

# **ULANGAN TENGAH SEMESTER**

**Mata Kuliah PBO & PEMWEB**



**UTS**

Dosen Pengampu:  
Willdan Aprizal Aripin, S.Pd., M.Kom.

Disusun Oleh:  
Nama: Putri Wahyuni  
NIM: 2310271  
Kelas: SIK 3A

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN  
UNIVERSITAS PENDIDIKAN INDONESIA**

**2024**

**1. Link GitHub:**

[https://github.com/putriwhyn29/UTS\\_PBO\\_PEMWEB\\_putriwahyun\\_i.git](https://github.com/putriwhyn29/UTS_PBO_PEMWEB_putriwahyun_i.git)

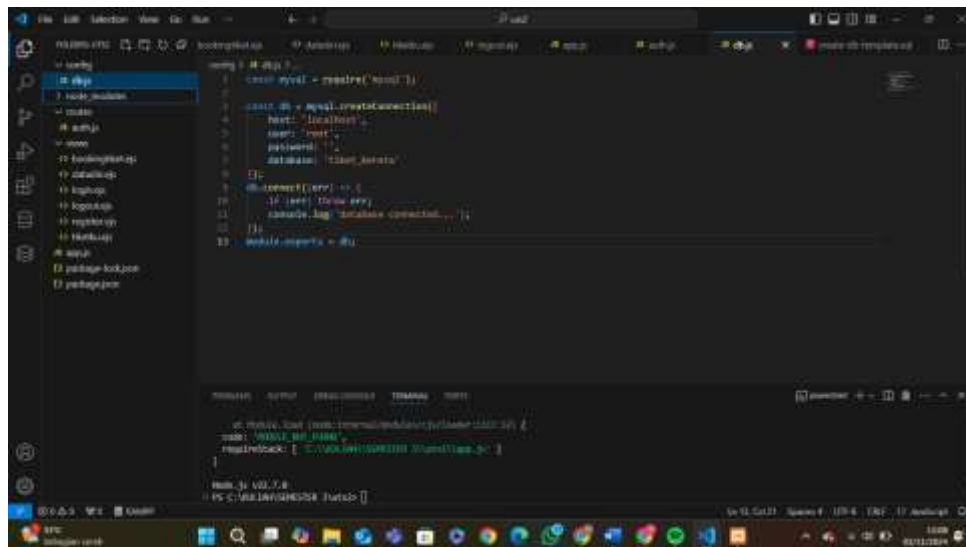
**2. Link Video Presentasi web:**

[https://drive.google.com/drive/folders/1GV\\_vjzJyoDPOAgDXBiA1T1ivT9nKIImk?usp=sharing](https://drive.google.com/drive/folders/1GV_vjzJyoDPOAgDXBiA1T1ivT9nKIImk?usp=sharing)

**3. Dokumentasi Kode dan Panduan Penggunaan**

- **DOKUMENTASI KODE:**

➤ db.js



```
const mysql = require('mysql');

const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'uts_pbo'
});

db.connect((err) => {
  if (err) throw err;
  console.log('Database connected...');
});

module.exports = db;
```

```
const db = require('mysql');

const db = require('mysql');

const db = require('mysql');
```

## ➤ routes/auth.js

```
routes.get('/register', (req, res) => {
  const { username, email, password } = req.body;

  const query = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
  db.query(query, [username, email, password], (err, result) => {
    if (err) {
      console.error(err);
      return res.status(500).send("Server error occurred.");
    }
    res.redirect('/auth/login');
  });
});

routes.get('/login', (req, res) => {
  res.render('login');
});
```

```
routes.get('/login', (req, res) => {
  const { username, password } = req.body;

  const query = "SELECT * FROM users WHERE username = ?";
  db.query(query, [username], (err, result) => {
    if (err) {
      console.error(err);
      return res.status(500).send("Server error occurred.");
    }

    if (result.length > 0) {
      const user = result[0];

      if (bcrypt.compareSync(password, user.password)) {
        req.session.user = user;
        res.redirect('/auth/profile');
      } else {
        res.send("Password does not match");
      }
    } else {
      res.send("Username does not exist");
    }
  });
});

routes.get('/profile', (req, res) => {
  if (req.session.user) {
    res.render('profile');
  } else {
    res.redirect('/auth/login');
  }
});
```

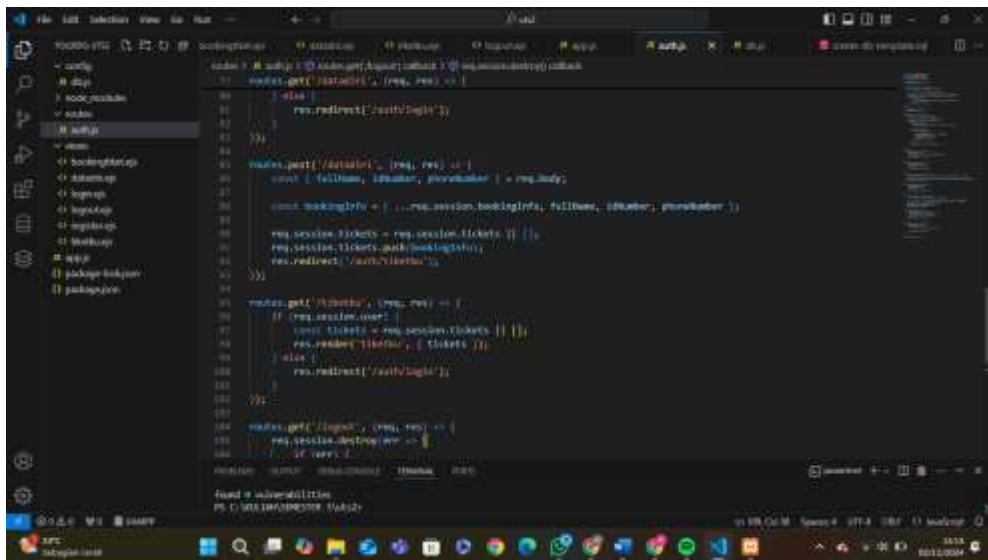
```
routes.get('/profile', (req, res) => {
  if (req.session.user) {
    res.render('profile', { user: req.session.user });
  } else {
    res.redirect('/auth/login');
  }
});

routes.get('/booking', (req, res) => {
  if (req.session.user) {
    res.render('booking');
  } else {
    res.redirect('/auth/login');
  }
});

routes.post('/booking', (req, res) => {
  const { departureStation, arrivalStation, departureDate, passengerCount } = req.body;

  req.session.bookingInfo = { departureStation, arrivalStation, departureDate, passengerCount };
  res.redirect('/auth/track');
});

routes.get('/track', (req, res) => {
  if (req.session.user) {
    res.render('track');
  } else {
    res.redirect('/auth/login');
  }
});
```



```
routes.get('/auth/login', (req, res) => {
  const { fullName, idNumber, password } = req.body;

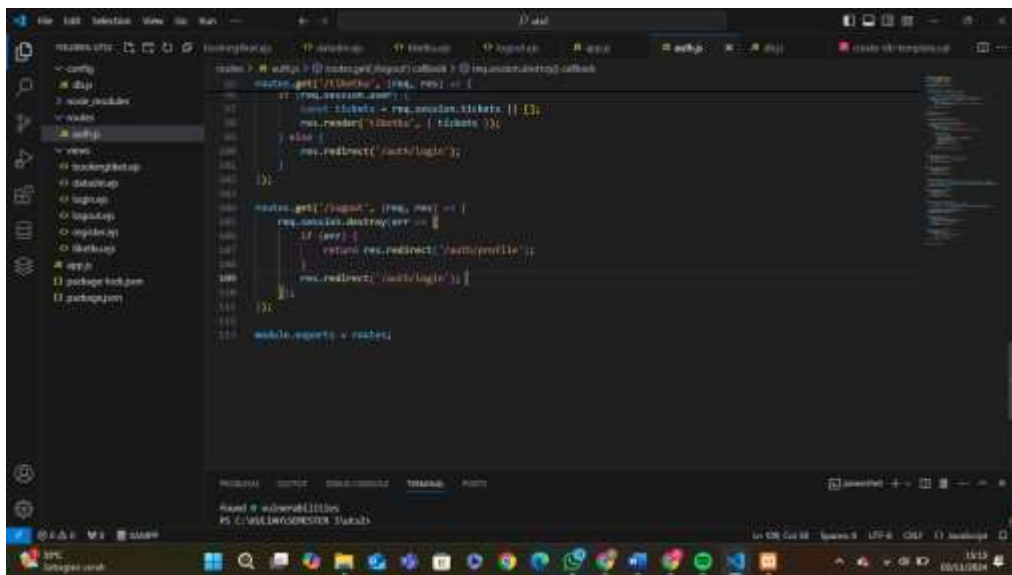
  const bookingInfo = { ...req.session.bookingInfo, fullName, idNumber, password };

  req.session.tickets = req.session.tickets || [];
  req.session.tickets.push(bookingInfo);
  res.redirect('/auth/login');
});

routes.get('/auth/register', (req, res) => {
  if (req.session.user) {
    const tickets = req.session.tickets || [];
    res.render('tickets', { tickets });
  } else {
    res.redirect('/auth/login');
  }
});

routes.get('/auth/logout', (req, res) => {
  req.session.destroy(err => {
    if (err) {
      return res.redirect('/auth/profile');
    }
    res.redirect('/auth/login');
  });
});

module.exports = routes;
```



```
routes.get('/auth/login', (req, res) => {
  const { fullName, idNumber, password } = req.body;

  const bookingInfo = { ...req.session.bookingInfo, fullName, idNumber, password };

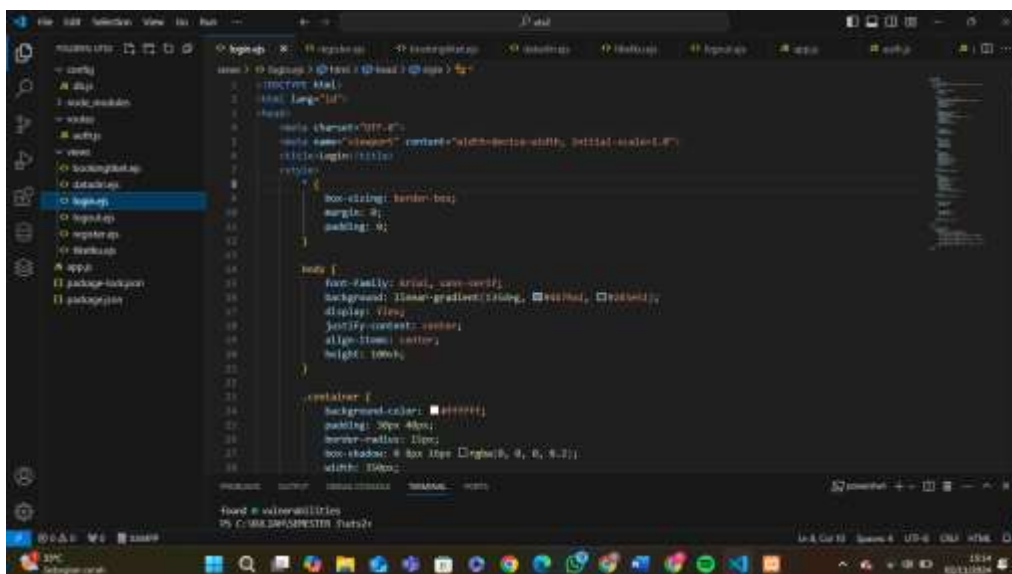
  req.session.tickets = req.session.tickets || [];
  req.session.tickets.push(bookingInfo);
  res.redirect('/auth/login');
});

routes.get('/auth/register', (req, res) => {
  if (req.session.user) {
    const tickets = req.session.tickets || [];
    res.render('tickets', { tickets });
  } else {
    res.redirect('/auth/login');
  }
});

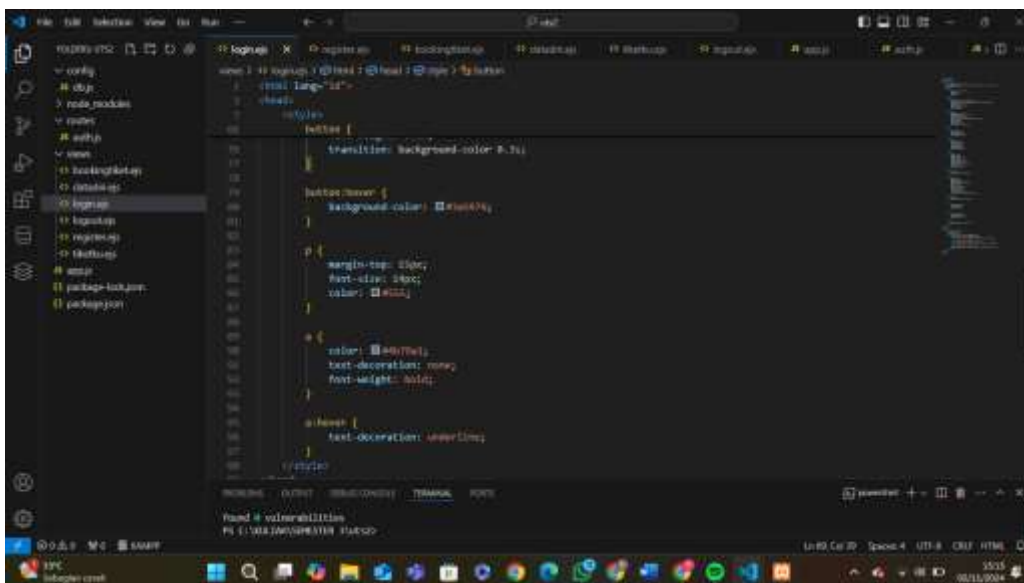
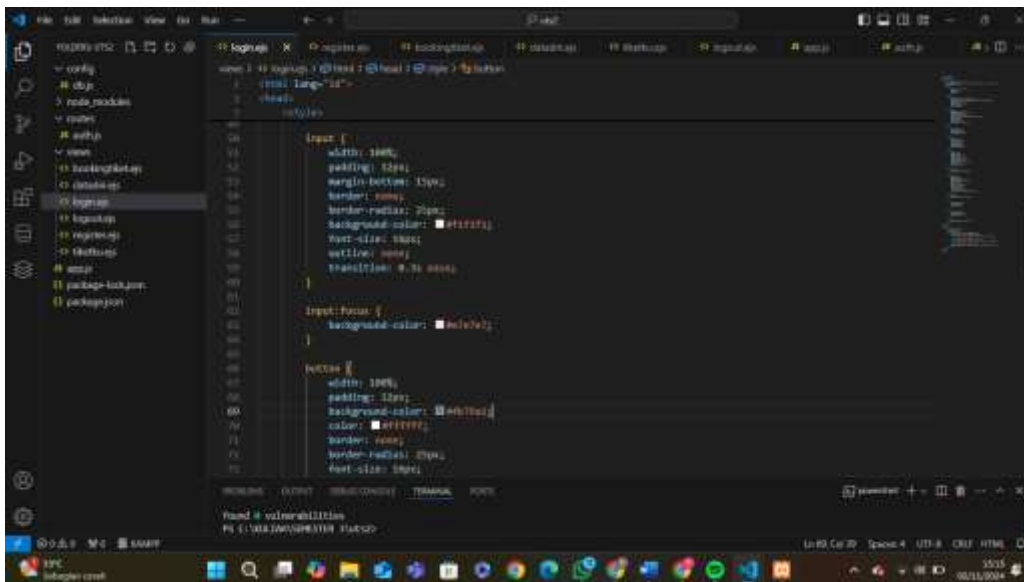
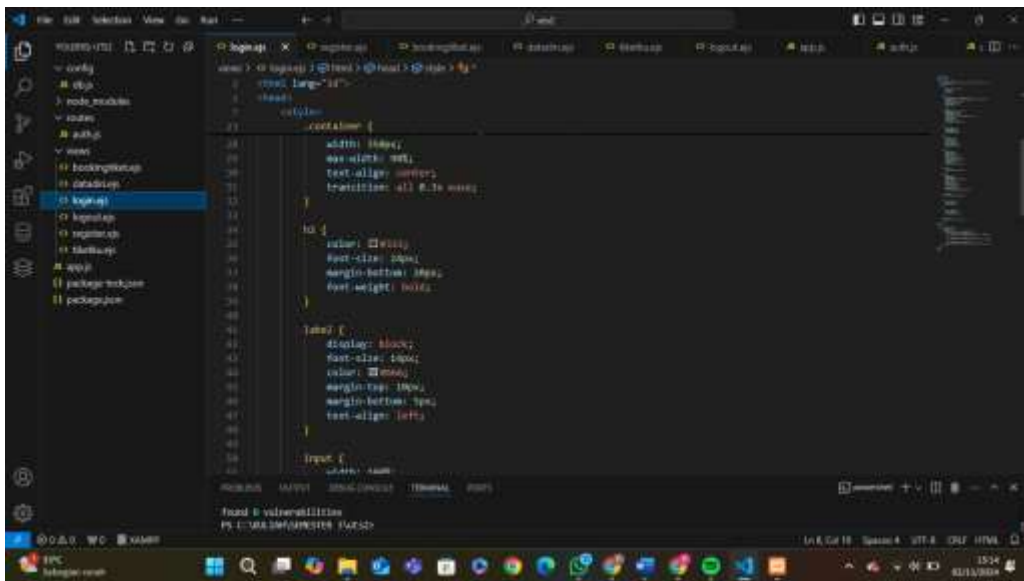
routes.get('/auth/logout', (req, res) => {
  req.session.destroy(err => {
    if (err) {
      return res.redirect('/auth/profile');
    }
    res.redirect('/auth/login');
  });
});

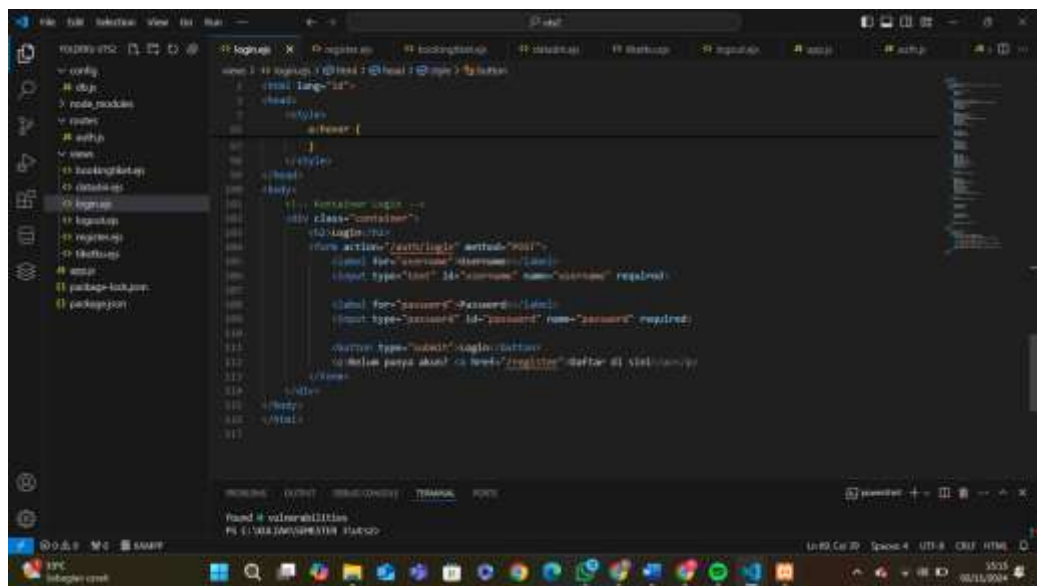
module.exports = routes;
```

➤ views/login.ejs

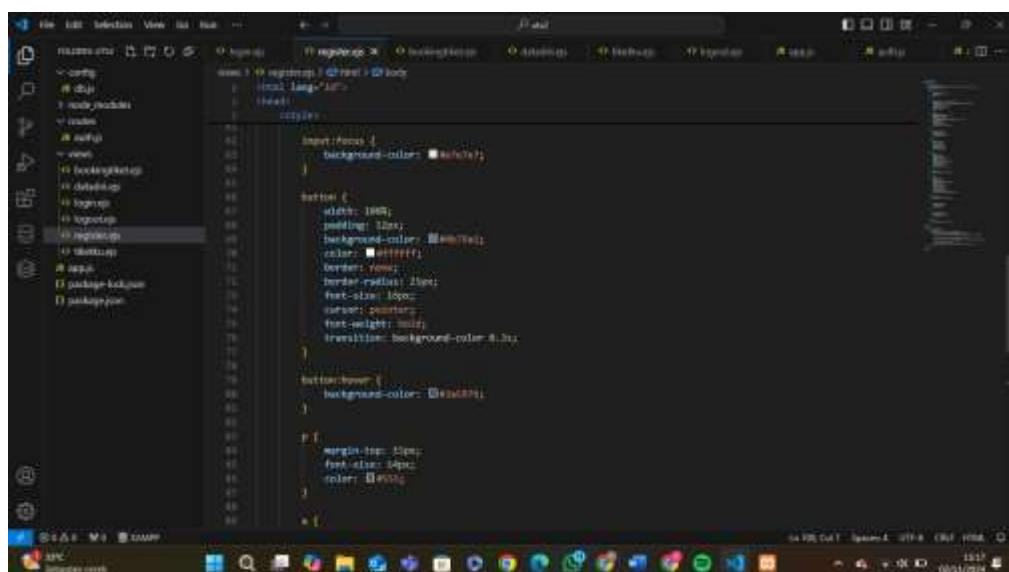
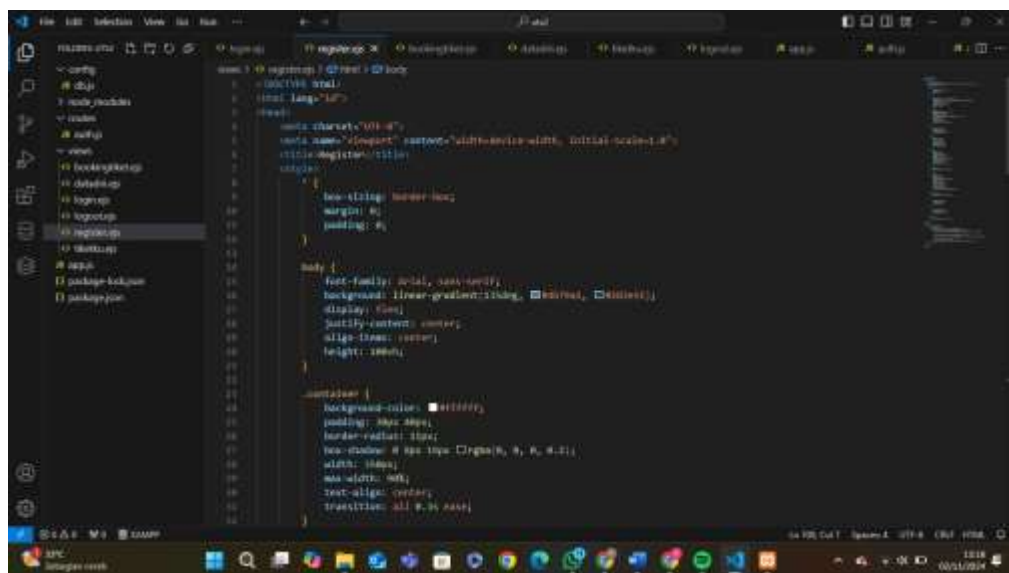


```
<!-- login.ejs -->
<%= include('partials/header') %>
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <div class="card">
        <div class="card-body">
          <div class="text-center">
            <h2>Login</h2>
          </div>
          <div class="form-group">
            <input type="text" class="form-control" placeholder="Email" />
          </div>
          <div class="form-group">
            <input type="password" class="form-control" placeholder="Password" />
          </div>
          <div class="text-center">
            <button type="submit" class="btn btn-primary">Login</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

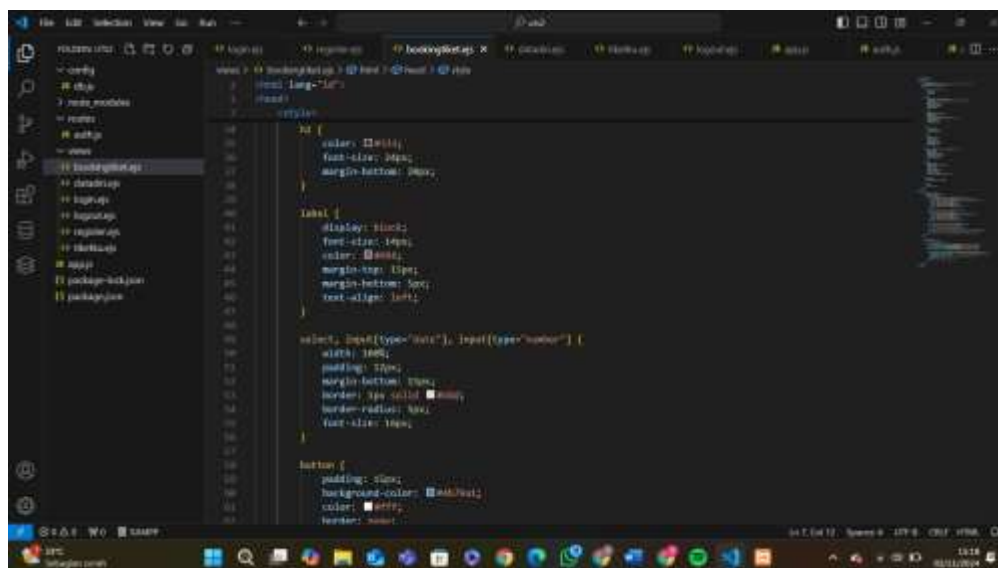


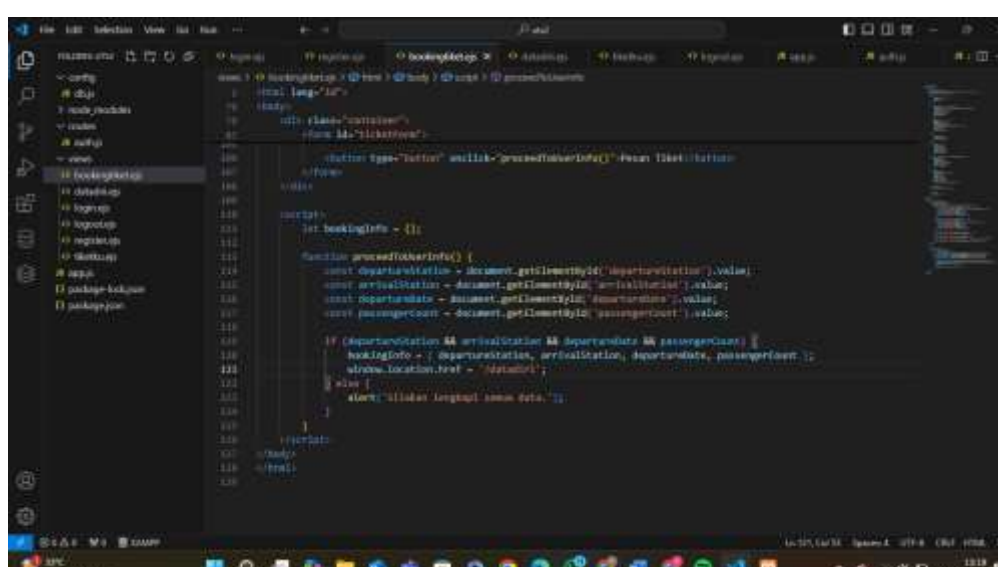


➤ views/register



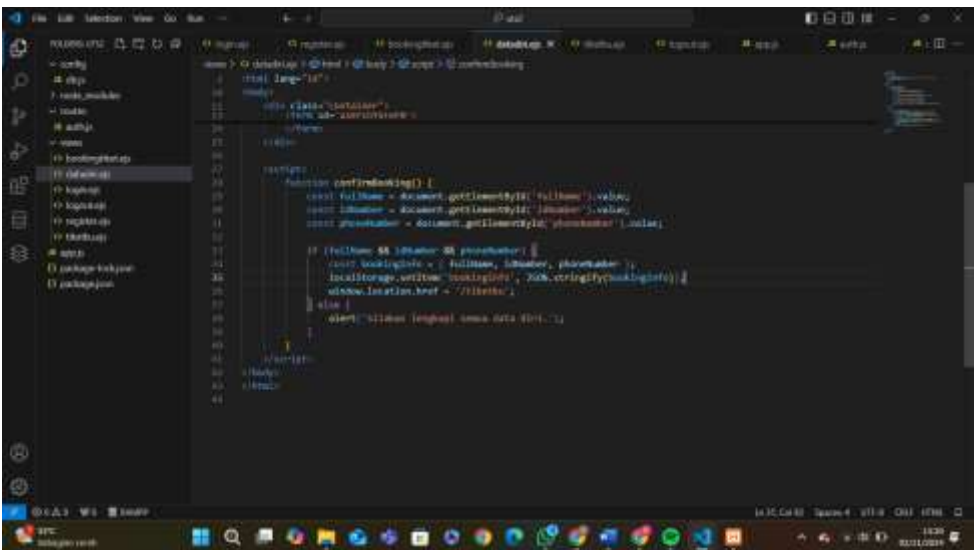
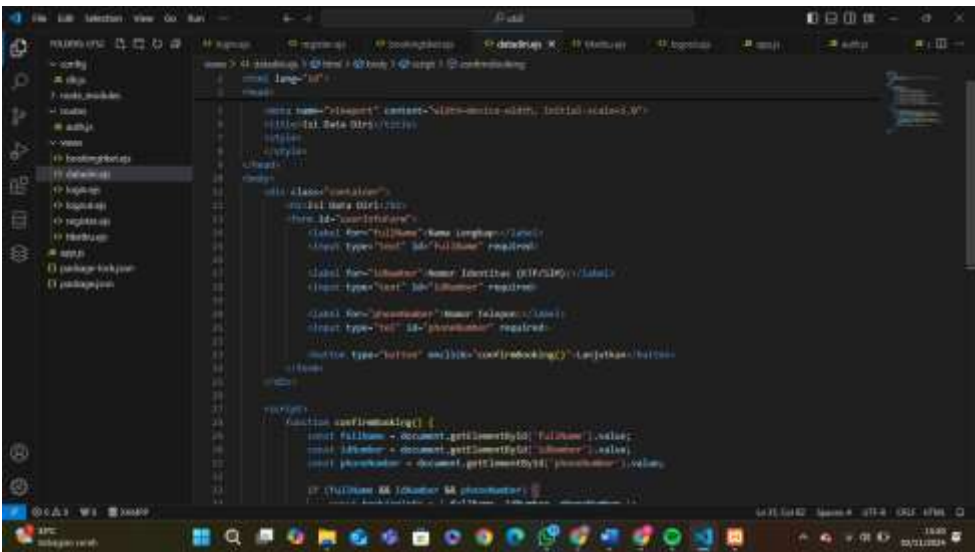




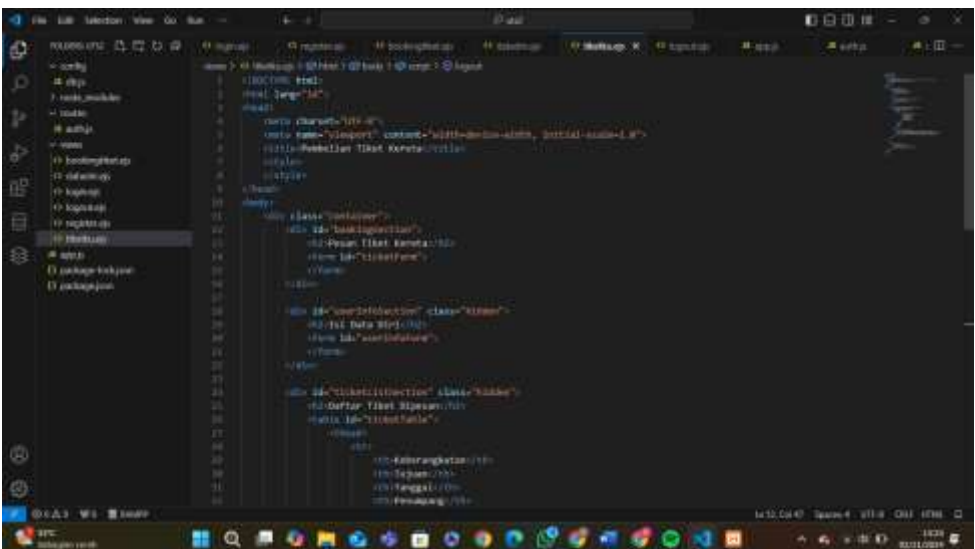




```
➤ views/datadiri.ejs
```



➤ `views/tiketku.ejs`



```

1 <!-- login -->
2 <html lang="id">
3 <head>
4
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>login</title>
8 </head>
9 <body>
10 <div class="container">
11 <div class="row">
12 <div class="col-md-4 offset-md-4">
13 <div class="card">
14 <div class="card-body">
15 <div class="text-center">
16 <h3>login</h3>
17 </div>
18 <div>
19 <input type="text" class="form-control" placeholder="Email" />
20 <input type="password" class="form-control" placeholder="Password" />
21 <button type="button" class="btn btn-primary" value="login">login</button>
22 </div>
23 </div>
24 </div>
25 </div>
26 </div>
27 </body>
28 </html>
29
30 <script>
31 $(document).ready(function() {
32 <!-- login -->
33 <!-- login -->
34 <!-- login -->
35 <!-- login -->
36 <!-- login -->
37 <!-- login -->
38 <!-- login -->
39 <!-- login -->
40 <!-- login -->
41 <!-- login -->
42 <!-- login -->
43 <!-- login -->
44 <!-- login -->
45 <!-- login -->
46 <!-- login -->
47 <!-- login -->
48 <!-- login -->
49 <!-- login -->
50 <!-- login -->
51 <!-- login -->
52 <!-- login -->
53 <!-- login -->
54 <!-- login -->
55 <!-- login -->
56 <!-- login -->
57 <!-- login -->
58 <!-- login -->
59 <!-- login -->
60 <!-- login -->
61 <!-- login -->
62 <!-- login -->
63 <!-- login -->
64 <!-- login -->
65 <!-- login -->
66 <!-- login -->
67 <!-- login -->
68 <!-- login -->
69 <!-- login -->
70 <!-- login -->
71 <!-- login -->
72 <!-- login -->
73 <!-- login -->
74 <!-- login -->
75 <!-- login -->
76 <!-- login -->
77 <!-- login -->
78 <!-- login -->
79 <!-- login -->
80 <!-- login -->
81 <!-- login -->
82 <!-- login -->
83 <!-- login -->
84 <!-- login -->
85 <!-- login -->
86 <!-- login -->
87 <!-- login -->
88 <!-- login -->
89 <!-- login -->
90 <!-- login -->
91 <!-- login -->
92 <!-- login -->
93 <!-- login -->
94 <!-- login -->
95 <!-- login -->
96 <!-- login -->
97 <!-- login -->
98 <!-- login -->
99 <!-- login -->
100 <!-- login -->

```

➤ views/logout

```

1 <!-- login -->
2 <html lang="id">
3 <head>
4
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>login</title>
8 </head>
9 <body>
10 <div class="container">
11 <div class="row">
12 <div class="col-md-4 offset-md-4">
13 <div class="card">
14 <div class="card-body">
15 <div class="text-center">
16 <h3>login</h3>
17 </div>
18 <div>
19 <input type="text" class="form-control" placeholder="Email" />
20 <input type="password" class="form-control" placeholder="Password" />
21 <button type="button" class="btn btn-primary" value="login">login</button>
22 </div>
23 </div>
24 </div>
25 </div>
26 </body>
27 </html>
28
29 <script>
30 $(document).ready(function() {
31 <!-- login -->
32 <!-- login -->
33 <!-- login -->
34 <!-- login -->
35 <!-- login -->
36 <!-- login -->
37 <!-- login -->
38 <!-- login -->
39 <!-- login -->
40 <!-- login -->
41 <!-- login -->
42 <!-- login -->
43 <!-- login -->
44 <!-- login -->
45 <!-- login -->
46 <!-- login -->
47 <!-- login -->
48 <!-- login -->
49 <!-- login -->
50 <!-- login -->
51 <!-- login -->
52 <!-- login -->
53 <!-- login -->
54 <!-- login -->
55 <!-- login -->
56 <!-- login -->
57 <!-- login -->
58 <!-- login -->
59 <!-- login -->
60 <!-- login -->
61 <!-- login -->
62 <!-- login -->
63 <!-- login -->
64 <!-- login -->
65 <!-- login -->
66 <!-- login -->
67 <!-- login -->
68 <!-- login -->
69 <!-- login -->
70 <!-- login -->
71 <!-- login -->
72 <!-- login -->
73 <!-- login -->
74 <!-- login -->
75 <!-- login -->
76 <!-- login -->
77 <!-- login -->
78 <!-- login -->
79 <!-- login -->
80 <!-- login -->
81 <!-- login -->
82 <!-- login -->
83 <!-- login -->
84 <!-- login -->
85 <!-- login -->
86 <!-- login -->
87 <!-- login -->
88 <!-- login -->
89 <!-- login -->
90 <!-- login -->
91 <!-- login -->
92 <!-- login -->
93 <!-- login -->
94 <!-- login -->
95 <!-- login -->
96 <!-- login -->
97 <!-- login -->
98 <!-- login -->
99 <!-- login -->
100 <!-- login -->

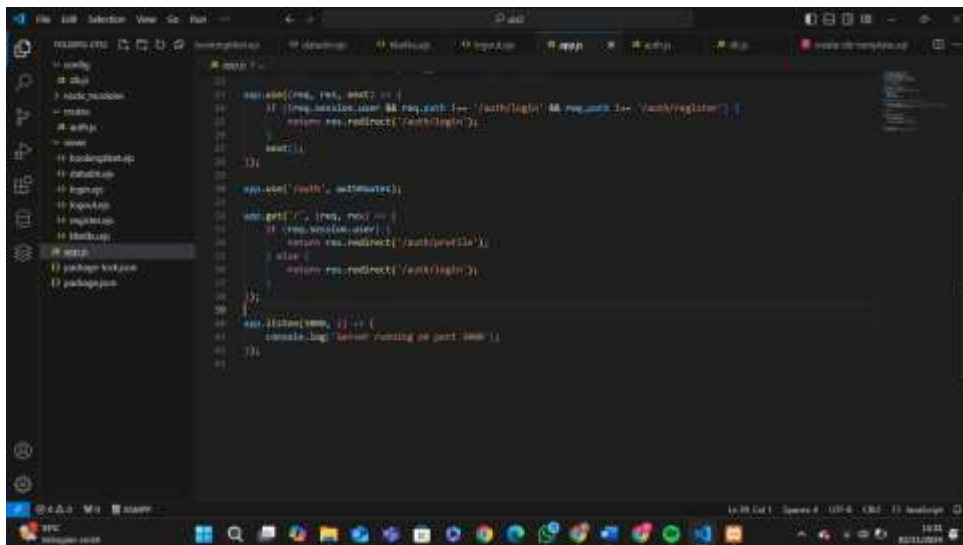
```

➤ app.js

```

1 const express = require('express');
2 const app = express();
3 const port = 3000;
4 const bodyParser = require('body-parser');
5 const session = require('express-session');
6 const auth = require('./auth');
7
8 const app = express();
9
10 app.use(bodyParser.json());
11 app.use(bodyParser.urlencoded({ extended: true }));
12
13 app.use(session({
14   secret: 'secret',
15   resave: false,
16   saveUninitialized: true
17 }));
18
19 app.use(express.static(path.resolve(__dirname, 'public')));
20
21 app.use((req, res, next) => {
22   if (req.session.user && req.session.user.role === 'admin') {
23     return next();
24   }
25   res.redirect('/login');
26 });
27
28 app.use(auth);
29
30 app.listen(port, () => {
31   console.log(`Server is running on port ${port}`);
32 });

```



## • PANDUAN PENGGUNAAN

### 1. Halaman Register

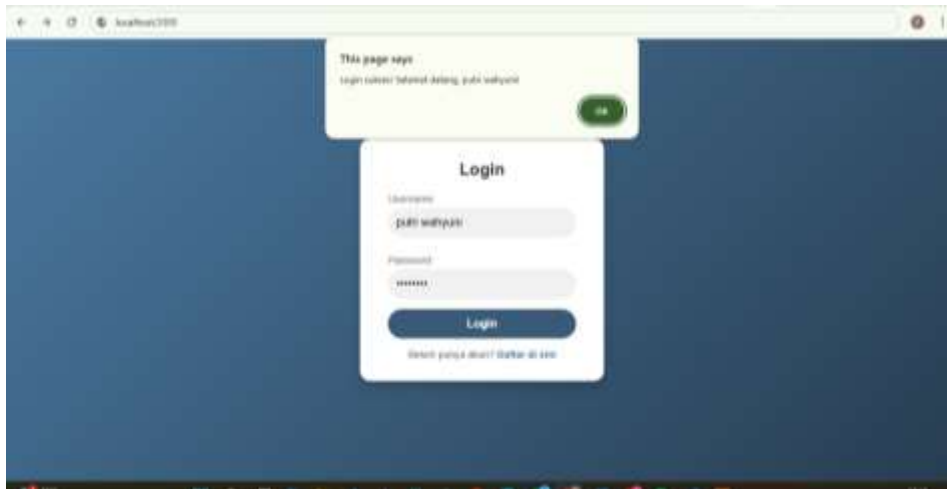
Sebelum dapat menggunakan website ini, pengguna diwajibkan untuk melakukan registrasi terlebih dahulu guna membuat akun. Pada proses registrasi, pengguna akan diminta untuk mengisi beberapa informasi penting, seperti username dan password. Setelah register pengguna dapat mengklik button daftar. Maka setelah itu akan dialihkan kehalaman login.



### 2. Halaman Login

Pada halaman login, pengguna diminta untuk memasukkan Username dan Password pada kolom yang disediakan. Setelah mengisi kolom tersebut, pengguna dapat menekan tombol Login untuk masuk ke dalam halaman selanjutnya yaitu halaman untuk memesan tiket kereta. Jika login berhasil, akan muncul notifikasi bertuliskan "Login sukses! Selamat datang, [nama pengguna]" untuk mengonfirmasi bahwa pengguna berhasil masuk. Di bagian bawah formulir login, terdapat opsi bagi pengguna yang

belum memiliki akun untuk melakukan pendaftaran dengan mengklik Daftar di sini dan akan kembali kehalaman register.

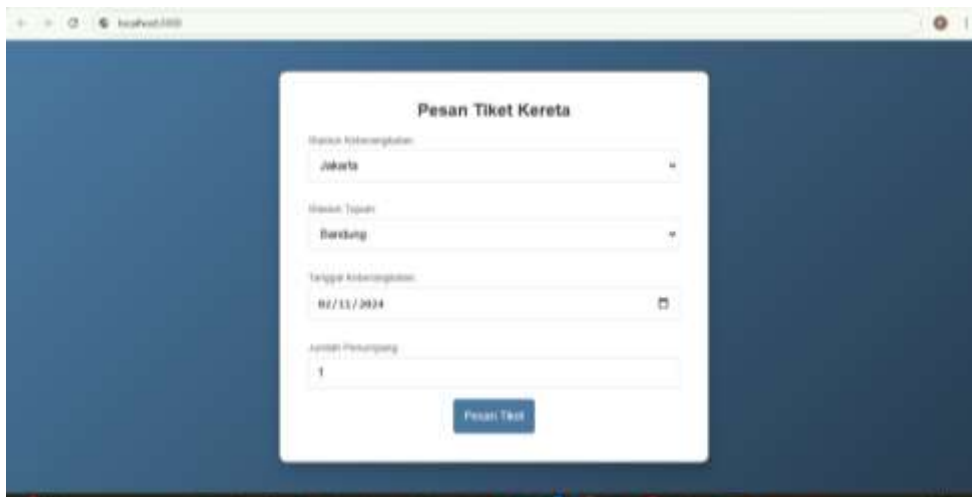


### 3. Halaman Pesan Tiket Kereta

Pada halaman ini, pengguna dapat melakukan pemesanan tiket dengan mengisi beberapa informasi berikut:

1. Stasiun Keberangkatan: Pengguna memilih stasiun asal keberangkatan dari daftar pilihan yang tersedia. Pada contoh ini, stasiun yang dipilih adalah "Jakarta."
2. Stasiun Tujuan: Pengguna memilih stasiun tujuan dari daftar pilihan yang tersedia. Pada contoh ini, stasiun tujuan yang dipilih adalah "Bandung."
3. Tanggal Keberangkatan: Pengguna memilih tanggal keberangkatan dengan memilih tanggal pada kolom yang tersedia atau menggunakan ikon kalender untuk mempermudah pemilihan tanggal. Tanggal yang dipilih dalam contoh ini adalah "02/11/2024."
4. Jumlah Penumpang: Pengguna memasukkan jumlah penumpang yang ingin memesan tiket pada kolom yang disediakan. Pada contoh ini, jumlah penumpang yang dimasukkan adalah "1."

Setelah semua data terisi, pengguna dapat menekan tombol Pesan Tiket untuk melanjutkan proses pemesanan. Setelah itu pengguna akan dialihkan kehalaman selanjutnya, yaitu halaman untuk mengisi data diri.



**Pesanan Tiket Kerata**

Dari Kota: Jakarta

Ke Kota: Bandung

Tanggal Keberangkatan: 02/11/2024

Jumlah Penumpang: 1

Pesan Tiket

#### 4. Halaman Data Diri

Pada halaman ini, pengguna diminta untuk memasukkan data diri mereka sebagai bagian dari proses pemesanan. Berikut adalah data yang harus diisi:

1. Nama Lengkap: Pengguna mengisi kolom ini dengan nama lengkap sesuai identitas resmi. Pada contoh ini, nama yang diisi adalah "putri wahyuni."
2. Nomor Identitas (KTP/SIM): Pengguna memasukkan nomor identitas resmi, seperti KTP atau SIM.
3. Nomor Telepon: Pengguna mengisi nomor telepon yang dapat dihubungi.

Setelah semua data terisi, pengguna dapat menekan tombol Lanjutkan untuk melanjutkan ke langkah berikutnya dalam proses pemesanan tiket, dimana akan dialihkan kehalaman selanjutnya, yaitu halaman Daftar Tiket Dipesan.



**Isi Data Diri**

Nama Lengkap: putri wahyuni

Nomor Identitas (KTP/SIM): 3201150000020000

Nomor Telepon: 081202501400

Lanjutkan

#### 5. Halaman Daftar Tiket yang Sudah Dipesan

Pada halaman ini menampilkan daftar tiket kereta yang telah dipesan oleh pengguna. Di bagian atas tabel, terdapat kolom-kolom yang berisi informasi detail mengenai tiket yang dipesan, seperti Keberangkatan, Tujuan, Tanggal, Penumpang, dan Aksi.



- Kolom Keberangkatan: Menunjukkan stasiun asal tempat keberangkatan kereta.
- Kolom Tujuan: Menampilkan stasiun tujuan dari perjalanan kereta.
- Kolom Tanggal: Menunjukkan tanggal keberangkatan kereta yang dipilih.
- Kolom Penumpang: Berisi jumlah penumpang yang dipesan untuk perjalanan tersebut.
- Kolom Aksi: Di sini terdapat tombol Batalkan yang memungkinkan pengguna untuk membatalkan tiket yang telah dipesan.

Di bawah tabel, terdapat tiga tombol utama yang berfungsi sebagai berikut:

- Hapus Semua Tiket: Menghapus semua tiket yang ada dalam daftar.
- Pesan Tiket Baru: Membuka halaman pemesanan tiket baru jika pengguna ingin memesan perjalanan lainnya.
- Logout: Keluar dari akun pengguna dan mengarahkan kembali ke halaman login.

