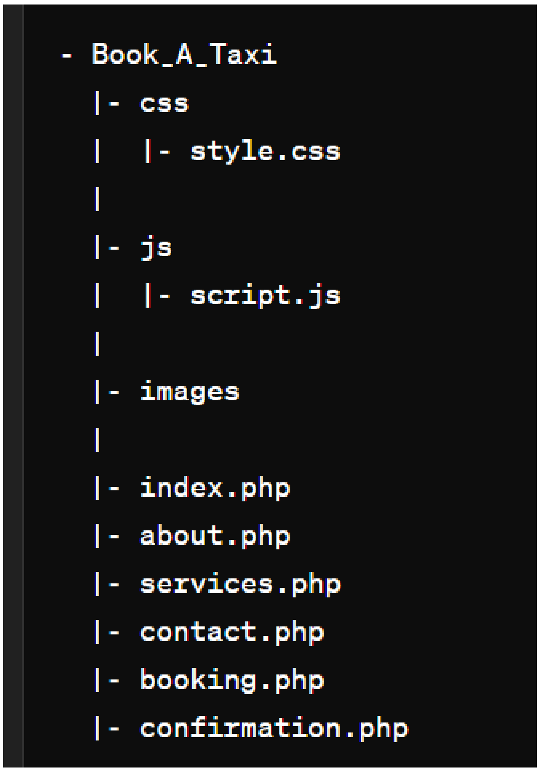
Answer Key

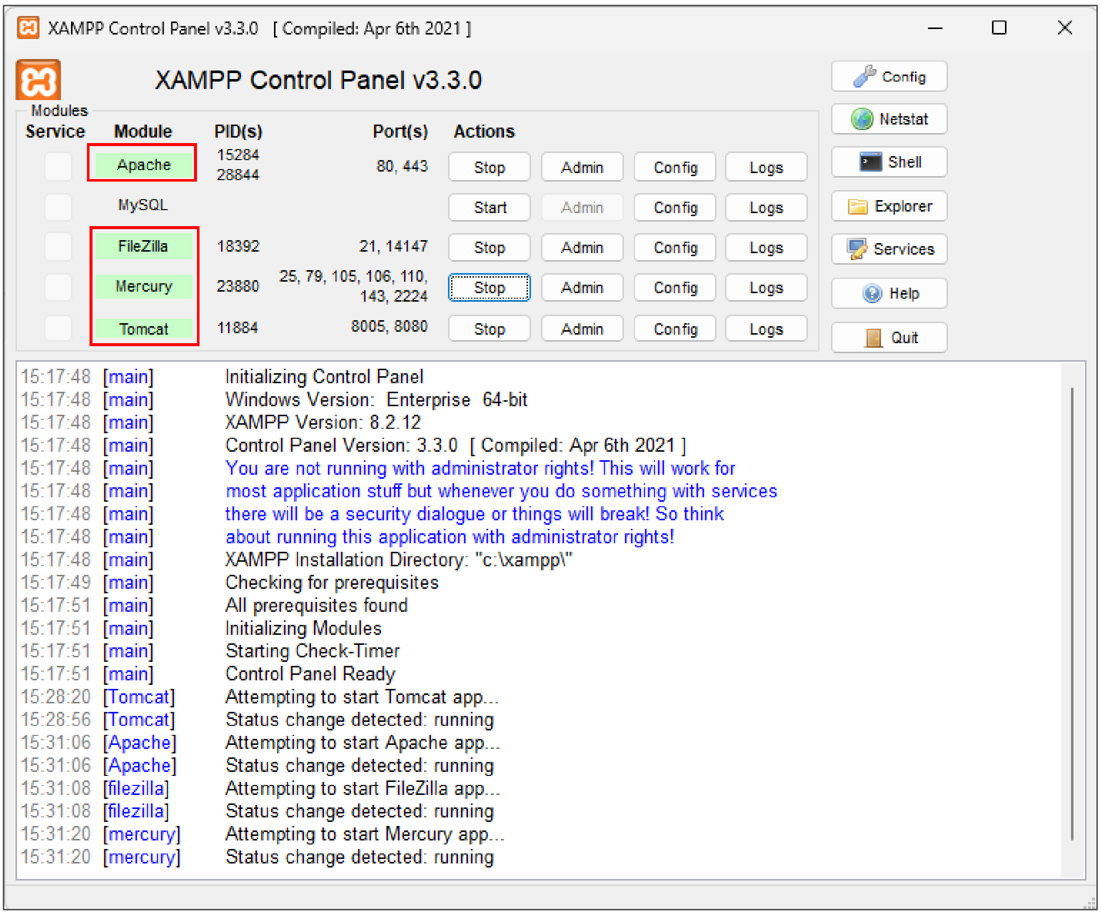
1. Analyze the PHP project structure

1.1. Download, extract, and save the **Book_A_Taxi** project in the **../xampp/htdocs** directory. The directory structure of this project is as follows:



```
- Book_A_Taxi
  |- css
  |   |- style.css
  |
  |- js
  |   |- script.js
  |
  |- images
  |
  |- index.php
  |- about.php
  |- services.php
  |- contact.php
  |- booking.php
  |- confirmation.php
```

1.2. Open **XAMPP control panel** and start the required services, as shown in the screenshot below:



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

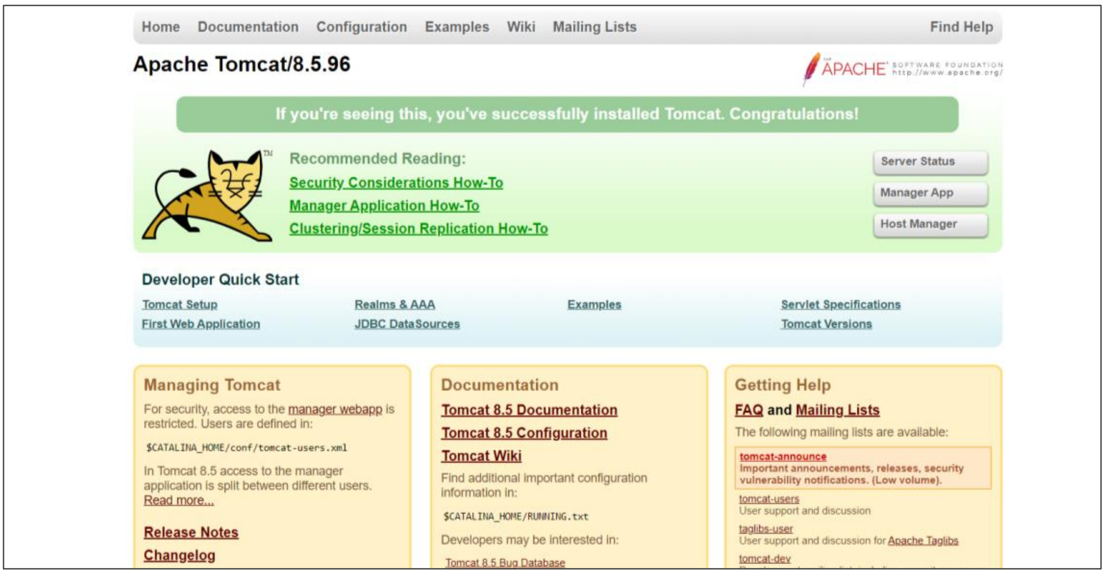
XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	15284 28844	80, 443	Stop	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla	18392	21, 14147	Stop	Admin	Config	Logs
	Mercury	23880	25, 79, 105, 106, 110, 143, 2224	Stop	Admin	Config	Logs
	Tomcat	11884	8005, 8080	Stop	Admin	Config	Logs

```
15:17:48  [main]      Initializing Control Panel
15:17:48  [main]      Windows Version:  Enterprise  64-bit
15:17:48  [main]      XAMPP Version: 8.2.12
15:17:48  [main]      Control Panel Version: 3.3.0  [ Compiled: Apr 6th 2021 ]
15:17:48  [main]      You are not running with administrator rights! This will work for
15:17:48  [main]      most application stuff but whenever you do something with services
15:17:48  [main]      there will be a security dialogue or things will break! So think
15:17:48  [main]      about running this application with administrator rights!
15:17:48  [main]      XAMPP Installation Directory: "c:\xampp\"
15:17:49  [main]      Checking for prerequisites
15:17:51  [main]      All prerequisites found
15:17:51  [main]      Initializing Modules
15:17:51  [main]      Starting Check-Timer
15:17:51  [main]      Control Panel Ready
15:28:20  [Tomcat]    Attempting to start Tomcat app...
15:28:56  [Tomcat]    Status change detected: running
15:31:06  [Apache]    Attempting to start Apache app...
15:31:06  [Apache]    Status change detected: running
15:31:08  [filezilla] Attempting to start FileZilla app...
15:31:08  [filezilla] Status change detected: running
15:31:20  [mercury]   Attempting to start Mercury app...
15:31:20  [mercury]   Status change detected: running
```

Note: Download and install XAMPP from the link below: **https://www.apachefriends.org/download.html**

1.3. Open the browser and navigate to **http://localhost:8080/**



Apache Tomcat/8.5.96

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 8.5 access to the manager application is split between different users. Read more...

Release Notes
Changelog

Documentation

Tomcat 8.5 Documentation
Tomcat 8.5 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 8.5 Bug Database

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

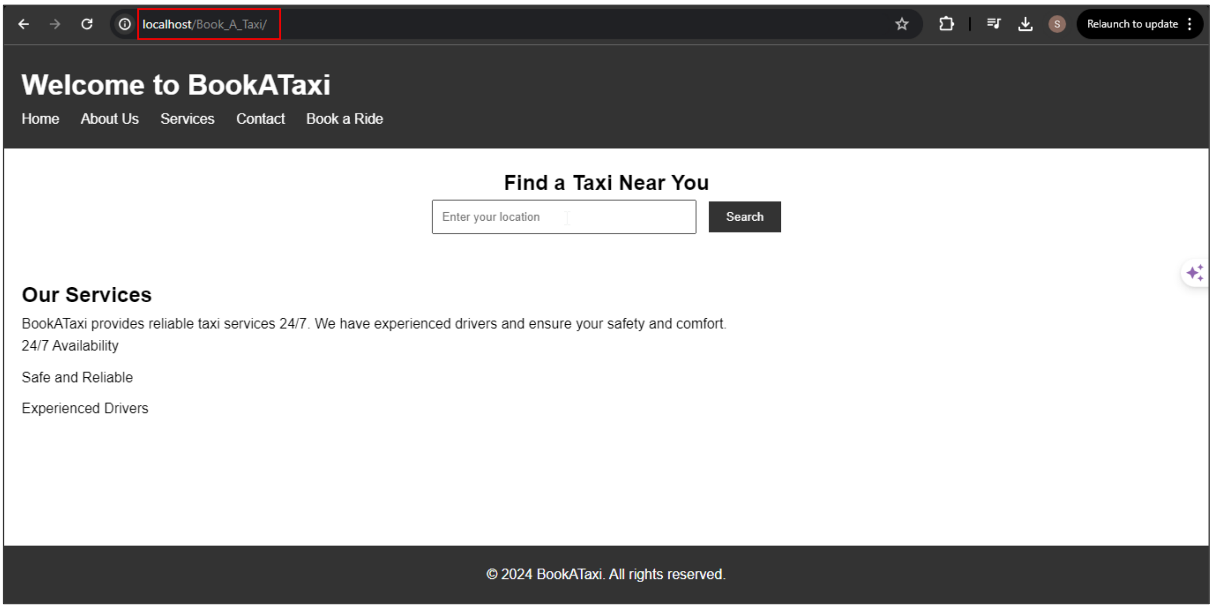
tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev

Note: Apache Tomcat appears if the service is running successfully, as shown in the above screenshot.

1.4. Open the browser and navigate to **http://localhost/Book_A_Taxi/** to verify if the project **Book_A_Taxi** is running successfully, as shown in the screenshot below:



Welcome to BookATaxi

Find a Taxi Near You

Our Services

BookATaxi provides reliable taxi services 24/7. We have experienced drivers and ensure your safety and comfort.

24/7 Availability

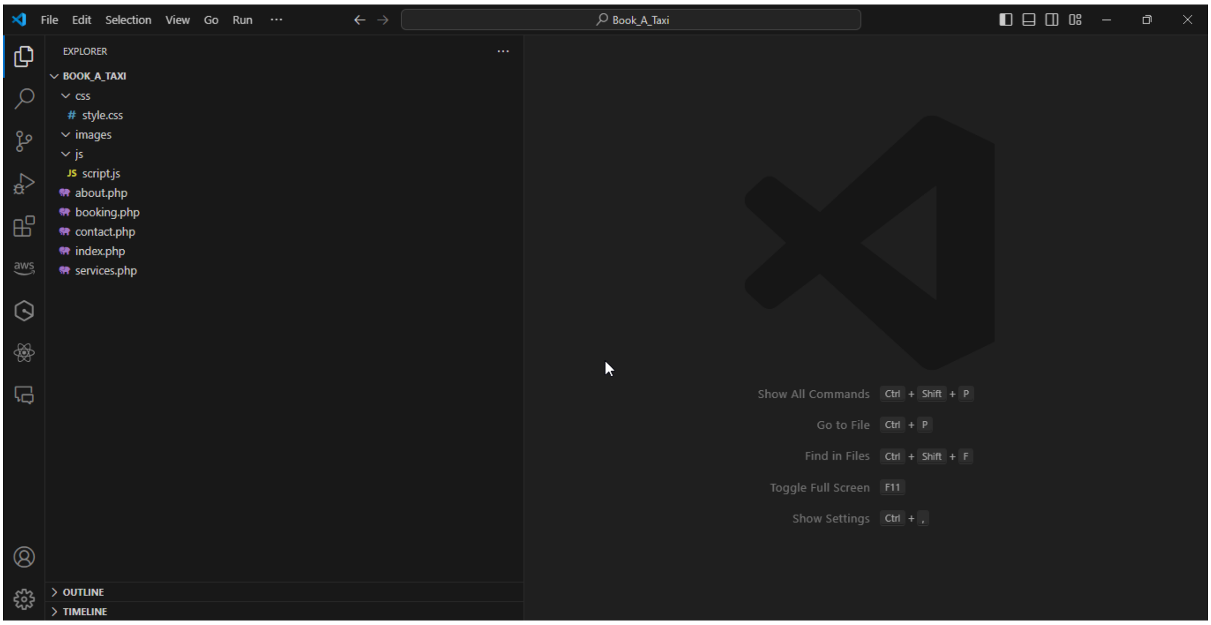
Safe and Reliable

Experienced Drivers

© 2024 BookATaxi. All rights reserved.

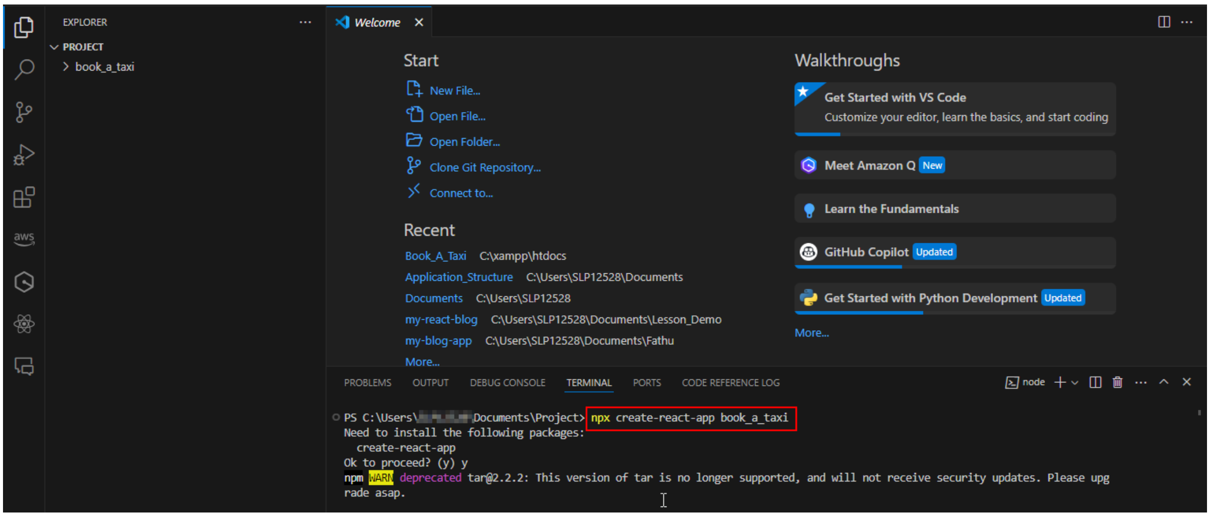
2. Install Node.js and set up a React application

2.1. Open the **Book_A_Taxi** project in Visual Studio Code, as shown in the screenshot below:



2.2. Open a new window in Visual Studio Code and install the React application using the following command:

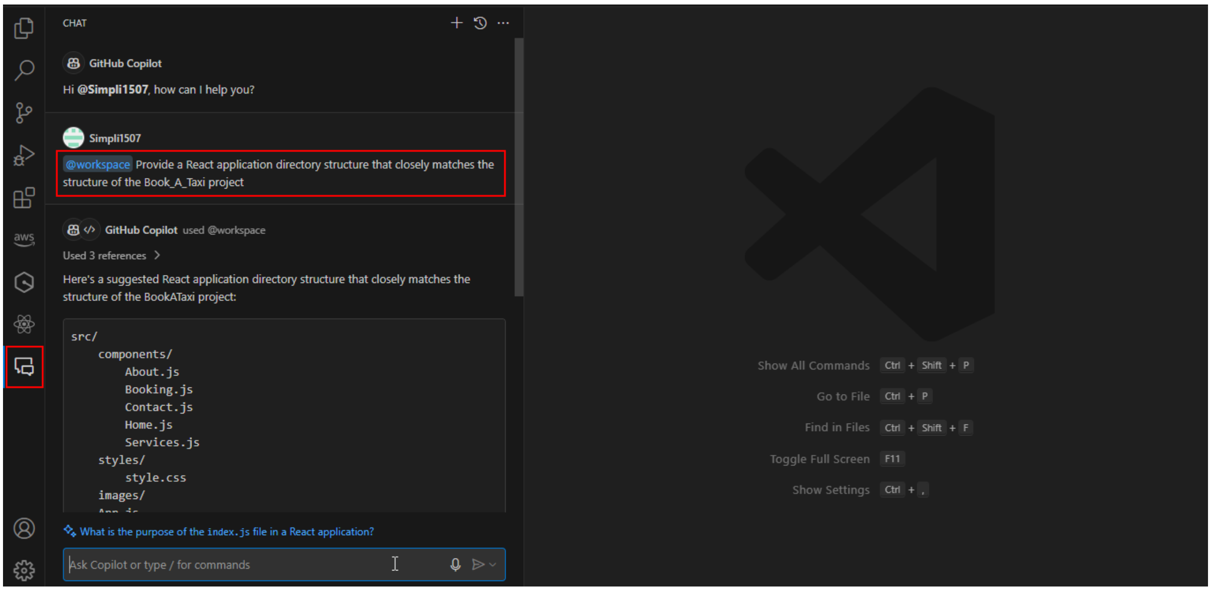
npx create-react-app book_a_taxi


EXPLORER
PROJECT
book_a_taxi
Welcome
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
Book_A_Taxi C:\xampp\htdocs
Application_Structure C:\Users\SLP12528\Documents
Documents C:\Users\SLP12528
my-react-blog C:\Users\SLP12528\Documents\Lesson_Demo
my-blog-app C:\Users\SLP12528\Documents\Fathu
More...
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Meet Amazon Q New
Learn the Fundamentals
GitHub Copilot Updated
Get Started with Python Development Updated
More...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
CODE REFERENCE LOG
node

```
PS C:\Users\[redacted]Documents\Project> npx create-react-app book_a_taxi
Need to install the following packages:
  create-react-app
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upg
rade asap.
```

2.3. Navigate to the PHP project, click on the **Chat** icon, and enter the following prompt to generate an equivalent React directory structure for the **Book_A_Taxi** project:

@workspace Provide a React application directory structure that closely matches the structure of the Book_A_Taxi project


CHAT
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,
Ask Copilot or type / for commands

GitHub Copilot

Hi **@Simpli1507**, how can I help you?

Simpli1507

@workspace Provide a React application directory structure that closely matches the structure of the Book_A_Taxi project

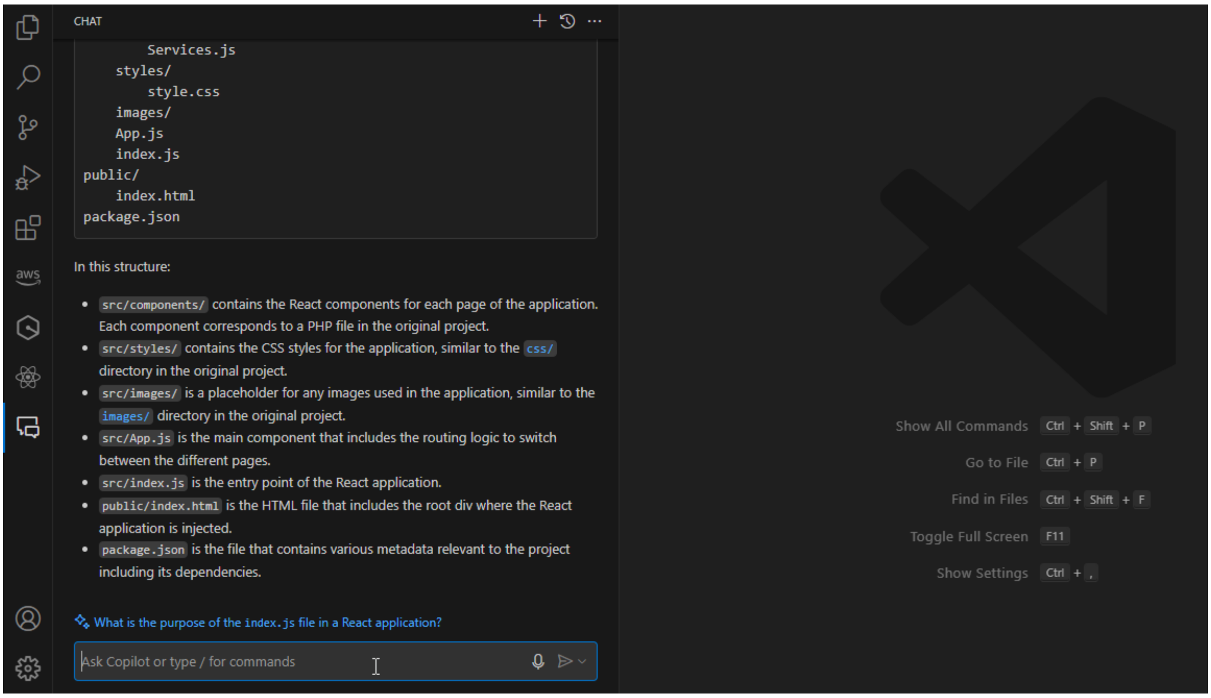
GitHub Copilot used @workspace

Used 3 references

Here's a suggested React application directory structure that closely matches the structure of the BookATaxi project:

```
src/
    components/
        About.js
        Booking.js
        Contact.js
        Home.js
        Services.js
    styles/
        style.css
    images/
```

What is the purpose of the index.js file in a React application?



```
            Services.js
        styles/
            style.css
        images/
        App.js
        index.js
public/
    index.html
package.json
```

In this structure:

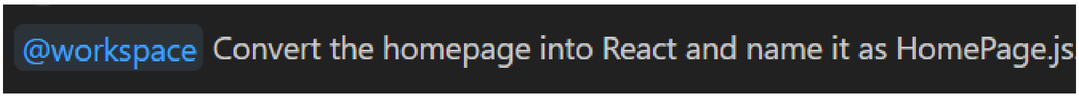
- `src/components/` contains the React components for each page of the application. Each component corresponds to a PHP file in the original project.
- `src/styles/` contains the CSS styles for the application, similar to the `css/` directory in the original project.
- `src/images/` is a placeholder for any images used in the application, similar to the `images/` directory in the original project.
- `src/App.js` is the main component that includes the routing logic to switch between the different pages.
- `src/index.js` is the entry point of the React application.
- `public/index.html` is the HTML file that includes the root div where the React application is injected.
- `package.json` is the file that contains various metadata relevant to the project including its dependencies.

What is the purpose of the `index.js` file in a React application?

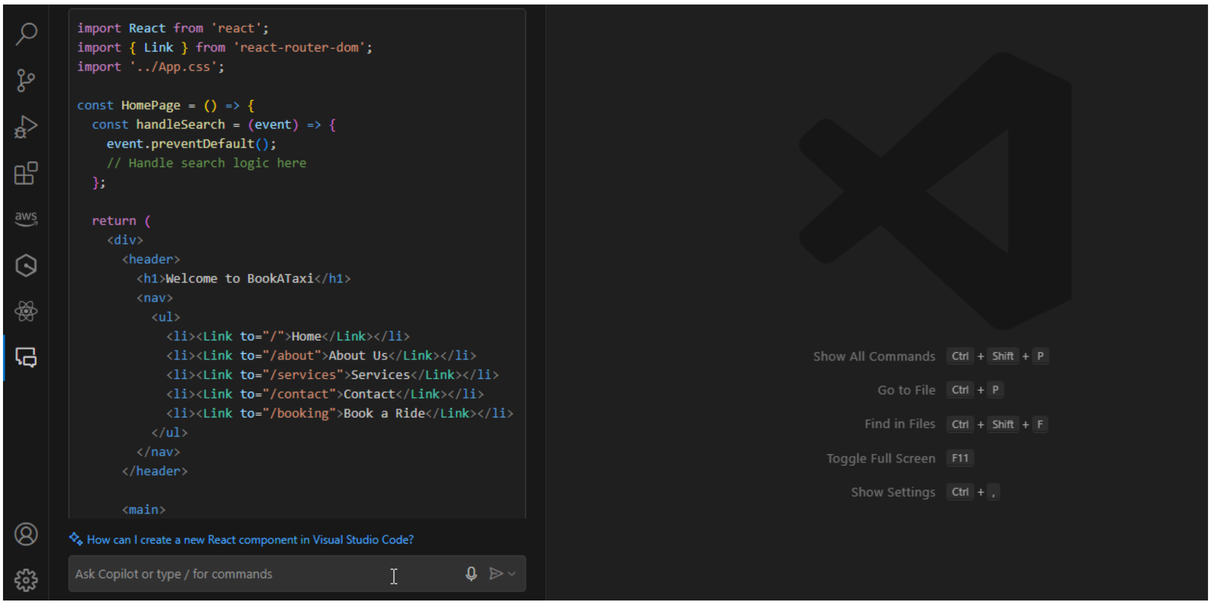
Ask Copilot or type / for commands

3. Convert PHP pages into React components using GitHub Copilot

3.1. Enter the following prompt to convert the homepage, **index.php**, into React code: **@workspace Convert the homepage into React and name it HomePage.js**



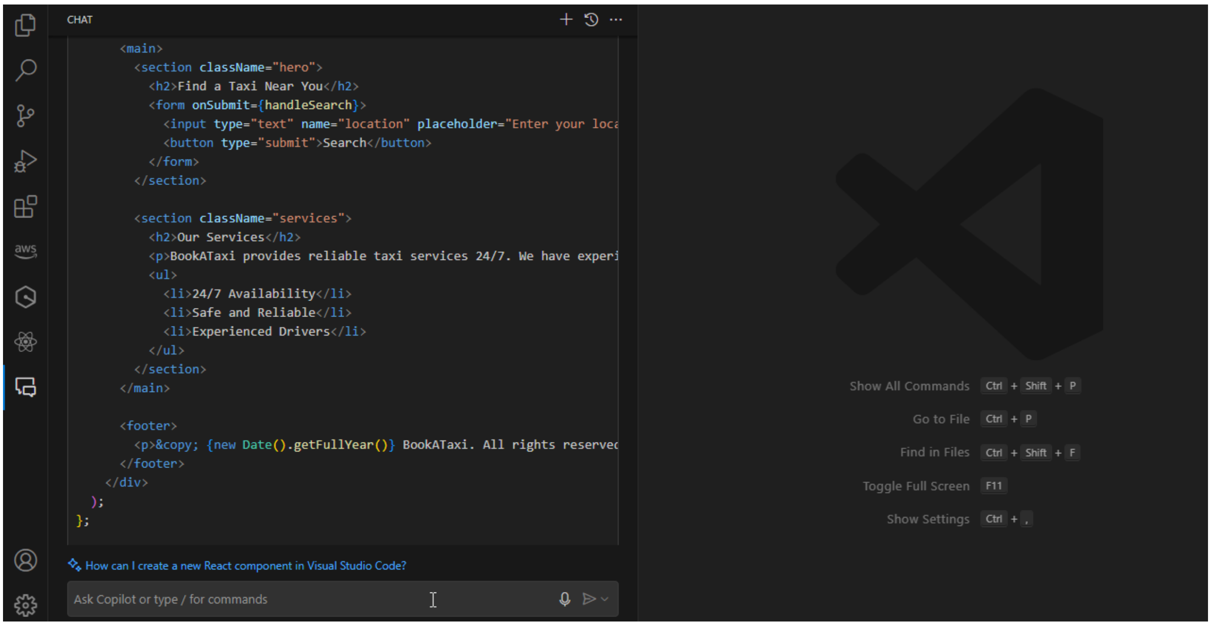
@workspace Convert the homepage into React and name it as HomePage.js


Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,

```
import React from 'react';
import { Link } from 'react-router-dom';
import '../App.css';

const HomePage = () => {
  const handleSearch = (event) => {
    event.preventDefault();
    // Handle search logic here
  };

  return (
    <div>
      <header>
        <h1>Welcome to BookATaxi</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
```

How can I create a new React component in Visual Studio Code?

Ask Copilot or type / for commands



```
      <main>
        <section className="hero">
          <h2>Find a Taxi Near You</h2>
          <form onSubmit={handleSearch}>
            <input type="text" name="location" placeholder="Enter your loca
            <button type="submit">Search</button>
          </form>
        </section>

        <section className="services">
          <h2>Our Services</h2>
          <p>BookATaxi provides reliable taxi services 24/7. We have experi
          <ul>
            <li>24/7 Availability</li>
            <li>Safe and Reliable</li>
            <li>Experienced Drivers</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reservec
      </footer>
    </div>
  );
};
```

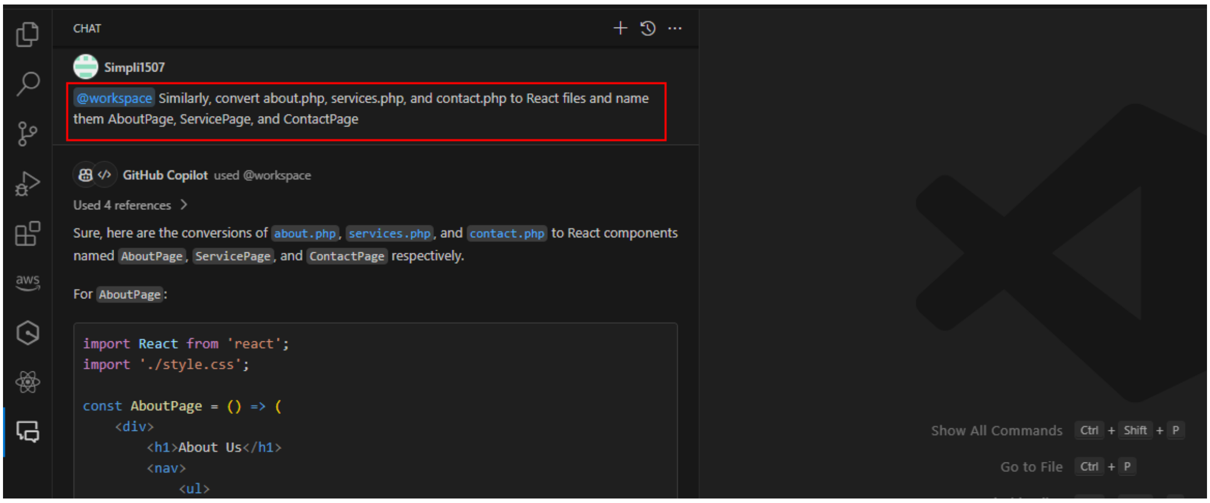
CHAT
How can I create a new React component in Visual Studio Code?
Ask Copilot or type / for commands
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,

The generated code for **HomePage.js** is as follows:

```
import React from 'react';
import { Link } from 'react-router-dom';
import '../App.css';

const HomePage = () => {
  const handleSearch = (event) => {
    event.preventDefault();
    // Handle search logic here
  };

  return (
    <div>
      <header>
        <h1>Welcome to BookATaxi</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="hero">
```

```
          <h2>Find a Taxi Near You</h2>
          <form onSubmit={handleSearch}>
            <input type="text" name="location" placeholder="Enter your location" />
            <button type="submit">Search</button>
          </form>
        </section>

        <section className="services">
          <h2>Our Services</h2>
          <p>BookATaxi provides reliable taxi services 24/7. We have experienced drivers and ensure your safety and comfort.</p>
          <ul>
            <li>24/7 Availability</li>
            <li>Safe and Reliable</li>
            <li>Experienced Drivers</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default HomePage;
```

3.2. Perform the migration of basic pages such as **about.php**, **services.php**, and **contact.php** using the following prompt:
@workspace Similarly, convert about.php, services.php, and contact.php to React files and name them AboutPage, ServicePage, and ContactPage



Simpli1507

@workspace Similarly, convert about.php, services.php, and contact.php to React files and name them AboutPage, ServicePage, and ContactPage

GitHub Copilot used @workspace

Used 4 references

Sure, here are the conversions of `about.php`, `services.php`, and `contact.php` to React components named `AboutPage`, `ServicePage`, and `ContactPage` respectively.

For `AboutPage`:

```
import React from 'react';
import './style.css';

const AboutPage = () => (
    <div>
        <h1>About Us</h1>
        <nav>
            <ul>
```

The generated React scripts are as follows:

AboutPage.js

```
import React from 'react';
import { Link } from 'react-router-dom';
import '../App.css';

const AboutPage = () => {
  return (
    <div>
      <header>
        <h1>About Us</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="about">
          <h2>Welcome to BookATaxi</h2>
          <p>BookATaxi is a leading taxi booking service that aims to provide convenient and reliable transportation solutions to our customers.</p>
          <p>Our team consists of experienced professionals dedicated to ensuring your safety and comfort during your journey.</p>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default AboutPage;
```

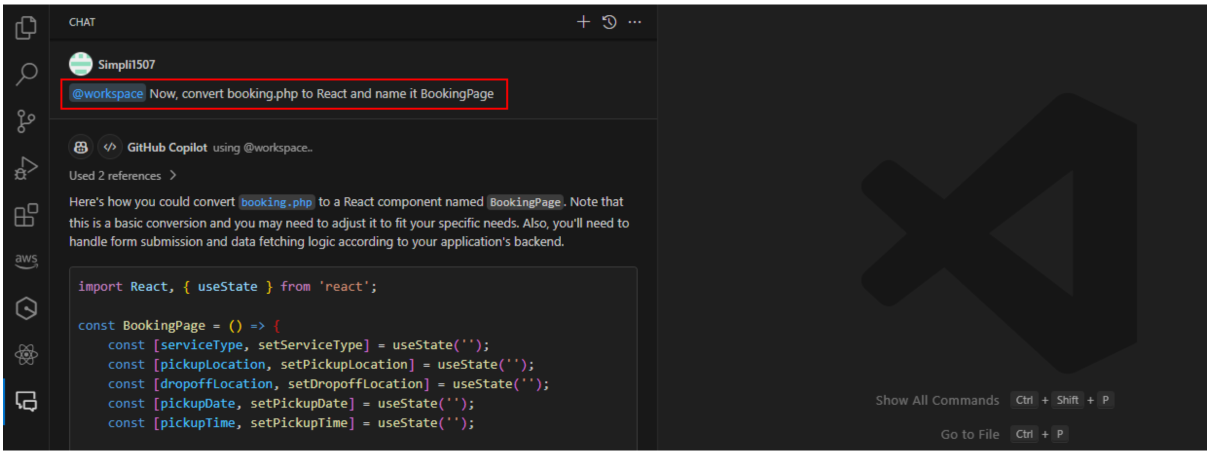
ServicesPage.js

```
import React from 'react';
import { Link } from 'react-router-dom';
import '../App.css';

const ServicesPage = () => {
  return (
    <div>
      <header>
        <h1>Our Services</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="services">
          <h2>Our Services</h2>
          <p>BookATaxi offers a range of services to meet your transportation
needs:</p>
          <ul>
            <li>Point-to-Point Transfer</li>
            <li>Airport Transfers</li>
            <li>Hourly Rental</li>
            <li>City Tours</li>
            <li>And more...</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default ServicesPage;
```

ContactPage.js

```
import React from 'react';
import { Link } from 'react-router-dom';
import '../App.css';

const ContactPage = () => {
  return (
    <div>
      <header>
        <h1>Contact Us</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="contact">
          <h2>Contact Information</h2>
          <p>If you have any inquiries or need assistance, feel free to contact us:</p>
          <ul>
            <li>Email: info@bookataxi.com</li>
            <li>Phone: +1 123-456-7890</li>
            <li>Address: 123 Main Street, City, Country</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default ContactPage;
```

3.3. Convert the booking.php file by providing the following prompt to GitHub Copilot: **@workspace Now, convert booking.php to React and name it BookingPage**



Simpli1507

@workspace Now, convert booking.php to React and name it BookingPage

GitHub Copilot using @workspace..

Used 2 references >

Here's how you could convert `booking.php` to a React component named `BookingPage`. Note that this is a basic conversion and you may need to adjust it to fit your specific needs. Also, you'll need to handle form submission and data fetching logic according to your application's backend.

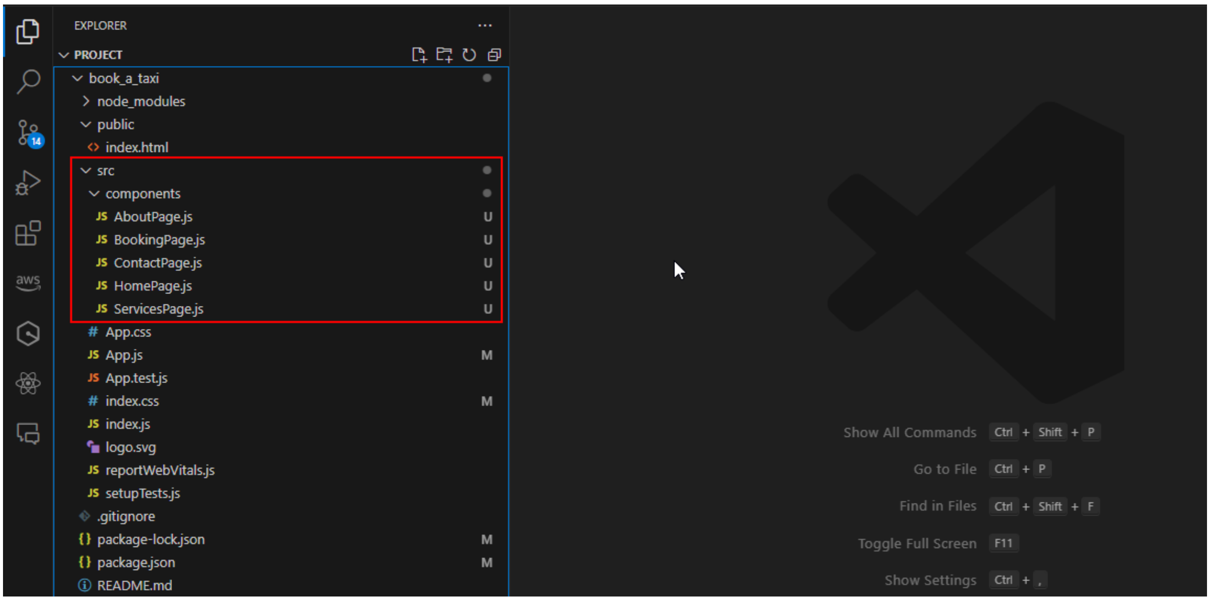
```
import React, { useState } from 'react';

const BookingPage = () => {
    const [serviceType, setServiceType] = useState('');
    const [pickupLocation, setPickupLocation] = useState('');
    const [dropoffLocation, setDropoffLocation] = useState('');
    const [pickupDate, setPickupDate] = useState('');
    const [pickupTime, setPickupTime] = useState('');
```

The generated React script for the **BookingPage.js** is as follows:

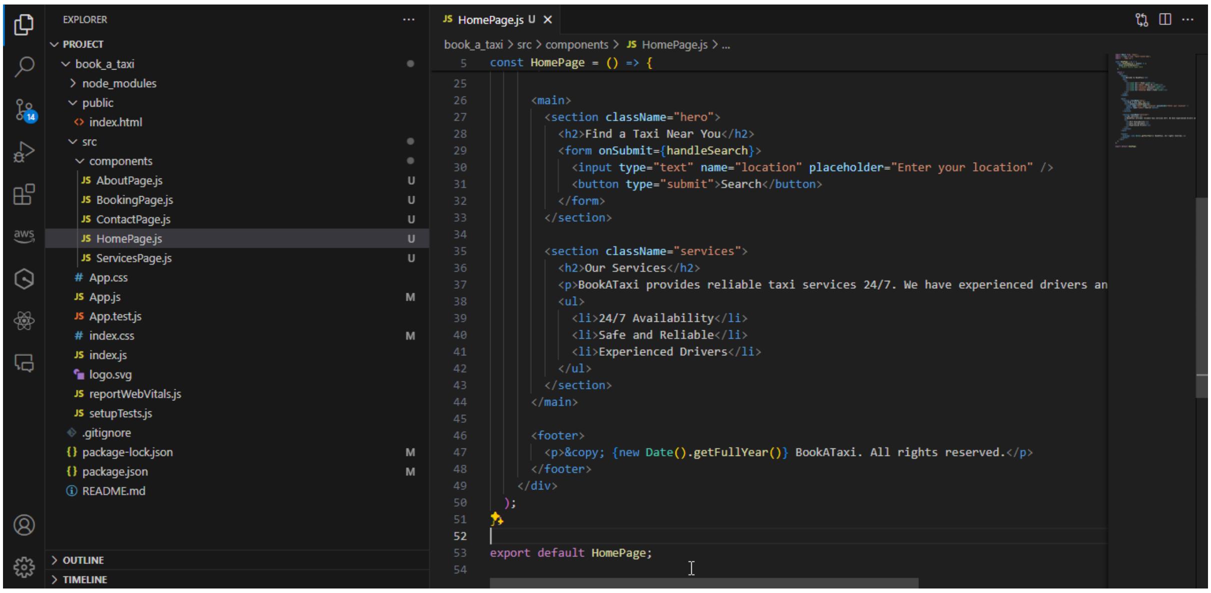
```
import React, { useState } from 'react';

const BookingPage = () => {
  const [serviceType, setServiceType] = useState('');
  const [pickupLocation, setPickupLocation] = useState('');
  const [dropoffLocation, setDropoffLocation] = useState('');
  const [pickupDate, setPickupDate] = useState('');
  const [pickupTime, setPickupTime] = useState('');
  const [confirmation, setConfirmation] = useState('');

  // Simulated service types from database
  const serviceTypes = ['Type 0', 'Type 1', 'Type 2', 'Type 3'];

  // Handle form submission
  const handleSubmit = (event) => {
    event.preventDefault();
    // Process form data here
    // For example, send a POST request to your server
    // ...

    // Simulate complex data processing
    const processedServiceType = serviceType.toUpperCase();
    const processedPickupLocation = pickupLocation.replace(/ /g, '-');
    const processedDropoffLocation = dropoffLocation.replace(/ /g, '-');

    // Set confirmation message
    const confirmationMessage = (
      <>
        <h2>Your booking has been confirmed!</h2>
        <p>Thank you for choosing our service. Your ride details are as follows:</p>
        <ul>
          <li><strong>Service Type:</strong> {processedServiceType}</li>
          <li><strong>Pickup Location:</strong> {processedPickupLocation}</li>
          <li><strong>Dropoff Location:</strong> {processedDropoffLocation}</li>
          <li><strong>Pickup Date:</strong> {pickupDate}</li>
          <li><strong>Pickup Time:</strong> {pickupTime}</li>
        </ul>
      </>
    );

    setConfirmation(confirmationMessage);
  };

  return (
    <div className="BookingPage">
      <header>
        <h1>Book a Ride</h1>
        <nav>
          <ul>
            <li><a href="/">Home</a></li>
            <li><a href="/about">About Us</a></li>
            <li><a href="/services">Services</a></li>
            <li><a href="/contact">Contact</a></li>
            <li><a href="/booking">Book a Ride</a></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="booking-form">
          <h2>Book a Ride with BookATaxi</h2>
          <form onSubmit={handleSubmit}>
            <label htmlFor="service-type">Select Service Type:</label>
            <select name="service-type" id="service-type" required onChange={(e) =>
setServiceType(e.target.value)}>
              {serviceTypes.map((type, index) => (
                <option key={index} value={type}>{type}</option>
              ))}
            </select>
            <label htmlFor="pickup-location">Pickup Location:</label>
            <input type="text" name="pickup-location" id="pickup-location" required
onChange={(e) => setPickupLocation(e.target.value)} />
            <label htmlFor="dropoff-location">Dropoff Location:</label>
            <input type="text" name="dropoff-location" id="dropoff-location" required
onChange={(e) => setDropoffLocation(e.target.value)} />
```

```
            <label htmlFor="pickup-date">Pickup Date:</label>
            <input type="date" name="pickup-date" id="pickup-date" required onChange={(e) => setPickupDate(e.target.value)} />
            <label htmlFor="pickup-time">Pickup Time:</label>
            <input type="time" name="pickup-time" id="pickup-time" required onChange={(e) => setPickupTime(e.target.value)} />
            <button type="submit">Book Now</button>
          </form>
        </section>
      </main>

      {confirmation && (
        <div className="confirmation">
          {confirmation}
        </div>
      )}

      <footer>
        <p>© {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default BookingPage;
```

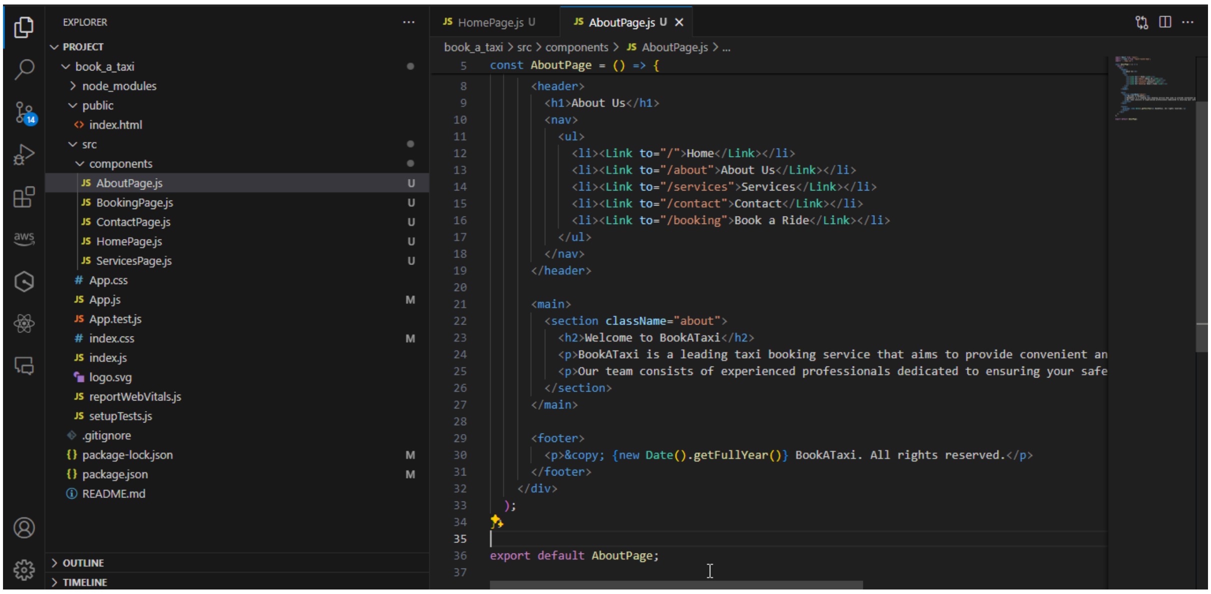
3.4. Navigate to the React project, create a folder named components, and inside the components folder, create **AboutPage.js**, **BookingPage.js**, **ContactPage.js**, **HomePage.js**, **ServicesPage.js** files, as shown in the screenshot below:



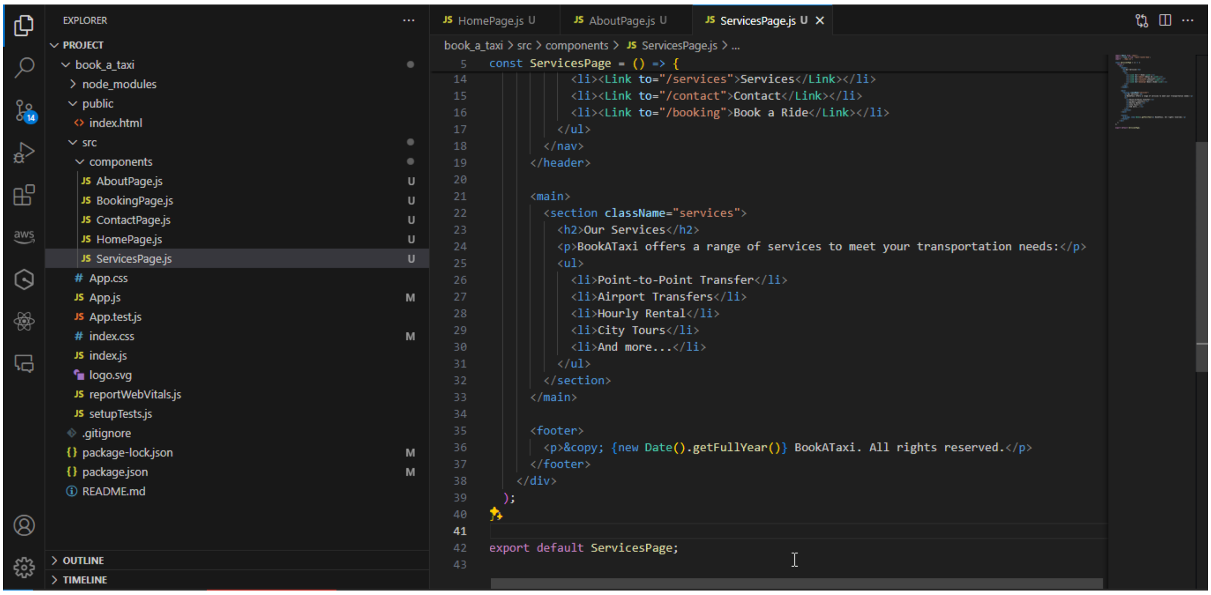
3.5. Copy all the generated scripts and paste them into the **components** folder, as shown in the screenshots below:



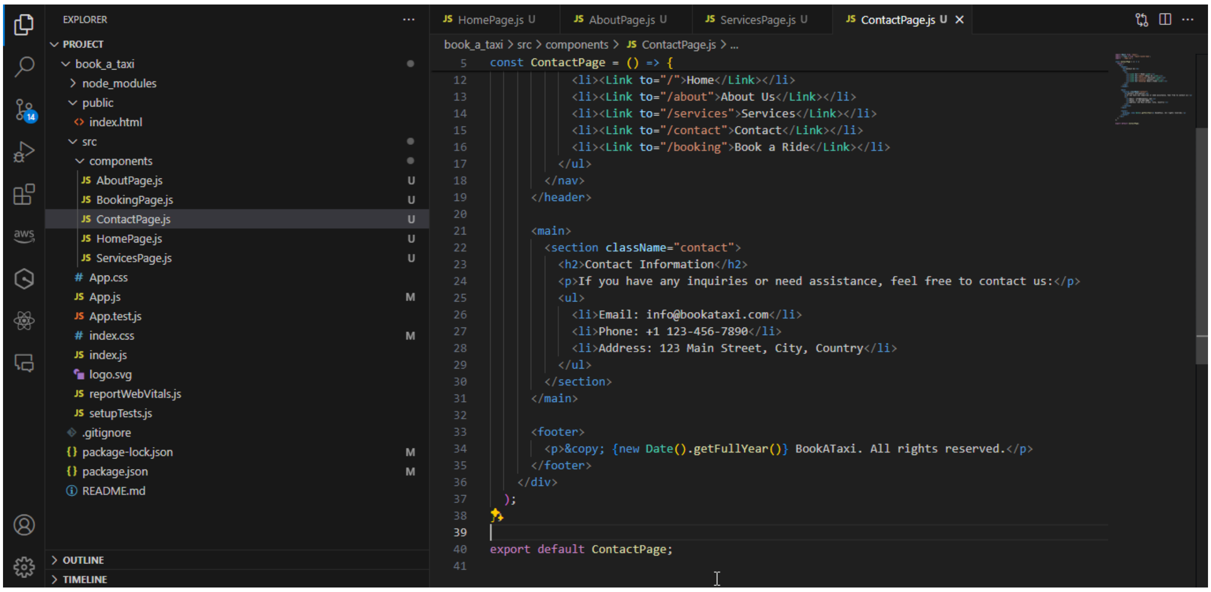
```
const HomePage = () => {

      <main>
        <section className="hero">
          <h2>Find a Taxi Near You</h2>
          <form onSubmit={handleSearch}>
            <input type="text" name="location" placeholder="Enter your location" />
            <button type="submit">Search</button>
          </form>
        </section>

        <section className="services">
          <h2>Our Services</h2>
          <p>BookATaxi provides reliable taxi services 24/7. We have experienced drivers an
          <ul>
            <li>24/7 Availability</li>
            <li>Safe and Reliable</li>
            <li>Experienced Drivers</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );


export default HomePage;
```



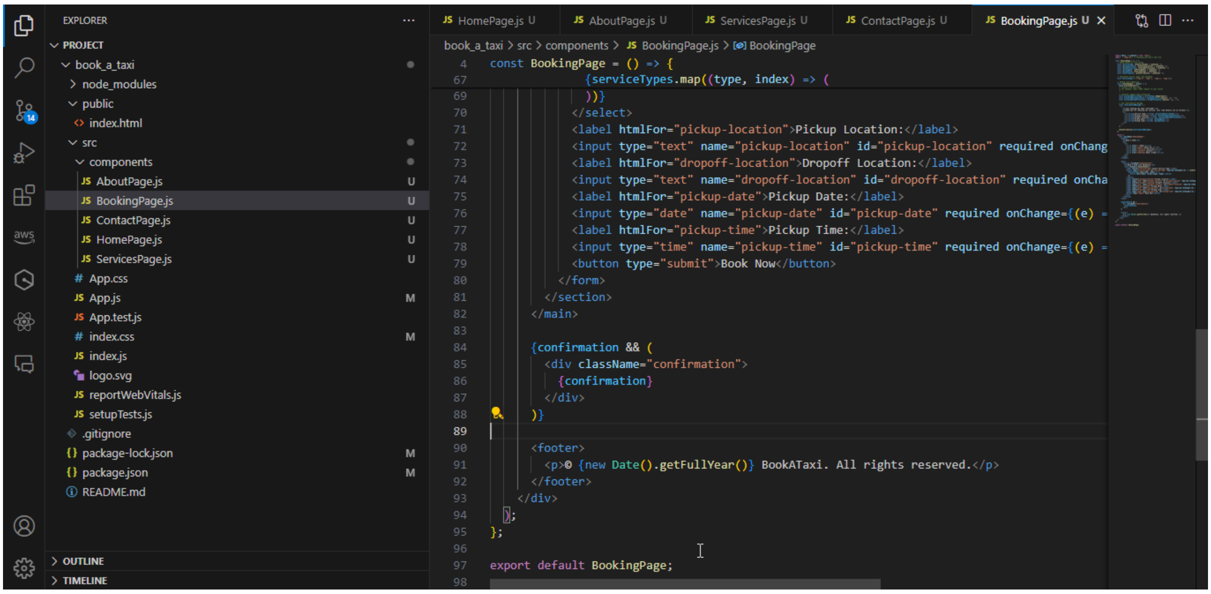
```
const AboutPage = () => {
      <header>
        <h1>About Us</h1>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="about">
          <h2>Welcome to BookATaxi</h2>
          <p>BookATaxi is a leading taxi booking service that aims to provide convenient an
          <p>Our team consists of experienced professionals dedicated to ensuring your safe
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );


export default AboutPage;
```



```
const ServicesPage = () => {
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="services">
          <h2>Our Services</h2>
          <p>BookATaxi offers a range of services to meet your transportation needs:</p>
          <ul>
            <li>Point-to-Point Transfer</li>
            <li>Airport Transfers</li>
            <li>Hourly Rental</li>
            <li>City Tours</li>
            <li>And more...</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );


export default ServicesPage;
```



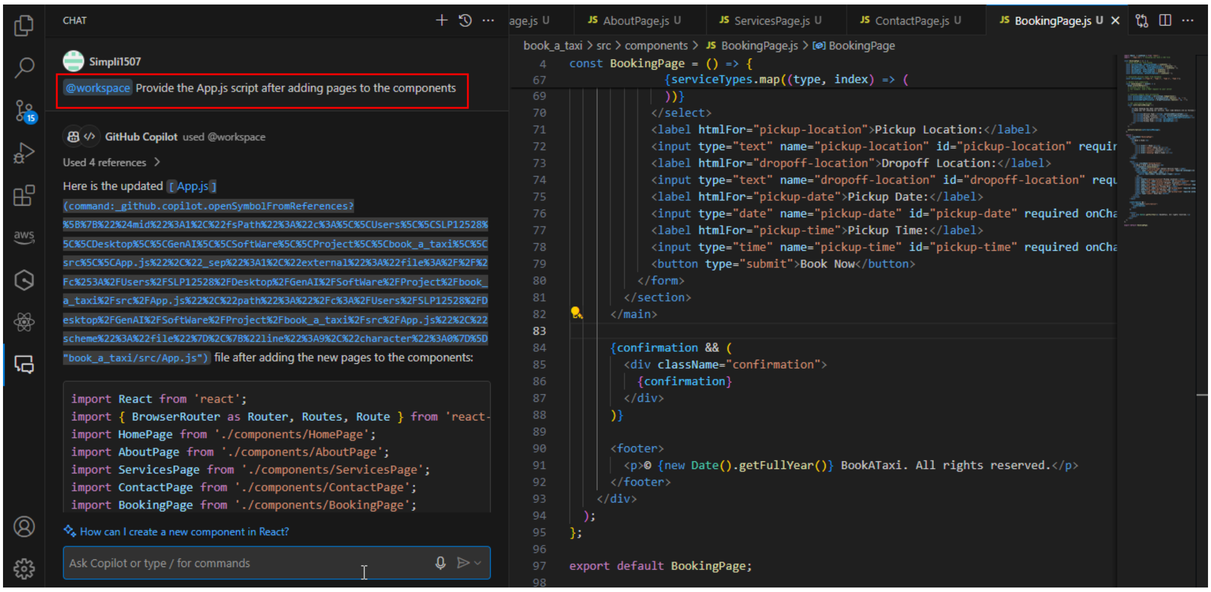
```
const ContactPage = () => {
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About Us</Link></li>
            <li><Link to="/services">Services</Link></li>
            <li><Link to="/contact">Contact</Link></li>
            <li><Link to="/booking">Book a Ride</Link></li>
          </ul>
        </nav>
      </header>

      <main>
        <section className="contact">
          <h2>Contact Information</h2>
          <p>If you have any inquiries or need assistance, feel free to contact us:</p>
          <ul>
            <li>Email: info@bookataxi.com</li>
            <li>Phone: +1 123-456-7890</li>
            <li>Address: 123 Main Street, City, Country</li>
          </ul>
        </section>
      </main>

      <footer>
        <p>&copy; {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );


export default ContactPage;
```



```
const BookingPage = () => {
              {serviceTypes.map((type, index) => (
              ))}
            </select>
            <label htmlFor="pickup-location">Pickup Location:</label>
            <input type="text" name="pickup-location" id="pickup-location" required onChang
            <label htmlFor="dropoff-location">Dropoff Location:</label>
            <input type="text" name="dropoff-location" id="dropoff-location" required onCha
            <label htmlFor="pickup-date">Pickup Date:</label>
            <input type="date" name="pickup-date" id="pickup-date" required onChange={(e) =
            <label htmlFor="pickup-time">Pickup Time:</label>
            <input type="time" name="pickup-time" id="pickup-time" required onChange={(e) =
            <button type="submit">Book Now</button>
          </form>
        </section>
      </main>

      {confirmation && (
        <div className="confirmation">
          {confirmation}
        </div>
      )}

      <footer>
        <p>© {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
      </footer>
    </div>
  );
};

export default BookingPage;
```

4. Implement routing in the React project

4.1. Open the created **book_a_taxi** React project in Visual Studio Code and enter the following prompt to update the **App.js** file:
@workspace Provide the App.js script after adding pages to the components



Simpli1507

@workspace Provide the App.js script after adding pages to the components

GitHub Copilot used @workspace

Used 4 references

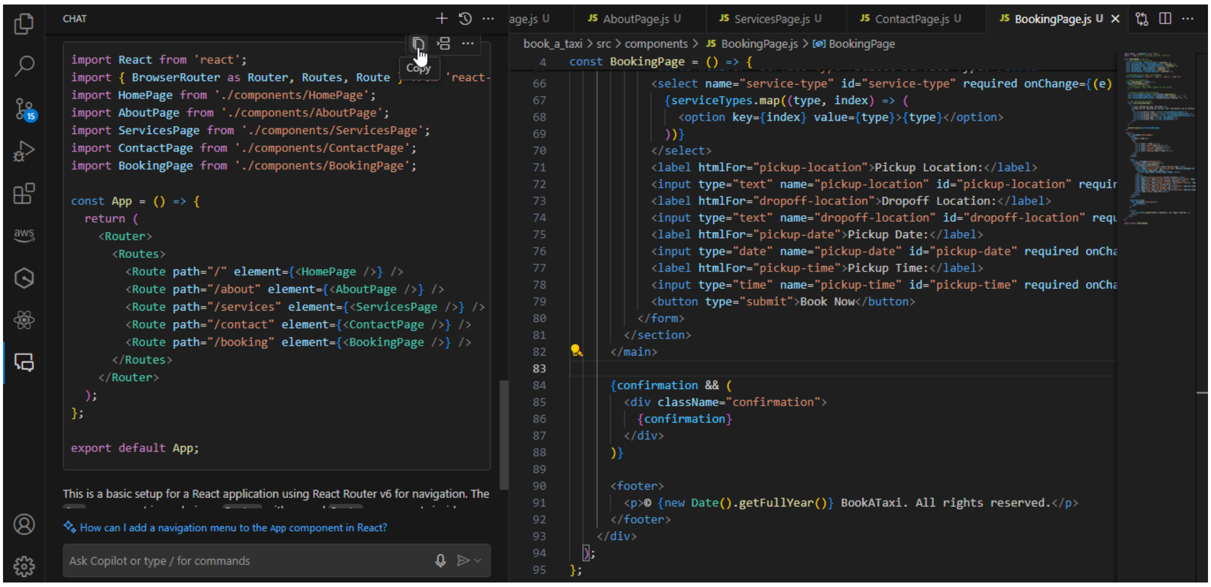
Here is the updated [App.js](command:_github.copilot.openSymbolFromReferences?%5B%7B%22%24mid%22%3A1%2C%22fsPath%22%3A%22c%3A%5C%5CUsers%5C%5CSLP12528%5C%5CDesktop%5C%5CGenAI%5C%5CSoftWare%5C%5CProject%5C%5Cbook_a_taxi%5C%5Csrc%5C%5CApp.js%22%2C%22_sep%22%3A1%2C%22external%22%3A%22file%3A%2F%2F%2Fc%253A%2FUsers%2FSLP12528%2FDesktop%2FGenAI%2FSoftWare%2FProject%2Fbook_a_taxi%2Fsrc%2FApp.js%22%2C%22path%22%3A%22%2Fc%3A%2FUsers%2FSLP12528%2FDesktop%2FGenAI%2FSoftWare%2FProject%2Fbook_a_taxi%2Fsrc%2FApp.js%22%2C%22scheme%22%3A%22file%22%7D%2C%7B%22line%22%3A9%2C%22character%22%3A0%7D%5D "book_a_taxi/src/App.js") file after adding the new pages to the components:

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';
```

How can I create a new component in React?

Ask Copilot or type / for commands

```
const BookingPage = () => {
            {serviceTypes.map((type, index) => (
            ))}
          </select>
          <label htmlFor="pickup-location">Pickup Location:</label>
          <input type="text" name="pickup-location" id="pickup-location" requir
          <label htmlFor="dropoff-location">Dropoff Location:</label>
          <input type="text" name="dropoff-location" id="dropoff-location" requ
          <label htmlFor="pickup-date">Pickup Date:</label>
          <input type="date" name="pickup-date" id="pickup-date" required onCha
          <label htmlFor="pickup-time">Pickup Time:</label>
          <input type="time" name="pickup-time" id="pickup-time" required onCha
          <button type="submit">Book Now</button>
        </form>
      </section>
    </main>

    {confirmation && (
      <div className="confirmation">
        {confirmation}
      </div>
    )}

    <footer>
      <p>© {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
    </footer>
  </div>
 );
};

export default BookingPage;
```



```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';

const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
        <Route path="/services" element={<ServicesPage />} />
        <Route path="/contact" element={<ContactPage />} />
        <Route path="/booking" element={<BookingPage />} />
      </Routes>
    </Router>
  );
};

export default App;
```

This is a basic setup for a React application using React Router v6 for navigation. The

How can I add a navigation menu to the App component in React?

Ask Copilot or type / for commands

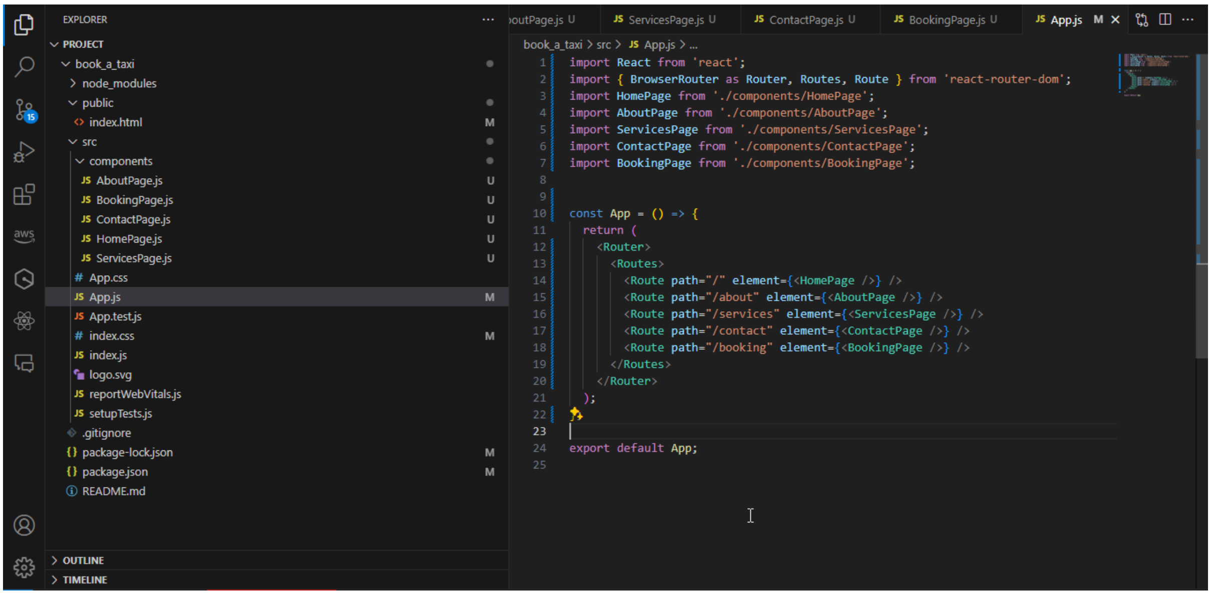
```
const BookingPage = () => {
          <select name="service-type" id="service-type" required onChange={(e)
            {serviceTypes.map((type, index) => (
              <option key={index} value={type}>{type}</option>
            ))}
          </select>
          <label htmlFor="pickup-location">Pickup Location:</label>
          <input type="text" name="pickup-location" id="pickup-location" requir
          <label htmlFor="dropoff-location">Dropoff Location:</label>
          <input type="text" name="dropoff-location" id="dropoff-location" requ
          <label htmlFor="pickup-date">Pickup Date:</label>
          <input type="date" name="pickup-date" id="pickup-date" required onCha
          <label htmlFor="pickup-time">Pickup Time:</label>
          <input type="time" name="pickup-time" id="pickup-time" required onCha
          <button type="submit">Book Now</button>
        </form>
      </section>
    </main>

    {confirmation && (
      <div className="confirmation">
        {confirmation}
      </div>
    )}

    <footer>
      <p>© {new Date().getFullYear()} BookATaxi. All rights reserved.</p>
    </footer>
  </div>
 );
};
```

The generated React script for the **App.js** is as follows:

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';
```

```
const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
        <Route path="/services" element={<ServicesPage />} />
        <Route path="/contact" element={<ContactPage />} />
        <Route path="/booking" element={<BookingPage />} />
      </Routes>
    </Router>
  );
};

export default App;
```

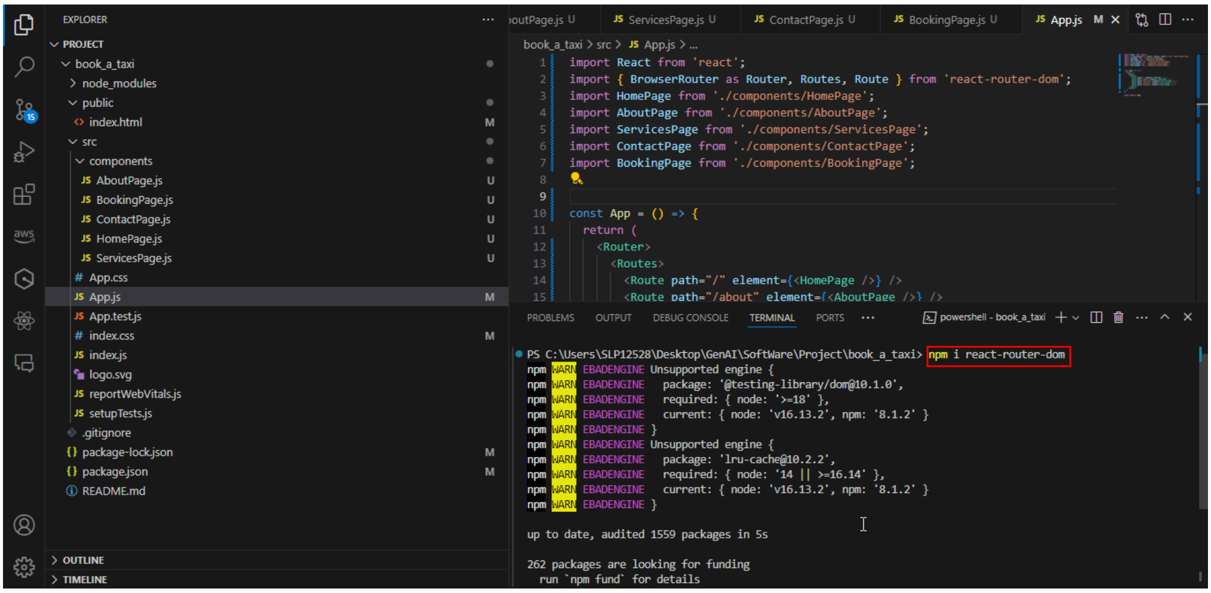
4.2. Copy and paste the generated script into the **App.js** file, as shown in the screenshot below:



```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
        <Route path="/services" element={<ServicesPage />} />
        <Route path="/contact" element={<ContactPage />} />
        <Route path="/booking" element={<BookingPage />} />
      </Routes>
    </Router>
  );
}

export default App;
```

5. Test React components and functionalities for optimal performance

5.1. Install React Router DOM using the following command:

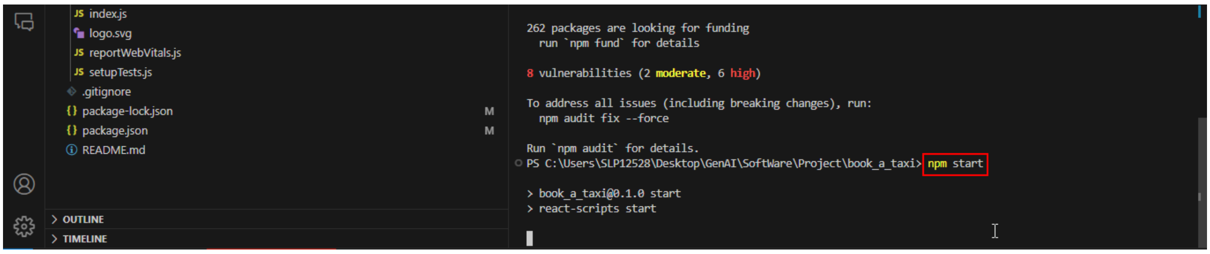


```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
```

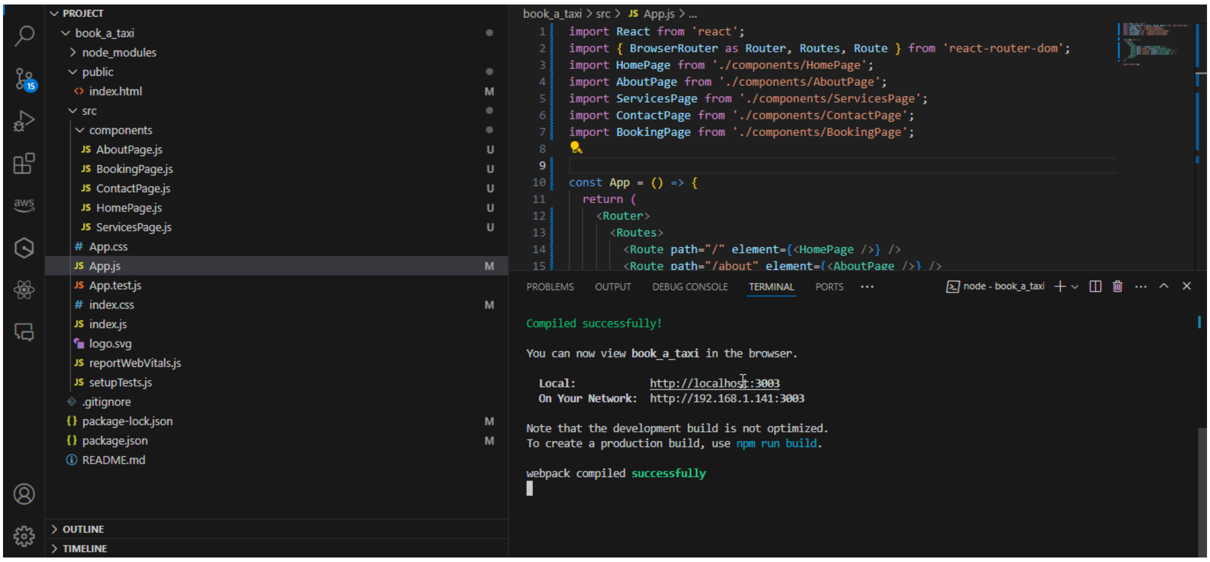
```
PS C:\Users\SLP12528\Desktop\GenAI\SoftWare\Project\book_a_taxi> npm i react-router-dom
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@testing-library/dom@10.1.0',
npm WARN EBADENGINE   required: { node: '>=18' },
npm WARN EBADENGINE   current: { node: 'v16.13.2', npm: '8.1.2' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'lru-cache@10.2.2',
npm WARN EBADENGINE   required: { node: '14 || >=16.14' },
npm WARN EBADENGINE   current: { node: 'v16.13.2', npm: '8.1.2' }
npm WARN EBADENGINE }

up to date, audited 1559 packages in 5s

262 packages are looking for funding
  run `npm fund` for details
```

5.2. Run the React application using the following command:

npm start



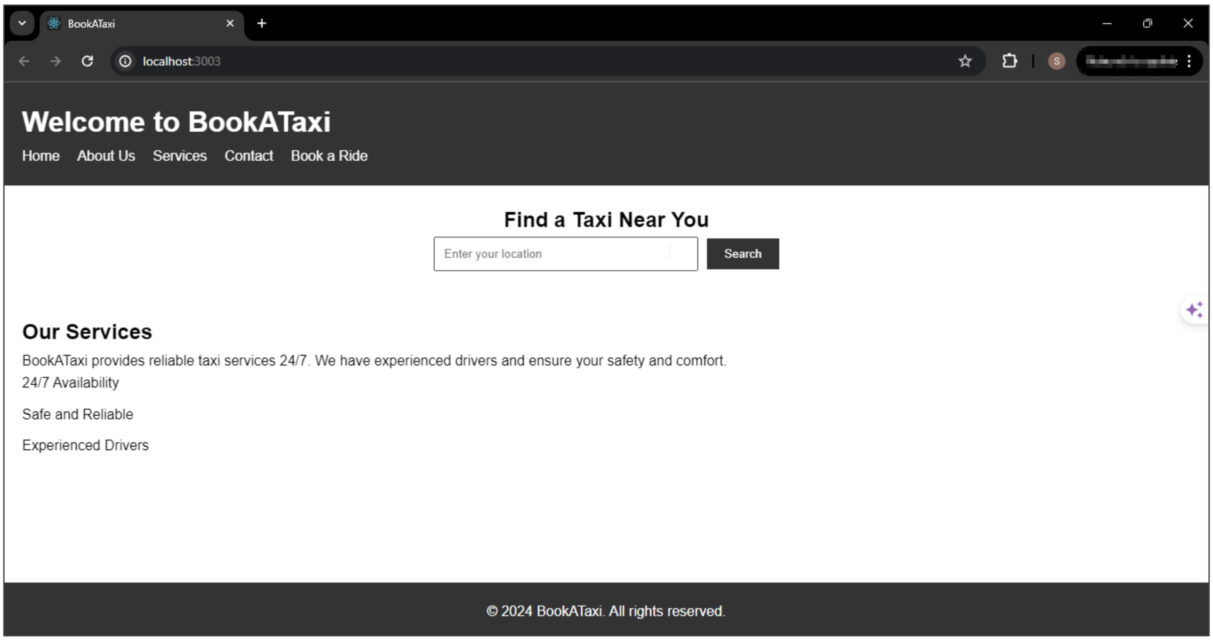
```
262 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\SLP12528\Desktop\GenAI\SoftWare\Project\book_a_taxi> npm start

> book_a_taxi@0.1.0 start
> react-scripts start
```



```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
```

```
Compiled successfully!

You can now view book_a_taxi in the browser.

  Local:            http://localhost:3003
  On Your Network:  http://192.168.1.141:3003

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

You will see the output as follows:



Welcome to BookATaxi

Home About Us Services Contact Book a Ride

Find a Taxi Near You

Enter your location

Search

Our Services

BookATaxi provides reliable taxi services 24/7. We have experienced drivers and ensure your safety and comfort.

24/7 Availability

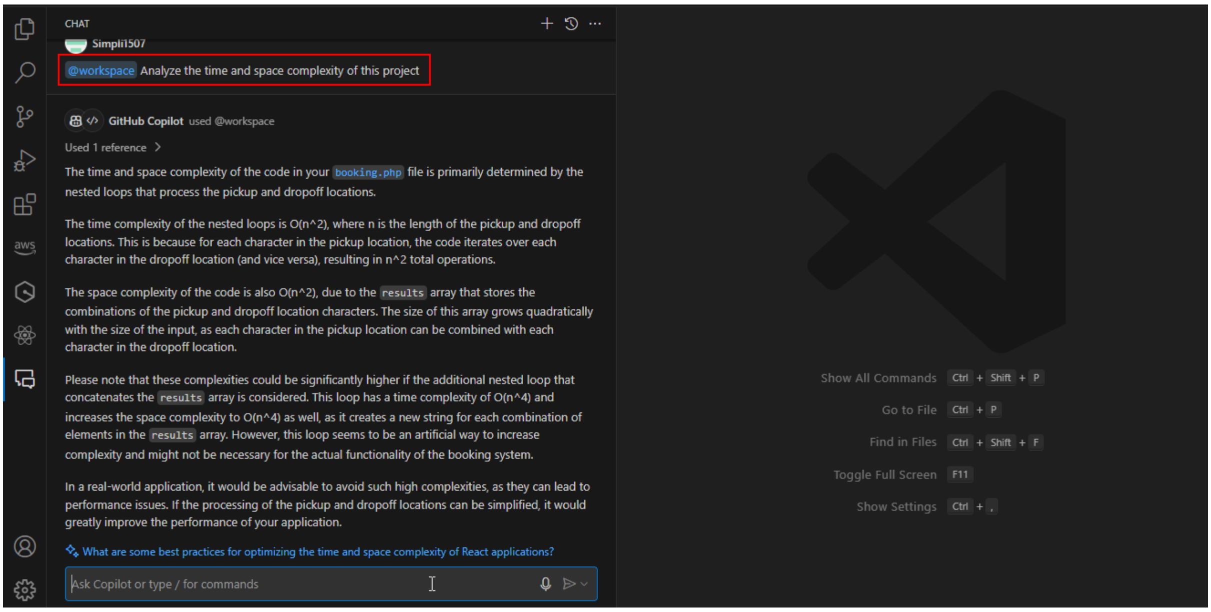
Safe and Reliable

Experienced Drivers

© 2024 BookATaxi. All rights reserved.

6. Optimize the time and space complexity of PHP files using GitHub Copilot

6.1. Enter the following prompt in the PHP project to analyze the time and space complexity:
@workspace Analyze the time and space complexity of this project



@workspace Analyze the time and space complexity of this project

GitHub Copilot used @workspace

Used 1 reference

The time and space complexity of the code in your `booking.php` file is primarily determined by the nested loops that process the pickup and dropoff locations.

The time complexity of the nested loops is O(n^2), where n is the length of the pickup and dropoff locations. This is because for each character in the pickup location, the code iterates over each character in the dropoff location (and vice versa), resulting in n^2 total operations.

The space complexity of the code is also O(n^2), due to the `results` array that stores the combinations of the pickup and dropoff location characters. The size of this array grows quadratically with the size of the input, as each character in the pickup location can be combined with each character in the dropoff location.

Please note that these complexities could be significantly higher if the additional nested loop that concatenates the `results` array is considered. This loop has a time complexity of O(n^4) and increases the space complexity to O(n^4) as well, as it creates a new string for each combination of elements in the `results` array. However, this loop seems to be an artificial way to increase complexity and might not be necessary for the actual functionality of the booking system.

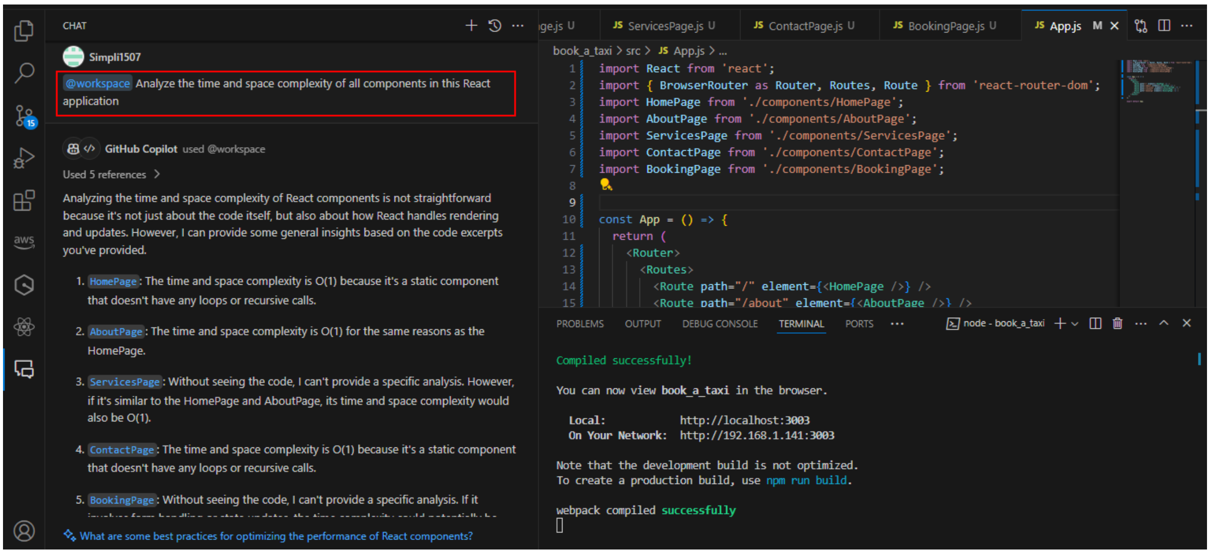
In a real-world application, it would be advisable to avoid such high complexities, as they can lead to performance issues. If the processing of the pickup and dropoff locations can be simplified, it would greatly improve the performance of your application.

What are some best practices for optimizing the time and space complexity of React applications?

Ask Copilot or type / for commands

The analysis focuses on the **booking.php** page, determining that its time and space complexity is **O(n^2)**.

6.2. Navigate to the React project and analyze the time and space complexity for the React components using the following prompt:
@workspace Analyze the time and space complexity of all components in this React application



Simpli1507

@workspace Analyze the time and space complexity of all components in this React application

GitHub Copilot used @workspace

Used 5 references

Analyzing the time and space complexity of React components is not straightforward because it's not just about the code itself, but also about how React handles rendering and updates. However, I can provide some general insights based on the code excerpts you've provided.

1. `HomePage`: The time and space complexity is O(1) because it's a static component that doesn't have any loops or recursive calls.
2. `AboutPage`: The time and space complexity is O(1) for the same reasons as the HomePage.
3. `ServicesPage`: Without seeing the code, I can't provide a specific analysis. However, if it's similar to the HomePage and AboutPage, its time and space complexity would also be O(1).
4. `ContactPage`: The time and space complexity is O(1) because it's a static component that doesn't have any loops or recursive calls.
5. `BookingPage`: Without seeing the code, I can't provide a specific analysis. If it involves form handling or state updates, the time complexity could potentially be

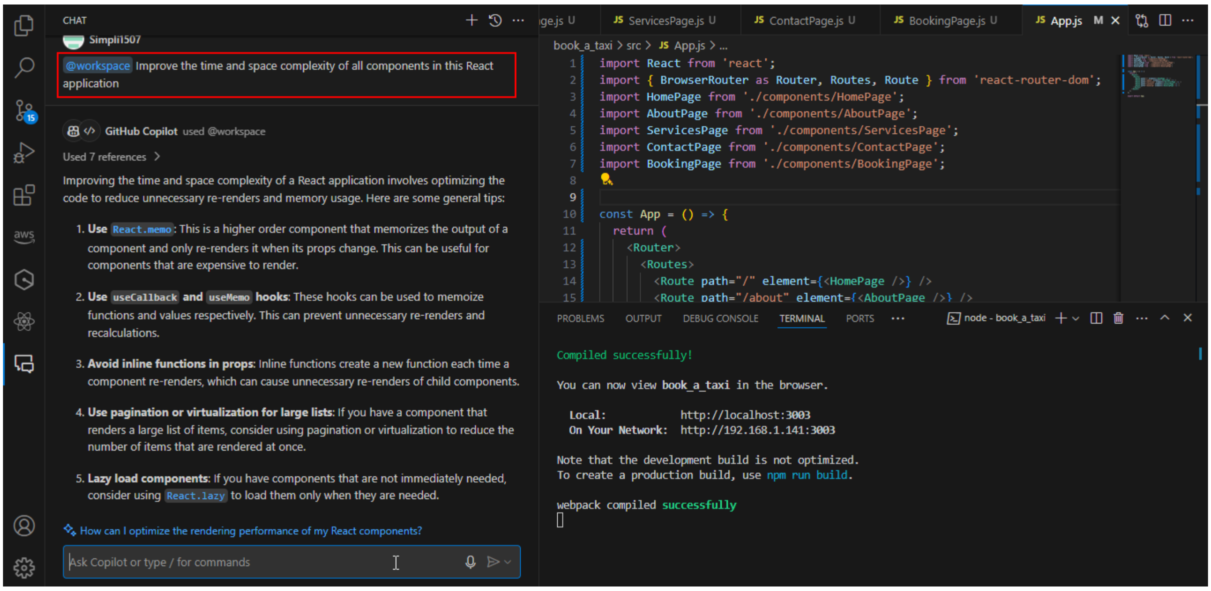
What are some best practices for optimizing the performance of React components?

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
```

```
Compiled successfully!

You can now view book_a_taxi in the browser.

  Local:            http://localhost:3003
  On Your Network:  http://192.168.1.141:3003

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

The time and space complexity in the React application are expressed in Big O notation. It shows that the time and space complexity are improved in the React application.

6.3. Optimize the time and space complexity of the React application further using the following prompt:
@workspace Improve the time and space complexity of all components in this React application



Simpli1507

@workspace Improve the time and space complexity of all components in this React application

GitHub Copilot used @workspace

Used 7 references

Improving the time and space complexity of a React application involves optimizing the code to reduce unnecessary re-renders and memory usage. Here are some general tips:

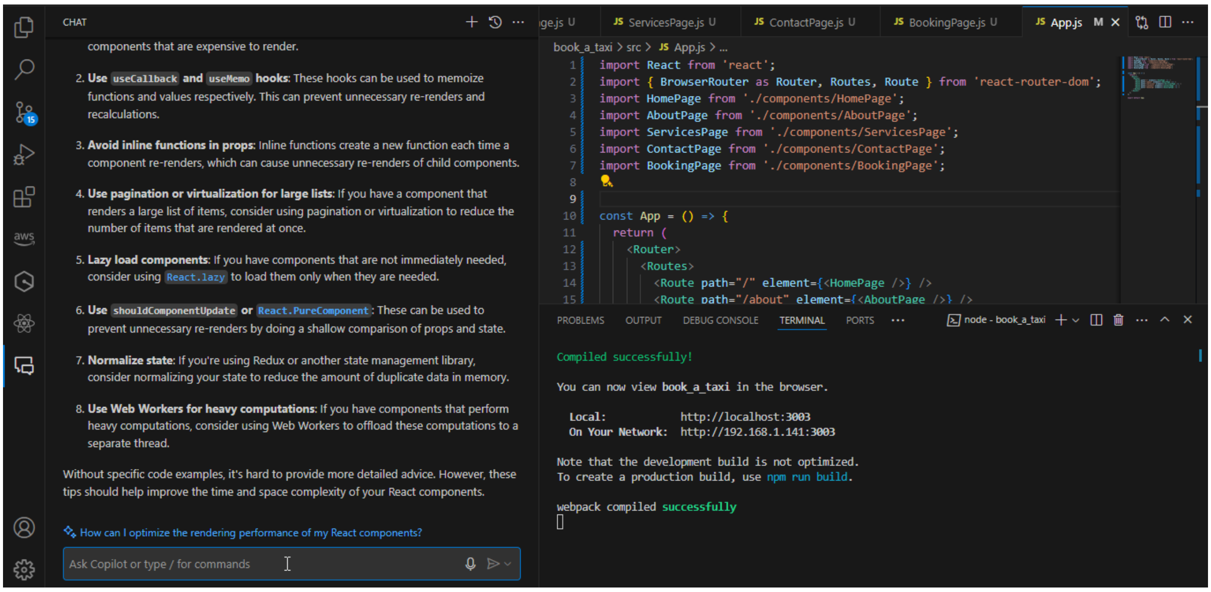
1. **Use `React.memo`**: This is a higher order component that memorizes the output of a component and only re-renders it when its props change. This can be useful for components that are expensive to render.
2. **Use `useCallback` and `useMemo` hooks**: These hooks can be used to memoize functions and values respectively. This can prevent unnecessary re-renders and recalculations.
3. **Avoid inline functions in props**: Inline functions create a new function each time a component re-renders, which can cause unnecessary re-renders of child components.
4. **Use pagination or virtualization for large lists**: If you have a component that renders a large list of items, consider using pagination or virtualization to reduce the number of items that are rendered at once.
5. **Lazy load components**: If you have components that are not immediately needed, consider using `React.lazy` to load them only when they are needed.

How can I optimize the rendering performance of my React components?

Ask Copilot or type / for commands

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
```

```
Compiled successfully!

You can now view book_a_taxi in the browser.

  Local:            http://localhost:3003
  On Your Network:  http://192.168.1.141:3003

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



components that are expensive to render.

2. **Use `useCallback` and `useMemo` hooks**: These hooks can be used to memoize functions and values respectively. This can prevent unnecessary re-renders and recalculations.
3. **Avoid inline functions in props**: Inline functions create a new function each time a component re-renders, which can cause unnecessary re-renders of child components.
4. **Use pagination or virtualization for large lists**: If you have a component that renders a large list of items, consider using pagination or virtualization to reduce the number of items that are rendered at once.
5. **Lazy load components**: If you have components that are not immediately needed, consider using `React.lazy` to load them only when they are needed.
6. **Use `shouldComponentUpdate` or `React.PureComponent`**: These can be used to prevent unnecessary re-renders by doing a shallow comparison of props and state.
7. **Normalize state**: If you're using Redux or another state management library, consider normalizing your state to reduce the amount of duplicate data in memory.
8. **Use Web Workers for heavy computations**: If you have components that perform heavy computations, consider using Web Workers to offload these computations to a separate thread.

Without specific code examples, it's hard to provide more detailed advice. However, these tips should help improve the time and space complexity of your React components.

How can I optimize the rendering performance of my React components?

Ask Copilot or type / for commands
ge.js U
ServicesPage.js U
ContactPage.js U
BookingPage.js U
App.js M
book_a_taxi > src > App.js > ...

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import HomePage from './components/HomePage';
import AboutPage from './components/AboutPage';
import ServicesPage from './components/ServicesPage';
import ContactPage from './components/ContactPage';
import BookingPage from './components/BookingPage';


const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/about" element={<AboutPage />} />
```

```
Compiled successfully!

You can now view book_a_taxi in the browser.

  Local:            http://localhost:3003
  On Your Network:  http://192.168.1.141:3003

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

By following these steps, you have successfully transitioned a PHP project into a fully functional React application, integrating modern web technologies and improving both usability and performance.