ASSIGNMENT-4

IMPLEMENTATION OF BACK-END FUNCTIONALITY

HAUL-SHARE

Team- RUNTIME TERROR

ADVANCED WEB SERVICES-CSCI570

Submitted by:

Name : Archanaapriya Nallasivan

Banner number : B00811602

A.4.1. GIT REPOSITORY LINK

INSTALLATION AND SETUP

I have pushed my code in the branch ‘Login-Reg-Backend-B00811602’ .

Repository link - https://git.cs.dal.ca/holla/haul-share-project [1] .

Branch worked on : _https://git.cs.dal.ca/holla/haul-share-project/tree/Login-Reg-Backend-B00811602 [1] .

PROJECT SETUP ON LOCAL MACHINE FROM GITLAB

1. Clone project from gitlab.cs.dal.ca

2. Go to the directory 'Haul-Share-Project_Group5' on the terminal/console.

3. Run 'npm install' on the terminal to download necessary libraries to run the application. (Install node[2] and npm[7] before this step)

4. After the installation, run 'npm start' on the terminal. The server files need not be explicitly run as they are concurrently run with front end.

5. A browser will open locally and run the application on 'localhost:3000' (The port might

change depending on the applications running on the user's browser)

6. ‘server.js’ could be run using node server.js command.

PROJECT LINK ON HEROKU

https://haul-share-official.herokuapp.com/ [8] .

LOGIN DETAILS

You can also register with the system and then login using that credentials, or use the login credentials which is already registered and the user details are stored in database.

Username: haul@share.com

Password: hauls

A.4.2. FEATURE AND ITS DESCRIPTION

As a part of assignment 4, front end and back end of User Management feature has been completed.

TASKS INCLUDED

1. Login page
2. Registration page

Login page is the landing screen of Haul-Share application. The user should be registered in order to access any of its features. In other words, first time users should be registered with the application before logging in.

TECHNOLOGIES USED

Front-end- ReactJS, Bootstrap, Material-UI.

Back- end-services- Node.js, Express, nodemon, bcrypt (to hash passwords), Cors.

Database - MongoDB, Mongoose is used to connect back end services to database.

MongoDB Atlas is used as it is a cloud based services .

The application is developed using the MERN stack

COMPATABILITY

The application is responsive and compatible with different browsers.

The application is W3C compliant.

Login and signup is made with all validations and it is also compatible in different browsers

A.4.3. BOILERPLATE USED

joi : Object schema description language and validator for JavaScript objects [5].

joi-objectid: MongoDB objectId validator [6].

Note : These are used for validations.

A.4.4. IMAGES AND LOGOS - AUTHOR ATTRIBUTION

Given author attribution in code comments.

In Signup and Login, icons were used from the project proposal submission. (designed by meghna)

A.4.5. CODE MODIFICATIONS

I have used the below piece of code for validations using boilerplates. [9]

```
   uname: Joi.string().min(2).max(25).required(),
   email_reg: Joi.string().min(2).max(100).required().email(),
   new_pass: Joi.string().min(5).max(100).required(),
   confirm_pass: Joi.string().min(5).max(100).required()
```

I have used the below code from stack overflow to resolve the issue, [11].

```
   app.use(function(req, res, next) {
   res.setHeader('Access-Control-Allow-Origin', '*');
   res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH,
DELETE');
   res.setHeader('Access-Control-Allow-Headers', 'Origin,X-Requested-With,Content-
type,Accept');
   next();
```

A.4.6. FOLDER STRUCTURE

1. All the front end files are in the ‘src/components’ folder.
2. All images are in ‘public/images’ folder.
3. All the backend files are present in ‘src/server’ folder.

4. Routes is present in utils.js.

A.4.7.FRONT END VALIDATIONS

FILES WORKED AND VALIDATIONS

1. I have written a ‘register’ POST method to create add the user to the database when he/she registers in our application. If the user is already registered, he/she is taken to the login page. Proper validations are done in the registration page.
2. I have written a ‘login’ POST method to allow the user to login to our application. The credentials are validated and then they are allowed to enter the home page.
3. Password stored in database is encrypted.
   MongoDB Atlas credentials:
   User - ar803514@dal.ca , Password - B00811602
4. The files I worked on ‘server.js’ (Backend) and ‘Login.js’ (Frontend) . Few validation changes are made in the existing front end code. Proper error recovery messages added on both front-end and back-end.

A.4.8. REFERENCES

[1] "Sign in", GitLab, 2019. [Online]. Available: https://git.cs.dal.ca/users/sign_in. [Accessed: 17- Jul- 2019].

[2] N. Foundation, "Node.js", Node.js, 2019. [Online]. Available: https://nodejs.org/en/. [Accessed: 17- Jul- 2019].

[3]"Starting with Authentication (A tutorial with Node.js and MongoDB)", *Medium*, 2019. [Online]. Available: https://medium.com/createdd-notes/starting-with-authentication-a-tutorial-with-node-js-and-mongodb-25d524ca0359. [Accessed: 14- Jul- 2019].

[4]"Using CORS - HTML5 Rocks", *HTML5 Rocks - A resource for open web HTML5 developers*, 2019. [Online]. Available: https://www.html5rocks.com/en/tutorials/cors/. [Accessed: 14- Jul- 2019].

[5] "joi", *npm*, 2019. [Online]. Available: https://www.npmjs.com/package/joi. [Accessed: 12-Jul- 2019].

[6] "joi-objectid", *npm*, 2019. [Online]. Available: https://www.npmjs.com/package/joi-objectid. [Accessed: 12- Jul- 2019].

[7] "npm | build amazing things", *Npmjs.com*, 2019. [Online]. Available: https://www.npmjs.com/. [Accessed: 17- Jul- 2019].

[8] Heroku. [Online]. Available: https://dashboard.heroku.com/apps. [Accessed: 17-Jul-2019].

[9]"Composing React Components – Vegibit", *Vegibit.com*, 2019. [Online]. Available: https://vegibit.com/composing-react-components/. [Accessed: 14- Jul- 2019].

[10] "Node.js MongoDB User Registration – Vegibit", *Vegibit.com*, 2019. [Online]. Available: https://vegibit.com/node-js-mongodb-user-registration/. [Accessed: 14- Jul- 2019].

[11] D. node.js and M. Panah, "Difference between res.setHeader and res.header in node.js", Stack Overflow, 2019. [Online]. Available: https://stackoverflow.com/questions/40840852/difference-between-res-setheader-and-res-header-in-node-js. [Accessed: 18- Jul- 2019].

[12] "React – A JavaScript Library for Building User Interfaces." – A JavaScript Library for Building User Interfaces, reactjs.org/.

[13] "React Bootstrap." React, react-bootstrap.github.io/

[14] "The World's Most Popular React UI Framework - Material-UI." Material, material-ui.com/

IMAGE REFERENCES

[1]"Free Logo Maker - Create Your Own Logo in Minutes!" Free Logo Maker - Create Your Own Logo in Minutes!, logomakr.com/.

[2] "User Icons." Free Download, PNG and SVG, https://icons8.com/icons/set/user

[3] "Key Icons." Free Download, PNG and SVG, http://icons8.com/icons/set/key

[4] "Email Icons." Free Download, PNG and SVG, http://icons8.com/icons/set/email

[5] "Phone Icons." Free Download, PNG and SVG, http://icons8.com/icons/set/phone

[6] "Lock Icons." Free Download, PNG and SVG, http://icons8.com/icons/set/lock