

Express i Node.JS

asinhrona chat aplikacija

Predmet: Klijent Server Sistemi

Profesor:

Dr Mirko Kosanović

Miloš Kosanović

Student:

Goran Đukić

REr 38/17

07.01.2020.

1. Uvod	3
2. Instalacija i podešavanje projekta	3
2.1. Instaliranje modula	3
5. Literatura	4

1. Uvod

U ovom projektu obrađena je izrada veb aplikacije za saradnju između članova tima preko interneta. Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS sa Bootstrap framework-om i JavaScript sa jQuery bibliotekom, dok su na serverskoj strani korišćeni NodeJS sa Express framework-om. Za komunikaciju u realnom vremenu (eng. Realtime Communication) korišćena je JavaScript biblioteka Socket.IO koja je slična WebSocket protokolu. Za autentikaciju korisnika korišćen je Passport, a za komunikaciju sa MongoDB bazom podataka biblioteka Mongoose. Alati koji su korišćeni prilikom izrade aplikacije su Google Chrome i Safari pretraživači uz Visual Studio Code kao text editor i Robo 3T kao navigator baze.

Aplikacija se sastoji iz dva dela. Prvi deo sadrži formu za login i registrovanje, kao i dugmeta za ulaz u aplikaciju pomoću Facebook ili Google naloga. Drugi deo aplikacije, odnosno glavni deo, sadrži dugme za logout, podešavanja gde biramo do kog datuma želimo da nam aplikacija učitava poruke, uz opciono brisanje poruka starijih od tog datuma, i prostor za dopisivanje, koji prilikom login-a automatski učitava poslednje dve nedelje poruka iz baze. Kad god se novi korisnik konektuje na aplikaciju, ostalim korisnicima će se prikazati poruka da je novi korisnik konektovan. Dok neki korisnik kuca poruku, ostali korisnici su obavešteni. U bilo kom trenutku je moguće upload-ovati i podeliti neki fajl sa drugim korisnicima putem običnog drag-and-dropa.

2. Instalacija i podešavanje projekta

Da bismo pokrenuli Node.JS aplikaciju, potrebno je da instaliramo Node.JS, a za ovaj projekat koristimo i MongoDB, tako da je potrebno i njega instalirati.

2.1. Instaliranje modula

U ovom projektu nalazi se **package.json** fajl, to je fajl koji se inicijalizuje sa projektom, naravno na zahtev programera, i ukoliko popunimo sve informacije ispravno biće kreiran fajl. Zatim kad krenemo da instaliramo nove module i ukoliko upotrebimo neke ključne reči (-S , -save) prilikom instalacije u ovom fajlu biće nam upisani svi moduli koje koristimo za našu aplikaciju. U koliko je sve to ispravno kreirano, da bi smo na nekom drugom računaru pokrenuli i instalirali module, potrebno je ukucati sledeću naredbu **npm install** ova naredba

prvo pretražuje **package.json** fajl i u njemu traži i instalira sve dependence (tj. Module ili biblioteke) koji su potrebni za ovaj projekat.

5. Literatura

How to Build a Real-time Chat App With NodeJS, Socket.IO, and MongoDB – Eze Sunday

- <https://dev.to/rexeze/how-to-build-a-real-time-chat-app-with-nodejs-socketio-and-mongodb-2kho>

Starting with Authentication (A tutorial with Node.js and MongoDB) – Daniel Deutsch

- <https://medium.com/createdd-notes/starting-with-authentication-a-tutorial-with-node-js-and-mongodb-25d524ca0359>

Add Facebook Login to your Node.js App with Passport.js – Felistas Ngumi

- <https://www.twilio.com/blog/facebook-oauth-login-node-js-app-passport-js>

How to add Passport.js Google OAuth Strategy to your website – Jon Preece

- <https://developerhandbook.com/passport.js/how-to-add-passportjs-google-oauth-strategy/>

Passport dokumentacija

- <http://www.passportjs.org/docs>

Bootstrap Datepicker in Modal Popup Window - Fengyuan Chen

- <https://www.codehim.com/date-time/bootstrap-datepicker-in-modal-popup-window/>

How To Make A Drag-and-Drop File Uploader With Vanilla JavaScript

- Joseph Zimmerman

- <https://www.smashingmagazine.com/2018/01/drag-drop-file-uploader-vanilla-js/>

Node.js Upload Files - W3Schools

- https://www.w3schools.com/nodejs/nodejs_uploadfiles.asp