# 嚴選唱片 網站

## Web程式設計
## 期末專題報告

S10350315 陳宏達

# TABLE OF CONTENTS

# 動機

01


- 分頁開太多看起來很亂
- 唱片管理網站 (基本CRUD功能)
- 嚴選唱片 網站 (只有R功能)

**Rachmaninoff: Piano Concerto No. 2 in C Minor, Op. 18 - II. Adagio sostenuto**

**Rachmaninoff: The Piano Concertos &**

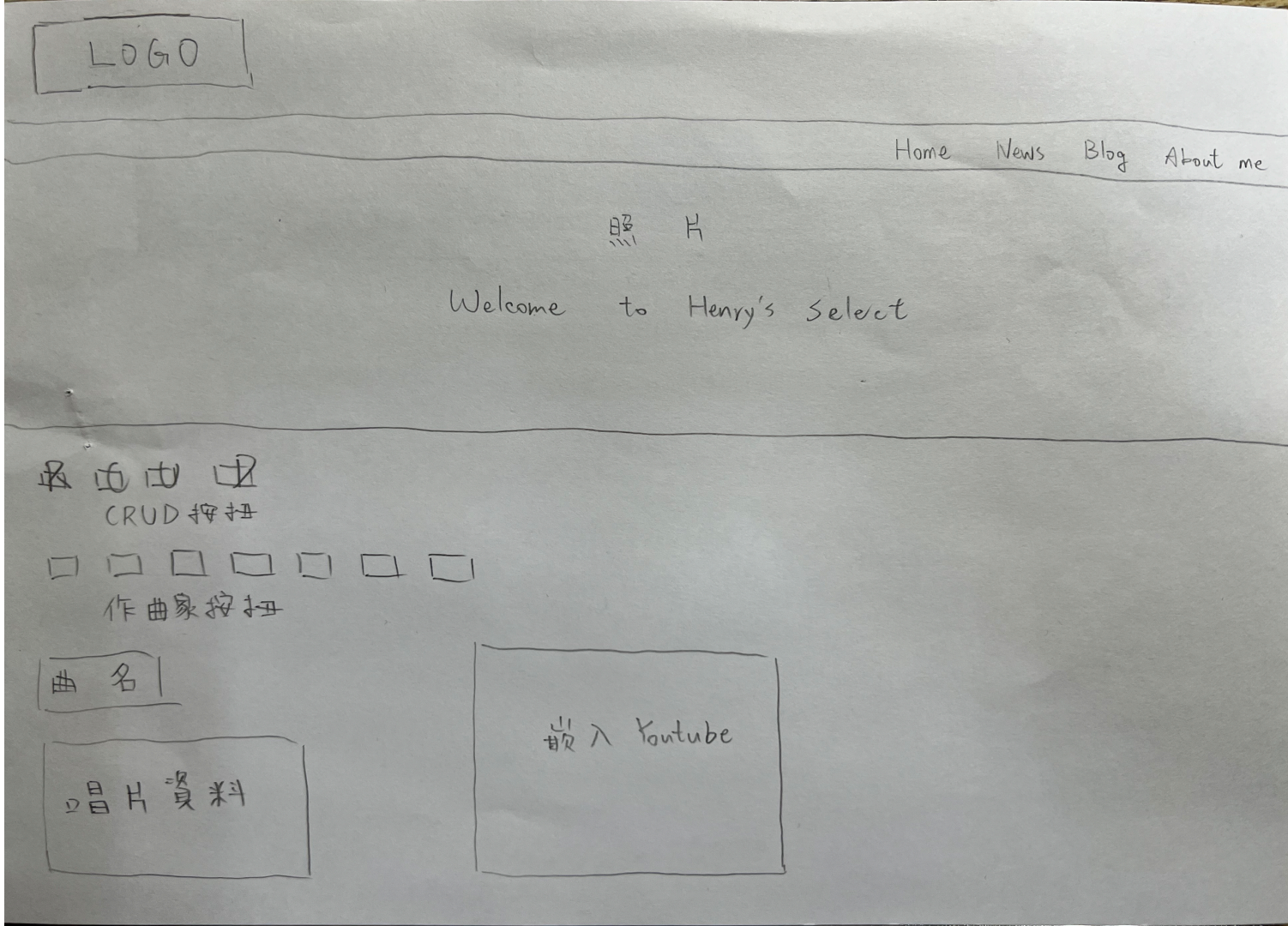
Yuja Wang, Los Angeles Philharmonic, Gustavo Duda... - 2/37

1. Rachmaninoff: Piano Concerto No. 2 in C Minor, Op. 18 - I.... 11:01 — Yuja Wang
2. ▶ Rachmaninoff: Piano Concerto No. 2 in C Minor, Op. 18 - II.... 11:04 — Yuja Wang
3. Rachmaninoff: Piano Concerto No. 2 in C Minor, Op. 18 - III.... 11:00 — Yuja Wang
4. Rachmaninoff: Piano Concerto No. 1 in F-Sharp Minor, Op. 1... 12:30 — Yuja Wang
5. Rachmaninoff: Piano Concerto No. 1 in F-Sharp Minor, Op. 1... 5:49 — Yuja Wang
6. Yuja Wang, LA Phil, Gustavo Dudamel – Rachmaninoff:... 8:09 — Deutsche Grammophon - DG
7. Rachmaninoff: Piano Concerto No. 4 in G Minor, Op. 40 (194... 9:03 — Yuja Wang

# Story Board

02

- Story Board


LOGO
Home
News
Blog
About me
照片
Welcome to Henry's Select
CRUD按扭
作曲家按扭
曲名
唱片資料
嵌入 Youtube

# 使用到的技術

03

# WHAT WE ARE WORKING ON

## 前端語言: ejs

可以寫JavaScript的html

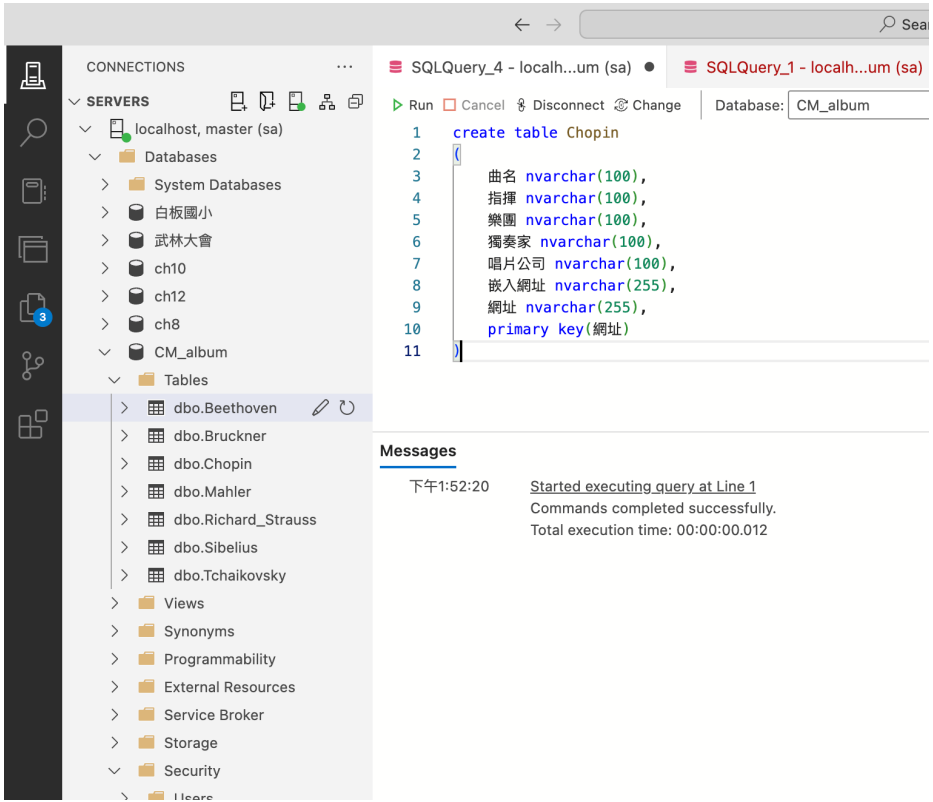
## 後端語言: node.js

可用 JavaScript 來寫伺服器端的程式碼

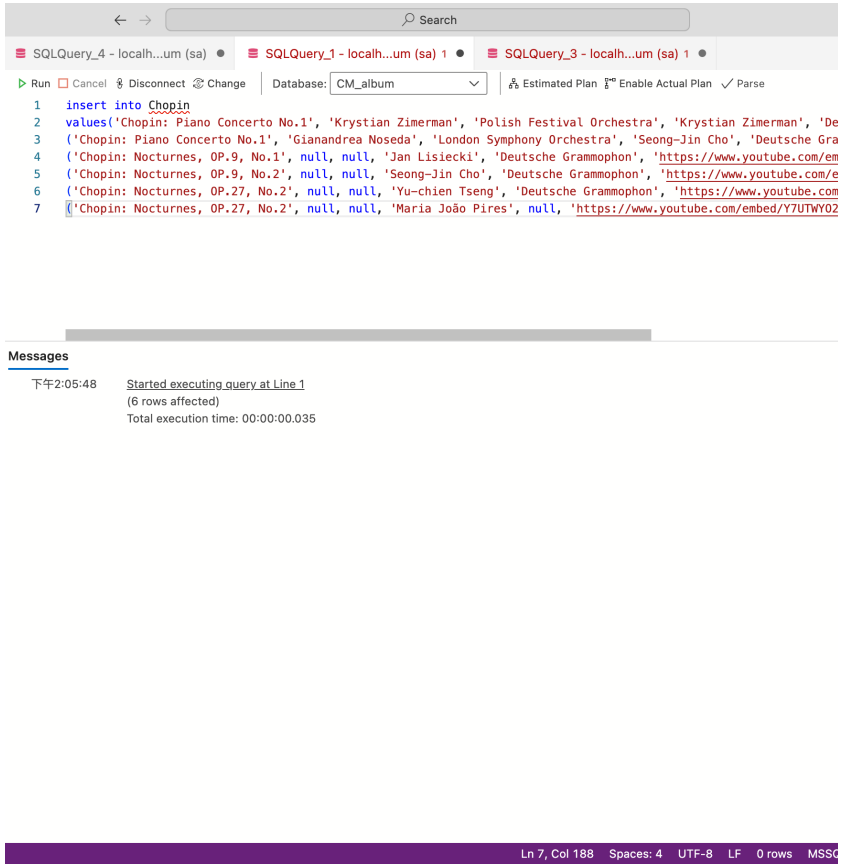
## 資料庫: sql server

express 模組
mssql模組
app.set('view engine', 'ejs');

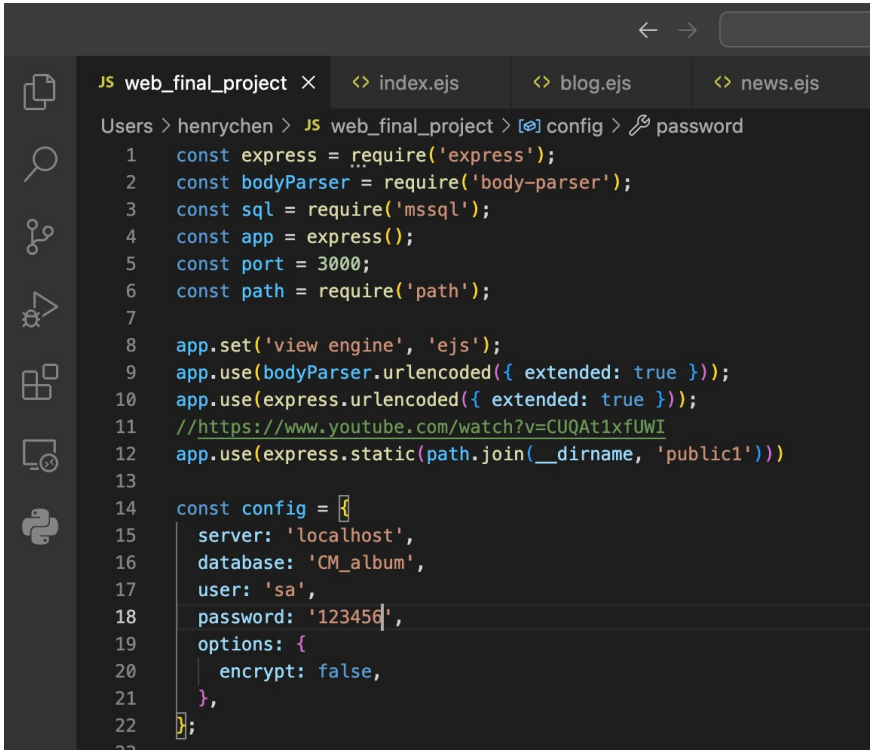
# 建資料表


CONNECTIONS
SERVERS
localhost, master (sa)
Databases
System Databases
白板國小
武林大會
ch10
ch12
ch8
CM_album
Tables
dbo.Beethoven
dbo.Bruckner
dbo.Chopin
dbo.Mahler
dbo.Richard_Strauss
dbo.Sibelius
dbo.Tchaikovsky
Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
SQLQuery_4 - localh...um (sa)
SQLQuery_1 - localh...um (sa)
Run
Cancel
Disconnect
Change
Database: CM_album
create table Chopin
(
曲名 nvarchar(100),
指揮 nvarchar(100),
樂團 nvarchar(100),
獨奏家 nvarchar(100),
唱片公司 nvarchar(100),
嵌入網址 nvarchar(255),
網址 nvarchar(255),
primary key(網址)
)
Messages
下午1:52:20
Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:00.012

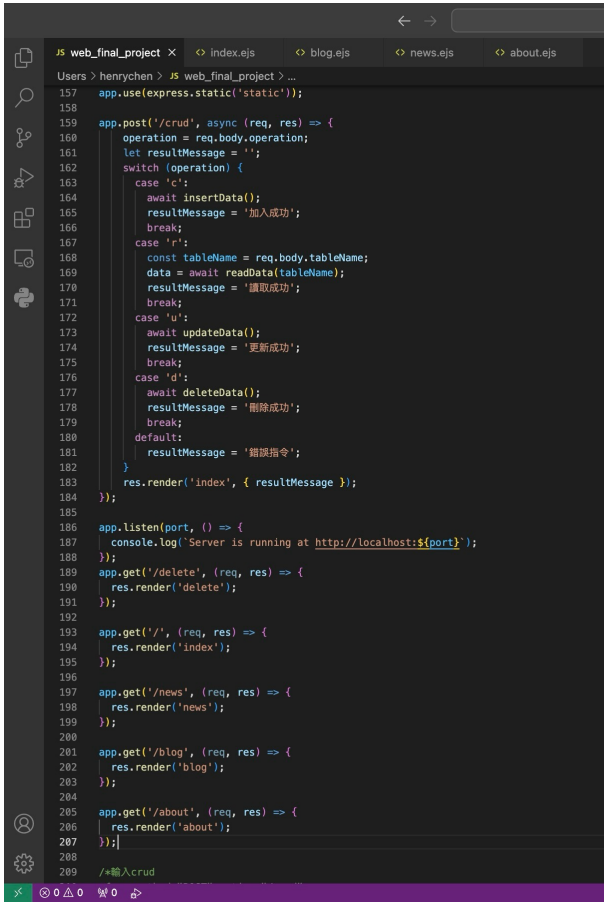
# 新增資料


Search
SQLQuery_4 - localh...um (sa)
SQLQuery_1 - localh...um (sa) 1
SQLQuery_3 - localh...um (sa) 1
Run
Cancel
Disconnect
Change
Database: CM_album
Estimated Plan
Enable Actual Plan
Parse
insert into Chopin
values('Chopin: Piano Concerto No.1', 'Krystian Zimerman', 'Polish Festival Orchestra', 'Krystian Zimerman', 'De
('Chopin: Piano Concerto No.1', 'Gianandrea Noseda', 'London Symphony Orchestra', 'Seong-Jin Cho', 'Deutsche Gra
('Chopin: Nocturnes, OP.9, No.1', null, null, 'Jan Lisiecki', 'Deutsche Grammophon', 'https://www.youtube.com/em
('Chopin: Nocturnes, OP.9, No.2', null, null, 'Seong-Jin Cho', 'Deutsche Grammophon', 'https://www.youtube.com/e
('Chopin: Nocturnes, OP.27, No.2', null, null, 'Yu-chien Tseng', 'Deutsche Grammophon', 'https://www.youtube.com
('Chopin: Nocturnes, OP.27, No.2', null, null, 'Maria João Pires', null, 'https://www.youtube.com/embed/Y7UTWYO2
Messages
下午2:05:48
Started executing query at Line 1
(6 rows affected)
Total execution time: 00:00:00.035
Ln 7, Col 188
Spaces: 4
UTF-8
LF
0 rows
MSSQ

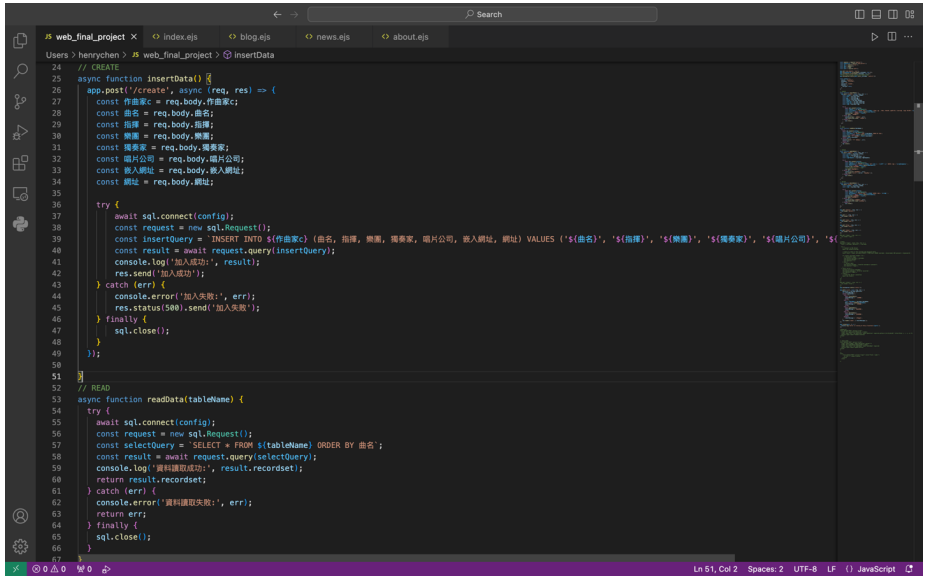
# 使用到的工具




# main程式



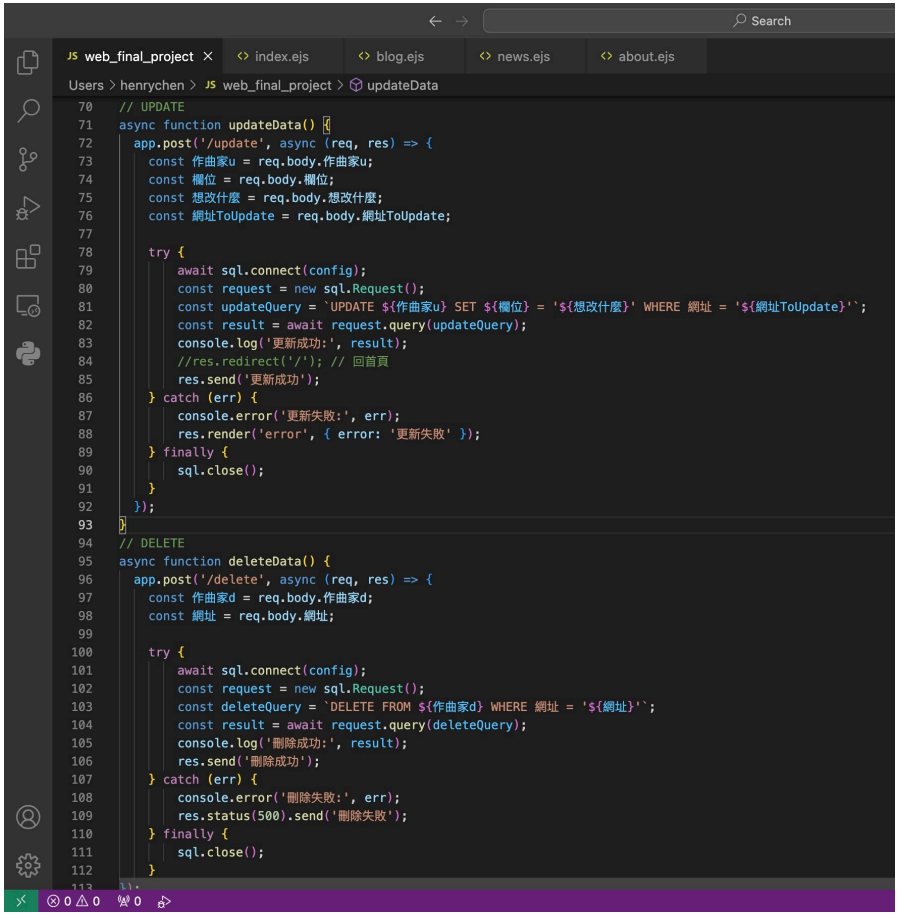
## 新增、讀取的函式



```
// CREATE
async function insertData() {
  app.post('/create', async (req, res) => {
    const 作曲家c = req.body.作曲家c;
    const 曲名 = req.body.曲名;
    const 指揮 = req.body.指揮;
    const 樂團 = req.body.樂團;
    const 獨奏家 = req.body.獨奏家;
    const 唱片公司 = req.body.唱片公司;
    const 嵌入網址 = req.body.嵌入網址;
    const 網址 = req.body.網址;

    try {
        await sql.connect(config);
        const request = new sql.Request();
        const insertQuery = `INSERT INTO ${作曲家c} (曲名, 指揮, 樂團, 獨奏家, 唱片公司, 嵌入網址, 網址) VALUES ('${曲名}', '${指揮}', '${樂團}', '${獨奏家}', '${唱片公司}', '${
        const result = await request.query(insertQuery);
        console.log('加入成功:', result);
        res.send('加入成功');
    } catch (err) {
        console.error('加入失敗:', err);
        res.status(500).send('加入失敗');
    } finally {
        sql.close();
    }
  });
}
// READ
async function readData(tableName) {
  try {
    await sql.connect(config);
    const request = new sql.Request();
    const selectQuery = `SELECT * FROM ${tableName} ORDER BY 曲名`;
    const result = await request.query(selectQuery);
    console.log('資料讀取成功:', result.recordset);
    return result.recordset;
  } catch (err) {
    console.error('資料讀取失敗:', err);
    return err;
  } finally {
    sql.close();
  }
}
```

## 更新、刪除的函式



```
// UPDATE
async function updateData() {
  app.post('/update', async (req, res) => {
    const 作曲家u = req.body.作曲家u;
    const 欄位 = req.body.欄位;
    const 想改什麼 = req.body.想改什麼;
    const 網址ToUpdate = req.body.網址ToUpdate;

    try {
        await sql.connect(config);
        const request = new sql.Request();
        const updateQuery = `UPDATE ${作曲家u} SET ${欄位} = '${想改什麼}' WHERE 網址 = '${網址ToUpdate}'`;
        const result = await request.query(updateQuery);
        console.log('更新成功:', result);
        //res.redirect('/'); // 回首頁
        res.send('更新成功');
    } catch (err) {
        console.error('更新失敗:', err);
        res.render('error', { error: '更新失敗' });
    } finally {
        sql.close();
    }
  });
}
// DELETE
async function deleteData() {
  app.post('/delete', async (req, res) => {
    const 作曲家d = req.body.作曲家d;
    const 網址 = req.body.網址;

    try {
        await sql.connect(config);
        const request = new sql.Request();
        const deleteQuery = `DELETE FROM ${作曲家d} WHERE 網址 = '${網址}'`;
        const result = await request.query(deleteQuery);
        console.log('刪除成功:', result);
        res.send('刪除成功');
    } catch (err) {
        console.error('刪除失敗:', err);
        res.status(500).send('刪除失敗');
    } finally {
        sql.close();
    }
```

# 碰到的問題

04

問題 問題 問題

## python sql server連接

改用node.js跟ejs

## js跟ejs連接

在Users建views檔案夾

## ejs讀照片

在Users建Public檔案夾

## YouTube 嵌入網址

專屬的嵌入網址
重建資料庫

# 未來計畫

05

1.刪除CUD功能

2.更新唱片、新聞、部落格

# Thank You Very Much

Mercury is the closest planet to the Sun and the smallest one in the Solar System–it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal since it was named after the Roman messenger god, Mercury