

Tyler Baxter - 924372420 - Github: agge3
Kat Powell - 924373005 - Github: kpowkitty
Ian Kligman - 924122742 - Github: ikligman AND Faten848
Team Blinux
Professor Scott
4 May 2025

Group Project - Milestone 3

Description:

Website with Javascript and database.

Approach:

We decided that we are going to use Docker, as it would be a convenient development environment for database management on linux. The first step was configuring docker in the project. Once this was set up, we could start interfacing with MariaDB, which is a fork of mysql. We set up dependencies and npm packages: mysql, bcrypt, express-session. Setting up express-session involved generating a unique session-secret that will be included in .gitignore to maintain security. Next, we incorporated our javascript routes for creating an account, and for logging in, while using database calls to cache new accounts and verify existing email-password pairs in the database for authentication. This javascript also incorporated our use of bcrypt, which hashes the passwords before storing them in the database, and makes use of bcrypt.compare() to verify a cleartext password to its hashed equivalent. We did not make use of salt, but did opt for several hashing iterations, in our case 10. A final crucial step was to configure the appropriate routes in our pre-existing login.ejs and create.ejs.

Issues and Resolutions:

There was a slight confusion regarding naming convention between the javascript file and the ejs files. Initially the route name for creating an account was "create". Somewhere along the line, I got confused and was trying to add a route named "register" to the .ejs files, while in fact the route in server317.js is named "create". This resulted in errors that did not immediately indicate that the names mismatched, and thus we were sent down a debugging rabbit hole that ended up being unnecessary.

Screenshots of some code:

Tyler Baxter - 924372420 - Github: agge3

Kat Powell - 924373005 - Github: kpowkitty

Ian Kligman - 924122742 - Github: ikligman AND Faten848

Team Blinux

Professor Scott

4 May 2025

```
10 const bcrypt = require('bcrypt');
11 const session = require('express-session');
12
13 // USER AUTH vvvv
14
15 const db = mysql.createPool({
16   host: 'localhost',
17   user: 'root',
18   password: '',
19   database: 'authdb'
20 });
21
22 app.use(session({
23   secret: process.env.SESSION_SECRET,
24   resave: false,
25   saveUninitialized: false
26 }));
27
28 // Route for creating a new account
29 app.post('/create', express.urlencoded({ extended: true }), async (req, res) => {
30   const { email, password } = req.body;
31
32   try {
33     // Check if the email already exists
34     const [existingUser] = await db.query('SELECT * FROM users WHERE email = ?', [email]);
35     if (existingUser.length > 0) {
36       return res.status(400).send('Email already exists');
37     }
38   }
39
40   // Save user to database
41   const [rows] = await db.query('INSERT INTO users (email, password) VALUES (?, ?)');
42   const user = rows[0];
43
44   // Save user ID in session
45   req.session.userId = user.id;
46   res.redirect('/profile'); // Redirect to the profile page after successful login
47 }
48
49 // Route for logging in
50 app.post('/login', express.urlencoded({ extended: true }), async (req, res) => {
51   const { email, password } = req.body;
52
53   try {
54     // Fetch the user from the database
55     const [rows] = await db.query('SELECT * FROM users WHERE email = ?', [email]);
56     const user = rows[0];
57
58     if (user && await bcrypt.compare(password, user.password)) {
59       // Save user ID in session
60       req.session.userId = user.id;
61       res.redirect('/profile'); // Redirect to the profile page after successful login
62     } else {
63       res.status(401).send('Invalid email or password');
64     }
65   } catch (err) {
66     console.error(err);
67     res.status(500).send('Error logging in');
68   }
69 }
70
71 // USER AUTH ^^^^
72
73 // Ensure profile page is limited to authenticated users
74 function requireAuth(req, res, next) {
75   if (!req.session.userId) {
76     res.status(401).send('Unauthorized');
77   } else {
78     next();
79   }
80 }
```

Tyler Baxter - 924372420 - Github: agge3

Kat Powell - 924373005 - Github: kpowkitty

Ian Kligman - 924122742 - Github: ikligman AND Faten848

Team Blinux

Professor Scott

4 May 2025

```
25     saveUninitialized: false
26   }));
27
28   // Route for creating a new account
29   app.post('/create', express.urlencoded({ extended: true }), async (req, res) => {
30     const { email, password } = req.body;
31
32     try {
33       // Check if the email already exists
34       const [existingUser] = await db.query('SELECT * FROM users WHERE email = ?', [email]);
35       if (existingUser.length > 0) {
36         return res.status(400).send('Email already exists');
37       }
38
39       // Hash the password
40       const hashedPassword = await bcrypt.hash(password, 10);
41
42       // Insert the new user into the database
43       await db.query('INSERT INTO users (email, password) VALUES (?, ?)', [email, hashedPassword]);
44
45       res.redirect('/login'); // Redirect to the login page after successful registration
46     } catch (err) {
47       console.error(err);
48       res.redirect('/create'); // Redirect to the register page if there is an error
49     }
50   });
51
52   // Route for logging in
```

Tyler Baxter - 924372420 - Github: agge3

Kat Powell - 924373005 - Github: kpowkitty

Ian Kligman - 924122742 - Github: ikligman AND Faten848

Team Blinux

Professor Scott

4 May 2025

```
11 <body>
12   <%- include('partials/info.ejs') %>
13   <%- include('partials/home.ejs') %>
14   <div class="container">
39     </div>
40   </form>
41 </div>
42
43 <script>
44   function validateForm(event) {
45     const password = document.getElementById('password').value;
46     const confirmPassword = document.getElementById('confirmPassword');
47
48     if (password !== confirmPassword) {
49       alert('Passwords do not match!');
50       return false; // Prevent form submission
51     }
52
53     if (password.length < 8) {
54       alert('Password must be at least 8 characters long!');
55       return false;
56     }
57
58     return true; // Allow form submission
59   }
60 </script>
61 </body>
62 </html>
```

Tyler Baxter - 924372420 - Github: agge3

Kat Powell - 924373005 - Github: kpowkitty

Ian Kligman - 924122742 - Github: ikligman AND Faten848

Team Blinux

Professor Scott

4 May 2025

```
10 </head>
11 <body>
12   <%- include('partials/info.ejs') %>
13   <%- include('partials/home.ejs') %>
14   <div class="container">
15     <h1 class="logo">Aggregate</h1>
16     <p class="tagline">Create your account</p>
17
18     <form action="/create" method="POST" onsubmit="return validateForm(ev
19       <div class="form-group">
20         <label for="email">Email</label>
21         <input type="email" id="email" name="email" required>
22       </div>
23
24       <div class="form-group">
25         <label for="password">Password</label>
26         <input type="password" id="password" name="password" required>
27         <p class="password-requirements">Must be at least 8 characters</p>
28       </div>
29
30       <div class="form-group">
31         <label for="confirmPassword">Confirm Password</label>
32         <input type="password" id="confirmPassword" name="confirmPassword
33       </div>
34
35       <button type="submit" class="createButton">Create account</button>
36
37       <div class="login-link">
```

Tyler Baxter - 924372420 - Github: agge3

Kat Powell - 924373005 - Github: kpowkitty

Ian Kligman - 924122742 - Github: ikligman AND Faten848

Team Blinux

Professor Scott

4 May 2025

```
11 <body>
12   <%- include('partials/info.ejs') %>
13   <%- include('partials/home.ejs') %>
14   <div class="container">
15     <h1 class="logo">Aggregate</h1>
16     <p class="tagline">Access your dashboard</p>
17
18     <form action="/login" method="POST">
19       <div class="form-group">
20         <label for="email">Email</label>
21         <input type="email" id="email" name="email" required>
22       </div>
23
24       <div class="form-group">
25         <label for="password">Password</label>
26         <input type="password" id="password" name="password" required>
27       </div>
28
29       <button type="submit" class="login-button">Log in</button>
30     </form>
31
32     <div class="forgot-password">
33       <a href="#">Forgot password?</a>
34     </div>
35
36     <div class="signup-link">
37       Don't have an account? <a href="/create">Create account</a>
38     </div>
```