Programiranje u smart grid sistemima

Slađana Turudić - sladjanaturudic@uns.ac.rs
Teodora Ruvčeski - teodoraruvceski@uns.ac.rs

Kako se polaže predmet?

- Odbrana projekta 70 bodova; 30 bodova usmeni
- Projekat radi svaki student zasebno
- Svaki student treba da napravi repozitorijum na GitHub-u i doda asistente kako bi se pratio progres

Kako se polaže predmet?

- Gradivo:
  - Frontend - React Components
    - React Hooks
    - React Services
  - Backend - .NET Core 5.0
    - EntityFramework
    - SQL Server

Način izvedbe nastave

- Jedna pauza od 15 minuta

React

Šta je React?

- React je JavaScript *open-source* biblioteka za razvoj korisničkog interfejsa
- Kreirana od strane Facebook-a
- Bazira se na komponentama
- Može se koristiti za razvoj *single-page,* mobilnih, ili *server-side* aplikacija

Šta je React?

- TypeScript -> superset JavaScript-a (JS sa tipovima podataka)
- Prilikom izvršavanja se automatski prevodi u JavaScript

Šta je React?

- Bazira se na MVC šablonu
- Sav kod se izvršava u pretraživaču
- Dokumentacija Getting Started – React (reactjs.org)

Requirements and setup

Node.js & npm

- Provera trenutne verzije Node.js
  - `node -v`
- Provera trenutne verzije npm-a
  - `npm -v`
- Ukoliko verzija nije prikazana ili ukoliko dođe do greške, instalirajte Node.js sa sledećeg linka
- Za instalaciju Node.js-a koristimo npm (node package manager)
- Uputstvo za instalaciju https://docs.npmjs.com/downloading-and-installing-node-js-and-npm
- Napomena - poželjna verzija Node version >= 8.10 i NPM version >= 5.6

Instalacija

- Command Line Interface omogućava korisnicima da kreiraju i upravljaju projektima iz komande linije
- Automatizuje procese kreiranja projekata, dodavanja novih komponenti, modula itd.
- Instalacija
  - `npm install -g create-react-app`

Kreiranje i pokretanje aplikacije

- Kreiranje nove React aplikacije
  - `npx create-react-app my-react-app`
  - `npx create-react-app my-react-app --template typescript`
- Pokretanjem komanda će se:
  - kreirati folder sa istim nazivom
  - skinuti neophodne React biblioteke
- Pozicioniranje u folder aplikacije
  - `cd my-react-app`
- Pokretanje aplikacije
  - `npm start`
- Aplikacija će po defaultu biti podignuta na portu 3000
  - http://localhost:3000/

Kreiranje i pokretanje aplikacije

- Sledeća greška ukazuje da je port trenutno zauzet
  - `Port 3000 is already in use`
- U tom slučaju potrebno je definisati port unutar fajla package.json

```
"scripts": {
  "start": "set PORT=3001 && react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "eslint": "eslint --ext .js --ext .jsx src",
  "eslint:fix": "eslint --fix --ext .js --ext .jsx src"
},
```

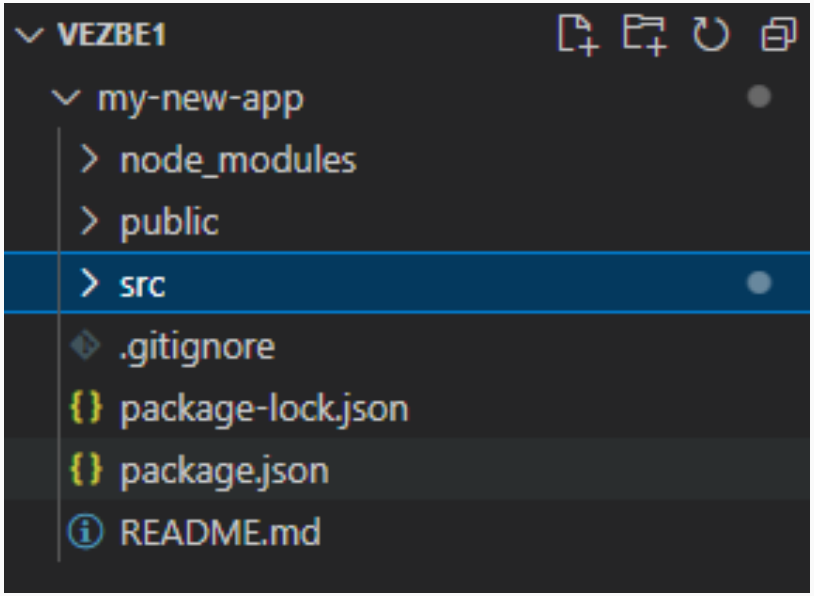
Proxy

- Git
  - `git config --global http.proxy http://192.168.77.100:8080`
- npm
  - `npm config set proxy http://192.168.77.100:8080`
  - `npm config set https-proxy http://192.168.77.100:8080`

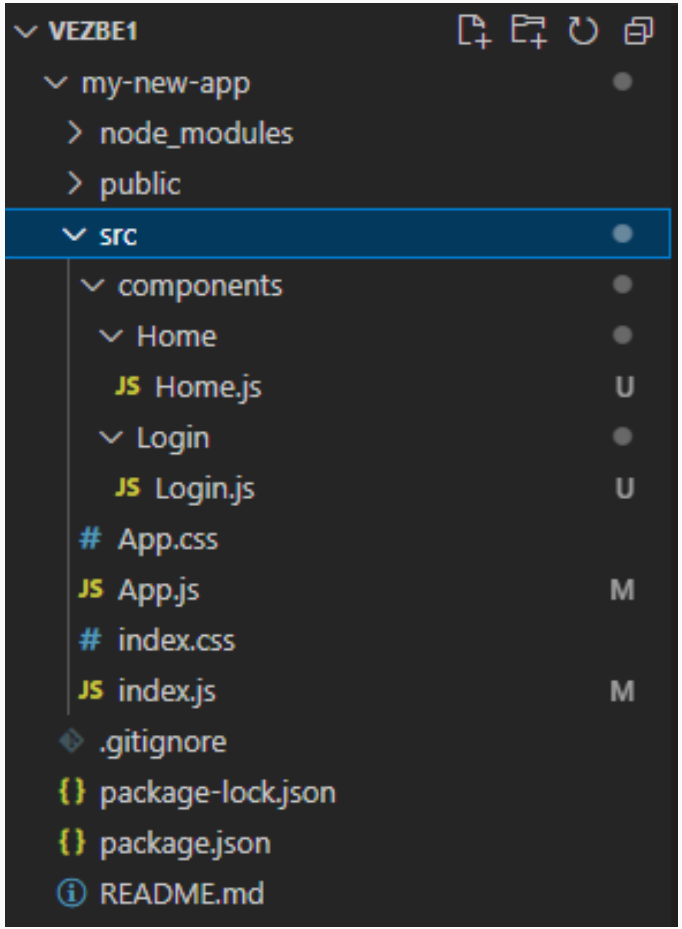
Edit `src/App.js` and save to reload.

Learn React

React Project Structure



- /node_modules/
  - folder unutar koga npm instalira sve pakete
  - ukoliko ovaj folder ne postoji potrebno je u root-u aplikacije pokrenuti `npm install`
- /src/
  - svi fajlovi koji čine projekat se nalaze u ovom folderu
- package.json
  - fajl unutar koga se handle-ju sve third-party biblioteke i moduli aplikacije
- tsconfig.json (za typescript projekte)
  - TypeScript konfiguracija



- /src/
  - folder sa svim fajlovima projekta
- /components/
  - folder sa svim komponentama
- /Home/
  - folder u kom se nalazi Home komponenta i svi fajlovi koji su potrebni za njeno renderovanje
- index.js
  - fajl unutar kog se poziva render()
- App.css
  - style sheet fajl za definisanje stilova
- App.js
  - glavna komponenta generisana po default-u unutar koje renderujemo ostale komponente
- .gitignore
  - fajl unutar kog definišemo tipove fajlova koje ne želimo da uključimo u git commit

React Components

React Components

- Funkcije koje vraćaju HTML elemente
- Komponente su nezavisni delovi koda koji podstiču *reusability* koda
- Naziv komponente mora imati veliko početno slovo
- Tipovi komponenti:
  - Class components
  - Function components

Class component

- Klasna komponenta mora sadržati `extends React.Component` statement
- Kreira se nasleđivanje `React.Component` komponente što omogućava pristup njenim funkcijama kao što je `render()` za *return* H
- Primer:

```
class Car extends React.Component {
  render() {
    return <h2>Hi, I am a Car!</h2>;
  }
}
```

Function component

- Function komponenta takođe vraća HTML elemente
- Ponaša se slično kao i klasna komponenta, ali nije potrebno nasleđivati React Component
- Primer

```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}
```

Rendering a Component

- Ako imamo komponentu Car koja vraća <h2> element možemo je iskoristiti za renderovanje:

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```

- Da bismo iskoristili komponentu koristimo sintaksu sličnu HTML-u:

```
<Car />
```

Component in Component

- Možemo da koristimo komponente unutar drugih komponenti
- Podstiče se decoupling

```
function Car() {
  return <h2>I am a Car!</h2>;
}

function Garage() {
  return (
    <>
      <h1>Who lives in my Garage?</h1>
      <Car />
    </>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Garage />);
```

```
function Car() {
  return <h2>I am a Car!</h2>;
}

function Garage() {
  return (
    <>
      <h1>Who lives in my Garage?</h1>
      <Car />
    </>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Garage />);
```

Components in Files

- React je veoma fokusiran na re-usability
- Preporuka je da se komponente nalaze u zasebnim fajlovima
- Za svaku komponentu kreiramo novi .js fajl
- Primer:

```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}

export default Car;
```

Components in Files

- Kako bismo mogli da koristimo kreiranu komponentu unutar index.js fajla, moramo je importovati
- Primer:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import Car from './Car.js';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```

React Props

React Props

- Props = properties
- Props su argumenti koji se prosleđuju React componentama
- Props prosleđujemo pomoću HTML atributa

```
function Garage() {
  return (
    <>
      <h1>Who lives in my garage?</h1>
      <Car brand="Ford" />
    </>
  );
}
```

```
function Car(props) {
  return <h2>I am a { props.brand }!</h2>;
}
```

React Props

- Upotreba JS konstanti i varijabli unutar HTML koda:

```
function Car(props) {
  return <h2>I am a { props.brand }!</h2>;
}

function Garage() {
  const carName = "Ford";
  return (
    <>
      <h1>Who lives in my garage?</h1>
      <Car brand={ carName } />
    </>
  );
}
```

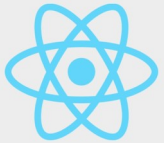
```
function Car(props) {
  return <h2>I am a { props.brand.model }!</h2>;
}

function Garage() {
  const carInfo = { name: "Ford", model: "Mustang" };
  return (
    <>
      <h1>Who lives in my garage?</h1>
      <Car brand={ carInfo } />
    </>
  );
}
```

Termin 1 - zadatak

- Kreirati React projekat sa 4 komponente:
  - prva komponenta treba u svom sadržaju da ima logo Angulara
  - druga komponenta treba u svom sadržaju da ima logo Reacta
  - treća komponenta treba u sadržaju da ima logo Vue.js
  - u App.js se prikazuju sve tri komponente jedna do druge

*Slike skinuti sa interneta
*Iskoristiti PROPS


Angular React Vue