데이터베이스설계 (ICE4016)

실습 10주차

Web, Login

Prof. Wonik Choi

데이터 인텔리전스 연구실
Data Intelligence Laboratory

인하대학교
INHA UNIVERSITY

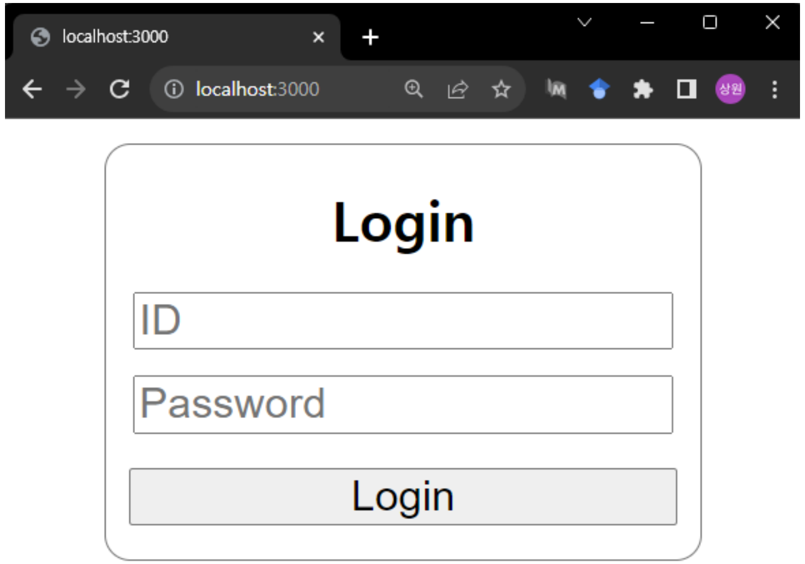
Week 10 practice

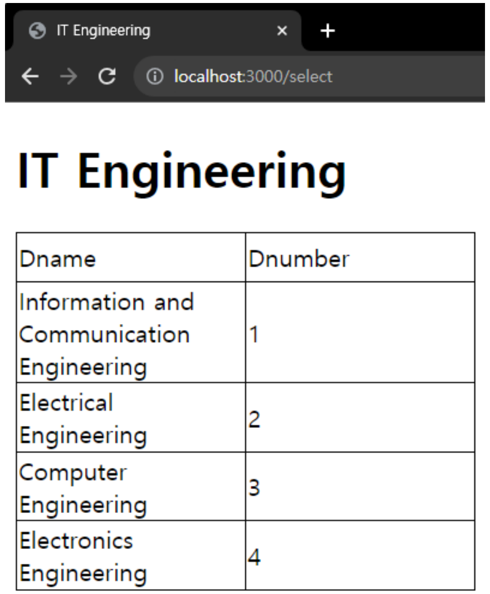
- 웹 상에서 로그인 기능 구현 및 권한 설정
  - '/'(로그인 화면), '/select' (조회 화면),
  - '/delete' (데이터 삭제 화면)

localhost:3000/
(로그인 페이지)

localhost:3000/select
(전체 데이터 조회 페이지)

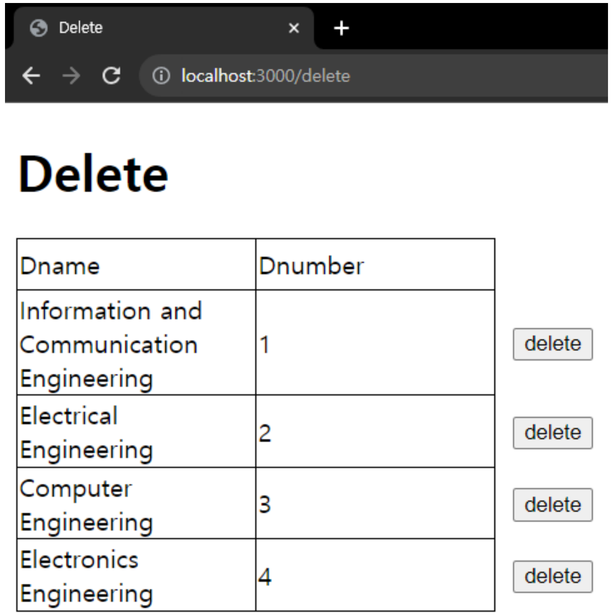
localhost:3000/delete
(데이터 선택 삭제 페이지)





IT Engineering

Dname	Dnumber
Information and Communication Engineering	1
Electrical Engineering	2
Computer Engineering	3
Electronics Engineering	4

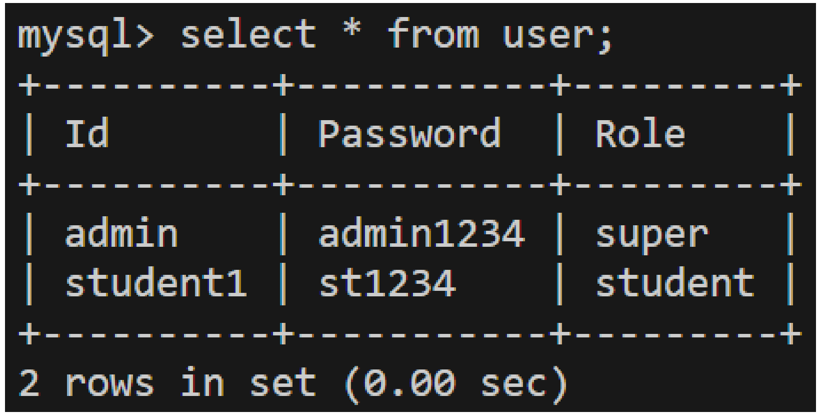


Delete

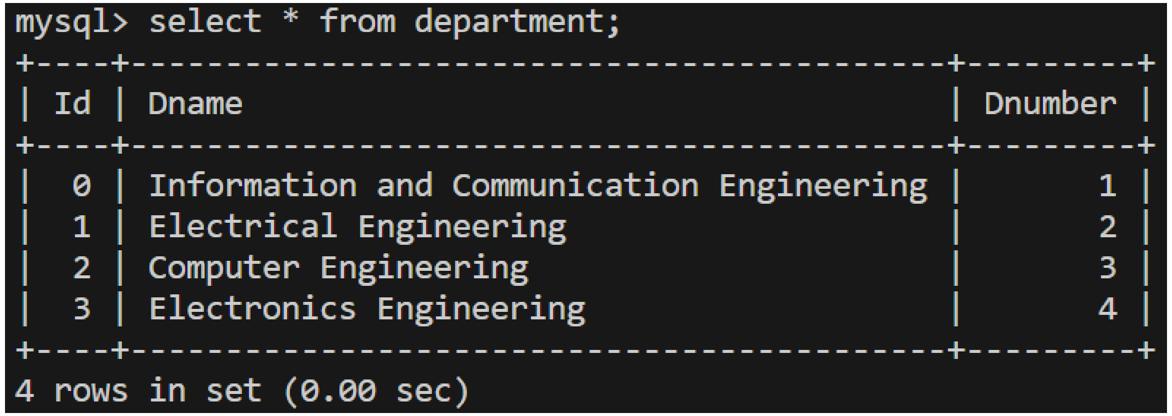
Dname	Dnumber	
Information and Communication Engineering	1	delete
Electrical Engineering	2	delete
Computer Engineering	3	delete
Electronics Engineering	4	delete

Week 10 practice

- 데이터베이스 생성 https://github.com/leesw9501/Database/tree/main/week10
  - Source week10.sql



```
mysql> select * from user;
+----------+-----------+---------+
| Id       | Password  | Role    |
+----------+-----------+---------+
| admin    | admin1234 | super   |
| student1 | st1234    | student |
+----------+-----------+---------+
2 rows in set (0.00 sec)
```

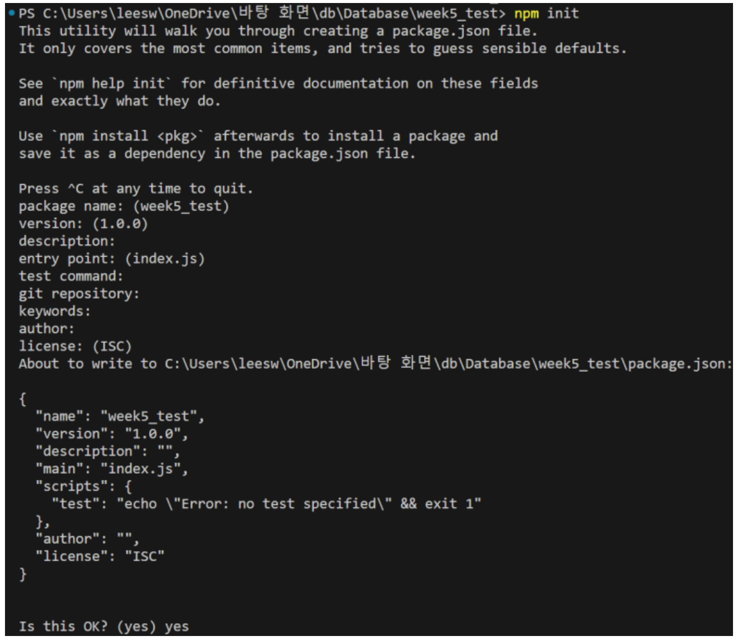


```
mysql> select * from department;
+----+----------------------------------------+---------+
| Id | Dname                                  | Dnumber |
+----+----------------------------------------+---------+
|  0 | Information and Communication Engineering |       1 |
|  1 | Electrical Engineering                 |       2 |
|  2 | Computer Engineering                   |       3 |
|  3 | Electronics Engineering                |       4 |
+----+----------------------------------------+---------+
4 rows in set (0.00 sec)
```

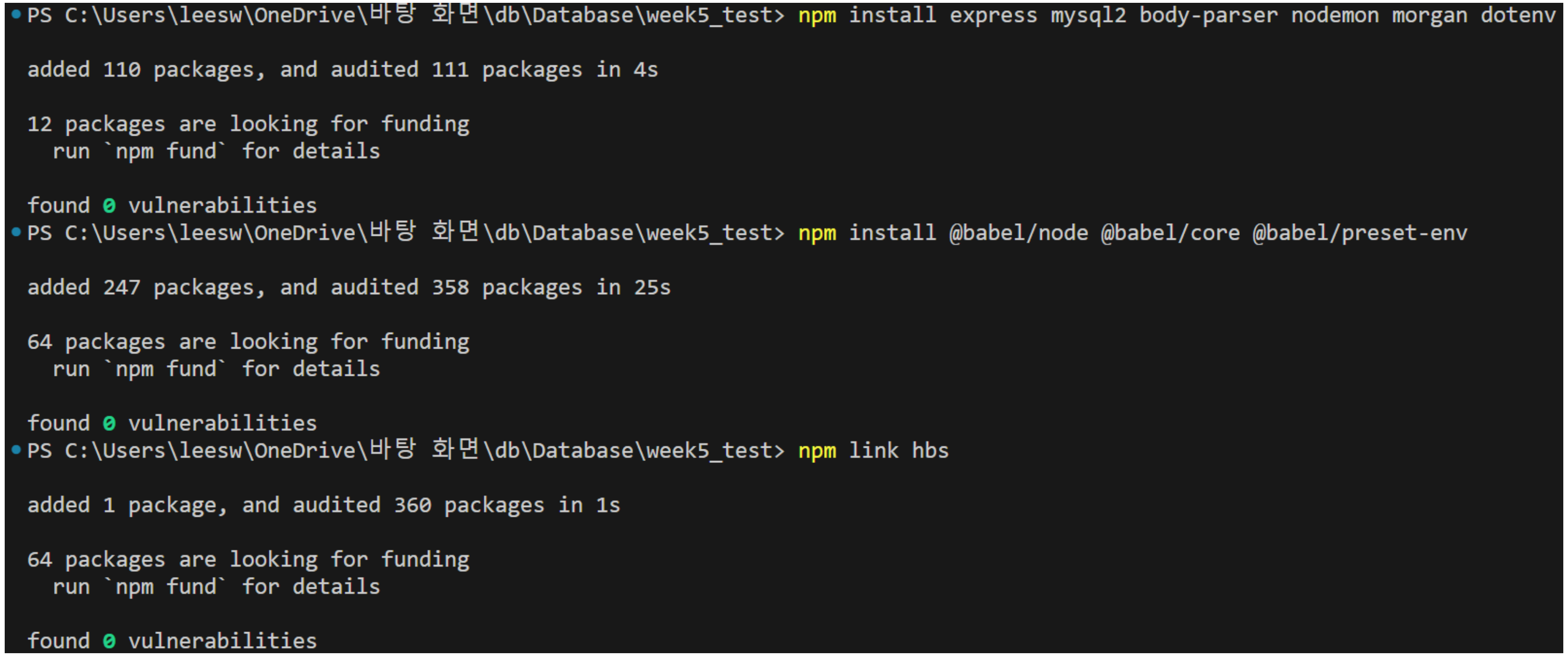
Week 10 practice

○ 프로젝트 생성 및 환경 세팅

- npm init
- npm install express mysql2 body-parser nodemon morgan dotenv express-session
- npm install @babel/node @babel/core @babel/preset-env
- npm link hbs



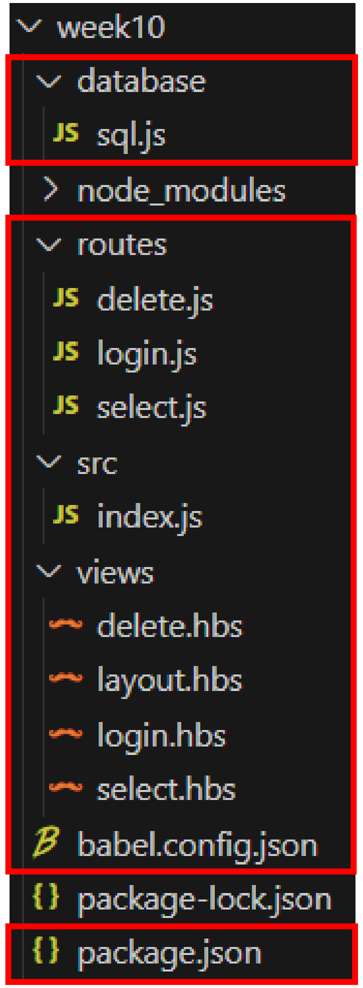
```
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week5_test> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (week5_test)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\leesw\OneDrive\바탕 화면\db\Database\week5_test\package.json:

{
  "name": "week5_test",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
```



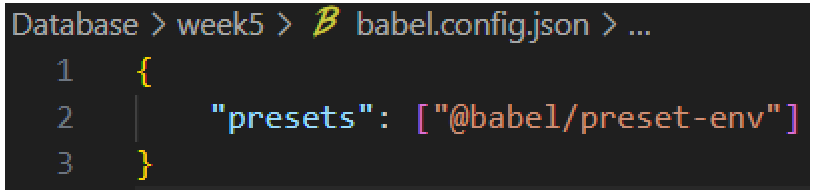
```
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week5_test> npm install express mysql2 body-parser nodemon morgan dotenv

added 110 packages, and audited 111 packages in 4s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week5_test> npm install @babel/node @babel/core @babel/preset-env

added 247 packages, and audited 358 packages in 25s

64 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\leesw\OneDrive\바탕 화면\db\Database\week5_test> npm link hbs

added 1 package, and audited 360 packages in 1s

64 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Week 10 practice

○ 프로젝트 생성 및 환경 세팅

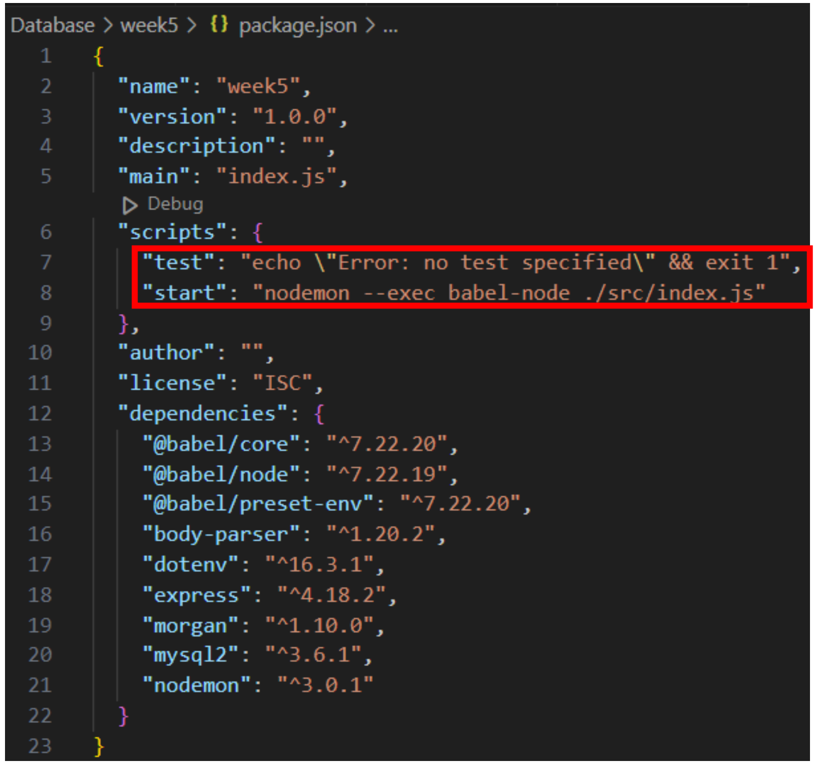


```
week10
  database
    sql.js
  node_modules
  routes
    delete.js
    login.js
    select.js
  src
    index.js
  views
    delete.hbs
    layout.hbs
    login.hbs
    select.hbs
  babel.config.json
  package-lock.json
  package.json
```



Database > week5 > babel.config.json > ...

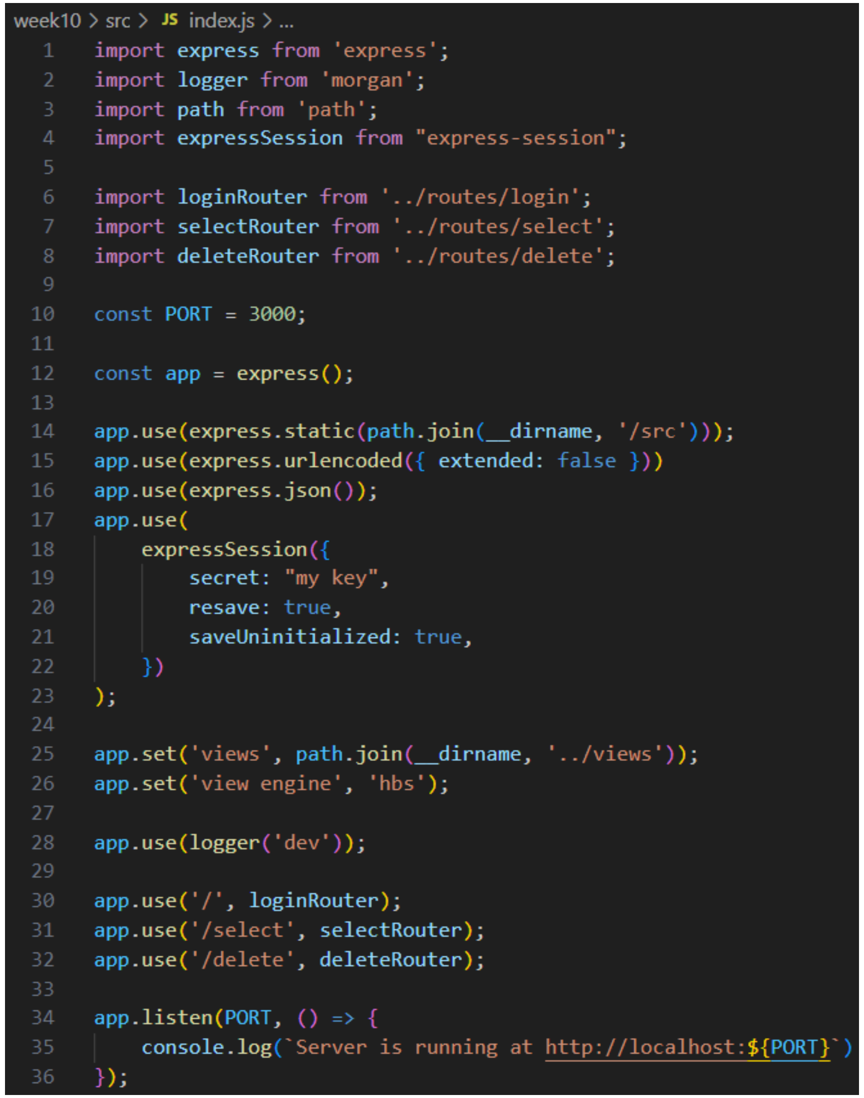
```json
{
    "presets": ["@babel/preset-env"]
}
```



Database > week5 > package.json > ...

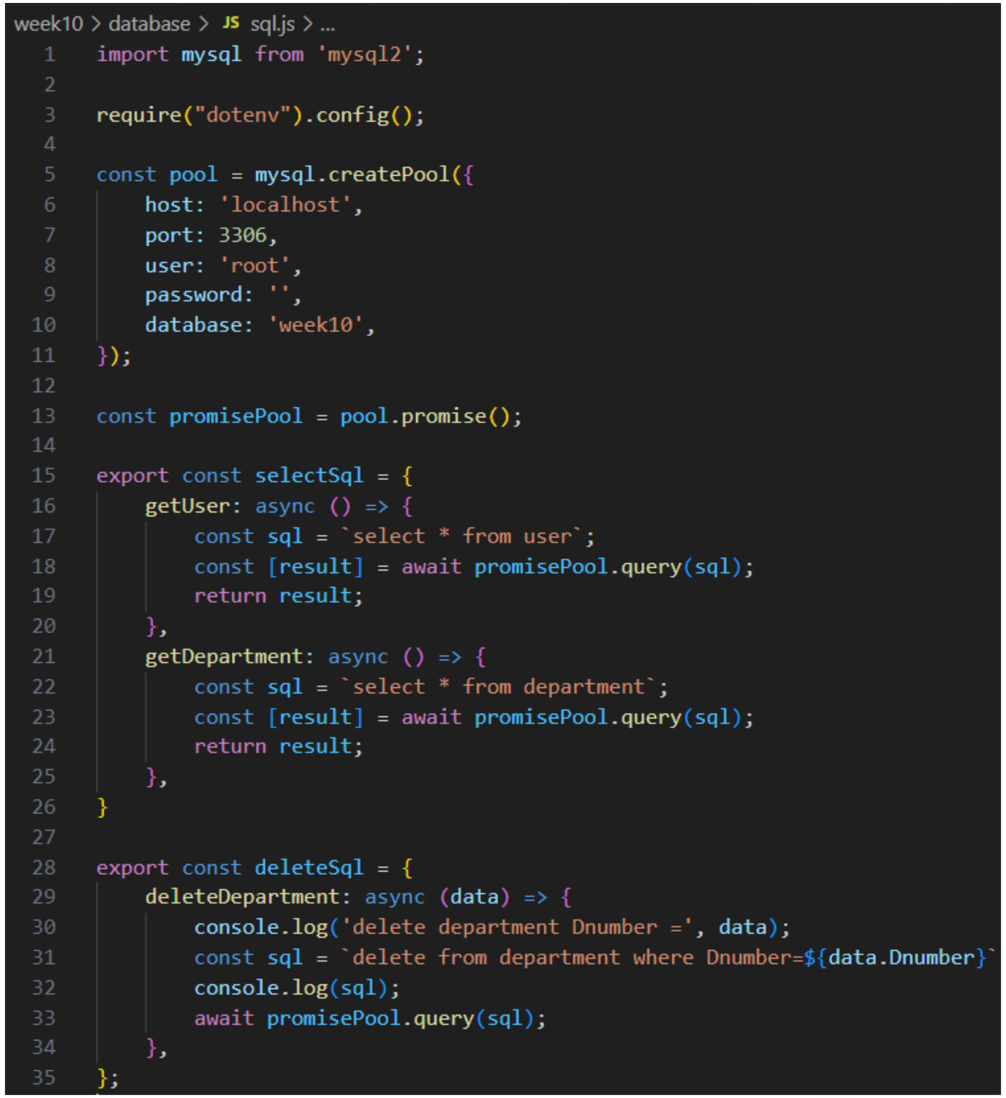
```json
{
  "name": "week5",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon --exec babel-node ./src/index.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@babel/core": "^7.22.20",
    "@babel/node": "^7.22.19",
    "@babel/preset-env": "^7.22.20",
    "body-parser": "^1.20.2",
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "morgan": "^1.10.0",
    "mysql2": "^3.6.1",
    "nodemon": "^3.0.1"
  }
}
```

Week 10 practice



week10 > src > JS index.js > ...

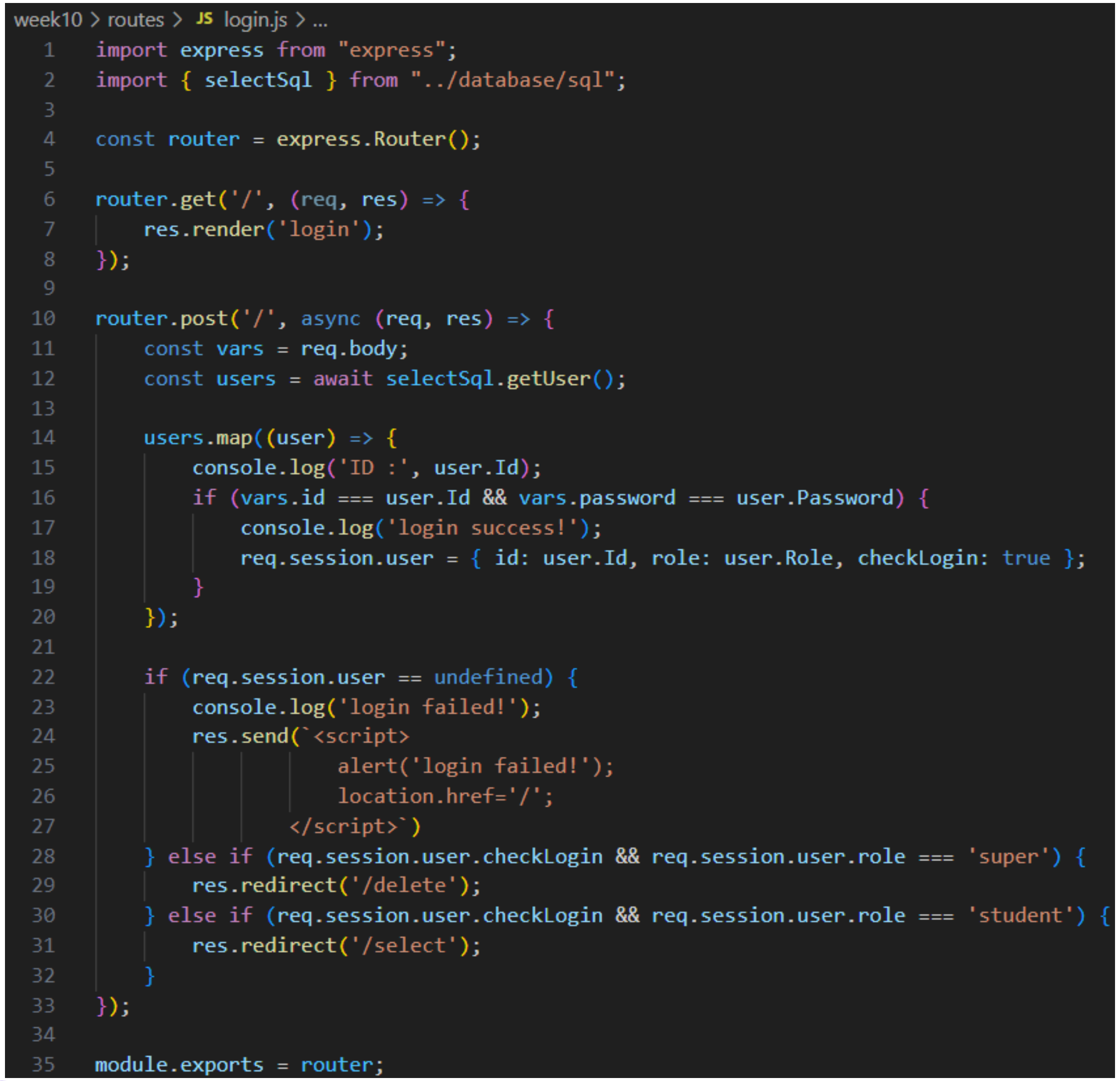
```js
import express from 'express';
import logger from 'morgan';
import path from 'path';
import expressSession from "express-session";

import loginRouter from '../routes/login';
import selectRouter from '../routes/select';
import deleteRouter from '../routes/delete';

const PORT = 3000;

const app = express();

app.use(express.static(path.join(__dirname, '/src')));
app.use(express.urlencoded({ extended: false }))
app.use(express.json());
app.use(
    expressSession({
        secret: "my key",
        resave: true,
        saveUninitialized: true,
    })
);

app.set('views', path.join(__dirname, '../views'));
app.set('view engine', 'hbs');

app.use(logger('dev'));

app.use('/', loginRouter);
app.use('/select', selectRouter);
app.use('/delete', deleteRouter);

app.listen(PORT, () => {
    console.log(`Server is running at http://localhost:${PORT}`)
});
```



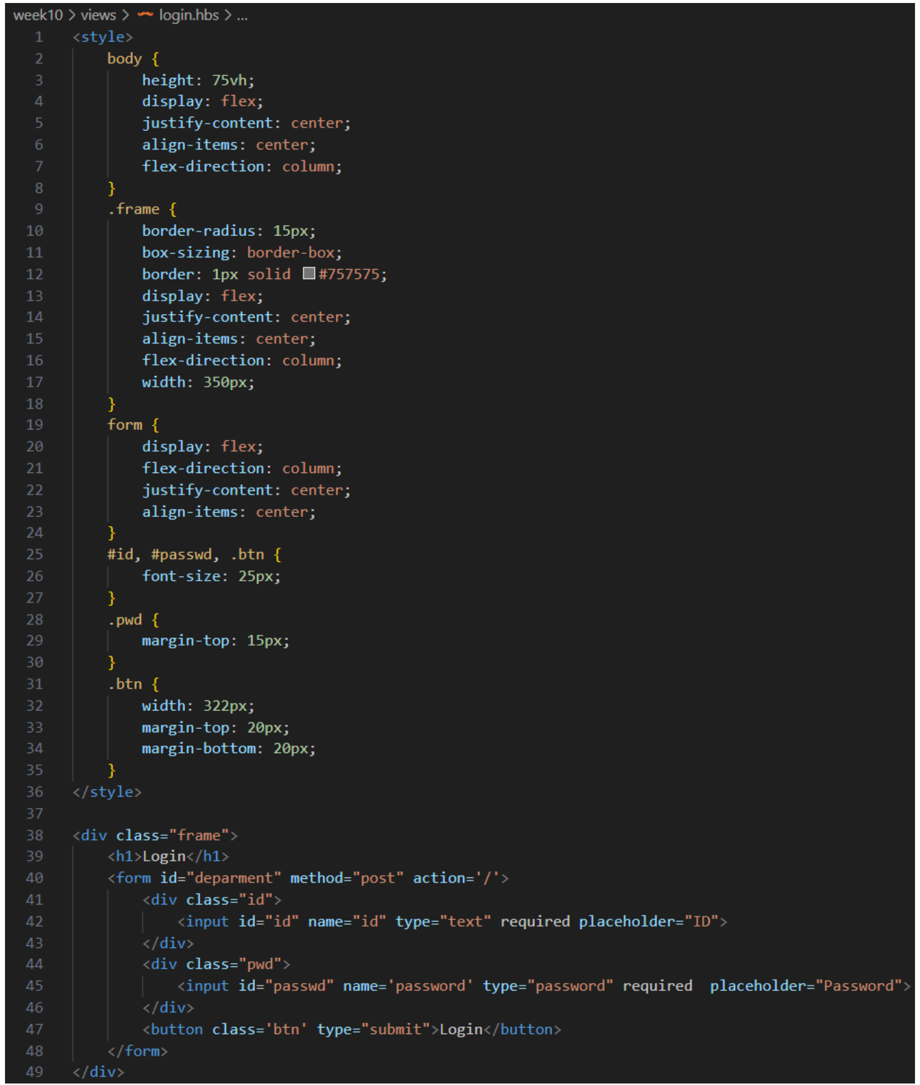
week10 > database > JS sql.js > ...

```js
import mysql from 'mysql2';

require("dotenv").config();

const pool = mysql.createPool({
    host: 'localhost',
    port: 3306,
    user: 'root',
    password: '',
    database: 'week10',
});

const promisePool = pool.promise();

export const selectSql = {
    getUser: async () => {
        const sql = `select * from user`;
        const [result] = await promisePool.query(sql);
        return result;
    },
    getDepartment: async () => {
        const sql = `select * from department`;
        const [result] = await promisePool.query(sql);
        return result;
    },
}

export const deleteSql = {
    deleteDepartment: async (data) => {
        console.log('delete department Dnumber =', data);
        const sql = `delete from department where Dnumber=${data.Dnumber}`
        console.log(sql);
        await promisePool.query(sql);
    },
};
```

Week 10 practice

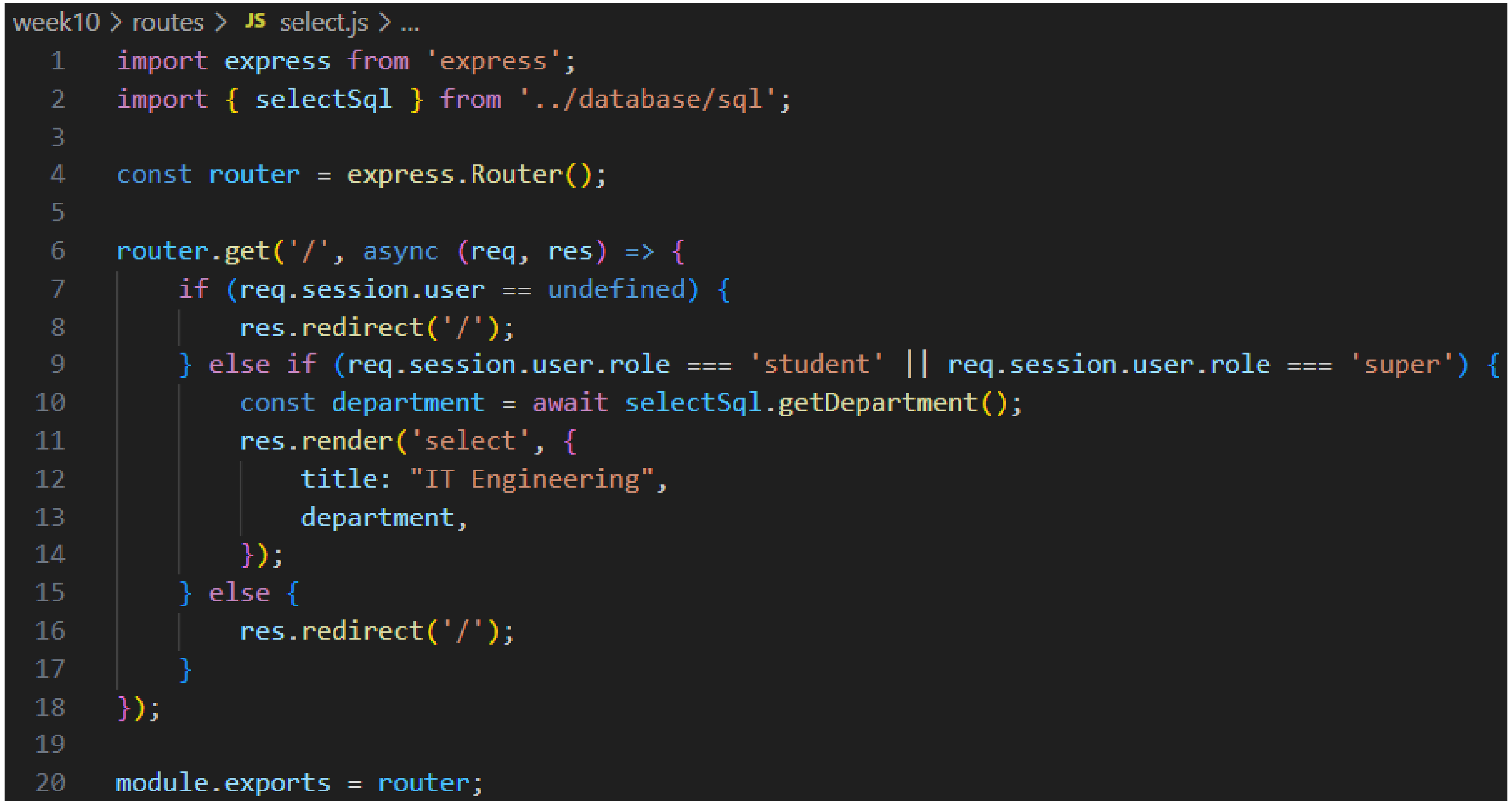


```javascript
import express from "express";
import { selectSql } from "../database/sql";

const router = express.Router();

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

router.post('/', async (req, res) => {
    const vars = req.body;
    const users = await selectSql.getUser();

    users.map((user) => {
        console.log('ID :', user.Id);
        if (vars.id === user.Id && vars.password === user.Password) {
            console.log('login success!');
            req.session.user = { id: user.Id, role: user.Role, checkLogin: true };
        }
    });

    if (req.session.user == undefined) {
        console.log('login failed!');
        res.send(`<script>
                    alert('login failed!');
                    location.href='/';
                </script>`)
    } else if (req.session.user.checkLogin && req.session.user.role === 'super') {
        res.redirect('/delete');
    } else if (req.session.user.checkLogin && req.session.user.role === 'student') {
        res.redirect('/select');
    }
});

module.exports = router;
```

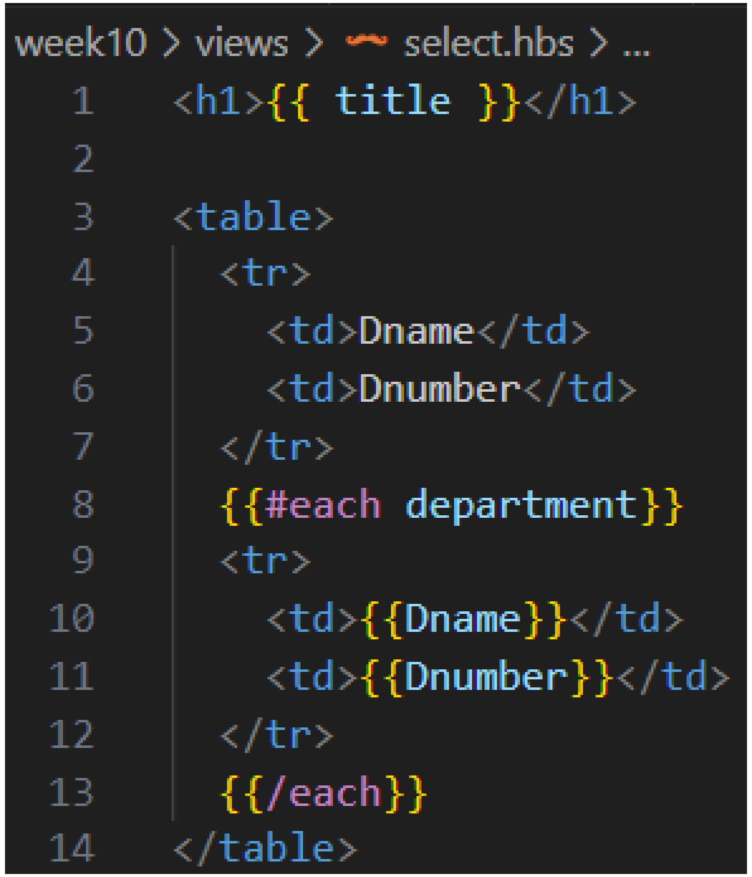


```html
<style>
    body {
        height: 75vh;
        display: flex;
        justify-content: center;
        align-items: center;
        flex-direction: column;
    }
    .frame {
        border-radius: 15px;
        box-sizing: border-box;
        border: 1px solid #757575;
        display: flex;
        justify-content: center;
        align-items: center;
        flex-direction: column;
        width: 350px;
    }
    form {
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
    }
    #id, #passwd, .btn {
        font-size: 25px;
    }
    .pwd {
        margin-top: 15px;
    }
    .btn {
        width: 322px;
        margin-top: 20px;
        margin-bottom: 20px;
    }
</style>

<div class="frame">
    <h1>Login</h1>
    <form id="deparment" method="post" action='/'>
        <div class="id">
            <input id="id" name="id" type="text" required placeholder="ID">
        </div>
        <div class="pwd">
            <input id="passwd" name='password' type="password" required  placeholder="Password">
        </div>
        <button class='btn' type="submit">Login</button>
    </form>
</div>
```

Week 10 practice

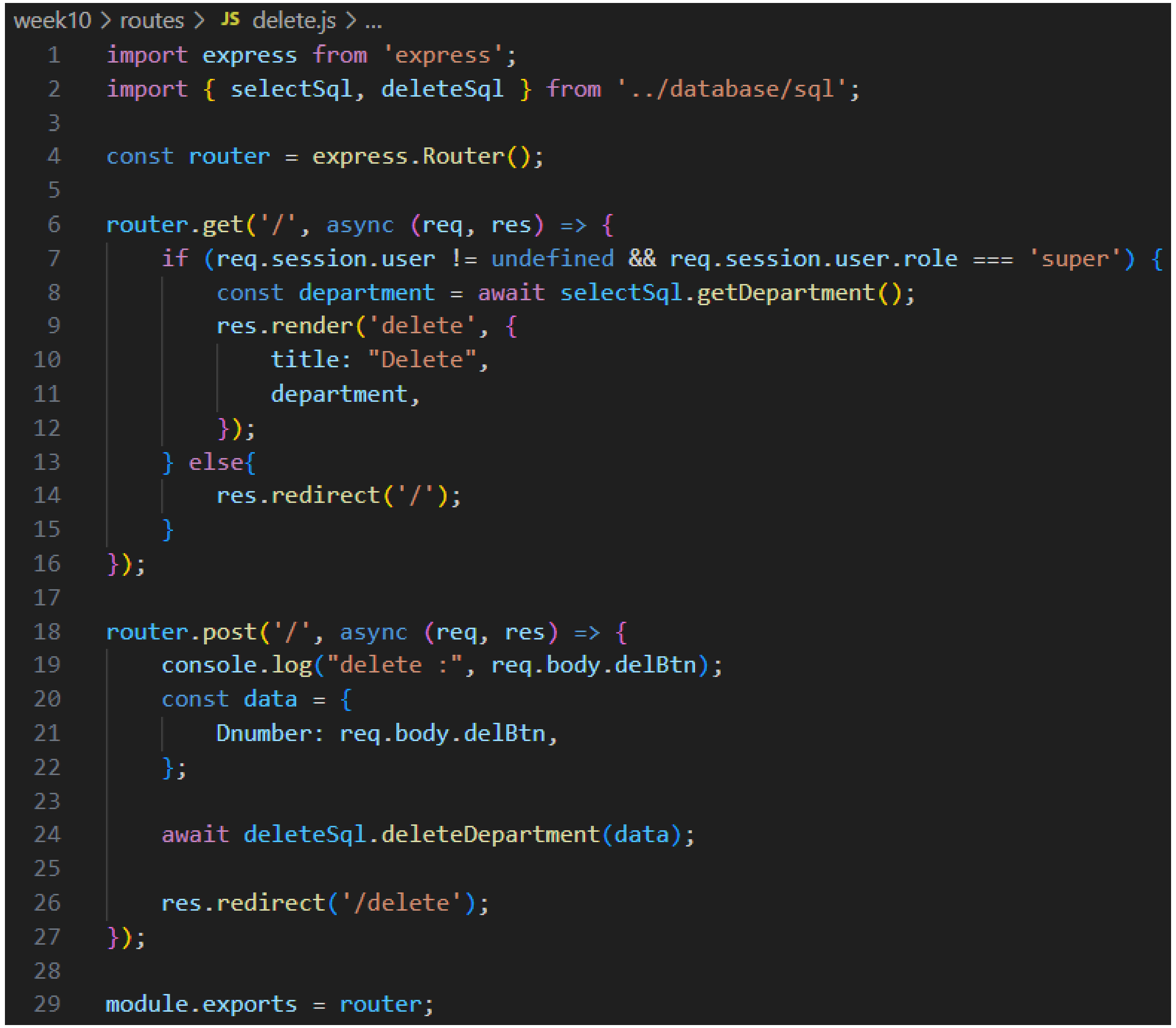


```javascript
import express from 'express';
import { selectSql } from '../database/sql';

const router = express.Router();

router.get('/', async (req, res) => {
    if (req.session.user == undefined) {
        res.redirect('/');
    } else if (req.session.user.role === 'student' || req.session.user.role === 'super') {
        const department = await selectSql.getDepartment();
        res.render('select', {
            title: "IT Engineering",
            department,
        });
    } else {
        res.redirect('/');
    }
});

module.exports = router;
```



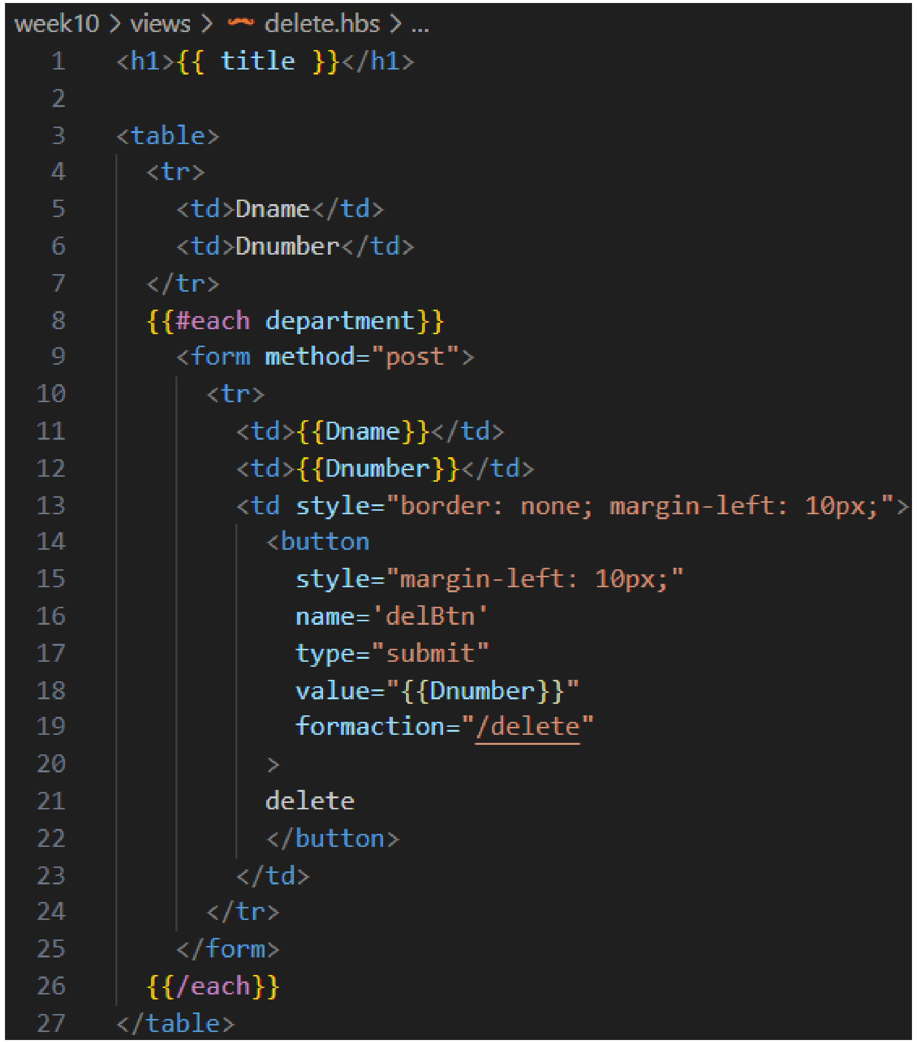
```handlebars
<h1>{{ title }}</h1>

<table>
  <tr>
    <td>Dname</td>
    <td>Dnumber</td>
  </tr>
  {{#each department}}
  <tr>
    <td>{{Dname}}</td>
    <td>{{Dnumber}}</td>
  </tr>
  {{/each}}
</table>
```

Week 10 practice



week10 > routes > JS delete.js > ...

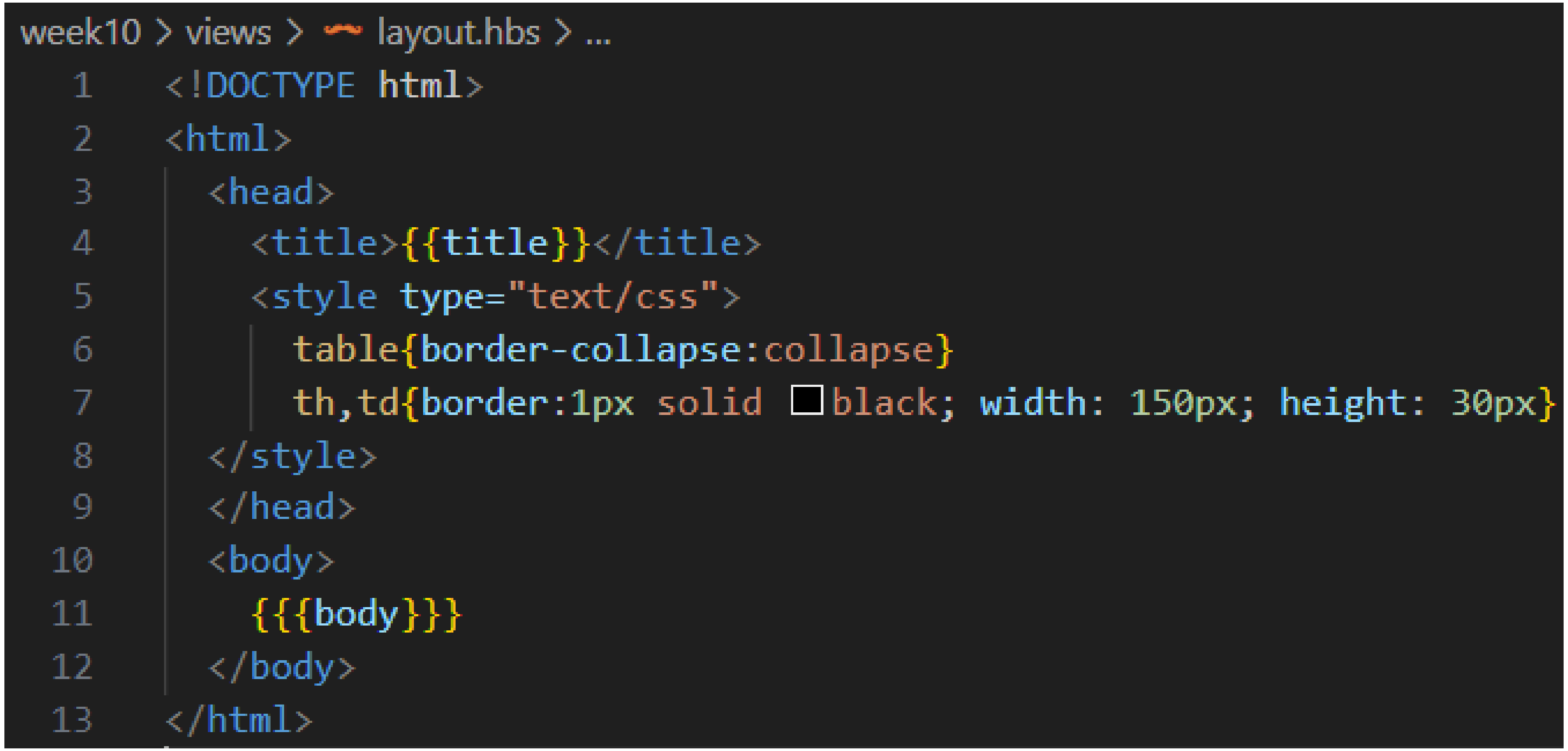
```javascript
import express from 'express';
import { selectSql, deleteSql } from '../database/sql';

const router = express.Router();

router.get('/', async (req, res) => {
    if (req.session.user != undefined && req.session.user.role === 'super') {
        const department = await selectSql.getDepartment();
        res.render('delete', {
            title: "Delete",
            department,
        });
    } else{
        res.redirect('/');
    }
});

router.post('/', async (req, res) => {
    console.log("delete :", req.body.delBtn);
    const data = {
        Dnumber: req.body.delBtn,
    };

    await deleteSql.deleteDepartment(data);

    res.redirect('/delete');
});

module.exports = router;
```



week10 > views > delete.hbs > ...

```handlebars
<h1>{{ title }}</h1>

<table>
  <tr>
    <td>Dname</td>
    <td>Dnumber</td>
  </tr>
  {{#each department}}
    <form method="post">
      <tr>
        <td>{{Dname}}</td>
        <td>{{Dnumber}}</td>
        <td style="border: none; margin-left: 10px;">
          <button
            style="margin-left: 10px;"
            name='delBtn'
            type="submit"
            value="{{Dnumber}}"
            formaction="/delete"
          >
          delete
          </button>
        </td>
      </tr>
    </form>
  {{/each}}
</table>
```

Week 10 practice



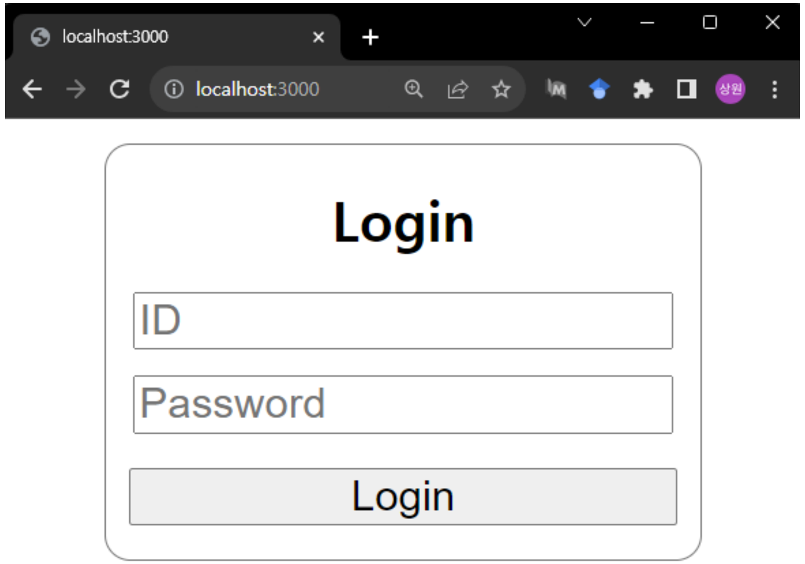
week10 > views > layout.hbs > ...

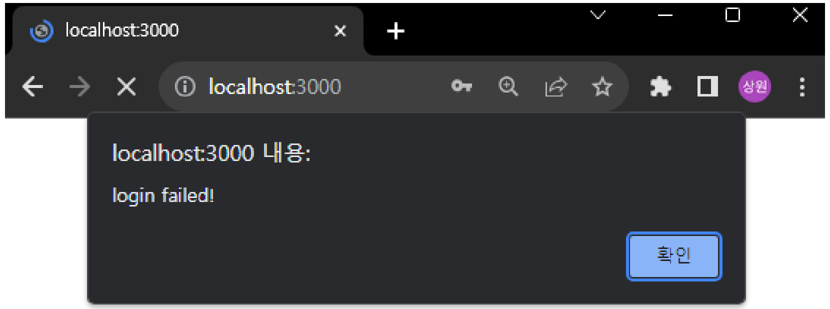
```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <style type="text/css">
      table{border-collapse:collapse}
      th,td{border:1px solid black; width: 150px; height: 30px}
  </style>
  </head>
  <body>
    {{{body}}}
  </body>
</html>
```

Week 10 practice

○ 서버 실행

- 콘솔창에서 “npm run start” 명령어로 서버 실행하여 기능 확인
  - 웹 브라우저에서 localhost:3000 접속
    - ➢ 로그인 화면 확인
      - ▪ DB에 저장된 유저 정보 확인
      - ▪ 로그인 성공 시 화면 이동
      - ▪ 로그인 실패 시 경고 창 팝업





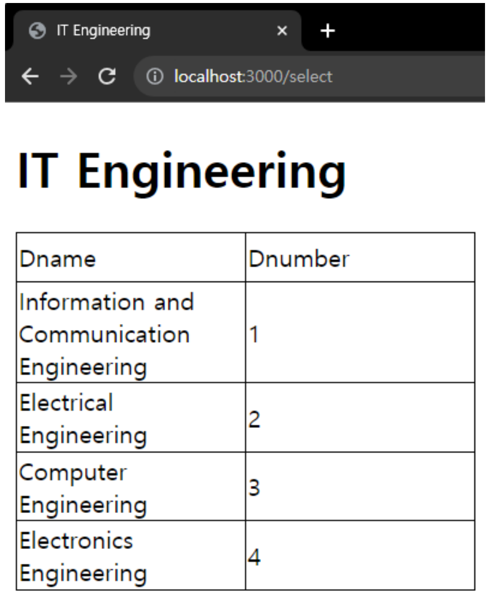
localhost:3000 내용:

login failed!

Week 10 practice

○ 서버 실행

- 콘솔창에서 “npm run start” 명령어로 서버 실행하여 기능 확인
  - 웹 브라우저에서 localhost:3000/select 접속
    - ➢ Department 테이블 데이터 조회

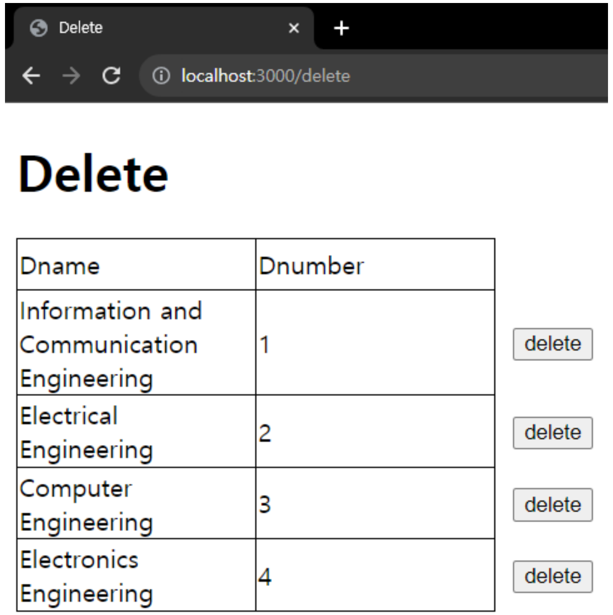


IT Engineering

Dname	Dnumber
Information and Communication Engineering	1
Electrical Engineering	2
Computer Engineering	3
Electronics Engineering	4

Week 10 practice

- 서버 실행
  - 콘솔창에서 "npm run start" 명령어로 서버 실행하여 기능 확인
    - 웹 브라우저에서 localhost:3000/delete 접속
      - ➢ 삭제 버튼을 클릭하여 해당 데이터 삭제



Delete

Dname	Dnumber	
Information and Communication Engineering	1	delete
Electrical Engineering	2	delete
Computer Engineering	3	delete
Electronics Engineering	4	delete

Week 10 Assignment

- STEP 1 : 7주차 Inha DB 웹에 로그인 기능 적용
  - 학생의 학번 및 비밀번호 (전화번호)로 로그인 하는 기능 구현할 것
    - 학생은 학생 정보를 조회할 수 있는 페이지로 이동

- STEP 2 : localhost:3000/delete/class 페이지 만들기
  - 학생으로 로그인 한 경우에만 수강하는 수업(class)을 수강취소(삭제) 할 수 있도록 구현

- 소스코드 제출 안해도 됨, 보고서 작성하여 제출
  - 로그인, 삭제 기능 관련 캡처 및 분석