Bar management system

Design document

Conor McBride

CA2

# Introduction

The application I have created is a bar management system. The idea behind it is that a person who manages multiple bars will be able to remotely manage them.

**Features**

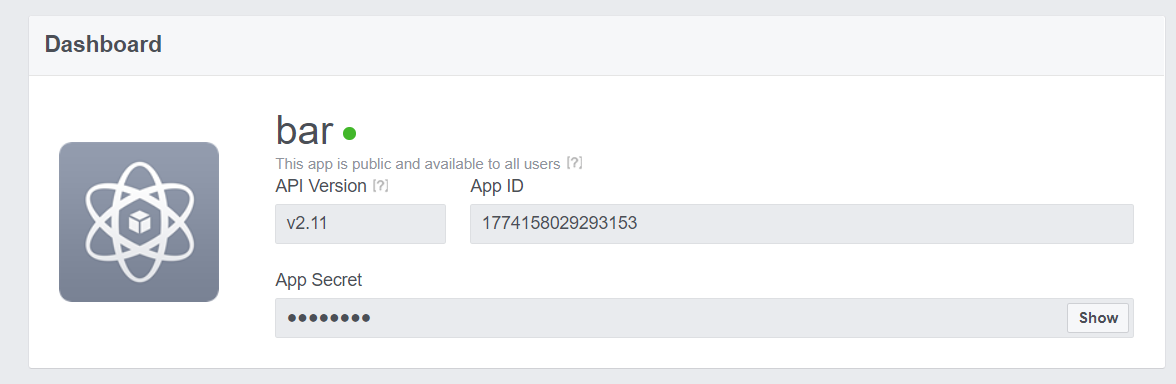
* Register.
* Log in- Using facebook, gmail, twitter.
* List of bars/staff.
* Add bar/staff.
* Delete bar/staff.
* Edit a bars earnings, edit a staff member’s wage.
* Git hub: <https://github.com/conormcbride/barmangV2>
* Mongo persistence.
* Back end testing that is run by putting in the command “npm test”. (to run these tests the server cannot be running.)
* Search via bar/staff name on the list pages.

# API’s

In this project I have three different apis. They are all in my log in screen. They are facebook, twitter and gmail. To access the application through these apis the account created in the register screen must have the same email address as the option chosen for example if you want to log in through facebook you have to have registered with the same email that you use to log in with on facebook. With all of the apis you need a privacy policy so I just found a generator for my purposes [https://privacypolicies.com/privacy/view/oAH3u3](https://l.facebook.com/l.php?u=https%3A%2F%2Fprivacypolicies.com%2Fprivacy%2Fview%2FoAH3u3&h=ATOKq1fuOHfaiBueCV2rTEdpuTok1jraHkHzp2Yau04TVltvZ8NBQXzZ8HJ6EMvfVPNuGBaOowNHMBAD8mhyWEL9xcecBTF11jqW7ZomOUfJxFsOoFzFRf6AnY0dvzxuv4CZrHQc1Pk)

## Facebook log in

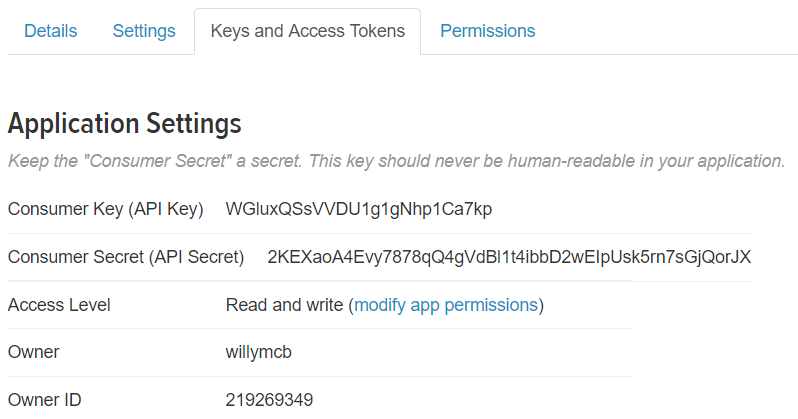
The facebook api is found on the developer.facebook.com site. They give you the code to put into your project and you also need keys so they work.



Here is where the keys are found.

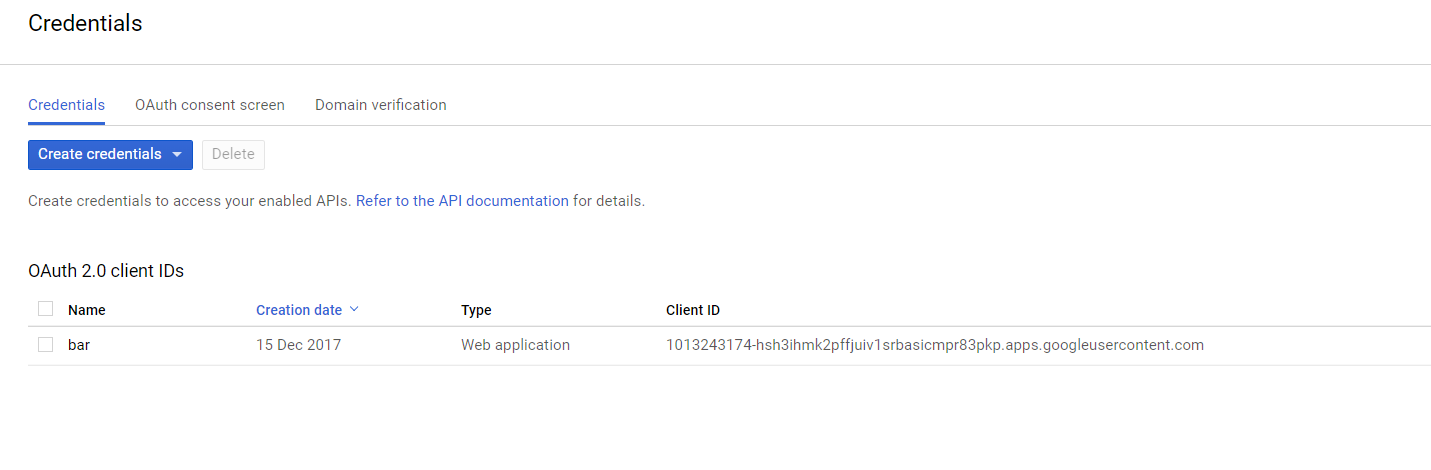
## Twitter log in

The twitter api is found on the apps.twitter.com page. Again they provide you with the code needed. The keys to get the feature to work are found here



## Gmail log in

The gmail api is found here console.developers.google.com. They provide the code here and they also provide the keys here also.



(GitHub, 2017)

The code for all these apis can be found in the passport.js file.







# Add feature

The add works the same on both pages. It is very straight forwards, all the fields are mandatory and the earnings/wage fields both have to be numbers. When the submit button is selected the correct list page is brought up confirming the add worked.

# List pages

The rest of the functionality of my program lies within my list pages and again both work the same. When delete is selected a prompt comes on the screen to confirm if this is desired. The edit works for the earnings and the wage fields, when edit is selected the save bar at the bottom of the list becomes available and the edit is done here. There is a search function at the top of the page that searches by all of the different.

# Profile

The profile shows who is logged in and their email address.

# Automatic tests

When npm test is run the database is cleared and added with hard coded objects so the test can run properly.

# References

GitHub. (2017). gugui3z24/meanstacktutorial. [online] Available at: https://github.com/gugui3z24/meanstacktutorial [Accessed 17 Dec. 2017].