

Assignment 7

In this assignment we are going to set up a project from scratch with a scaffolding tool called “degit”. We will use Vite to replace Parcel for code bundling. We will call the Open Library API for searching book titles with certain keywords.

1. Scaffolding [1pt]

1. Open your terminal (Mac: iTerm2, Windows: WSL Ubuntu). Use the following command to create your project folder under your workspaces folder.

```
~/workspaces  
➤ mkdir assignment-7
```

2. Change directory into the newly created folder.

```
~/workspaces  
➤ cd assignment-7
```

3. Copy the `.devcontainer` folder from your previous assignment. Assume your assignment-6 folder is located at the same level as your assignment-7 folder.

```
~/workspaces/assignment-7  
➤ cp -R ../assignment-6/.devcontainer .
```

4. Verify the directory and files contained in this directory have been copied.

```
~/workspaces/assignment-7  
➤ ls -al .devcontainer  
total 8  
drwxr-xr-x  3 amoschen  staff  96 Oct 30 22:52 .  
drwxr-xr-x  3 amoschen  staff  96 Oct 30 22:52 ..  
-rw-r--r--  1 amoschen  staff  92 Oct 30 22:52 devcontainer.json
```

5. Open assignment-7 in VSCode using “Open Folder ...”. When prompted, choose “**Reopen in Container**” so that we can use the NodeJS environment to scaffold our project.

- Open a terminal panel in the VSCode. Then run the following command to set up the project using the [jvidalv/vital](#) template.

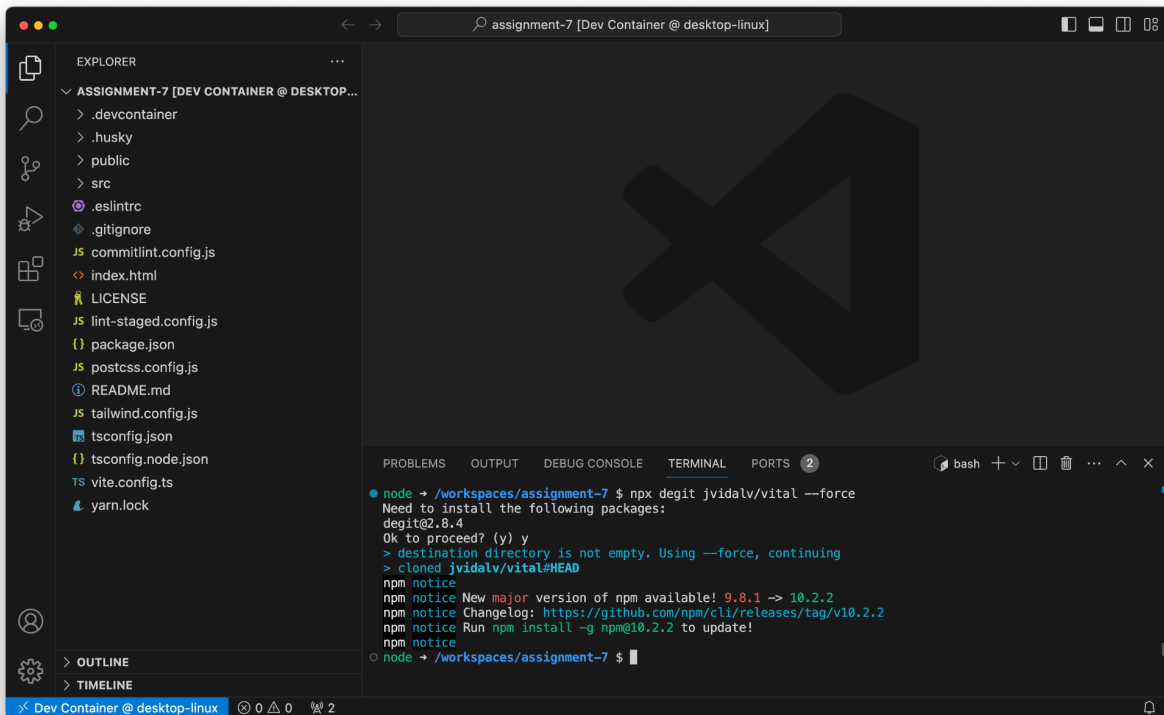
```
~/workspaces/assignment-7  
➤ npx degit jvidalv/vital --force
```

This command uses a scaffolding tool called “degit” to set up folder structure and configuration files according to the `jvidalv/vital` template.

- When prompted, please use “y” to proceed.

```
node ➔ /workspaces/assignment-7 $ npx degit jvidalv/vital --force  
Need to install the following packages:  
degity@2.8.4  
Ok to proceed? (y) y
```

The “--force” parameter is required because we have a `.devcontainer` folder. Otherwise it’s not needed when the folder is empty. In the instructions it asked to provide a project name which will in turn create a subdirectory in the assignment-7 folder which we don’t want. The scaffolding process may take some time. Please be passionate. Once it’s finished we will see the folder for assignment-7 looks like below.



It does have a lot of files!

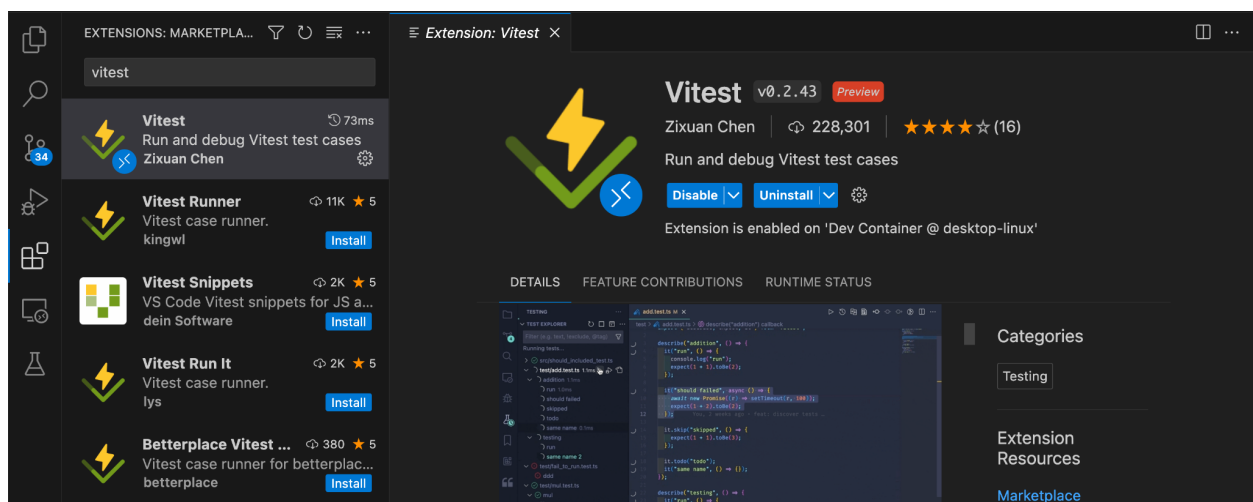
8. Before installing all the Node.JS modules let's initiate a Git repository for this folder.

```
~/workspaces/assignment-7  
➤ git init
```

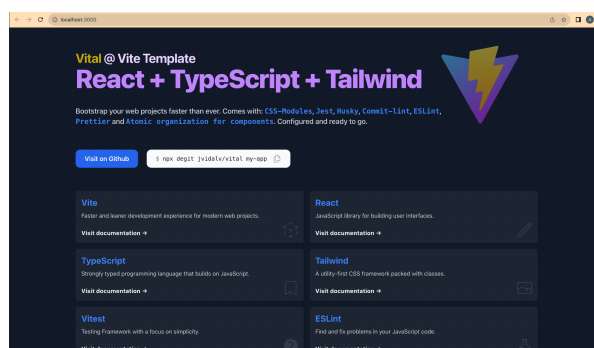
9. Then run the following command to start. This project has a git pre-commit check to ensure the commit message follows a certain format. So it's important to have a git repository initialized.
- yarn
 - yarn dev

Yarn is a relatively new package manager developed by Facebook. [Here](#) is a comparison between yarn and npm.

While waiting for packages being installed, let's add a new VSCode extension for better testing support.



Once the development server starts the browser should be launched automatically with the following page displayed.

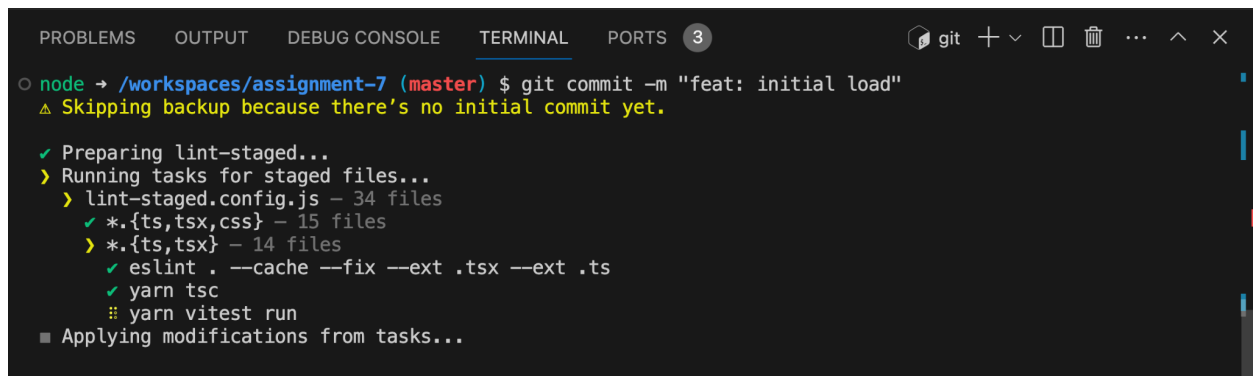


10. A few words about commit messages. It's more convenient to write commit messages following a consistent format. When looking back in the commit history messages following conventions helps us to find the place where certain changes were made faster. [Conventional Commits 1.0.0](#) is one of these conventions. "Husky" is the tool that enforces this convention.

Please run the following command to commit your initial load.

```
~/workspaces/assignment-7
> git add .
> git commit -m "feat: initial load"
```

Note, the "feat:" is required to start the commit message. Then you will see the pre-commit hook runs the ESLint, and other checking before the commit is added to git. This ensures that only the working code can be added to the git repository.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 3
○ node → /workspaces/assignment-7 (master) $ git commit -m "feat: initial load"
  ▲ Skipping backup because there's no initial commit yet.

  ✓ Preparing lint-staged...
  > Running tasks for staged files...
    > lint-staged.config.js - 34 files
      ✓ *.ts,tsx,css - 15 files
      > *.ts,tsx - 14 files
        ✓ eslint . --cache --fix --ext .tsx --ext .ts
        ✓ yarn tsc
        :: yarn vitest run
    ■ Applying modifications from tasks...
```

2. Search Books [1pt]

1. Go to the src/ folder and open the app/. Open the **app.tsx** file and replace it with the following content. This replacement app.tsx file is also available for downloading from the W9-Requesting Data1 assignment in Canvas.

```
import { ChangeEvent, MouseEvent, useState } from "react";

function App() {
  const [bookName, setBookName] = useState("");
  const [loading, setLoading] = useState(false);
  const [hasResult, setHasResult] = useState(false);
  const [booksCount, setBooksCount] = useState(null);

  const handleInputChange = (e: ChangeEvent<HTMLInputElement>) => {
    setBookName(e.target.value);
```

```

};

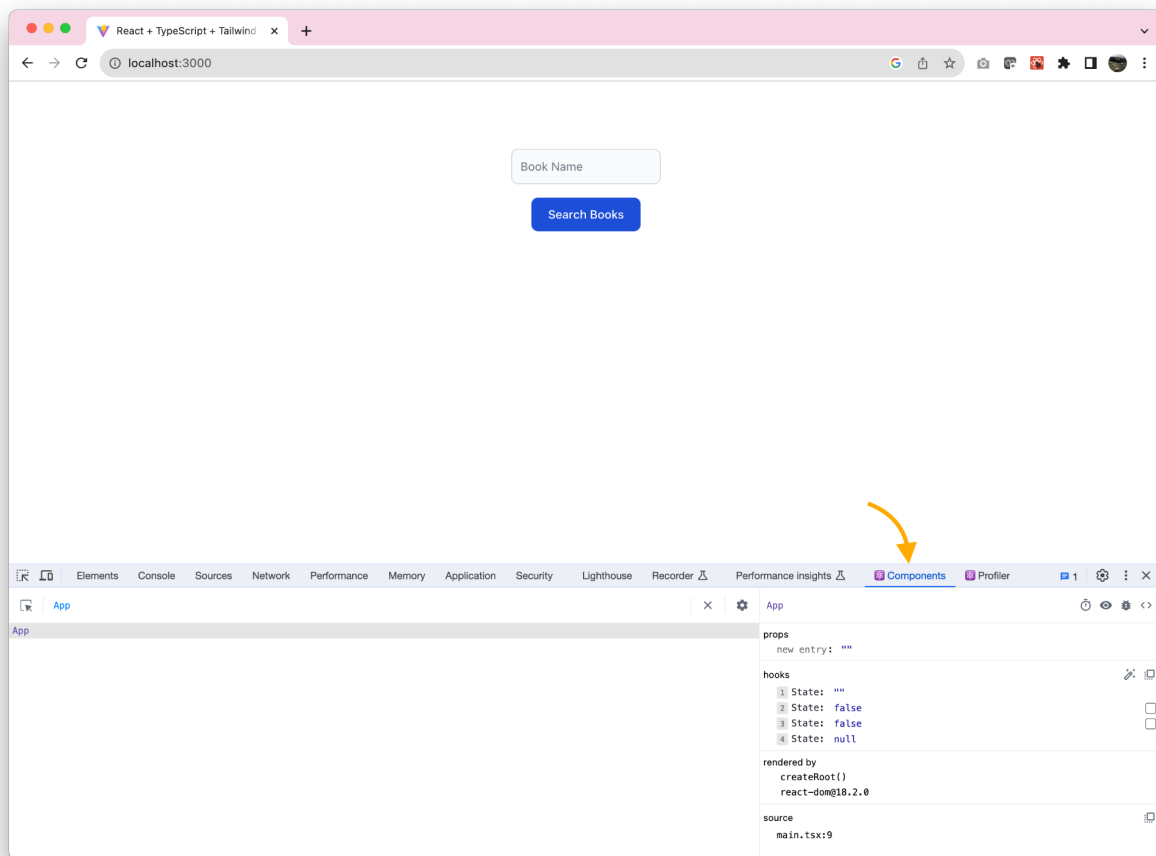
const handleClick = async (event: MouseEvent<HTMLInputElement>) => {
  event.preventDefault();
  setHasResult(false);
  const response = await fetch("");
  const data = await response.json();
  window.console.log(data);
  setLoading(false);
  setBooksCount(data.numFound);
};

return (
  <div className="flex min-h-[60vh] flex-col items-center justify-start gap-4 mt-20 text-center">
    <div>
      <input
        type="text"
        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"
        placeholder="Book Name"
        required
        value={bookName}
        onChange={handleInputChange}
      />
    </div>
    <button
      type="button"
      onClick={handleClick}
      className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"
    >
      Search Books
    </button>
    {loading ? <div>Loading ...</div> : ""}
    {hasResult ? <div>Found {booksCount} books</div> : ""}
  </div>
);
};
}

export default App;

```

2. After getting the project running the site should appear as the following. It's recommended to install the [React Developer Tools](#) as an extension to your browser. This improves your development experience with React.



3. Fetch Search Result [2pt]

Currently clicking the search button doesn't do anything. The URL is blank in the fetch() function. We are going to use the [Search API](#) and pass the value from the input text box into our search request.

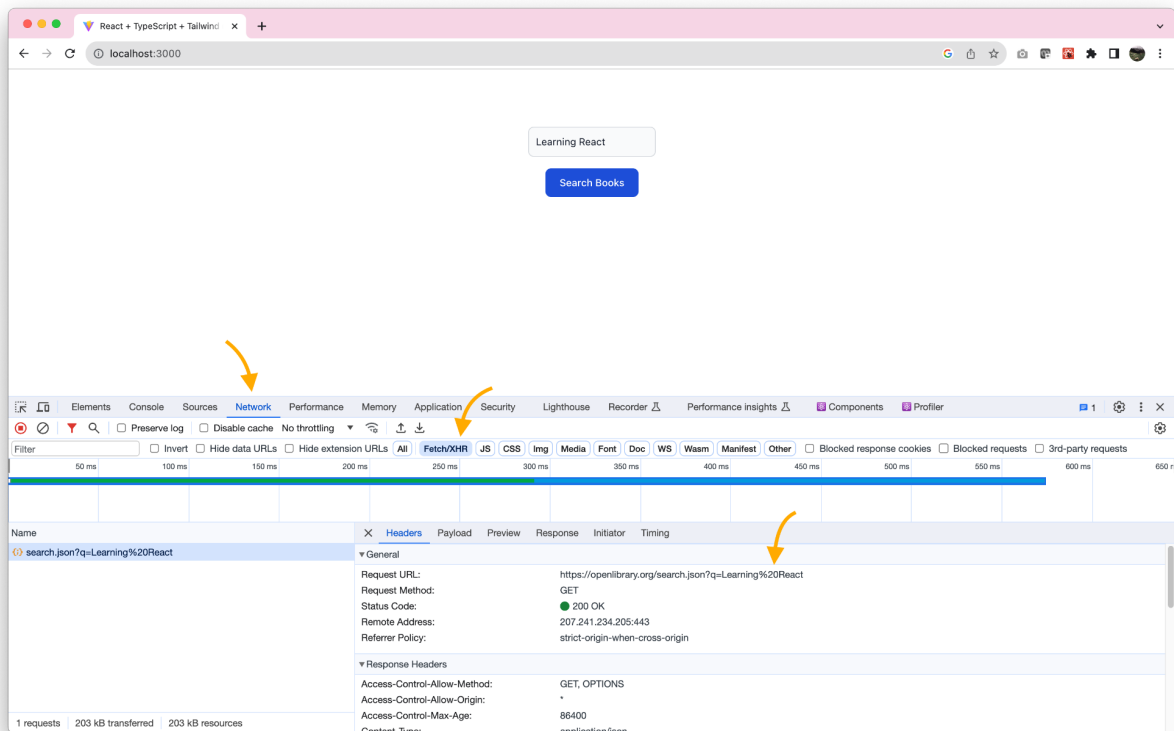
1. A change needs to be made to the following line.

```
const response = await fetch("");
```

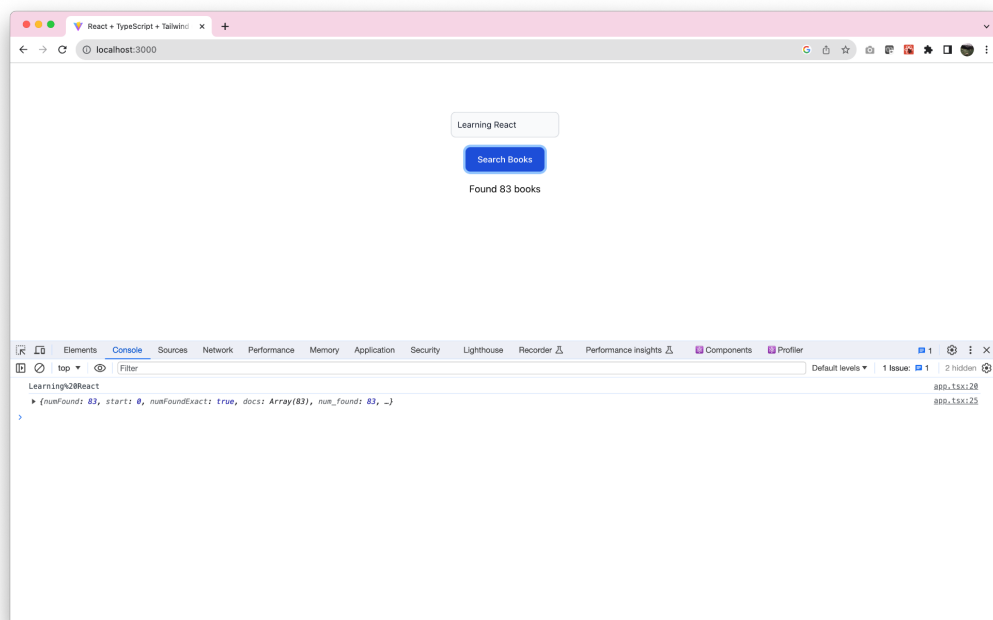
So that if I put "Learning React" in the text box I should form a request that looks like the following.

```
"https://openlibrary.org/search.json?q=Learning React"
```

Apart from using window.console.log() to verify what the URL looks like you can also use Chrome's Developer Tool to verify the HTTP request being sent.



If I switch to the Console view in the Developer Tool I should be able to see the returning data. But the page isn't displaying anything. Please make changes to the `handleClick` function so that the state variable `hasResult` and `loading` can be set appropriately.



The page should show "Loading ..." while waiting for the response to come back.

4. Deploy [1pt]

Please commit all your code to the master/main branch and push to your personal GitHub repository. Deploy it to Netlify. Please invite the instructor as a collaborator into your personal repository.

Instructor's GitHub username is **amoschenfamt**.

When submitting your assignment please include both Netlify URL as well as the GitHub repo link. Thanks

