What you need to accomplish in this assignment:

1. Watch video: https://www.youtube.com/watch?v=Q3ixb1w-QaY Also watch this video to explain the code.

2. Run the sample code according to the video and the instructions given below.

3. Install nodejs and npm, and packages including express, mysql, cors, and nodemon.

4. Change the frontend code (index.html, main.jsx, and App.jsx) and the backend code (server.js) from the sample code to display the results of running 10 SELECT statements, each of them should select from at least TWO tables and its where clause has at least two conditions.

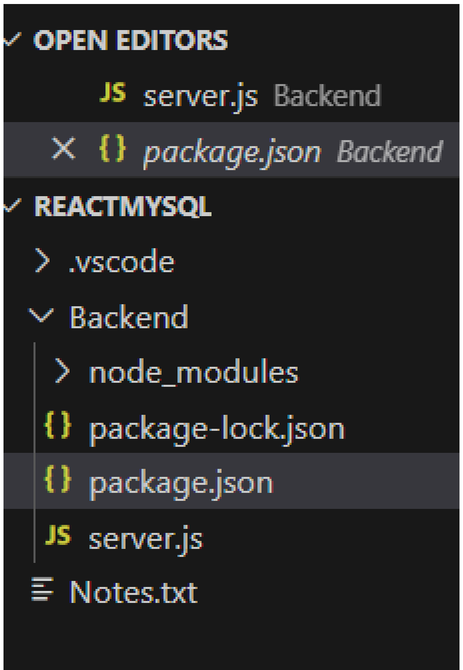
   Example 1: `SELECT crscode FROM Students, Transcript WHERE students.id = Transcript.studentid AND transcipt.semester = 'Fall2007'` Exmaple 2: `SELECT People.name FROM People, Hobbies WHERE People.id = Hobbies.peopleid and Hobbies.hobby = 'swimming'`

5. The code should be running to deserve full credits. Submit all SQL statements in a file called sql.txt.

Note: grading is based on your fluency of SQL, the success execution result of server.js showing on the web browser and your explanation skill of the result.

How you go about accomplishing the above: (by first simulating it with the sample code)

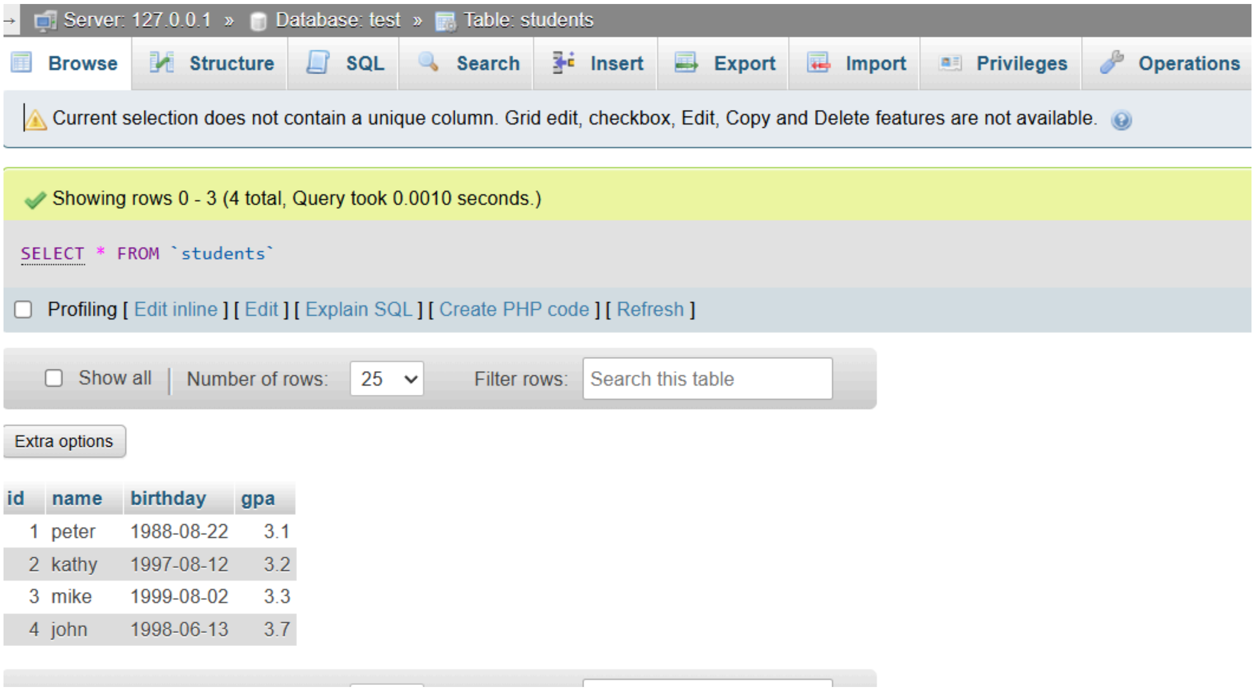
1. In your file system, create a directory called `reactmysql` as your project directory.
2. Under `reactmysql`, create directory `Backend`.
3. Copy the files under Backend (under https://github.com/shiyonglu/database_javascript/tree/main/exercise3) to Backend.



```
OPEN EDITORS
    JS server.js Backend
  × {} package.json Backend
REACTMYSQL
  > .vscode
  ⌄ Backend
    > node_modules
    {} package-lock.json
    {} package.json
    JS server.js
  Notes.txt
```

4. Create a table called students in the `test` database as follows:

```
CREATE TABLE students (id SMALLINT, name VARCHAR(100), birthday DATE, gpa FLOAT);

INSERT INTO students VALUE (1, "peter", '1988-08-22', 3.1);

INSERT INTO students VALUE (2, "kathy", '1997-08-12', 3.2);

INSERT INTO students VALUE (3, "mike", '1999-08-02', 3.3);

INSERT INTO students VALUE (4, "john", '1998-06-13', 3.7);
```



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

```
SELECT * FROM `students`
```

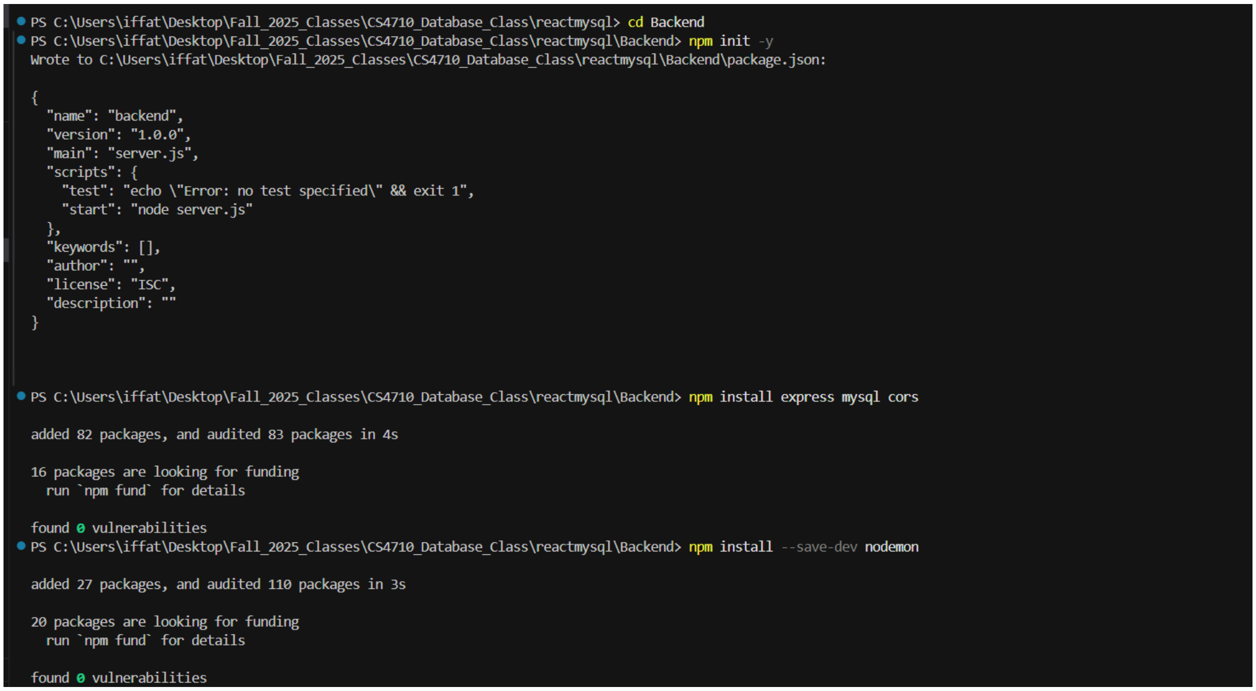
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

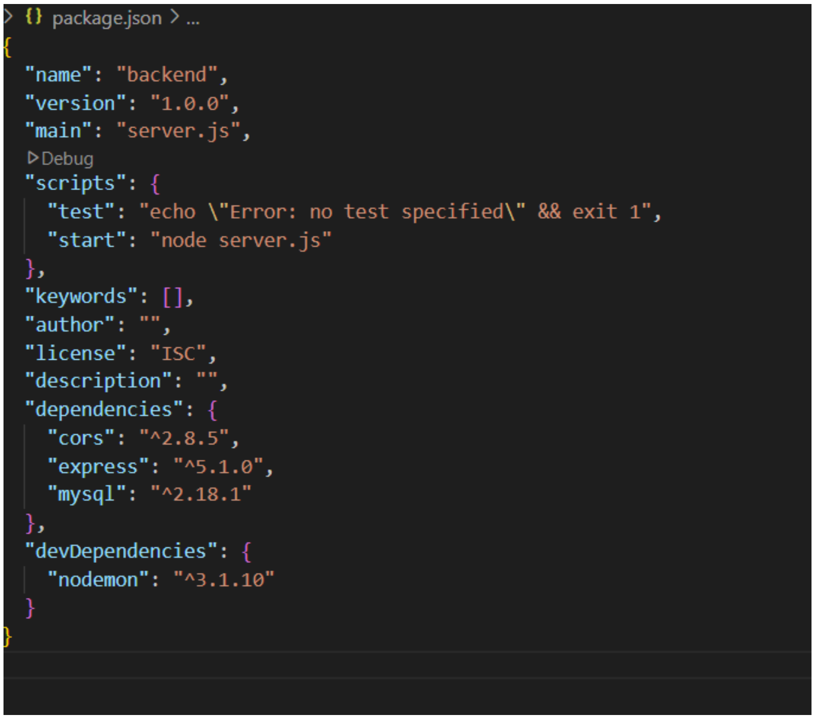
Extra options

id	name	birthday	gpa
1	peter	1988-08-22	3.1
2	kathy	1997-08-12	3.2
3	mike	1999-08-02	3.3
4	john	1998-06-13	3.7

5. cd Backend
6. npm init -y
7. npm install express mysql cors nodemon
8. Modify the scripts section of the Backend/package.json as follows:



```
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql> cd Backend
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend> npm init -y
Wrote to C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend\package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}


PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend> npm install express mysql cors

added 82 packages, and audited 83 packages in 4s

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

found 0 vulnerabilities
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend> npm install --save-dev nodemon

added 27 packages, and audited 110 packages in 3s

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

found 0 vulnerabilities
```



```
{
  "name": "backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^5.1.0",
    "mysql": "^2.18.1"
  },
  "devDependencies": {
    "nodemon": "^3.1.10"
  }
}
```

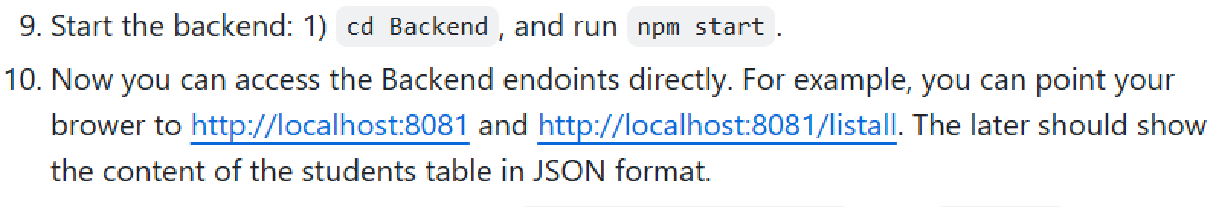
*** For step 5) why do we switch to backend and then do the following command in the terminal for step 6)?** The backend is a directory that refers to the Backend portion of the program, that is everything that occurs behind the scene or what the user is unable to see. This includes the data being handled on the server side of things. The server.js file is file that fits this perfectly, specifically, this file acts as the middle man between the database and the frontend file app.jsx (where app.jsx outputs the preferred data on the web application) where server.js gives app.jsx the needed data.

*** For step 6) why do we do the command npm init -y?:** This command creates the package.json. The package.json file is the heart of a node.js project. It describes your project and keeps track of its dependencies (packages via npm install, scripts, and settings).

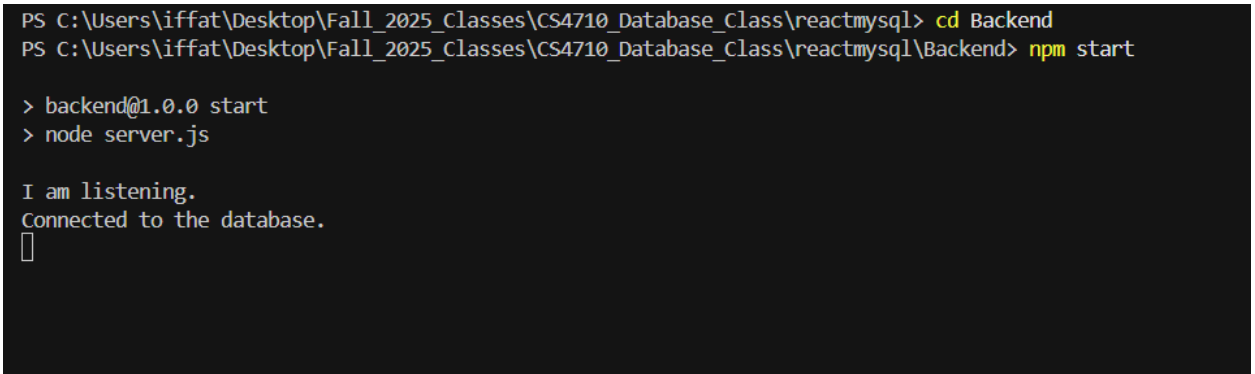
1) Describes your project -> Project metadata: Stores info like project name, version, description, author, license.
2) Keeps track of dependencies -> dependencies: Note that this section of package.json does not store the code of the packages installed via npm install but rather their names. The actual code is stored node_modules/. A package can be a library, a framework, or a tool, or some piece of code.
3) Scripts (lets you define custom commands for your project) -> scripts: (refer to def for this section)
    - At this point, you do the command npm start. This will start the file in the Backend directory, which is sever.js.

*** For step 7) why do we use the command npm install express mysql cors & then npm install --save-dev nodemon ?:** This is where we utilize the NPM to install the following packages/libraries:

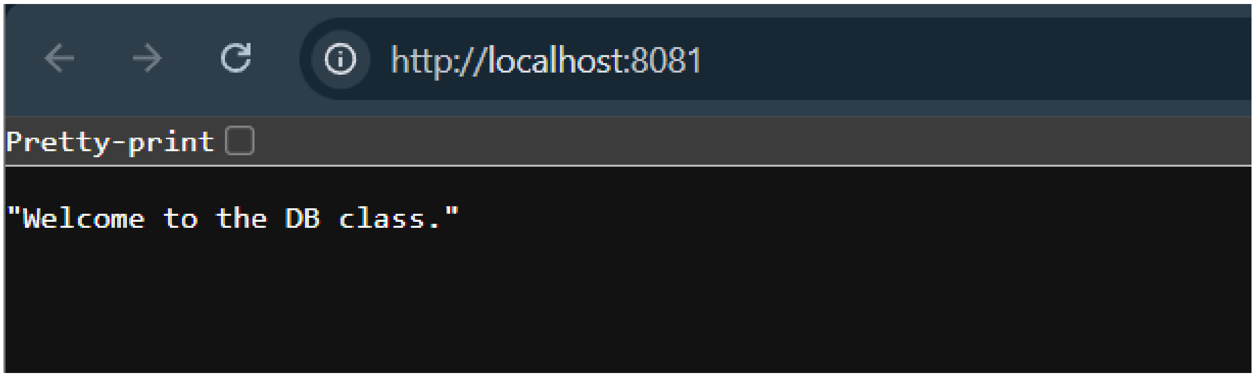
- express → web framework
- mysql → connector for mySQL database server
- cors → middleware to handler cross origin
- nodemon -> this is a dev dependency. Anytime we edit our code, this will restart our server automatically.



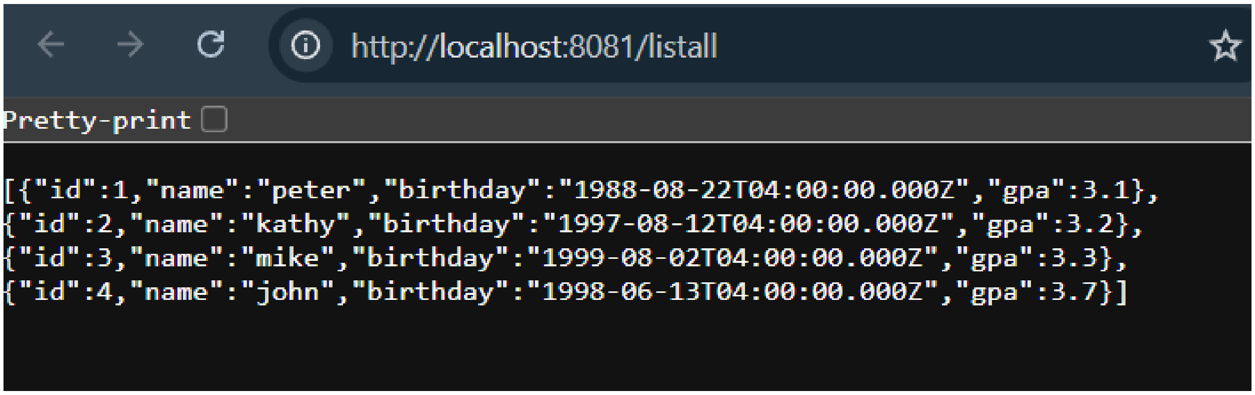
9. Start the backend: 1) `cd Backend`, and run `npm start`.
10. Now you can access the Backend endoints directly. For example, you can point your brower to http://localhost:8081 and http://localhost:8081/listall. The later should show the content of the students table in JSON format.



```
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql> cd Backend
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend> npm start

> backend@1.0.0 start
> node server.js

I am listening.
Connected to the database.
```

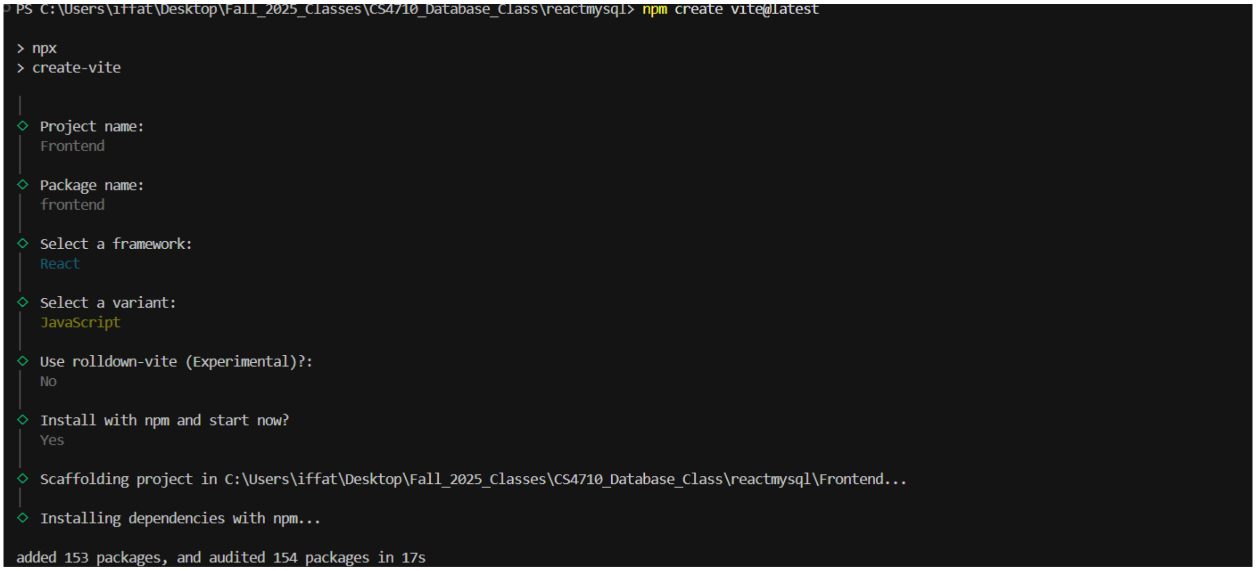


```
"Welcome to the DB class."
```



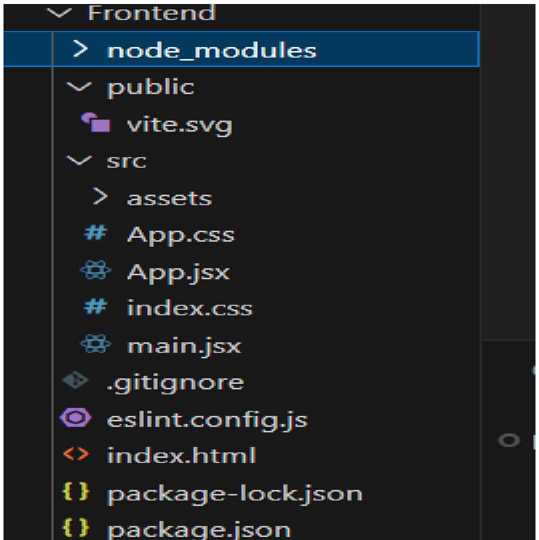
```
[{"id":1,"name":"peter","birthday":"1988-08-22T04:00:00.000Z","gpa":3.1},
{"id":2,"name":"kathy","birthday":"1997-08-12T04:00:00.000Z","gpa":3.2},
{"id":3,"name":"mike","birthday":"1999-08-02T04:00:00.000Z","gpa":3.3},
{"id":4,"name":"john","birthday":"1998-06-13T04:00:00.000Z","gpa":3.7}]
```

11. At the project home directory, run `npm create vite@latest`, type `Frontend` as the project name, and choose `react` and `javascript`.
12. Replace the App.jsx file under Frontend/src by the one under https://github.com/shiyonglu/database_javascript/tree/main/exercise3/Frontend. Open a new terminal to run the following commands.
13. cd Frontend
14. Run `npm install`
15. Run `npm run dev` and then point your browser to `http://localhost:5173/` which should show the page that displays the query result of the students table.



```
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql> npm create vite@latest

> npx
> create-vite

│
◇  Project name:
│  Frontend
│
◇  Package name:
│  frontend
│
◇  Select a framework:
│  React
│
◇  Select a variant:
│  JavaScript
│
◇  Use rolldown-vite (Experimental)?:
│  No
│
◇  Install with npm and start now?
│  Yes
│
◇  Scaffolding project in C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Frontend...
│
◇  Installing dependencies with npm...

added 153 packages, and audited 154 packages in 17s
```

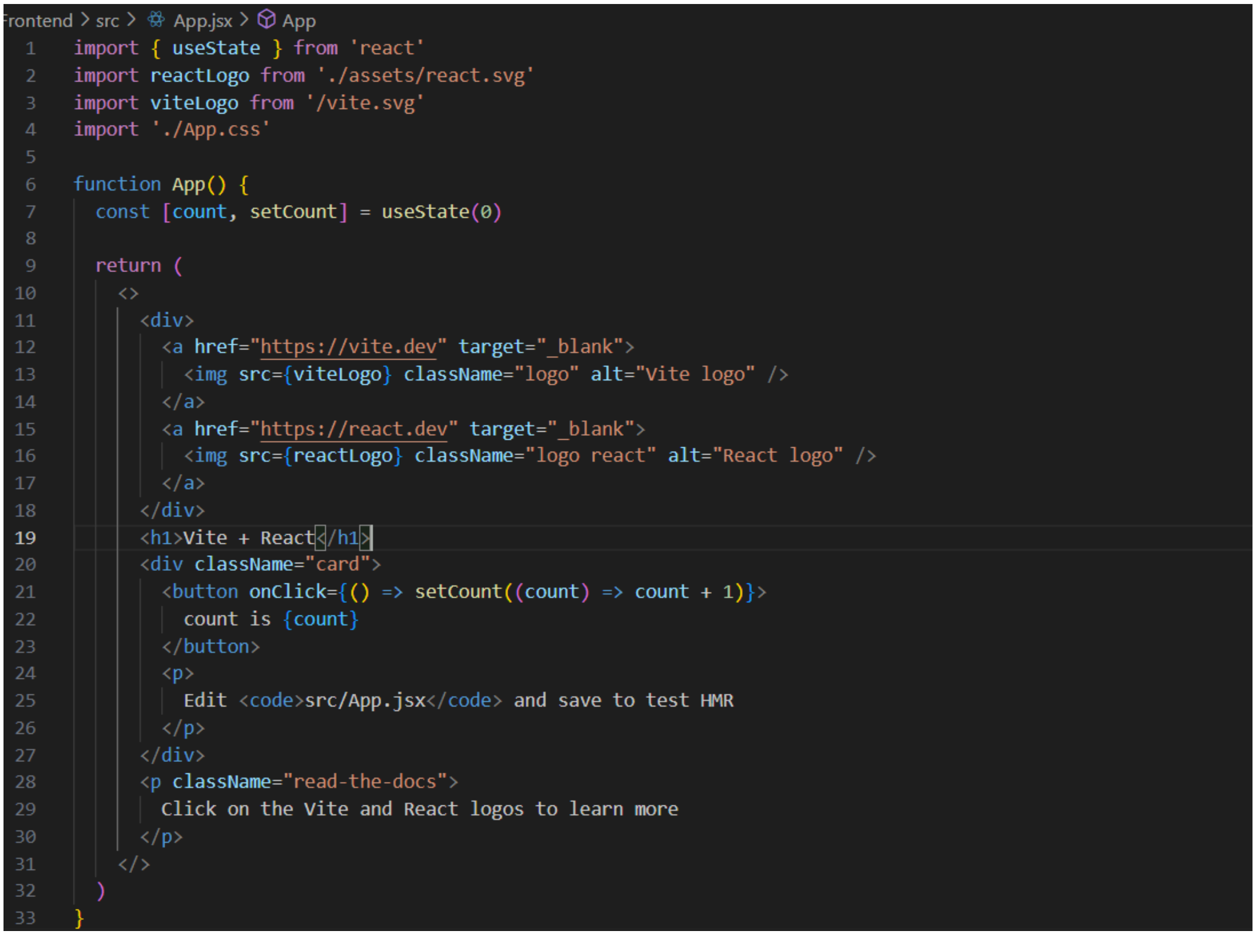
Result:



```
∨ Frontend
  > node_modules
  ∨ public
     vite.svg
  ∨ src
    > assets
     App.css
     App.jsx
     index.css
     main.jsx
   .gitignore
   eslint.config.js
   index.html
   package-lock.json
   package.json
```

- **The Frontend directory, mentioned in the assignment, is finally created and in your project directory reactmysql.**

In step 12) Here is what the App.jsx file looks like by default (before replacing) after running the commands in step 11)



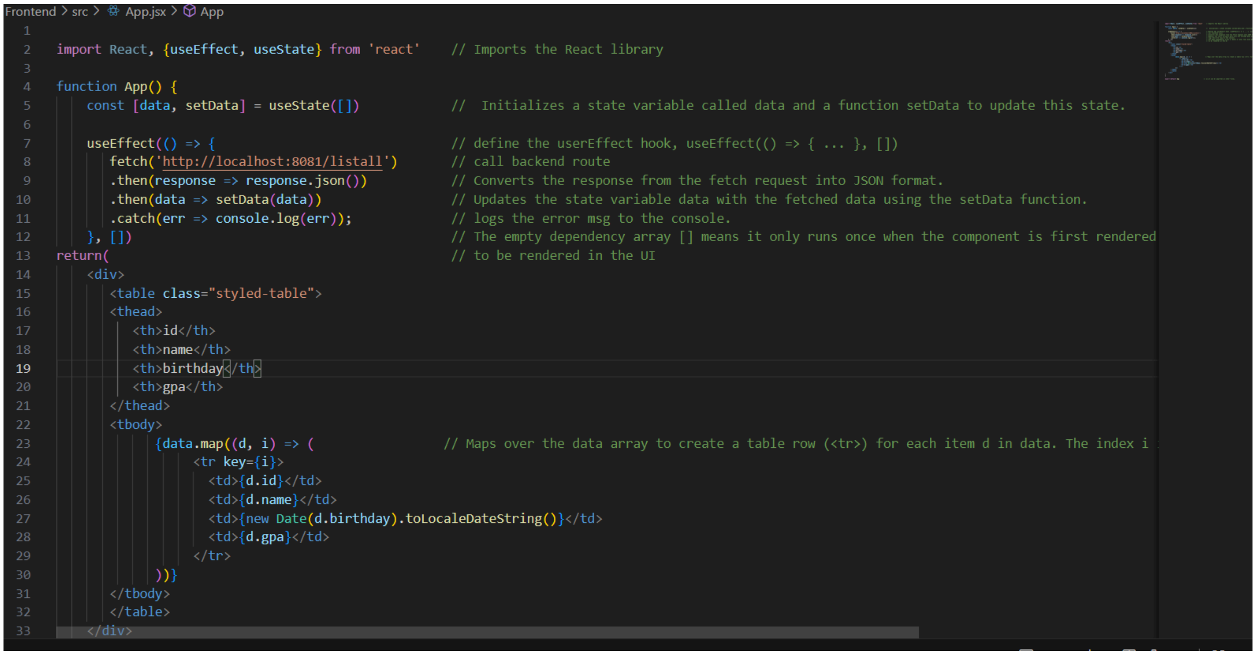
```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function App() {
  const [count, setCount] = useState(0)

  return (
    <>
      <div>
        <a href="https://vite.dev" target="_blank">
          <img src={viteLogo} className="logo" alt="Vite logo" />
        </a>
        <a href="https://react.dev" target="_blank">
          <img src={reactLogo} className="logo react" alt="React logo" />
        </a>
      </div>
      <h1>Vite + React</h1>
      <div className="card">
        <button onClick={() => setCount((count) => count + 1)}>
          count is {count}
        </button>
        <p>
          Edit <code>src/App.jsx</code> and save to test HMR
        </p>
      </div>
      <p className="read-the-docs">
        Click on the Vite and React logos to learn more
      </p>
    </>
  )
}
```

*** What does the command npm create vite@latest do?:** This creates a new Vite React Project. The terminal will react by asking for **Project Name:** Frontend (this is where the Frontend directory will be created and placed into the project directory). The terminal will react by asking for the **Package name:** frontend (this info goes under the name section for the package.json file. This must be lowercase and no spaces as it should also be the same name as the Project Name. For **Framework** you choose React and for **Varaint** you choose **Javascript.**

* **What does the command npm install do (when executed at the Frontend directory after doing all the prior commands)?:** This will install the dependencies (library/framework) react

and vite. You only need to do this command once on your local machine (assuming said machine as the data/project directory above).

*** What does the command npm run dev do?:** This command will start a local dev server to view your app.
Here’s how the updated App.jsx file looks like now after replacing it:

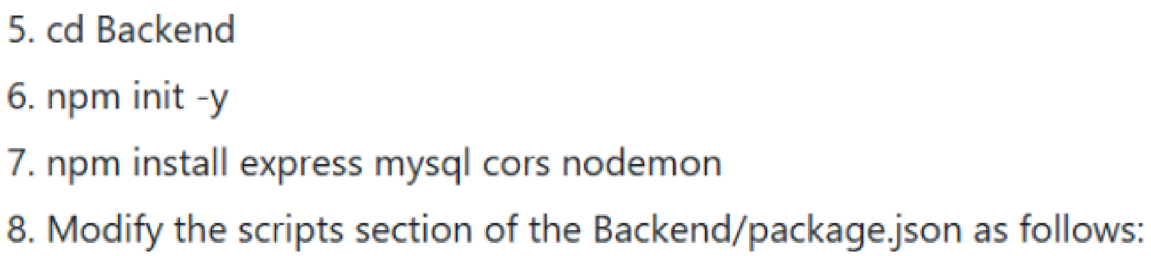


```

import React, {useEffect, useState} from 'react'    // Imports the React library

function App() {
    const [data, setData] = useState([])           //  Initializes a state variable called data and a function setData to update this state.

    useEffect(() => {                              // define the userEffect hook, useEffect(() => { ... }, [])
       fetch('http://localhost:8081/listall')      // call backend route
       .then(response => response.json())          // Converts the response from the fetch request into JSON format.
       .then(data => setData(data))                // Updates the state variable data with the fetched data using the setData function.
       .catch(err => console.log(err));            // logs the error msg to the console.
    }, [])                                         // The empty dependency array [] means it only runs once when the component is first rendered
return(                                            // to be rendered in the UI
    <div>
       <table class="styled-table">
       <thead>
          <th>id</th>
          <th>name</th>
          <th>birthday</th>
          <th>gpa</th>
       </thead>
       <tbody>
             {data.map((d, i) => (                // Maps over the data array to create a table row (<tr>) for each item d in data. The index i
                  <tr key={i}>
                    <td>{d.id}</td>
                    <td>{d.name}</td>
                    <td>{new Date(d.birthday).toLocaleDateString()}</td>
                    <td>{d.gpa}</td>
                  </tr>
             ))}
       </tbody>
       </table>
    </div>
```

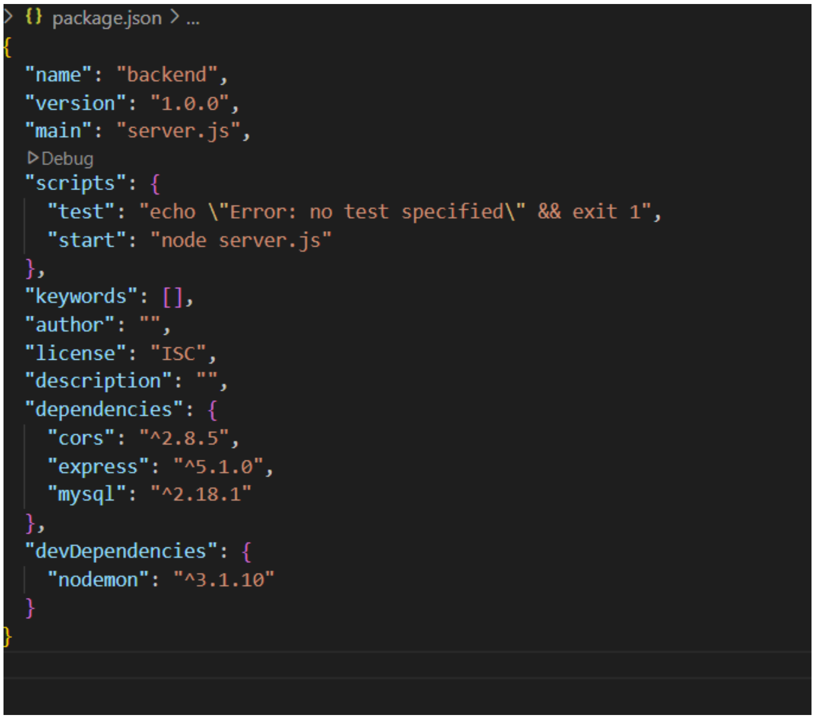
It should be noted that once you do the npm commands for the Backend and Frontend directory with the following packages, the creation of the project will be complete. The rest depends on the source code for the project.

How do we compile, deploy, and run your (Iffat's) code for database assignment #3?: The steps for doing x, y, z for my code in comparison to the sample code are very similar as all we are doing is changing the source code of the files mentioned above to output data of our choice from our database! Let's go over them

* **Compile, Deploy, & Run :**
   - Make sure you have an IDE (like Visual Studio Code) installed along with [Node.js](). Verify you have [Node.js]() installed by doing node -v in your terminal
   - Clone the following repo on to your local machine and then cd Backend on the terminal: [https://github.com/Iffat313/CS4710_IntroToDatabase/tree/main/reactmysql]()
   - Make sure you have an XAMPP control panel and use the test database to upload the necessary tables using the proper queries via the phpmyadmin web application. (The queries are on the repo as "DataForDatabase.sql"
   - In your terminal do the following to initialize the package.json file and modules.

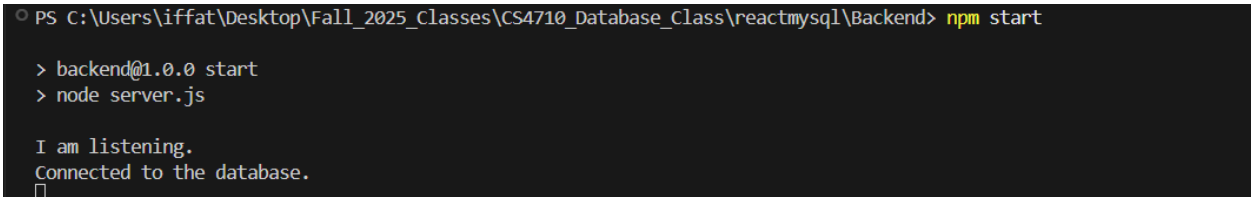


5. cd Backend
6. npm init -y
7. npm install express mysql cors nodemon
8. Modify the scripts section of the Backend/package.json as follows:



```
{
  "name": "backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^5.1.0",
    "mysql": "^2.18.1"
  },
  "devDependencies": {
    "nodemon": "^3.1.10"
  }
}
```

- In your Backend directory via your terminal run the following command: “npm start.” You should be able to see the following:



```
PS C:\Users\iffat\Desktop\Fall_2025_Classes\CS4710_Database_Class\reactmysql\Backend> npm start

 > backend@1.0.0 start
 > node server.js

I am listening.
Connected to the database.
```

- Now you’ll be able to access the following four Backend endpoints directly, where you should be able to see the content of the corresponding tables in JSON format:
    1) http://localhost:8081/Cams&Consoles
    2) http://localhost:8081/Michigan
    3) http://localhost:8081/Other_Tech&Consoles
    4) http://localhost:8081/Countries
- Unlike the sample code, I did not have one endpoint of /listall just for the sake of preference/consistency as Im working with various queries that are working with different tables in the same database.
- In your terminal, cd to the project directory and type the following command: “npm create vite@latest”. This command will ask to do the following below which essentially
- Project name = “Frontend”
- Package name = “frontend” (all lowercase)
- Framework = “React”
- Variant = “Javascript”
- In your terminal, cd to Frontend and type the following commands: “npm install” and then “npm run dev”. If the terminal offers to do this for you, hit yes.
- Replace the source for App.jsx with the one in the project directory of the repository given above
- Now you can go to the Frontend local host website to see the result: http://localhost:5173/