# CSE 5234 – Distributed Enterprise Computing

Lab 5 – Building presentation tier of a web application

# Objective

Build a simple e-commerce web site which allows you to select from a list of available items for purchase, enter payment (i.e. credit card) and shipping information, and submit your order. See detailed requirements, and implementation instructions using ReactJS library in subsequent sections. Note, this lab is (and all subsequent labs are) a **group** project.

# Detailed Requirements

1. The website will have 5 pages. The data items to present on the page, the data entry fields, and the functionality to support on each page is summarized below.

|  |  |  |
| --- | --- | --- |
| URL | Required info | function |
| /purchase | list of items. Item contains name, price, and quantity fields | Allow users to select items and quantities they want to purchase. Save it in shopping cart. |
| /purchase/paymentEntry | credit card number, expiration date, cvvCode, and card holder name | Allow users to enter credit card information. Save it. |
| /purchase/shippingEntry | name, addressLine1, addressLine2, city, state, and zip as properties. | Allow users to provide shipping details. Save it. |
| /purchase/viewOrder |  | Present a summary of items selected by user and total cost of the order. Ask the user to validate and confirm the order. |
| /purchcase/viewConfirmation |  | display greetings, confirmation number, and order summary. |

1. On the purchase page, display a catalog of 5 items to sell. Do not worry about making this list dynamic. The data on the page can be hardcoded. We will return to clean this up in a subsequent lab, when we build the business tier & database tier (i.e., the backend).
2. On the purchase page, preferably display an image along with the name of the item. Allow the user to enter the quantity they want to purchase and add to the shopping cart.
3. Support a shopping cart, which will collect user specified items and quantities to purchase. User can click on it at any time and modify the order (i.e., change quantity & cancel line item).
4. Build pages for users to enter payment and shipping information. Save all the user supplied data in the DOM (or supporting object).

# Implementation

Please note that this lab can be implemented using a web application framework of your choice. Brief instructions are provided for building using React library. Please take the time to learn, design and build your web application. Consider the implementation instructions provided here as suggestions. Do not become a code monkey!

Talk to the instructor if you want to use a server-side framework such as Spring MVC – detailed instructions are available.

# Implementation with React JS Library

## Resources:

React tutorial (<https://reactjs.org/tutorial/tutorial.html>

React styling and css (<https://reactjs.org/docs/faq-styling.html>)

React Router v6 (<https://reactrouter.com/en/v6.3.0/getting-started/tutorial>)

## Prerequisite:

Nodejs: <https://nodejs.org/en/>

npm install react-router-dom@6

npm install react-router-dom  
npm install --save styled-components

npm install react-icons

npm install mysql –save

npm install sync-mysql --save

npm install express --save

npm install axios

npm install cors

## Implementation – Basic Set Up

(1) Create a new project using [**create-react-app**](https://www.geeksforgeeks.org/reactjs-setting-development-environment/) so open your terminal and type:

npx create-react-app <project name>

<project name> should be named after the business venture your team decided to get into.

(2)Now go to your folder by typing the given command in the terminal:

cd <project name>

(3) Run this command to execute the React application my-react-app:

npm start

A new browser window will pop up with your newly created React App! If not, open your browser and type localhost:3000 in the address bar.

## Implementation - Building Pages

1. Create components folder in (/<project name>/src) and 5 blank JavaScript files (\*.js) under /<project name>/src folder.

a. purchase.js

b. paymentEntry.js

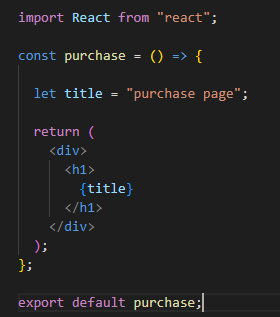
c. shippingEntry.js

d. viewOrder.js

e. Confirmation.js

2. Edit various pages in the project.

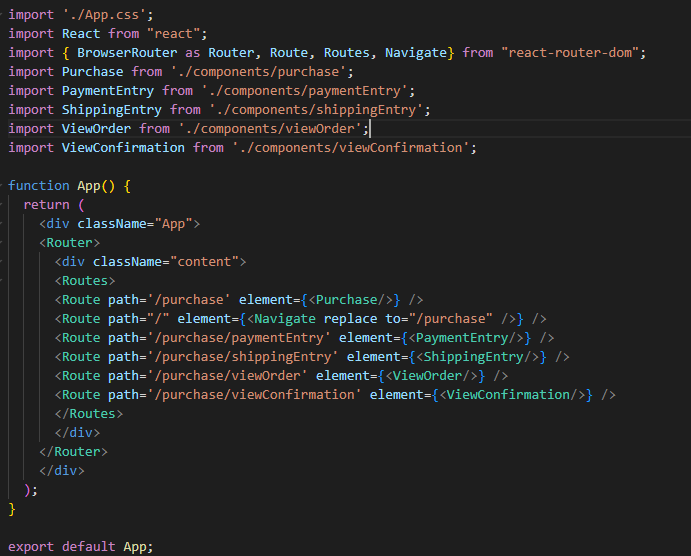
ex., purchase.js looks like this:



where title is the variable defined in the js file and it would be used when the function is called.

3. Open app.js and import react library and other components. Next, import React router components. (The purpose of the App component is to handle all the screens in one component. It will configure routing and enable navigation to all other components.)

It should look like this:



4. Next, serve the application using the npm command. (i.e., npm start) Now, you can view created pages by entering [http://localhost:3000/<path](http://localhost:3000/%3cpath)> (ex. <http://localhost:3000/purchase>) In the browser.

5. Develop purchase.js to allow collect user requested quantity and then direct to the next page (/purchase/paymentEntry). Here, we demonstrate how to use “useState”, “forms” and “useNavigate” react components to achieve it. (For more information, pls refer to <https://reactjs.org/docs/forms.html>, <https://reactjs.org/docs/hooks-state.html,> <https://reactrouter.com/en/main/hooks/use-navigate>)

The result would look like this:

A screen shot of a computer program

Description automatically generated

Besides, we need to add the “useLocation” component (<https://reactrouter.com/en/main/hooks/use-location>) in paymentEntry.js to get the variable sent from /purchase. It would look like this:

A screen shot of a computer program

Description automatically generated

6. Develop paymentEntry.js, shippingEntry.js to collect user payment information (i.e., credit card number, expiration date, cvvCode, and card holder name) and shipping information (i.e., name, addressLine1, addressLine2, city, state, and zip as properties). Besides, you should redirect users to the next page.

8. Develop viewOrder.js to display order details (i.e., items with user requested quantities), payment information, and shipping details.

9. DevelopConfirmation.js which thanks the user for completing the order and displays a unique confirmation number / code. Hard code this confirmation number for now.

10. Deploy, run (npm start), and test. If needed, use config.log("") for debugging.

A screenshot of a computer

Description automatically generated with medium confidence

Console.log(“tmp”);