

LAPORAN PRAKTIKUM MATA KULIAH
PEMOGRAMAN WEB



Pertemuan 5 (Tugas 2) Praktikum 5

Dosen Pengampu :

Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh :

Syifa Fauzia (2309845)

SIK B/3

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA

2024

1. Pendahuluan

Pada praktikum ini, saya akan membuat sebuah aplikasi web sederhana dengan fitur) menggunakan Node.js dengan framework Express, MySQL sebagai basis data, serta EJS (Embedded JavaScript) sebagai template engine untuk menampilkan data di sisi frontend. Session adalah mekanisme yang digunakan dalam aplikasi web untuk menyimpan informasi pengguna secara sementara saat mereka berinteraksi dengan aplikasi tersebut. Dalam konteks Node.js, session digunakan untuk melacak dan mengingat status pengguna di antara permintaan HTTP (request) yang berbeda.

Point penting session

1. Penyimpanan Informasi Sementara: Ketika pengguna login, informasi seperti ID pengguna, username, atau token disimpan dalam session, sehingga server dapat mengingat pengguna tersebut selama sesi mereka berlangsung.
2. Terikat pada Pengguna: Setiap pengguna yang terhubung ke aplikasi memiliki session unik. Ketika pengguna melakukan request baru (misalnya berpindah halaman), session yang terkait dengannya akan diidentifikasi dan diakses.
3. Berbasis Cookie: Session biasanya disimpan di server, dan untuk melacak session tersebut, browser pengguna akan menyimpan cookie yang berisi ID session. Server akan memeriksa cookie ini untuk mengidentifikasi session pengguna.
4. Contoh Penggunaan:
 - Ketika pengguna login, aplikasi menyimpan informasi login dalam session.
 - Pada request berikutnya, aplikasi dapat memeriksa session untuk memastikan bahwa pengguna sudah login.
 - Jika pengguna logout, session akan dihapus, sehingga pengguna harus login kembali.

2. Penjelasan code

1. penjelasan app.js

```
JS app.js > ...
1  const express = require ('express');
2  const bodyParser = require ('body-parser');
3  const session = require ('express-session');
4  const authRoutes = require ('./routes/auth');
5  const path = require ('path');
6
7  const app = express () ;
8
9  //set EJS sebagai template engine
10 app.set ('view engine', 'ejs');
11
12 app.use(express.static(path.join(__dirname, 'public')));
13
14 //middleware
15 app.use (bodyParser.json () ) ;
16 app.use (bodyParser.urlencoded ({ extended: true }));
17 app.use (session({
18     secret: 'secret',
19     resave: false,
20     saveUninitialized: true
21 }));
22
23
24
25 //middleware to chech login status
26 app.use( (req, res, next) => {
27     if (!req.session.user && req.path !== '/auth/login' && req.path !== '/auth/register') {
28         // if the user is not logged in and trying to access any other page except login/register
29         return res.redirect ('/auth/login');
30     } else if (req.session.user && req.path === '/') {
31         // if user is logged in and tries to access the root route, redirect to profile
32         return res.redirect ('/auth/profile');
33     }
34     next ();
35 });
36
37 // Routes
38 app.use ('/auth', authRoutes);
39
40 //Root Route: Redirect to /auth/login or /auth/profile based on session
41 app.get ('/', (req, res) => {
42     if (req.session.user) {
43         return res.redirect ('/auth/profile');
44     } else {
45         return res.redirect ('/auth/login');
46     }
47 });
48
49 // Menjalankan Server
50 app.listen (3000, () => {
51     console.log ('Server running on port 3000, open web via http://localhost:3000');
52 });
```

□ Imports dan Setup Awal:

- express, body-parser, express-session, path, dan modul routing (authRoutes) digunakan untuk membangun server.
- app.set('view engine', 'ejs') digunakan untuk mengatur EJS sebagai template engine.

□ Static Files:

- `app.use(express.static(path.join(__dirname, 'public')))` untuk menyajikan file statis dari folder public.

□ Middleware:

- `body-parser` digunakan untuk mem-parsing JSON dan data form.
- `express-session` digunakan untuk mengatur sesi pengguna, dengan secret sebagai kunci enkripsi sesi.
- Middleware untuk mengecek status login memeriksa apakah pengguna sudah login, dan menentukan apakah akan diarahkan ke halaman login, register, atau profil.

□ Routing:

- Rute `/auth` menggunakan `authRoutes`, yang mungkin mengelola proses login dan register.
- Rute root `(/)` akan mengarahkan pengguna ke `/auth/profile` jika sudah login atau ke `/auth/login` jika belum login.

□ Menjalankan Server:

- Server dijalankan pada port 3000, dan pesan akan ditampilkan untuk menunjukkan bahwa server sudah aktif.

2. penjelasan `auth.js`

```
JS auth.js X
routes > JS auth.js > ...
1  const express = require ('express') ;
2  const router = express.Router () ;
3  const bcrypt = require ('bcryptjs') ;
4  const db = require ('../config/db') ;
5
6  // Render halaman register
7  router.get ('/register', (req, res) => {
8    res.render('register');
9  }) ;
10
11 // Proses register user
12 router.post ('/register', (req, res) => {
13   const { Username, Email, Password}= req.body;
14
15   const hashedPassword = bcrypt.hashSync (Password, 10) ;
16
17   const query = "INSERT INTO users (Username, Email, Password) VALUES (?, ?, ?) ";
18   db.query (query, [Username, Email, hashedPassword], (err, result) => {
19     if (err) throw err;
20     res.redirect ('/auth/login');
21   }) ;
22 }) ;
23
24 // Render halaman login
25 router.get ('/login', (req, res) => {
26   res.render ('login');
27 });
```

```

28
29 // Proses login user
30 router.post ('/login', (req, res) => {
31     const { Username, Password } = req. body;
32     const query = "SELECT * FROM users WHERE Username = ?";
33     db.query (query, [Username], (err, result) => {
34         if (err) throw err;
35
36         if (result.length > 0) {
37             const user = result [0];
38
39             if (bcrypt.compareSync (Password, user.Password)) {
40                 req.session.user = user;
41                 res.redirect ('/auth/profile');
42             } else {
43                 res.send ('Incorrect password' );
44             }
45         } else {
46             res.send ('user not found' );
47         }
48     });
49 });
50
51 // Render halaman profil user
52 router.get ('/profile', (req, res) => {
53     if (req.session.user) {
54         res.render ('profile', { user: req.session.user });
55     } else {
56         res.redirect ('/auth/login');
57     }
58 });
59
60 // Proses logout
61 router.get ('/logout', (req, res) => {
62     req.session.destroy ();
63     res.redirect ('/auth/login' );
64 });
65
66 module.exports = router;

```

□ Imports dan Setup:

- express digunakan untuk membuat router.
- bcryptjs digunakan untuk mengenkripsi kata sandi.
- db adalah konfigurasi koneksi ke database.

□ Rute Register:

- GET /register: Merender halaman form register.
- POST /register: Memproses pendaftaran pengguna baru dengan:
 - Mengambil data Username, Email, dan Password dari body request.
 - Mengenkripsi Password menggunakan bcrypt.
 - Menyimpan pengguna baru ke database dalam tabel users.
 - Setelah berhasil, mengarahkan ke halaman login (/auth/login).

□ **Rute Login:**

- GET /login: Merender halaman form login.
- POST /login: Memproses login pengguna dengan:
 - Mengambil Username dan Password dari body request.
 - Mengecek apakah Username ada di database.
 - Jika ditemukan, membandingkan Password yang dienkripsi.
 - Jika kata sandi benar, menyimpan data pengguna di session dan mengarahkan ke halaman profil.
 - Jika salah, mengirim pesan "Incorrect password" atau "User not found".

□ **Rute Profil:**

- GET /profile: Merender halaman profil pengguna jika sudah login; jika tidak, mengarahkan ke halaman login.

□ **Rute Logout:**

- GET /logout: Menghapus sesi pengguna dan mengarahkan kembali ke halaman login.

□ **Ekspor Router:**

- `module.exports = router` digunakan untuk mengekspor router agar bisa digunakan di file utama aplikasi.

3. penjelasan *db.js*

```
JS db.js X
config > JS db.js > [db] database
1  const mysql = require ('mysql');
2
3  const db = mysql.createConnection ({
4    host: 'localhost',
5    user: 'root',
6    password: '',
7    database: 'pemweb5'
8  })
9
10 db.connect ((err) => {
11   if (err) throw err;
12   console.log('Database connected...');
13 });
14 module.exports = db;
```

□ Imports dan Setup:

- mysql digunakan untuk membuat koneksi ke database MySQL.

□ Konfigurasi Koneksi:

- mysql.createConnection membuat koneksi ke database dengan konfigurasi berikut:
 - host: 'localhost': Database berjalan secara lokal.
 - user: 'root': Nama pengguna untuk mengakses database.
 - password: '': Kata sandi pengguna, yang dalam contoh ini dibiarkan kosong.
 - database: 'pemweb5': Nama database yang digunakan adalah pemweb5.

□ Membuka Koneksi:

- db.connect digunakan untuk terhubung ke database.
- Jika ada error saat menghubungkan, program akan menghentikan proses (throw err).
- Jika berhasil, akan menampilkan pesan "Database connected...".

□ Ekspor Modul:

- module.exports = db memungkinkan objek db untuk diakses oleh file lain dalam aplikasi, seperti saat menjalankan query.

4. penjelasan login.ejs

```
login.ejs
views > login.ejs > ? > html > body > div.container
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="/styles.css">
7   <title>login</title>
8 </head>
9 <body>
10   <div class="container">
11     <h2>Login</h2>
12     <form action="/auth/login" method="POST">
13       <label for="Username">Username</label>
14       <input type="text" id="Username" Name="Username" required> <br>
15
16       <label for="Password">Password</label>
17       <input type="Password" id="Password" Name="Password" required> <br>
18
19       <label for="confirmPassword">Confirm Password</label>
20       <input type="password" name="confirmPassword" id="confirmPassword" required><br>
21
22       <button type="submit">Login</button>
23     </form>
24     <p>Don't have an account? <a href="/auth/register">Register
25     here</a></p>
```

```

27     <script>
28         document.querySelector('form').addEventListener('submit', function(event) {
29             const Password = document.getElementById('Password').value;
30             const confirmPassword = document.getElementById('confirmPassword').value;
31
32             if (Password !== confirmPassword) {
33                 event.preventDefault(); // prevent form submission
34                 alert('Passwords do not match!');
35             }
36         });
37     </script>
38 </div>
39 </body>
40 </html>

```

□ Header HTML:

- <!DOCTYPE html>: Mendefinisikan dokumen sebagai HTML5.
- <html lang="en">: Menentukan bahasa dokumen sebagai bahasa Inggris.
- <meta charset="UTF-8"> dan <meta name="viewport" content="width=device-width, initial-scale=1.0">: Menyediakan pengaturan karakter dan membuat halaman responsif pada berbagai perangkat.
- <link rel="stylesheet" href="/styles.css">: Menghubungkan halaman ke file CSS eksternal bernama styles.css untuk menambahkan gaya.

□ Body HTML:

- <div class="container">: Pembungkus utama untuk konten halaman.
- <h2>Login</h2>: Menampilkan judul "Login".
- <form action="/auth/login" method="POST">: Formulir yang mengirim data ke server melalui metode POST ke rute /auth/login.
 - label dan input untuk Username, Password, dan Confirm Password, memastikan bahwa semua kolom harus diisi (required).
 - Tombol submit untuk mengirim formulir.

□ Link Register:

- <p>Don't have an account? Register here</p>: Tautan yang mengarahkan pengguna ke halaman register jika belum memiliki akun.

□ JavaScript Validation:

- Script JavaScript ditambahkan untuk memvalidasi apakah Password dan Confirm Password cocok.
- Jika tidak cocok, form tidak akan dikirim (event.preventDefault()), dan pesan peringatan "Passwords do not match!" akan muncul.

□ Penutupan Tag:

- Struktur penutup untuk </div>, </body>, dan </html> untuk menyelesaikan dokumen.

5. penjelasan profile.ejs

```
<> profile.ejs X
views > <> profile.ejs > ? > html > head > title
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="/styles.css">
7      <title>profile</title>
8  </head>
9  <body>
10
11      <div class="container">
12          <h2>Welcome, <%= user.Username %></h2>
13          <p>Email: <%= user.Email %></p>
14          <a href="/auth/logout">Logout</a>
15      </div>
16  </body>
17  </html>
```

□ Header HTML:

- `<!DOCTYPE html>`: Mendefinisikan dokumen sebagai HTML5.
- `<html lang="en">`: Menentukan bahasa dokumen sebagai bahasa Inggris.
- `<meta charset="UTF-8">` dan `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Menetapkan karakter dan membuat halaman responsif pada berbagai perangkat.
- `<link rel="stylesheet" href="/styles.css">`: Menghubungkan halaman ke file CSS eksternal styles.css untuk menambahkan gaya.
- `<title>profile</title>`: Menentukan judul halaman sebagai "profile".

□ Body HTML:

- `<div class="container">`: Pembungkus utama untuk konten halaman.
- `<h2>Welcome, <%= user.Username %></h2>`: Menampilkan pesan sambutan dengan nama pengguna yang sedang login. `<%= user.Username %>` merupakan sintaks EJS yang mengambil data Username dari objek user yang dikirim dari server.
- `<p>Email: <%= user.Email %></p>`: Menampilkan alamat email pengguna yang diambil dari objek user dengan sintaks EJS `<%= user.Email %>`.
- `Logout`: Tautan yang memungkinkan pengguna untuk keluar dari sesi saat ini dengan mengakses rute /auth/logout.

□ Penutupan Tag:

- Menutup tag `</div>`, `</body>`, dan `</html>` untuk menyelesaikan dokumen HTML.

6. penjelasan register.ejs

```
register.ejs X
views > register.ejs > ? > html > body > div.container > form > label > button
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="/styles.css">
7   <title>register</title>
8 </head>
9 <body>
10   <div class="container">
11     <h2>register</h2>
12     <form action="/auth/register" method="POST">
13       <label for="Username">Username</label>
14       <input type="text" id="Username" Name="Username" required> <br>
15
16       <label for="Email">Email</label>
17       <input type="Email" id="Email" Name="Email" required> <br>
18
19       <label for="confirmPassword">Confirm Password</label>
20       <input type="Password" name="confirmPassword" id="confirmPassword" required><br>
21
22       <label for="Password">Passwords</label>
23       <input type="Password" id="Password" Name="Password" required> <br>
24
25       <button type="submit">register</button>
26     </form>
27     <p>Already have an account? <a href="/auth/login"> Login here</a></p>
28   </div>
29 <script>
30   document.querySelector('form').addEventListener('submit', function(event) {
31     const password = document.getElementById('Password').value;
32     const confirmPassword = document.getElementById('confirmPassword').value;
33
34     if (password !== confirmPassword) {
35       event.preventDefault(); // prevent form submission
36       alert('Passwords do not match!');
37     }
38   });
39 </script>
40 </body>
41 </html>
```

□ Header HTML:

- <!DOCTYPE html>: Menentukan dokumen sebagai HTML5.
- <html lang="en">: Menyatakan bahasa halaman sebagai bahasa Inggris.
- <meta charset="UTF-8"> dan <meta name="viewport" content="width=device-width, initial-scale=1.0">: Menetapkan karakter dan membuat halaman responsif pada berbagai perangkat.
- <link rel="stylesheet" href="/styles.css">: Menghubungkan halaman dengan file CSS eksternal styles.css untuk menambahkan gaya.

□ Body HTML:

- <div class="container">: Pembungkus utama untuk konten halaman.
- <h2>register</h2>: Menampilkan judul "register".

- `<form action="/auth/register" method="POST">`: Formulir yang mengirim data ke server melalui metode POST ke rute `/auth/register`.
 - Input untuk Username, Email, Password, dan Confirm Password dengan masing-masing kolom memiliki atribut `required` untuk memastikan bahwa mereka tidak boleh kosong.
 - Terdapat kesalahan kecil dalam kode: `<label for="Password">Passwords/label>` seharusnya ditulis sebagai `<label for="Password">Password</label>`.

□ Link Login:

- `<p>Already have an account? Login here</p>`: Tautan yang mengarahkan pengguna ke halaman login jika sudah memiliki akun.

□ Validasi JavaScript:

- Script JavaScript digunakan untuk memeriksa apakah Password dan Confirm Password sesuai.
- Jika tidak cocok, form tidak akan dikirim (`event.preventDefault()`), dan akan muncul pesan peringatan "Passwords do not match!".

□ Penutupan Tag:

- Menutup tag `</div>`, `</script>`, `</body>`, dan `</html>` untuk menyelesaikan struktur dokumen.

7. penjelasan styles.css

```
# styles.css
public > # styles.css > body
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f0f0f0;
4      background-image: url("bg5.jpeg");
5      background-size: cover;
6      background-position: center;
7      background-repeat: no-repeat;
8      display: flex;
9      justify-content: center;
10     align-items: center;
11     height: 100h;
12     margin: 0;
13 }
14
15 .container {
16     background-color: #fff;
17     padding: 20px;
18     border-radius: 10px;
19     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
20     width: 300px;
21 }
22
23 h2 {
24     text-align: center;
25 }
26
27 label {
28     display: block;
29     margin-bottom: 5px;
30 }
```

```

32 input {
33   width: 100%;
34   padding: 8px;
35   margin-bottom: 15px;
36   border: 1px solid #ccc;
37   border-radius: 5px;
38 }
39
40 button {
41   width: 100%;
42   padding: 10px;
43   background-color: #fd65ac;
44   color: white;
45   border: none;
46   border-radius: 5px;
47   cursor: pointer;
48 }
49
50 button:hover {
51   background-color: #fd65ac;
52 }
53
54 p {
55   text-align: center;
56 }
57
58 a {
59   color: #fd65ac;
60   text-decoration: none;
61 }
62
63 a:hover {
64   text-decoration: underline;
65 }

```

□ body Styling:

- Menggunakan font Arial, sans-serif.
- Mengatur latar belakang menjadi abu-abu muda (#f0f0f0) dengan gambar latar "bg5.jpeg", yang diatur agar menutupi seluruh layar (background-size: cover), berada di tengah (background-position: center), dan tidak mengulangi gambar (background-repeat: no-repeat).
- Menampilkan konten secara fleksibel dengan display: flex, serta memposisikan konten di tengah secara horizontal dan vertikal (justify-content: center dan align-items: center).
- Mengatur tinggi halaman penuh (height: 100vh) dan margin nol.

□ .container Styling:

- Latar belakang putih (background-color: #fff), dengan padding 20px, dan sudut membulat (border-radius: 10px).
- Menambahkan bayangan (box-shadow: 0 0 10px rgba(0, 0, 0, 0.1)) untuk memberikan efek elevasi.
- Lebar kontainer diatur menjadi 300px.

□ h2 Styling:

- Mengatur judul (<h2>) agar rata tengah (text-align: center).

□ label Styling:

- Ditampilkan dalam bentuk blok (display: block) dengan jarak bawah 5px agar rapi.

□ **input Styling:**

- Lebarnya 100%, dengan padding 8px, jarak bawah 15px, batas (border) berwarna abu-abu muda (#ccc), dan sudut yang membulat (border-radius: 5px).

□ **button Styling:**

- Lebarnya 100%, padding 10px, dengan warna latar belakang merah muda (#fd65ac), teks putih, tanpa batas, dan sudut yang membulat.
- Menambahkan kursor pointer untuk menunjukkan bahwa tombol dapat diklik.

□ **button: hover Styling:**

- Mengatur warna latar belakang yang sama saat tombol dilewati kursor (#fd65ac).

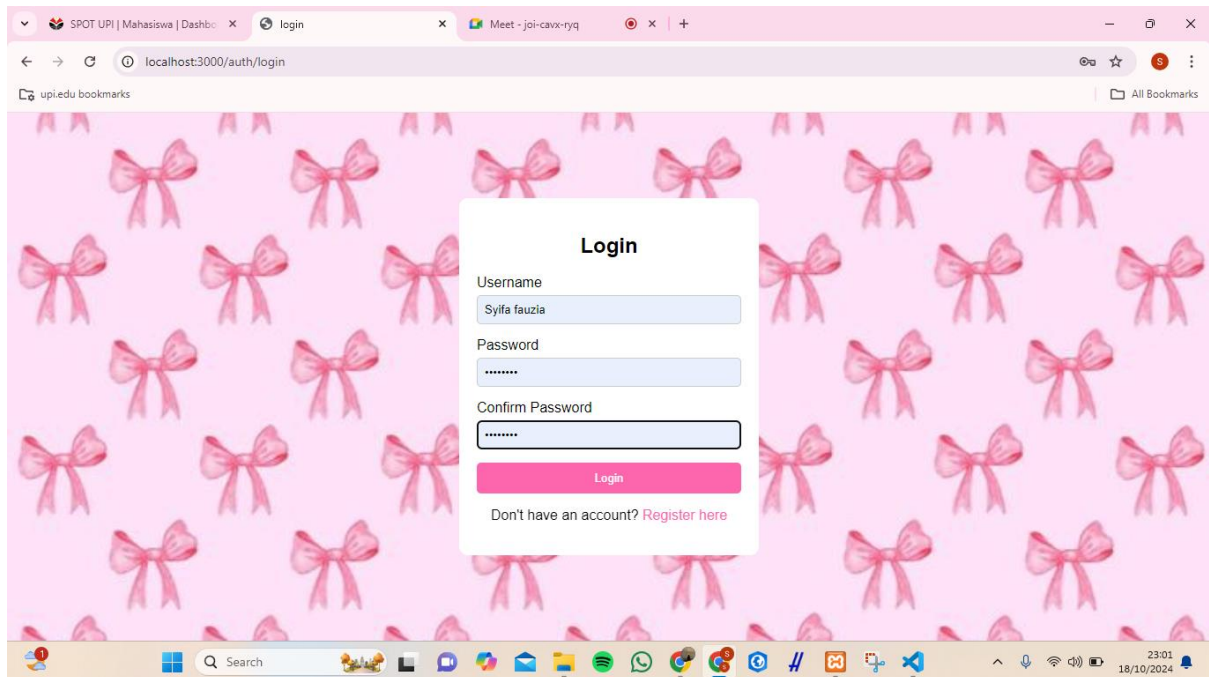
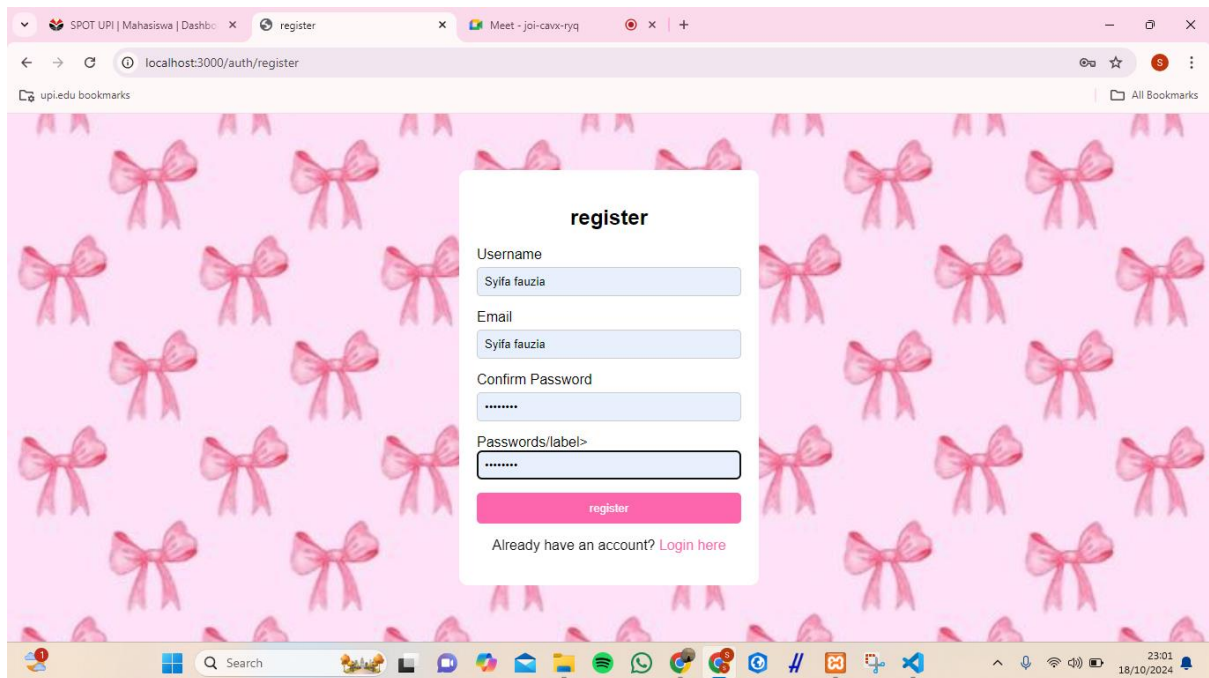
□ **p Styling:**

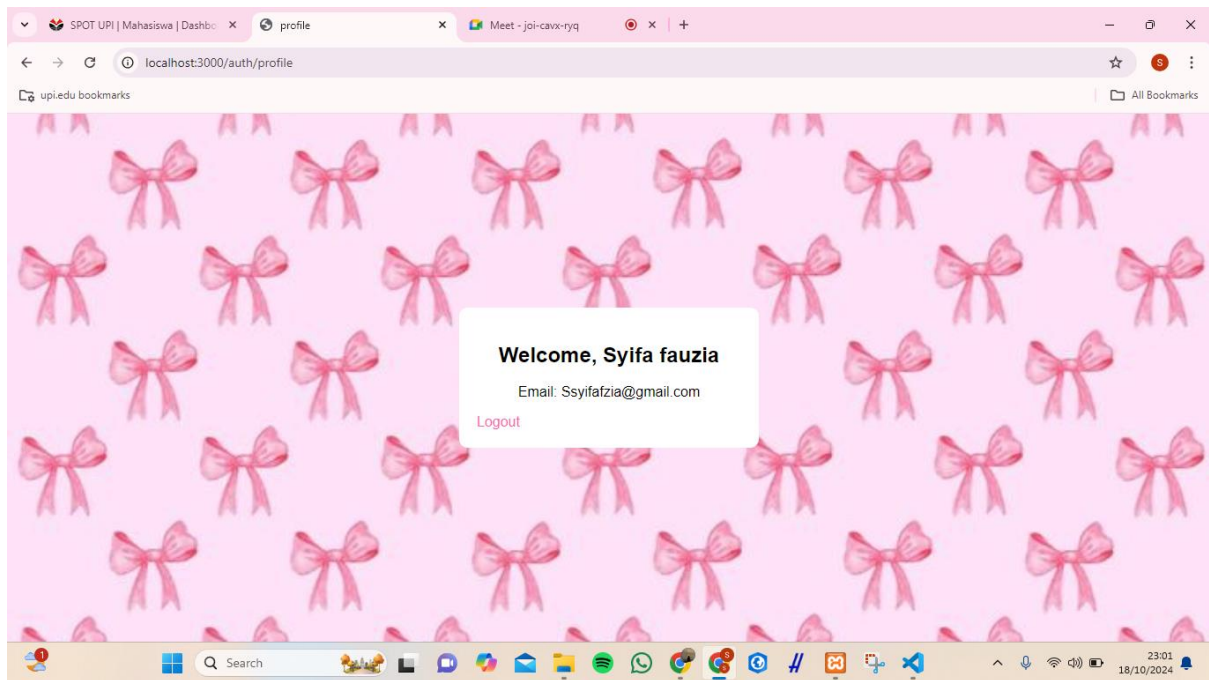
- Teks paragraf diratakan di tengah (text-align: center).

□ **a Styling:**

- Mengatur warna teks tautan menjadi merah muda (#fd65ac) dan menghilangkan garis bawah (text-decoration: none).
- Menambahkan garis bawah pada tautan saat dilewati kursor (a: hover).

HASIL AKHIR WEB





Web memiliki tiga halaman utama:

- **Halaman Register:** Untuk pengguna baru mendaftar dengan memasukkan informasi seperti username, email, dan password.
- **Halaman Login:** Untuk pengguna yang telah terdaftar dapat masuk dengan username dan password mereka.
- **Halaman Profil:** Menampilkan informasi pengguna yang sudah login, termasuk nama dan email mereka. Terdapat juga opsi untuk logout.

3. Kesimpulan

Kesimpulan dari laporan praktikum Pemrograman Web ini adalah bahwa aplikasi web sederhana telah berhasil dibuat menggunakan Node.js dengan framework Express, MySQL sebagai basis data, dan EJS sebagai template engine. Fitur utama aplikasi ini termasuk sistem login, register, dan profil pengguna yang menggunakan session untuk melacak status pengguna. Aplikasi ini melibatkan pengelolaan pengguna melalui otentikasi dengan bcryptjs untuk keamanan kata sandi. Hasil akhir aplikasi memiliki tiga halaman utama: register, login, dan profil, dengan desain front-end yang responsif dan terstruktur rapi.