

*Prolećni semestar, 2020/21*

*PREDMET: SE211 KONSTRUISANJE SOFTVERA*

**Projektni Zadatak**

Ime i prezime: **Aleksa Cekić**

Broj indeksa: **4173**

Datum izrade: **29.05.2021.**

# Sadržaj:

1. Opis projekta
2. Activity diagram za contact
3. Uvod u React
4. Opis Seleniuma
5. Kreiranje projekta
6. Dodavanje biblioteka u projektu
7. Dodavanje animacija
8. Struktura projekta
9. Testovi
10. Zaključak
11. Literatura

# Opis projekta:

Ovaj projekat predstavlja portfolio koji je pisan u **JavaScript** biblioteci **React**.

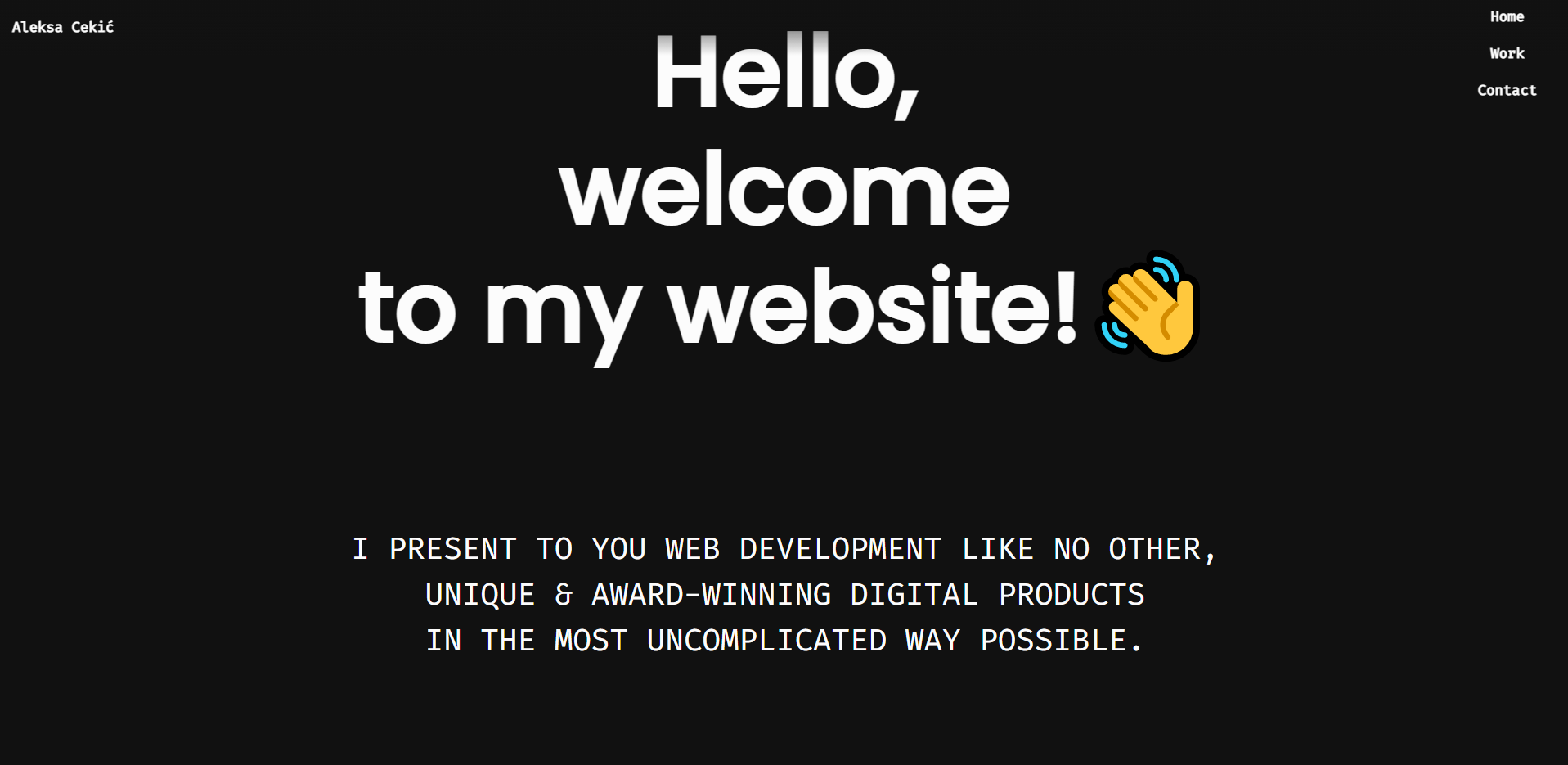
Projekat je rađen korišćenjem raznih biblioteka kao što su node-sass, bootstrap, material-ui za stilizovanje, json-server za čuvanje kontakt informacija, uuid za generisanje id-eva svake kontakt forme.

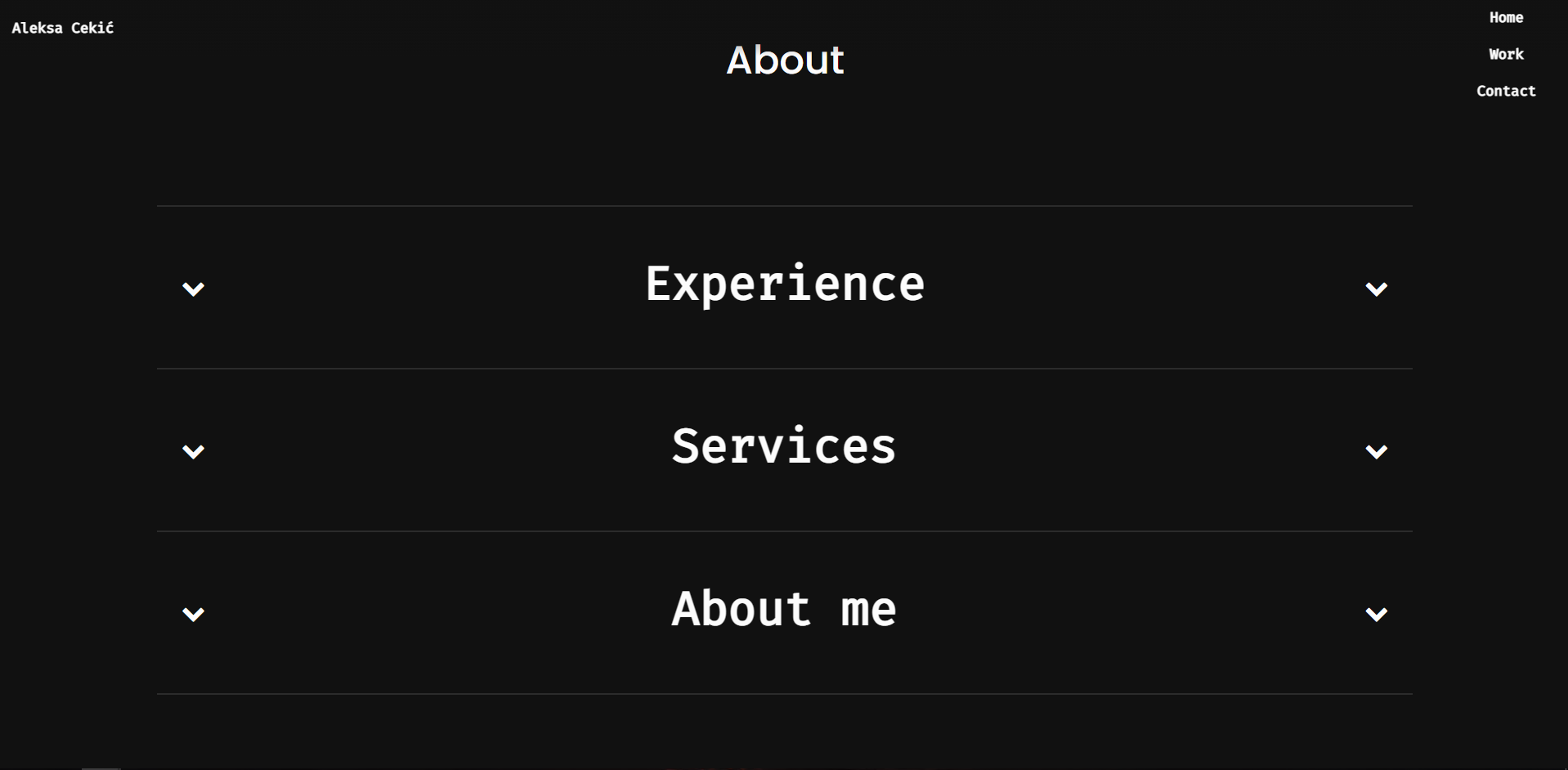
Za testiranje ove aplikacije korišćen je **Selenium** i **Mocha**.

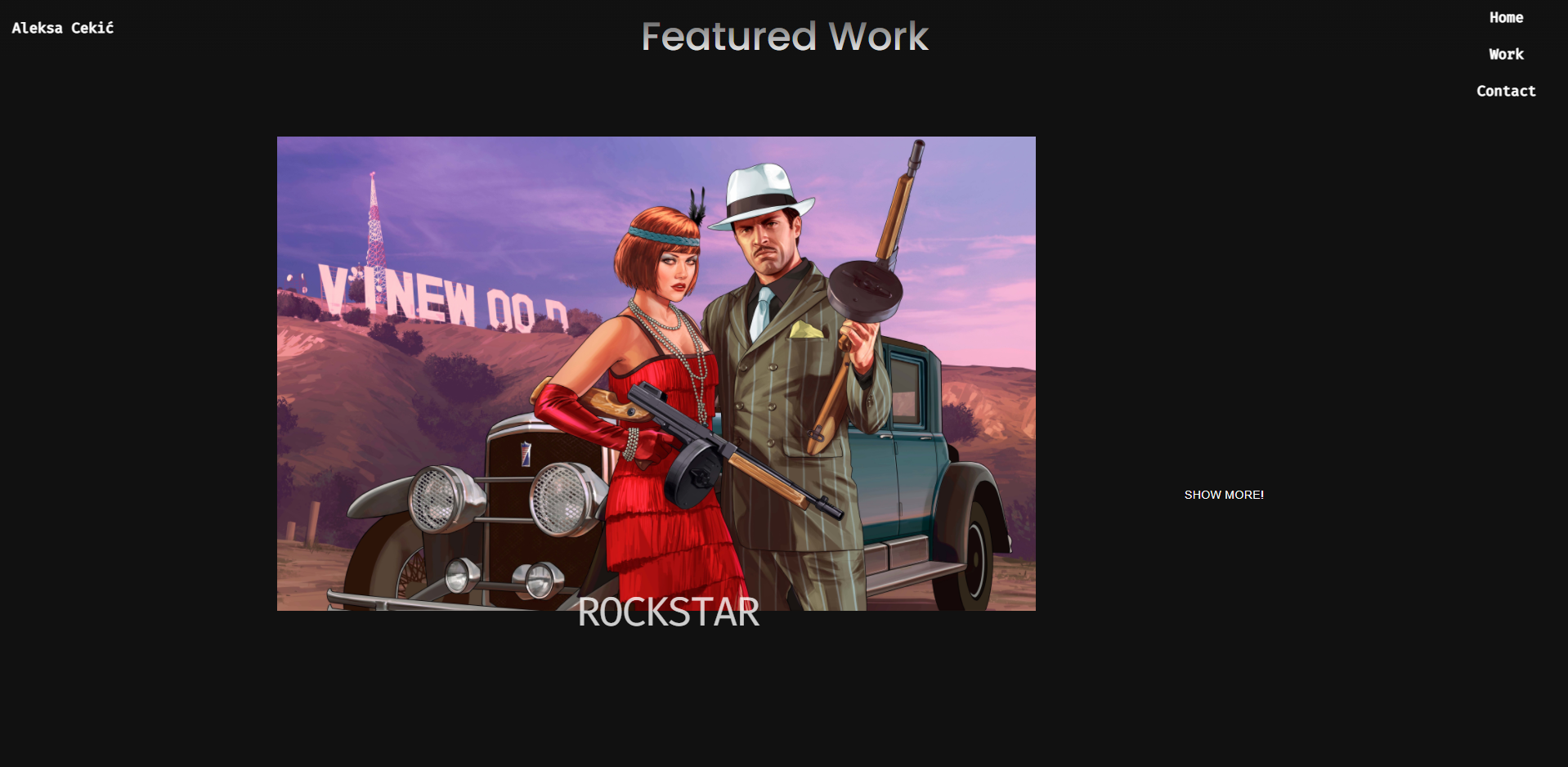
Projekat koristi biblioteku **AOS** (Animate on scroll) za animacije.

Projekat ima 3 strane: **Home**, **Work** I **Contact**

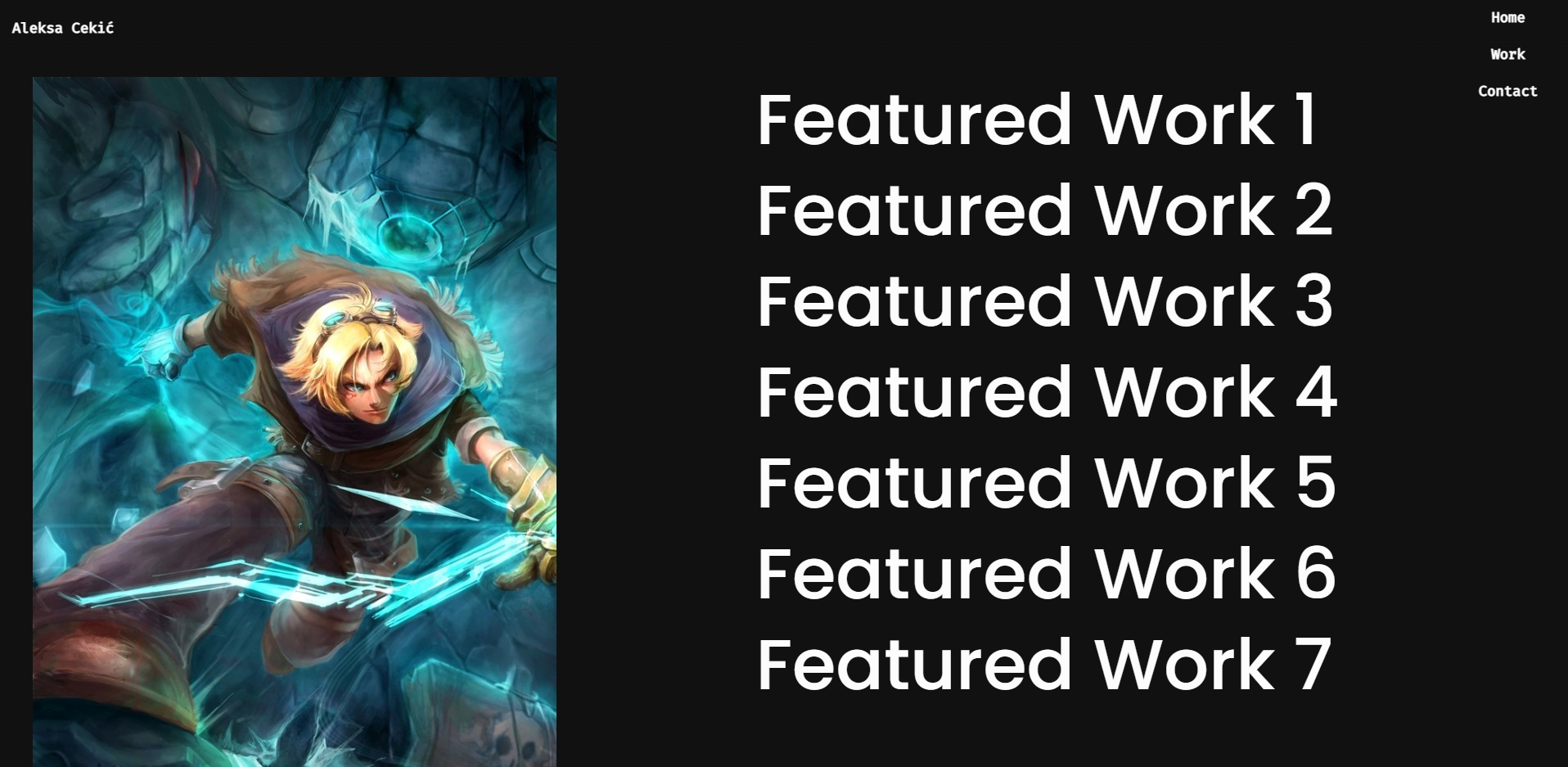
Strana Home predstavlja glavnu stranu koja sadrži sve bitne informacije, opise, iskustva, projekte.



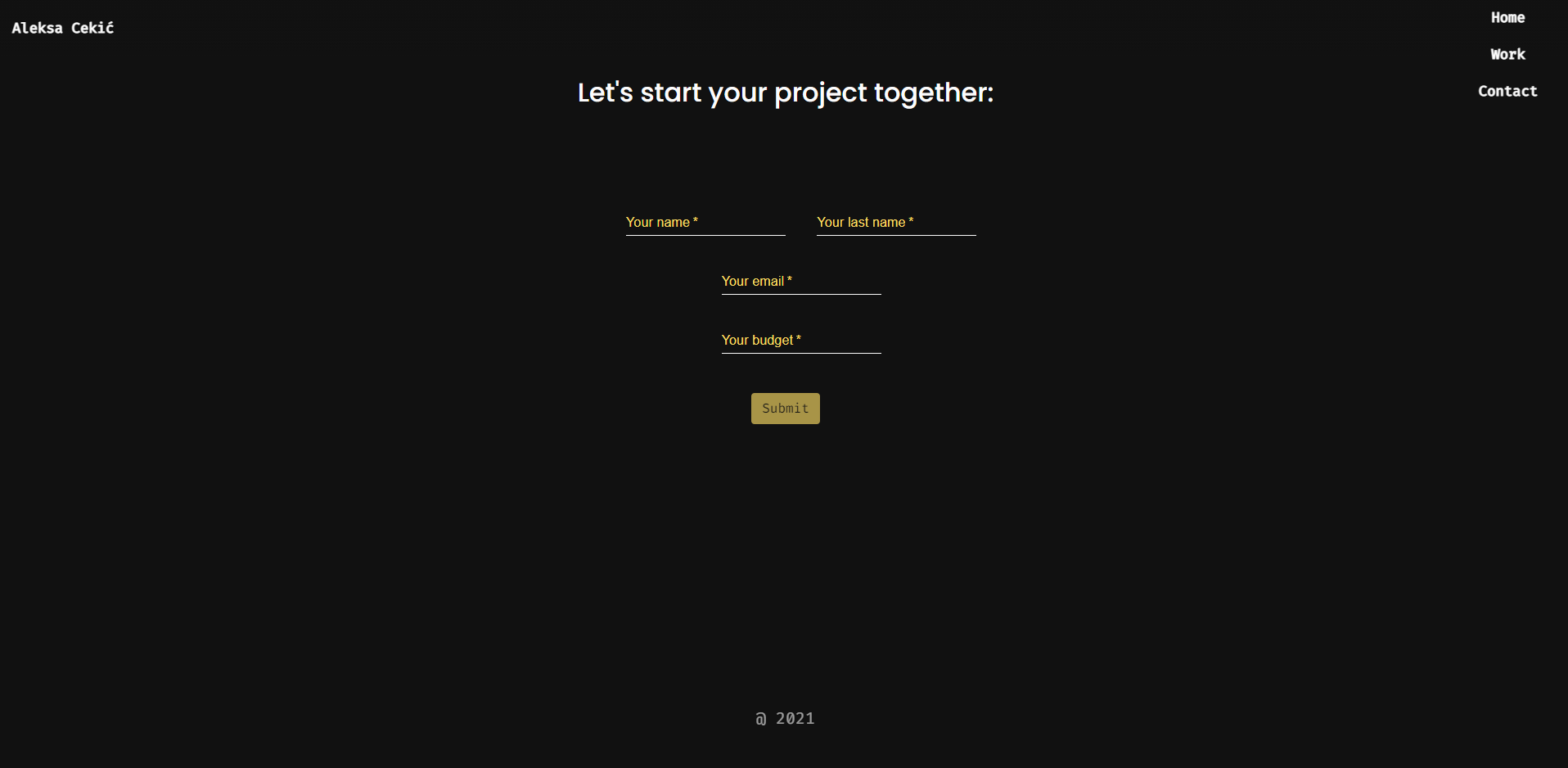




Strana **Work** sadrži sve poslove na kojima je programer radio.

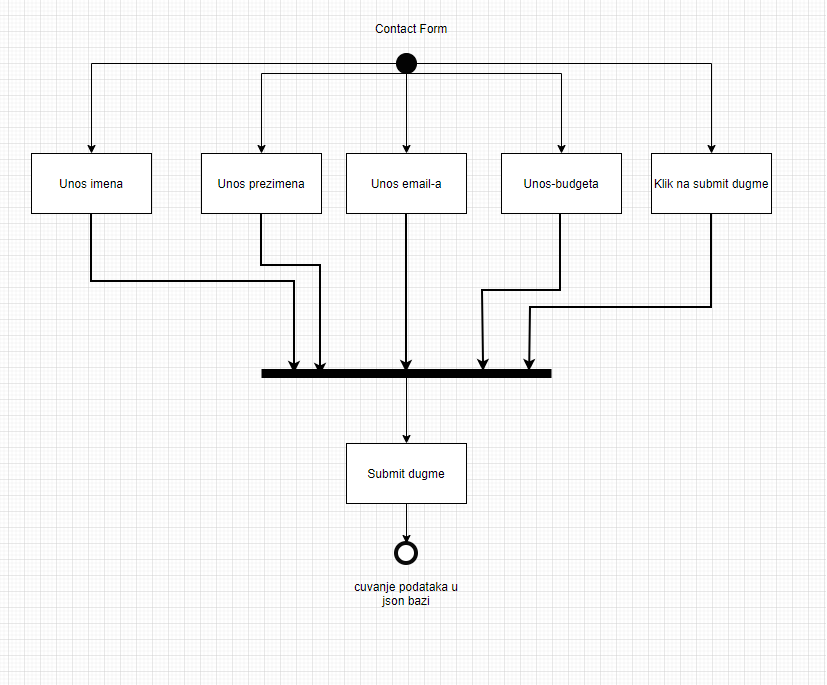


Strana **Contact** sadrži formu za unos imena, prezimena email adrese i budžeta koji se čuva u json-server bazi.



Dugme **submit** je trenutno isključeno zato što korisnik mora popuniti sva polja da bi se aktiviralo.

# Activity diagram za contact:



# Uvod u react:

React je biblioteka za JavaScript kreirana od strane **Facebook**-a.

React je **korisnički interfejs** kao i alat za kreiranje **UI komponenata**.

On je nastao u 2013 godini kao front-end biblioteka. Pre toga smo imali samo klasičan HTML, CSS i JAVASCRIPT. Postojao je jQuery ali to je bilo to.

Kako su web sajtovi postale sve veće i veće aplikacije kao recimo sada **Facebook** gde ljudi mogu imati interakcije a ne samo da pretražuju kao neke obične blogove, zbog toga su javascript fajlovi postajali sve veći i veći pa je samim tim i izašla jedna interesantna biblioteka pod nazivom **backbone.js** koja nam omogućuje da organizujemo naše fajlove lakše. Samim tim je postajalo lakše da radimo sa **DOM**-om (Document Object Model).

To je stvorilo želju za korišćenjem, takozvane, “single-page application” (Aplikacija na jednoj stranici).

U 2010 nastao je famework **AngularJS** kreiran od strane **Google** koji je koristijo baš taj metod za kreiranje aplikacija na jednoj strain.

Sa njegovim nastankom je I revolucionisao razvoj ovakvih web aplikacija. On je koristio jednu specifičnu metodu za organizaciju fajlova, koji su se zvali “kontejneri“.

Međutim sa njim je postojao i problem.

Kako su aplikacije postajale sve veće i veće, samim tim je i potreba za funkcionalnošću postajala veća i kompleksnija.



Facebook je našao rešenje, da ne dolazi do ovakvih kompleksnosti. U 2013 je pustio biblioteku pod nazivom „**React**” u javnost.

React je samim tim postao veoma popularan zato što je koristio drugačiji pristup razvijanju web aplikacija.

U 2014 godini Google je zbog kompleksnosti zahteva web aplikacija, napisao od početka ceo framework koji je sada poznat kao **Angular** (bez JS na kraju) koji je trenutno drugi najpopularniji web framework i trenutno ima 12 verzija.

React je trenutno najpopularnija biblioteka na svetu i trenutno najtraženija na tržištu. Sajtovi kao što su Tumblr, Walmart, Facebook, Uber itd. koriste react.

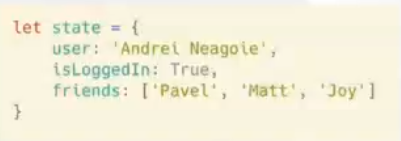
React kreira „**virtualni DOM**“ od pravog DOM-a ali je to samo javascript verzija pravog DOM-a.

Problem kod ostalih biblioteka je to što su oni manipulisali sa podacima na pravom DOM-u gde je dolazilo do velikih bagova I smanjenja brzine pretraživača.

Mi manipulišemo virtualnim DOM-om gde pravi DOM proverava ako je došlo do nekih izmena nakon čega se I on sam ažurira.

React sadrži takozvan “state“ objekat je običan javascript objekat koji sadži određene podatke koje opisuju našu aplikaciju.

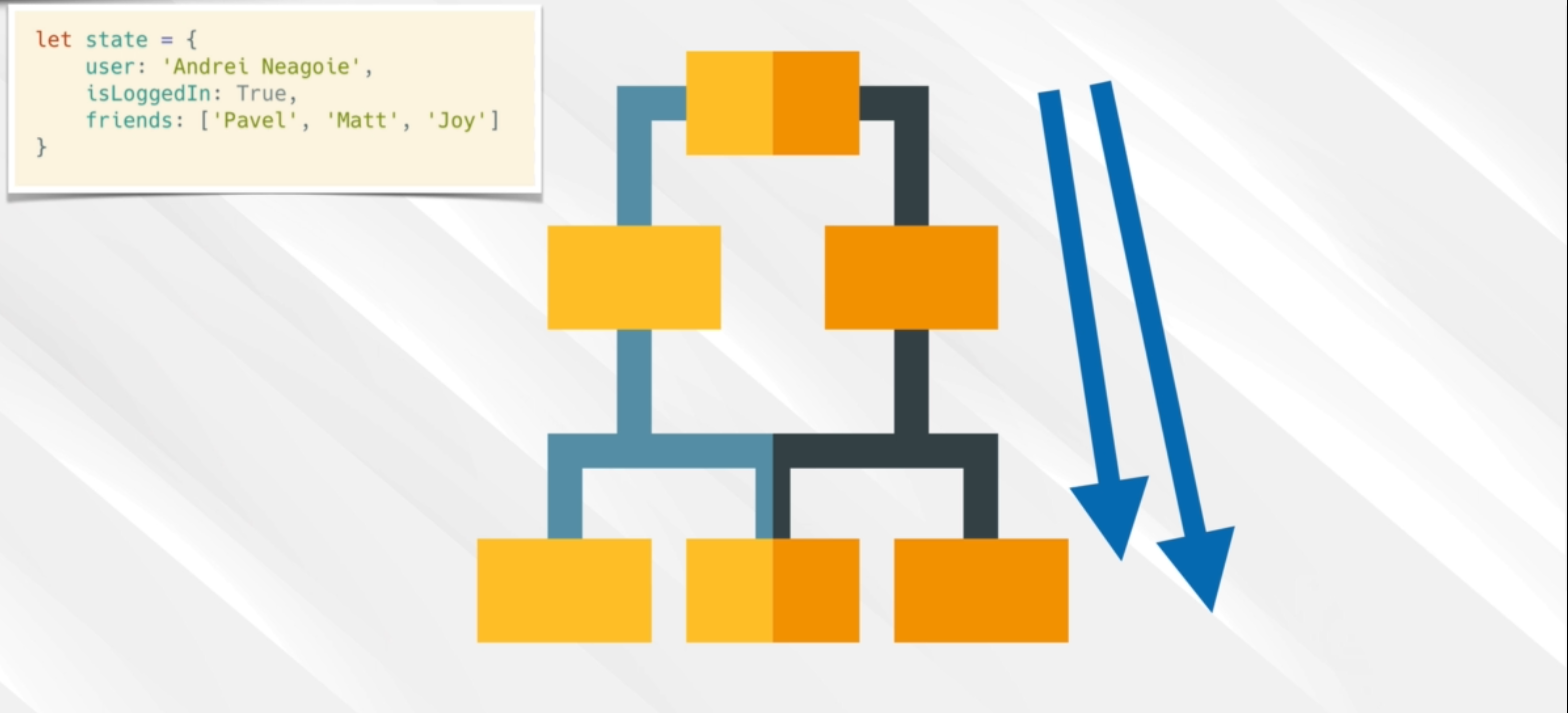
Naziv **React** zapravo i potiče odatle, ako dođe do nekih promena u state-u, samim tim dom reaguje na to.



Svaka komponenta može da sadrži svoj state koji se može spuštati niz komponenti.

Komponente su napravljene korišćenjem takozvane JSX sintakse, što je u suštini HTML sintaksa ugnježdena unutar JavaScript.

Podaci nikad ne mogu da se vraćaju unazad, tako da ako bi se state promenio u nekoj drugoj komponenti u kojoj se spušta, ta promena podataka živi u toj komponenti ali se može i spustiti dalje.



# Opis seleniuma:

Selenium je paket alata za automatizaciju web pretraživača.

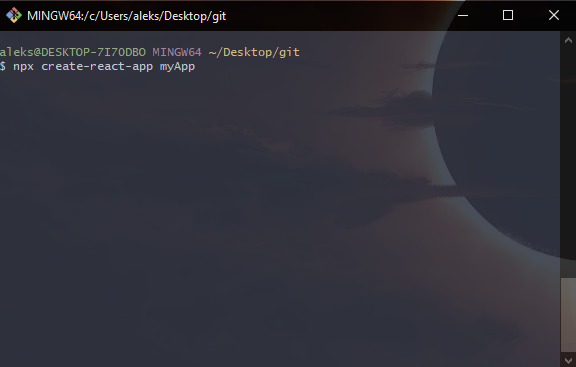
Selenium je nastao u 2004 godini u Čikagu kao alat za testiranje JavaScript koda.

U ovom projektu ja sam iskoristio **selenium-webdriver** zajedno sa **mocha-js** što predstavlja javascript test framework koji radi na **Node.js** i na pretraživaču.

# Kreiranje projekta:

Da bi kreirali projekat prvo moramo instalirati najnoviju verziju Node.js a samim tim i npm koji dolazi sa njim.

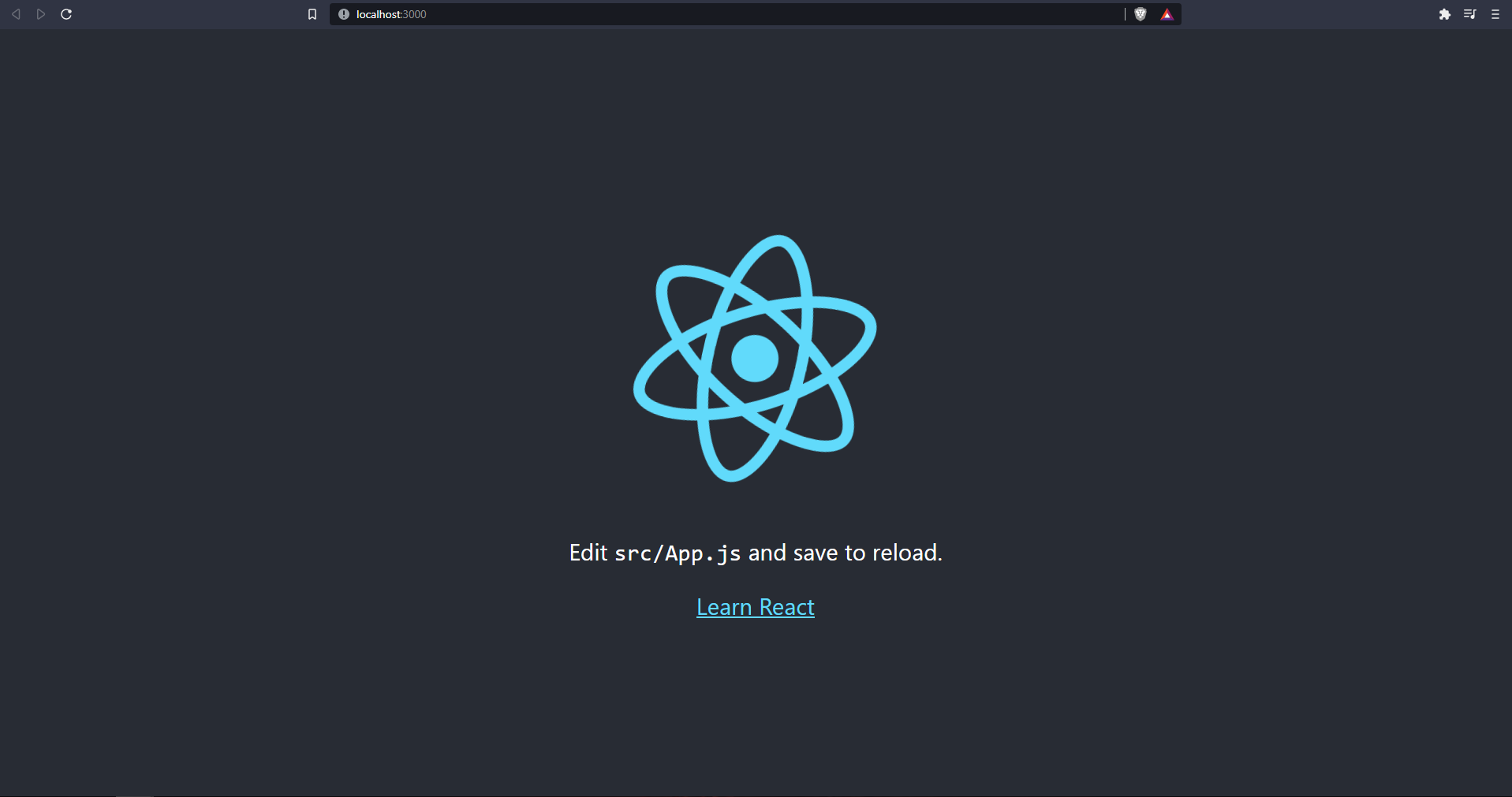
Nakon toga u konzoli ili terminalu kucamo **npx** **create-react-app** **imeProjekta** što nam kreira prazan react projekat.



U ovom slučaju koristimo CLI (Command Line Interface) Yarn tako da kucanjem komande **yarn start**, pokreće se aplikacija na localhost sa portom 3000.

Međutim pre svega treba instalirati paket node\_modules sa svim neophodnim bibliotekama, to se može izvesti kucanjem samo yarn komande ili kucanjem komande yarn install.

Kada se aplikacija pokrene dobija se glavna strana koja izgleda ovako:



# Dodavanje biblioteka u projektu

Sada kada imamo instaliranu I spremnu za rad aplikaciju, moramo dodati sve neophodne biblioteke.

U ovom projektu ja sam koristio biblioteke kao što su:

1. Bootstrap
2. Material-ui
3. Uuid
4. Node-sass
5. AOS
6. Axios
7. Selenium-webdriver
8. Mocha
9. Chromedriver
10. Geckodriver
11. React-router-dom

Sve ove biblioteke mogu se dodati komandom **yarn add nazivBiblioteke**.

Primer:

**yarn add selenium-webdriver**

# Dodavanje animacija

Aplikacija sadrži animacije pri skrolovanju koje se mogu dodati instaliranjem biblioteke AOS (animate on scroll).

Instaliranje biblioteke izvršava se:

**yarn add aos@next**

Nakon toga se mora dodati u projektu sledećim načinom i da se pokrene metodom **init().**

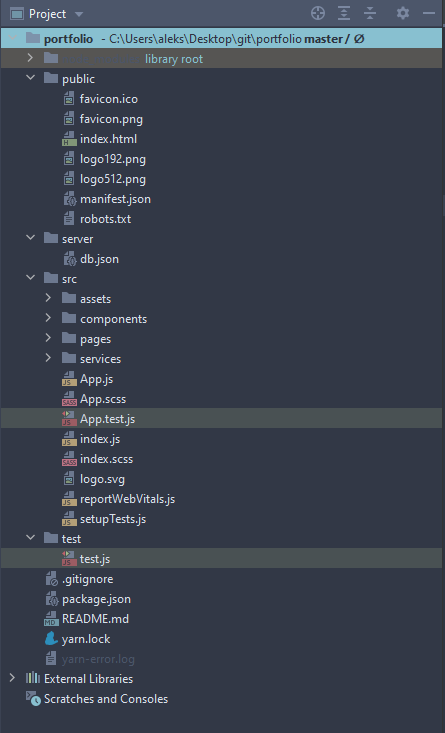
import AOS from 'aos';

import 'aos/dist/aos.css';

// ..

AOS.init();

# Struktura projekta



Struktura projekta je izvučena po raznovrsnim direktorijuma koji su organizovani tako da opisuju šta rade.

Komponente mogu se pozivati svuda gde je potrebno dok strane predstavljaju strane koje se prikazuju na aplikaciji.

Servisi predstavljaju servise napisanim tako da može da se izvrši CRUD nad json bazom.

Server sadrži bazu.

Assets sadrzi zajednicke komponente koje mogu da se pozivaju kroz aplikaciju kao sto su neki zajednicki stilovi, ili slike.

Tests sadrze selenium I mocha testove napisane u javascriptu.

# Testovi

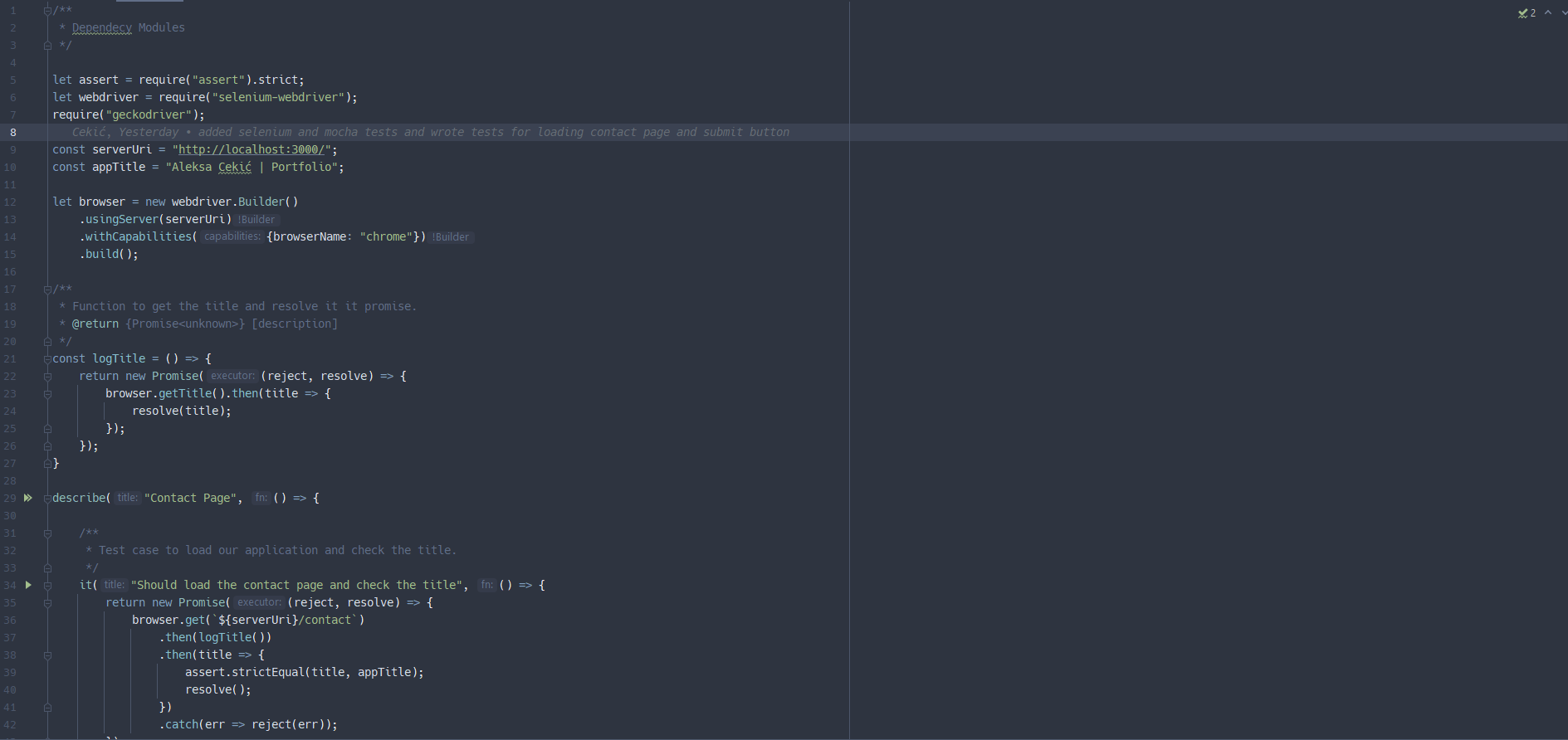
Da bismo izvršili testove mi moramo da instaliramo neke biblioteke.

1. Selenium WebDriver
2. Chromedriver & Geckodriver
3. Mocha
4. Chrome browser

Instaliranje se izvršava kroz CLI:

**yarn add selenium-webdriver chromedriver geckodriver mocha**

Nakon toga možemo pozivati razne module ovih biblioteka.





Prvo je kreiran assert I webdriver. Ovo su bitne biblioteka da bi automatizovale naš razvoj.

**Assert** je defaultna Node.js biblioteka tako da nema potrebe da se instalira.

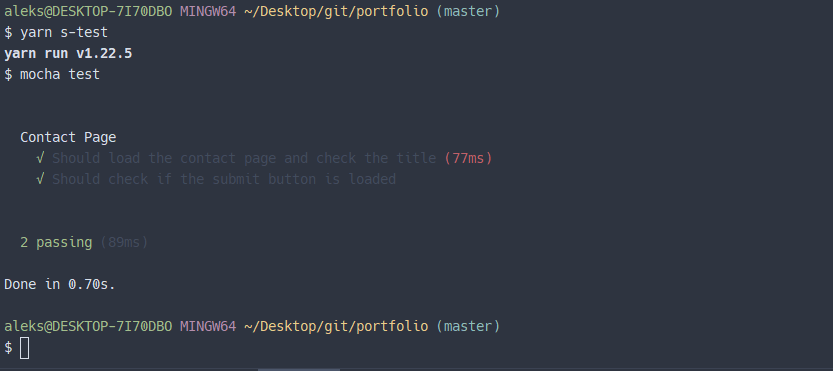
**Slenium-webdriver** je bazni drajver za selenium da bi mogo da se pokrene i **geckodriver** je bitan da bismo mogli da pokrenemo naše test slučajeve na drugim pretraživačima a da nije chrome.

Postavljamo naš aplikacioni server sa URI-jem kao i sa titlom.

Izvršavamo 2 testa:

1. Mora da očita Contact Page i da proveri da li je appTitle isti kao title stranice, to proverava funkcija logTitle.
2. Proverava da li je učitan submit dugme na contact stranici.

Pokretanjem skripte **yarn s-test** treba da pokrene mocha test I da prodje ova 2 testa.



# Zaključak

Web aplikacija, portfolio, koja može da se proširuje I da se nadograđuje.

Napisana je u React-u I koristi razne biblioteke I module za prikaz podataka I stilizovanje.

Sadrži 3 strane za prikazivanje, **Home**, **Work** i **Contact** strane.

Home strana sadrži opise, informacije i poslove.

Work strana sadrži trenutne poslove.

Contact strana ima contact formu koja se čuva u json bazi sa jedinstvenim id-em koji generiše slučajan 128-bit kod koji je unikatan.

Aplikacija se nalazi na Github-u: <https://github.com/yxngalex/se211-portfolio>

# Korišćena literatura: (opciono)

1. <http://lams.metropolitan.ac.rs/>
2. <https://www.udemy.com/course/complete-react-developer-zero-to-mastery/>
3. <https://www.selenium.dev/>