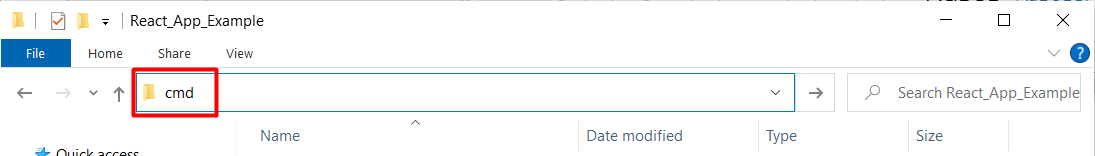
**Lab Work #2. Building a React.js**

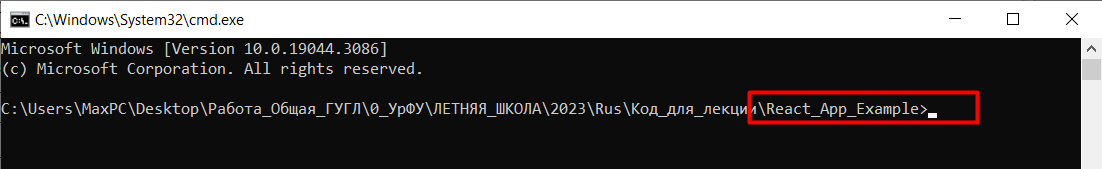
Create a new folder on the desktop with the name " React\_App\_Example".

Open the command prompt from the current folder.

To do this, you can simply enter the «**cmd**» command in the search bar of the current folder and press Enter.



After that, the command prompt opens:



Now that the preparation is over, let’s create a React application that connects to the Django application's API and performs some operations on the posts.

**Step 1: Set Up a React Application**

1. Make sure you have Node.js and npm installed on your machine.
2. Run the following command to create a new React application:

npx create-react-app my-app

Once the command finishes executing, navigate into the newly created directory:

cd my-app

**Step 2: Install Required Packages**

Install **axios**, which is a popular HTTP client for making API requests:

npm install axios

We will also install an additional module "react-router-dom" to organize navigation in the application:

npm install react-router-dom

Now we need to open the project in VS Code:

* + close the command prompt,
  + run VS Code,
  + open the project folder via the *File -> Open Folder* menu.
  + launch the terminal from the current project. This can be done via the *Terminal -> New Terminal* menu.

In the *terminal*, we need to go to the project folder so that further commands are executed in the appropriate folder.

To do this, type the command in the terminal:

cd my-app

**Step 3: Create Components**

Create a new folder «**components**» in «**src**» folder.

Create a **Home.js** component in the **src/components** directory and paste there the code below:

import React from 'react'

function Home() {

  return (

    <div>

        <h1>Main Page</h1>

        <p>Here you can place some content of the main page {`:)`}</p>

    </div>

    )

}

export default Home

Create a **PostList.js** component in the **src/components** directory and paste there the code below:

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

import axios from 'axios';

import { Link } from 'react-router-dom';

const PostList = () => {

  const [posts, setPosts] = useState([]);

  useEffect(() => {

    fetchPosts();

  }, []);

  const fetchPosts = async () => {

    try {

      const response = await axios.get('http://localhost:8000/posts/');

      setPosts(response.data);

      console.log(response.data);

    } catch (error) {

      console.error(error);

    }

  };

  return (

    <div>

      <h1>Posts</h1>

      <ul>

        {posts.map((post) => (

          <li key={post.id}>

            {post.title}

          </li>

        ))}

      </ul>

      <Link to="/posts/create">Create New Post</Link>

    </div>

  );

};

export default PostList;

Create a **PostCreate.js** component in the **src/components** directory:

import React, { useState } from 'react';

import axios from 'axios';

const PostCreate = () => {

  const [title, setTitle] = useState('');

  const [content, setContent] = useState('');

  const handleSubmit = async (e) => {

    e.preventDefault();

    try {

      await axios.post('http://127.0.0.1:8000/posts/', { title, content });

      setTitle('');

      setContent('');

      console.log('Post created successfully.');

    } catch (error) {

      console.error(error);

    }

  };

  return (

    <div>

      <h1>Create Post</h1>

      <form onSubmit={handleSubmit}>

        <div>

          <label>Title:</label>

          <input type="text" value={title} onChange={(e) => setTitle(e.target.value)} />

        </div>

        <div>

          <label>Content:</label>

          <textarea value={content} onChange={(e) => setContent(e.target.value)}></textarea>

        </div>

        <button type="submit">Create</button>

      </form>

    </div>

  );

};

export default PostCreate;

**Step 4: Create a Main App Component**

Replace the content of **src/App.js** with the following code:

import React from "react";

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

import PostList from "./components/PostList";

import PostCreate from "./components/PostCreate";

import Home from "./components/Home";

const App = () => {

  return (

    <Router>

      <div className="container">

        <h1>React Post App</h1>

        <nav>

          <ul>

            <li>

              <Link to="/">Home</Link>

            </li>

            <li>

              <Link to="/posts">Posts</Link>

            </li>

          </ul>

        </nav>

        <Routes>

          <Route exact path="/" element={<Home />} />

          <Route exact path="/posts" element={<PostList />} />

          <Route exact path="/posts/create" element={<PostCreate />} />

        </Routes>

      </div>

    </Router>

  );

};

export default App;

In the code above, an additional React module named "react-router-dom" is used to navigate between pages. It allows you to create navigation through various parts of the application. Initially, the React application is SAP app (SPA - single page application). With this additional module, we can create the appearance of a page transition, but in reality we just move from one component to another without redrawing the entire page as a whole.

Let's add some CSS styles to the main page and the page with posts.

**For this:**

In the **src/components** folder, create a **PostList.module.css** file and add the following styles to it:

.wraper{

    padding: 20px;

    display: flex;

    flex-direction: column;

    align-items: center;

}

.header{

    text-align: center;

    margin-bottom: 20px;

}

.post{

    width: 70%;

    /\* border: 1px solid gray; \*/

    border-radius: 5px;

    padding: 5px;

    margin-bottom: 10px;

    box-shadow: 0px 0px 10px 10px rgba(194, 194, 198, 0.4);

}

.postTitle{

    color: #000;

    font-weight: 600;

    margin-bottom: 10px;

    padding: 5px;

}

.postContent{

    background-color:bisque;

    border-radius: 5px;

    padding: 10px;

}

.createNew{

    display: flex;

    justify-content: center;

    align-items: center;

    width: 150px;

    border-radius: 5px;

    text-decoration: none;

    color: #000;

    font-weight: bold;

    height: 50px;

    background-color: rgb(202, 86, 4);

    margin-top: 20px;

    margin-left: 10px;

    margin-right: 10px;

    transition: all 0.2s;

}

.createNew:hover {

    box-shadow: 0px 0px 10px 10px rgba(155, 155, 157, 0.8);

}

.buttons{

    display: flex;

    flex-direction: row;

    justify-content: space-between;

    align-items: center;

}

Add the necessary css classes to the component code **src/components/PostList.js**. Replace the entire component code with the code below:

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

import axios from 'axios';

import { Link } from 'react-router-dom';

import styles from '../components/PostList.module.css';

const PostList = () => {

  const [posts, setPosts] = useState([]);

  useEffect(() => {

    fetchPosts();

  }, []);

  const fetchPosts = async () => {

    try {

      const response = await axios.get('http://localhost:8000/posts/');

      setPosts(response.data);

      console.log(response.data);

    } catch (error) {

      console.error(error);

    }

  };

  return (

    <div className={styles.wraper}>

      <h1 className={styles.header}>Posts</h1>

        {posts.map((post) => (

          <div className={styles.post} key={post.id}>

            <div className={styles.postTitle}>

              {post.title}

            </div>

            <div className={styles.postContent}>

              {post.content}

            </div>

          </div>

        ))}

      <div className={styles.buttons}>

        <Link className={styles.createNew} to="/posts/create">Create New Post</Link>

        <Link className={styles.createNew} to="/">Main page</Link>

      </div>

    </div>

  );

};

export default PostList;

In the **src** folder, create a **App.module.css** file and add the following styles to it:

.container{

    width: 95%;

    margin: 0 auto;

  }

  h1 {

    text-align: center;

  }

  nav {

    display: flex;

    flex-direction: row;

    justify-content: left;

    align-items: center;

background-color: #343a40;

height: 50px;

  }

  .navLink {

    color: white;

    text-decoration: none;

    margin-left: 10px;

    font-weight: bold;

  }

Add the necessary css classes to the component code **src/App.js**. Replace the entire component code with the code below:

import React from "react";

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

import PostList from "./components/PostList";

import PostCreate from "./components/PostCreate";

import Home from "./components/Home";

import styles from '../src/App.module.css';

const App = () => {

  return (

    <Router>

      <div className={styles.container}

        <h1 >React Post App</h1>

        <nav>

          <div><Link className={styles.navLink} to="/">Home</Link></div>

          <div><Link className={styles.navLink} to="/posts">Posts</Link></div>

        </nav>

        <Routes>

          <Route exact path="/" element={<Home />} />

          <Route exact path="/posts" element={<PostList />} />

          <Route exact path="/posts/create" element={<PostCreate />} />

        </Routes>

      </div>

    </Router>

  );

};

export default App;

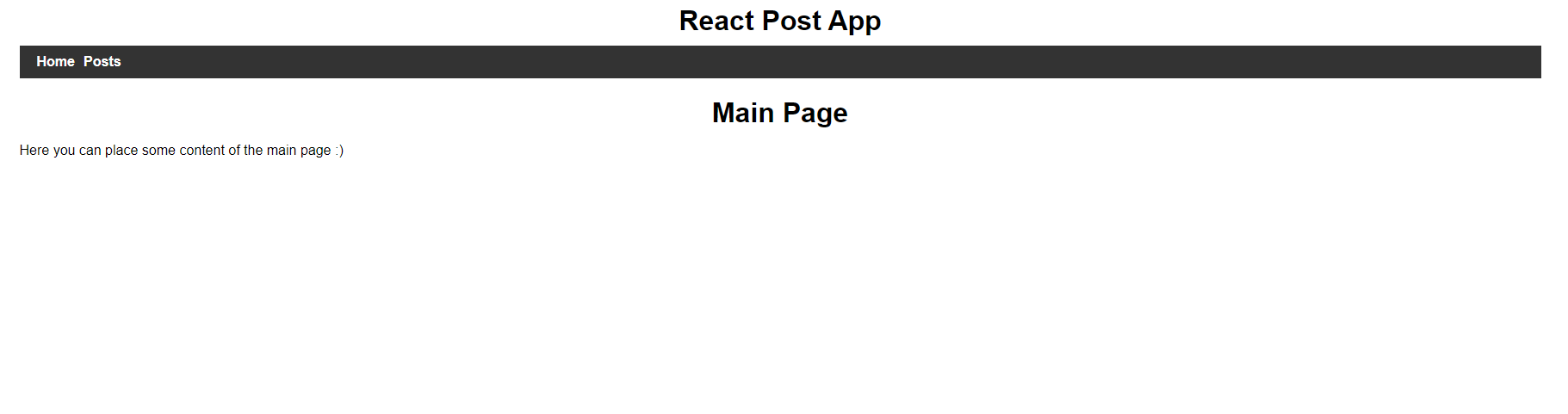
**Step 5: Start the React Development Server**

**Save** all the files and run the following command in your terminal:

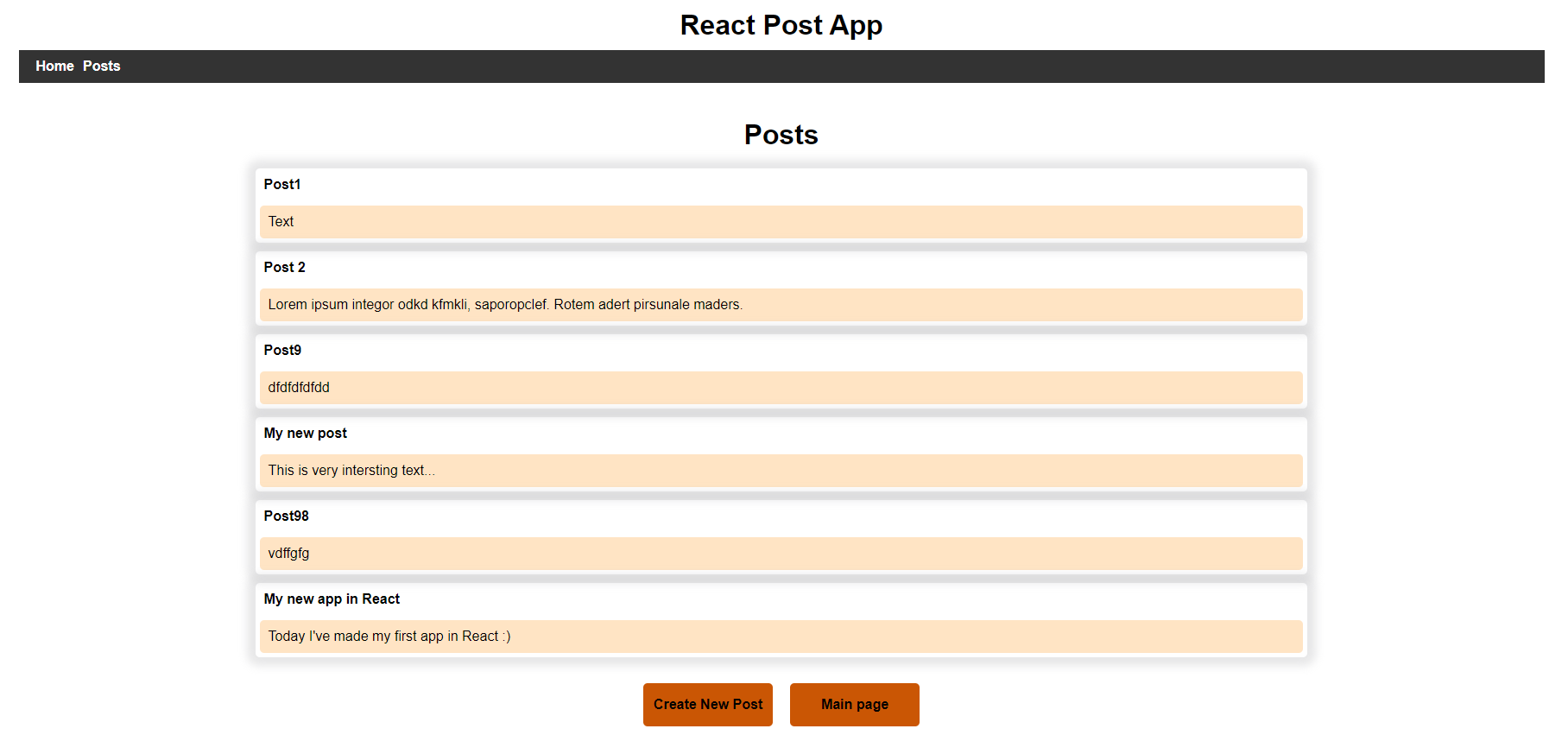
npm start

The React application will be running on **http://localhost:3000**. You can access it in your browser.

***Note:*** *For this React app it is necessary to launch the Django Post App project, which we created in the last laboratory work. This project will work as a remote API for our React application.*



If you click on the "Post" link, the application will send a request to the server (this is our previous Django application), receive data from the database and display it on the screen.



When you click on the "Create New Post" button, we go to a page where you can enter the data of a new post and save them in the database.

**Task:**

1. Add another "Contacts" component. Place markup with phone, email and address in it. Do your own styling.
2. On the main page, under the navigation bar, add a similar layout as shown below (you can use any content):

