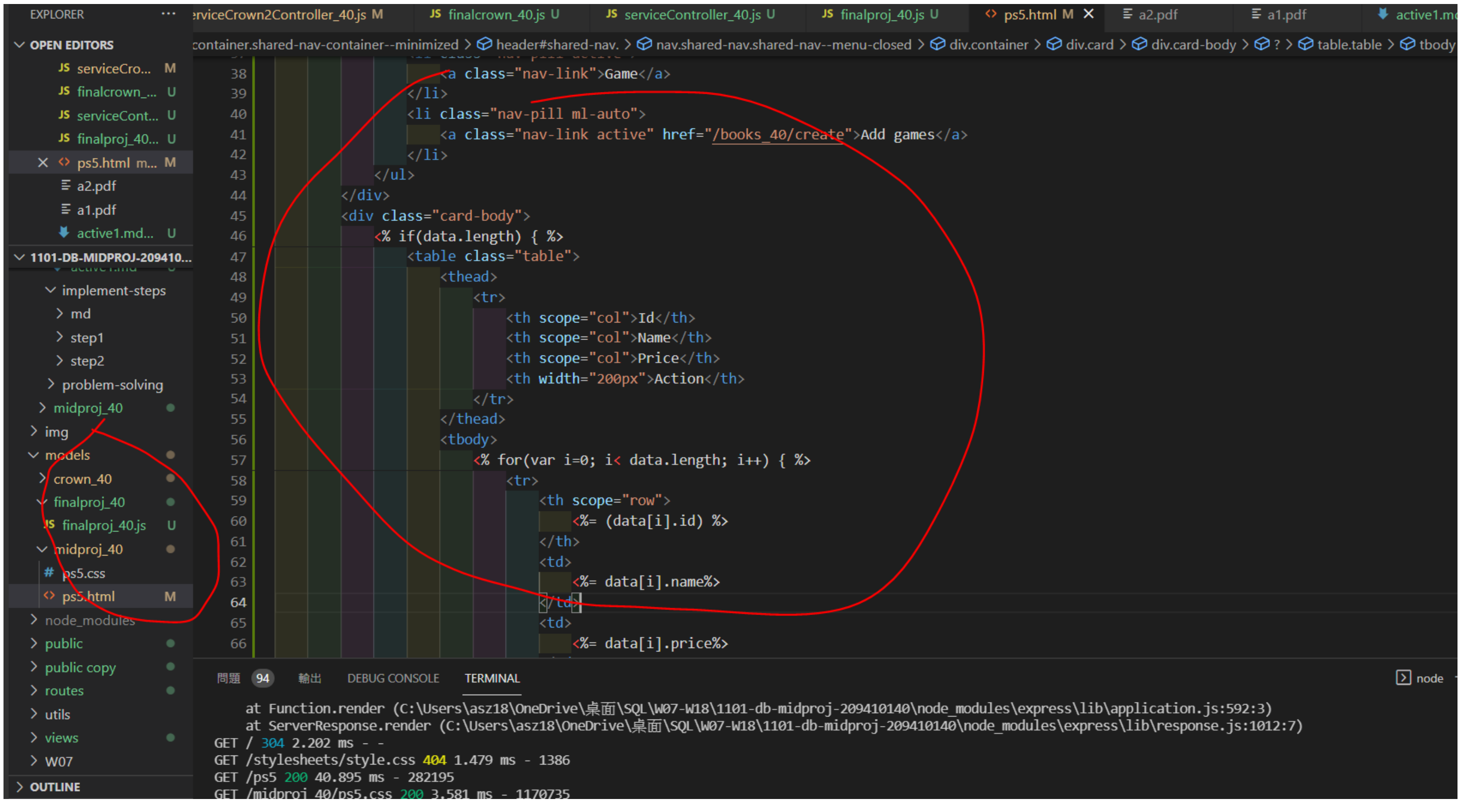
Active 1

嘗試加入CRUD的html介面，但弄了很久卻還是以失敗告終。

我是參考老師的上課教材來新增的，但成果不盡人意。

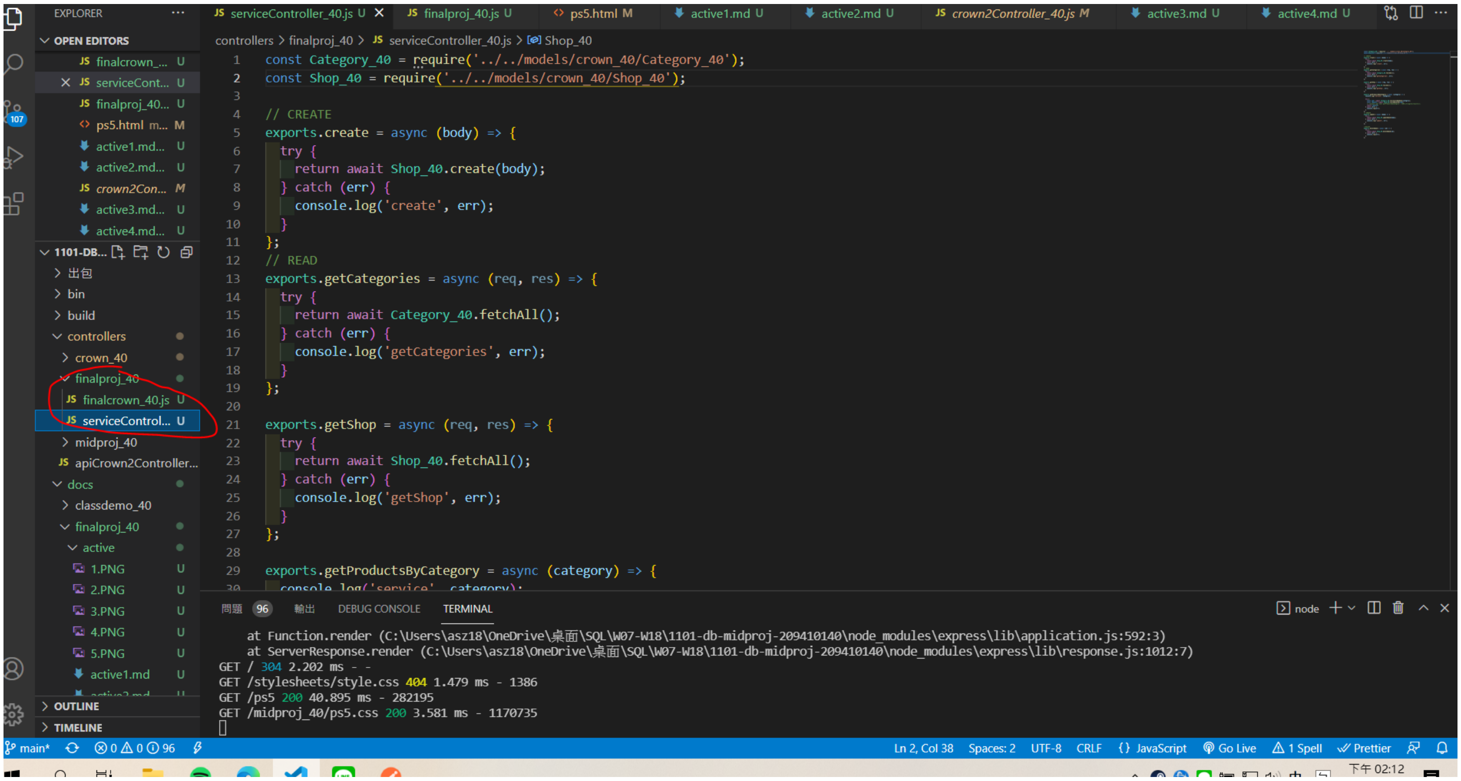


```
            <a class="nav-link">Game</a>
        </li>
        <li class="nav-pill ml-auto">
            <a class="nav-link active" href="/books_40/create">Add games</a>
        </li>
    </ul>
</div>
<div class="card-body">
    <% if(data.length) { %>
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">Id</th>
                    <th scope="col">Name</th>
                    <th scope="col">Price</th>
                    <th width="200px">Action</th>
                </tr>
            </thead>
            <tbody>
                <% for(var i=0; i< data.length; i++) { %>
                    <tr>
                        <th scope="row">
                            <%= (data[i].id) %>
                        </th>
                        <td>
                            <%= data[i].name%>
                        </td>
                        <td>
                            <%= data[i].price%>
```

```
    at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
    at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```

Active2

把controllers等const設定好，雖然沒弄好，但起碼有試過也跑過流程，加深印象。

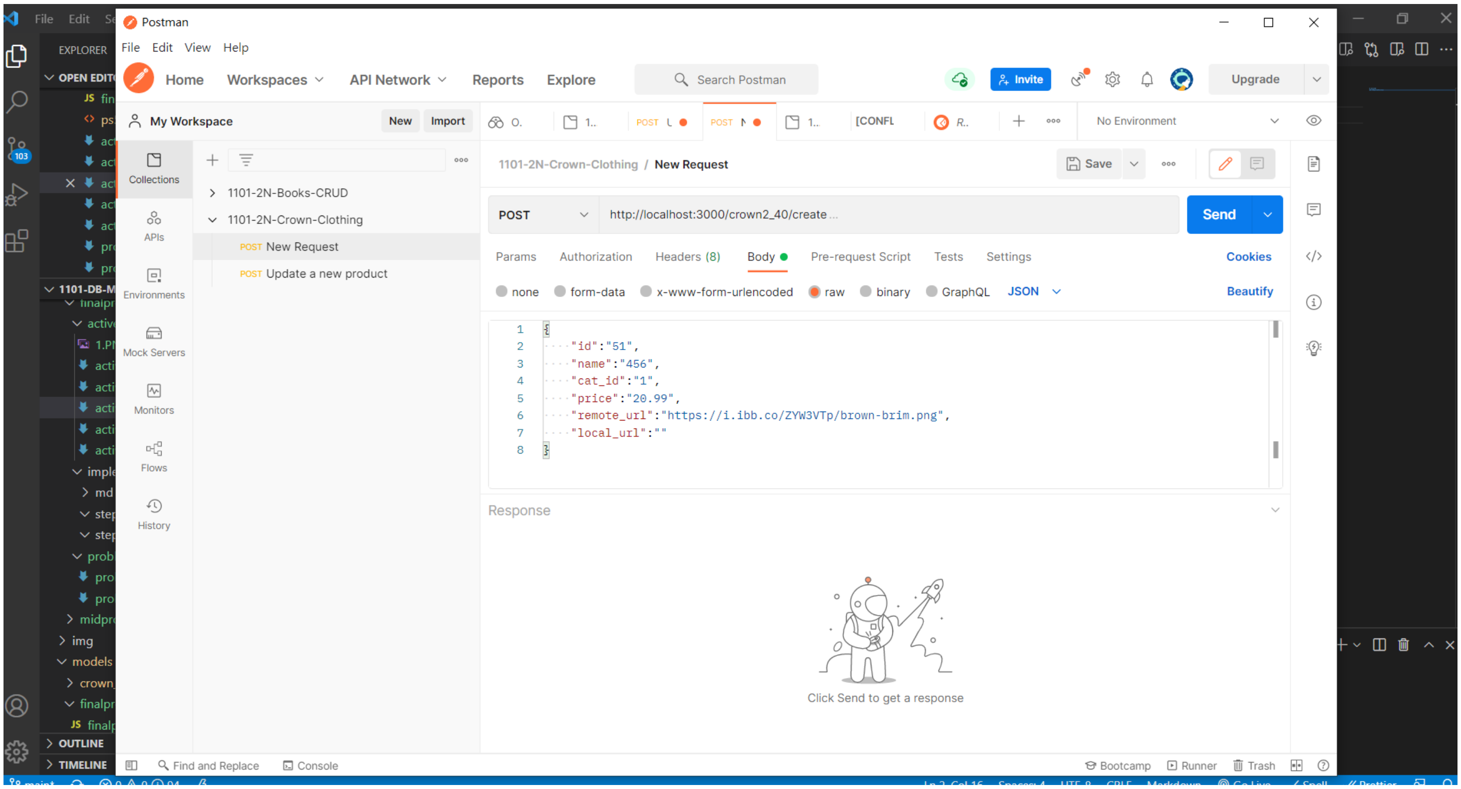


```javascript
const Category_40 = require('../../models/crown_40/Category_40');
const Shop_40 = require('../../models/crown_40/Shop_40');

// CREATE
exports.create = async (body) => {
  try {
    return await Shop_40.create(body);
  } catch (err) {
    console.log('create', err);
  }
};
// READ
exports.getCategories = async (req, res) => {
  try {
    return await Category_40.fetchAll();
  } catch (err) {
    console.log('getCategories', err);
  }
};

exports.getShop = async (req, res) => {
  try {
    return await Shop_40.fetchAll();
  } catch (err) {
    console.log('getShop', err);
  }
};

exports.getProductsByCategory = async (category) => {
  console.log('service', category);
```

```
    at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
    at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```

Active3

重新研究Postman，想知道它的運作邏輯，但也不確定自己有沒有看懂。



POST http://localhost:3000/crown2_40/create

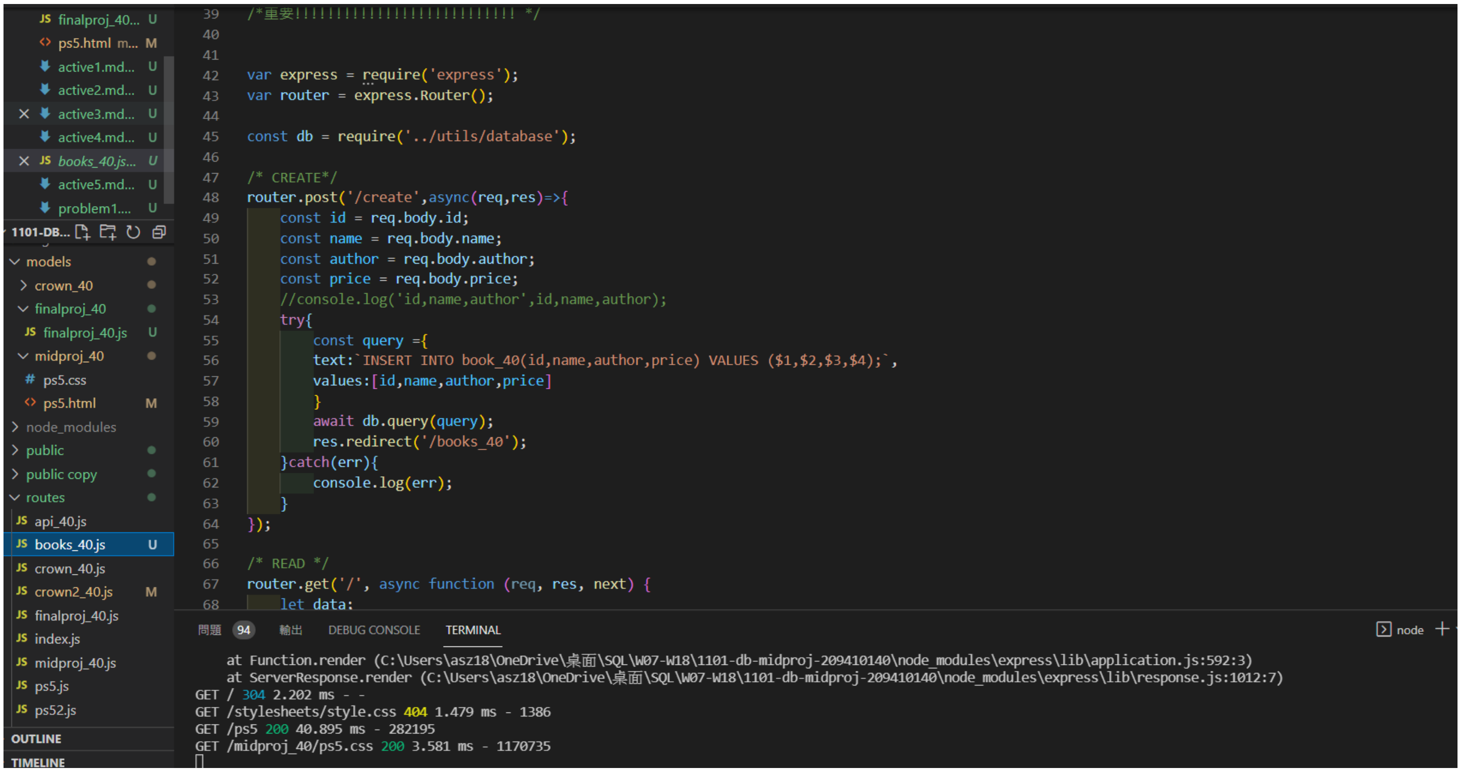
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```json
{
    "id":"51",
    "name":"456",
    "cat_id":"1",
    "price":"20.99",
    "remote_url":"https://i.ibb.co/ZYW3VTp/brown-brim.png",
    "local_url":""
}
```

Active4

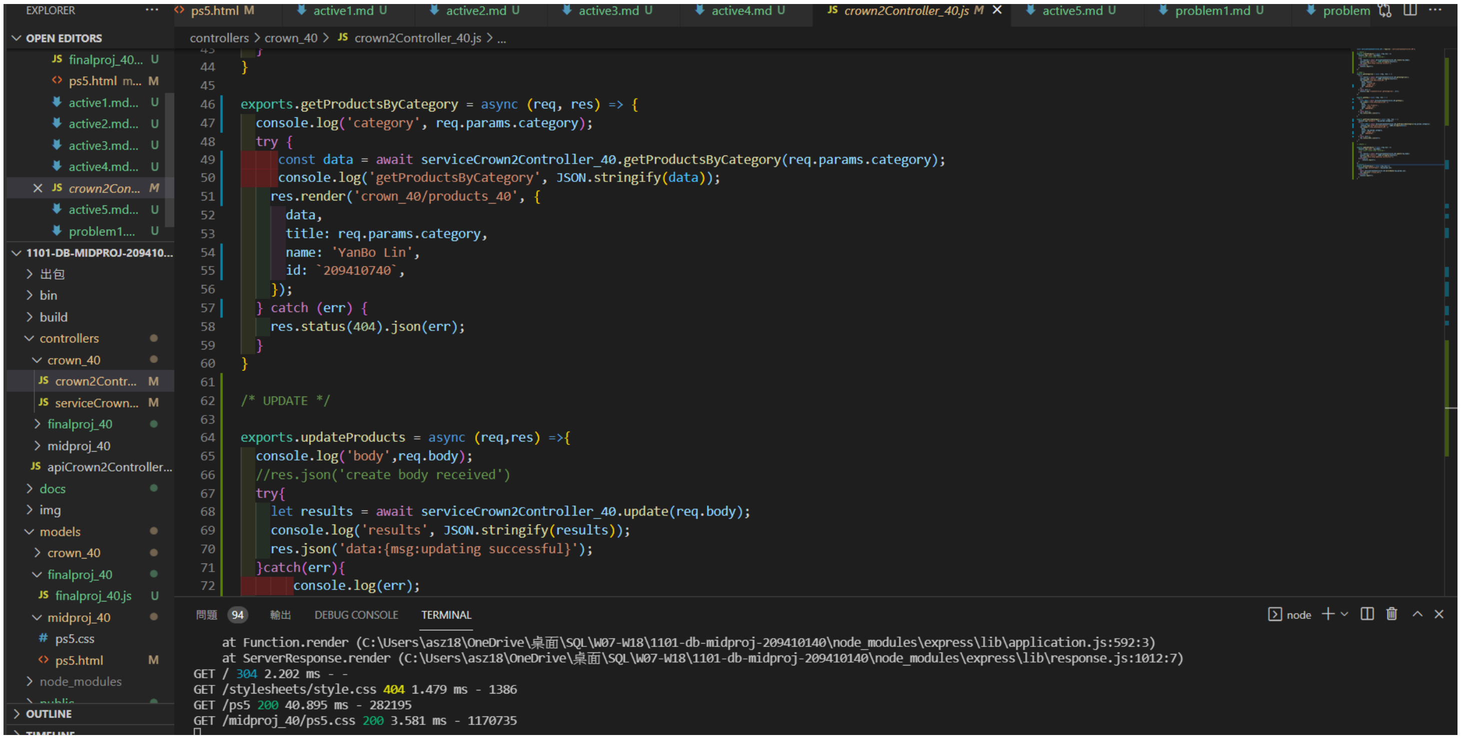
重新看過老師的上課影片，加深印象以對付考試。

這一區花很多時間，但還是沒有完全搞懂，感覺不妙。



```
/*重要!!!!!!!!!!!!!!!!!!!!!!!!!!! */


var express = require('express');
var router = express.Router();

const db = require('../utils/database');

/* CREATE*/
router.post('/create',async(req,res)=>{
    const id = req.body.id;
    const name = req.body.name;
    const author = req.body.author;
    const price = req.body.price;
    //console.log('id,name,author',id,name,author);
    try{
        const query ={
        text:`INSERT INTO book_40(id,name,author,price) VALUES ($1,$2,$3,$4);`,
        values:[id,name,author,price]
        }
        await db.query(query);
        res.redirect('/books_40');
    }catch(err){
        console.log(err);
    }
});

/* READ */
router.get('/', async function (req, res, next) {
    let data;
```

```
    at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
    at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```



```
}

exports.getProductsByCategory = async (req, res) => {
  console.log('category', req.params.category);
  try {
      const data = await serviceCrown2Controller_40.getProductsByCategory(req.params.category);
      console.log('getProductsByCategory', JSON.stringify(data));
    res.render('crown_40/products_40', {
      data,
      title: req.params.category,
      name: 'YanBo Lin',
      id: `209410740`,
    });
  } catch (err) {
    res.status(404).json(err);
  }
}

/* UPDATE */

exports.updateProducts = async (req,res) =>{
  console.log('body',req.body);
  //res.json('create body received')
  try{
    let results = await serviceCrown2Controller_40.update(req.body);
    console.log('results', JSON.stringify(results));
    res.json('data:{msg:updating successful}');
  }catch(err){
      console.log(err);
```

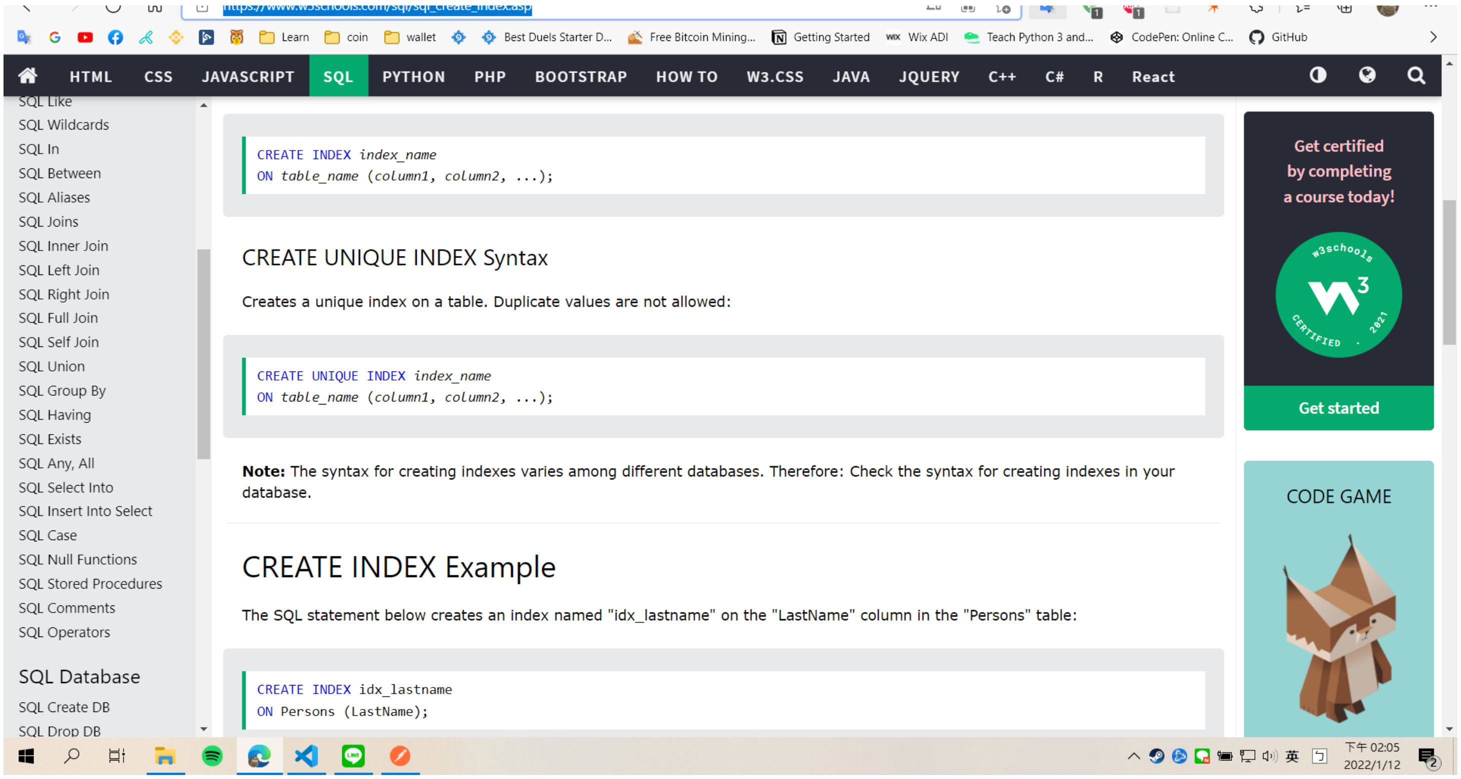
```
    at Function.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\application.js:592:3)
    at ServerResponse.render (C:\Users\asz18\OneDrive\桌面\SQL\W07-W18\1101-db-midproj-209410140\node_modules\express\lib\response.js:1012:7)
GET / 304 2.202 ms - -
GET /stylesheets/style.css 404 1.479 ms - 1386
GET /ps5 200 40.895 ms - 282195
GET /midproj_40/ps5.css 200 3.581 ms - 1170735
```

Active5

最後就是複習SQL的語法，透過助教的上課影片和W3School。

複習過了就看腦子有沒有記住了。

W3School



```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

```
CREATE UNIQUE INDEX index_name
ON table_name (column1, column2, ...);
```

Note: The syntax for creating indexes varies among different databases. Therefore: Check the syntax for creating indexes in your database.

CREATE INDEX Example

The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:

```
CREATE INDEX idx_lastname
ON Persons (LastName);
```