



Etsy Prototype - Lab 1

Snigdha Chaturvedi | 015957435

Youtube Demo: <https://www.youtube.com/watch?v=T60fTZe3tNE>
Git Repository: <https://github.com/Snichat97/EtsyPrivateAWS>

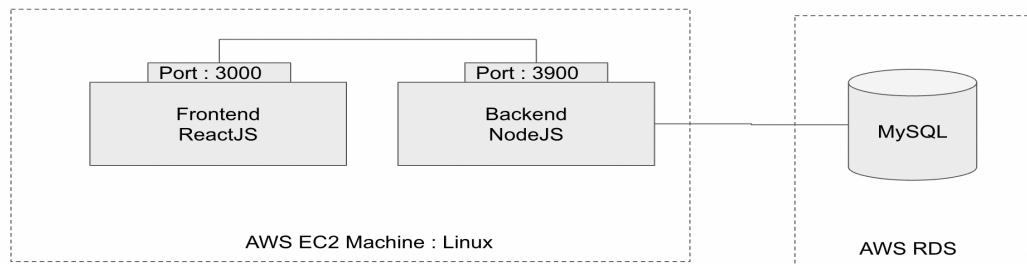
Goals and Purpose:

My goals from this project are to design and deploy an end-to-end application and explore in-depth backend API construction, various deployment strategies, and using the Redux library for state management across multiple components. The system is a prototype of the “Etsy” Application which allows users to buy products and host their own shops.

Salient features of the System:

The application uses ReactJS for the frontend and NodeJS running on Express Server for the backend. A relational MySQL database is used.

An AWS EC2 instance holds the complete application. On port 3000 the frontend of the application is hosted and on 3900 the backend is running. A separate RDS has been created to hold the SQL database.



Important Libraries:

Backend Libraries

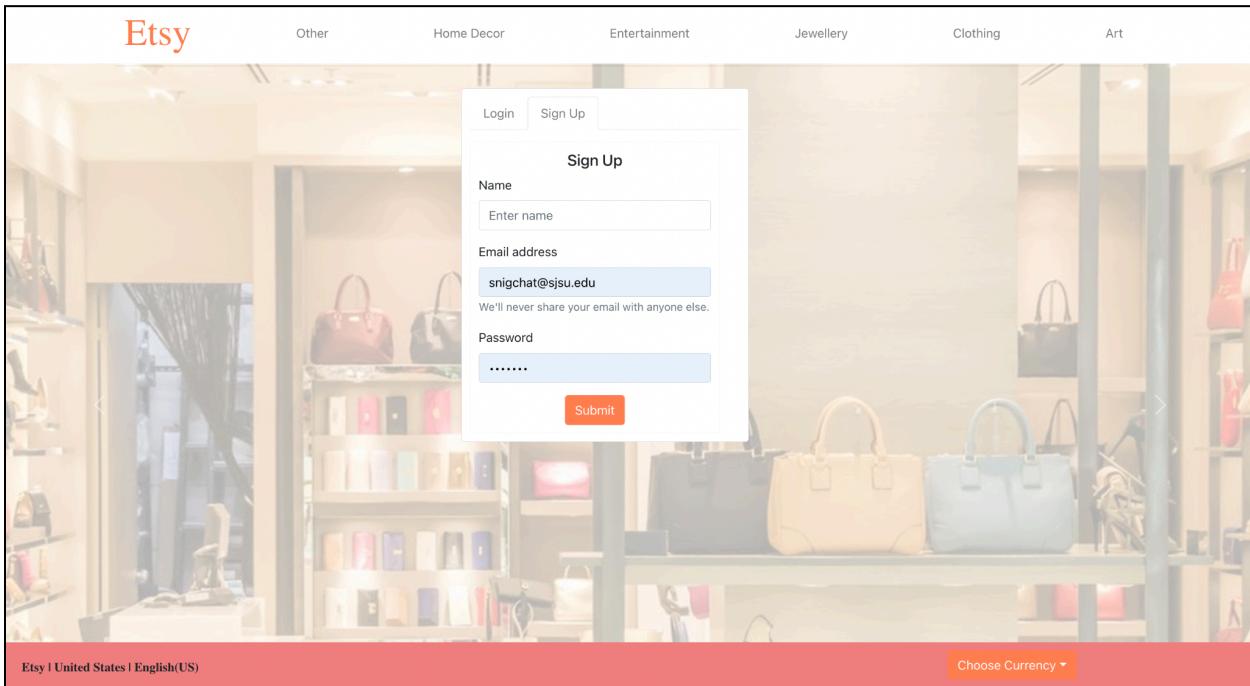
- "bcrypt": "^5.0.1" - To decrypt passwords
- "mocha": "^9.1.2" - For testing purposes
- "mysql2": "^2.3.3" - To establish connection between the backend and SQL server

Frontend Libraries

- "@testing-library/user-event": "^13.5.0" - For testing purposes of frontend components
- "axios": "^0.26.0" - For communication between the frontend and backend
- "bcryptjs": "^2.4.3" - To encrypt passwords
- "react-bootstrap": "^2.1.2" - For maintaining styling
- "react-redux": "^7.2.6" - For maintaining global state
- "react-router-dom": "^6.2.1" - For routing between components
- "redux": "^4.1.2" - For maintaining global state

Screen Previews

Login / Signup Screen



Encrypted Password

A screenshot of a network traffic capture tool, likely Wireshark, showing a request payload. The payload contains an object with an email and password field. The email field is "snigchat@sjtu.edu" and the password field is "U2FsdGVkX19gykkBG+DUJ/C9/rKoogVYp+BrqZXtUoI=". The interface includes tabs for Headers, Payload, Preview, Response, Initiator, and Timing. The Payload tab is selected, and the Request Payload section is expanded to show the JSON-like structure of the data.

Profile Page / Favorites Page- Basic:

The screenshot shows the user's profile page on Etsy. At the top, there is a search bar and a navigation menu with categories like Other, Home Decor, Entertainment, Jewellery, Clothing, Art, and Logout. A notification icon in the top right corner shows '2'. Below the header, there is a profile picture of a woman and her contact information: Snigdha2, USA, 33 s third st #411, san jose, 95115. There is also a link to 'See Purchase History' and a pencil icon for editing. The main content area is titled 'Favourite Items' and includes search, filter, and sort options. Below this, there are sections for 'Minimum Price', 'Maximum Price', and 'Submit'. The bottom of the page features a red footer bar with links for 'Etsy | United States | English(US)' and 'Choose Currency ▾'.

The screenshot shows the user's favorites page on Etsy. The interface is similar to the profile page, with a 'Favourite Items' section at the top featuring search, filter, and sort options. Below this, there is a grid of five items, each with a heart icon indicating it is a favorite. The items listed are:

- Turtle (\$30) - Category: ENTERTAINMENT
- Diamond Stud Earrings (\$1000) - Category: JEWELLERY
- Juhি Scent Box (\$80) - Category: OTHER
- Mainstays 8-Piece Black Bed (\$100) - Category: HOME DECOR
- A small image of a dessert, likely baklava, which is partially visible.

The bottom of the page features a red footer bar with links for 'Etsy | United States | English(US)' and 'Choose Currency ▾'.

Profile Page - Update:

The screenshot shows the 'Profile Update' section of an Etsy profile. The form contains the following fields:

- Name: Snigdha2
- Email: snigchat@sjtu.edu
- Image: A URL pointing to an image file.
- Phone Number: 6692137769
- Date of Birth: 03/17/2022
- Country: USA
- Street: 33 s third st #411
- City: san jose
- Pincode: 95115

A red 'Submit' button is located at the bottom right of the form.

Sell on Etsy Page - First time:

The screenshot shows the 'Name your Shop' step on the Etsy sign-up process. At the top, there's a navigation bar with links for Other, Home Decor, Entertainment, Jewellery, Clothing, Art, and a Logout button. To the right of the navigation are icons for a heart, a shopping cart with a '2' notification, and a user profile. Below the navigation is a search bar with the word 'Search'. The main form area has a title 'Name your Shop' and a subtitle 'Choose a memorable name that reflects your style.' A 'Shop Name' input field contains 'Snigdha Art Store'. Next to it is a small icon with a checkmark. Below the input field are two buttons: 'Check Availability' (orange) and 'Submit' (orange). The entire form is enclosed in a light gray box.

This screenshot shows the same 'Name your Shop' step, but the shop name has been changed to 'Mugdha Art Store'. The rest of the interface is identical to the first screenshot, including the navigation bar, search bar, and form layout.

Shop Home Page:

The screenshot shows the homepage of a shop on the Etsy platform. At the top, there's a navigation bar with categories like Other, Home Decor, Entertainment, Jewellery, Clothing, and Art, along with a search bar and a user profile for 'Snigdha2'. Below the navigation, a featured product is displayed: 'Paddys Mystery Box' with a total sales count of 80. A button to 'Add a Product' is visible next to it. To the right, there's a profile picture of a woman and contact information for the shop owner. A large orange button labeled 'See Purchase History' is present. Below this, a section titled 'Products' contains search and filter options, and buttons for 'Minimum Price', 'Maximum Price', and 'Submit'. At the bottom of the page, there are links for 'Etsy | United States | English(US)' and 'Choose Currency ▾'.

Adding and editing an item to a shop:

This screenshot shows the same shop homepage as above, but with an 'Add Product' modal window overlaid. The modal has fields for Name ('Scent Box -Viva La Juicy Gold Couture'), Add image ('/scentbox.com/images/product/thumbs/Ari-w.jpg'), Category ('Other'), Description ('Viva La Juicy Gold Couture'), Price ('30'), and Quantity ('20'). A 'Submit' button is at the bottom right of the modal. The background of the shop page is dimmed to indicate the modal is active.

See Purchase History

Products

Search Filter Sort

Minimum Price Maximum Price Submit



Superhero Gift Box, Gift Bo ...
\$ 80
Total Sold : 1 ENTERTAINMENT



Juhi Scent Box
\$ 80
Total Sold : 0 OTHER



Scent Box -Viva La Juicy Go ...
\$ 30
Total Sold : 0 OTHER

Etsy | United States | English(US) Choose Currency ▾

See Purchase History

Edit Product

Name
Juhi Scent Box

Add image
<https://i.guim.co.uk/img/media/177f871940f3dab/>

Category
OTHER

Description
The American Kennel Club has voted the labrador

Price
80

Quantity
20

Submit

Superhero Gift Box, Gift Bo ...
\$ 80
Total Sold : 1 ENTERTAINMENT

Juhi Scent Box
\$ 80
Total Sold : 0

Etsy | United States | English(US) Choose Currency ▾

Dashboard – Landing page:

The screenshot shows the Etsy dashboard landing page. At the top, there's a search bar with the word "Search" and a placeholder "Search". To the right of the search bar are icons for a heart (2), a shopping cart, and a user profile. Below the search bar is a navigation menu with categories: Other, Home Decor, Entertainment, Jewellery, Clothing, Art, and a Logout button. A large orange banner at the top says "Welcome back, Snigdha2 !". Below the banner are buttons for "Search", "Filter", and "Sort". Further down are buttons for "Minimum Price", "Maximum Price", and "Submit". The main content area displays four items in a grid:

- Mainstays 8-Piece Black Bed (\$100) - Total Sold: 2, HOME DECOR
- Mainstays 8-Piece Black Bed (\$80) - Total Sold: 7, ENTERTAINMENT
- Diamond Stud Earrings (\$1000) - Total Sold: 19, JEWELLERY
- Turtle (\$30) - Total Sold: 2, ENTERTAINMENT

At the bottom of the page, there are links for "Etsy | United States | English(US)", social media icons for Spotify and Apple TV, and a "Choose Currency" dropdown.

Searching Landing Page:

The screenshot shows the Etsy searching landing page with the search term "beautiful" entered into the search bar. The rest of the interface is identical to the dashboard landing page, including the navigation menu, the "Welcome back, Snigdha2 !" banner, and the item grid. The items listed are:

- Mainstays 8-Piece Black Bed (\$80) - Total Sold: 7, ENTERTAINMENT
- Diamond Stud Earrings (\$1000) - Total Sold: 19, JEWELLERY
- Pao Bhaji (\$30) - Total Sold: 0, OTHER
- Mainstays 8-Piece Black Bed (\$1000) - Total Sold: 0, ENTERTAINMENT

At the bottom of the page, there are links for "Etsy | United States | English(US)", social media icons for Spotify and Apple TV, and a "Choose Currency" dropdown.

Shopping Item Overview page:

The screenshot shows the item details for a perfume bottle. The product image is a faceted pink bottle with a silver cap, labeled 'ARTAN'. To its left is a white spherical object. On the right, the product title is 'Scent Box -Viva La Juicy Gold Couture' with a heart icon. Below it is the brand name 'Viva La Juicy Gold Couture'. The price is listed as '\$ 30' with a currency dropdown showing '1'. An 'Add to Cart' button is at the bottom right. The top navigation bar includes categories like Other, Home Decor, Entertainment, Jewellery, Clothing, Art, and a Logout button.

Cart Page:

The screenshot shows the shopping cart containing three items. The first item is a perfume bottle labeled 'Scent Box -Viva La Juicy Gold Couture' with a quantity of 3 and a price of \$90. A 'Complete Purchase' button is at the bottom right. The top navigation bar includes categories like Other, Home Decor, Entertainment, Jewellery, Clothing, Art, and a Logout button.

My Purchases page:

The screenshot shows the 'Order Listing' section of the Etsy My Purchases page. It displays two separate orders.

Order Number : 6

- Mainstays 8-Piece Black Bed**
Quantity : 2
Price : 200
Shop Name : Shop 1
- Blue Sapphire Teardrop Pendant with Diamond**
Quantity : 11
Price : 19800
Shop Name : Suchetas Jewelery !
- Turtle**
Quantity : 3
Price : 90
Shop Name : Shradha Pet Shop

Order Number : 7

- Turtle**
Quantity : 3

Etsy | United States | English(US) Choose Currency ▾

Orders can't be placed without the user adding his address

A modal window is displayed in the center of the screen, overlaid on the 'Cart' section. The modal has a dark background and contains the following text:

localhost:3000 says
Please fill in your address details.

OK

The 'Cart' section below the modal shows the following items:

Cart

- Diamond Stud Earrings
Quantity: 1
Price: 1000

Complete Purchase

Etsy | United States | English(US) Choose Currency ▾

Footer with currency change option

Welcome back, Snigdha2 !

Search Filter Sort

Minimum Price Maximum Price Submit



Mainstays 8-Piece Black Bed
₹ 100
Total Sold : 2 HOME DECOR



Mainstays 8-Piece Black Bed
₹ 80
Total Sold : 7 ENTERTAINMENT



Diamond Stud Earrings
₹ 1000
Total Sold : 19 JEWELLERY



Turtle
₹ 30
Total Sold : 2 ENTERTAINMENT



Etsy | United States | English(US)
localhost:3000/home#



Ethan



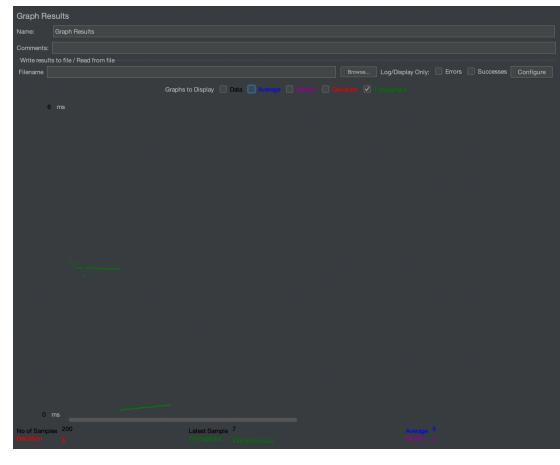
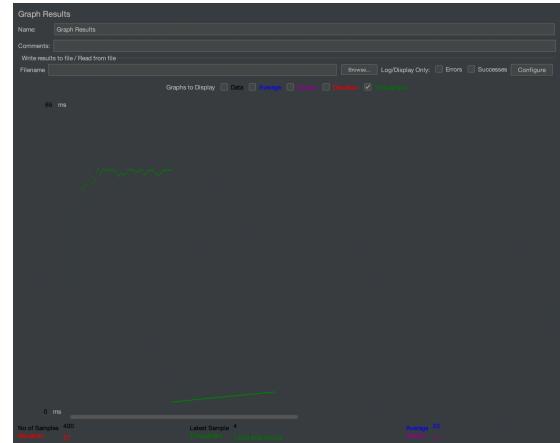
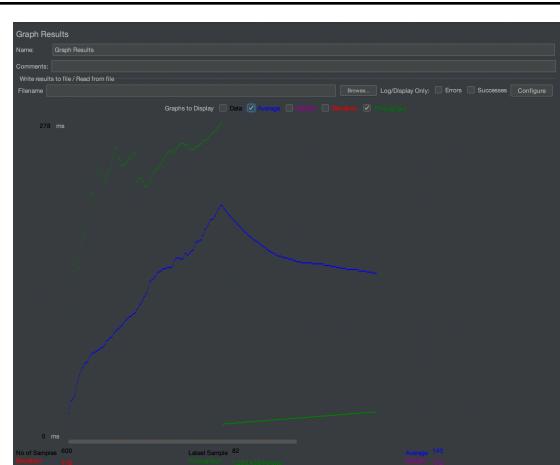
Choose Currency ▾



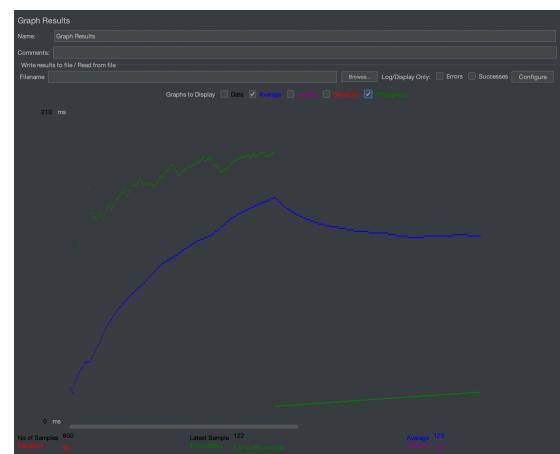
INR
USD

Testing

1. JMeter Testing

Number of concurrent users	Graph pooling vs single connection
100 Users	 A screenshot of the JMeter Graph Results interface for 100 concurrent users. The graph shows a distribution of response times. The x-axis represents response time in milliseconds (ms), ranging from 0 to 5 ms. The y-axis represents the number of samples. A green line represents the distribution, which is relatively flat and centered around 0 ms. Key statistics at the bottom of the graph are: No. of Samples: 200, Deviation: 0 ms, Latest Sample: 7 ms, Throughput: 14.3 samples/sec, Average: 5 ms, and Median: 3 ms.
200 Users	 A screenshot of the JMeter Graph Results interface for 200 concurrent users. The graph shows a distribution of response times. The x-axis represents response time in milliseconds (ms), ranging from 0 to 5 ms. The y-axis represents the number of samples. A green line represents the distribution, which is slightly more spread out than at 100 users. Key statistics at the bottom of the graph are: No. of Samples: 400, Deviation: 37 ms, Latest Sample: 4 ms, Throughput: 10.0 samples/sec, Average: 10 ms, and Median: 11 ms.
300 Users	 A screenshot of the JMeter Graph Results interface for 300 concurrent users. The graph shows a distribution of response times. The x-axis represents response time in milliseconds (ms), ranging from 0 to 279 ms. The y-axis represents the number of samples. A blue line represents the distribution, which is significantly more spread out and shifted towards higher values compared to 100 and 200 users. Key statistics at the bottom of the graph are: No. of Samples: 600, Deviation: 116 ms, Latest Sample: 82 ms, Throughput: 7.4 samples/sec, Average: 149 ms, and Median: 115 ms.

400 Users



500 Users



2. Mocha Testing

```
JS db.js JS test.js X ProductItem.jsx U # App.css M Login.jsx U JS query.js JS user.js .../routes JS user.js .../n ...  
Lab1 > backend > test > JS test.js > [e] apiUrl  
1  const axios = require("axios");  
2  
3  const assert = require("assert");  
4  
5  const apiUrl = "http://localhost:3900/api";  
6  
7  describe("POST - check for sign in", () => {  
8    it("/user/login", (done) => {  
9      axios  
10        .post(apiUrl + "/user/login", { email: "snigdha.chaturvedi@sjtu.edu", password: "snigdha" })  
11        .then((response) => {  
12          console.log(response.data);  
13          assert.equal(response.status, 200);  
14          done();  
15        })  
16        .catch((err) => {  
17          done(err);  
18        });  
19    });
20  });
21  
22  describe("GET - get customer details by id", () => {  
23    it("/user/25", (done) => {  
24      axios  
25        .get(apiUrl + "/user/25")  
26        .then((response) => {  
27          console.log(response.data);  
28          assert.equal(response.status, 200);  
29          done();  
30        })  
31        .catch((err) => {  
32          done(err);  
33        });
34    });
35  });
36  
37  describe("GET - product details", () => {  
38    it("/product/14", (done) => {  
39      axios  
40        .get(apiUrl + "/product/14")  
41        .then((response) => {  
42          console.log(response.data);  
43          assert.equal(response.status, 200);  
44          done();  
45        })  
46        .catch((err) => {  
47          done(err);  
48        });
49    });
50  });
51  
52  describe("GET - shop details", () => {  
53    it("/shop/16", (done) => {  
54      axios  
55        .get(apiUrl + "/shop/16")  
56        .then((response) => {  
57          console.log(response.data);  
58          assert.equal(response.status, 200);  
59          done();  
60        })  
61        .catch((err) => {  
62          done(err);  
63        });
64    });
65  });
66  
67  describe("GET - user's shop ", () => {  
68    it("/user/25", (done) => {  
69      axios  
70        .get(apiUrl + "/user/25")  
71        .then((response) => {  
72          console.log(response.data);  
73          assert.equal(response.status, 200);  
74          done();  
75        })  
76        .catch((err) => {  
77          done(err);  
78        });
79    });
80  });
81
```

3. React Testing

Component/Page	React Testing Library
Shop Page	<pre>import React from "react"; import { render, screen } from "@testing-library/react"; import AddShopImage from "../atom/AddShopImage.jsx"; import { BrowserRouter } from "react-router-dom"; import { Provider } from "react-redux"; import store from "../redux/store"; test("renders Shop Page", () => { render(<Provider store={store}> <React.StrictMode> <BrowserRouter> <AddShopImage /> </BrowserRouter> </React.StrictMode> </Provider>); const linkElement = screen.queryByText(/Add image/); expect(linkElement).toBeInTheDocument(); });</pre>
Login Page	<pre>import React from "react"; import { render, screen } from "@testing-library/react"; import LoginConnected from "../molecules/Login.jsx"; import { BrowserRouter } from "react-router-dom"; import { Provider } from "react-redux"; import store from "../redux/store"; test("renders Login Page", () => { render(<Provider store={store}> <React.StrictMode> <BrowserRouter> <LoginConnected /> </BrowserRouter> </React.StrictMode> </Provider>); const linkElement = screen.queryByText(/Login/); expect(linkElement).toBeInTheDocument(); });</pre>
Header	<pre>import React from "react"; import { render, screen } from "@testing-library/react"; import Header from "../molecules/Header.jsx"; import { BrowserRouter } from "react-router-dom"; import { Provider } from "react-redux"; import store from "../redux/store"; test("renders Header Component", () => { render(<Provider store={store}> <React.StrictMode> <BrowserRouter> <Header /> </BrowserRouter> </React.StrictMode> </Provider>); const linkElement = screen.queryByText(/Etsy/); console.log("hdhjbshhj",linkElement) expect(linkElement).toBeInTheDocument(); });</pre>

Product Tile

```
import React from "react";
import { render, screen } from "@testing-library/react";
import { BrowserRouter } from "react-router-dom";
import { Provider } from "react-redux";
import store from "../redux/store";
import ProductItem from "../atom/ProductItem.jsx";

const prodconst = {"category": "https://i5.walmartimages.com/asr/93bb1cce-4af9-48f0-8c0e-b8aabe78b58b.0d1293e7fe185df77d6d4bc3cd193", "id": 15, "idshop": 15, "name": "Mainstays 8-Piece Black Bed", "photo": "https://i5.walmartimages.com/asr/93bb1cce-4af9-48f0-8c0e-b8aabe78b58b.0d1293e7fe185df77d6d4bc3cd193", "productPrice": 160, "purchaseQuantity": 2, };

test("renders Product Tile", () => {
  render(
    <Provider store={store}>
      <React.StrictMode>
        <BrowserRouter>
          <ProductItem productItem={prodconst} favlist={[]} />
        </BrowserRouter>
      </React.StrictMode>
    </Provider>
  );
  const linkElement = screen.queryByText("Mainstays 8-Piece Black Bed");
  expect(linkElement).toBeInTheDocument();
});
```

Cart Item

```
import React from "react";
import { render, screen } from "@testing-library/react";
import { BrowserRouter } from "react-router-dom";
import { Provider } from "react-redux";
import store from "../redux/store";
import CartItem from "../atom/CartItem.jsx";

const prodconst = {"category": "https://i5.walmartimages.com/asr/93bb1cce-4af9-48f0-8c0e-b8aabe78b58b.0d1293e7fe185df77d6d4bc3cd193", "id": 15, "idshop": 15, "name": "Mainstays 8-Piece Black Bed", "photo": "https://i5.walmartimages.com/asr/93bb1cce-4af9-48f0-8c0e-b8aabe78b58b.0d1293e7fe185df77d6d4bc3cd193", "productPrice": 160, "purchaseQuantity": 2, };

test("renders Cart Item", () => {
  render(
    <Provider store={store}>
      <React.StrictMode>
        <BrowserRouter>
          <CartItem product={prodconst} />
        </BrowserRouter>
      </React.StrictMode>
    </Provider>
  );
  const linkElement = screen.queryByText("Quantity:");
  expect(linkElement).toBeInTheDocument();
});
```

Git Commit history

The screenshot shows the GitHub commit history for the 'main' branch of the repository 'Snichat97/EtsyPrivateAWS'. The commits are organized by date:

- Commits on Mar 17, 2022:**
 - http fix. (commit 68ae88) - Snichat97 committed 18 hours ago
 - Fix the backend url. (commit 5cfe57) - Snichat97 committed 19 hours ago
 - Crypto Password addition. (commit 0f94e4) - Snichat97 committed 21 hours ago
 - Minor fixes. (commit 584413) - Snichat97 committed 21 hours ago
 - Filing of the domain. (commit 9c1a2a) - Snichat97 committed 21 hours ago
 - Filing favourites. Encrypting passwords. More shifting to redux. (commit d510ed) - Snichat97 committed 22 hours ago
- Commits on Mar 16, 2022:**
 - Search Page and Landing Page fix. (commit b84406) - Snichat97 committed 2 days ago
 - Update Signup.jsx (commit f1d0c2) - Snichat97 committed 2 days ago
 - fixing the api routes. (commit 0404ff) - Snichat97 committed 2 days ago
 - Shift to redux of account and search and currency. (commit 212006) - Snichat97 committed 2 days ago
 - updating aws vpc. (commit c6516e) - Snichat97 committed 2 days ago
- Commits on Mar 15, 2022:**
 - host setting (commit e830e1) - Snichat97 committed 3 days ago
 - DB stuff. (commit 7cc6e2) - Snichat97 committed 3 days ago
- Commits on Mar 14, 2022:**
 - frontend. (commit d2fcfa) - Snichat97 committed 4 days ago
 - All Push (commit a8513a) - Snichat97 committed 4 days ago

At the bottom, there are 'Never' and 'Older' buttons.

Performance:

Performance testing was done using JMeter. Visible across all loads (100,200,300,400,500 users) the connection pooling of 500 connections outperformed the single connection. It is significantly notable in heavier loads.

With 100 user test case

- Connection Pooling | 0.3 ms average throughput
- Single connection | 3 ms average throughput

With 500 user test case

- Connection Pooling | 1 ms average throughput
- Single connection | 613 ms average throughput

Questions

Compare the results of graphs with and without in-built MySQL connection pooling of the database. Explain the result in detail and describe the connection pooling algorithm if you need to implement connection pooling on your own. (3 pts)

As mentioned in the performance testing segment 2 main features noted were

- Visible across all loads (100,200,300,400,500 users) the connection pooling of 500 connections outperformed the single connection.
- It is significantly notable in heavier loads.

With 100 user test case

- Connection Pooling | 0.3 ms average throughput
- Single connection | 3 ms average throughput

With 500 user test case

- Connection Pooling | 1 ms average throughput
- Single connection | 613 ms average throughput

Connection Pooling Algorithm

- On initialization create 500 connections to the SQL server
- Maintain this in a queue held by the node application.
- Whenever a query is made to the backend one of the connections is picked and the result is fetched from the database, **but the connection is not terminated because connection establishment takes the majority of the time**. After retrieving the data the connection simply joins back into the queue.
- The queries can be distributed to the connections in a round-robin fashion.

What are the ways to improve SQL Performance? List at least 3 strategies. Explain why those strategies improve the performance. (2 pts)

Three strategies to improve SQL Performance:

- Avoid using * in the SELECT query, rather mention only the required columns that need to be fetched this helps in reducing the response time to the queries.
- Simplify the join queries. This can be achieved very easily sometimes just by rearranging the order in which the tables are being joined. Further, filter down the search by using Left inner or Right inner JOIN queries.
- By using clustered SQL Server indexes we can improve the retrieval time of the queries. For example, if I can cluster the products based on the shop in the ETSY application it will significantly reduce the retrieval time for products.