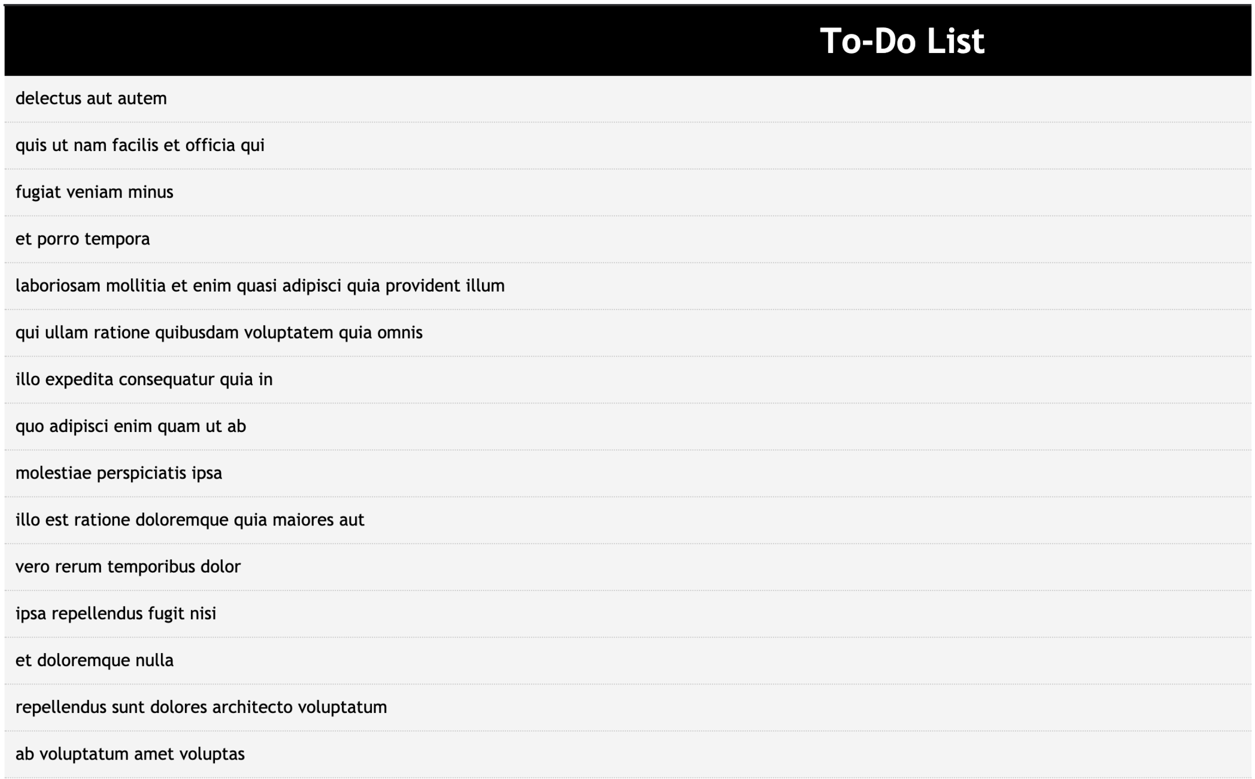
DEVPSU **Project Details**

React JS Project

Due: March 30, 5:59 PM EST

Welcome to your second project of DevPSU Learning. In this project, you are going to build a **To-Do List React application** that renders dummy JSON data from a fake API (https://jsonplaceholder.typicode.com/todos).

Once done, the web application will look something like this:



To-Do List

delectus aut autem

quis ut nam facilis et officia qui

fugiat veniam minus

et porro tempora

laboriosam mollitia et enim quasi adipisci quia provident illum

qui ullam ratione quibusdam voluptatem quia omnis

illo expedita consequatur quia in

quo adipisci enim quam ut ab

molestiae perspiciatis ipsa

illo est ratione doloremque quia maiores aut

vero rerum temporibus dolor

ipsa repellendus fugit nisi

et doloremque nulla

repellendus sunt dolores architecto voluptatum

ab voluptatum amet voluptas

Getting Started

Step 1

In order to successfully compile a React application, you need to install several packages and modules. We will be using the Create React App environment for learning React, as it sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. **You don’t need to install anything just yet.**

Additionally, you’ll need to have Node.js on your machine. **You need to install this at** https://nodejs.org/en/. Once downloaded, open your preferred IDE (such as VSCode, IntelliJ etc.) to create the project.

Step 2

In the terminal inside your IDE run:

```
npx create-react-app <your-app-name>
```

For example,

```
npx create-react-app my-first-app
```

Note that npm will take a while to download all dependencies and modules. Once this installation is done, navigate to inside your app folder using the below command.

```
cd my-first-app/src
```

Step 3

Just to make sure you’ve installed everything properly, run the below command to see boilerplate React code getting deployed!

```
npm start
```

Step 4

For developing this application, inside the src folder you’ll need to create separate JS files to put your code in. Finally, when you’re ready to deploy to production, run the below command to see your application in action!

```
npm start
```

Project Description and Requirements

The aim of this project is to practice fetching, cleaning (JSON to text) and rendering data on the frontend using API calls. For the purpose of this project we are using a fake API with dummy JSON data, as mentioned above. You are required to use the concept of “everything-is-a-component” within React to accomplish this task.

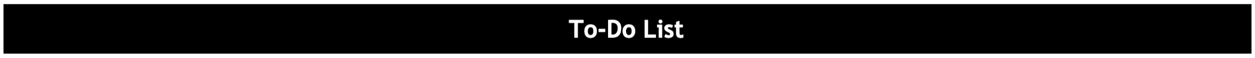
You need to submit three JavaScript files whose descriptions are as below:

App.js

This JavaScript file is already present in your application when you first run create-react-app and acts as the top-level JS file. You need to **research and implement** the concept of **npm-axios()** and a React component called **componentDidMount().** Using these together, you need to make a call to the fake API and store its response. Additionally, in order to practice the “component structure” of React, you’ll need to render the other two JS files (as described below) within App.js.

Header.js

This JavaScript file will allow you to practice some basic level CSS as it forms a core part of web/application development. In this file, you need to design and render the header of our application. Feel free to explore this CSS documentation to style your header. An example header would look like:



To-Do List

Todos.js

This JavaScript file will allow you to practice the use of user-defined functions within your React code. Additionally, this is where you’ll be looping through and rendering the first 30 “title” of the data you receive from our API.

Notice that the API is returning a list of data which look like:

```
 {
    "userId": 1,
    "id": 17,
    "title": "quo laboriosam deleniti aut qui",
    "completed": true
  },
```

Use the below provided getStyle() function to style your title while you render it. Again, feel free to play around with this.

```
class Todos extends Component {
 getStyle = () => {
   return {
     background: "#f4f4f4",
     padding: "10px",
     borderBottom: "1px #ccc dotted",
   };
 };
  ………… {* Todos.js code implemented by you *}
}
```

Challenge (Optional)

Kudos to you if you were able to create the basic implementation of our To-Do list! If you further want to practice React development and earn some extra-credit, let's get you started!

In this portion of the project (optional) you would be to improve the usability of our app, by allowing the users to be able to add new tasks in their To-Do lists. You need to accomplish this by adding a **textbox and a submit button** just below the header.

You will need to **research and implement** the concept of React forms and React state to successfully complete the implementation. You might also have to look into additional resources or tutorials.

Submission

GitHub (Recommended)

You will need to submit a link to your **public** GitHub repository which contains your application. Please note that you need to submit all the files and folders and not just the above-mentioned JS files.

Additionally, make sure to **add a new file called ".gitignore"** in your application. You'll just need to copy paste the below:

```
# See https://help.github.com/articles/ignoring-files/ for more about
ignoring files.

# dependencies
/node_modules
/.pnp
.pnp.js

# testing
/coverage

# production
/build

# misc
.DS_Store
.env.local
.env.development.local
.env.test.local
```

```
.env.production.local

npm-debug.log*
yarn-debug.log*
yarn-error.log*
```

Adding this file will ignore unnecessary files and folders. **Add it outside the src folder**, that is on the same level as package-lock.json and package.json files, before you finally push your code on GitHub.

Canvas Submission

You can also compress your application folder and submit a .zip file on Canvas. No need to add the .gitignore file in this case.

Help

If you have any questions or concerns, please do not hesitate to reach out to us at devpsu@acm.psu.edu or through Canvas.