

KEEP YOUR FILES FOR LAB 7 USE COMPLETED LAB 7 TO START ASSIGNMENT 3.

CPSC Lab 7: ES Module Project

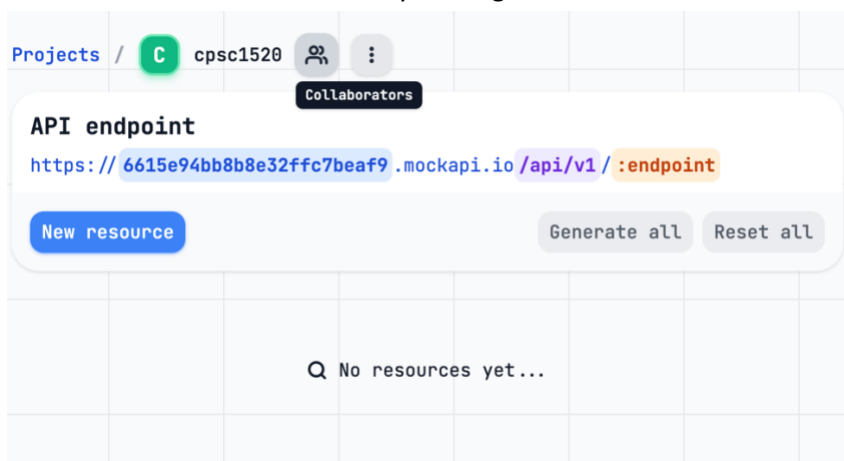
Introduction

You've increased the search functionality on your album creator application by having a backend implementation for the album data. You will be using the mock api service that we have been using in class. The album data for the application is inside the public/data folder.

Task 1: Create the Mock API Server Data Schema

If you do not have a current project for the mock api service, then you will need to create one as demonstrated in class. The free account only allows you to have a single project with two resources. For the lab and the assignment, you are going to create an albums and a favorites resource.

1 Add the albums and resource by clicking on the new resource button.



Task 2 Albums Resource

In the new resource panel give the resource the name albums.

NEW RESOURCE

X

Resource name

Enter meaningful resource name, it will be used to generate API endpoints.

albums

Task 3 Albums Schema

Define the album schema to match the albums.json data file (located in: public/data/albums.json). Make sure to spell the property names correctly so they match the albums.json property names. Once you have finished this step you can click on the create button at the bottom of the resource panel. Next you're going to add another resource.

Schema

Define Resource schema, it will be used to generate mock data.

id	Object ID	
albumName	Faker.js	lorem.words
artistName	Faker.js	lorem.words
releaseDate	Faker.js	lorem.words
descriptors	Faker.js	lorem.words
averageRating	Faker.js	lorem.words
numberRatings	Faker.js	lorem.words
numberReviews	Faker.js	lorem.words
uid	Faker.js	random.word



Task 4 Favorites Resource

Create another resource and name this one favorites. This is where the albums that have been added to your favorites list will be store. You will perform this task in Assignment 3

NEW RESOURCE



Resource name

Enter meaningful resource name, it will be used to generate API endpoints.

favorites

Setup Task 4 Favorites Schema

Create the favorites schema it should be the same as the album's schema. When you have finished with the schema click on the create button.

Schema



Define Resource schema, it will be used to generate mock data.

id	Object ID	
albumName	Faker.js	lorem.words
artistName	Faker.js	lorem.words
releaseDate	Faker.js	lorem.words
descriptors	Faker.js	lorem.words
averageRating	Faker.js	lorem.words
numberRatings	Faker.js	lorem.words
numberReviews	Faker.js	lorem.words
uid	Faker.js	random.word



Task 5 Adding the album data.

Now you are going to copy the album data into the albums resource. Click on the data button for the albums resource.

[Projects](#) / C cpssc1520  

API endpoint

<https://6615e94bb8b8e32ffc7beaf9.mockapi.io/api/v1/> :endpoint


New resource

albums

0

Data

Edit



favorites

0

Setup Task 6 Adding the album data.

You are going to copy the array from the album.json data in the assignment into the resource data panel.

RESOURCE DATA - ALBUMS 

Resource data

Edit/replace data for **albums** resource. Data must be an array and a valid JSON.

[]

Close

Update

Task 7 Copy the album data. (75 albums total)

Open the albums data in the assignment folder path: ***public/data/albums.json***. Copy the contents of this file.

```
{
  "id": "0",
  "albumName": "OK Computer",
  "artistName": "Radiohead",
  "releaseDate": "16 June 1997",
  "genres": "Alternative Rock, Art Rock",
  "descriptors": "melancholic, anxious, futuristic, alienation, existential, male vocals, at
introspective",
  "averageRating": "4.23",
  "numberRatings": "70,382",
  "numberReviews": "1531",
  "uid": "ST33easu"
},
{
  "id": "1",
  "albumName": "Wish You Were Here",
  "artistName": "Pink Floyd",
  "releaseDate": "12 September 1975",
  "genres": "Progressive Rock, Art Rock",
  "descriptors": "melancholic, atmospheric, progressive, male vocals, concept album, introsp
bittersweet, meditative",
  "averageRating": "4.29",
  "numberRatings": "48,662",
  "numberReviews": "983",
  "uid": "VipgI9pu"
},
{
  "id": "2",
  "albumName": "In the Court of the Crimson King",
  "artistName": "King Crimson",
  "releaseDate": "10 October 1969",
  "genres": "Progressive Rock, Art Rock",
  "descriptors": "fantasy, epic, progressive, philosophical, complex, surreal, poetic, male
technical",
  "averageRating": "4.30",
  "numberRatings": "44,943",
  "numberReviews": "870",
  "uid": "VuTqs8jU"
},
{
  "id": "3",
  "albumName": "Kid A",
  "artistName": "Radiohead",
```

Task 8 Paste the album data into the albums Resource data panel

Copy the contents of this file and paste it into the resource data panel. Click the update button to save this data.

RESOURCE DATA - TEST



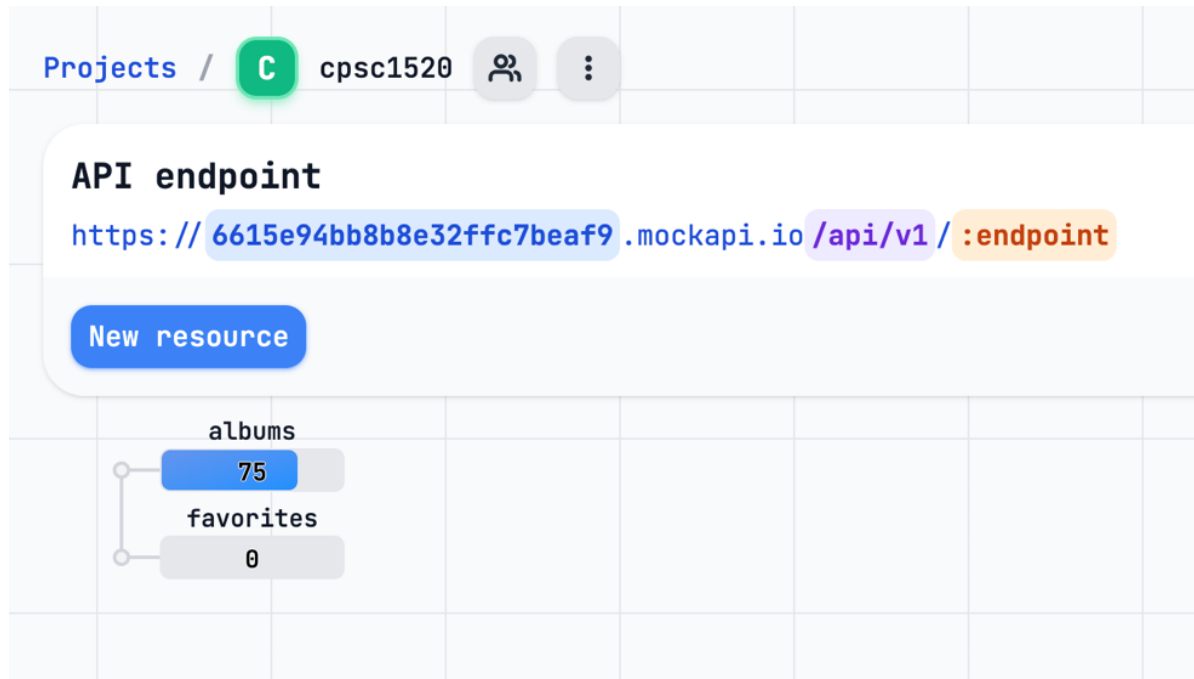
Resource data

Edit/replace data for **test** resource. Data must be an array and a valid JSON.

```
[
  {
    "id": "0",
    "albumName": "OK Computer",
    "artistName": "Radiohead",
    "releaseDate": "16 June 1997",
    "genres": "Alternative Rock, Art Rock",
    "descriptors": "melancholic, anxious, futuristic, alienation,
existential, male vocals, atmospheric, lonely, cold, introspective",
    "averageRating": "4.23",
    "numberRatings": "70,382",
    "numberReviews": "1531",
    "uid": "ST33easu"
  },
  {
    "id": "1",
    "albumName": "Wish You Were Here",
    "artistName": "Pink Floyd",
    "releaseDate": "12 September 1975",
    "genres": "Progressive Rock, Art Rock",
    "descriptors": "melancholic, atmospheric, progressive, male vocals,
concept album, introspective, serious, longing, bittersweet, meditative",
    "averageRating": "4.29",
    "numberRatings": "48,662",
    "numberReviews": "983",
    "uid": "VipgI9pu"
  },
  {
    "id": "2",
    "albumName": "In the Court of the Crimson King",
    "artistName": "King Crimson",
```

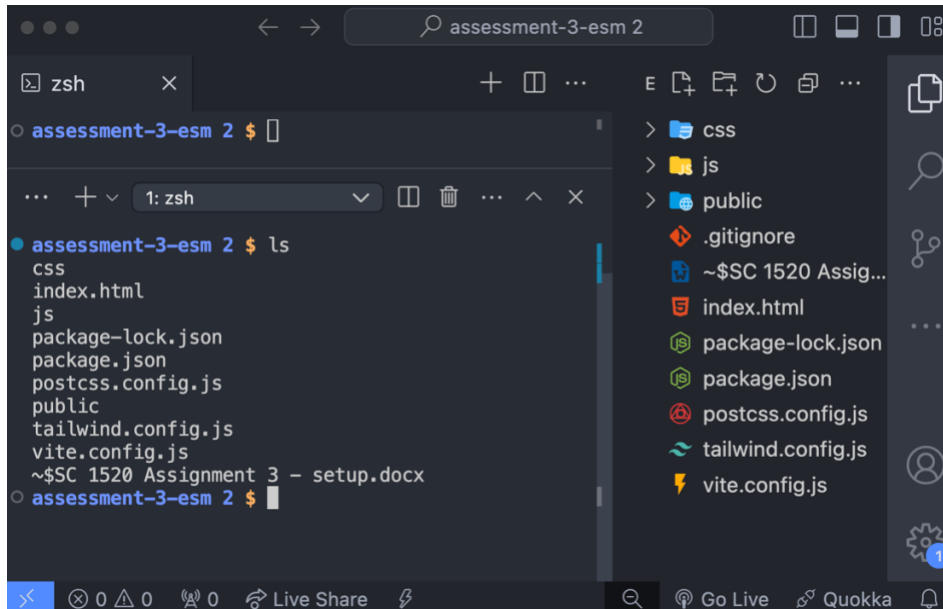
Task 9: Check Your Setup

You have now completed the setup of your mock album api service. Below is an image of how your project panel should look. At this point click on the API endpoint url. You should see the album data in a new browser tab, copy the url you will need it for the next task.



Task 10 Installing Node Modules.

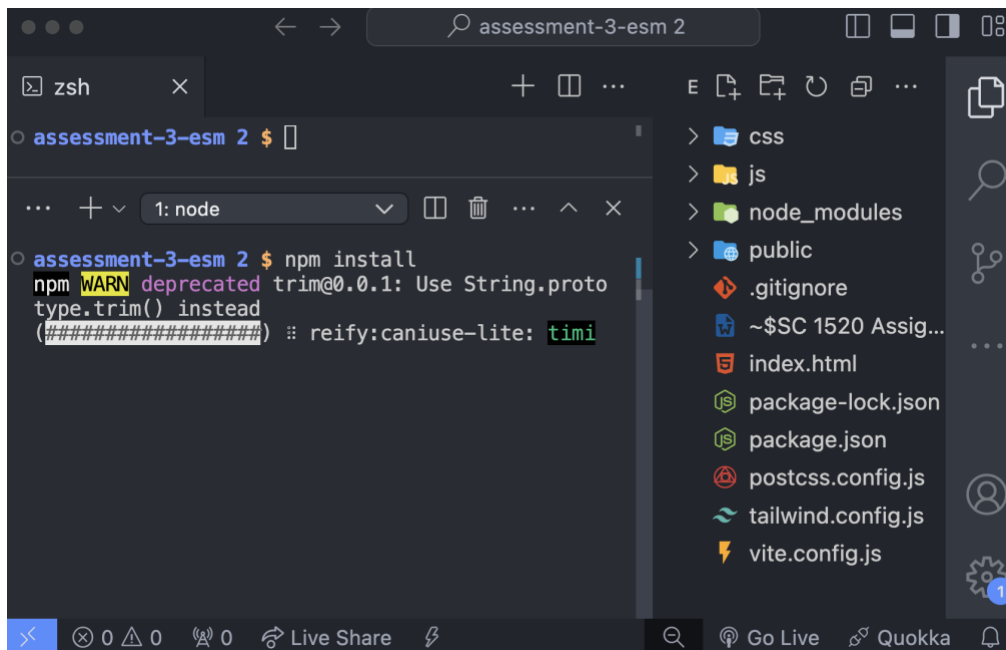
Before you fetch the album data you need to setup your node modules. Open the VS Code terminal. Make sure that you are that you are in the project folder. Inside the terminal issue an ls command you should see all of the project files as the output. If not, you are not in the root directory close vscode and open only the project folder.



```
assessment-3-esm 2 $ ls
css
index.html
js
package-lock.json
package.json
postcss.config.js
public
tailwind.config.js
vite.config.js
~$SC 1520 Assignment 3 - setup.docx
```

The screenshot shows the VS Code interface with a terminal window open. The terminal is in the directory 'assessment-3-esm 2'. The user has entered the 'ls' command, and the output lists the files and directories in the current directory: 'css', 'index.html', 'js', 'package-lock.json', 'package.json', 'postcss.config.js', 'public', 'tailwind.config.js', 'vite.config.js', and '~\$SC 1520 Assignment 3 - setup.docx'. The file explorer on the right shows the same files and directories.

Clear the terminal and then install the node modules using the npm install command.



```
assessment-3-esm 2 $ npm install
npm WARN deprecated trim@0.0.1: Use String.prototype.trim() instead
(#####) :: reify:caniuse-lite: timi
```

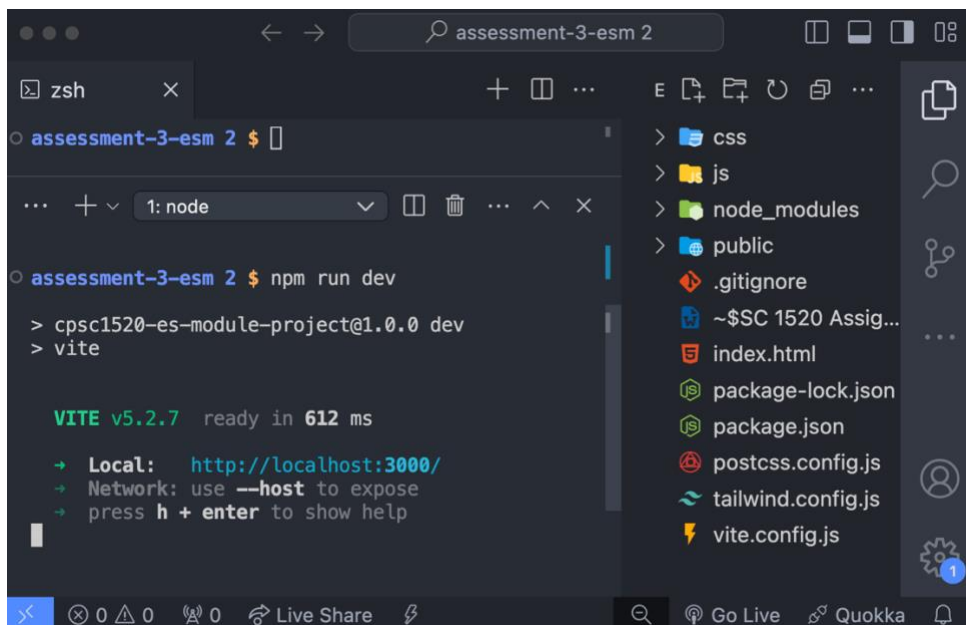
The screenshot shows the VS Code interface with a terminal window open. The terminal is in the directory 'assessment-3-esm 2'. The user has entered the 'npm install' command. The output shows a warning message: 'npm WARN deprecated trim@0.0.1: Use String.prototype.trim() instead'. Below the warning, there is a progress bar for the installation of 'reify:caniuse-lite' with the name 'timi'.

Task 11 Fetch the Album Data.

Open up the index.js file and create a fetch request to get the albums from the your albums resource on the mock api service. Replace the key shown in the image with your key/url.

```
async function appInit() {  
  const res = await fetch(['https://your-project-key.mockapi.io/api/v1/test'])  
  const payload = await res.json()  
}  
  
appInit()
```

Start the dev server using the **npm run dev** command in the terminal. Make sure that you see your data in the console.



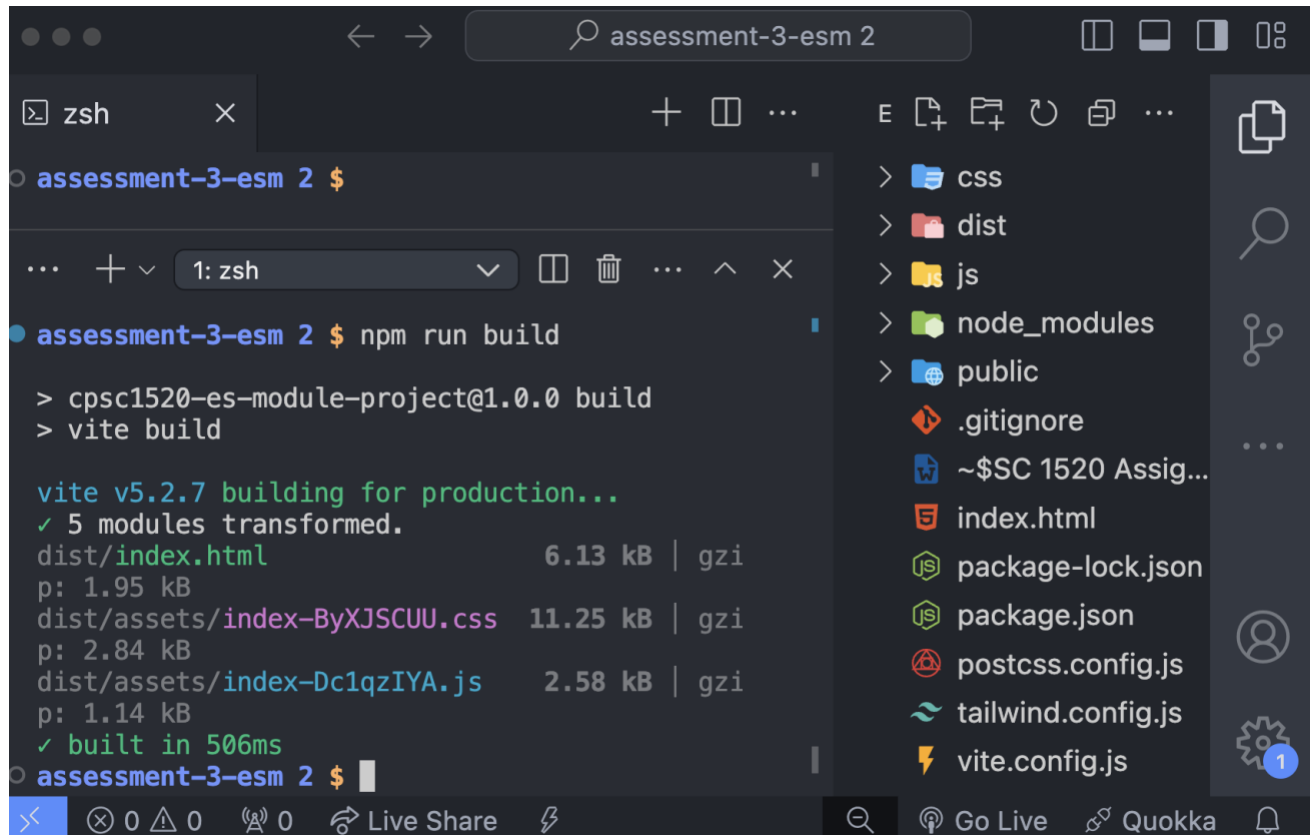
Console Output Example

[index.js:3](#)

```
▼ Array(75) i  
  ▶ 0: {id: '1.', albumName: 'OK Computer', artistName: 'Radiohead', releaseDate: '16 June 1997', g  
  ▶ 1: {id: '2.', albumName: 'Wish You Were Here', artistName: 'Pink Floyd', releaseDate: '12 Septe  
  ▶ 2: {id: '3.', albumName: 'In the Court of the Crimson King', artistName: 'King Crimson', releas  
  ▶ 3: {id: '4.', albumName: 'Kid A', artistName: 'Radiohead', releaseDate: '3 October 2000', genre  
  ▶ 4: {id: '5.', albumName: 'To Pimp a Butterfly', artistName: 'Kendrick Lamar', releaseDate: '15 |  
  ▶ 5: {id: '6.', albumName: 'Loveless', artistName: 'My Bloody Valentine', releaseDate: '4 Novembe
```

Task 12 Create a production build.

Once you see all 75 albums being fetched you can now run a production build of the lab. Open the vscode terminal and type **the npm run build** command. This will create a **dist** directory inside your project folder. This is the folder you will upload to Netlify.



The screenshot shows a VS Code editor window with a terminal open. The terminal is running a command prompt in a shell (zsh) within a project named 'assessment-3-esm 2'. The user has entered the command 'npm run build'. The output shows the build process for 'cpsc1520-es-module-project@1.0.0' using Vite v5.2.7. The build is successful, creating a 'dist' directory with three files: 'index.html' (6.13 kB), 'assets/index-ByXJSCUU.css' (11.25 kB), and 'assets/index-Dc1qzIYA.js' (2.58 kB). The build was completed in 506ms. The file explorer on the right shows the project structure, including the newly created 'dist' directory.

```
assessment-3-esm 2 $ npm run build

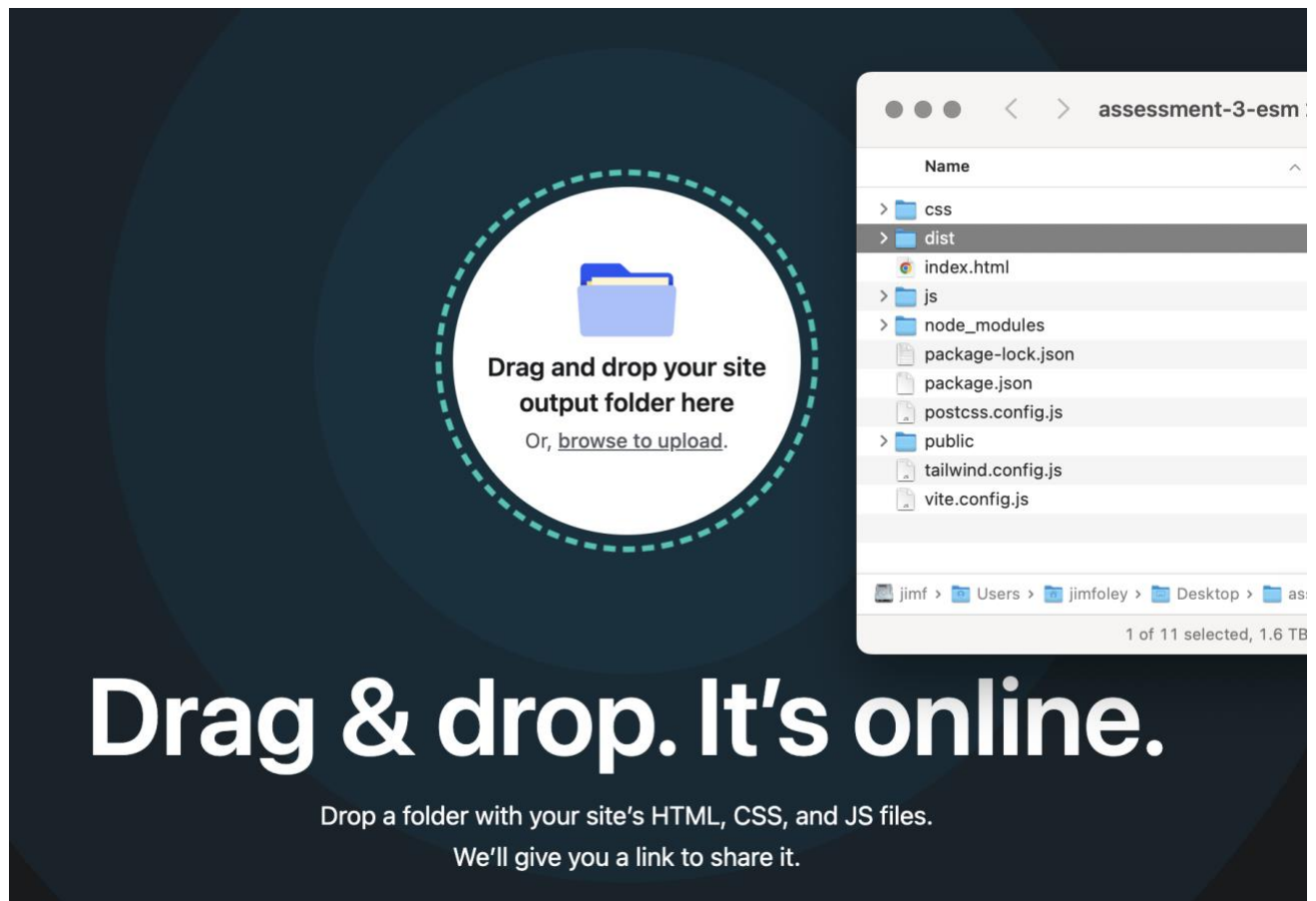
> cpsc1520-es-module-project@1.0.0 build
> vite build

vite v5.2.7 building for production...
✓ 5 modules transformed.
dist/index.html                6.13 kB | gzi
p: 1.95 kB
dist/assets/index-ByXJSCUU.css 11.25 kB | gzi
p: 2.84 kB
dist/assets/index-Dc1qzIYA.js  2.58 kB | gzi
p: 1.14 kB
✓ built in 506ms
assessment-3-esm 2 $
```

Task 13 Deploy to Netlify

Login to your Netlify account and create a new site deploy. **Drag and drop the dist** folder onto the deployment target. Make sure that you only add the dist folder or your site will not build as expected.

Check the production deployment link and make sure that you see the 75 albums listed in the console.



Submission

Add the following files and folders to Moodle for marking.

Netlify deployment URL

GitHub repository URL

Complete Lab 7 files as a zip file.

**KEEP YOUR FILES YOU WILL USE THEM FOR ASSIGNMENT 3
YOU ARE NOW READY TO START ASSIGNMENT 3.**