Department of Computer Engineering 

University of Puerto Rico

Mayaguez Campus



**Team Members:**

Kevin Santiago Ortíz

José A. Rodríguez Rivera

Jariel O. Laureano Cruz

**Emails:**

[kevin.santiago1@upr.edu](mailto:kevin.santiago1@upr.edu)

[jose.rodriguez37@upr.edu](mailto:jose.rodriguez37@upr.edu)

[jariel.laureano@upr.edu](mailto:jariel.laureano@upr.edu)

**Introduction**

We live on a society that depends on science and technology. As technology advance our goal is making things easier. Things like cars, houses, and clothes advertisement are now performed online. The reason is not a mystery, is a cheaper and more efficient way to do it.

Moving on this line of reasoning, DealIt is a web application that offers a user friendly interface were can be placed sales/trades and special offers. The sales/trade transactions could be anything such as: vehicles, houses, electronics, furniture, and miscellaneous. As mentioned before sales/trades could be anything, so this web application targets everyone willing to sale/buy/trade something that fits in one of the above categories.

The technologies employed to develop DealIt are as follows:

*Front End*

* Angular JS
* Javascript
* Html
* Css
* Angular Material

*Middleware*

* Java Play

*Back End*

* PostgreSql

**Client App Description**

DealIt web application will have the following capabilities:

1. *User registration:*

Users that want to sign up would need to supply some information such as:

full name, city, cell phone number, email and a password. After submitting their

credentials a notification for email validation would be send for completing the

process. Another option

b) *Selling:*

For posting something for sell, a form will be supplied. To those that are

already registered, the contact information will be automatically fill up. In

change, those that want to continue as guest will need to supply all the required

data, and a post code will be send via text to the user for future post editing.

A description field would be supplied and the option for uploading a picture of the

item.

c) Buy/Trade:

A few methods will be supported in terms of buying or trading items. From

the most traditional ones such as: contacting the owner through the information

supplied on the advertisement. Also is possible to pay through credit card in

some cases. All the above mentioned would depend on the product owner and

obviously what classification we are talking about.

d) Product Filtering:

The interface will count with a filter that would be capable of search trough

the classifications making easier find things within the web app.

**Server App Description**

The server side of the web application will make possible the communication between the client and the database. In this way the information from the item and the people who own them will be stored at the database. The client side of the application would be worked using Angularjs which facilitates a lot things like the ajax calls for getting or posting data and other things such as the interface response.

Our database system is PostgreSql based and will probably have some tables like the ones described below:

*Registered Users --* Will provide an ID and store all the information from the user. Will have

have an active column to distinguish between the active users, and the

inactive ones.

*Account* – Contains information about the products that the user owns, and the orders he has made.

*Deleted User --* The deleted registered users will be stored here (Their ID) after setting their

active column to false on the Registered Users table.

*Products --* The items posted will be stored here. Each item need a classification field that has to be consistent with the available classifications on the Items Classification table. As the Users table this table will also have a active column.

*Product Details* – All information about a product, description etc.

*Product Classification --* This table will contain all available classifications assigning them an

ID that can be used for the classification field on the Items table.

*Product Deleted --* The deleted items post would be stored here (Item ID) after setting their

active column to false on the Items table.

*User Info --* This table will bind each user with the credit card that each one supplied.

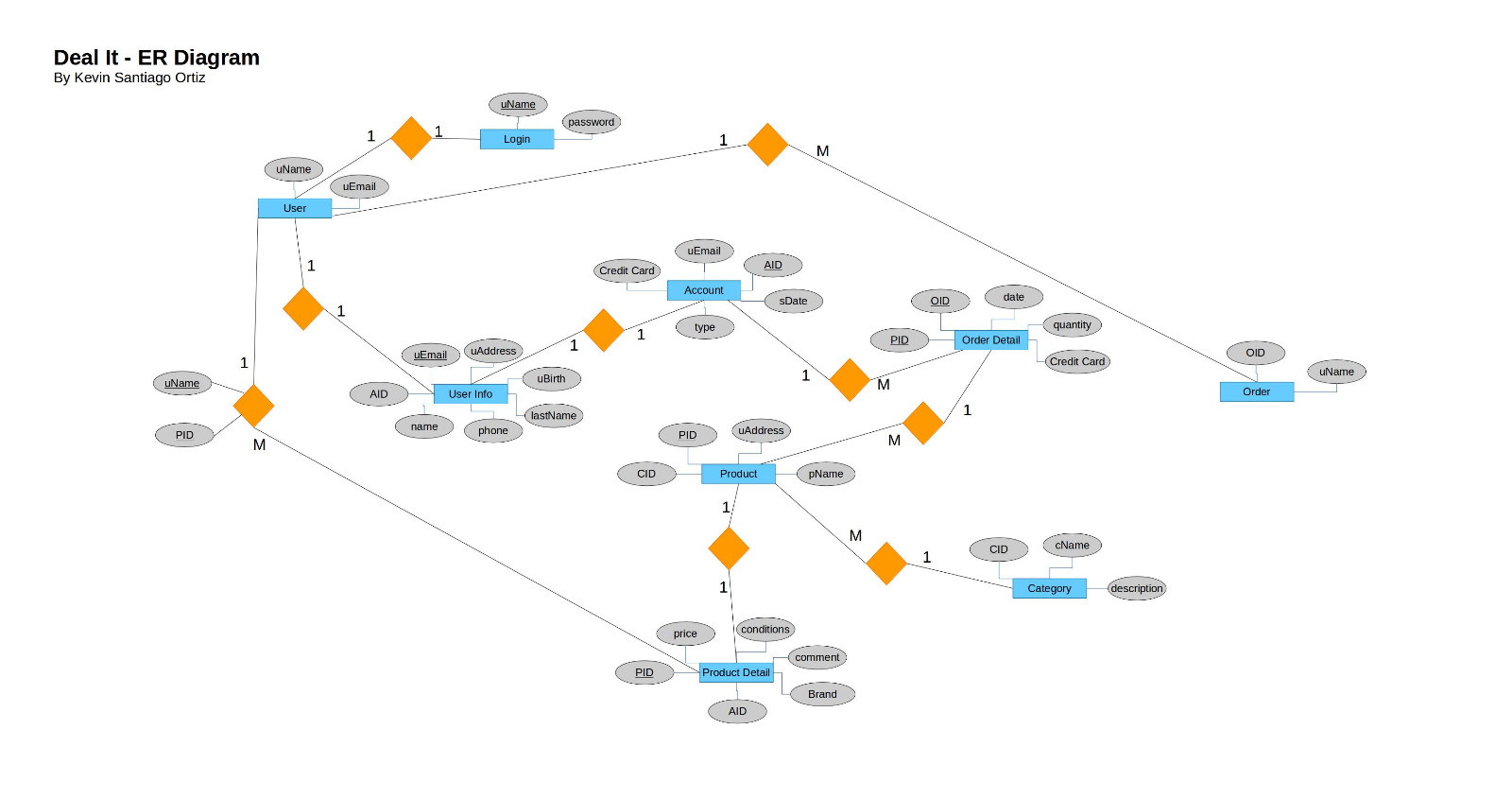
*User Watch List --* Bind the user with the items that he adds to the watch list.

*Login* – Credentials of each registered user.

*Order* – Order number and the name of the user who makes the order.

*Order Detail* - Contains information about the order such as which products are on the order and the quantity, etc.

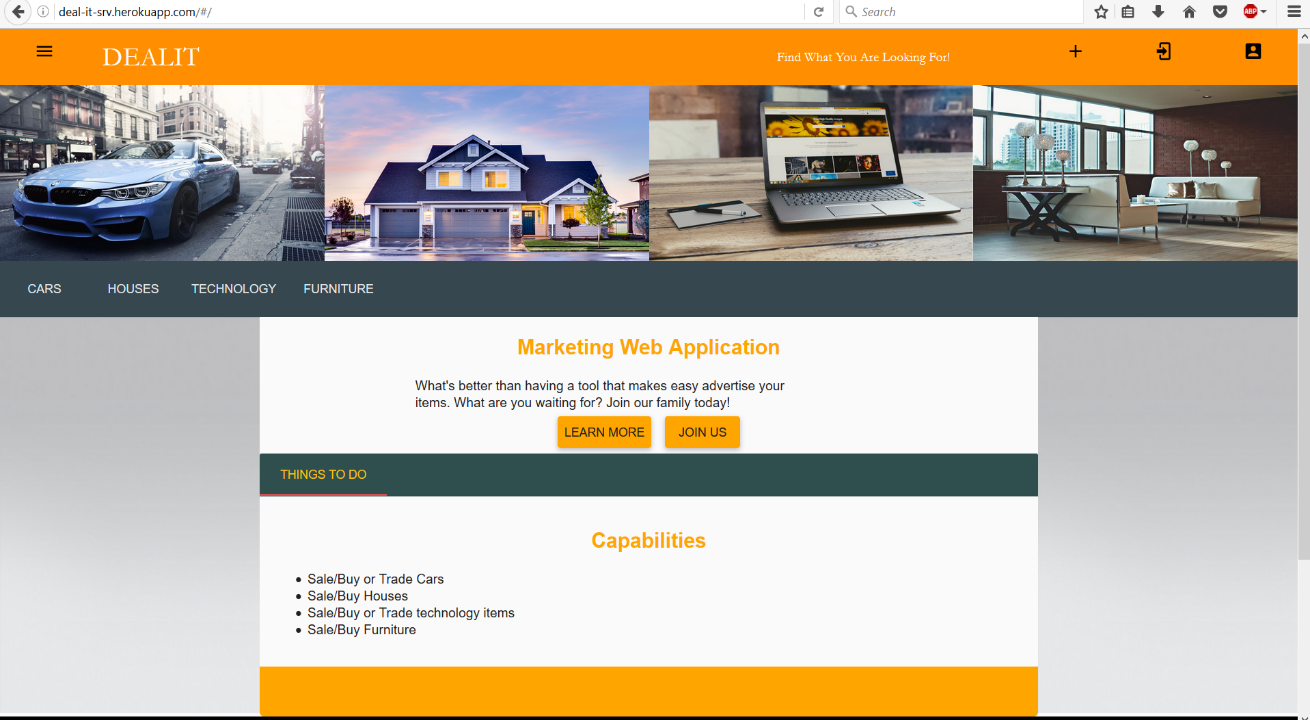
**E-R Diagram**

****

Each user will need credentials for login in. Every user has some information (User Info) that is bind to the account. User have the capability to acquire products or sale them. Each product has product details where is also stored a relation with the user that put it on sale. Each product also has a category under which it’s label. There is an entity order that contains the user and the order to create a relation with the user and the order details. The order detail contains all the information about the order such as how much product is buy, which ones are in the order and credit card used to pay for the order.

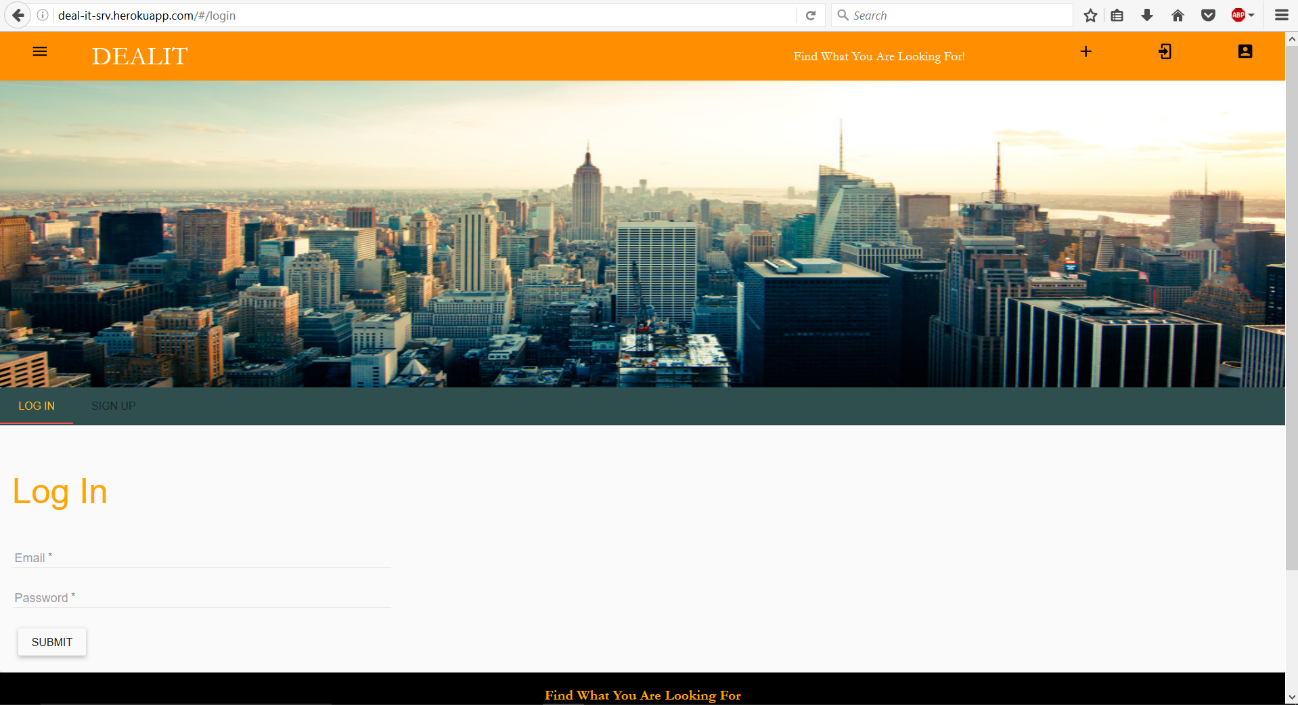
**Client Side -Screenshots**

Home Page Screenshot



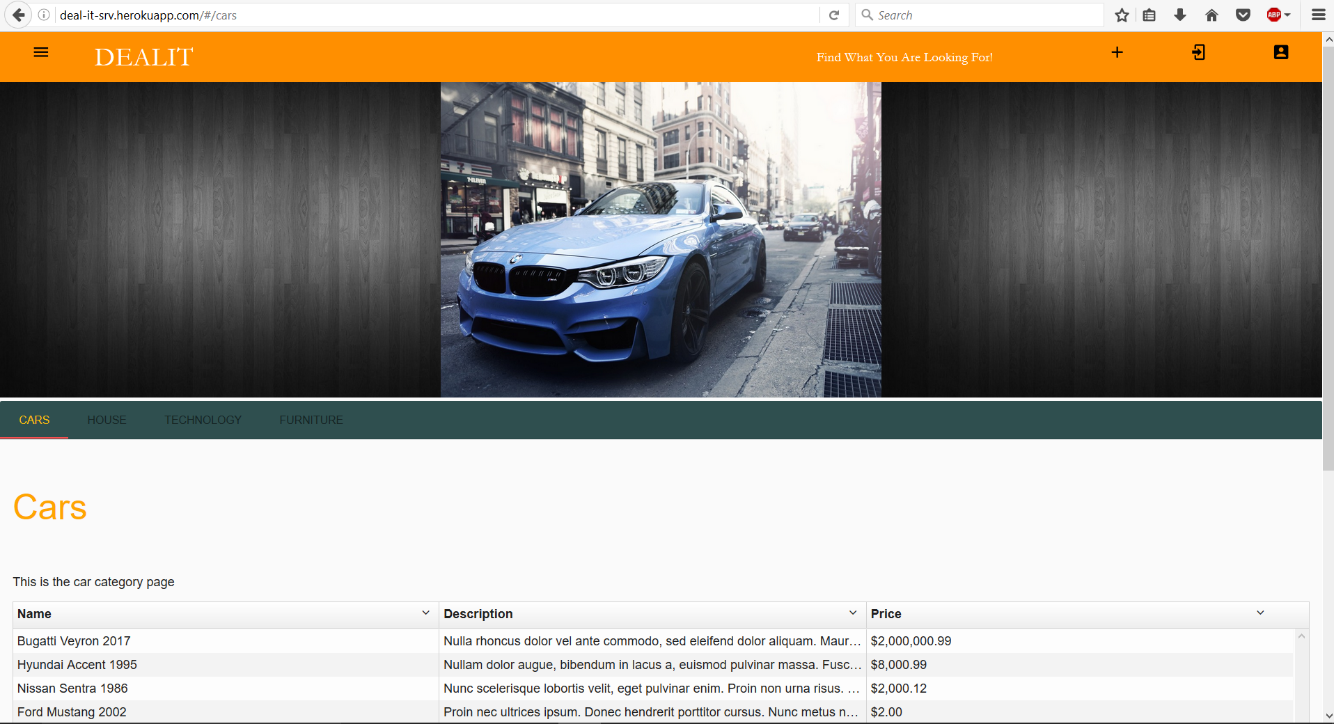
On the home page the user has access to information about the web application. Also this page contains menus and buttons with links to navigate through the app. The menu shown above have all the categories in which the user could search for anything they are looking for. Also there is a side navigation menu that popup on the left of the window with the use of the top left icon.

Login/Signup Screenshot



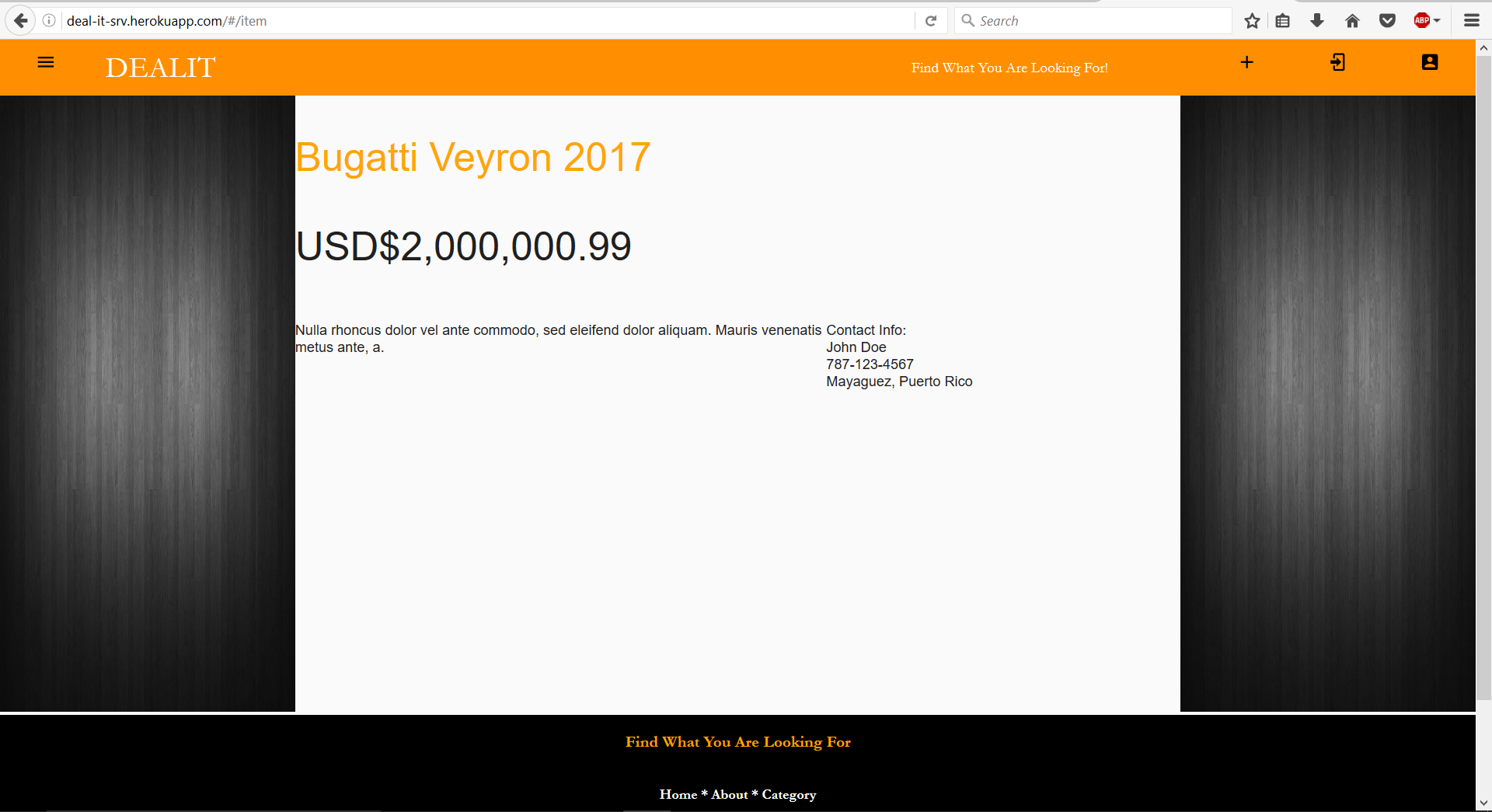
The login/signup view serve as an interface in which the user can login to his account. If the user doesn’t have an account, the form for signup is also available.

Category View Screenshot



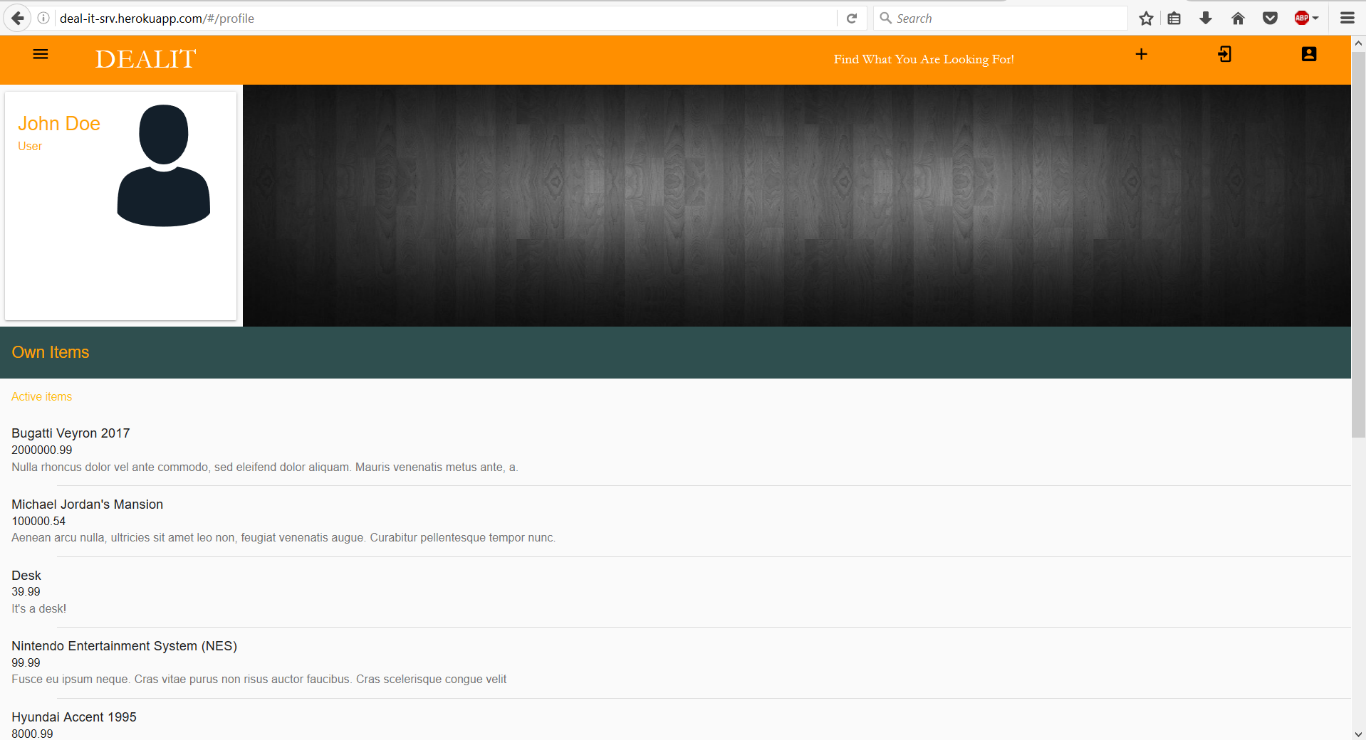
On the category view it’s presented a table with all the items which are labelled under that category, in this case cars. The user can click on any of the items for a more detailed view for that item.

Individual Item View



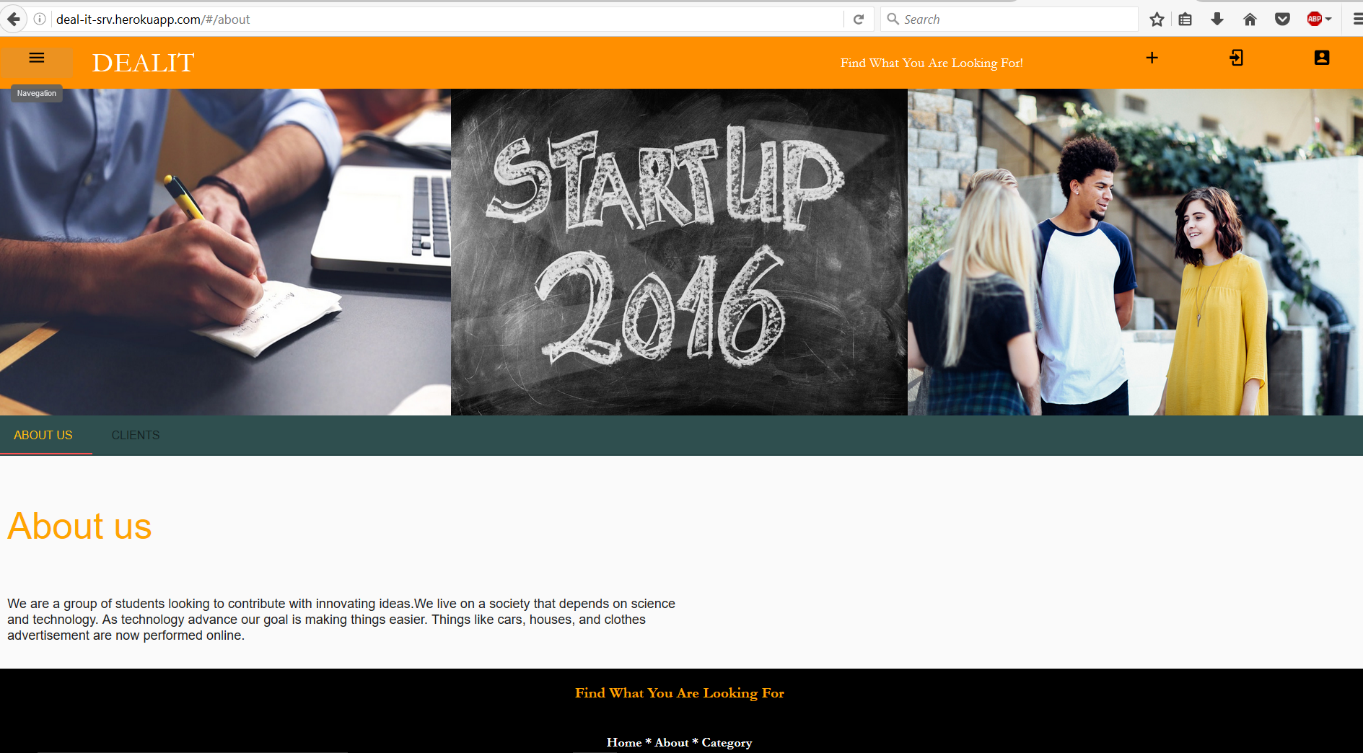
The individual item view its where all the important data about an item can be accessed by the user. Information such as price, description and contact information is something that can be expected.

Profile View



The profile view contains a list of items that are currently post and still active on the application by the user.

About View



The about view contains a little more information about the purpose of the web application and which is the scope of clients that it aims.