CAB230 WEB COMPUTING REPORT

Client and Server Side

Tuan Thanh Diep
Student ID: n9607234

Table of Contents

1. Client Side

1.1 Registration Page

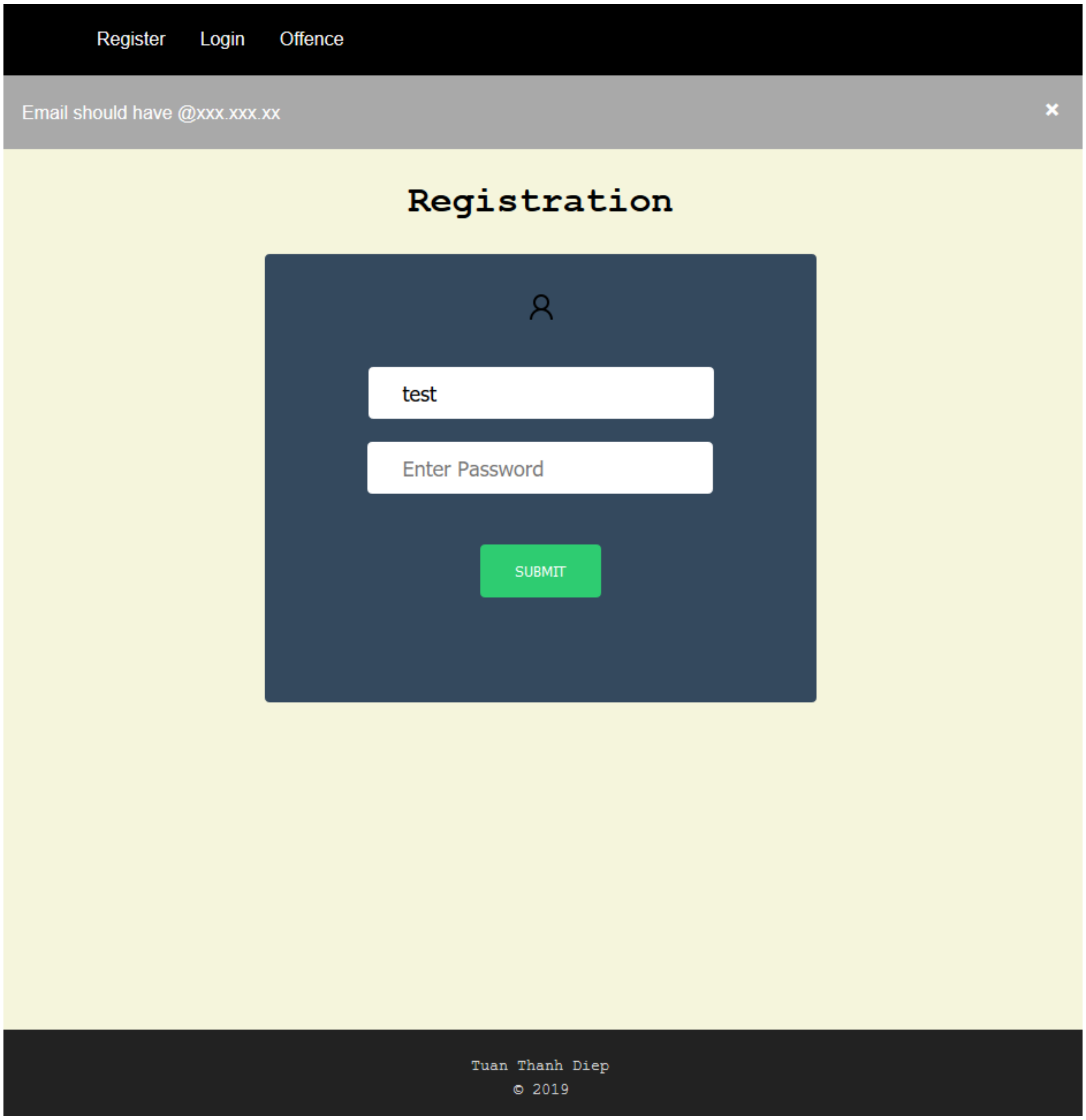
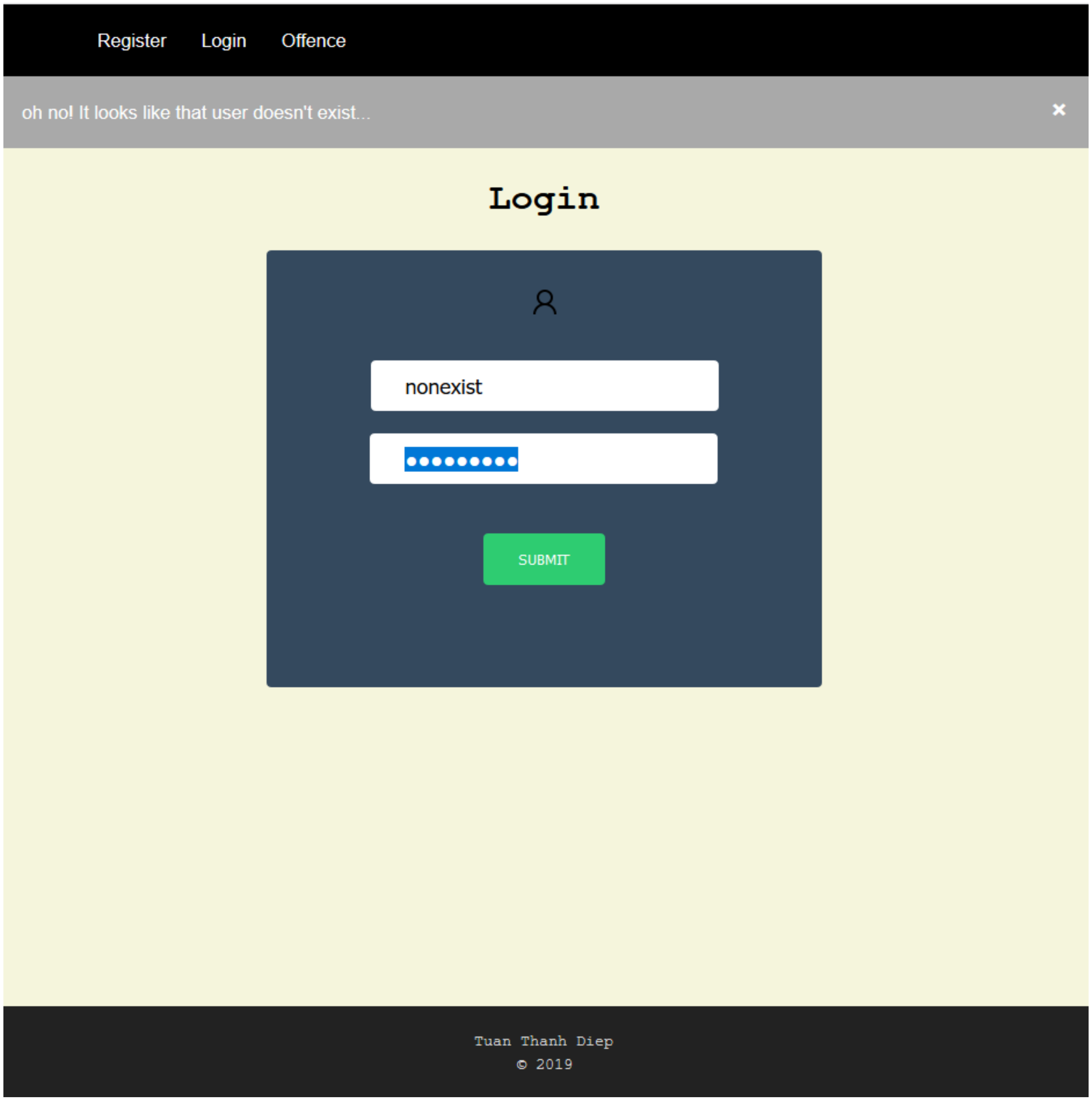


Figure 1 Registration Page

When user register an existing account or with a non-email username, an alert appear as the figure display above. However, with a successful registration, the page will redirect to the login page where your details are automatically filled.

1.2 Login Page



oh no! It looks like that user doesn't exist...

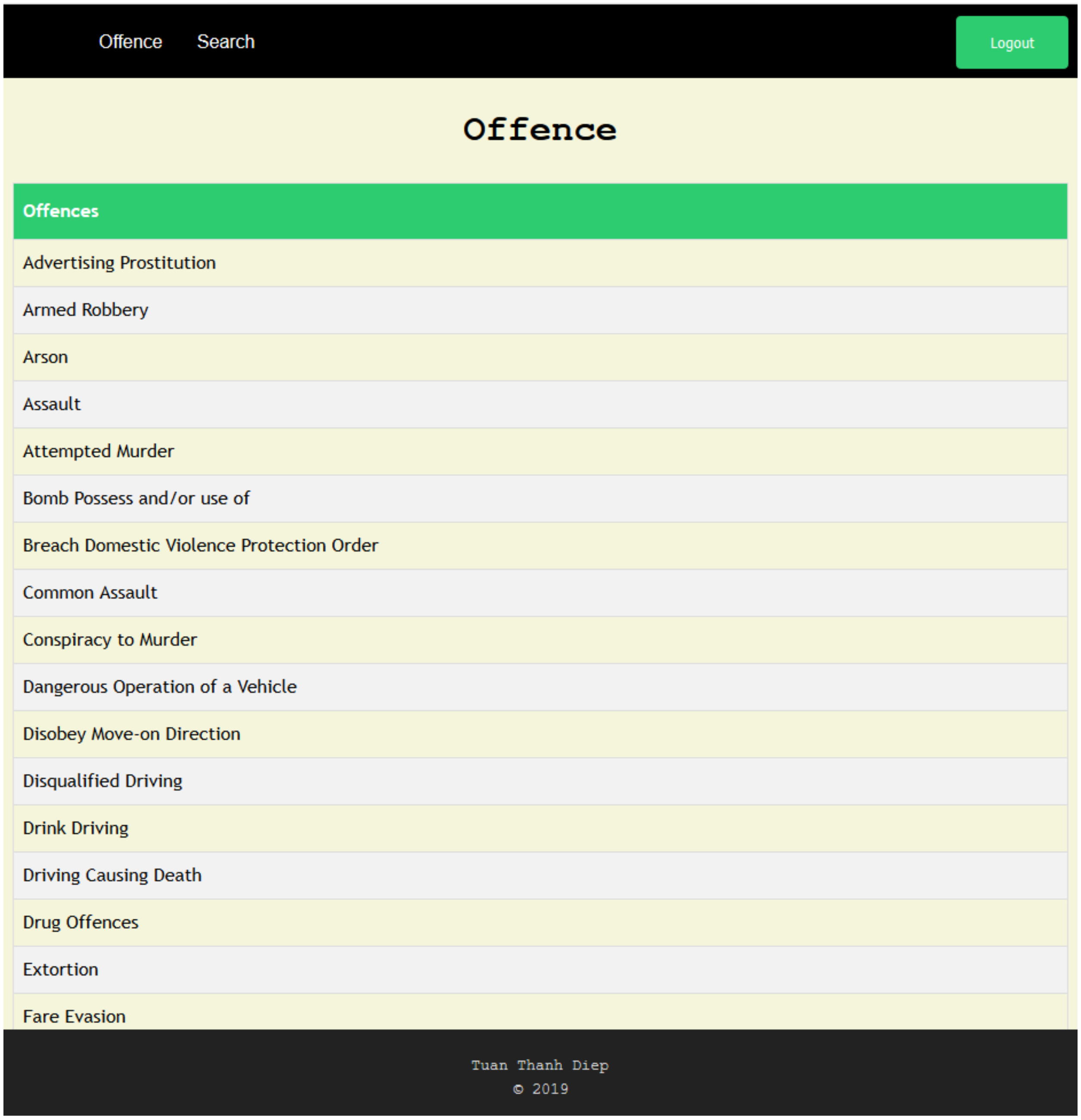
Login

nonexist

Figure 2 Login Page

Similarly, the login site will have the similar alert for invalid authentication such as entering a non-existing account, wrong password or email. When the user has successfully log in, a JWT token is received and the user will be redirected to the Offence page where a list of offences is displayed in a table as presented in **Figure 3**. The user is no longer able to access to both register and login page until they have successfully log out.

1.3 Offence Page



Offence

Offences
Advertising Prostitution
Armed Robbery
Arson
Assault
Attempted Murder
Bomb Possess and/or use of
Breach Domestic Violence Protection Order
Common Assault
Conspiracy to Murder
Dangerous Operation of a Vehicle
Disobey Move-on Direction
Disqualified Driving
Drink Driving
Driving Causing Death
Drug Offences
Extortion
Fare Evasion

Tuan Thanh Diep
© 2019

Figure 3 Offence Page

The Offence page is accessible for unauthenticated and authenticated users.

1.4 Search Page

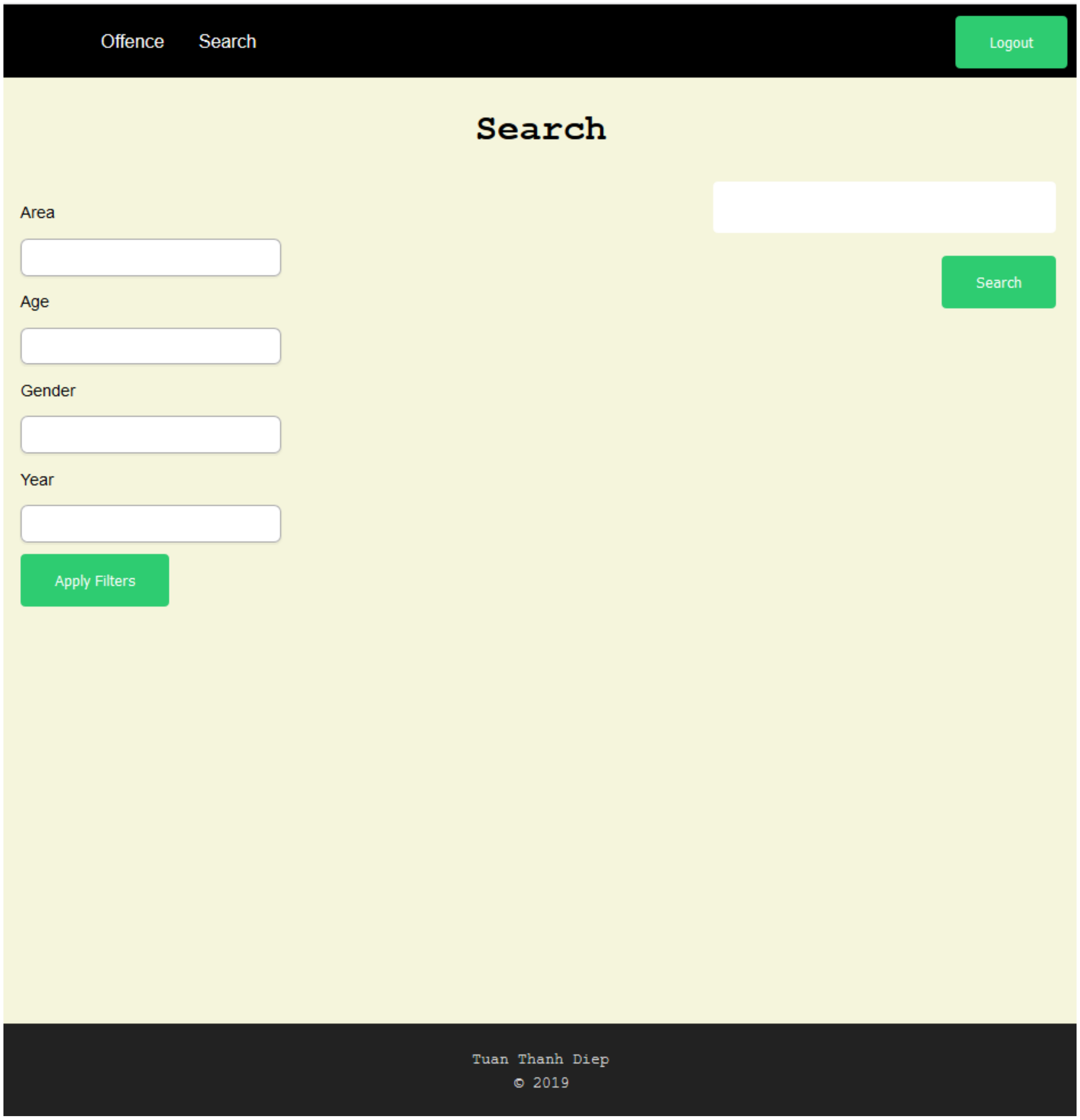
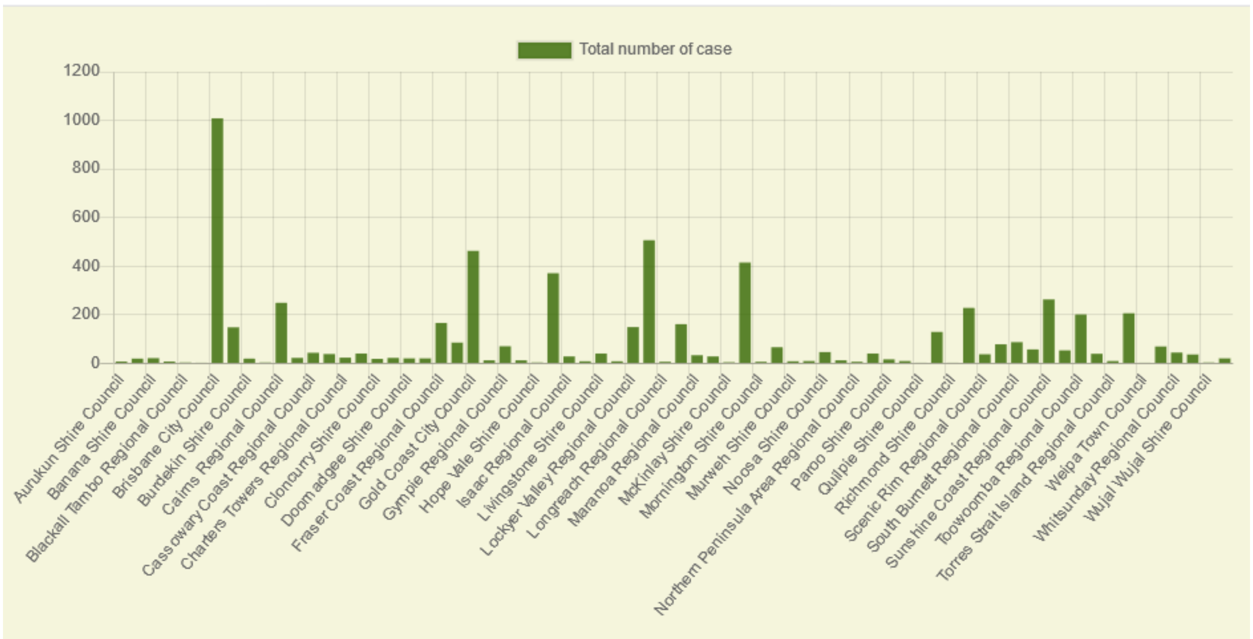
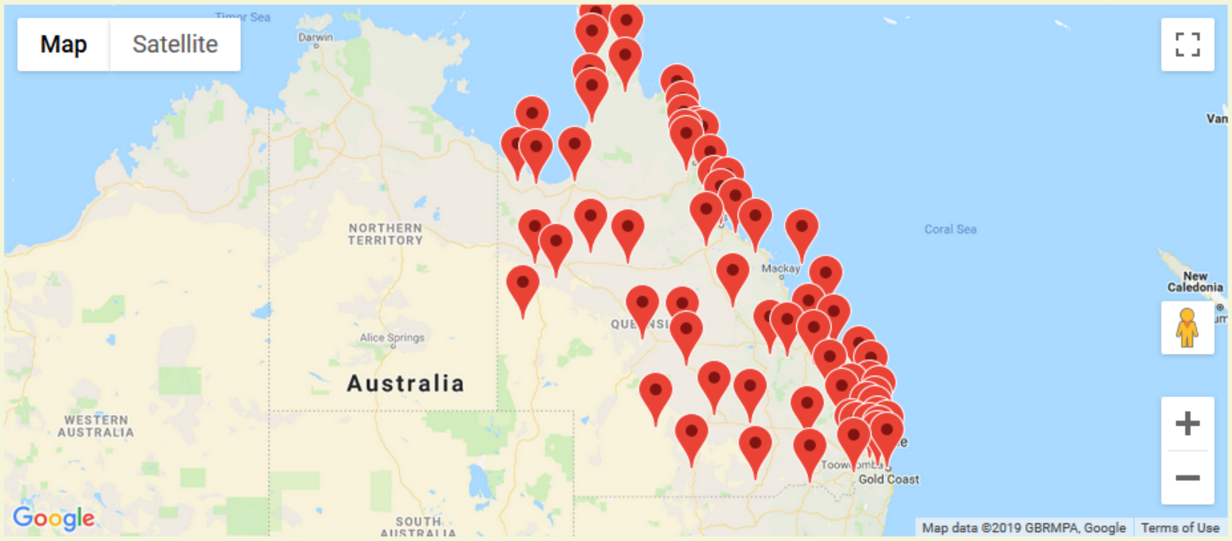


Figure 4 Search Page

The Search page presented in **Figure 4**, requires access token after log in authentication in order to have the accessibility. When an offence is entered in the search bar and submitted, a bar chart displaying the total of cases against the locations of that particular offence. Additionally, a world map will be displayed with markers generated by the latitude and long from the retrieved API data from the search. An example from a search result is in **Figure 5**.



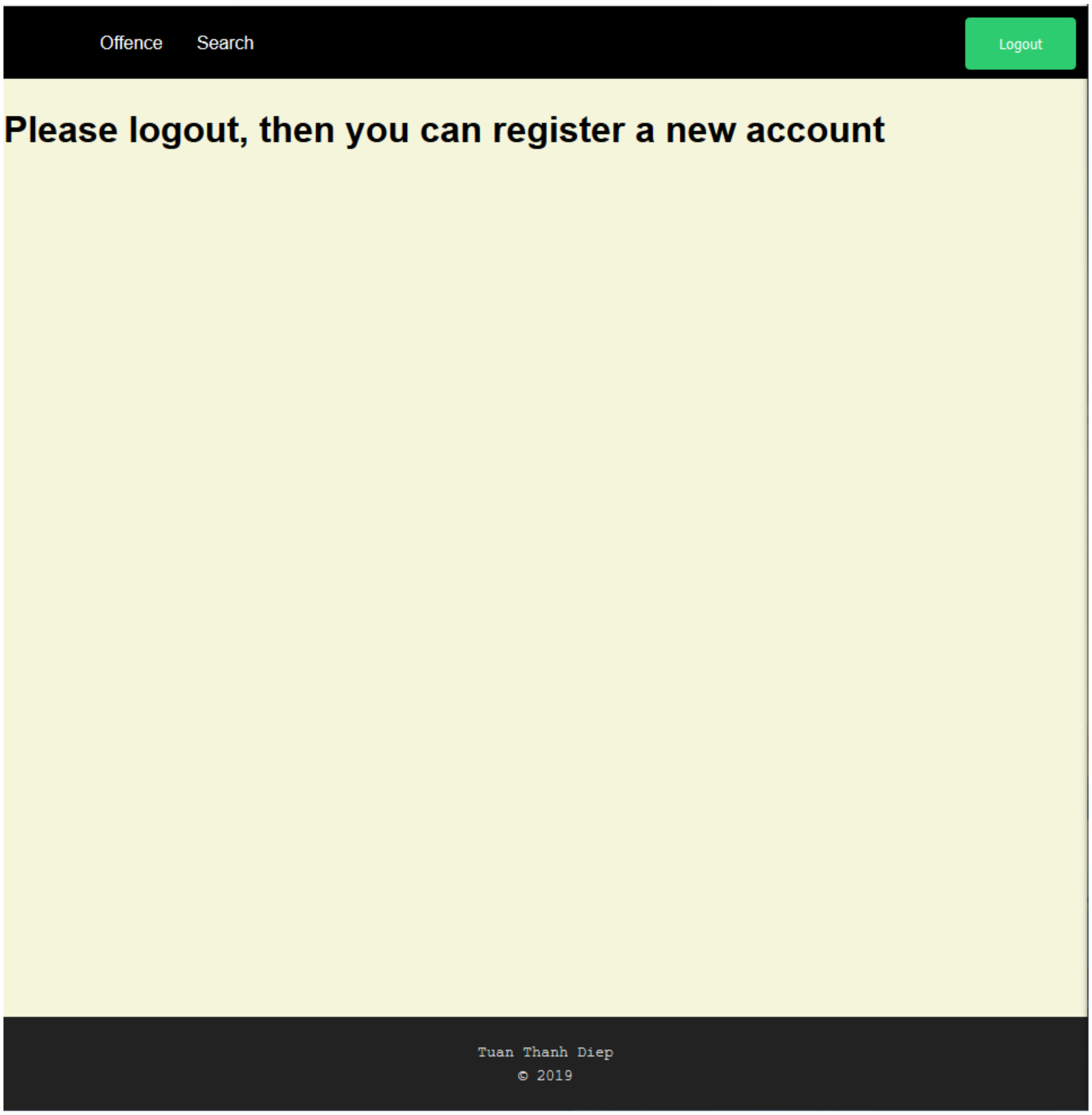


Tuan Thanh Diep
© 2019

Figure 5 An example of a search request

Notice that results with the value of zero in total cases have been filtered out before the chart and map are generated.

1.5 Invalid Access Attempts



Please logout, then you can register a new account

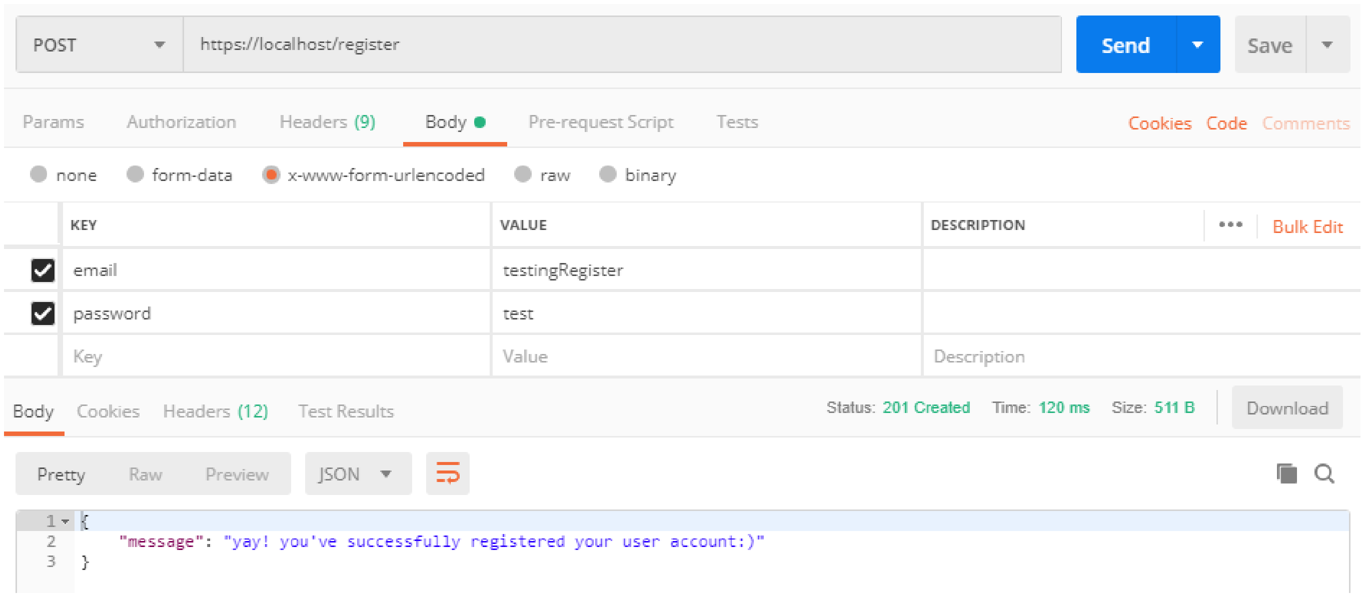
Figure 6 An attempt of accessing Login or Register when logged in

As obvious as it is, user can enter Login and Registration page route into the browser. In **Figure 6,** the user has logged in and attempted to access the register and login while are still logged in. Similarly, when accessing the search page when not logged in, a similar block page appears like above.

2. Server Side

2.1 Routes

2.1.1 /Register



https://localhost/register

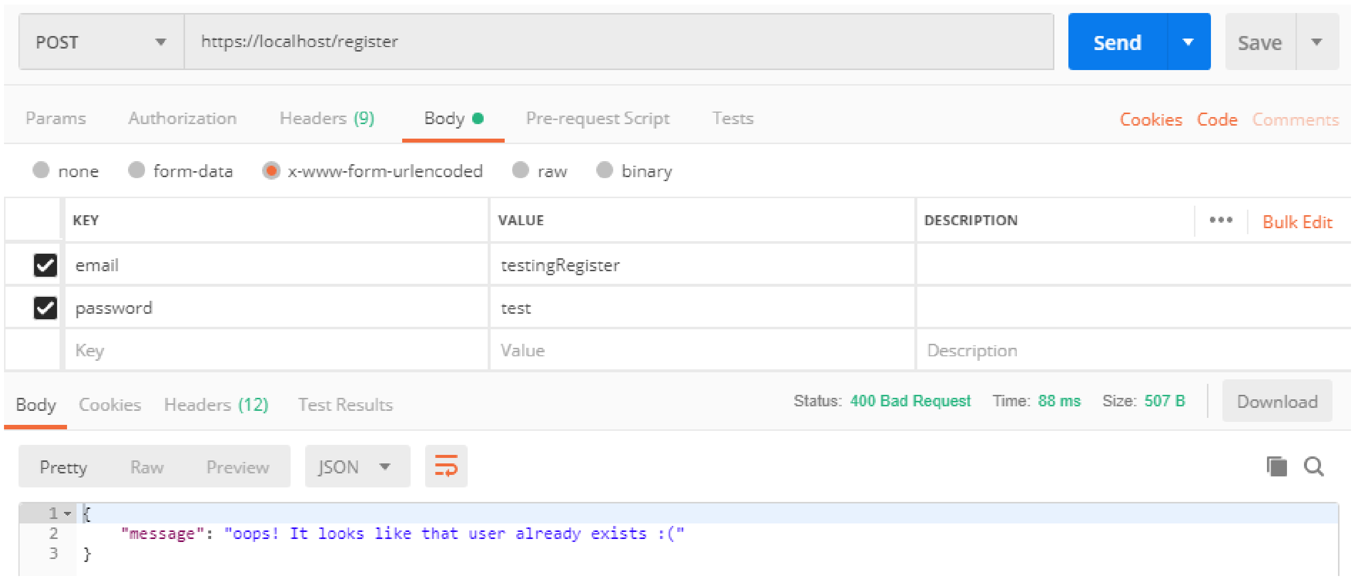
KEY	VALUE	DESCRIPTION
email	testingRegister	
password	test	
Key	Value	Description

Status: 201 Created Time: 120 ms Size: 511 B

```
{
    "message": "yay! you've successfully registered your user account:)"
}
```

Figure 7 A POST request on Register route

In **Figure 7**, a POST request was made to the register route with a new account details embedded the body using Postman application. Hitting the same request with the same account details, the database will respond with an alert message like below



https://localhost/register

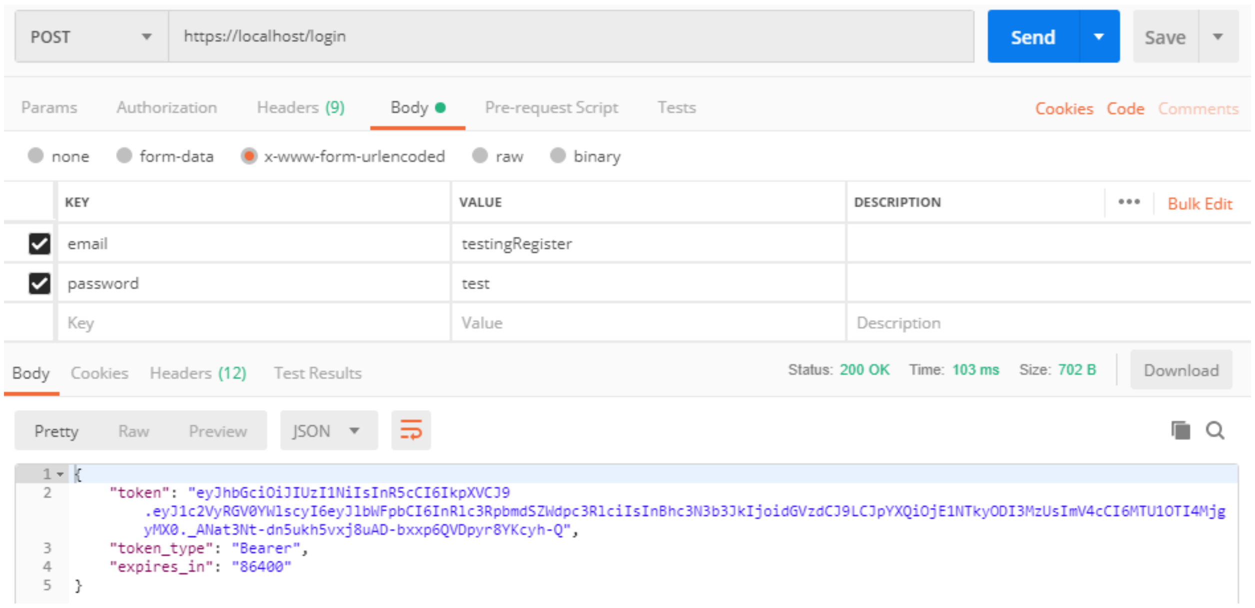
KEY	VALUE	DESCRIPTION
email	testingRegister	
password	test	
Key	Value	Description

Status: 400 Bad Request Time: 88 ms Size: 507 B

```
{
    "message": "oops! It looks like that user already exists :("
}
```

Figure 8 Database response for creating an existing account

2.1.2 /Login



POST https://localhost/login

Body: x-www-form-urlencoded

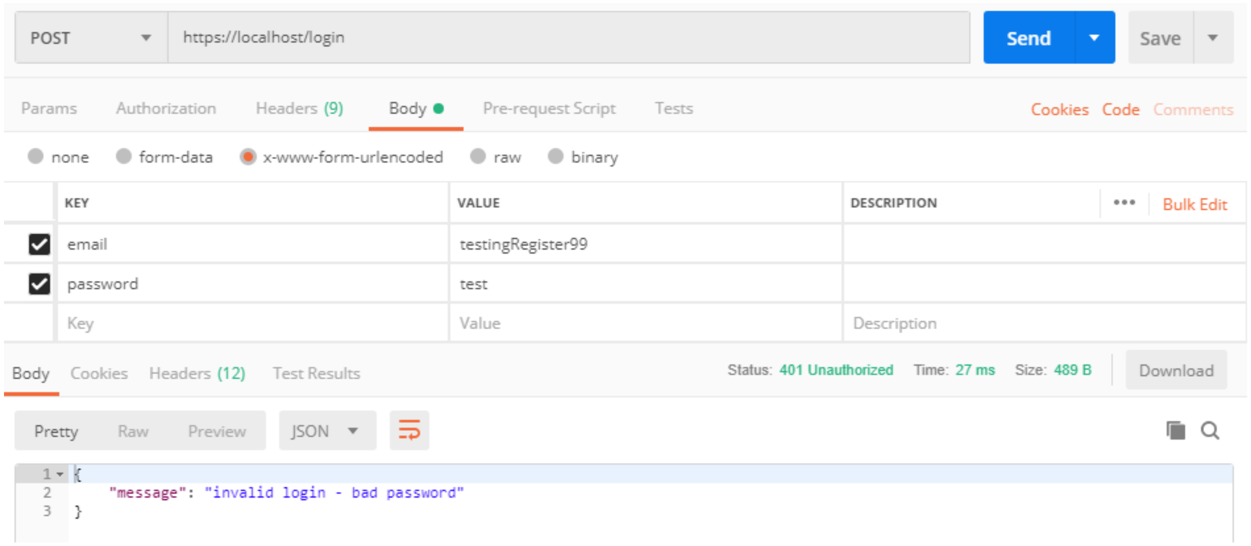
	KEY	VALUE	DESCRIPTION
✓	email	testingRegister	
✓	password	test	

Status: 200 OK Time: 103 ms Size: 702 B

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
        .eyJ1c2VyRGV0YWlscyI6eyJlbWFpbCI6InRlc3RpbmdSZWdpc3RlciIsInBhc3N3b3JkIjoidGVzdCJ9LCJpYXQiOjE1NTkyODI3MzUsImV4cCI6MTU1OTI4Mjg
        yMX0._ANat3Nt-dn5ukh5vxj8uAD-bxxp6QVDpyr8YKcyh-Q",
    "token_type": "Bearer",
    "expires_in": "86400"
}
```

Figure 9 Login with the new account

"token", "token_access" and "expires_in" are return on a 200 response and the token has a time limit for 24 hours.



POST https://localhost/login

Body: x-www-form-urlencoded

	KEY	VALUE	DESCRIPTION
✓	email	testingRegister99	
✓	password	test	

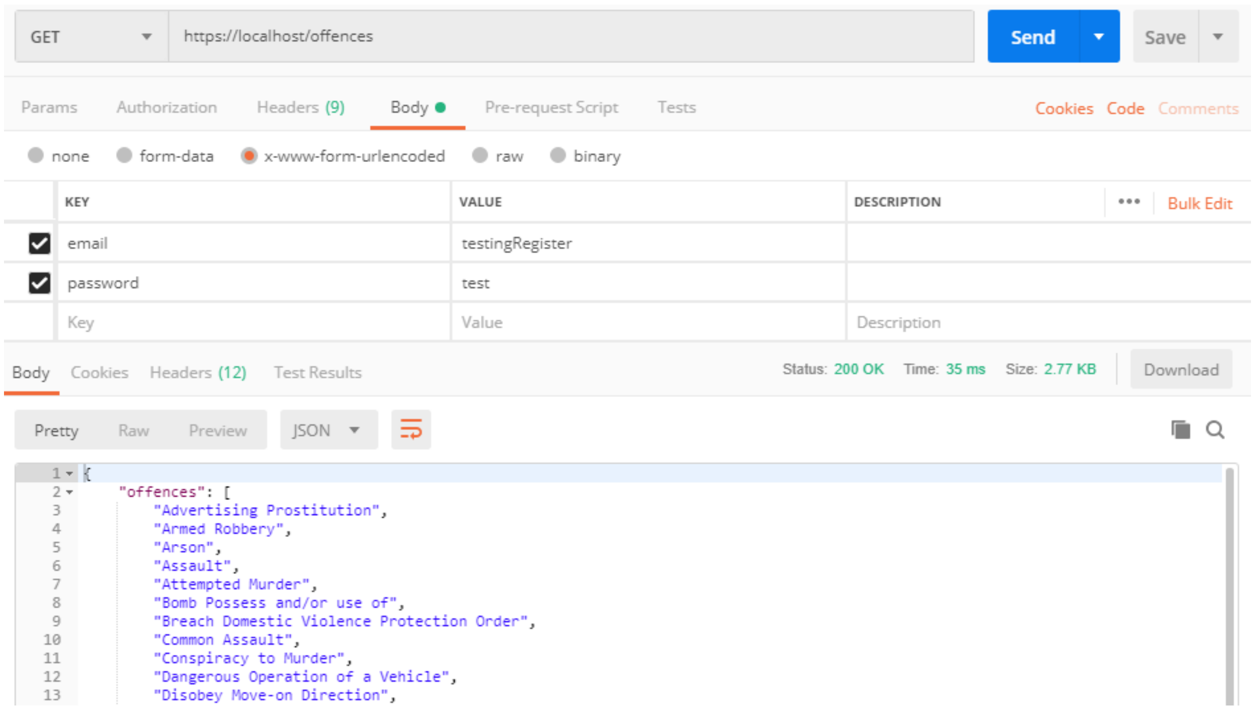
Status: 401 Unauthorized Time: 27 ms Size: 489 B

```
{
    "message": "invalid login - bad password"
}
```

Figure 10 Status 401 Unauthorized

Logging in with incorrect account details, user will be returned with a JSON message and a 401-error response.

2.1.3 /Offences



https://localhost/offences

KEY	VALUE	DESCRIPTION
email	testingRegister	
password	test	
Key	Value	Description

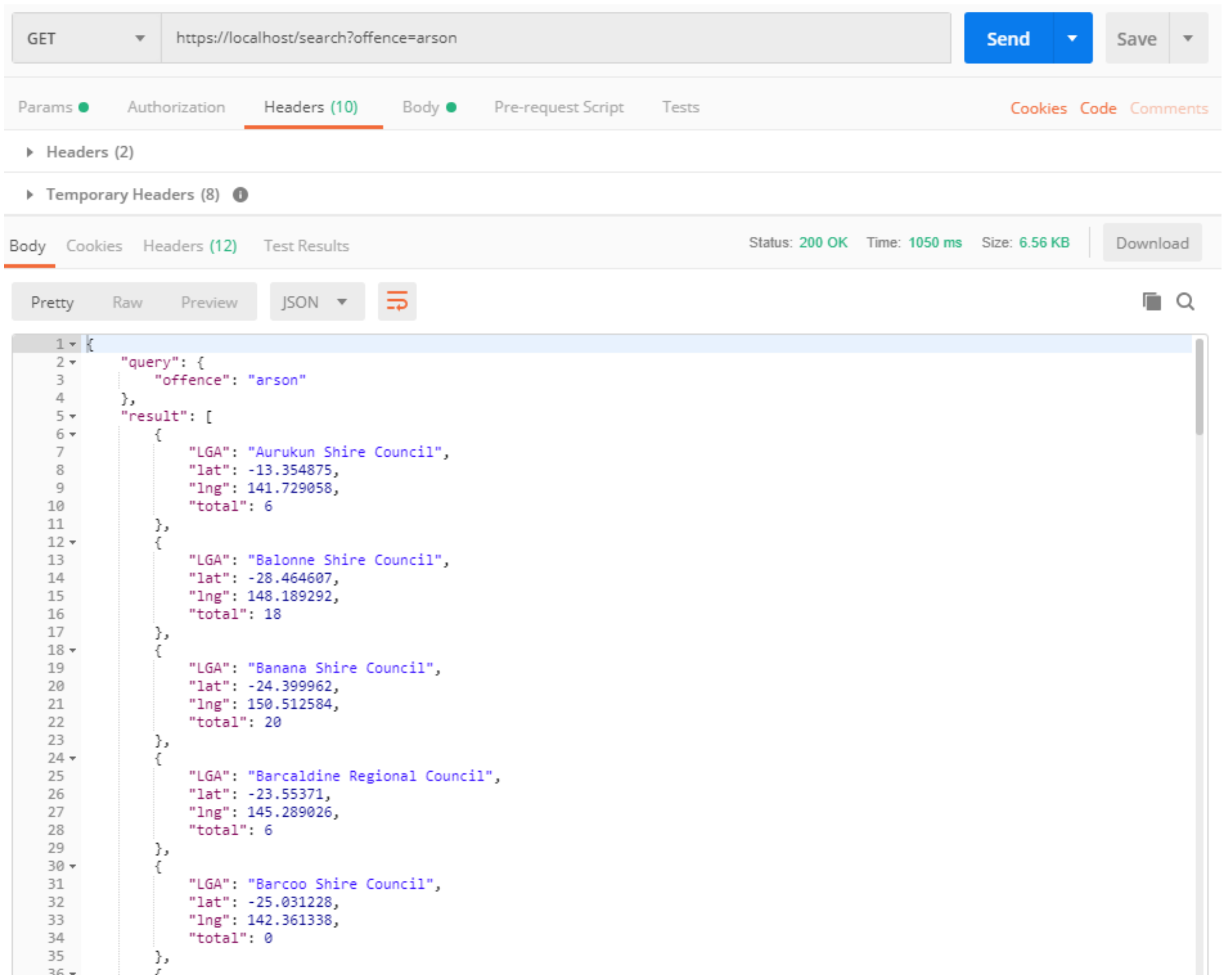
Status: 200 OK Time: 35 ms Size: 2.77 KB

```
{
    "offences": [
        "Advertising Prostitution",
        "Armed Robbery",
        "Arson",
        "Assault",
        "Attempted Murder",
        "Bomb Possess and/or use of",
        "Breach Domestic Violence Protection Order",
        "Common Assault",
        "Conspiracy to Murder",
        "Dangerous Operation of a Vehicle",
        "Disobey Move-on Direction",
```

Figure 11 GET request at Offences route

By hitting the offences route, the response is a JSON with an array of all the offences in the database.

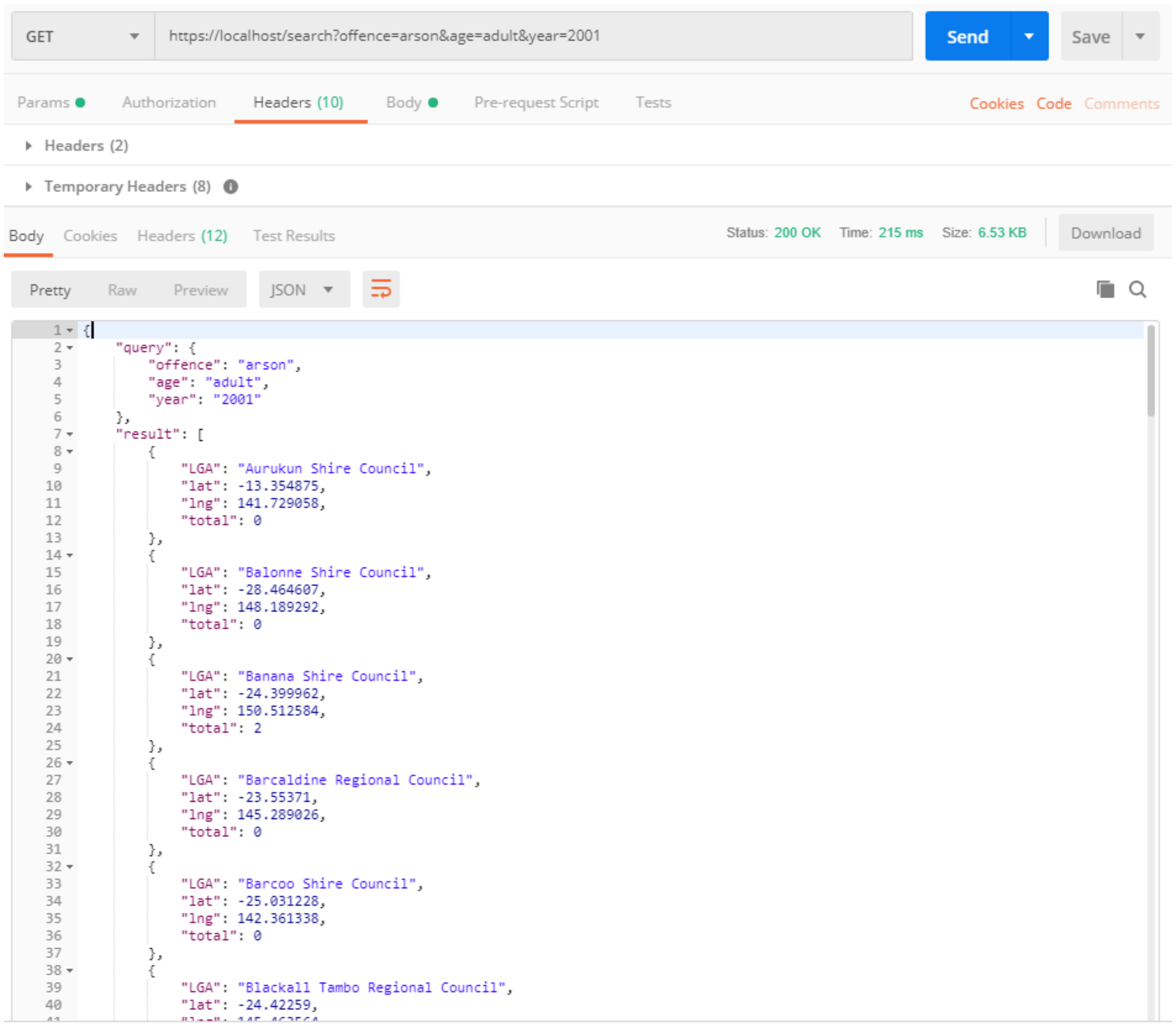
2.1.4 /Search



```
{
    "query": {
        "offence": "arson"
    },
    "result": [
        {
            "LGA": "Aurukun Shire Council",
            "lat": -13.354875,
            "lng": 141.729058,
            "total": 6
        },
        {
            "LGA": "Balonne Shire Council",
            "lat": -28.464607,
            "lng": 148.189292,
            "total": 18
        },
        {
            "LGA": "Banana Shire Council",
            "lat": -24.399962,
            "lng": 150.512584,
            "total": 20
        },
        {
            "LGA": "Barcaldine Regional Council",
            "lat": -23.55371,
            "lng": 145.289026,
            "total": 6
        },
        {
            "LGA": "Barcoo Shire Council",
            "lat": -25.031228,
            "lng": 142.361338,
            "total": 0
        },
```

Figure 12 A result of the protected route Search without filters

By using knex and some middleware, the search route is protected route which only allow authorized users to have access. Moreover, adding more parameters such as “&area=”, “&age=”, “&gender=” and “&year=” will filter out the results from **Figure 12.**



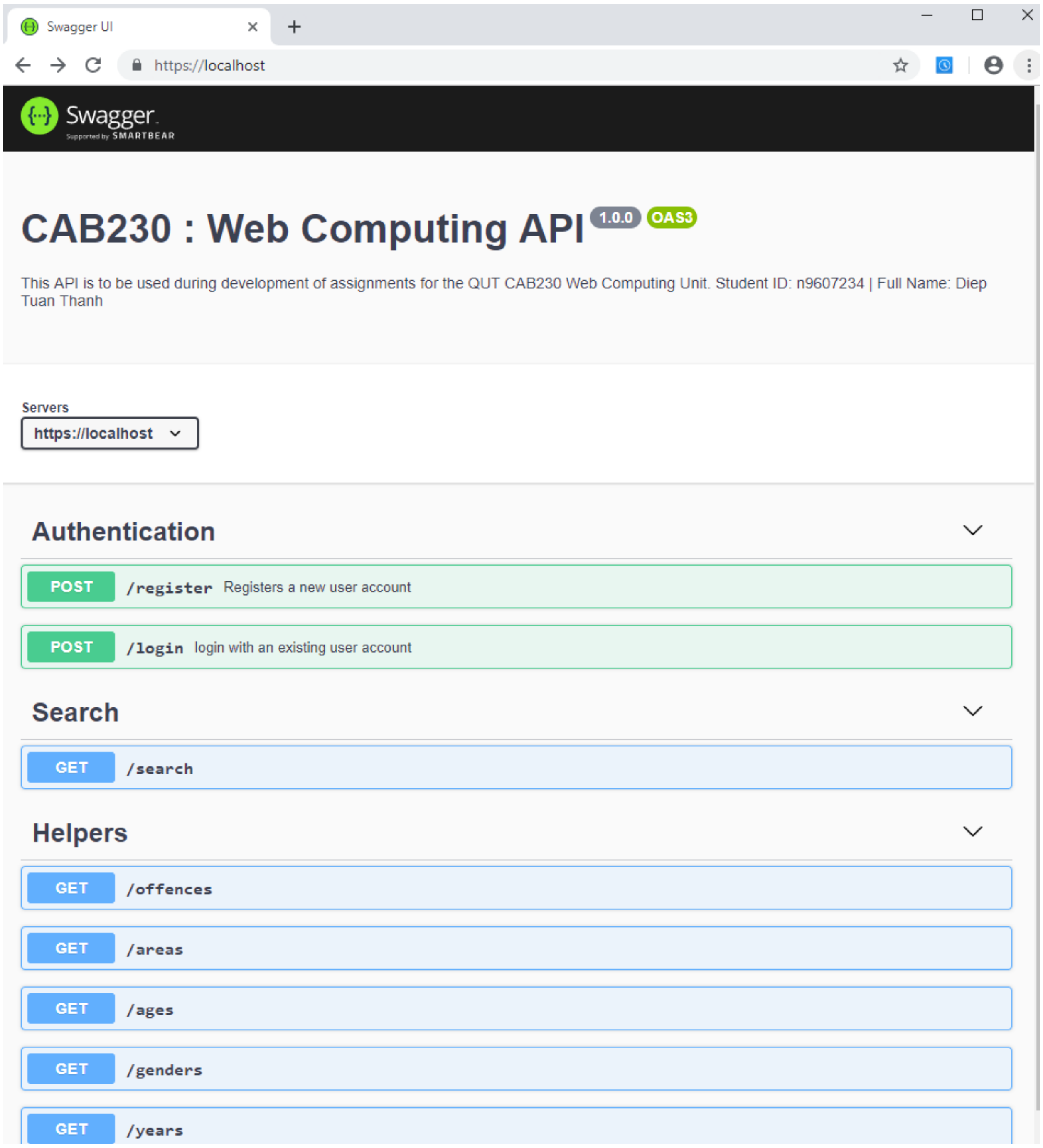
https://localhost/search?offence=arson&age=adult&year=2001

Status: 200 OK Time: 215 ms Size: 6.53 KB

```
{
    "query": {
        "offence": "arson",
        "age": "adult",
        "year": "2001"
    },
    "result": [
        {
            "LGA": "Aurukun Shire Council",
            "lat": -13.354875,
            "lng": 141.729058,
            "total": 0
        },
        {
            "LGA": "Balonne Shire Council",
            "lat": -28.464607,
            "lng": 148.189292,
            "total": 0
        },
        {
            "LGA": "Banana Shire Council",
            "lat": -24.399962,
            "lng": 150.512584,
            "total": 2
        },
        {
            "LGA": "Barcaldine Regional Council",
            "lat": -23.55371,
            "lng": 145.289026,
            "total": 0
        },
        {
            "LGA": "Barcoo Shire Council",
            "lat": -25.031228,
            "lng": 142.361338,
            "total": 0
        },
        {
            "LGA": "Blackall Tambo Regional Council",
            "lat": -24.42259,
```

Figure 13 Search request with filters

2.2 Swagger Docs


Swagger UI

https://localhost

CAB230 : Web Computing API 1.0.0 OAS3

This API is to be used during development of assignments for the QUT CAB230 Web Computing Unit. Student ID: n9607234 | Full Name: Diep Tuan Thanh

Servers

https://localhost

Authentication

POST /register Registers a new user account

POST /login login with an existing user account

Search

GET /search

Helpers

GET /offences

GET /areas

GET /ages

GET /genders

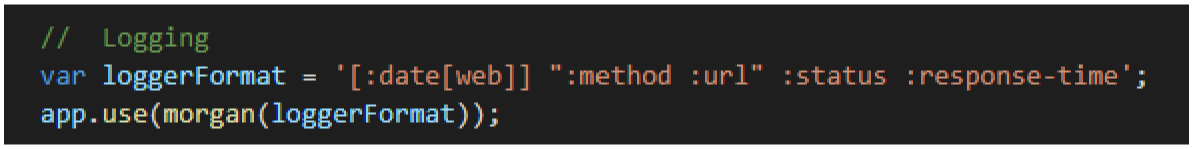
GET /years

Figure 14 Swagger docs for the serverside

The Swagger document for this server side is implemented and heavily based on the Swagger document on https://cab230.hackhouse.sh/. All the results from both POST and GET request are identical to the example above.

2.3 Logging

For server logging, the format is as follow:



```
//  Logging
var loggerFormat = '[:date[web]] ":method :url" :status :response-time';
app.use(morgan(loggerFormat));
```

Figure 15 Logging Format for the server