

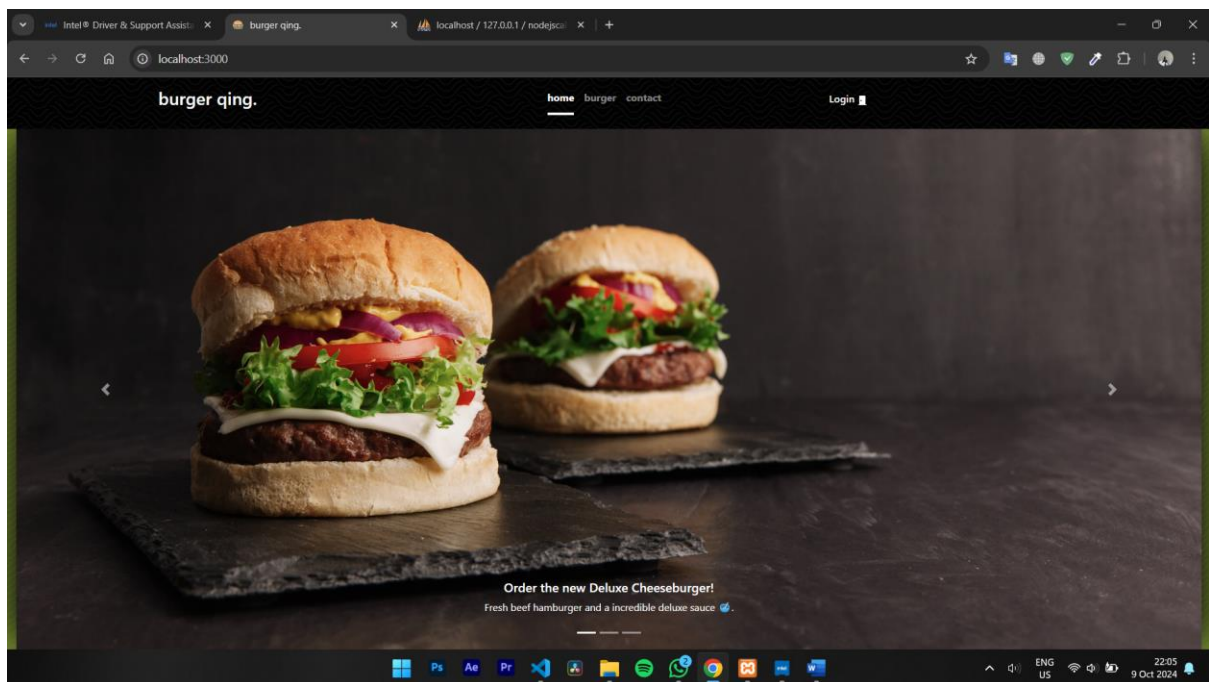
Devanti Ryo Asgawa

21060122140100

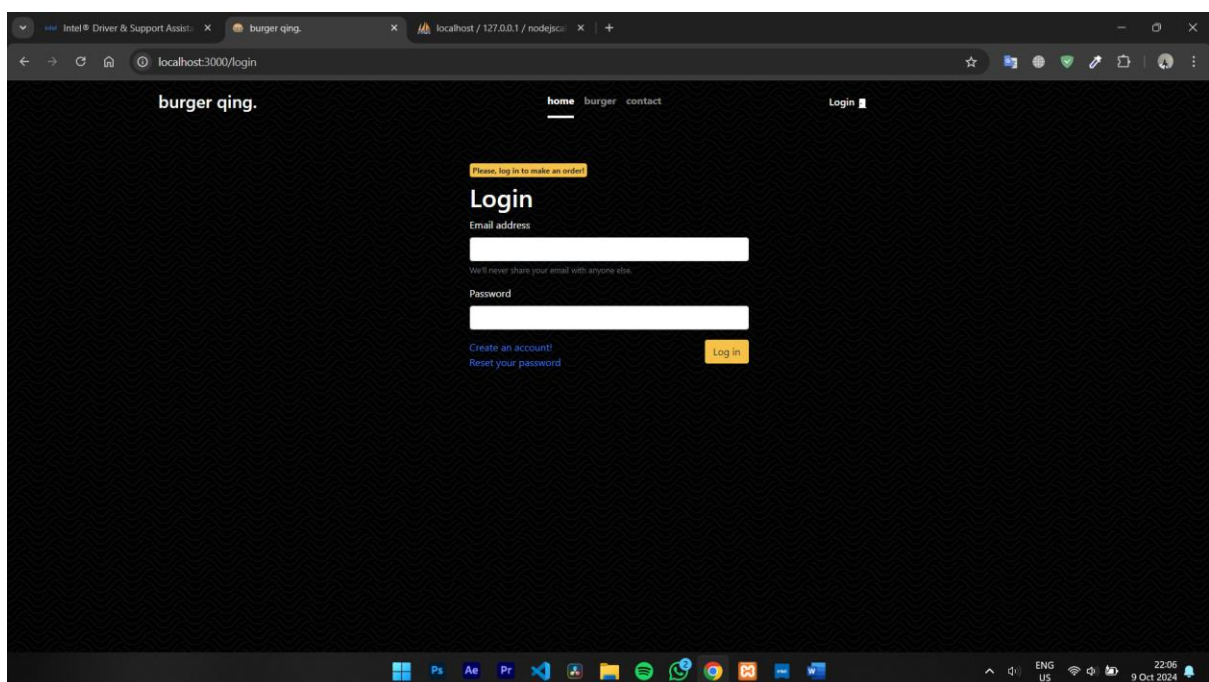
Pengembangan Web – A

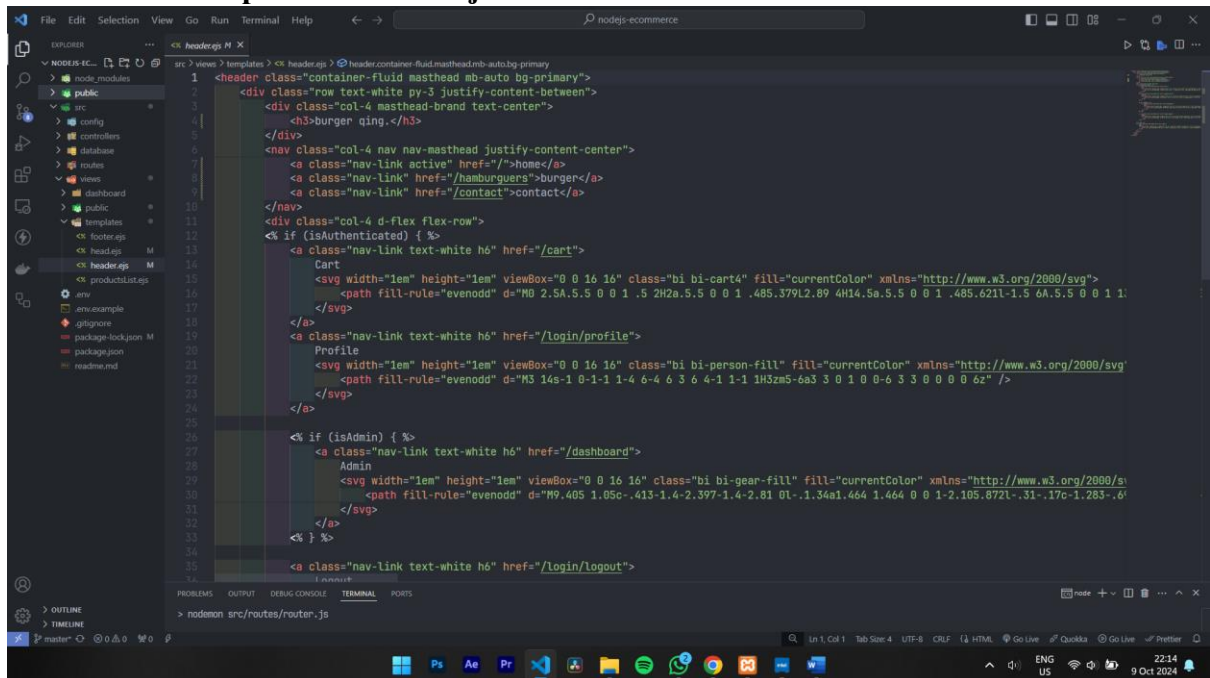
Ujian Tengah Semester

Tampilan depan



Tampilan login account oleh user





```
src > routes > login.js
```

The image shows a VS Code editor window with a Node.js application. The file explorer on the left shows the project structure, including folders like 'node_modules', 'src', and 'routes'. The main editor displays the 'routes/router.js' file, which contains the following code:

```
1 // express initialization
2 const express = require('express');
3 const router = express.Router();
4
5 // config and controller
6 require('dotenv').config();
7 const config = require('../config/app-config.js');
8 const UserController = require('../controllers/users.js');
9 const User = new UserController();
10
11 // required libraries
12 const bcrypt = require('bcrypt');
13 const session = require('express-session');
14 const MySQLStore = require('express-mysql-session');
15 const sessionStore = new MySQLStore(config.dbName);
16 const bodyParser = require('body-parser');
17 const passport = require('passport');
18 const passportLocalStrategy = require('passport-local');
19 const LocalStrategy = require('passport-local').Strategy;
20 const cookieParser = require('cookie-parser');
21 const csrf = require('csurf');
22
23 // global middleware
24 router.use(session({
25   name: process.env.SESSION_NAME,
26   key: process.env.SESSION_KEY,
27   secret: process.env.SESSION_SECRET,
28   store: sessionStore,
29   resave: false,
30   saveUninitialized: false
31 }));
32
33 router.use(passport.initialize());
34 router.use(passport.session());
35
36 router.use(async function(req, res, next) {
37   res.locals.isAuthenticated = req.isAuthenticated();
38
39   try {
40     res.locals.isAdmin = await User.isAdmin(req.session.passport.user);
41   } catch {
42     res.locals.isAdmin = false;
43   }
44 });
```

Views > Publik > login.ejs

```

src > views > public > <- login.ejs ->
1 <!DOCTYPE html>
2 <html lang="en">
3   <<- include(../templates/head.ejs) ->>
4
5   <body class="bg-primary">
6     <<- include(../templates/header.ejs) ->>
7     <div class="container mt-5">
8       <div class="row">
9         <div class="col"></div>
10        <div class="col-5 text-white">
11          <div class="badge badge-warning text-wrap d-inline">Please, log in to make an order</div>
12          <div class="mt-2">Login</div>
13          <% if (messages.error) { %>
14            <div class="badge badge-danger my-3"><%= messages.error %></div>
15          <% %>
16          <form method="POST" action="/login">
17            <input type="hidden" name="_csrf" value="<%= csrfToken %>">
18            <div class="form-group">
19              <label for="exampleInputEmail">Email address</label>
20              <input type="email" class="form-control" id="username" name="username">
21              <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
22            </div>
23            <div class="form-group">
24              <label for="exampleInputPassword">Password</label>
25              <input type="password" class="form-control" id="password" name="password">
26            </div>
27            <button type="submit" class="btn btn-warning float-right">Log In</button>
28          </form>
29          <a class="d-block wret=/login/register">Create an account</a>
30          <a class="d-block wret=/login/reset">Reset your password</a>
31        </div>
32      </div class="col"></div>
33    </div>
34  </body>
35 </html>

```

Kode HTML

Kode HTML yang ditampilkan adalah bagian dari sebuah halaman login yang menggunakan **EJS (Embedded JavaScript)** sebagai template engine untuk Node.js. Berikut penjelasan dari kode tersebut:

Bagian 1: Deklarasi DOCTYPE dan Tag HTML

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
```

- `<!DOCTYPE html>`: Menandakan bahwa dokumen ini adalah dokumen HTML5.
- `<html lang="en">`: Tag pembuka elemen HTML yang menunjukkan bahwa bahasa yang digunakan dalam halaman ini adalah bahasa Inggris (`lang="en"`).

Bagian 2: Penyertaan Template Eksternal

```
html
Copy code
<%- include('../templates/head.ejs') -%>
```

- Baris ini menggunakan sintaks EJS untuk menyertakan file `head.ejs`. File ini mungkin berisi elemen `<head>` seperti `title`, link ke CSS, atau metadata lainnya, sehingga tidak perlu ditulis ulang di setiap halaman.

Bagian 3: Elemen Body dan Header

```
html
Copy code
<body class="bg-primary">
  <%- include('../templates/header.ejs') -%>
```

- `body class="bg-primary"`: Elemen `<body>` dimulai dan diberi kelas `bg-primary`, yang mungkin merujuk pada class CSS untuk memberi latar belakang berwarna biru (sesuai dengan framework Bootstrap).
- `include('../templates/header.ejs')`: Menyertakan template `header.ejs`, yang mungkin berisi bagian header navigasi atau elemen-elemen umum di bagian atas halaman.

Bagian 4: Layout Formulir Login

```
html
Copy code
<div class="container mt-5">
  <div class="row">
    <div class="col"></div>
    <div class="col-5 text-white">
      <div class="badge badge-warning text-wrap d-inline">Please, log
in to make an order!</div>
      <h1 class="mt-2">Login</h1>
      <% if (messages.error) { %>
```

```

        <div class="badge badge-danger my-3"><%= messages.error
%></div>
    <% } %>

```

- `container mt-5`: Kelas Bootstrap untuk memberikan margin atas (spacing) yang lebih besar.
- `row`: Membuat sebuah baris Bootstrap untuk mengatur layout dengan grid.
- `col` dan `col-5`: Mengatur kolom dengan grid, `col-5` mengambil 5/12 bagian dari lebar kontainer.
- `text-white`: Memberikan teks berwarna putih (Bootstrap).
- `<% if (messages.error) { %>`: Ini adalah sintaks EJS untuk kondisi, di mana jika terdapat pesan error dalam variabel `messages.error`, maka pesan tersebut akan ditampilkan dalam bentuk badge merah.
- `<%= messages.error %>`: Menampilkan pesan error yang diteruskan dari server jika otentikasi gagal.

Bagian 5: Form Login

```

html
Copy code
<form method="POST" action="/login">
  <input type="hidden" name="_csrf" value="<%= csrfToken %>">
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="username"
name="username">
    <small id="emailHelp" class="form-text text-muted">We'll never
share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="password"
name="password">
  </div>
  <button type="submit" class="btn btn-warning float-right">Log
in</button>
</form>

```

- `<form method="POST" action="/login">`: Form ini menggunakan metode POST untuk mengirim data ke rute `/login`.
- `<input type="hidden" name="_csrf" value="<%= csrfToken %>">`: Ini adalah token CSRF (Cross-Site Request Forgery) untuk mencegah serangan CSRF, dihasilkan oleh server dan disisipkan di form.
- Bagian email dan password diatur dalam `form-group` yang merupakan elemen Bootstrap untuk memformat input form.
- Tombol "Log in" diberi kelas `btn btn-warning` yang memberi gaya warna kuning pada tombol dengan posisi di sebelah kanan (`float-right`).

Bagian 6: Link Tambahan

```

html
Copy code
<a class="d-block" href="/login/register">Create an account!</a>
<a class="d-block" href="/login/reset">Reset your password</a>

```

- `d-block`: Ini adalah class Bootstrap untuk membuat link ditampilkan sebagai blok (sehingga setiap link berada di baris terpisah).
- Link pertama mengarah ke halaman pendaftaran (`/login/register`) untuk membuat akun baru.
- Link kedua mengarah ke halaman reset password (`/login/reset`) untuk mereset kata sandi jika pengguna lupa.

Bagian 7: Penutup HTML

```
html
Copy code
</div>
<div class="col"></div>
</div>
</div>
</body>

</html>
```

- Mengakhiri layout dengan menutup tag `div`, `body`, dan `html`. Bagian ini menyelesaikan struktur grid layout dan kontainer Bootstrap.

Kesimpulan

- Kode ini merupakan halaman login yang menggunakan template engine EJS untuk rendering dinamis di server-side. Desainnya memanfaatkan framework **Bootstrap** untuk pengaturan layout dan gaya visual.
- Elemen penting seperti proteksi CSRF dan pesan error juga diimplementasikan, serta terdapat link untuk mendaftar dan mereset kata sandi jika pengguna membutuhkan.

Penjelasan dari framework yang terlibat:

Kode yang Anda berikan menggunakan kombinasi dari berbagai teknologi dan framework, dengan fokus utama pada **Node.js** dan **Express.js**, serta beberapa alat lain untuk templating dan tata letak antarmuka. Berikut adalah penjelasan dari framework yang terlibat:

1. **Node.js**: Ini adalah lingkungan runtime yang menjalankan fungsi server-side dari aplikasi. Node.js memungkinkan JavaScript untuk dijalankan di sisi server.
2. **Express.js**: Ini adalah framework backend utama yang digunakan untuk menangani routing, logika server, dan middleware di dalam aplikasi ini. Aksi POST dari form, perlindungan token CSRF, dan rendering sisi server dari tampilan dikelola oleh Express.
3. **EJS (Embedded JavaScript Templates)**: EJS adalah mesin templating yang digunakan untuk secara dinamis menghasilkan konten HTML. Sintaks seperti `<%-include('../templates/head.ejs') -%>` dan `<%= csrfToken %>` menunjukkan bahwa EJS digunakan untuk menyuntikkan variabel server-side dan konten ke dalam HTML sebelum dikirim ke browser.
4. **Bootstrap**: Kelas seperti container, form-group, form-control, btn btn-warning, dan badge merupakan bagian dari framework CSS **Bootstrap**. Bootstrap digunakan di sini

untuk memberikan gaya pada halaman, membuatnya responsif, dan menciptakan tata letak yang terstruktur.

5. **Perlindungan CSRF (Cross-Site Request Forgery):** Kehadiran token CSRF (`<input type="hidden" name="_csrf" value="<%= csrfToken %>">`) menunjukkan penggunaan middleware untuk perlindungan CSRF. Dalam Express, ini biasanya diterapkan menggunakan middleware `csrf` untuk mengamankan form dari serangan CSRF.

Kesimpulan:

Kode ini merupakan bagian dari aplikasi web **Node.js** dan **Express.js** yang menggunakan **EJS** untuk templating dan **Bootstrap** untuk styling. Selain itu, kode ini juga mencakup fitur keamanan seperti perlindungan **CSRF**. Framework-framework ini bekerja sama untuk membangun aplikasi web yang aman, responsif, dan dinamis.