CPSC1520 – JavaScript 7 Exercise: ES Modules

Introduction

A large part of using JavaScript is interacting and using the ecosystem of packages that are available. In this assessment we're going to setup a project so that we can use this ecosystem and use a common external npm package called validator (it just validates strings). We're going to validate a list of contacts and check if their email is valid or not based on the validators' isEmail function.

Exercise Step 1: Setup NPM, Packages and Parcel

1. Open your command line and go to the folder of your extracted in-class-assessment.
2. In your terminal (in the same folder as your index.html) run the command "npm init" and just press enter until a "package.json" file is in the same folder as your index.html file.
3. Install the required packages that we'll need for this project. In the terminal run the following commands.
   a. npm install validator
   b. npm install bootstrap
   c. npm install parcel -- save-dev
   Note: You need to see this in your dependencies for your package.json file like so (except newer versions most likely):

```
"dependencies": {
   "bootstrap": "^5.1.3",
   "validator": "^13.7.0"
},
"devDependencies": {
   "parcel": "^2.4.0"
}
```

4. In your package.json replace the scripts from what's there to the following.
   From:

```
"scripts": {
   "test": "echo \"Error: no test specified\" && exit 1"
},
```

   To this:

```
"source": "index.html",
"scripts": {
   "start": "parcel",
   "build": "parcel build"
},
```

5. When running your parcel web build with "npm run start" and opening the site at the port that parcel gives you you should see the following site. NOTE: If you're using liveserver from vs code for the rest of this assessment it just won't work.

Valid Users

No Valid Users Loaded.

6. In your "js" folder create an "index.js" file and link it in your html as follows.

```
<script type="module" src="js/index.js"></script>
```

Note: To ensure that your index.js is built and is using it successfully you should just display something for yourself. Only after you've done all of these steps you should actually move to the next step.

Exercise Step 2: Use ES Modules for styling and filtering active uses.

1. In your index.js import bootstrap so that you have some default styles. Referring to bootstraps' documentation you should be able to import it ([reference here](#)) as the first thing in your index.js like so:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Note: If everything is working successfully it should look like the image below (If it doesn't refer back to the previous steps):

Valid Users
No Valid Users Loaded.

2. In your index.js import your "main.js" using your knowledge of relative file paths (hint: it's in the same folder).
3. In your main.js import the "CONTACTS" from the "contacts.js" file. The "CONTACTS" variable is already exported so you should be able to access them.
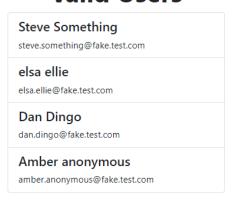
a. Once imported render all of the contacts on the page (using the renderContacts function), if successful it should look like the image below.

Valid Users

Steve Something
steve.something@fake.test.com

Bad Barry
bad.barry@fake

elsa ellie
elsa.ellie@fake.test.com

Dan Dingo
dan.dingo@fake.test.com

Amber anonymous
amber.anonymous@fake.test.com

Terry terrible
terrible.terry@.org.org

Jeff Jubilant
.jeff.jubilant@tester.com

4. There's some poorly formatted emails here so you're going to filter out users that have badly formatted emails. Thankfully we've imported the library/package "validator.js" we're going to use the "isEmail" function to filter out users that don't have a valid email.
   a. Go to the documentation here https://github.com/validatorjs/validator.js/#es6 and import validator the way it's described.
   b. In the "renderContacts" function instead of rendering all contacts only render the contacts that you know are valid using the "isEmail" function on the validator by passing in the email of the contact. (Hint: you could use filter if you want).
      The End result should look like the following:

Valid Users

Steve Something
steve.something@fake.test.com

elsa ellie
elsa.ellie@fake.test.com

Dan Dingo
dan.dingo@fake.test.com

Amber anonymous
amber.anonymous@fake.test.com

Once complete you can go to the

Exercise Step 3: Upload and Zip your site to moodle.

1. Once Complete, stop the parcel process, and delete the parcel-cache folders and node_modules folders.
2. Zip up your project and ensure that your entire project is included.
3. Submit that to moodle.
   Note if it's too large then you probably have included your node_modules (you'll get no marks if they're included anyways because you read the Grading section below so you'll delete it.)

Grading

I'll give full marks if:

- There is NO node_modules and NO .parcel_cache folders included in the project.
- Everything is imported correctly.
- The contacts file is untouched and not manipulated.
- The end result looks like the following picture

If you don't follow the instructions you'll get a zero. There are no marks in between.