Illustrations

Initial Step

First, We need to start the database and the app.

current state of database (after initial database setup):

To view `appusers` table, follow Steps `15` to `18` from notes.


```
cd sqlibasic/
node sessions.js
```

Visit https://localhost:3000/

Click `Advanced`

Click `Accept the Risk and Continue`

Click `Create account.`

That multiple users can create an account.

Create Two Accounts

Account 1 Details (use chrome client)

```
username: daisy
password: A2345678b+
info: The sky is blue
```

Create An Account

Username:
daisy

Password (10 characters minimum):
••••••••••

Optional Field:

Information:
The sky is blue

Register

Password Rules

at least 10 characters

at least 1 uppercase character (A-Z)

at least 1 lowercase character (a-z)

at least 1 digit (0-9)

at least 1 special character (punctuation)

Already have an account? Login.

Account Created! Please Login

Login

Account 2 Details (use Firefox client)

```
username: john
password: Bpd2*09w_P
info: I need to go to the store.
```

Create An Account

Username:
john

Password (10 characters minimum):
••••••••••

Optional Field:

Information:
I need to go to the store.

Register

Password Rules

at least 10 characters

at least 1 uppercase character (A-Z)

at least 1 lowercase character (a-z)

at least 1 digit (0-9)

at least 1 special character (punctuation)

Already have an account? Login.

Account Created! Please Login

Login

```
SELECT * from appusers;
```

```
MariaDB [users]> SELECT * from appusers;
+----------+--------------------------------------------------------------+----------------------------+---------------+
| username | password                                                     | info                       | session       |
+----------+--------------------------------------------------------------+----------------------------+---------------+
| daisy    | $2b$10$e7r827aDmajZgyTq1sxHVuuj1/uFj321pbyNvI4e6WsNQaumwC2C2 | The sky is blue            | not logged in |
| john     | $2b$10$8VKW.miHf0/t9z47PXflvuyciXzLKsWzb5ygxY/WvmqeVlYHHw0l. | I need to go to the store. | not logged in |
| mikhail  | $2b$10$MPwhYcT.NTOy6z0C0CrP4eNsQkd4GczQTm7GTdqGQRi9CT7nb5cRC | Where are the cats?        | not logged in |
| testuser | $2b$10$DPYpj2zzZk7WpQ18Z6jmZOeva5o5zcVExMtK4nbRLnxhO9cqCrdmq | Hellllo???                 | not logged in |
+----------+--------------------------------------------------------------+----------------------------+---------------+
4 rows in set (0.000 sec)

MariaDB [users]>
```

That multiple users can login and correct information is shown upon login/visiting the website with a valid session.

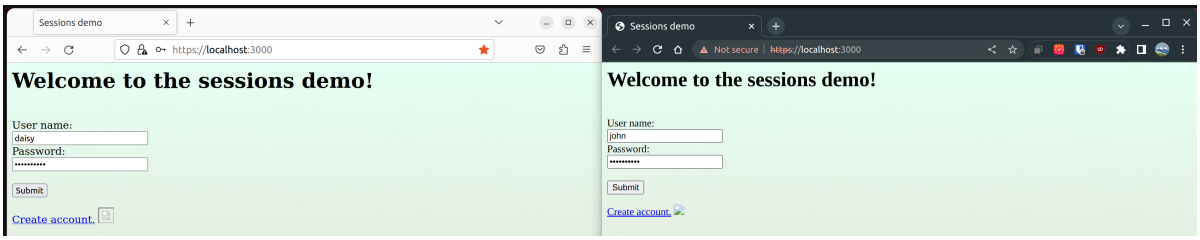
Login Two Accounts

Account 1 Details (use chrome client)

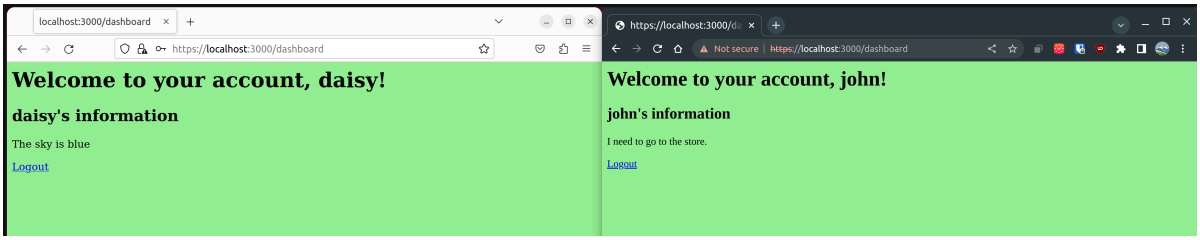
```
username: daisy
password: A2345678b+
```

Account 2 Details (use Firefox client)

```
username: john
password: Bpd2*09w_P
```



Press `submit`



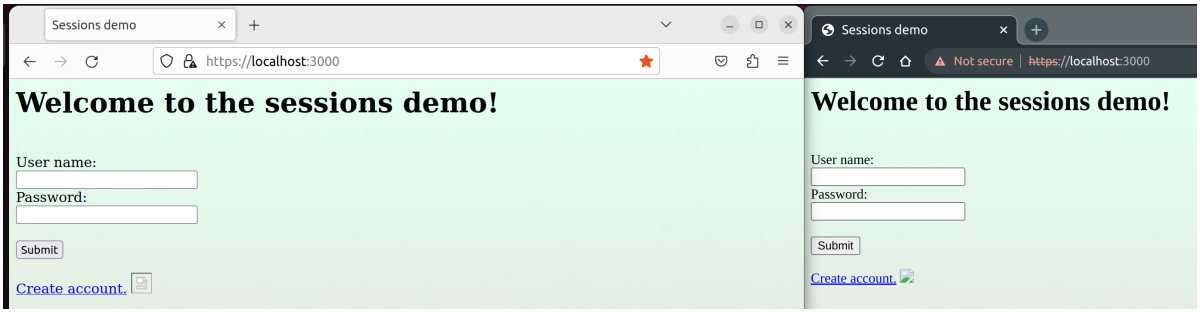
current state of database, after multiple account login:

```
SELECT * from appusers;
```

```
MariaDB [users]> SELECT * from appusers;
+----------+--------------------------------------------------------------+-------------------------+--------------------------------------+
| username | password                                                     | info                    | session                              |
+----------+--------------------------------------------------------------+-------------------------+--------------------------------------+
| daisy    | $2b$10$e7r827aDmajZgyTq1sxHVuuj1/uFj321pbyNvI4e6WsNQaumwC2C2 | The sky is blue         | 5f063018-f46e-4f1d-8d3e-245aa79a9b0f |
| john     | $2b$10$8VKW.miHf0/t9z47PXflvuyciXzLKsWzb5ygxY/WvmqeVlYHHw0l. | I need to go to the store. | 96f9208a-f15f-48c8-aea8-85af432ad89d |
| mikhail  | $2b$10$MPwhYcT.NTOy6z0C0CrP4eNsQkd4GczQTm7GTdqGQRi9CT7nb5cRC | Where are the cats?     | not logged in                        |
| testuser | $2b$10$DPYpj2zzZk7WpQ18Z6jmZOeva5o5zcVExMtK4nbRLnxhO9cqCrdmq | Hellllo???              | not logged in                        |
+----------+--------------------------------------------------------------+-------------------------+--------------------------------------+
4 rows in set (0.000 sec)

MariaDB [users]>
```

That user's can logout and after logout the user is directed to login if they visit the site.

Press `Logout` on both `daisy` and `john` accounts. Page should redirect after pressing `Logout`.



current state of database, after multiple account Logout:

```
SELECT * from appusers;
```

```
MariaDB [users]> SELECT * from appusers;
+----------+--------------------------------------------------------------+-------------------------+--------------+
| username | password                                                     | info                    | session      |
+----------+--------------------------------------------------------------+-------------------------+--------------+
| daisy    | $2b$10$e7r827aDmajZgyTq1sxHVuuj1/uFj321pbyNvI4e6WsNQaumwC2C2 | The sky is blue         | not logged in |
| john     | $2b$10$8VKW.miHf0/t9z47PXflvuyciXzLKsWzb5ygxY/WvmqeVlYHHw0l. | I need to go to the store. | not logged in |
| mikhail  | $2b$10$MPwhYcT.NTOy6z0C0CrP4eNsQkd4GczQTm7GTdqGQRi9CT7nb5cRC | Where are the cats?     | not logged in |
| testuser | $2b$10$DPYpj2zzzZk7WpQ18Z6jmZOeva5o5zcVExMtK4nbRLnxhO9cqCrdmq | Hellllo???              | not logged in |
+----------+--------------------------------------------------------------+-------------------------+--------------+
4 rows in set (0.001 sec)

MariaDB [users]>
```

An explanation and screenshots explaining how the code meets each requirement.

Step 1. Add a "session" attribute to the appusers table.

Please look at step 11 in notes

This is what `appusers` should look like:

```
MariaDB [users]> Describe appusers;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| username | varchar(255) | NO   | PRI | NULL    |       |
| password | varchar(255) | YES  |     | NULL    |       |
| info     | varchar(255) | YES  |     | NULL    |       |
| session  | varchar(255) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
4 rows in set (0.001 sec)

MariaDB [users]>
```

Step 2. When the user logs in, store the session ID in the "session" attribute of the user's record. Please do not make any other changes to the database.

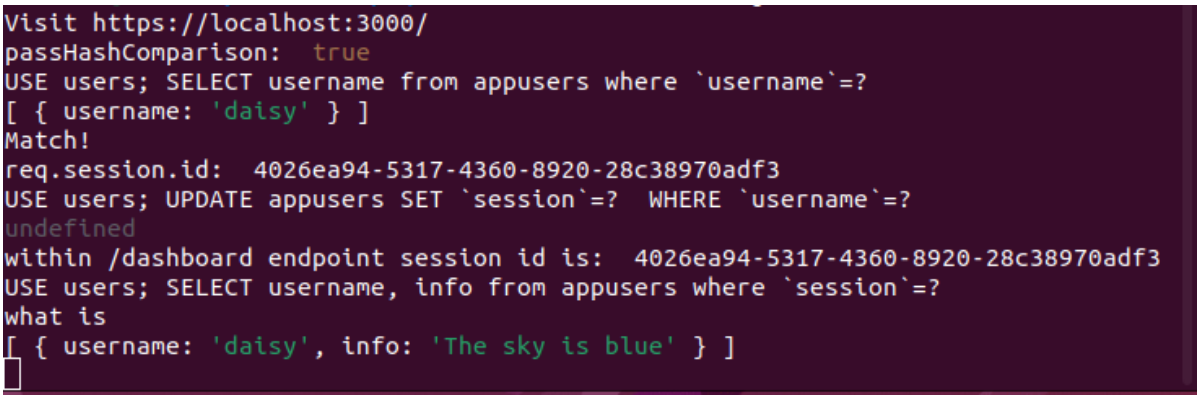
Please look at `/login` POST endpoint for code in `sessions.js`.

The most important piece of code is within:

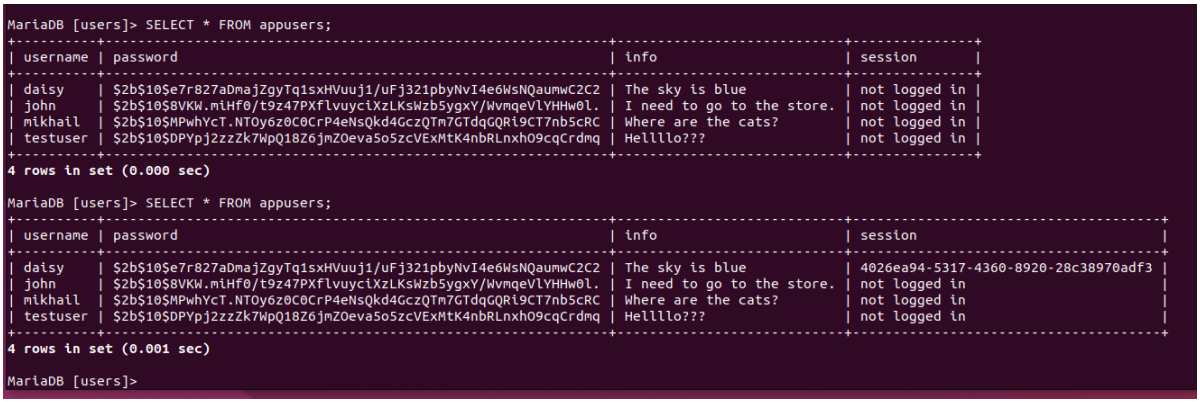
```
        if (match) {

            // create random id and store in cookie
            req.session.id = uuidv4()
            console.log("req.session.id: ", req.session.id)

            //store in db
            let query = "USE users; UPDATE appusers SET `session`=?  WHERE
`username`=?";
            console.log(query);

            mysqlConn.query(query, [req.session.id, userName],function (err, res)
{
                if (err) throw err;

                console.log(res[1]['message'])
```

```
            })


            res.redirect('/dashboard');
        }
        else {
            // If no matches have been found, we are done
            res.send("<b>Wrong</b>");
        }
```

Console messages when running `sessions.js`:



Database before and after login submission:



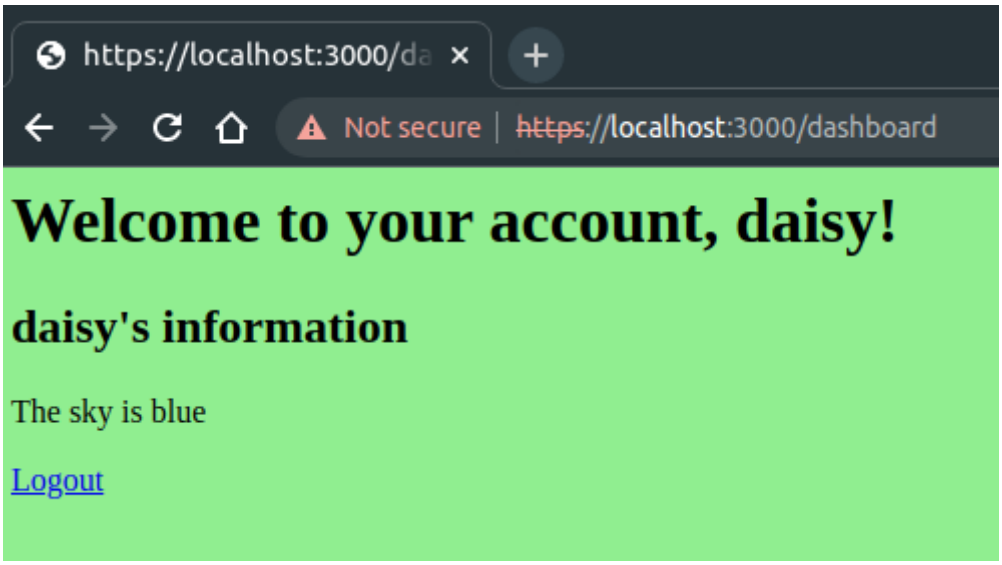
Step 4. When the user logs out, the session ID is deleted from the user's record (or is replaced with some place holder value such as "not logged in".

Please look at `/logout` GET endpoint for code in `sessions.js`.

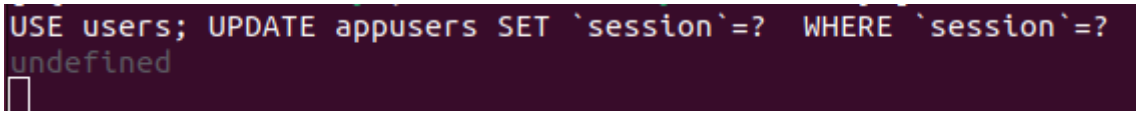
The most important piece of code is within:

```
    // clear session id from db
    let message = "not logged in"
    let query = "USE users; UPDATE appusers SET `session`=?  WHERE `session`=? ";
    console.log(query);

    mysqlConn.query(query, [message, req.session.id],function (err, res) {
        if (err) throw err;

        console.log(res[1]['message'])
    })
```

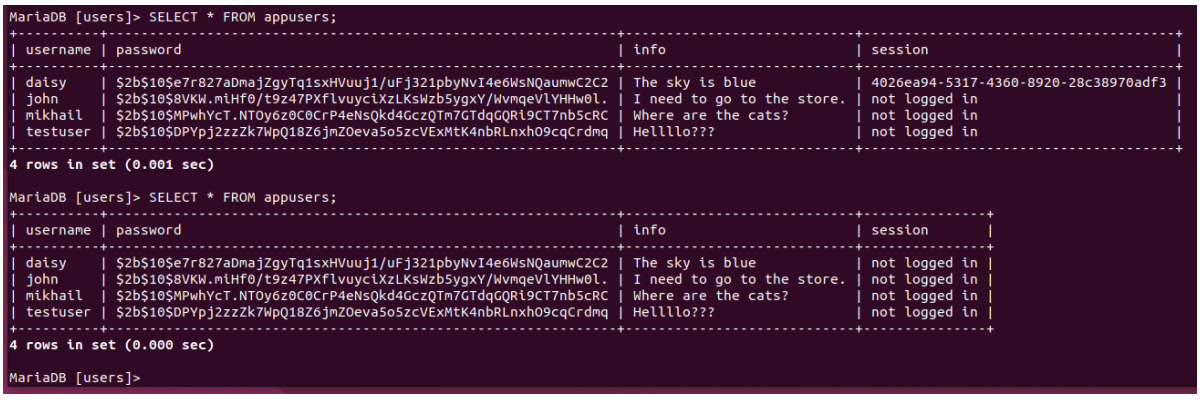
Press `Logout`



Console messages when running `sessions.js`:



Database before and after Logout submission:



Step 5. Add an option to allow users to register (i.e., add their user name and password)

Please look at `/create-account` endpoints for code in `sessions.js`.

The most important piece of code is within `/create-account` POST endpoint:

```
// Get the username and password data from the form
    let userName = req.body.username;
    let password = req.body.password;

    console.log("req.body.information: ", req.body.information);

    let initialInfo = req.body.information;

    let initialSession = "not logged in"

    bcrypt.hash(password, saltRounds, function (err, hash) {
        // Store hash in your password DB

        // Construct the query
        let query = "USE users; INSERT INTO appusers (`username`, `password`,
`info`, `session`) VALUES (?, ?, ?, ?)";


        mysqlConn.query(query, [userName, hash, initialInfo, initialSession],
function (err, qResult) {
console.log(query);
            if (err) throw err;

            console.log(qResult[1]);

            res.redirect('/successpage');

        });
    })
```

Also take a look at the created views: `create-account-page.ejs`, `successpage.ejs` within `views/` folder.

Here is `create-account-page.ejs` :

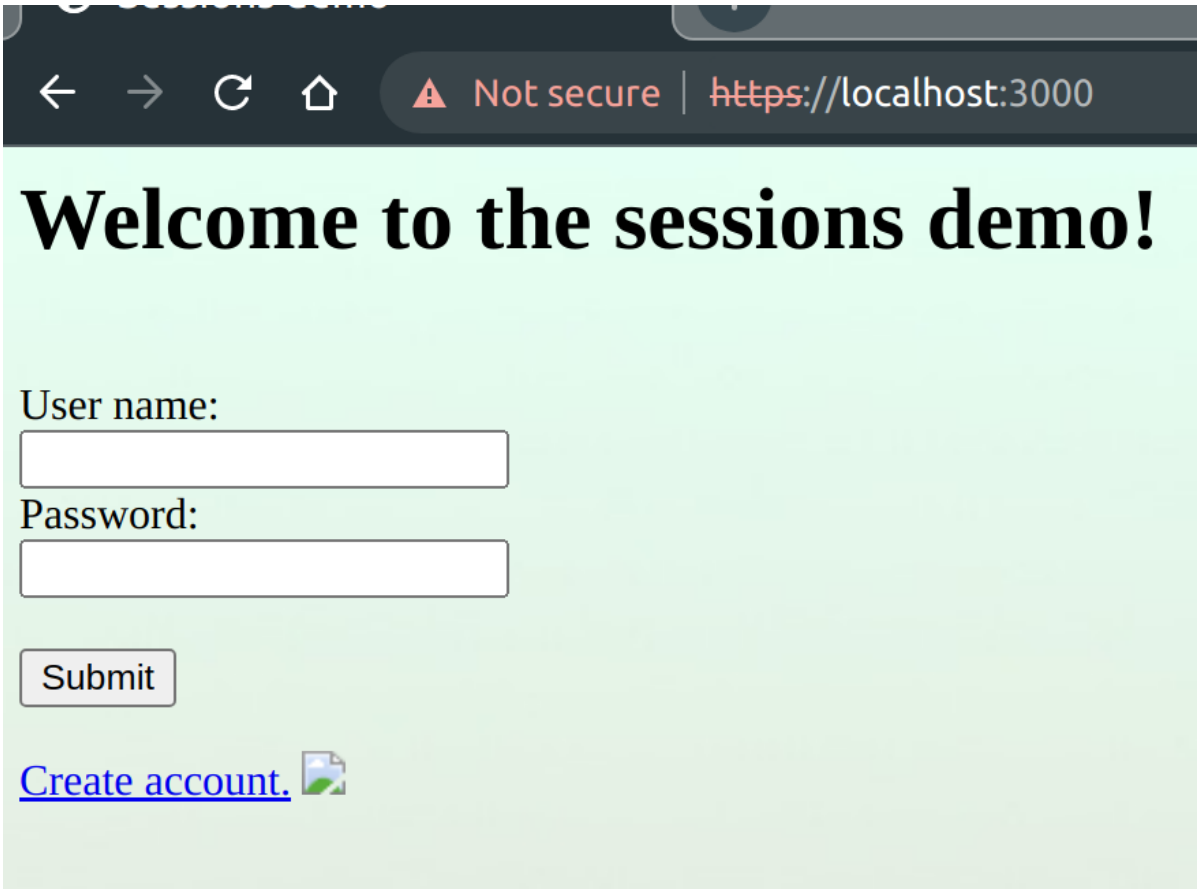
```
<html>
<title> Sessions demo </title>

<body bgcolor="lightyellow">
  <h1>Create An Account</h1>
  <br>

  <div>
    <form action="/create-account" method="POST">
```

```
      <label for="username">Username:</label><br>
      <input type="text" id="username" name="username"><br><br>
      <label for="pass">Password (10 characters minimum):</label><br>
      <input type="password" id="pass" name="password" minlength="10" required>
<br><br>

      <p>Optional Field:</p>
      <label for="information">Information:</label><br>
      <textarea name="information" placeholder='Enter information...'
maxlength='1000'></textarea>
      <br>
      <input type="submit" value="Register">
    </form>

    <div>
      <h2>Password Rules</h2>
      <p>at least 10 characters</p>
      <p>at least 1 uppercase character (A-Z)</p>
      <p>at least 1 lowercase character (a-z)</p>
      <p>at least 1 digit (0-9)</p>
      <p>at least 1 special character (punctuation) </p>


    </div>

    <br><br>

    <a href="/">Already have an account? Login.</a>


</body>

</html>
```

Here is `successpage.ejs`. This page will render if account creations is successful.

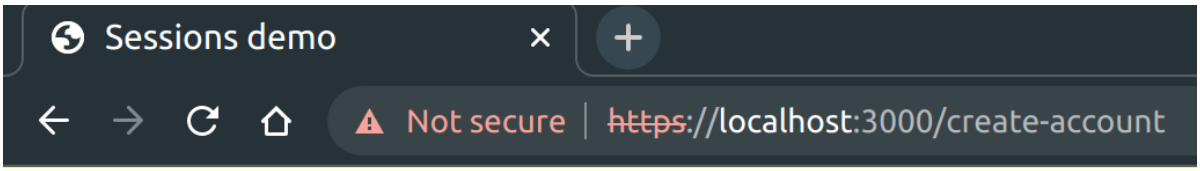
```
<html>
<body bgcolor="lightpink">

<h1> Account Created! Please Login</h1>
<div>
<img src="https://http.cat/201" width="500" height="auto"></img>
</div>

<h2><a href="/"> Login</a></h2>

</body>
</html>
```

Press `Create account.`



Create an account using info:

```
Username: mary
Password: Fp4$1ApaG3
Information: I am showing the creation of mary's account!
```



Create An Account

Username:
mary

Password (10 characters minimum):
••••••••••

Optional Field:

Information:
I am showing the creation of mary's account!

Register

Password Rules

at least 10 characters

at least 1 uppercase character (A-Z)

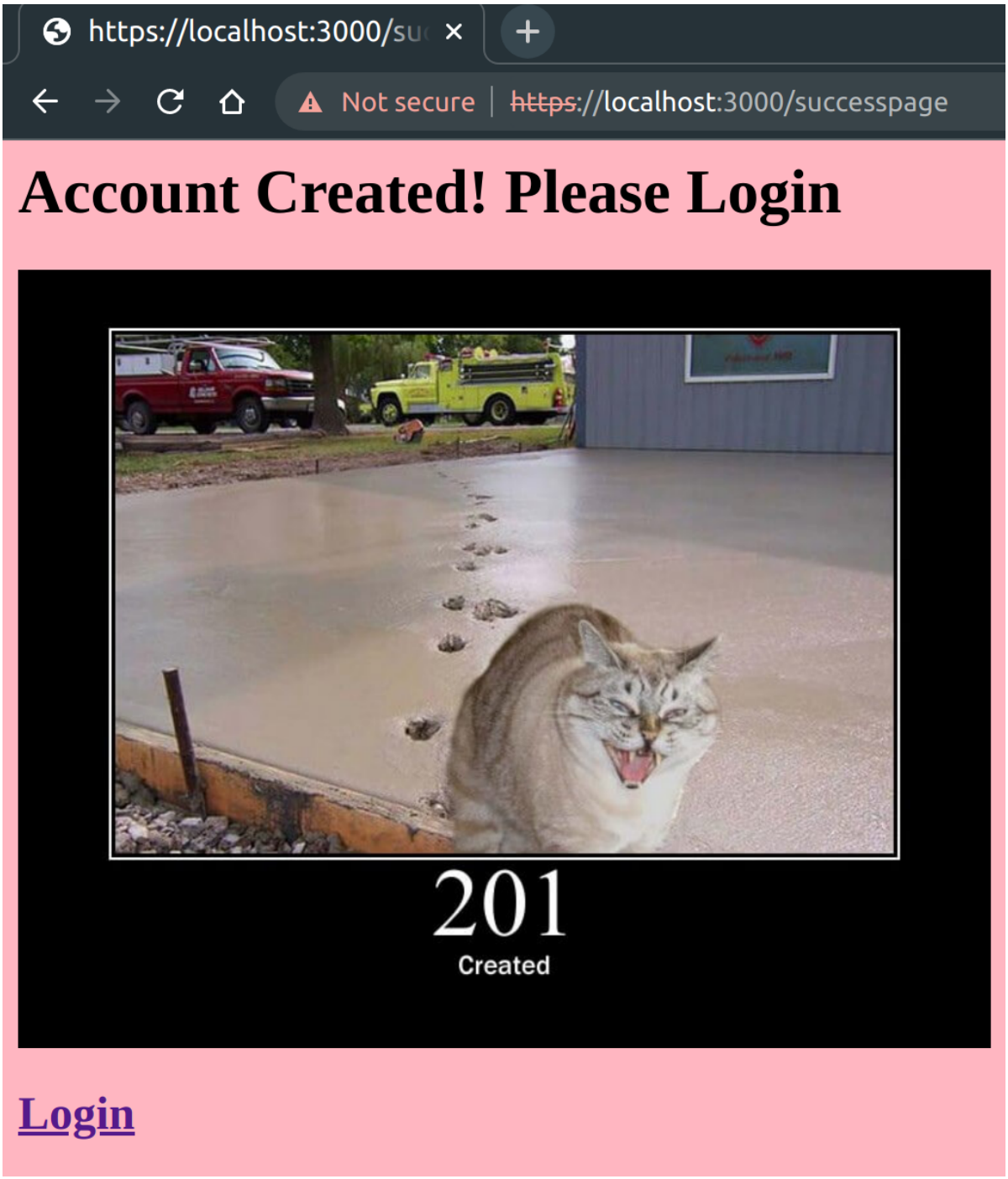
at least 1 lowercase character (a-z)

at least 1 digit (0-9)

at least 1 special character (punctuation)

Already have an account? Login.

Press `Register`



Console messages when running `sessions.js`:

```
middleware: checking password strength...
Password: Strong
Password: 10
Password: [ 'lowercase', 'uppercase', 'number', 'symbol' ]
req.body.information:  I am showing the creation of mary's account!
USE users; INSERT INTO appusers (`username`, `password`, `info`, `session`) VALUES (?, ?, ?, ?)
ResultSetHeader {
  fieldCount: 0,
  affectedRows: 1,
  insertId: 0,
  info: '',
  serverStatus: 2,
  warningStatus: 0
}
```

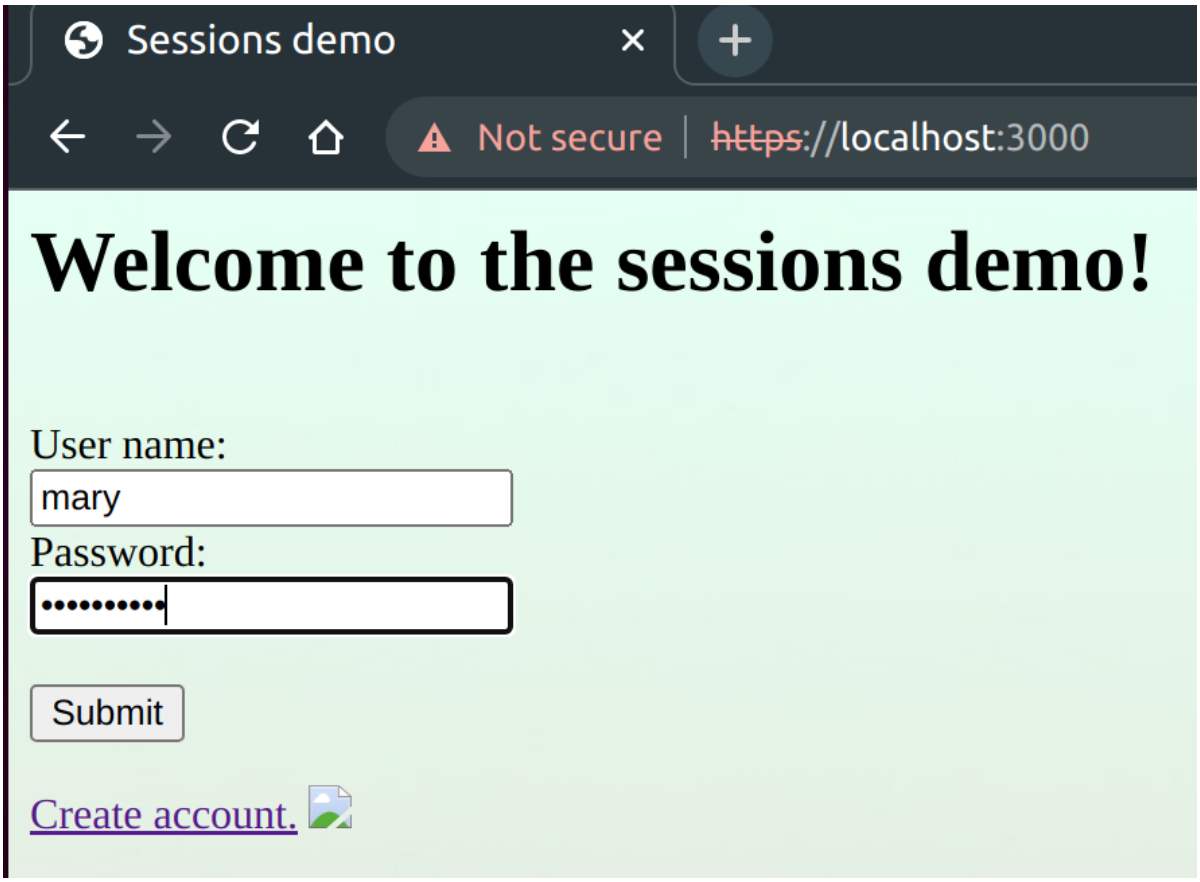
Database before and after `Register` submission:

```
MariaDB [users]> SELECT * FROM appusers;
+----------+--------------------------------------------------------------+-------------------------+--------------+
| username | password                                                     | info                    | session      |
+----------+--------------------------------------------------------------+-------------------------+--------------+
| daisy    | $2b$10$e7r827aDmajZgyTq1sxHVuuj1/uFj321pbyNvI4e6WsNQaumwC2C2 | The sky is blue         | not logged in |
| john     | $2b$10$8VKW.miHf0/t9z47PXflvuyciXzLKsWzb5ygxY/WvmqeVlYHHw0l. | I need to go to the store. | not logged in |
| mikhail  | $2b$10$MPwhYcT.NTOy6z0C0CrP4eNsQkd4GczQTm7GTdqGQRi9CT7nb5cRC | Where are the cats?     | not logged in |
| testuser | $2b$10$DPYpj2zzZk7WpQ18Z6jmZOeva5o5zcVExMtK4nbRLnxhO9cqCrdmq | Hellllo???              | not logged in |
+----------+--------------------------------------------------------------+-------------------------+--------------+
4 rows in set (0.000 sec)

MariaDB [users]> SELECT * FROM appusers;
+----------+--------------------------------------------------------------+-----------------------------------------+--------------+
| username | password                                                     | info                                    | session      |
+----------+--------------------------------------------------------------+-----------------------------------------+--------------+
| daisy    | $2b$10$e7r827aDmajZgyTq1sxHVuuj1/uFj321pbyNvI4e6WsNQaumwC2C2 | The sky is blue                         | not logged in |
| john     | $2b$10$8VKW.miHf0/t9z47PXflvuyciXzLKsWzb5ygxY/WvmqeVlYHHw0l. | I need to go to the store.              | not logged in |
| mary     | $2b$10$56025hEQ..Hql85IcMqDp.B.ytShJdG6NmmCfq.V5WMt1X5ATXoUm | I am showing the creation of mary's account! | not logged in |
| mikhail  | $2b$10$MPwhYcT.NTOy6z0C0CrP4eNsQkd4GczQTm7GTdqGQRi9CT7nb5cRC | Where are the cats?                     | not logged in |
| testuser | $2b$10$DPYpj2zzZk7WpQ18Z6jmZOeva5o5zcVExMtK4nbRLnxhO9cqCrdmq | Hellllo???                              | not logged in |
+----------+--------------------------------------------------------------+-----------------------------------------+--------------+
5 rows in set (0.000 sec)

MariaDB [users]>
```

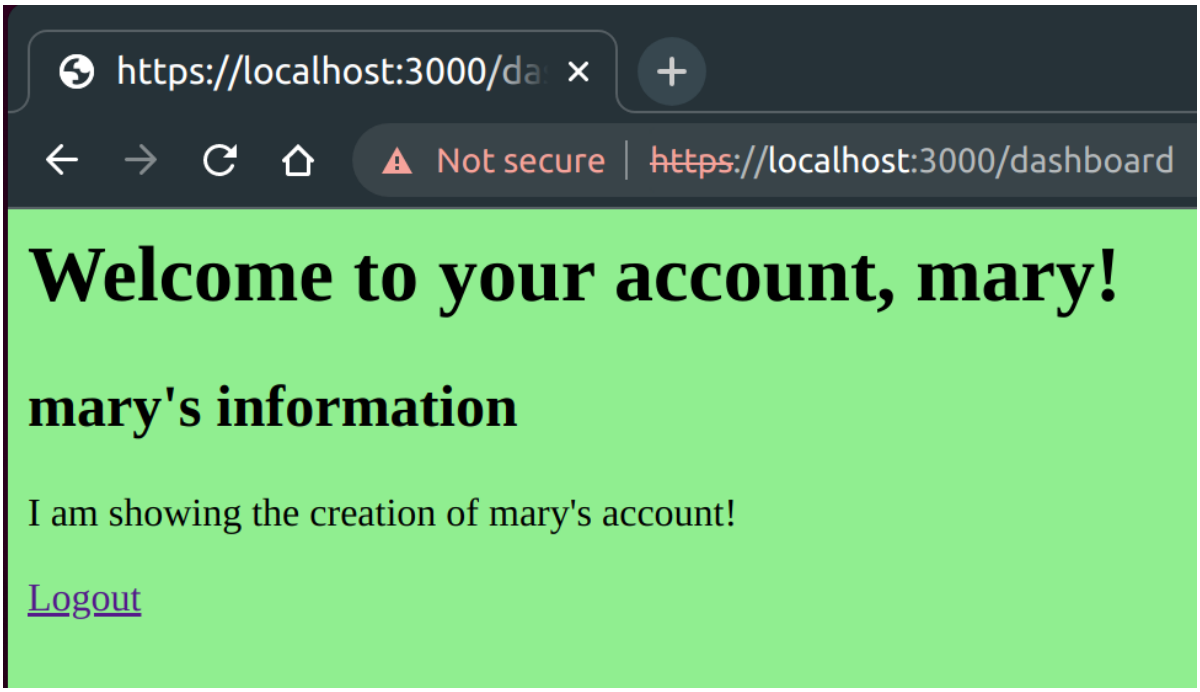
Press `Login` to test successful account creation:

Enter Credentials:

```
Username: mary
Password: Fp4$1ApaG3
```



Press `Submit`



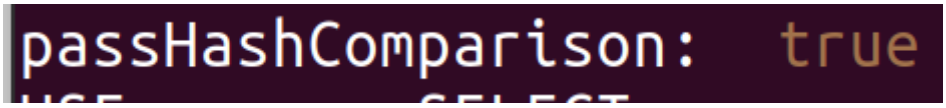
Step 5. Use the node.js's bcrypt package to securely store and verify passwords (in the SQL dabatabse). You can also find a very simple sample file here.

see the previous code snipping for storing passwords using bcrypt.

To verify, see `/login` POST endpoint. Below is a snippet showing the usage of bcrypt to verify passwords.

```
const hash = bcrypt.hashSync(password, saltRounds);
let passHashComparison = bcrypt.compareSync(password, hash);
console.log("passHashComparison: ", passHashComparison);

// use username comparison and passHashComparison to evaluate if match is found
```

Console messages when running `sessions.js`:

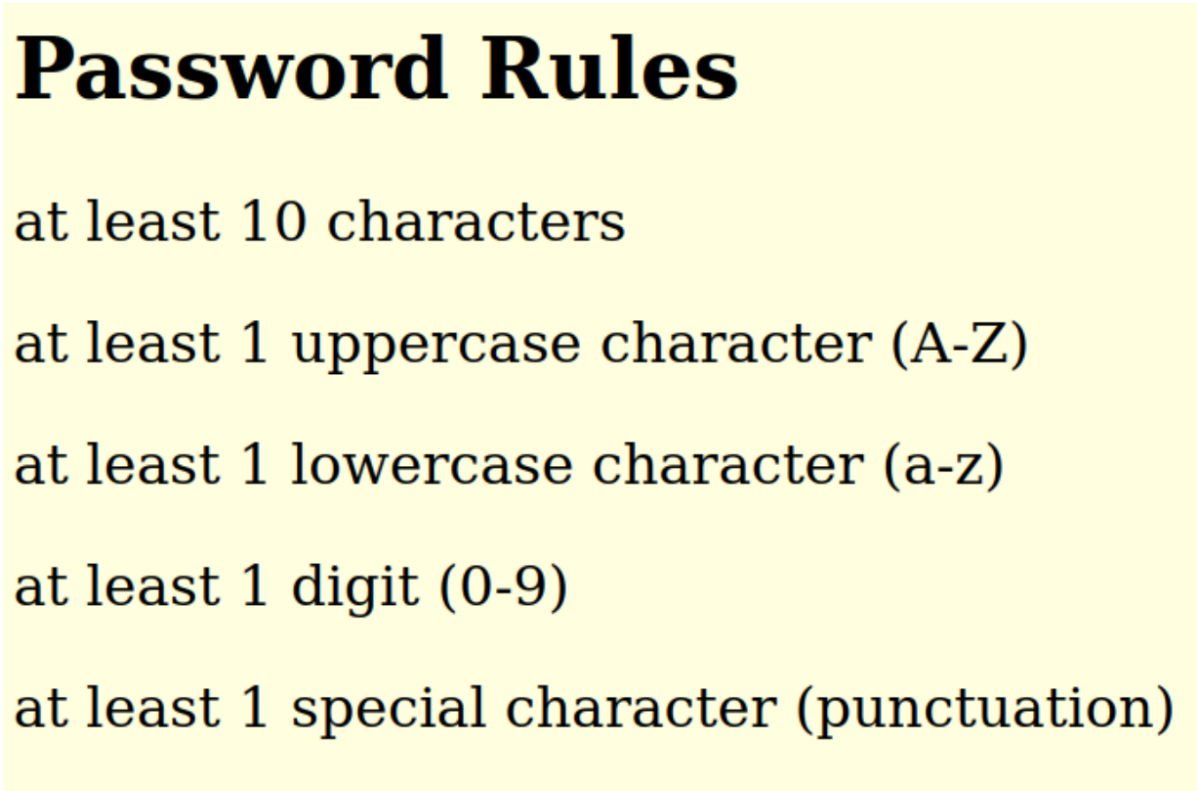


Step 6. Use node.js's password strength checker package to check whether the user's password is strong according to OWASP 10 requirements covered in class.

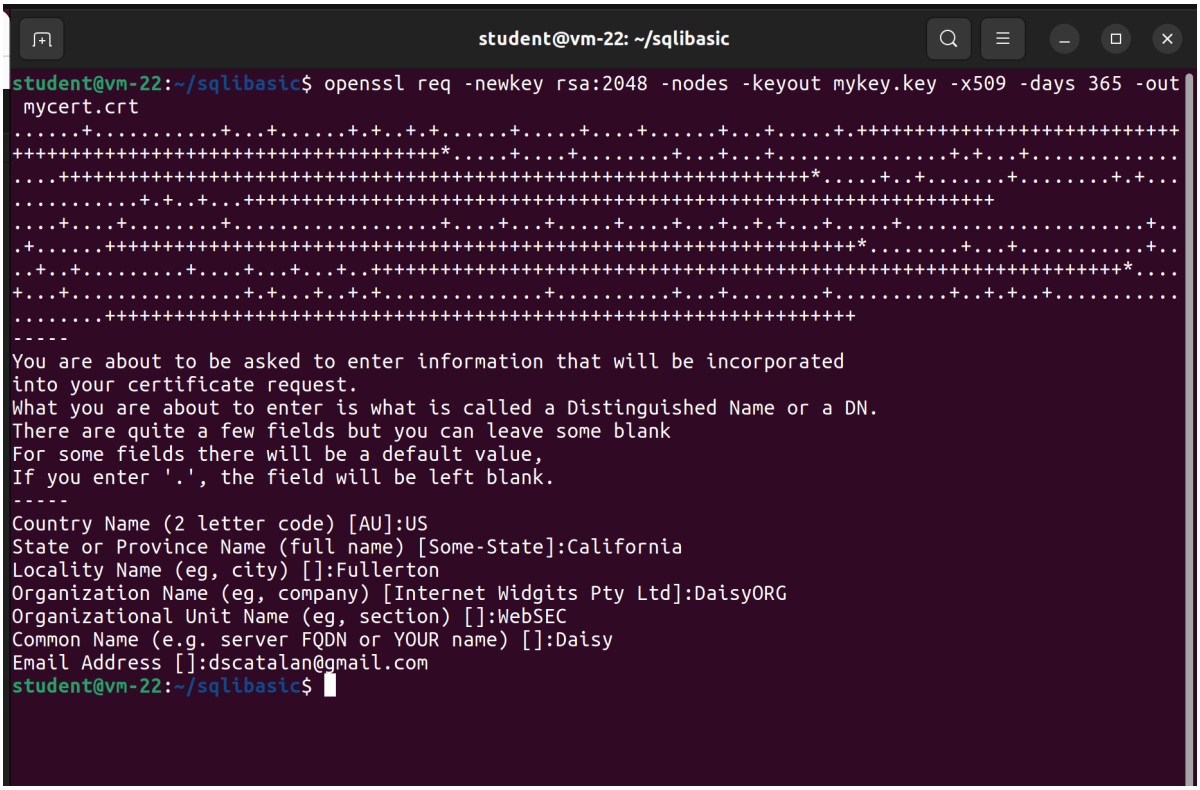
add middleware right before `/create-account` POST endpoint.

```
// middleware: check-password-strength
app.use("/create-account", (req, res, next) => {
    console.log("middleware: checking password strength...");
    console.log("Password:", passwordStrength(req.body.password).value);
    console.log("Password:", passwordStrength(req.body.password).length);
    console.log("Password:", passwordStrength(req.body.password).contains);

    if (passwordStrength(req.body.password).value === "Strong") {
        next();
    } else {
        res.send("Password Not Strong Enough! Please make sure to satisfy all
password rules!");
    }
});
```

Password rules are listed on `create-account-page.ejs`



Step 7. Add a self-signed HTTPs certificate.




```
cd sqlibasic/

openssl req -newkey rsa:2048 -nodes -keyout mykey.key -x509 -days 365 -out
mycert.crt
```

```
US
California
Fullerton
DaisyORG
WebSEC
Daisy
dscatalan@gmail.com
```

Step 8. Configure the client-sessions package to have the session expire after 10 mins inactivity (which the program already uses).

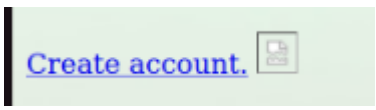
```
// The session settings middleware
app.use(sessions({
    cookieName: 'session',
    secret: 'random_string_goes_here',
    duration: 10 * 60 * 1000, // 10 mins
    activeDuration: 10 * 60 * 1000, // 10 mins
    cookie: {
        httpOnly: true // when true, cookie is not accessible from javascript
    }
}));
```

Step 9. Add CSP protection and make session cookies HTTPOnly to ensure some protection against XSS.

Add CSP protection

```
// csp middleware
app.use(
    contentSecurityPolicy({
        useDefaults: true,
        directives: {
            defaultSrc: ["'self'"],
            scriptSrc: ["'self'"],
            imgSrc: ["'self'", 'data:', 'http.cat'],
            objectSrc: ["'none'"],
            upgradeInsecureRequests: [],
        },
        reportOnly: false,
    })
);
```

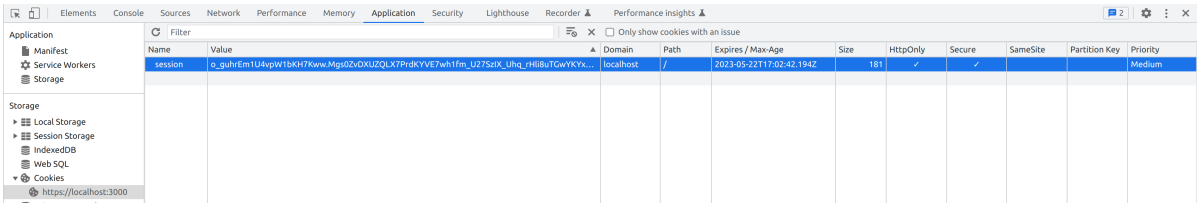
In the default page ("/") I have an image that should not load, and in create-account I have an image that should load.



This is because the image source is not allowed because I did not list it within the csp middleware.

I did list `http.cat` , so the cat image should work in the `successpage.ejs` page.

In Step 8, `httpOnly` is set to true when setting up the `client-sessions` cookie.



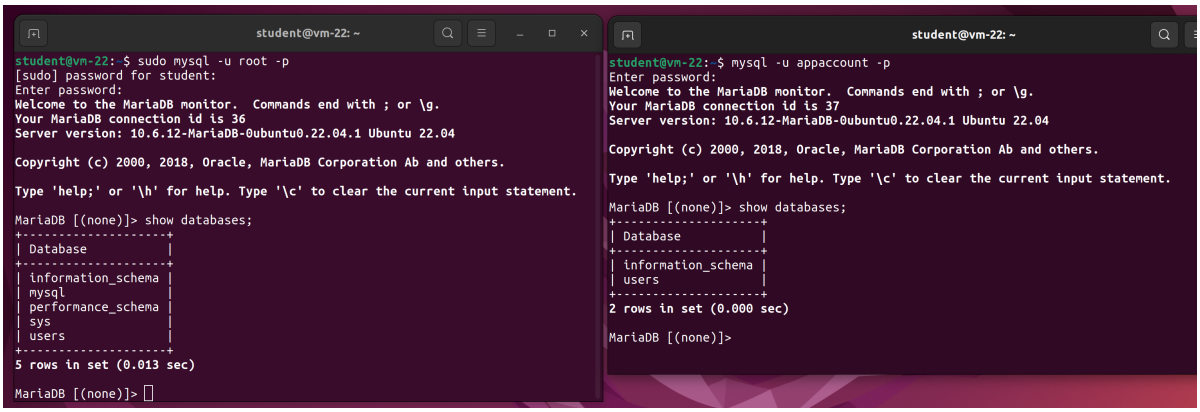
Step 10. Make sure that the webapp has a privilege-restricted database account.

See Step `13` from notes.

Code snippet from `sessions.js` showing use of privilege-restricted database account.

```
// Connect to the database
const mysqlConn = mysql.createConnection({
    host: "localhost",
    user: "appaccount",
    password: "apppass",
    multipleStatements: true

});
```

Notice the difference when using `root` vs `appaccount`


student@vm-22: ~
student@vm-22:~$ sudo mysql -u root -p
[sudo] password for student:
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
| Database |
| information_schema |
| mysql |
| performance_schema |
| sys |
| users |
5 rows in set (0.013 sec)
MariaDB [(none)]>
student@vm-22: ~
student@vm-22:~$ mysql -u appaccount -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
| Database |
| information_schema |
| users |
2 rows in set (0.000 sec)
MariaDB [(none)]>