

```
const { Pool } = require('pg');
require('dotenv').config();
```

```
const pool = new Pool({
  user: process.env.DB_USER || 'postgres',
  host: process.env.DB_HOST || 'localhost',
  database: process.env.DB_NAME ||
'transport_conference',
  password: process.env.DB_PASSWORD ||
'kamlu',
  port: process.env.DB_PORT || 5433,
});
```

```
module.exports = pool;
```

```
-----
-----
```

```
const pool = require('../db/index');
```

```
/******Login
******/
```

```
//
```

```
function isAdmin(req, res, next) {  
  if (req.session && req.session.user &&  
    req.session.user.roles &&  
    req.session.user.roles.includes('ADMIN')) {  
    return next(); // User logged in hai aur  
    ADMIN hai -> Aage jane do  
  }  
  // Logged-in nahi hai toh login page par  
  redirect kar do  
  res.redirect('/conference/login');  
}
```

```
// 1. Login Page open karne ke liye (GET  
Route)  
router.get('/login', (req, res) => {  
  if (req.session && req.session.user &&  
    req.session.user.roles.includes('ADMIN')) {  
    return  
    res.redirect('/conference/admin/agenda');
```

```
}  
  res.render('login', { error: null });  
});
```

**// 2. Login submit hone par (POST Route -
'app' ki jagah 'router' karein)**

```
router.post('/login', async (req, res) => {  
  const { username, password } = req.body;  
  
  try {  
    const query = `  
      SELECT u.id, u.username, r.role_name  
  
      FROM users u  
      JOIN user_roles ur ON u.id =  
ur.user_id  
      JOIN roles r ON ur.role_id = r.id  
      WHERE u.username = $1 AND  
u.password = $2  
    `;  
  
    const result = await pool.query(query,
```

```
[username, password]);
```

```
    if (result.rows.length > 0) {  
        const user = {  
            id: result.rows[0].id,  
            username: result.rows[0].username,  
            roles: result.rows.map(row =>  
row.role_name)  
        };  
    }
```

```
    req.session.user = user;
```

```
    if (user.roles.includes('ADMIN')) {  
        return  
res.redirect('/conference/admin/agenda');  
    } else {  
        return  
res.redirect('/conference/2026');  
    }  
    } else {  
        res.render('login', { error: 'Invalid  
Username or Password' });  
    }  
} catch (err) {  
    console.error(err);  
    res.status(500).send('Server Error');  
}
```

```
});
```

// 3. Logout Route

```
router.get('/logout', (req, res) => {  
  req.session.destroy(() => {  
    res.redirect('/conference/login');  
  });  
});
```