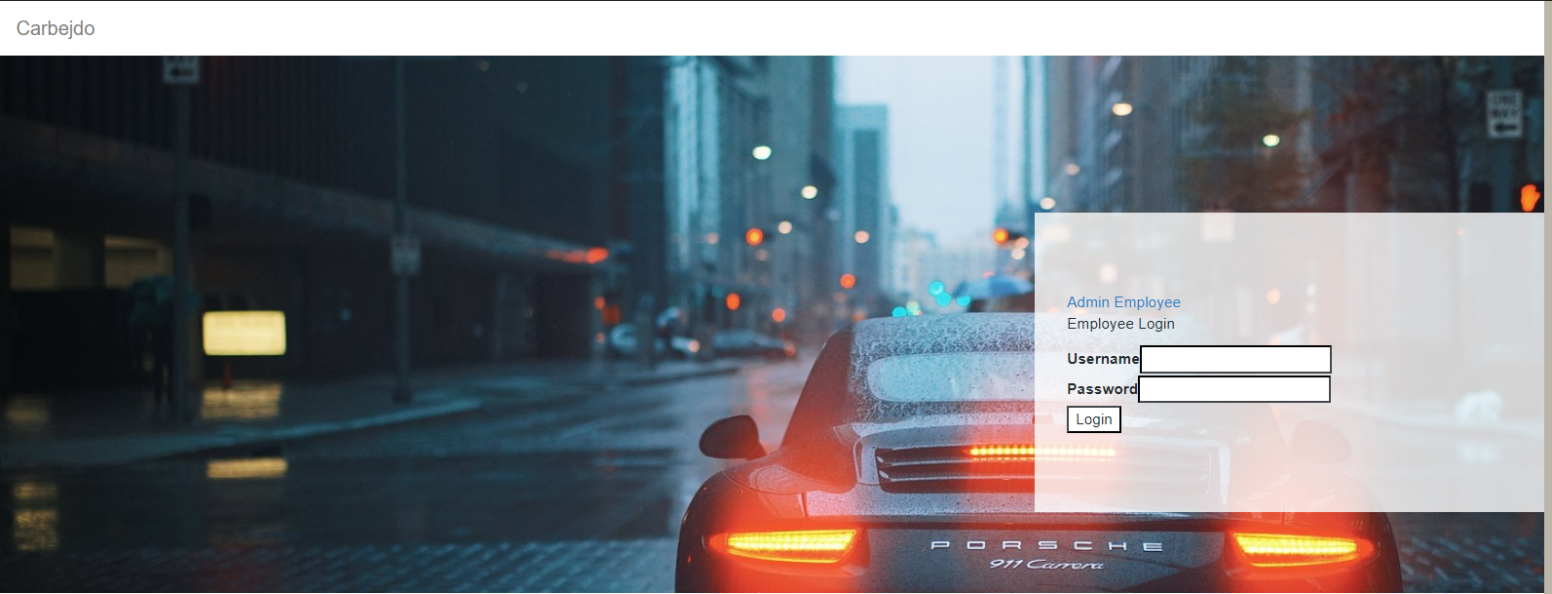
Jijnas S (DAC 36)

Swapnil Bhagat(DAC 104)

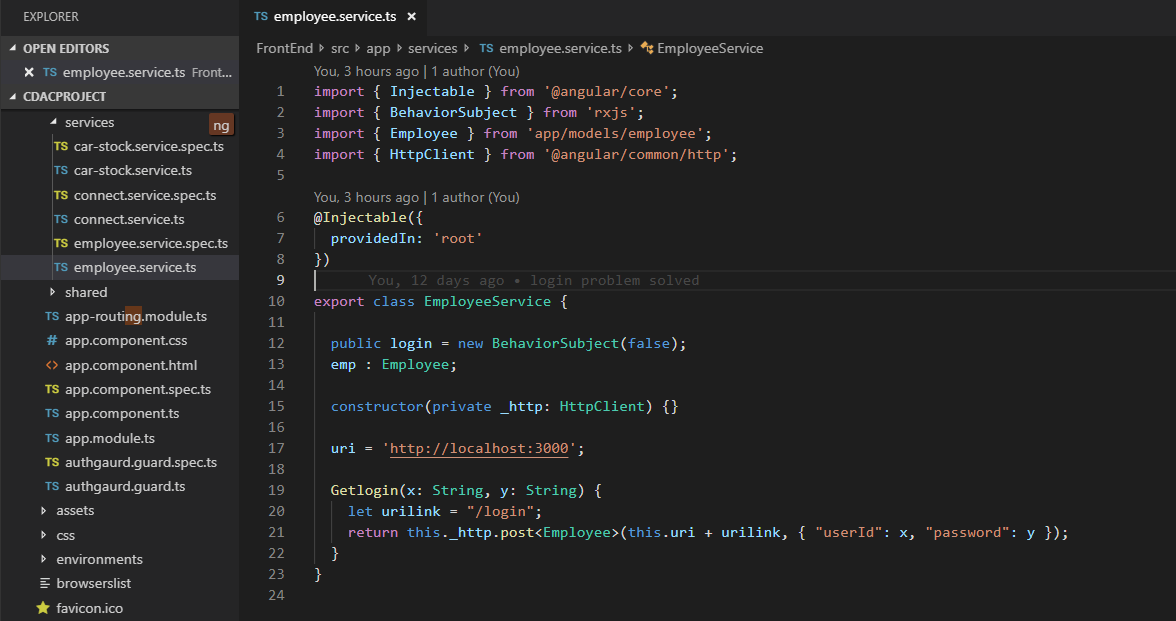
**Project Overview**

A Web application for managing the operations for second hand car retailers. It is implemented through REST API with Angular on Presentation Layer, NodeJS on the Service Layer and My SQL on DAO Layer. On **Presentation Layer** we provide a management system where there are two kinds of users namely:

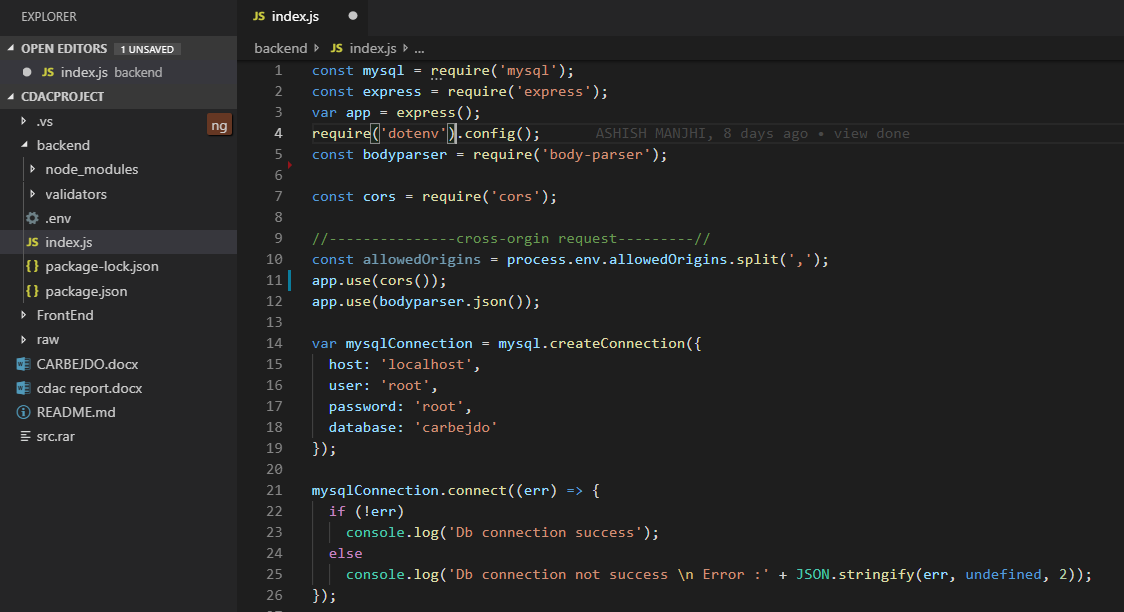
* Administrator.
* Employee.



Both the users have their own dashboards with common functionalities to manipulate details on customer list, car stock, and transaction history of the company. Apart from the common functionalities administrator will have a privileged Dashboard which deals with the special function like manipulate Employee List, add new employee etc. Also Employees will have a dashboard which deals with the updating of customer and car details needed for a new transactions and hence the invoice generation for the respective transactions.



On **Service Layer** we’ve provided the REST API on Express at NodeJS server which will handle the request from the presentation layer for login, manipulating details of transactions, employee list, and customer list and car stock from the tables of the database.



On **DAO Layer** MySQL Database has been integrated with the Service part where tables for customer, cars, employee and Invoice has been developed and made dependent by proper normalization procedures

## Introduction

* 1. **Purpose:**  Provides active web Application for the second hand Car Business Management which provides functionalities to manage and manipulate car stock, employees and other important entities.

## System Design:

* 1. **Class Diagram** - Class diagram is a static diagram. It represents the static view of an application.
  2. **ER Diagram**- ER Diagram is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities.

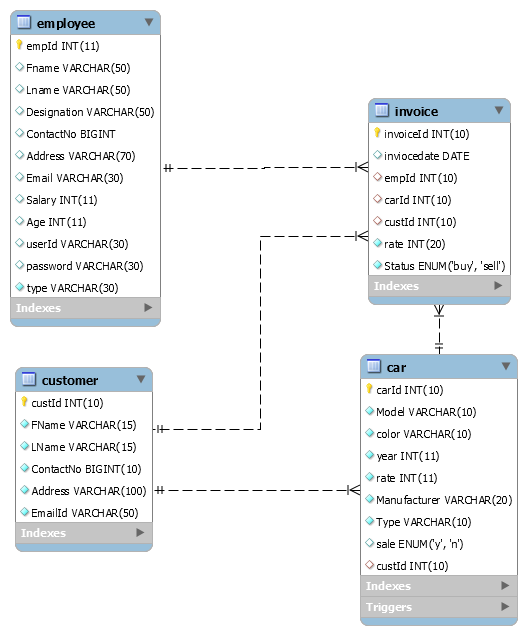


Figure 1

1. **Analysis:** The based on the existing system we’ve modified the way of operations in the Presentation layer and integrated the Service layer with the MySQL database for the proposed system.

# Technology used-

* 1. **MySQL** - It is an open source relational database management system which contains the database.
  2. **Angular –** The JavaScript MVC framework used for web applications It is a JavaScript framework for building web applications and apps in JavaScript, html, and Type Script, which is a superset of JavaScript. It’s the client side service which in connection to the server side helps the client to operate or use the application.
  3. **Node JS**. – Framework used for scalable server-side and networking applications. It’s the server side service which in connection to the client service is used by the client side to operate and perform various actions on the application. It can generate dynamic page content and used in back-end run time environment which facilitates event driven, asynchronous programming that makes it fast. The authentication part is done in this.
  4. **Express.js**- Express.js is a Node JS web application server framework, which is specifically designed for building single-page, multi-page, and hybrid web applications. It has become the standard server framework for node.js. Express is the backend part of something known as the MEAN stack. The Rest API are made using express js in the server side to perform different actions.
  5. **Github Repository Link** : <https://github.com/cdacactskp/CdacProject>

1. **Flow of Execution**

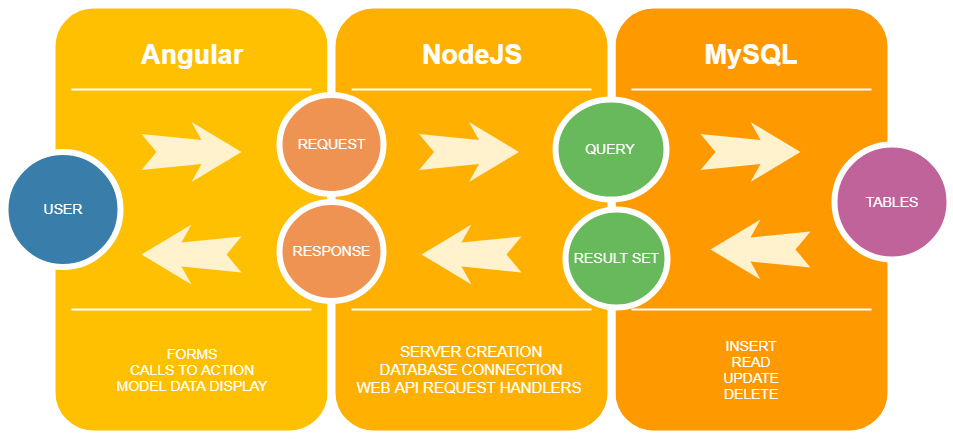


Figure 2

1. **Conclusion:**

• We learned to connect two servers front-end and back-end via HTTPCLIENT.

• Learned to use express JS to create Rest API.

• How to implement routing of Components and services in Angular.

• Learn to use different modules of Node JS to validate, protect the application servers and others for smooth implementation of different request of the client like Mysql, nodemon, cros, Body-parser.

• Used Mysql server for the database management of our project and also learned how to use the relational database.