LAPORAN PRAKTIKUM MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK

Pertemuan 7 Tugas 6

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 ters but 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 didentifikasi 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.

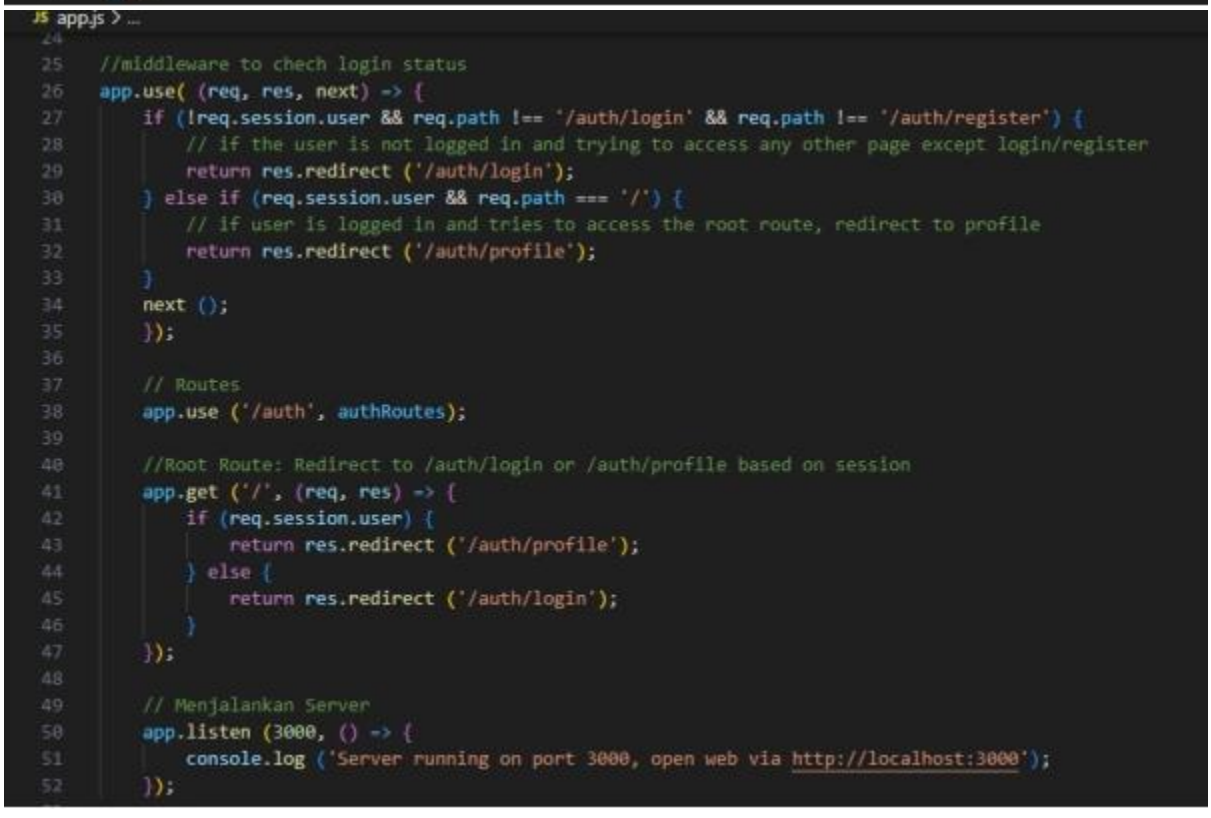
- Pada request berikutnya, aplikasi dapat memeriksa session untuk memastikan bahwa pengguna sudah login.

- 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  }));
```



```
JS app.js > ...
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.

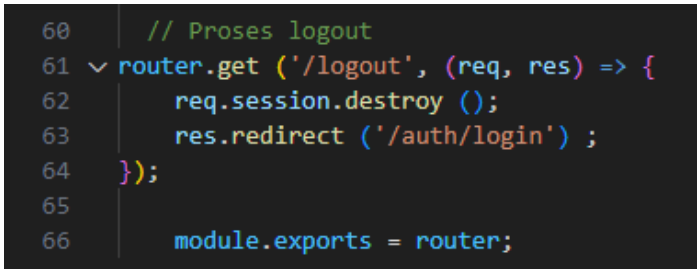
2. penjelasan auth.js

auth.js

routes > auth.js > router.post('/register') callback > query

```
const express = require ('express') ;
const router = express.Router () ;
const bcrypt = require ('bcryptjs') ;
const db = require ('../config/db') ;

// Render halaman register
router.get ('/register', (req, res) => {
    res.render('register');
}) ;

// Proses register user
router.post ('/register', (req, res) => {
    const { Username, Email, Phone, Instagram, Password}= req.body;

    const hashedPassword = bcrypt.hashSync (Password, 10) ;

    const query = "INSERT INTO users (Username, Email, Phone, Instagram, Password) VALUES (?,?,?,?,?) ";
    db.query (query, [Username, Email, Phone, Instagram, hashedPassword], (err, result) => {
        if (err) throw err;
        res.redirect ('/auth/login');
    }) ;
}) ;

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

```
// Proses login user
router.post ('/login', (req, res) => {
    const { Username, Password } = req. body;
    const query = "SELECT * FROM users WHERE Username = ?";
    db.query (query, [Username], (err, result) => {
        if (err) throw err;

        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 ('Incorrect password') ;
            }
        } else {
            res.send ('user not found' ) ;
        }
    });
});

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



```
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.

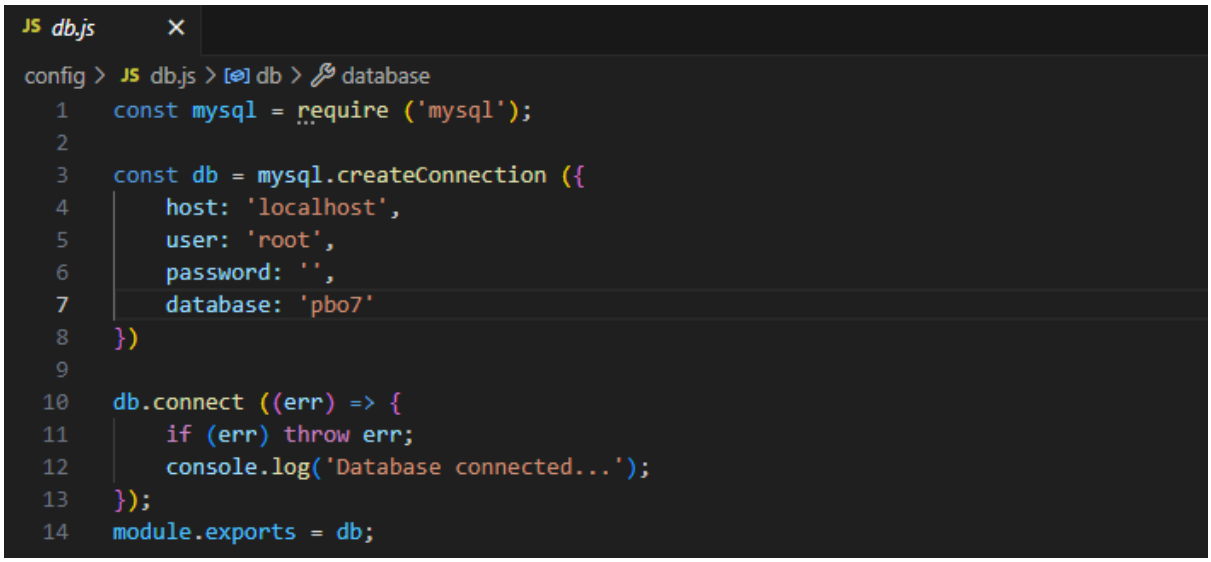
Rute Logout:

- 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



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: 'pbo7': Nama database yang digunakan adalah pbo7.

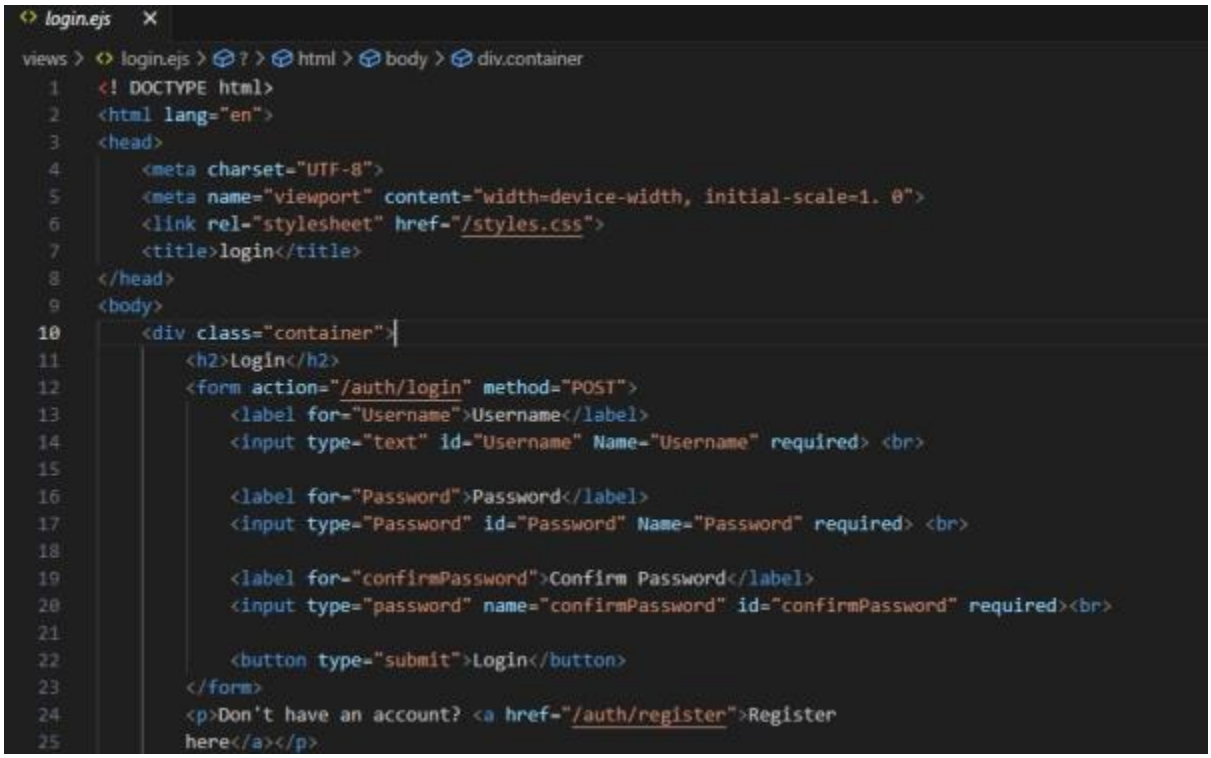
Membuka Koneksi:

- 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



```ejs
<! DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1. 0">
    <link rel="stylesheet" href="/styles.css">
    <title>login</title>
</head>
<body>
    <div class="container">
        <h2>Login</h2>
        <form action="/auth/login" method="POST">
            <label for="Username">Username</label>
            <input type="text" id="Username" Name="Username" required> <br>

            <label for="Password">Password</label>
            <input type="Password" id="Password" Name="Password" required> <br>

            <label for="confirmPassword">Confirm Password</label>
            <input type="password" name="confirmPassword" id="confirmPassword" required><br>

            <button type="submit">Login</button>
        </form>
        <p>Don't have an account? <a href="/auth/register">Register
        here</a></p>
```

```ejs
        <script>
            document.querySelector('form').addEventListener('submit', function(event) {
                const Password = document.getElementById('Password').value;
                const confirmPassword = document.getElementById('confirmPassword').value;

                if (Password !== confirmPassword) {
                    event.preventDefault(); // prevent form submission
                    alert('Passwords do not match!');
                }
            });
        </script>
    </div>
</body>
</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:

- : 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:

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

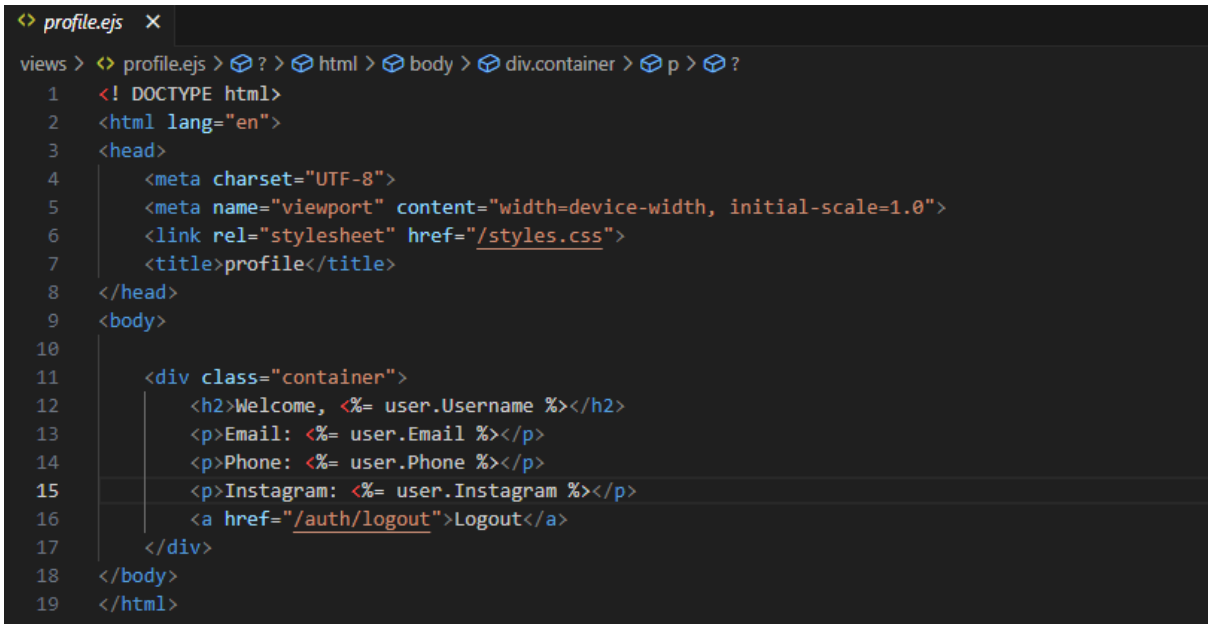
JavaScript Validation:

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

5. penjelasan profile.ejs



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".

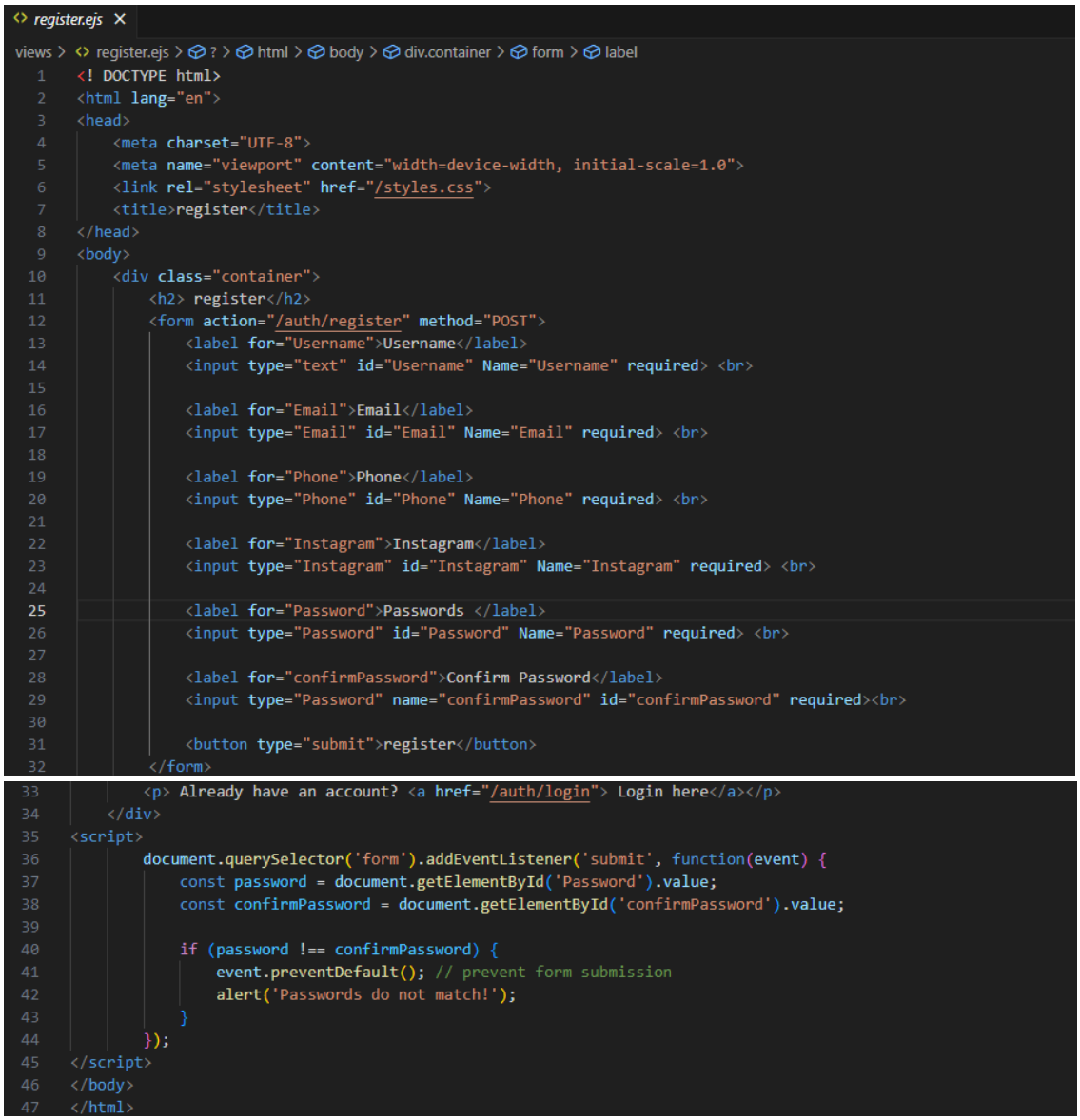
Body HTML:

- : 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.
- Email: <%= user.Email %> : Menampilkan alamat email pengguna yang diambil dari objek user dengan sintaks EJS <%= user.Email %>.
- Phone: <%= user.Phone %> : Menampilkan nomer telephone pengguna yang diambil dari objek user dengan sintaks EJS <%= user.Phone %>.
- Instagram: <%= user.Instagram %> : Menampilkan nama instagram 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 , </body>, dan </html> untuk menyelesaikan dokumen HTML.

6. penjelasan register.ejs



```ejs
<! DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/styles.css">
    <title>register</title>
</head>
<body>
    <div class="container">
        <h2> register</h2>
        <form action="/auth/register" method="POST">
            <label for="Username">Username</label>
            <input type="text" id="Username" Name="Username" required> <br>

            <label for="Email">Email</label>
            <input type="Email" id="Email" Name="Email" required> <br>

            <label for="Phone">Phone</label>
            <input type="Phone" id="Phone" Name="Phone" required> <br>

            <label for="Instagram">Instagram</label>
            <input type="Instagram" id="Instagram" Name="Instagram" required> <br>

            <label for="Password">Passwords </label>
            <input type="Password" id="Password" Name="Password" required> <br>

            <label for="confirmPassword">Confirm Password</label>
            <input type="Password" name="confirmPassword" id="confirmPassword" required><br>

            <button type="submit">register</button>
        </form>
        <p> Already have an account? <a href="/auth/login"> Login here</a></p>
    </div>
<script>
        document.querySelector('form').addEventListener('submit', function(event) {
            const password = document.getElementById('Password').value;
            const confirmPassword = document.getElementById('confirmPassword').value;

            if (password !== confirmPassword) {
                event.preventDefault(); // prevent form submission
                alert('Passwords do not match!');
            }
        });
</script>
</body>
</html>
```

HTML Structure:

- Terdapat elemen <!DOCTYPE html> yang menunjukkan bahwa dokumen ini adalah dokumen HTML5.
- Elemen <html lang="en"> menentukan bahasa dokumen sebagai bahasa Inggris.
- <head> berisi metadata seperti karakter yang digunakan (UTF-8), pengaturan tampilan responsif (viewport), dan tautan ke file CSS eksternal (/styles.css).

Body Content:

- Di dalam <body>, terdapat div dengan class container yang digunakan untuk mengatur tata letak form.
- Tag <h2> menampilkan judul "register".
- <form> membuat formulir pendaftaran yang mengirim data ke endpoint /auth/register menggunakan metode POST.

Input Fields:

- Formulir ini memiliki beberapa input yang wajib diisi, termasuk:
  - **Username:** Input teks untuk nama pengguna.
  - **Email:** Input untuk alamat email.
  - **Phone:** Input untuk nomor telepon.
  - **Instagram:** Input untuk nama akun Instagram.
  - **Password:** Input untuk kata sandi.
  - **Confirm Password:** Input untuk mengonfirmasi kata sandi.

JavaScript Validation:

- Terdapat kode JavaScript untuk memeriksa apakah kata sandi dan konfirmasi kata sandi cocok.
- Jika tidak cocok, formulir tidak akan dikirim dan menampilkan pesan peringatan "Passwords do not match!".

Footer:

- Terdapat link untuk pengguna yang sudah memiliki akun agar bisa menuju halaman login.

Styling dan Aksesibilitas:

- File CSS eksternal dihubungkan untuk menambahkan gaya pada halaman.
- Elemen input diberi atribut required untuk memastikan semua input harus diisi sebelum pengiriman formulir.

7. penjelasan styles.css

styles.css

public > # styles.css > button

```css
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    background-image: url("bg8.jpeg");
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100h;
    margin: 0;
}

.container {
    background-color: #FF8C94;
    padding: 20px;
    border-radius: 10px;
    box-shadow: #f0f0f0;
    width: 300px;
}

h2 {
    text-align: center;
    color: #994738;
}

label {
    display: block;
    color: #461d16;
    margin-bottom: 5px;
}
```

```css
input {
width: 100%;
padding: 8px;
margin-bottom: 15px;
border: 1px solid #994738;
border-radius: 5px;
}

button {
width: 100%;
padding: 10px;
background-color: #994738;
color: #eeebeb ;
border: none;
border-radius: 5px;
cursor: pointer;
}

button:hover {
    background-color: #994738;
}
```

```css
p {
    text-align: center;
    color: #461d16;
}

a {
    color: #771807;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}
```

Body Styles:

- **Font dan Background:** Menggunakan font "Arial" dan warna latar belakang #f0f0f0 dengan gambar latar bg8.jpeg.
- **Background Properties:** Mengatur ukuran gambar agar menutupi seluruh layar (cover), posisi di tengah (center), dan tidak mengulang (no-repeat).
- **Layout Flexbox:** Menggunakan Flexbox untuk menempatkan konten di tengah layar, baik secara horizontal maupun vertikal (justify-content: center dan align-items: center).
- **Height dan Margin:** Mengatur tinggi halaman (100h, seharusnya 100vh untuk tampilan penuh) dan menghapus margin default (margin: 0).

container Styles:

- **Background Color:** Mengatur warna latar belakang container menjadi #FF8C94.
- **Padding dan Border Radius:** Menambah padding sebesar 20px dan membulatkan sudut dengan radius 10px.
- **Box Shadow:** Ada properti box-shadow, tetapi tidak digunakan dengan benar (seharusnya ada nilai seperti 0px 4px 10px rgba(0, 0, 0, 0.1) untuk efek bayangan).
- **Width:** Lebar container diatur sebesar 300px.

h2 Styles:

- Mengatur teks judul agar rata tengah dengan warna #994738.

Label Styles:

- Menampilkan label sebagai blok dengan warna #461d16 dan menambahkan jarak di bawahnya.

Input Styles:

- Lebar penuh (100%), padding 8px, margin bawah 15px, border berwarna #994738, dan border-radius untuk sudut yang agak bulat.

Button Styles:

- Tombol memiliki lebar penuh, padding 10px, warna latar #994738, dan teks berwarna #eeebeb.
- Tidak ada border dan border-radius sebesar 5px.
- Mengubah kursor menjadi pointer saat diarahkan ke tombol (cursor: pointer).
- Saat di-hover, warna latar tetap sama.

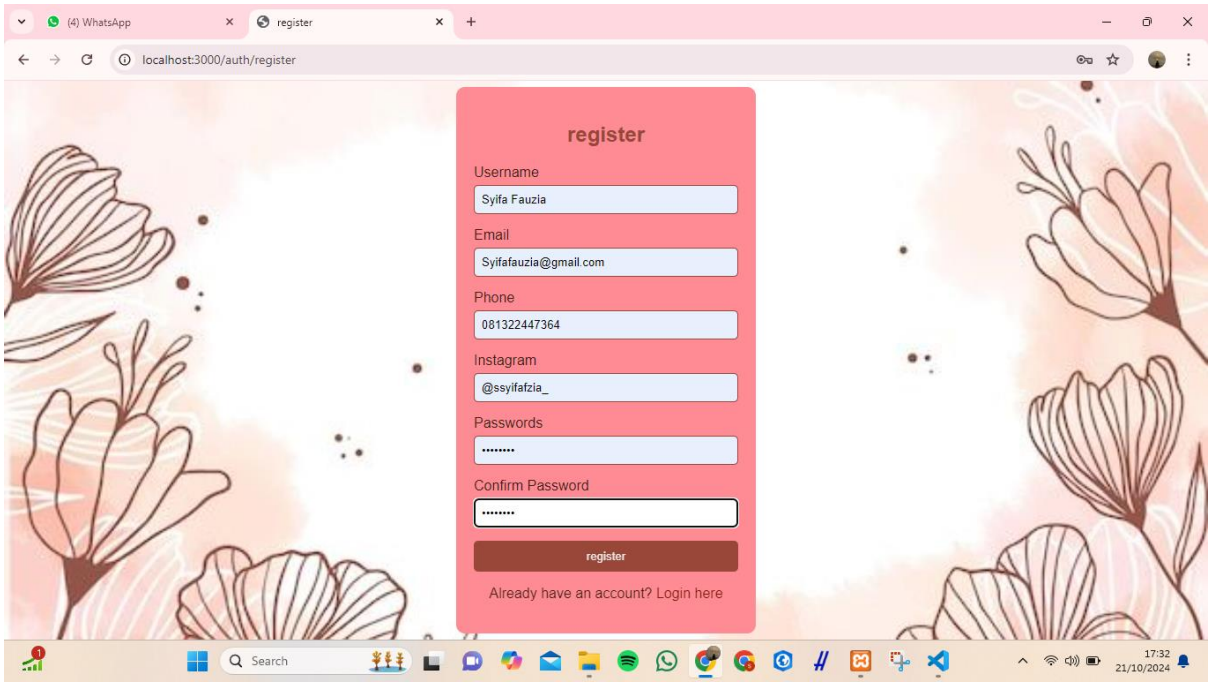
Paragraph Styles (p):

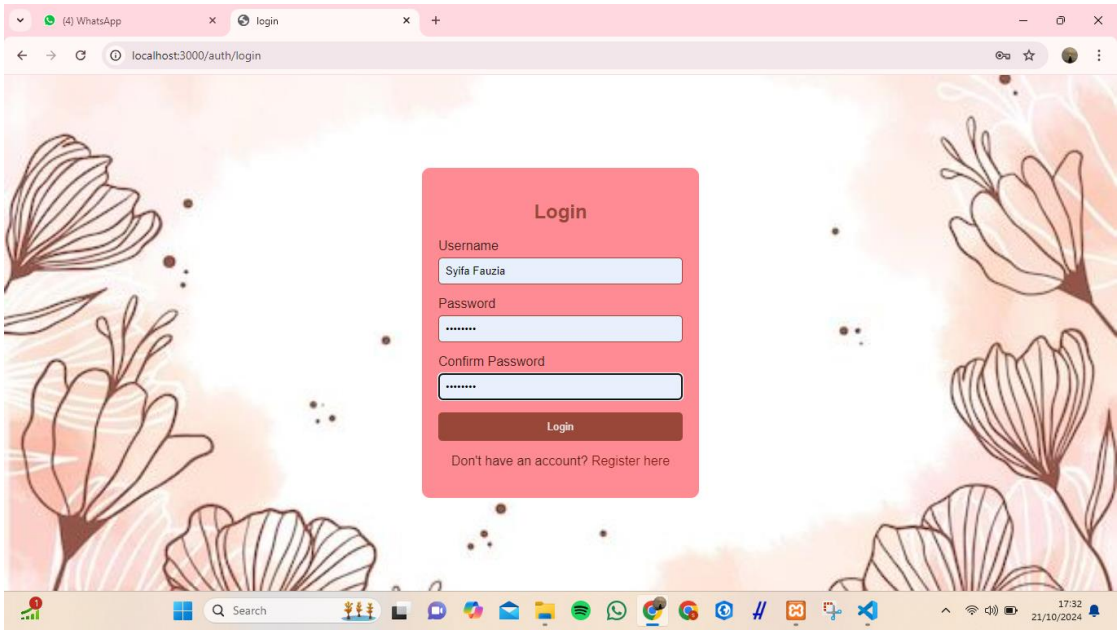
- Teks paragraf dirata tengah dan diberi warna #461d16.

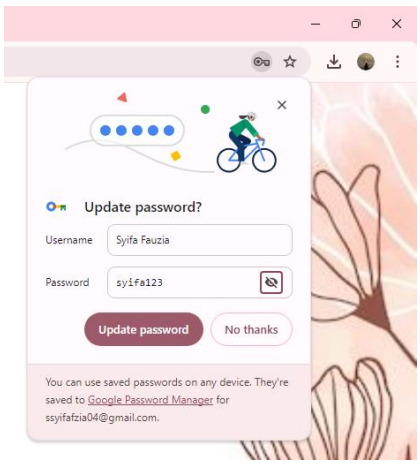
Anchor Styles (a):

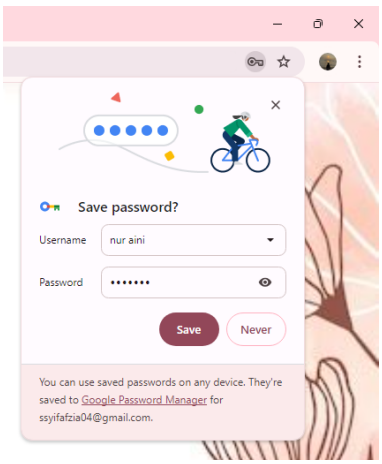
- Warna tautan diatur menjadi #771807 tanpa garis bawah.
- Saat di-hover, teks akan diberi garis bawah.

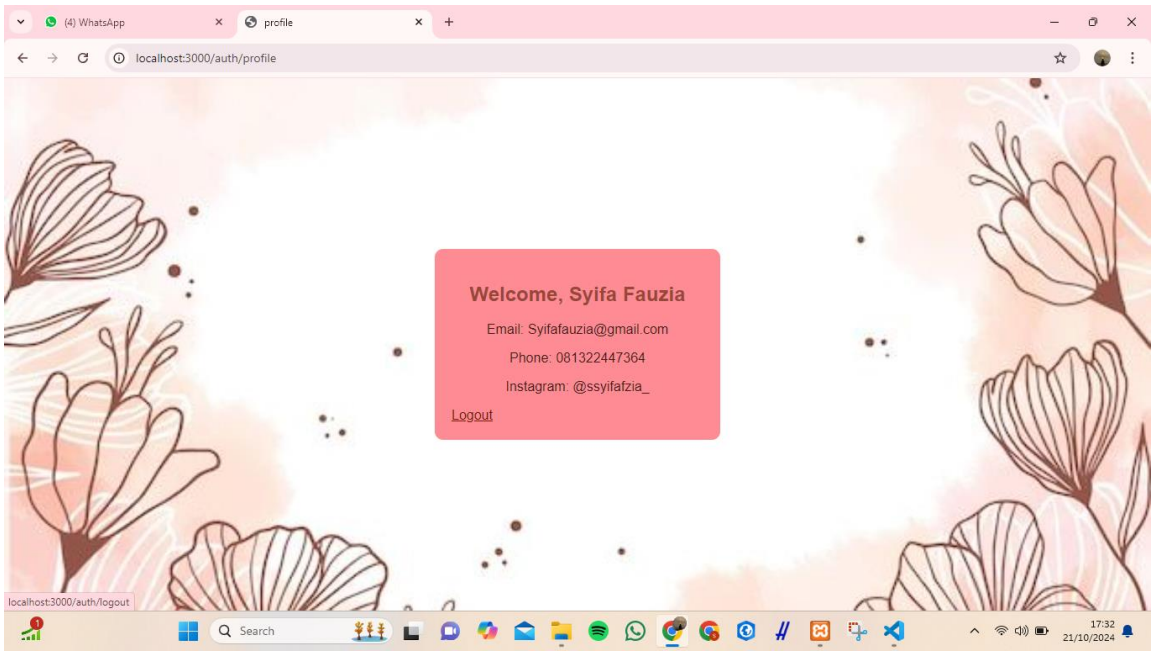
HASIL AKHIR WEB




(4) WhatsApp
login
localhost:3000/auth/login
Login
Username
Syifa Fauzia
Password
Confirm Password
Login
Don't have an account? Register here
Search
17:32
21/10/2024


Update password?
Username
Syifa Fauzia
Password
syifa123
Update password
No thanks
You can use saved passwords on any device. They're saved to Google Password Manager for ssyifafzia04@gmail.com.


Save password?
Username
nur aini
Password
Save
Never
You can use saved passwords on any device. They're saved to Google Password Manager for ssyifafzia04@gmail.com.


(4) WhatsApp
profile
localhost:3000/auth/profile
Welcome, Syifa Fauzia
Email: Syifafauzia@gmail.com
Phone: 081322447364
Instagram: @ssyifafzia_
Logout
localhost:3000/auth/logout
Search
17:32
21/10/2024

Web memiliki tiga halaman utama:

- **Halaman Register:** Untuk pengguna baru mendaftar dengan memasukkan informasi seperti username, email, phone, instagram dan password.
- **Halaman Login:** Untuk pengguna yang telah terdaftar dapat masuk dengan username dan password mereka.
- **Halaman Login (bagian atas kanan):** Untuk pengguna apabila sebelum login password ingin di save atau update password.
- **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.