CAB230 SERVER SPECIFICATION

Liam Wrigley – n9448381

1. Introduction

Implemented:

- Proper routes added for all endpoints.
- Middleware for managing connections, security and database
- Appropriate error handling matching the swagger documentation
- Use of helmet security
- Use of Knex
- Deployed on HTTPS
- Serves accurate swagger docs
- Storing passwords hashed
- Sending jwt token on successful login
- Search by all criteria at once on search endpoint

Not Implemented

- Comma separated search functionality
- Authentication on search endpoint

2. Technical description

Middleware that was used included:

- Knex for connection to the database
- Helmet for security
- Swagger for serving endpoint documentation
- Cors for security
- JWT for authentication
- Bcrypt for hashing passwords before storing them

All error handling matches the swagger documentation as closely as possible, some database queries requiring data processing to display in the correct form, such as the search endpoint:

```
var queryString = {
  "offence" : decodeURI(req.query.offence).split(" ").join("").toLowerCase(),
  "area" : decodeURI(req.query.area),
  "age" : decodeURI(req.query.age),
  "gender" : decodeURI(req.query.gender),
  "year" : decodeURI(req.query.year),
  "month" : decodeURI(req.query.month)
}
//remove blanks
Object.keys(queryString).forEach((key) => (queryString[key] == "") && delete queryString[key])

req.db.distinct().from('offences')
  .select(`offences.area as LGA`)
  .sum(`${queryString.offence} as total`)
  .select('areas.lat', 'areas.lng').join('areas', 'areas.area', '=', 'offences.area').groupBy('offences.area')
  .modify(function(queryBuilder) {
    for (var property in queryString) {
      if (queryString.hasOwnProperty(property)) {
        if(property != 'offence') {
          queryBuilder.where(`offences.${property}`, queryString[property])
        }
      }
    }
  })
.then((rows) => {
  res.json({"query":queryString,"result":rows});
})
```

Figure 1 /Search endpoint

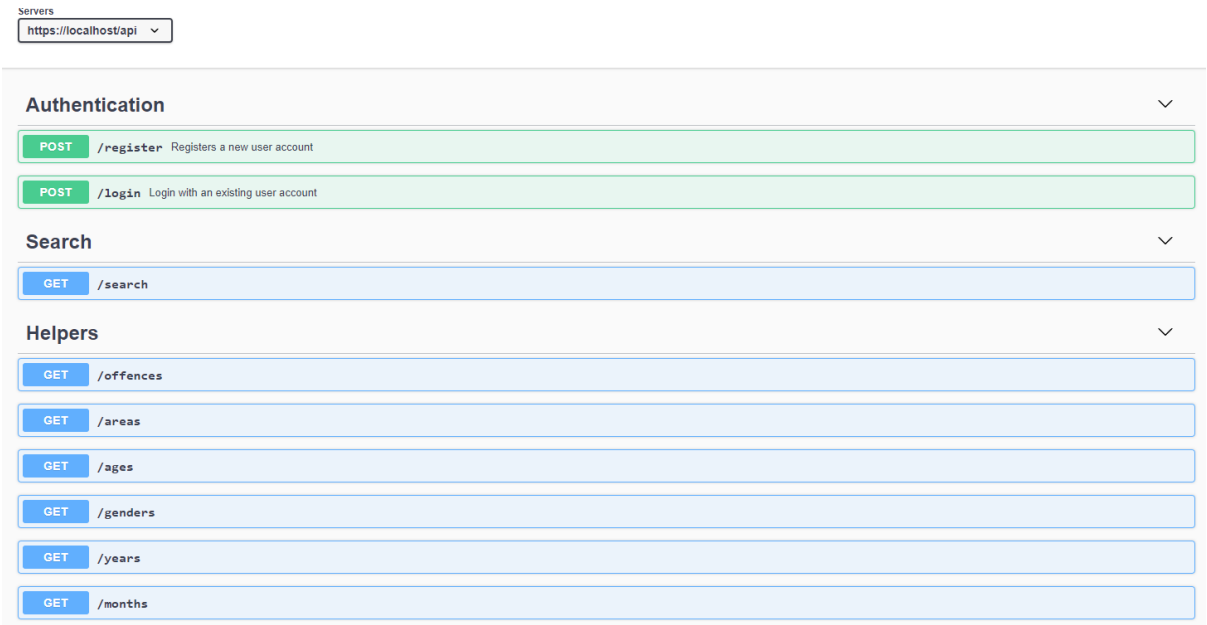
The search endpoint was having issues with receiving the authentication header and was in the end not implemented. Comma separated searches were also not implemented due to time constraints. The query is dynamically built depending on what fields were within the query. A loop goes over all values and adds additional filtering for the select query to the database.

3. Security

The whole server is serving to HTTPS with use of several packages to help with security. Security between the server and the database include bcrypt which ensures that passwords are not stored in the database in plaintext and are hashed 10 times in the middleware. When checking for account details, the server will get the stored hashed password and compare it with an inputted password that is hashed using the same package, this way no passwords are ever stored on the database. Knex is also used to mitigate the use of unwanted characters being entered for search parameters which helps with SQL injections. Jsonwebtoken is used the verify that a user cannot access the search endpoint without being logged in, unfortunately this was not fully implemented. However, upon successful login the user will be served a token, access token and expiry time for the token. Helmet is used to help maintain the headers of requests to ensure that they are only able to accept wanted header parameters.

4. Testing and limitations – test plan, results, compromises.

All testing was conducted on POSTMAN, hitting all endpoints with different data and comparing them to the hackhouse swagger documentation as well as the hackhouse API returns.



Each function has all schemas, examples, tests and error codes working as intended and serving the same information as the hackhouse swagger with the exception of the search endpoint

5. References

Packages used:

- Jsonwebtoken – for authentication
- Bcrypt - for encrypting passwords
- swagger-ui-express – for serving the swagger API documentation
- helmet - for security with headers
- CORS - for security

6. Appendix: brief user guide

To install and run

“npm install”

“npm start”