PROMINEO TECH

Week 15: Coding Assignment

URL to GitHub Repository:

https://github.com/Rcruzn33/ReactCRUD-Countries.git

URL to Your Coding Assignment Video:
https://youtu.be/4FnC1LJ64hw

Instructions:

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed.
  Submit this document as a .PDF file in the LMS.

Coding Steps:

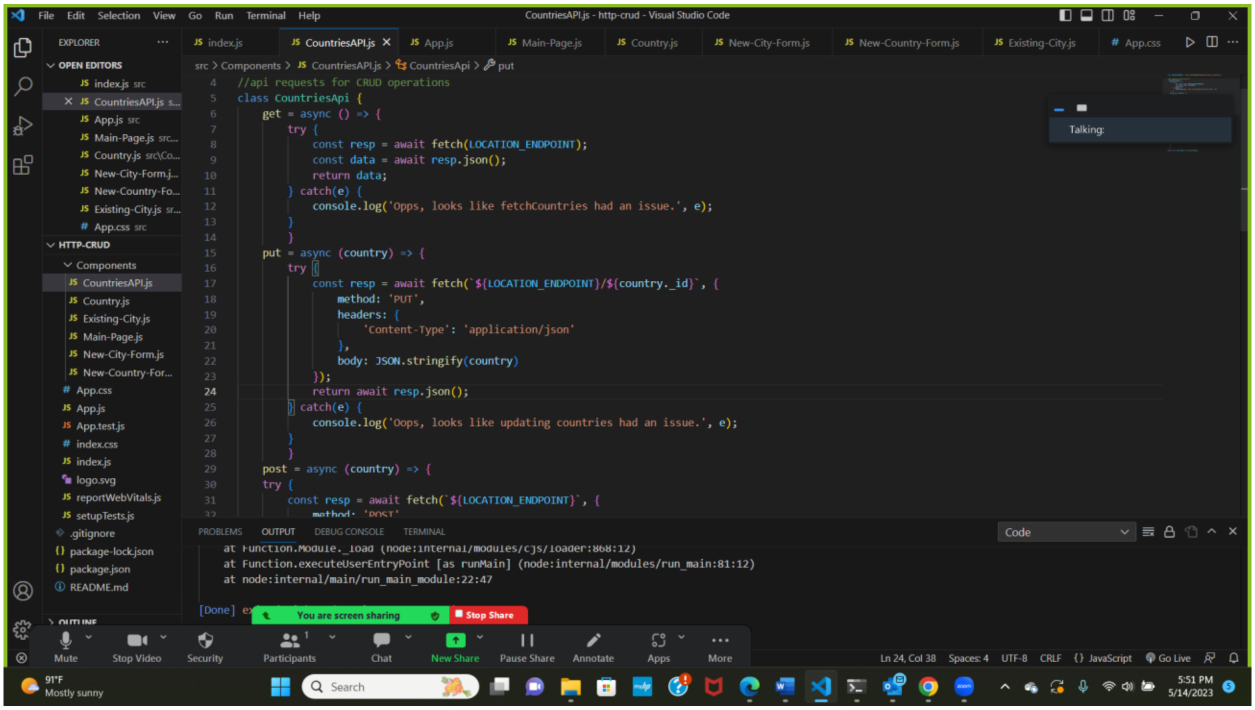
- Using the Houses API, or any open API of your choice you can find online, create a single page that allows for all 4 **CRUD** operations to be performed on a resource from the API.
- Create a React component (or more, if needed) to represent the resource.
- Make all forms and other necessary UI pieces their own components as reasonable.

Video Steps:

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
  - You can create a new meeting, start screen sharing, and start recording.
  - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.

Week 15: Coding Assignment

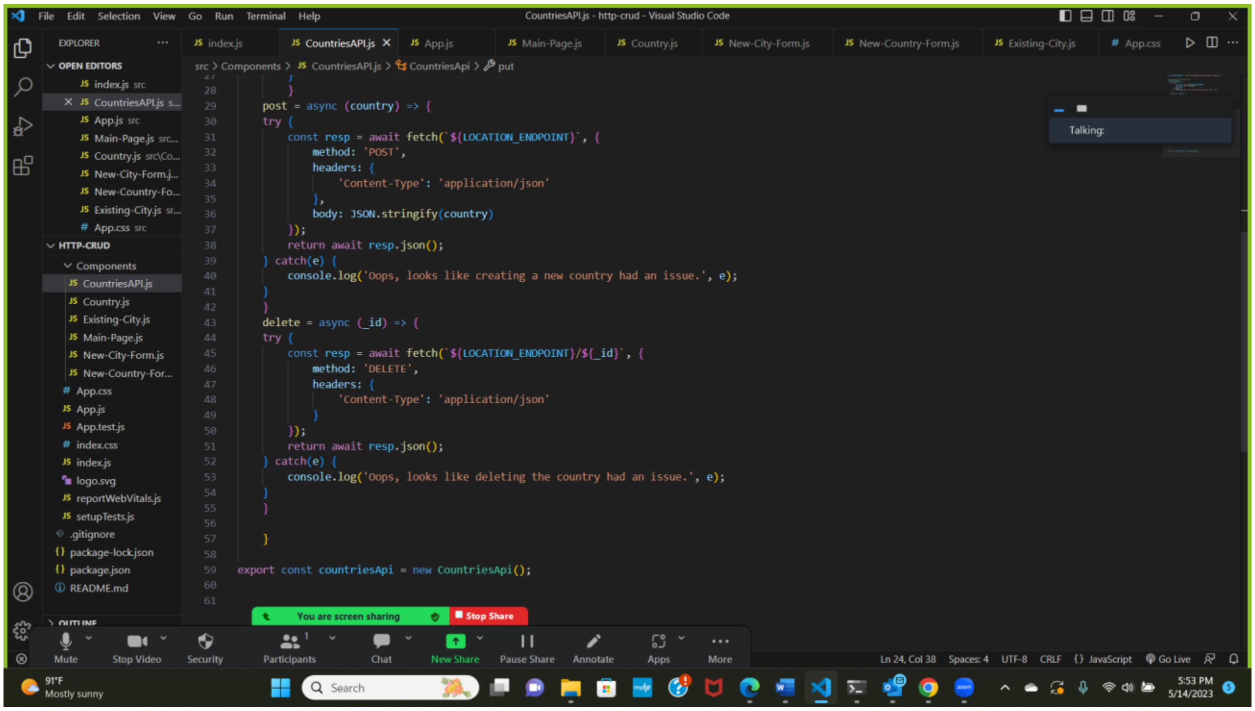
- Ensure the link you share is **PUBLIC** or **UNLISTED**!
- If it is not accessible by your grader, your project will be graded based on what they can access.



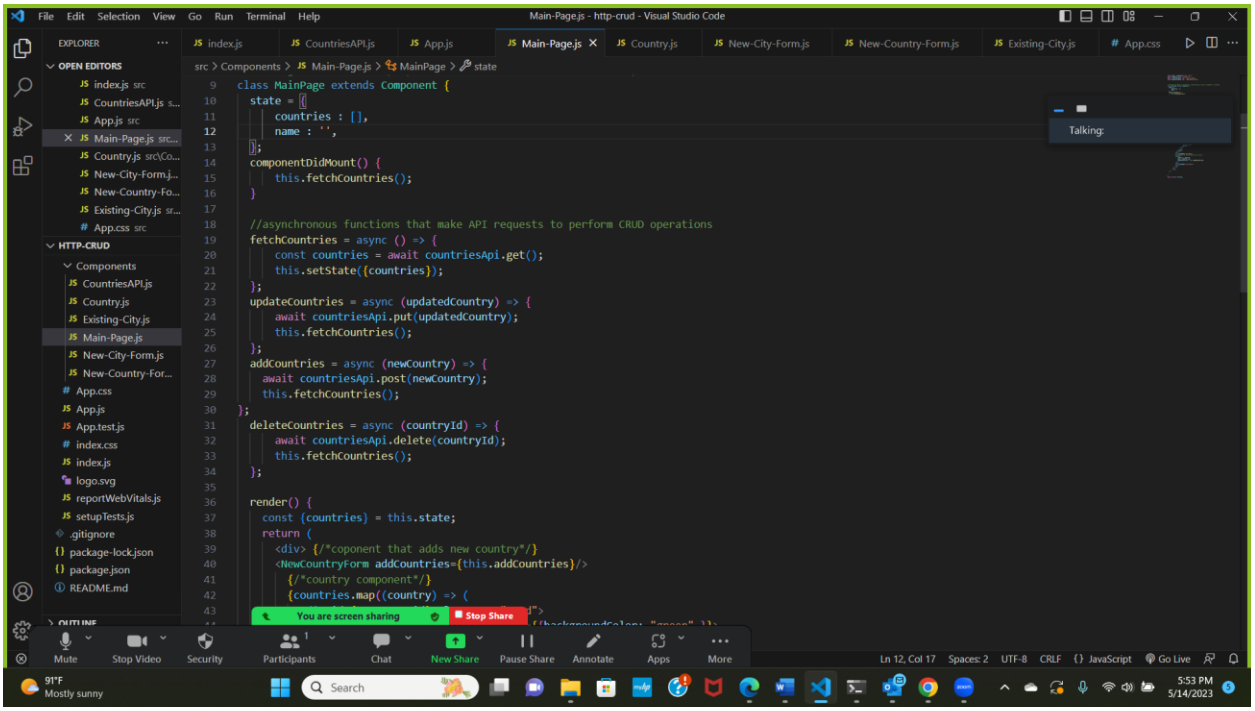
```javascript
//api requests for CRUD operations
class CountriesApi {
    get = async () => {
        try {
            const resp = await fetch(LOCATION_ENDPOINT);
            const data = await resp.json();
            return data;
        } catch(e) {
            console.log('Opps, looks like fetchCountries had an issue.', e);
        }
        }
    put = async (country) => {
        try {
            const resp = await fetch(`${LOCATION_ENDPOINT}/${country._id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(country)
            });
            return await resp.json();
        } catch(e) {
            console.log('Oops, looks like updating countries had an issue.', e);
        }
        }
    post = async (country) => {
    try {
        const resp = await fetch(`${LOCATION_ENDPOINT}`, {
            method: 'POST'
```

```
    at Function.Module._load (node:internal/modules/cjs/loader:868:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:22:47

[Done] ex
```

Week 15: Coding Assignment

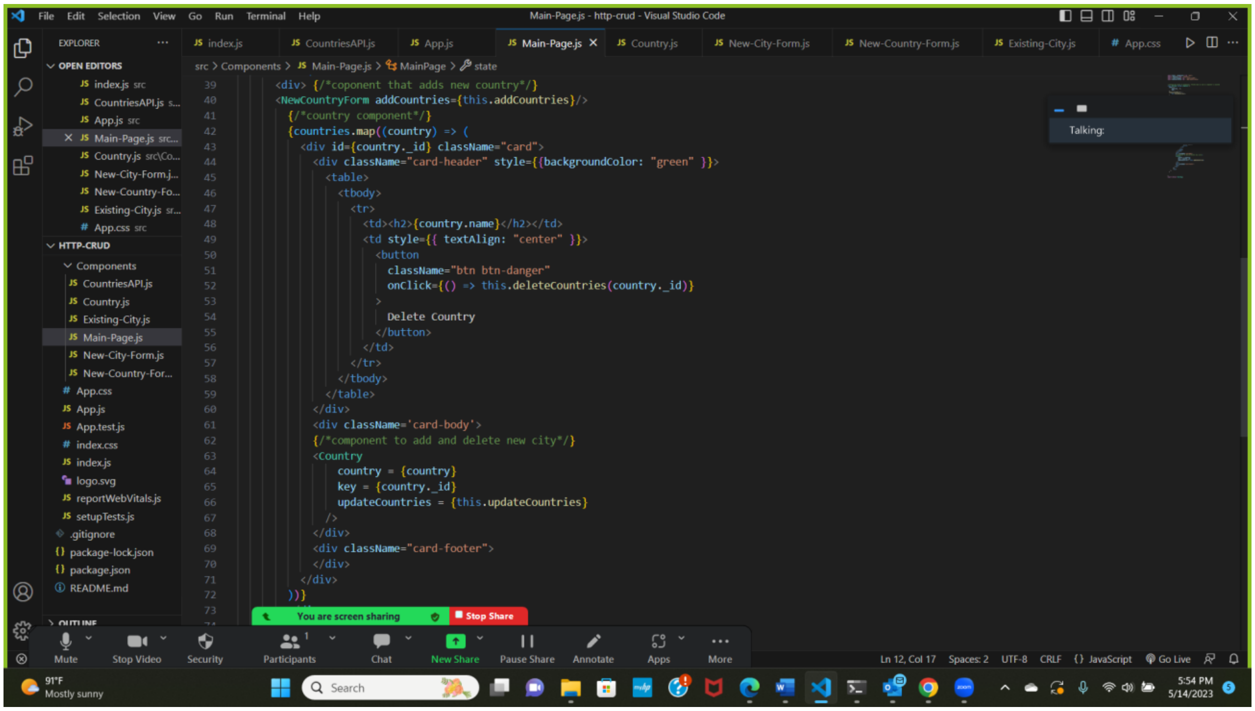

CountriesAPI.js - http-crud - Visual Studio Code
src > Components > CountriesAPI.js > CountriesApi > put
You are screen sharing
Stop Share
Ln 24, Col 38 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Talking:

```javascript
      }
  post = async (country) => {
  try {
      const resp = await fetch(`${LOCATION_ENDPOINT}`, {
          method: 'POST',
          headers: {
              'Content-Type': 'application/json'
          },
          body: JSON.stringify(country)
      });
      return await resp.json();
  } catch(e) {
      console.log('Oops, looks like creating a new country had an issue.', e);
  }
  }
  delete = async (_id) => {
  try {
      const resp = await fetch(`${LOCATION_ENDPOINT}/${_id}`, {
          method: 'DELETE',
          headers: {
              'Content-Type': 'application/json'
          }
      });
      return await resp.json();
  } catch(e) {
      console.log('Oops, looks like deleting the country had an issue.', e);
  }
  }

  }

export const countriesApi = new CountriesApi();
```

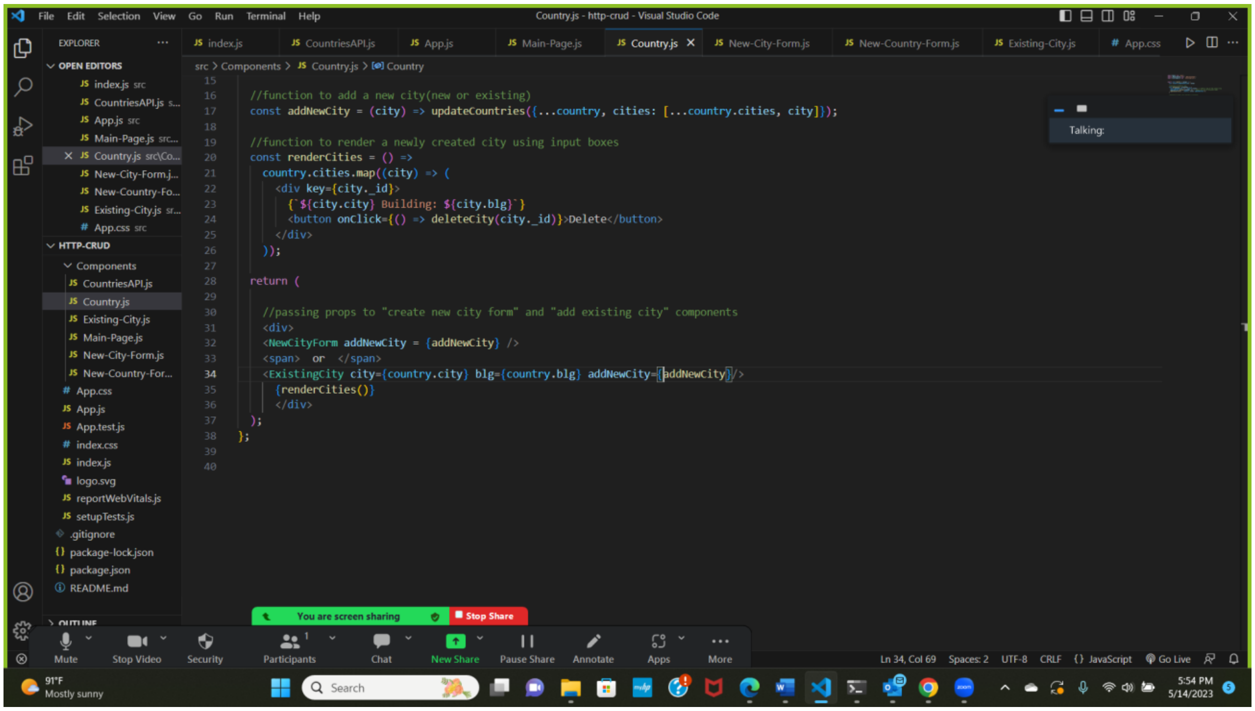

Main-Page.js - http-crud - Visual Studio Code
src > Components > Main-Page.js > MainPage > state
You are screen sharing
Stop Share
Ln 12, Col 17 Spaces: 2 UTF-8 CRLF JavaScript Go Live
Talking:

```javascript
class MainPage extends Component {
  state = {
    countries : [],
    name : '',
  };
  componentDidMount() {
    this.fetchCountries();
  }

  //asynchronous functions that make API requests to perform CRUD operations
  fetchCountries = async () => {
    const countries = await countriesApi.get();
    this.setState({countries});
  };
  updateCountries = async (updatedCountry) => {
    await countriesApi.put(updatedCountry);
    this.fetchCountries();
  };
  addCountries = async (newCountry) => {
   await countriesApi.post(newCountry);
   this.fetchCountries();
};
  deleteCountries = async (countryId) => {
    await countriesApi.delete(countryId);
    this.fetchCountries();
  };

  render() {
   const {countries} = this.state;
   return (
    <div> {/*coponent that adds new country*/}
    <NewCountryForm addCountries={this.addCountries}/>
     {/*country component*/}
     {countries.map((country) => (
```

Week 15: Coding Assignment


Main-Page.js - http-crud - Visual Studio Code
src > Components > Main-Page.js > MainPage > state
You are screen sharing
Stop Share
Ln 12, Col 17 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live

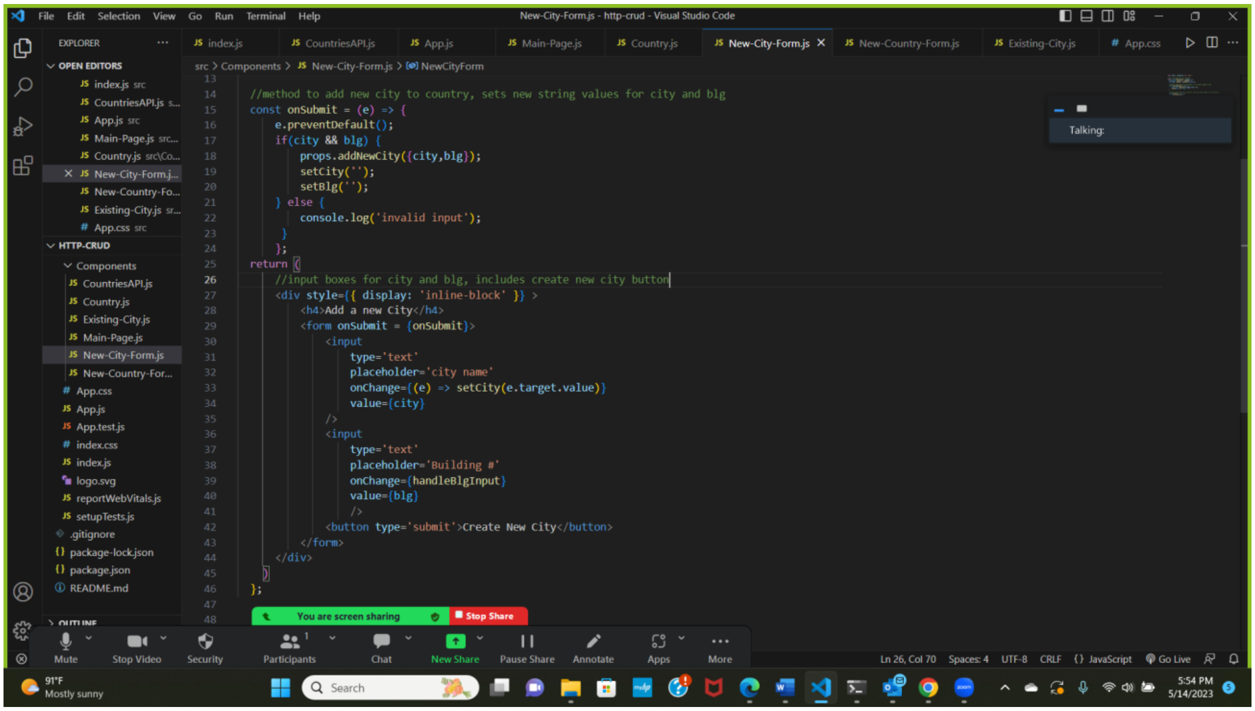
```
<div> {/*coponent that adds new country*/}
<NewCountryForm addCountries={this.addCountries}/>
  {/*country component*/}
  {countries.map((country) => (
    <div id={country._id} className="card">
      <div className="card-header" style={{backgroundColor: "green" }}>
        <table>
          <tbody>
            <tr>
              <td><h2>{country.name}</h2></td>
              <td style={{ textAlign: "center" }}>
                <button
                  className="btn btn-danger"
                  onClick={() => this.deleteCountries(country._id)}
                >
                  Delete Country
                </button>
              </td>
            </tr>
          </tbody>
        </table>
      </div>
      <div className='card-body'>
      {/*component to add and delete new city*/}
      <Country
          country = {country}
          key = {country._id}
          updateCountries = {this.updateCountries}
        />
      </div>
      <div className="card-footer">
      </div>
    </div>
  ))}
```


Country.js - http-crud - Visual Studio Code
src > Components > Country.js > Country
You are screen sharing
Stop Share
Ln 34, Col 69 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live

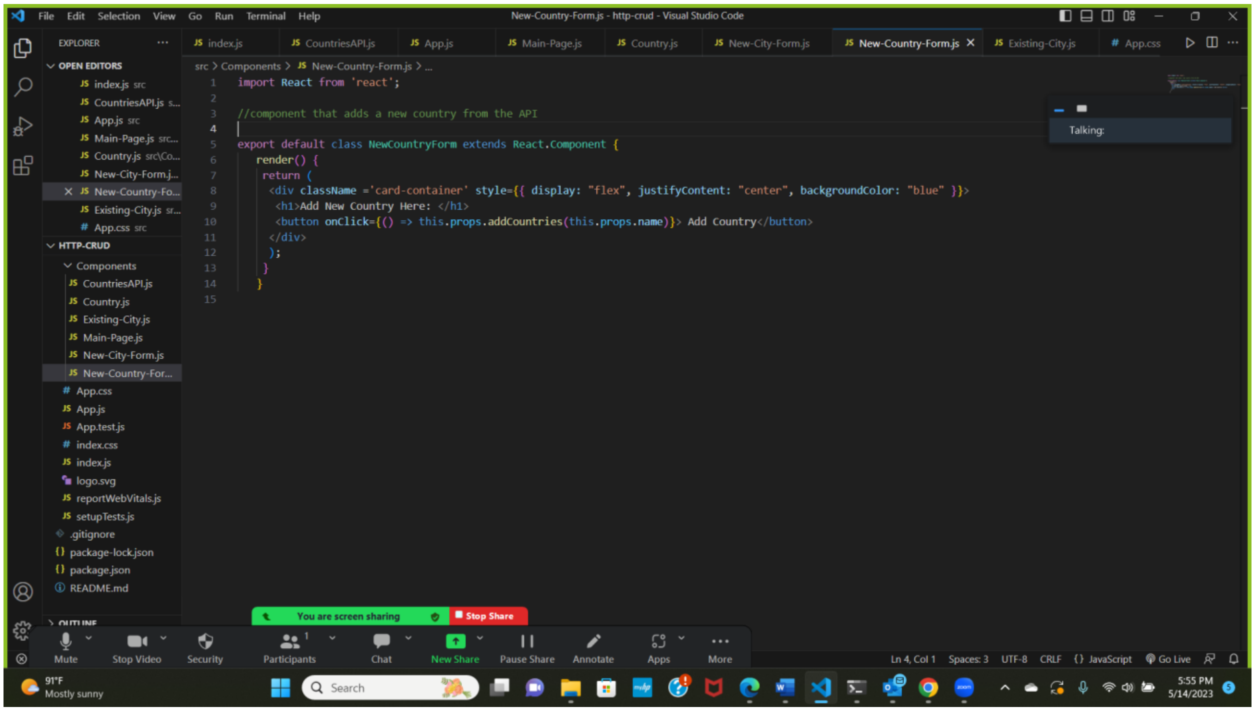
```

  //function to add a new city(new or existing)
  const addNewCity = (city) => updateCountries({...country, cities: [...country.cities, city]});

  //function to render a newly created city using input boxes
  const renderCities = () =>
    country.cities.map((city) => (
      <div key={city._id}>
        {`${city.city} Building: ${city.blg}`}
        <button onClick={() => deleteCity(city._id)}>Delete</button>
      </div>
    ));

  return (

    //passing props to "create new city form" and "add existing city" components
    <div>
    <NewCityForm addNewCity = {addNewCity} />
    <span>  or  </span>
    <ExistingCity city={country.city} blg={country.blg} addNewCity={addNewCity}/>
      {renderCities()}
      </div>
  );
};
```

PROMINEO TECH

Week 15: Coding Assignment

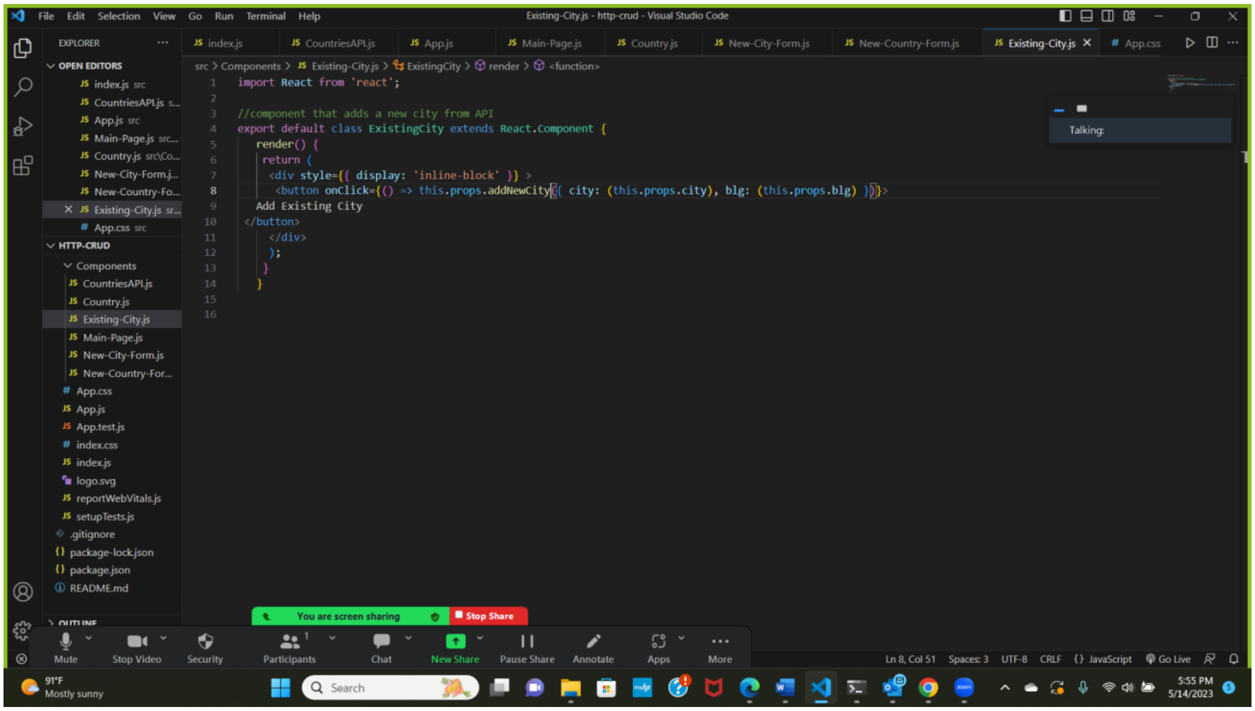


```

    //method to add new city to country, sets new string values for city and blg
    const onSubmit = (e) => {
        e.preventDefault();
        if(city && blg) {
            props.addNewCity({city,blg});
            setCity('');
            setBlg('');
        } else {
            console.log('invalid input');
         }
        };
    return (
        //input boxes for city and blg, includes create new city button
        <div style={{ display: 'inline-block' }} >
            <h4>Add a new City</h4>
            <form onSubmit = {onSubmit}>
                <input
                    type='text'
                    placeholder='city name'
                    onChange={(e) => setCity(e.target.value)}
                    value={city}
                />
                <input
                    type='text'
                    placeholder='Building #'
                    onChange={handleBlgInput}
                    value={blg}
                    />
                <button type='submit'>Create New City</button>
            </form>
        </div>
        )
    };


```



```
import React from 'react';

//component that adds a new country from the API

export default class NewCountryForm extends React.Component {
   render() {
    return (
     <div className ='card-container' style={{ display: "flex", justifyContent: "center", backgroundColor: "blue" }}>
      <h1>Add New Country Here: </h1>
      <button onClick={() => this.props.addCountries(this.props.name)}> Add Country</button>
     </div>
     );
    }
   }

```

Week 15: Coding Assignment



```
import React from 'react';

//component that adds a new city from API
export default class ExistingCity extends React.Component {
   render() {
    return (
     <div style={{ display: 'inline-block' }} >
      <button onClick={() => this.props.addNewCity({ city: (this.props.city), blg: (this.props.blg) })}>
   Add Existing City
  </button>
     </div>
    );
   }
  }
```