Arjun Bhagat	Github: smeerj
Thamizarasu Sankarasubramanian	Github: thamizarasus
Shaurya Chawla	Github: shauryachawla15
Joseph Behroz	Github: jbehroz

Syntax Squad Web Development CSC 317-03

Assignment 3 – SQL and JS

Description:
The main goal of this assignment was to get a connection to an SQL database working with JavaScript, as well as hashing the password.

Approach:
Once professor scott walked us through the steps of setting up our SQL database with a webpage in lecture, we simply followed the recorded lecture and set it up with our own webpage. We ended up having some issues with the hashing function that is mentioned in the next section, but this portion of our final project was pretty straight forward. Our next step is to implement products into the database and have the webpage display straight from there rather than the hardcoded product pages we have currently.

We also went back and ironed out some of the issues we had with our HTML and CSS in terms of consistency throughout the pages, as well as changing some overall design decisions such as how our products are layed out.

Issues and Resolutions:
MySQL: For this group, we had very little to no experience with MySQL so it was a struggle to get things to work. Even using basic features such as creating a table, adding data to said table, and what type of data each column should hold was a challenge and constantly changing. Even installing it was an issue for some of us since we use different OS's. Once we got the database working for one of us, that person exported their database and shared it with the other members so they could import it and make sure it works on their machine too. Additionally, Google helped us figure out how to write the expressions we used to add or select data from the database.
Node.js: While we talked about and used a little bit of Node previously in class, this was our first time getting into the meat of it and dealing with proper http requests and processing data the user enters. The largest issue was dealing with hashing and salting the passwords since we are not familiar with the asynchronous nature of the bcrpyt library. Once we figured out that we should be putting a majority of our processing within the .then functions, it became much easier.

Analysis:

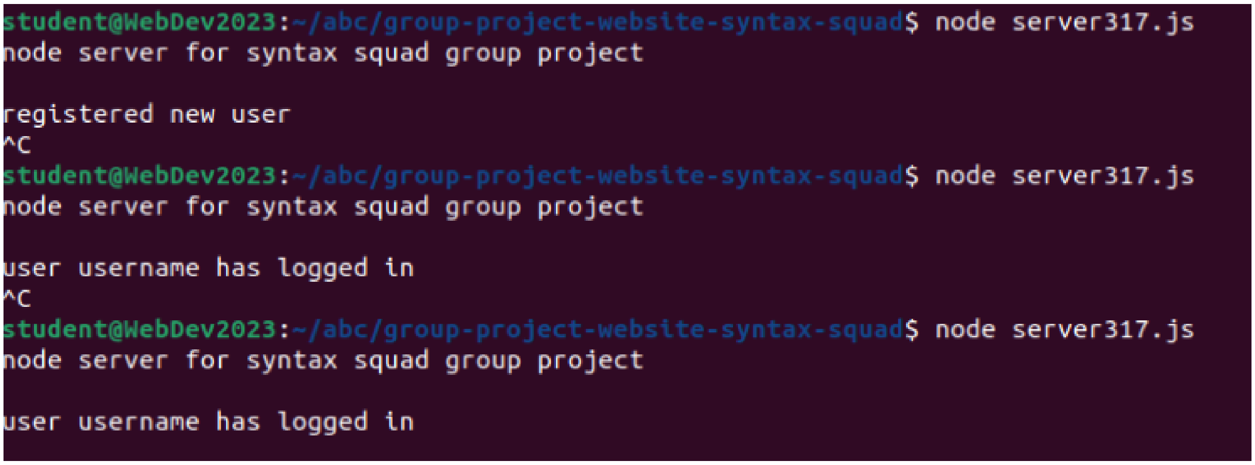
Arjun Bhagat Github: smeerj
Thamizarasu Sankarasubramanian Github: thamizarasus
Shaurya Chawla Github: shauryachawla15
Joseph Behroz Github: jbehroz

Syntax Squad Web Development CSC 317-03

Once again, implementing these new changes made us realize how as a developer you really need to think like a user and how they would use the website, as well as the options that they will need to have an enjoyable and complete experience. We found that after implementing the authentication, there were so many aspects of our website that we hadn't considered such as password changes, having not just a login page but a register page as well, and it opened our eyes to how much you must consider as a developer. Had we not taken the time to consider these, the experience of our website would have been awful and missing a lot of key features. We thought about what the user would need and implemented those changes accordingly.

<u>Screenshots:</u>

The node server running and showing when a user registers or logs in

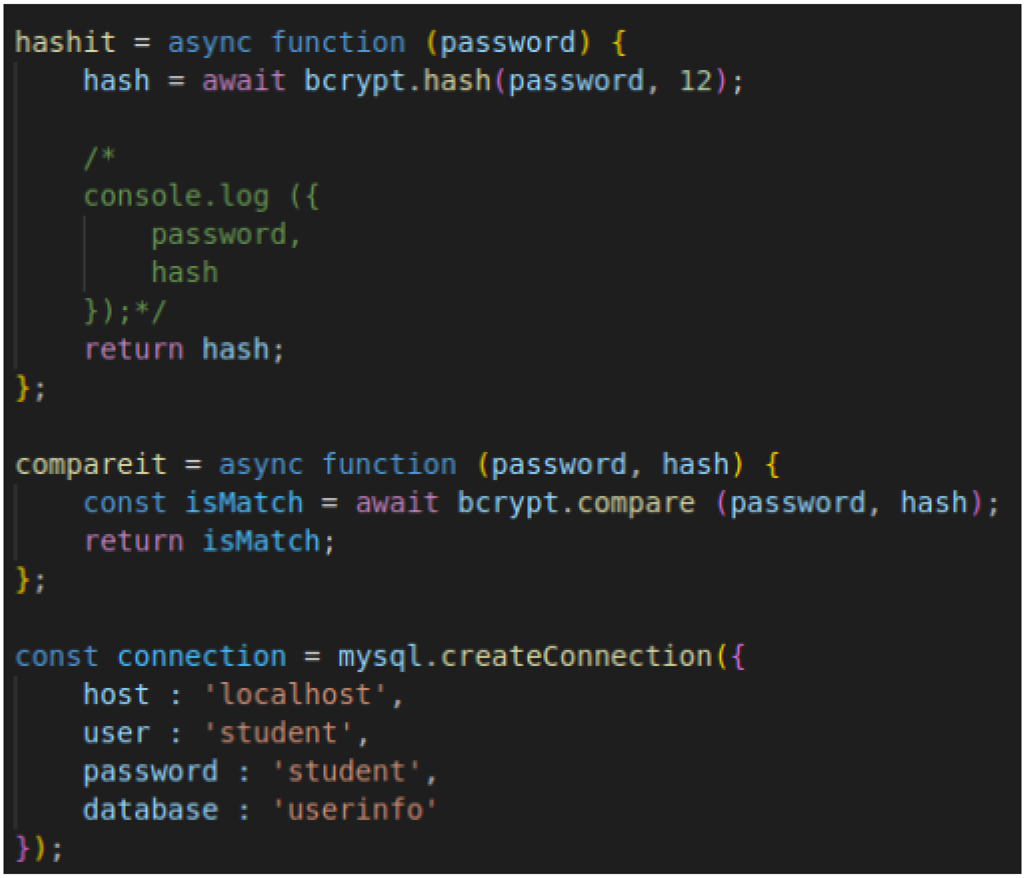


```
student@WebDev2023:~/abc/group-project-website-syntax-squad$ node server317.js
node server for syntax squad group project

registered new user
^C
student@WebDev2023:~/abc/group-project-website-syntax-squad$ node server317.js
node server for syntax squad group project

user username has logged in
^C
student@WebDev2023:~/abc/group-project-website-syntax-squad$ node server317.js
node server for syntax squad group project

user username has logged in
```

Arjun Bhagat Github: smeerj
Thamizarasu Sankarasubramanian Github: thamizarasus
Shaurya Chawla Github: shauryachawla15
Joseph Behroz Github: jbehroz

Syntax Squad Web Development CSC 317-03

Implementation of the hashing and comparison functions as well as the MySQL connection

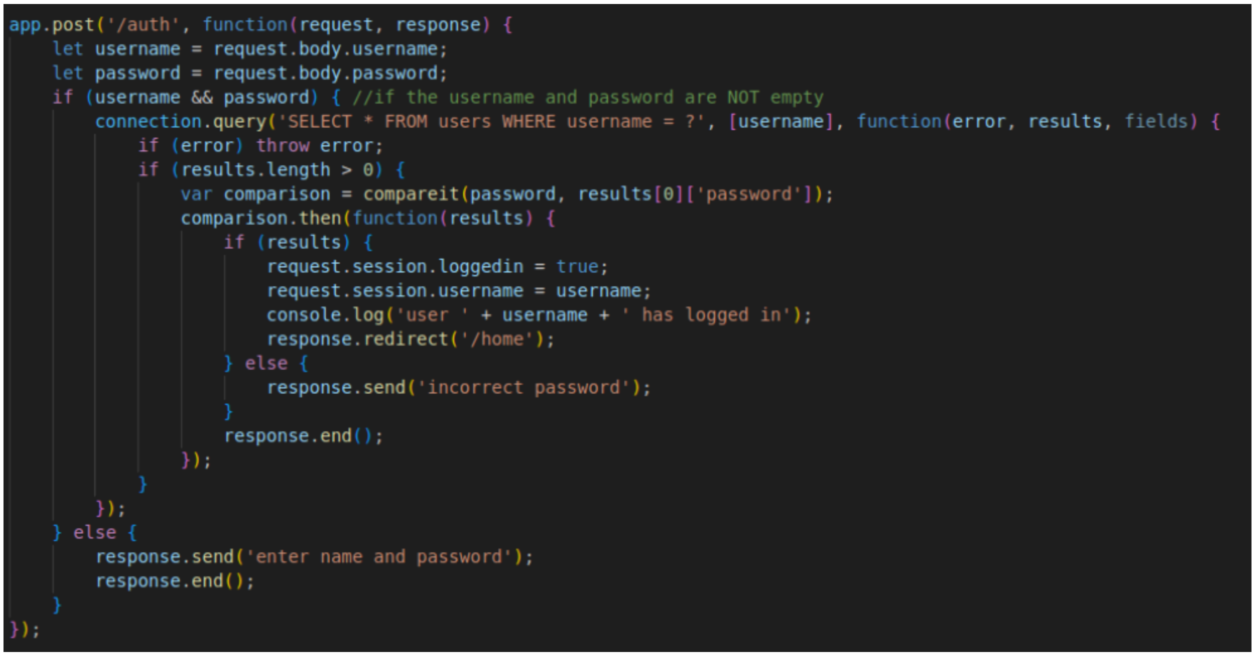


```
hashit = async function (password) {
    hash = await bcrypt.hash(password, 12);

    /*
    console.log ({
        password,
        hash
    });*/
    return hash;
};

compareit = async function (password, hash) {
    const isMatch = await bcrypt.compare (password, hash);
    return isMatch;
};

const connection = mysql.createConnection({
    host : 'localhost',
    user : 'student',
    password : 'student',
    database : 'userinfo'
});
```

Arjun Bhagat Github: smeerj
Thamizarasu Sankarasubramanian Github: thamizarasus
Shaurya Chawla Github: shauryachawla15
Joseph Behroz Github: jbehroz

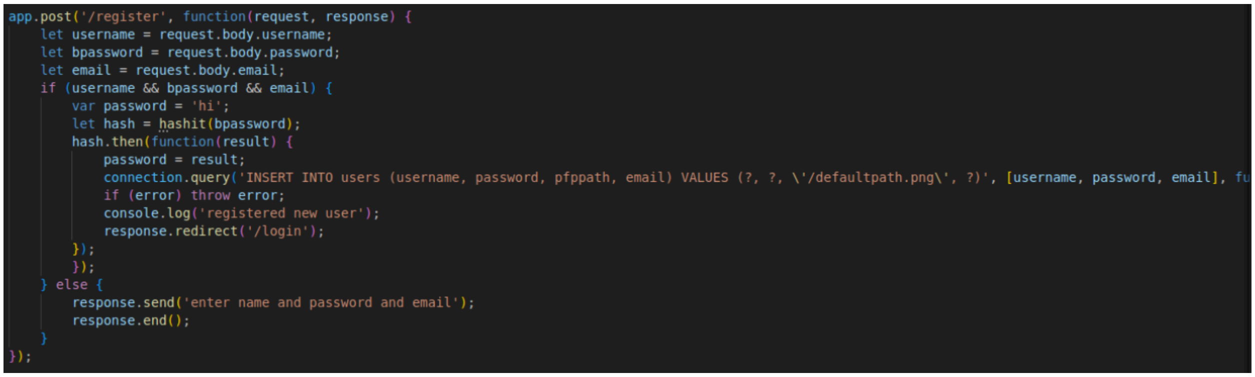
Syntax Squad Web Development CSC 317-03

Implementation of authenticating the users login request



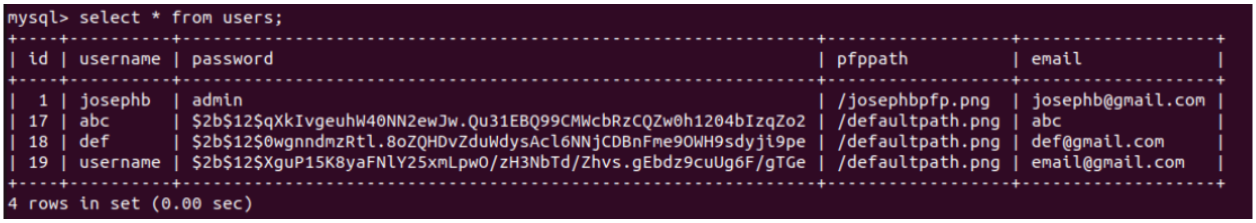
```
app.post('/auth', function(request, response) {
    let username = request.body.username;
    let password = request.body.password;
    if (username && password) { //if the username and password are NOT empty
        connection.query('SELECT * FROM users WHERE username = ?', [username], function(error, results, fields) {
            if (error) throw error;
            if (results.length > 0) {
                var comparison = compareit(password, results[0]['password']);
                comparison.then(function(results) {
                    if (results) {
                        request.session.loggedin = true;
                        request.session.username = username;
                        console.log('user ' + username + ' has logged in');
                        response.redirect('/home');
                    } else {
                        response.send('incorrect password');
                    }
                    response.end();
                });
            }
        });
    } else {
        response.send('enter name and password');
        response.end();
    }
});
```

Implementation of registering a new user



```
app.post('/register', function(request, response) {
    let username = request.body.username;
    let bpassword = request.body.password;
    let email = request.body.email;
    if (username && bpassword && email) {
        var password = 'hi';
        let hash = hashit(bpassword);
        hash.then(function(result) {
            password = result;
            connection.query('INSERT INTO users (username, password, pfppath, email) VALUES (?, ?, \'/defaultpath.png\', ?)', [username, password, email], fu
            if (error) throw error;
            console.log('registered new user');
            response.redirect('/login');
        });
        });
    } else {
        response.send('enter name and password and email');
        response.end();
    }
});
```

MySQL database with some test members



```
mysql> select * from users;
+----+----------+--------------------------------------------------------------+------------------+-------------------+
| id | username | password                                                     | pfppath          | email             |
+----+----------+--------------------------------------------------------------+------------------+-------------------+
|  1 | josephb  | admin                                                        | /josephbpfp.png  | josephb@gmail.com |
| 17 | abc      | $2b$12$qXkIvgeuhW40NN2ewJw.Qu31EBQ99CMWcbRzCQZw0h1204bIzqZo2 | /defaultpath.png | abc               |
| 18 | def      | $2b$12$0wgnndmzRtl.8oZQHDvZduWdysAcl6NNjCDBnFme9OWH9sdyji9pe | /defaultpath.png | def@gmail.com     |
| 19 | username | $2b$12$XguP15K8yaFNlY25xmLpwO/zH3NbTd/Zhvs.gEbdz9cuUg6F/gTGe | /defaultpath.png | email@gmail.com   |
+----+----------+--------------------------------------------------------------+------------------+-------------------+
4 rows in set (0.00 sec)
```