

# Login

Tuesday, June 13, 2023 9:53 PM

## Step 1: create the html

```
JS users.js 2  <> practiceSearch.html  <> Loginpage.html X  JS app.js 2  public > <> Loginpage.html > html > body > form
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>login Page</title>
5      <link rel="stylesheet" href="/stylesheets/style.css">
6  </head>
7
8  <body>
9      <h1>Login Page</h1>
10     <p>Enter your login details here:</p>
11     <form action="myPage.html" method="get">
12         <label for="user">Enter Username</label>
13         <input type="text" id="user" value="Write Username here"/>
14
15         <label for="pass">Enter Password</label>
16         <input type="password" id="pass" value="Write Password here">
17
18         <button type="button">click to login</button>
19     </form>
20 </body>
21 </html>
```

Login html created.

## Login Page

Enter your login details here:

Enter Username  Enter Password

Step 2: Create the JSON of the input details and send them to the server using AJAX

In myJavascript.js

```
public > javascripts > JS myJavascript.js > myloginFunc > loginData > password
1  function myloginFunc() {
2      let loginData = {
3          username: document.getElementById('user').value,
4          password: document.getElementById('pass').value
5      };
6
7      let myxhttp = new XMLHttpRequest();
8
9      myxhttp.open('POST', '/login');
10     myxhttp.setRequestHeader('content-type', 'application/json');
11     myxhttp.send(JSON.stringify(loginData));
12 }
```

Step 3:

Lets set the server side as well so that it can handle the post to /login path

In index.js

```
8
9  router.post('/login', function(req, res, next){
10    console.log(JSON.stringify(req.body));
11    res.end();
12  });
13
14  module.exports = router;
```

Pushing button sends response to the server through AJAX client:

```
[nodemon] starting `node ./bin/www`
{"username":"Turjoasd","password":"asdasd"}
POST /login 200 845.759 ms - -
{"username":"Turjoasd","password":"asdasd"}
POST /login 200 3.112 ms - -
^C
```

Step 4: Now we need to store them so let's create a server session

```
root ➔ /workspaces/23S1_WDC_prac7-Turjo009 (main) $ npm install --save express-session
(.....) :: idealTree:23S1_WDC_prac7-Turjo009: sill idealTree buildDeps
```

In the app.js setup the server session as well. NOTE: keep the name of the variable as "sessions"

```
<> Loginpage.html    JS myJavascript.js 2    JS index.js 3    JS app.js 4, M X
JS app.js > ...
1  var express = require('express');
2  var path = require('path');
3  var cookieParser = require('cookie-parser');
4  var logger = require('morgan');
5  var session = require('express-session');
6  var indexRouter = require('./routes/index');
7  var usersRouter = require('./routes/users');
8
9  var app = express();
10
11  app.use(logger('dev'));
12  app.use(express.json());
13  app.use(express.urlencoded({ extended: false }));
14  app.use(cookieParser());
15
16  app.use(session({
17    resave: false,
18    saveUninitialized: true,
19    secret: 'super secret string of my choice',
20    secure: false
21  }));
22
23  //this is used to log the logins
24  app.use(function(req, res, next){
25    console.log("The current user is: " + req.session.username);
26    next();
27  });
28
29
30  app.use(express.static(path.join(__dirname, 'public')));
31
32  app.use('/', indexRouter);
33  app.use('/users', usersRouter);
34
35
36  app.listen(3000, function() {
```

A slight modification in the server index.js is made so that the username is available in the app.js:

```
router.post('/login', function(req, res, next){
  req.session.username = req.body.username;
  console.log(req.body.username);
  res.end();
});
```

DONE:

NOW LETS CHECK USERS FROM STORED DATA

In index.js:

Create an object user, and set and if and else statement to

verify if the input matches the data of the user.

```
39
40 let users = {
41   Ami: {password: 'mypass1'},
42   Gandu: {password: 'mypass2'}
43 };
44
45 router.post('/login', function(req, res, next){
46
47   if(req.body.username in users){
48     if(req.body.password === users[req.body.username].password){
49       req.session.username = req.body.username;
50       console.log("you are loggin in, Congrats");
51       res.end();
52     }
53     else{
54       res.sendStatus(401);
55       console.log("Wrong password, sad");
56     }
57   }
58   else{
59     res.sendStatus(401);
60     console.log("User Not Found, sad");
61   }
62 }
63 });
64
65 module.exports = router;
```

DONE

Suppose you want to make a post only when a user is logged in:

```
34
35 router.post('/newpost', function(req,res,next){
36
37   if (!('username' in req.session)){
38     res.sendStatus(403);
39     return;
40   }
41
42   // Receive data from the client; we'll look more at post requests later this semester
43   let post = req.body;
44   /*
```

So if a username attribute is created for the session object then the server will allow to post new content.