Junior Web Developer Technical Assessment

Jad Saad
Web Developer
ja_saa@encs.concordia.ca

I. Brief

Tasks One and Two were combined in one web application that is published on github at https://github.com/jadmsaad/Assessment and deployed to heroku at https://assessmentkfl.herokuapp.com/.

The web application is composed of a front-end (assessment 2) that is done with React and a back-end server that has a user management API (assessment 1).

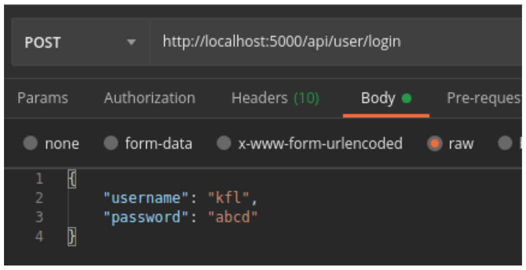
A. testing

To test the application you can either visit it at the heroku link or clone on your own machine from the github repository. To run it on the machine first run ”npm install” on both the /Assessment and /Assesent/client folders so both the dependecies for the server and client gets installed. Then at the server side (/Assessment) you can run the ”npm run dev” script which will concurrently fire up the back-end server and the react development server. The back-end is on port 5000 and the front-end is on port 3000.

To test assessment 1 you can use the web applications login, register and edit user forms which are accessible form the Navbar. The login and register links are found before the user is authorized and the edit user and logout are found in the dropdown titled ”user” after signing in and being authorized.

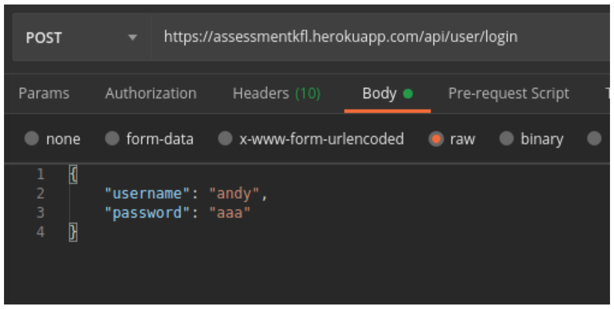
Assessment 1 can also be tested by making the following http requests to the below endpoints:

1) **Create a user:** Post request with raw json body that contains ”first_name”, ”last_name”,”username”, ”email”, ”password”. Include in headers content/type: application/json
heroku: https://assessmentkfl.herokuapp.com/api/auth/create
localhost: http://localhost:5000/api/auth/create
2) **update user info:** Post request with raw json body that contains ”first_name”, ”last_name”,”username”, ”email”, ”password”. Must happen after login and receiving authorization token. The back-end will infer the user from the token. Include in headers content/type: application/json and authorization: the bearer token received after login.
heroku: https://assessmentkfl.herokuapp.com/api/auth/update
localhost: http://localhost:5000/api/auth/update
3) **login:** Post request with raw json body that contains ”username”, ”password”. Include in headers content/type: application/json
heroku: https://assessmentkfl.herokuapp.com/api/auth/login
localhost: http://localhost:5000/api/auth/login



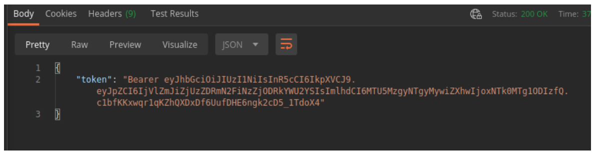
```
POST http://localhost:5000/api/user/login
{
    "username": "kfl",
    "password": "abcd"
}
```

Fig. 1. Login



```
POST https://assessmentkfl.herokuapp.com/api/user/login
{
    "username": "andy",
    "password": "aaa"
}
```

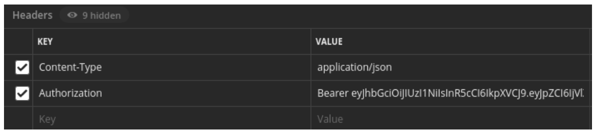
Fig. 2. Login heroku



Status: 200 OK Time: 37

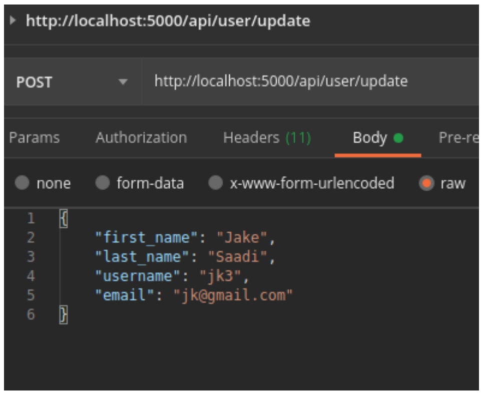
```
{
    "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJpZCI6IjVlZmJiZjUzZDRmN2FiNzZjODRkYWU2YSIsImlhdCI6MTU5MzgyNTgyMywiZXhwIjoxNTk0MTg1ODIzfQ.
        c1bfKKxwqr1qKZhQXDxDf6UufDHE6ngk2cD5_1TdoX4"
}
```

Fig. 3. Successful login sends back token



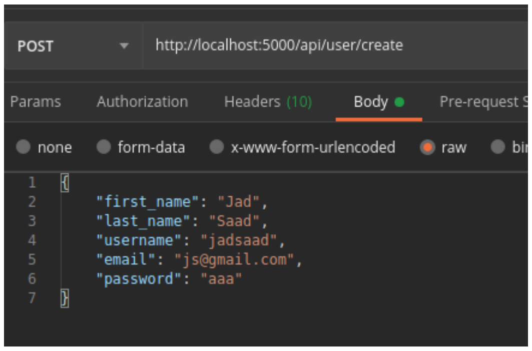
	KEY	VALUE
✓	Content-Type	application/json
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVl
	Key	Value

Fig. 4. Headers



```
{
    "first_name": "Jake",
    "last_name": "Saadi",
    "username": "jk3",
    "email": "jk@gmail.com"
}
```

Fig. 5. Update



```
{
    "first_name": "Jad",
    "last_name": "Saad",
    "username": "jadsaad",
    "email": "js@gmail.com",
    "password": "aaa"
}
```

Fig. 6. Register

Of course all of the above could be done on the web applications front-end on heroku https://assessmentkfl.herokuapp.com/ or http://localhost:3000

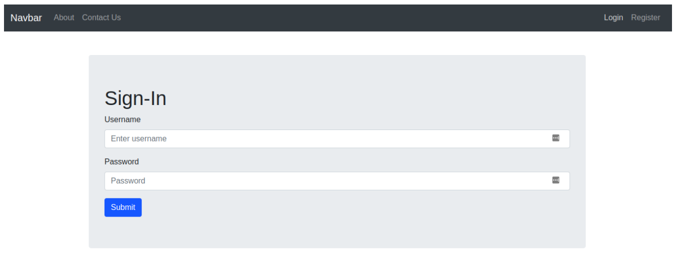
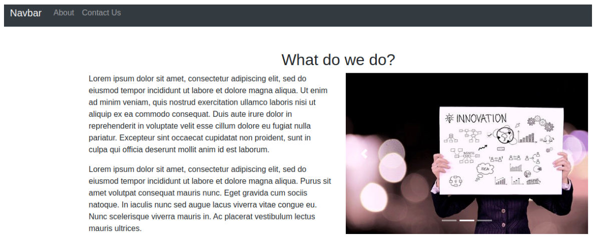


Fig. 7. Login form

II. Assessment 2

Assessment 2 was completed using react and react-bootstrap. The task results can be seen on the web app on https://assessmentkfl.herokuapp.com/ or after cloning and running the app at http://localhost:3000.

1) **login form:**
   heroku: https://assessmentkfl.herokuapp.com/login
   localhost: http://localhost:3000/login
2) **navigation bar:** including in all pages
   heroku: https://assessmentkfl.herokuapp.com
   localhost: http://localhost:3000
3) **contain info explaining about the business:**
   heroku: https://assessmentkfl.herokuapp.com/about
   localhost: http://localhost:3000/about
4) **contact form:**
   heroku: https://assessmentkfl.herokuapp.com/contact
   localhost: http://localhost:3000/contact



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Purus sit amet volutpat consequat mauris nunc. Eget gravida cum sociis natoque. In iaculis nunc sed augue lacus viverra vitae congue eu. Nunc scelerisque viverra mauris in. Ac placerat vestibulum lectus mauris ultrices.

Fig. 8. About

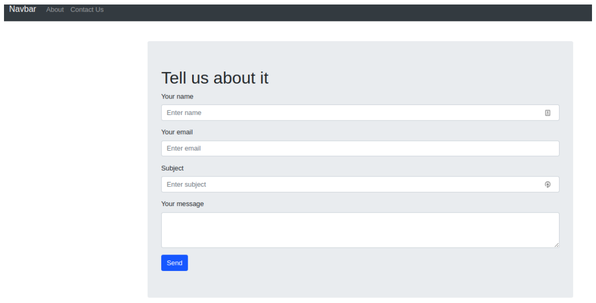


Fig. 9. Contact us

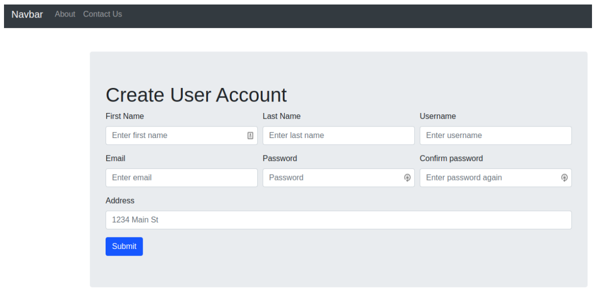


Fig. 10. Register

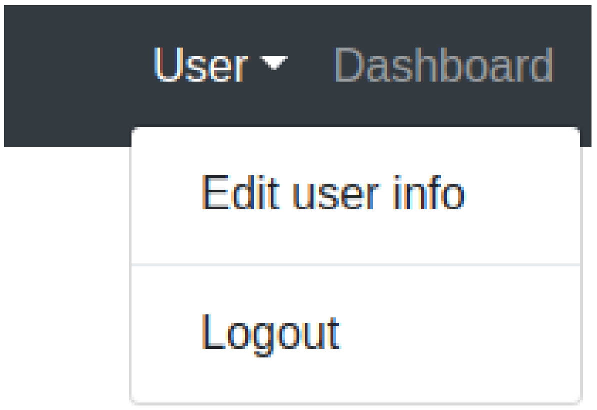
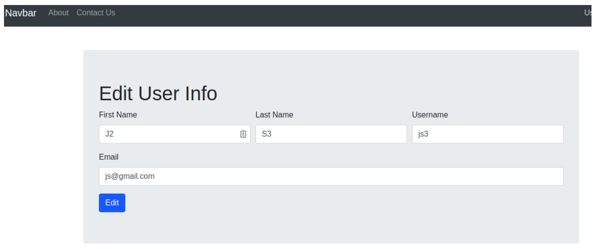


Fig. 11. Drop-down



Edit User Info

First Name	Last Name	Username
J2	S3	js3

Email: js@gmail.com

Fig. 12. Edit user info form

III. Assessment 1&2 in a web application

A. User Stories

1) As a new user, I shall be able to create a new account.
2) As a user, I shall be able to login with username and password
3) As a user, I shall be able to access private routes/
4) As a user, I shall be able to edit user information
5) As a user, I shall be able to view web pages that host info about the business.
6) As a user, I shall be able to visit a contact form with subject, email and description.

B. Front-end

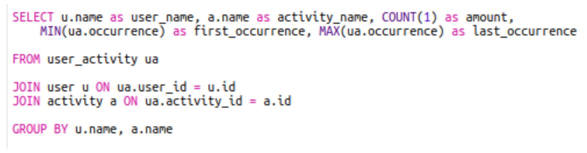
1) **Language:** JavaScript
2) **Library:** React, Axios, Bootstrap
3) **State Management:** Redux
4) **Middleware:** Thunk
5) **Other dependencies:** uuid

C. Back-end

1) **Run-time Environment:** NodeJS
2) **Library:** Express, Mongoose, Passport
3) **Database:** MongoDB
4) **Middleware:** Passport
5) **Other dependencies:** jsonwebtoken, cors, config, bcryptjs

IV. Assessment 3

The Select function for the given task is:



```
SELECT u.name as user_name, a.name as activity_name, COUNT(1) as amount,
    MIN(ua.occurrence) as first_occurrence, MAX(ua.occurrence) as last_occurrence

FROM user_activity ua

JOIN user u ON ua.user_id = u.id
JOIN activity a ON ua.activity_id = a.id

GROUP BY u.name, a.name
```

Fig. 13. SQL for Assessment 3