1. Question: Could you explain the reasoning behind importing specific packages in about/About.jsx for the Burger Shop's About Page? How did your choices align with the functionalities or visual elements required for this page?

Solution:

pages are imported to enable the functionality and styling required for different components not only About.jsx component. Let's break down the imports as possible as I can.

1. In the App.js I am importing this

```
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
```

and these packages are important for implementing client-side routing in a React application. They allow navigation between different pages (components) without a full page reload.
It follows node tree components as it will reload the nodes only after the node parent node gets updates
In other words, if the parent node gets updated then the children nodes will get updated as well.

The Router: Provides the context for routing in the application.
The Route: Declares a route that renders a component based on the current URL.
The Routes: A container for multiple `Route` components.

2. Component Imports
in my App.js I have components imported

```
import Home from "./components/home/Home";
import Footer from "./components/layout/Footer";
import Header from "./components/layout/Header";
import Contact from "./components/contact/Contact";
// Add module imports for CART
import Cart from "./components/cart/Cart";
// Add module imports for Shipping
import Shipping from "./components/cart/Shipping";
import Login from "./components/login/Login";
import Profile from "./components/profile/Profile";
// Add module imports for MY ORDERS
import MyOrders from "./components/myOrders/MyOrders";
import OrderDetails from "./components/myOrders/OrderDetails";
import About from "./components/about/About";
```

These imports bring in the specific React components that represent different pages of the application.

3. Styling Imports

```
import "./styles/app.scss";
import "./styles/header.scss";
import "./styles/home.scss";
import "./styles/founder.scss";
import "./styles/menu.scss";
import "./styles/footer.scss";
import "./styles/contact.scss";
import "./styles/cart.scss";
import "./styles/shipping.scss";
import "./styles/login.scss";
import "./styles/profile.scss";
import "./styles/table.scss";
import "./styles/orderDetails.scss";
import "./styles/about.scss";
import "./styles/DropdownMenu.scss";
import "./styles/colors.scss";
//I have created this styling just to make the order details look like an actual  reciept order
import "./styles/receipt.scss";
```

These imports include the stylesheets for different components, ensuring that each section is visually styled according to the design of the component itself.

Each component has its own styling for example the about.scss import is specifically for styling the About page.

4. Routing Configuration

```
<Router>
  <Header isAuthenticated={true} />
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/contact" element={<Contact />} />
    <Route path="/about" element={<About />} />
    <Route path="/cart" element={<Cart />} />
    {<Route path="/shipping" element={<Shipping />} />}
    <Route path="/login" element={<Login />} />
    <Route path="/me" element={<Profile />} />
    <Route path="/myOrders" element={<MyOrders />} />
    <Route path="/order/:id" element={<OrderDetails />} />
  </Routes>
  <Footer />
</Router>
```

This section configures the routes and associates each route with its corresponding React component.
For example, the "/about" route is associated with the `About` component.
And so on with other components accordingly.

2. Question: Explain the coding logic behind creating a Popup triggered by the send button in contact/Contact.jsx. How did you

structure the code to handle the click event and display the Popup?

Solution:

I have used the reactjs-popup library to create a popup that is triggered when the user clicks the "Send" button. Let's break down the coding logic behind what I have tried to code.

1. State Initialization

```
const Contact = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    message: '',
  });

  const [isSubmitted, setIsSubmitted] = useState(false);
```

The useState hook to manage the form data (formData) and to track whether the form has been submitted (isSubmitted).

2. Event Handlers

```
const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.name]: e.target.value,
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();

    // Check if the fields are empty
    if (!formData.name || !formData.email || !formData.message) {
      alert('Please fill in all the fields before submitting.');
      return;
    }

    // Handle form submission logic here
    // in the lecture course the guy said no database required for now
(e.g., send data to server) so I will just log the data to the console
    //the video is here
https://ibm.box.com/s/nlo8r939muitapht5ontxjgdms5bp8yb
    console.log('Form submitted:', formData);

    // Alert user about successful submission
    alert('Form submitted successfully!');
    // Redirect to the home page
    setIsSubmitted(true);
    navigate('/');
  };
```

handleChange: Updates the form data in the state as the user types in the input fields.
handleSubmit: Validates the form fields, logs the form data to the console, display an alert for a successful submission, sets the `isSubmitted` state to (true), and then redirects the user to the home page/home component.

3. Popup Component

```
<Popup trigger={<button type="submit">Send</button>}>
    <div>
      <h2>Your message is important to us!</h2>
      <p>Try to fill out the fields before submission!</p>
    </div>
  </Popup>
```

I have used reactjs-popup library to create a popup. It takes a trigger prop, which specifies the element that, when clicked, will open the popup. In this case, the "Send" button triggers the popup.

3. Question: Illustrate the coding logic behind creating Login and Signup Tabs within login/Login.jsx. What approach did you use to organize and switch between these tabs within the Login and Signup forms?

Solution:

The `Login` component uses MDBTabs and related components from the `mdb-react-ui-kit` library to create tabs for "Sign In" and "Register." Here's an explanation of the coding logic for creating the login and signup tabs and how my code is done;

1. State Initialization

```
const Login = () => {
  const navigate = useNavigate();
  const [justifyActive, setJustifyActive] = useState("tab1");
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });
```

justifyActive: This state variable is used to determine which tab is currently active. It is initially set to "tab1" for the "Sign In" tab.

Source: https://stackoverflow.com/questions/50415189/reactjs-justify-navbar-items

formData: Manages the form data for email and password.

Source: https://stackoverflow.com/questions/43085762/how-to-pass-form-values-as-formdata-in-reactjs-on-submit-function

2. Event Handlers:

```
const handleSignIn = () => {
  // Validate fields
  if (!formData.email || !formData.password) {
    alert("Please fill in all the fields before signing in.");
    return;
  }

  //Assume the user is logged in successfully
  alert("Logged in successfully!");

  // Redirect to the home page after successful login
  navigate("/");
};

const handleRegister = () => {
  // Validate fields
  if (!formData.email || !formData.password) {
    alert("Please fill in all the fields before registering.");
    return;
  }

  // Assume the user is registered successfully
  alert("Registered successfully!");

  // Redirect to the home page after successful registration
  navigate("/");
};

const handleChange = (e) => {
  setFormData({
    ...formData,
    [e.target.name]: e.target.value,
  });
};
```

handleChange: Updates the form data in the state as the user types in the input fields.

handleSignIn and `handleRegister`: These functions contain the logic for handling sign-in and registration, respectively. For now, they display alerts for successful sign-in or registration.

Noting it has been mentioned here https://ibm.box.com/s/nlo8r939muitapht5ontxjgdms5bp8yb

That no database is required. So I just made alert just for informing the user

3. Tab Creation:

```
        <MDBTabsItem>
          <MDBTabsLink show={justifyActive === "tab1"} onClick={() => setJustifyActive("tab1")}>
            Sign In?
          </MDBTabsLink>
        </MDBTabsItem>

        {/*Creating another tab which will be number 2 for registering note delete later, come back here if the registeration does not work*/}
        {/*Creating another tab for registeration is required in the brief*/}
        <MDBTabsItem>
          <MDBTabsLink show={justifyActive === "tab2"} onClick={() => setJustifyActive("tab2")}>
            Register?
          </MDBTabsLink>
        </MDBTabsItem>
```

- Two MDBTabsItem components are created for the "Sign In" and "Register" tabs.
- MDBTabsLink components are used for the tab labels. The show prop is set based on whether the tab is currently active (`justifyActive === "tab1"` or `justifyActive === "tab2"`).
- Clicking on a tab trigger the `setJustifyActive` function, updating the active tab.

4. Tab Content:

```
        <MDBTabsContent>
          <MDBTabsPane show={justifyActive === "tab1"}>
            <div className="text-center mb-3">
              <p>Sign in with:</p>
              <div className="d-flex justify-content-between mx-auto" style={{ width: "40%" }}>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                  <MDBIcon fab icon="facebook-f" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color: "#1266f1" }}>
                  <MDBIcon fab icon="twitter" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="google" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="github" size="sm" />
                </MDBBtn>
              </div>
              <p className="text-center mt-3">or:</p>
            </div>
            <MDBInput wrapperClass="mb-4" label="Email address" id="form1"
type="email" name="email" value={formData.email} onChange={handleChange} />
            <MDBInput wrapperClass="mb-4" label="Password" id="form2"
type="password" name="password" value={formData.password} onChange={handleChange}
/>
            <div className="d-flex justify-content-between mx-4 mb-4">
              <MDBCheckbox name="flexCheck" value="" id="flexCheckDefault"
label="Remember me" />
              <a href="!#">Forgot password?</a>
            </div>
            <MDBBtn className="mb-4 w-100" onClick={handleSignIn}>
              Sign in
            </MDBBtn>
          </MDBTabsPane>

          <MDBTabsPane show={justifyActive === "tab2"}>
            <div className="text-center mb-3">
              <p>Register with:</p>
              <div className="d-flex justify-content-between mx-auto" style={{
width: "40%" }}>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="facebook-f" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="twitter" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="google" size="sm" />
                </MDBBtn>
                <MDBBtn tag="a" color="none" className="m-1" style={{ color:
"#1266f1" }}>
                  <MDBIcon fab icon="github" size="sm" />
                </MDBBtn>
              </div>
              <p className="text-center mt-3">or:</p>
            </div>
            <MDBInput wrapperClass="mb-4" label="Email address" id="form1"
type="email" name="email" value={formData.email} onChange={handleChange} />
            <MDBInput wrapperClass="mb-4" label="Password" id="form2"
type="password" name="password" value={formData.password} onChange={handleChange}
/>
            <div className="d-flex justify-content-between mx-4 mb-4">
              <MDBCheckbox name="flexCheck" value="" id="flexCheckDefault"
label="Remember me" />
            </div>
            <MDBBtn className="mb-4 w-100" onClick={handleRegister}>
              Register
            </MDBBtn>
          </MDBTabsPane>
        </MDBTabsContent>
```

Two `MDBTabsPane` components contain the content for the "Sign In" and "Register" tabs. The `show` prop is set based on the active tab.