****

**PDFKit**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Sazdić Jovana**

**Miloš Kosanović Rer 37/16**

**25.05.2019.**

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 - 5 -](#_Toc512514144)

[3.2 Klijentski deo - 5 -](#_Toc512514145)

[3.4 Komunikacija - 6 -](#_Toc512514147)

[4. Rad aplikacije - 7 -](#_Toc512514148)

[4.1 Opis implementacije - 7 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 8 -](#_Toc512514150)

[5. Literatura - 9 -](#_Toc512514151)

# 1. Uvod

U ovom projektu obrađena je izrada veb aplikacije za računanje indeksa telesne mase (BMI kalkulator). Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS i JavaScript. Takođe, korišćena je i JavaScript PDFkit biblioteka za generisanje PDF-a za Node i pretraživač. Tehnologije koje su korišćene na serverskoj strani su NodeJS sa Express framework-om.

Alati koji su korišćeni prilikom izrade aplikacije su Google Chrome pretraživač i Visual Code kao text editor.

PDFKit je biblioteka za generisanje PDF dokumenata za Node i pretraživače koji olakšava kreiranje složenih, višestranih dokumenata za štampanje. API obuhvata lanac i uključuje i funkcije niskog nivoa kao i apstrakcije za funkcionalnost višeg nivoa. PDFKit API je dizajniran da bude jednostavan, tako da je generisanje složenih dokumenata često isto tako jednostavno kao nekoliko poziva funkcija.

**2. Instalacija i podešavanje projekta**

Da bismo pokrenuli nodejs aplikaciju potrebno je da instaliramo NodeJs, koji možemo preuzeti

sa sledeće stranice: <https://nodejs.org/en/>.

## 2.1. Instaliranje modula

Velika prednost Node.js-a je modularnost. Modularnost podrazumeva nadogradnju Node.js, inkorporaciju odredjenih funkcionalnosti u vidu programskog koda koji proširuju mogućnosti Node.js. Module možemo posmatrati kao neku vrstu JavaScript biblioteka, a zapravo predstavljaju set funkcija koje želimo uključiti u aplikacju.

Postoje ugrađeni moduli, takozvani Built-in, a takođe postoje i moduli- packages, koje instaliramo.

Module instliramo pomoću Node Package Managera ili skraćeno – **npm.**NPM je pisan u JavaScriptu i on na osnovu zadatih parametara pretražuje bazu modula-registar modula. Npm automatski dodaje module u **dependency** listu (popis modula), fajla **package.json.**

**package.json** je tekstualni fajl u JSON formatu i on je obavezan za svaki modul, uključujući i koren aplikacije koja je i sama jedan veliki modul.. Obavezna svojstva koje mora da sadrži su:

• ime (eng. name) - jedinstveni naziv modula pomoću kog moze da se registruje,

• verzija (eng. version) - verzija modula.

Ostali atributi su:

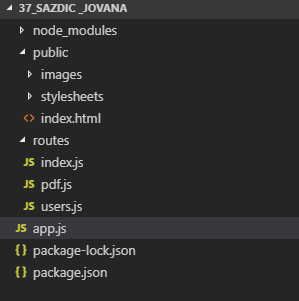
• autor (eng. author ) - svojstvo čuva naziv autora modula

• opis (eng. description) - svojstvo čuva opis modula koji je koristan ukoliko želimo da modul objavimo

• zavisnosti (eng. dependencies) - ovo svojstvo čuva listu svih instaliranih modula i njihovih podmodula koje koriste. Lista sadrži ime modula i njegovu verziju, kao i ostale module od kojih on zavisi.

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 **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.

# 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. '/public' direktorijum u kome je smešten klijentski deo aplikacije, ’/routes’ direktorijum koji sadrzi datoteke rutera (index.js , pdf.js i users.js), kao i dve datoteke '/package.json' koji predstavlja JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo aplikacije zavistan i app.js datoteka u kojoj je kod NodeJS servera. Svakoj datoteci iz '/public' direktorijuma je moguće pristupiti javno, odnosno direktno iz URL-a veb čitača.

## 3.1 Serverski deo

**Node.js** je programski jezik zasnovan na JavaScript jeziku. On je ne-blokirajući, event driven, lightweight, efikasan jezik čija je glavna namena da se koristi kod distribuiranih aplikacija koje rade na različitim platformama i koje imaju potrebu da rade sa velikim količinama zahteva ili podataka u realnom vremenu.

Node.js može generisati dinamičan sadržaj stranice, može da kreira,otvara,čita,piše,briše i zatvara datoteke na serveru. Takodje, može da dodaje,briše i menja podatke u bazi podataka.

**Node.js** ima veliki broj modula – **JavaScript** biblioteka ili funkcija koje obavljaju određene specifične zadatke i koje možete dodati svojoj aplikaciji.

**Express.js** frejmvork predstavlja najpoznatiji frejmvork za razvoj veb aplikacija zasnovanih na node.js programskom jeziku. U sebi sadrži podršku za rutiranje, konfiguraciju, templejt engine, POST parsiranje zahteva, i pristup različitim bazama podataka i druge funkcionalnosti. Iako predstavlja solidno rešenje, za sada se još uvek po funkcionalnostima ne može uporediti sa drugim fullstack frejmvorcima kao što su Rails, Larawell, Django i drugi…

## 3.2 Klijentski deo

## HTML je skraćenica od Hyper Text Markup Language, što se na srpski može prevesti kao Jezik za označavanje hiper tekstova. Hiper tekstovi su tekstovi koji pored reči sadrže i sliku, video i audio zapise. HTML je podskup jednog šireg jezika, SGML-a (Standard General Markup Language) i koristi se za definisanje izgleda  World Wide Web dokumenata (stranica) kao i za uspostavljanje veza (linkova) među dokumentima (podrazumeva se da dokument sadrži tekst, sliku, zvuk, grafiku).

## CSS (engl. Cascading Style Sheets) je jezik formatiranja pomoću kog se definiše izgled elemenata veb-stranice. Eksterni stilovi se čuvaju kao CSS fajlovi .CSS se koristi za definisanje stila vašeg veb sajta, uključujući dizajn za različite uređaje i veličine ekrana.

## JavaScript je skript jezik koji je obično deo HTML stranice i koji se izvršava od strane brauzera ( web klijenta ). JavaScript može da čita, piše i menja HTML elemente ( DOM objekte ), da validira ulazne podatke, da detektuje vrstu brauzera, da kreira kolačiće ( cookies ).

## 3.3 Komunikacija

## Express podržava sledeće metode rutiranja: get, post, put, head, delete, options. Prva stvar koju ćete videti u app.js su zahtevi. Funkcija zahteva je prilično jasna. To je ugrađena funkcija čvora koja uvozi objekat (module.exports) iz druge datoteke ili modula.

## 

## Module uključujemo pomoću komande require. U require() module spadaju i lokalni require() moduli koji upravljaju rutiranjem.

## 

## Kreiramo Express aplikaciju.

## 

## Postavljanje putanje pomocu app.use().

## 

## Na kraju imamo naredbu app.listen() koja govori aplikaciji koji broj porta treba da ’sluša’.

## 

# 4. Rad aplikacije

## 4.1 Opis implementacije

## 1.Kreiramo ime PDF dokumenta u kome će se sačuvati naš rezultat po izvršenju aplikacije. Biće izgenerisan pdf fajl „dokument.pdf“.

## 

## 2.

## Doc.fontSize() - podešavanje veličine fonta.

## Doc.text()- prikazivanje teksta u PDF dokumentu.

## Doc.image()- dodavanje slike u PDF dokument i podešavanje veličine slike.

## Doc.end()- Završetak PDF dokumenta.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

## Za pokretanje aplikacije, pre svega potrebno je instalirati NodeJS. Instalaciju možemo skinuti sa sajta : <https://nodejs.org/en/> . Nakon toga, da bi se aplikacija izvršila potrebno je instalirati module koji su definisani u package.json fajlu.

## 

## To cemo uraditi tako što cemo u command prompt-u otkucati naredbu npm install.

## 

## Nakon toga, aplikaciju pokrećemo naredbom node app.js.

## 

## Ukoliko je sve uspešno instalirano u komandnoj liniji dobijamo poruku „Listening on port 3000“.

## I na kraju, u web pretraživaču unosimo adresu : <http://localhost:3000/> .

# 5. Literatura

<http://pdfkit.org/>

<http://pdfkit.org/docs/getting_started.html>

<https://nodejs.org/en/>

<https://developer.ibm.com/tutorials/learn-nodejs-expressjs/>

<http://www.codeblocq.com/2016/05/PDF-Generation-with-Node-JS/>