****

**Athlone Institute of Technology**



**COURSE:** Master Of Science In Software Engineering

**STUDENT NAME:** Brendan Kenny

**STUDENT ID:** A00037189

**COURSE ID:** AL\_KSWFT\_R\_5G

**SUBJECT:** Rich Internet Applications

**YEAR:** 5

|  |  |  |
| --- | --- | --- |
| Student Number | Student Name | Signature (for hardcopy) |
| A00037189 | Brendan Kenny |  |

Contents

[***Introduction*** 3](#_Toc449709687)

[***Overview Of Website*** 4](#_Toc449709688)

[***About*** 4](#_Toc449709689)

[***Cars*** 5](#_Toc449709690)

[***Contact Us*** 6](#_Toc449709691)

[***Featured*** 6](#_Toc449709692)

[***Gallery*** 7](#_Toc449709693)

[***BackBone.*** 8](#_Toc449709694)

# ***Introduction***

The development of this website utilised a RESTful api, full CRUD functionality, modals, featured products, gallery and implement BackBone.js. The idea behind the site was to create a website for a car dealership called KennysCars.

Brendan Kenny

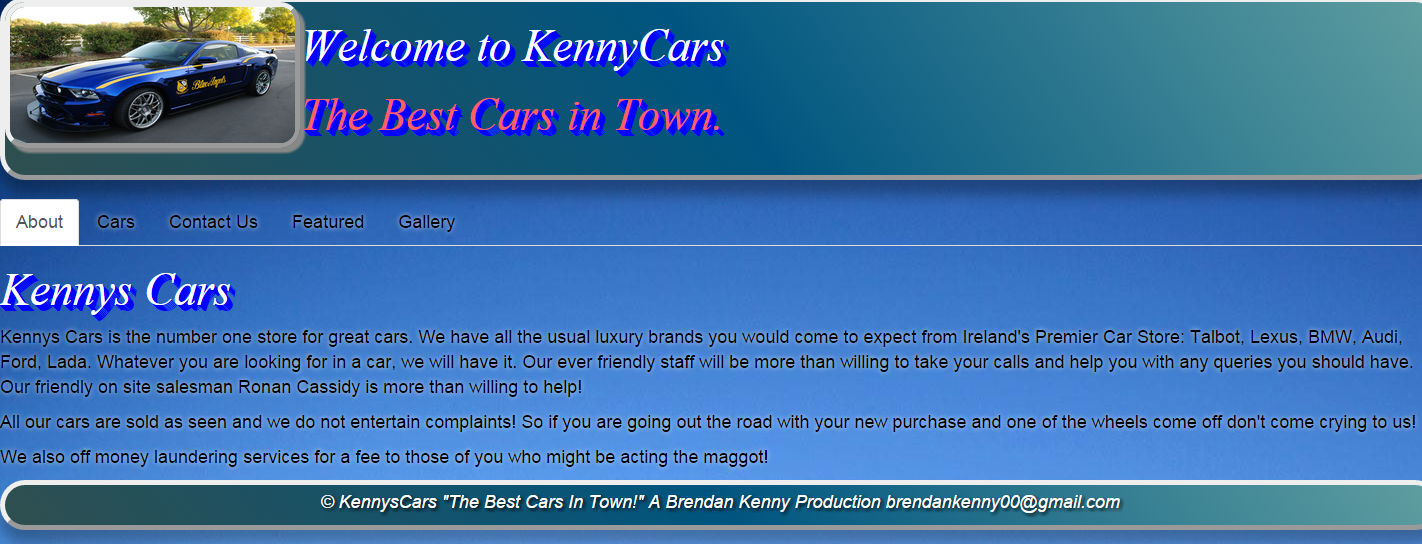
27 April 2016

# ***Overview Of Website***

The theme of my website is a car dealership ***KennysCars*** and is made up of five sections organised into the following tabs:

* About
* Cars
* Contact Us
* Featured
* Gallery

## ***About***



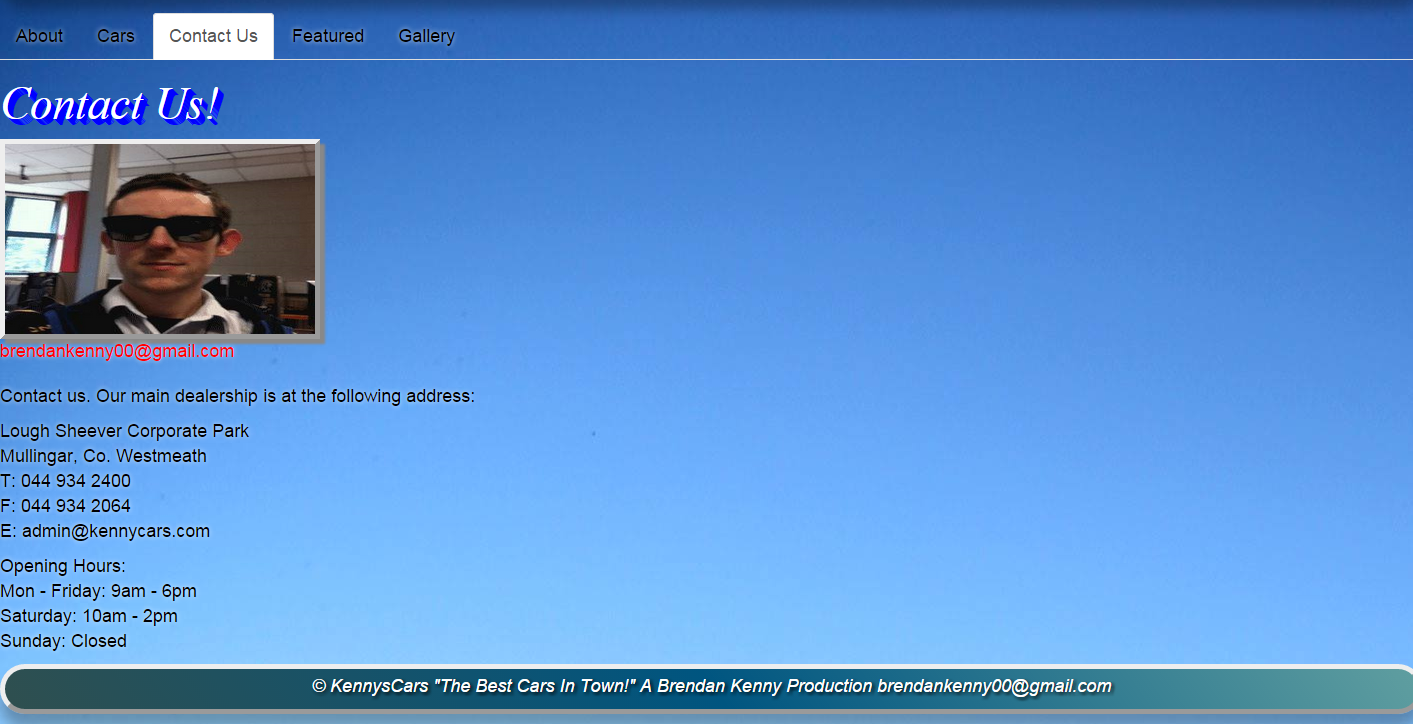
This page is the first page that a person will see when viewing the website for the first time. It gives the reader an overview of the business.

# ***Cars***



The cars page displays all the records in the database for the cars in stock. The user can search for a car and can apply the various CRUD options to update, delete or add a new record to the database by clicking on the edit option. The description option when clicked will give the user a brief overview of the car in question and also a picture of that car.

# ***Contact Us***



The contact us page gives the user useful information regarding how to contact the dealership should they wish to contact us to compliment us on the great business we are doing. Also the featured page displays some of the best cars we have in stock.

# ***Featured***



# ***Gallery***



The gallery page displays some more nice cars in a nice slider.

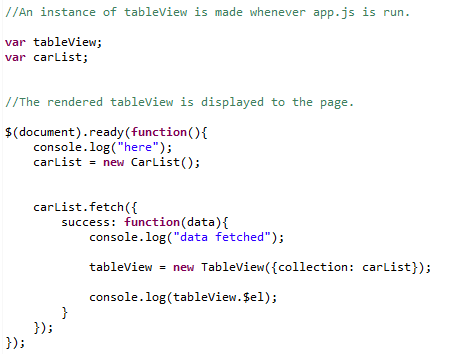
# ***BackBone.***

The whole idea of BackBone is to add better organisation to the developers approach in the website building process. With BackBone, we see the separation of concerns, in that business data is isolated and organised into *Models* and interfaces are organised into *Views*.

The benefits of BackBone become apparent when one considers the amount of code involved in a large website. In my opinion a framework such as BackBone would make life much easier for the developer due to the better organised nature of the code.

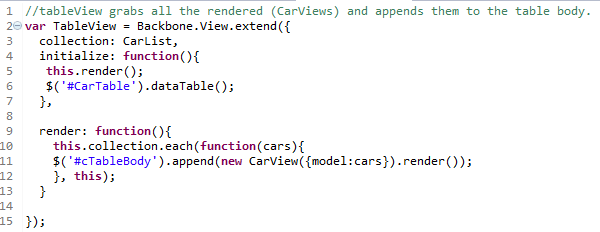
To give an idea of how BackBone works, I will go through how I implemented BackBone in the development of my website.





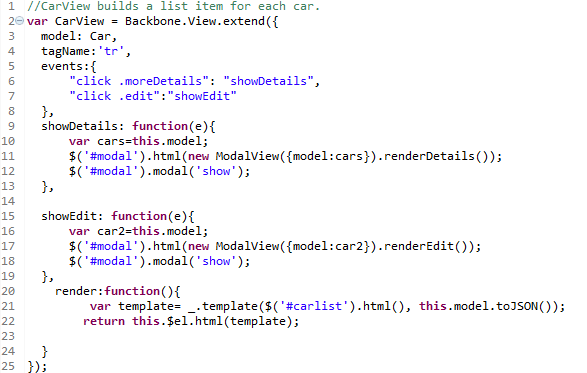
The app.js initiates the tableView when the website is first ran. The carList as defined in the Models.js is passed to the tableView. So, in other words, the business data is applied to the view.





The tableView in turn is concerned with the rendering of the datatable and grabs all the rendered (CarViews) and appends them to the table body. We can see that the model which deals with the data records is passed to the view and then rendered.





The CarView is concerned with building a list item for each car. So we can see here that I have a showDetails function and a showEdit function which are concerned with the rendering of the Modals for editing and displaying details about the cars. So if a person clicks on the moreDetails button, the Details modal will be displayed complete with the relevant info on that car.





The models.js is concerned with initialising the business data from the database as identified from the rootUrl.





The modalView is concerned with the CRUD actions that a user may wish to perform. For example, the user may wish to update a record, delete a record, or add/post a new record to the database. The events are: newCar, delete, save. So if a user wishes to delete a car, they will click on the delete button in the Edit modal. The deleteCar function will then be employed and the record in question will be deleted from the database. The newCar function simply blanks out the various fields on display for that modal.