****

**Sportski Klub**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Aleksandar Čolović**

**Miloš Kosanović Rer 68/17**

**07.02.2020.**

SADRŽAJ

[1. Uvod - 3 -](#_Toc512514140)

[2. Instalacija i podešavanje projekta - 3 -](#_Toc512514141)

[2.1. Instaliranje modula - 3 -](#_Toc512514142)

[3. Arhitektura aplikacije - 4 -](#_Toc512514143)

[3.1 Serverski deo - 4 -](#_Toc512514144)

[3.2 Klijentski deo - 4 -](#_Toc512514145)

[3.3 Baza podataka - 4 -](#_Toc512514146)

[3.4 Komunikacija - 4 -](#_Toc512514147)

[4. Rad aplikacije - 5 -](#_Toc512514148)

[4.1 Opis implementacije - 5 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 5 -](#_Toc512514150)

[5. Literatura - 5 -](#_Toc512514151)

# 1. Uvod

U ovom projektu obrađena je izrada MEAN Stack web aplikacije taekwondo kluba. Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS sa Bootstrap framework-om, Animate.css i JavaScript sa Angular 8 framework-om, dok su na serverskoj strani korišćeni NodeJS sa Express framework-om. Za bazu je korišćen MongoDB Atlas. Alati koji su korišćeni prilikom izrade aplikacije su Brave pretraživač ,Visual Studio Code text editor, Intellij IDEA IDE i Postman za proveru i testiranje Rest API-ja .

Aplikacija se sastoji iz tri dela(posetilačkog, korisničkog i admin dela) odnosno ima dve role (Član kluba i Admin). Korisnik (odnosno član kluba), kada pristupi veb aplikaciji može videti početnu stranicu sa novostima vezanih za klub, stranicu „O nama“ sa dodatnim informacijama o klubu i stranicu „Uplata“ gde, ako je ulogovan, može pristupiti tabeli sa informacijama o tome, koji član nije platio članarinu, kao i dugme za plaćanje članarine. Članarinu je moguće online platiti pomoću paypala, a informacije o članu koji je platio članarinu, određeni mesec za koji je platio i napomenu koju može opcionalno popuniti se šalju na mail adminu. Admin ima svoju posebnu stranicu gde može dodati novog člana, dodati novi članak (novosti) i izmeniti tabelu sa evidencijom zaduženih članova.

# 2. Instalacija i podešavanje projekta

Da bismo pokrenuli ovu MEAN aplikaciju potrebno je da instaliramo NodeJs, Angular i sve ostale module koji su korišćeni za izradu ove aplikacije. Jako je bitno da nam je dostupan internet prilikom pokretanja programa, jer aplikacija koristi MongoDB Atlas (nerelacionu cloud bazu) kao i Bootstrap i CSS.Active CDN linkove.

## 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, potreno je ukucati sledeću naredbu **npminstall** 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. Moduli koji su potrebni za pokretanje ove aplikacije su: bcrypt, body-parser, cors, jsonwebtoken, mongoose, multer, nodemailer, passport, rxjs.

# 3. Arhitektura aplikacije

Aplikacija u sebi sadrži korenski (engl. Root) direktorijum '/' koji u sebi sadrži '/node\_module' direktorijum koji sadrži module koji su potrebni NodeJS-u, '/src' direktorijum u kome je smešten klijentski deo aplikacije, /backend folder u kome je smešten serverski deo aplikacije poput kontrolera, modela, serverskih ruta, itd., dve datoteke '/package.json' koji predstavlja JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo aplikacije zavistan, četiri config.json datoteke koje predstavljaju konfiguraciju serverskog i klijentskog dela aplikacije, server.js datoteka u kojoj je kod NodeJS servera. Unutar /src direktorijuma se nalazi /app direktorijum koji sadrži komponente aplikacije (Components), kojima se većini može pristupiti preko URL adrese, module ts datoteke gde se implementuju sve komponente koje smo kreirali i routing modul gde definišemo frontend rute, /assests direktiva gde se nalaze alternativni css kao i slike koje su korišćene za realizaciju ove web aplikacije, /enviroments direktiva gde se nalaze dodatne typescript datoteke, glavna index.html stranica i glavna css datoteka.

## 3.1 Serverski deo

## Na serverskom delu se nalazi config, models, controllers, routes direktorijumi kao i app.js datoteka. U config folderu se nalazi db.js gde se nalazi konekcioni string sa bazom kao i json web token tajni ključ koji nam služi za verifikaciju tokena određenog korisnika, passportConfig gde se nalazi konfiguracija passporta. Passport je middleware koji služi za autentifikaciju korisnika. U controllers direktivi se nalaze kontroleri koji sluze da prime zahtev od klijenta i zatim šalju podatke bazi i modelu. Model u sebi sadrži korisničku šemu koja je sastavljena od imena svojstva i tipa vrednost. On takođe služi da prosleđuje podatke bazi. U direktorijumu routes se nalaze dve javascript datoteke u kome su smeštene serverske API rute. App.js datoteka sadrži konekcije sa modelima, kontrolerima, modulima... U njoj se nalazi i konfiguracija nodemailer modula, koji služi za slanje podataka na e-mail adresu.

## 3.2 Klijentski deo

Angular je framework za kreiranje SPA (Single Page Application) jer u sebi sadrži samo jednu html stranicu i komponente od koje se sastoji ta aplikacija. Komponente su specifične po tome da, kada se prelazi sa jedne na drugu, html stranica se ne osvežava već se ta komponenta odmah učita. To daje brzinu i efikasnost našoj aplikaciji. Komponenta se može tretirati kao posebna stranica, ali se može koristiti i kao deo određene stranice. Ako želimo da komponentu koristimo kao deo stranice, koristimo tag <app-imekomponente>, a ako želimo da je koristimo kao zasebnu stranicu, moramo definisati rutu. U app-routing ts datoteci možemo da definišemo rute naših komponenti kojima kasnije možemo pristupiti preko URL adrese.

Za dodatnu stilizaciju stranice kao i mogućstva za lakšim kreiranjem responsive stranice koristimo Bootstrap framework. On sadrži dosta predefinisanih funkcija za stilizovanje tako da dosta umanjuje rad sa našim css dokumentom i takođe omogućava da aplikacija izgleda savršeno na svim veličinama ekrana.

## 3.3 Baza podataka

MongoDB je nerelaciona baza podataka i u njoj se čuvaju podaci koje aplikacija šalje. Ona može biti lokalna ili na cloudu. Moja baza sadrži tri kolekcije. U kolekciji post se nalaze naslovi i sadržaj svakog članka koji je admin kreirao. U kolekciji noviKorisnik se nalazi permisija da li je korisnik admin ili član, username korisnika, hešovana šifra kao i saltSecret tajni ključ. U kolekciji placanje se nalazi ime člana kao i meseci u godini sa vrednostima koje unese admin.

## 3.4 Komunikacija

## 

## 

# 4. Rad aplikacije

## 4.1 Opis implementacije

## 

## 

## Ovo je deo koda za prikazivanje iste forme ali sa različitom funkcijom, u zavisnosti od toga koje je dugme kliknuto. Na početku, default vrednosti isDisabled i isEnabled su true. To su dodeljene vrednosti dugmadi i divovima sa naslovom u formi sa property hidden.(Znači da su na početku sakriveni). U tabeli na istoj stranici se nalaze dva dugmeta: ikonica za izmenu tabele i Dodaj Člana. Njihovom disabled property-ju su dodeljene promenljive isDisabledd i isEnabledd sa početnom vrednošću false(znači da su oba dugmeta na početku aktivna). Formi je na property-ju hidden dodeljena promenljiva isShown koja na početku ima vrednost true. Kada korisnik klikne dugme Dodaj Člana, ikonici za izmenu tabele, disabled property dobija vrednost true (ne može se kliknuti), isShown dobija vrednost false (forma se prikazuje), i sakriva se dugme i div sa naslovom u formi za izmenu, a prikazuje se dugme i div sa naslovom u formi za dodavanje. Kada se dugme u formi klikne, sve promenljive dobijaju početnu vrednost. Proces je suprotan u slučaju kad se klikne ikonica za izmenu tabele.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

## Da bi se pokrenula aplikacija, pored instaliranih modula, NodeJS, Angulara i interneta, potrebno je otvoriti dva terminala kako bi pokrenuli klijentski i serverski deo aplikacije. Pokretanje nodejs-a možemo izvršiti komandom npm run devStart(pokreće se nodemon). Pokretanje Angulara možemo izvršiti komandom ng serve. Kada se oba servera podignu, možemo pristupiti aplikaciji.

## 

## Za informacije o klubu, potrebno je kliknuti link “O Nama” koji se nalazi na navigacionom meniju. Uplatu mogu izvršiti samo članovi kluba, što implicira da morate biti ulogovani kako bi ste mogli pristupiti toj stranici.

## 

## 

## Menjanje tabele plaćanja, dodavanje novog korisnika ili dodavanje novog članka može se postići dugmetom klikni.

## 

## 

# 5. Literatura

1. <https://angular.io/docs>

2. <https://stackoverflow.com/>

3. <https://www.youtube.com/>

4. <https://github.com/>

5. <https://www.udemy.com/course/angular-2-and-nodejs-the-practical-guide/#instructor-1>

6. Dayley B.. (2018). Node.js, MongoDB i Angular Integrisane alatke za razvoj veb strana