****

**GraphicsMagick**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Marko Todorović**

**Miloš Kosanović Ser 55/15**

**13.09.2019.**

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 - 4 -](#_Toc512514144)

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

[3.3 Baza podataka - 5 -](#_Toc512514146)

[3.4 Komunikacija - 5 -](#_Toc512514147)

[4. Rad aplikacije - 6 -](#_Toc512514148)

[4.1 Opis implementacije - 6 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 6 -](#_Toc512514150)

[5. Literatura - 7 -](#_Toc512514151)

# 1. Uvod

U ovom projektu obrađena je obrada rezolucije slike preko GraphicMagick-a. Tehnologije koje su korišćene na klijentskoj strani su Jade, CSS sa Bootstrap framework-om, dok je na serverskoj strani korišćen NodeJS sa Express framework-om.

Koriscen je vec pomenuti GraphicMagick za promenu rezolucije, dok je za upload slike koriscen Multer.

# 2. Instalacija i podešavanje projekta

Prvo instaliramo Node.js i pokrećemo Command Prompt u folderu u kome nam se nalazi projekat. A nakon toga komandom **npm init** kreiramo **package.json** fajl. Menadžer paketa je alat zadužen da automatizuje procese vezane za rad sa paketima.

Instaliramo Express application generator koji nam kreira skelet aplikacije, **npm install express-generator –g**, flag –g znači da instaliramo globalno.

U folderu u kome želimo da kreiramo Express projekat, kucamo **express Marko**. Ovo će kreirati folder Marko sa svim potrebnim fajlovima i folderima koji su nam potrebni.

U folderu bin, fajl www predstavlja naš server. Ovde je konfigurisan server da sluša na portu 3000.

Potrebno je komandom **npm install** instalirati sve dependence koje su potrebne ovom projektu. Ovo će kreirati folder node\_modules. Sada možemo pokrenuti server komandom npm start. U web pretraživaču kucamo <http://localhost:3000/>

## 2.1. Instaliranje modula

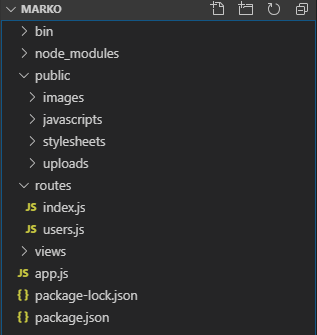
Kad instaliramo nove module u **package.json** fajlu biće nam upisani svi moduli koje koristimo za našu aplikaciju. Da bi smo na drugom računaru pokrenuli i instalirali module, potrebno je ukucati naredbu **npm install** ova naredba prvo pretražuje **package.json** fajl i u njemu traži i instalira sve dependence (tj. module ili biblioteke) koji su potrebni za ovaj projekat.

Za projekat je potrebno instalirati express-validator. Njega instaliramo komandom npm install express-validator –save. Potrebni su nam I body-parser (npm install body-parser) kao I Multer (npm install multer). U fajlu app.js dodajemo funkcionalnost za:

Var bodyParser = require(‘body-parser’)

# 3. Arhitektura aplikacije

Uz pomoć Express application generator-a kreiran folder sa u okviru koga su kreirani potrebni fajlovi i folderi.



Projekat 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 projekta, kao i datoteku '/package.json' koji predstavlja JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo projekta zavistan i www.js datoteka u kojoj je kod NodeJS servera.

## 3.1 Serverski deo

Node.js je programski jezik zasnovan na JavaScript jeziku i on je ne-blokirajući, vođen događajima i koristi se za realizaciju distribuiranih aplikacija koje rade sa velikim količinama podataka u realnom vremenu.

Uz pomoć Node.js možemo da razvijamo programe na raznim nivoima, od prostih programa do HTTP servera, odnosno web aplikacija.

Jedan od težih problema u programiranju aplikacija sa kojima se komunicira preko mreže je kako upravljati ulazom i izlazom, odnosno kako čitati (read) i pisati (write) podatke sa mreže, diskova i drugih uređaja. Prenos podataka kroz mrežu zahteva vreme pa se postavlja pitanje na koji način postići potrebnu brzinu rada takvih aplikacija.

Node je prvobitno bio namenjen da se lako ostavri asinhroni I/O. Mi smo se sa asinhronim interfejsom sreli ranije kada somo koristili XMLHttpRequest ( Ajax call ). Asinhroni interfejs omogućava JavaScript programu da nastavi izvršavanje bez zaustavljanja i da pozove povratnu (callback) funkciju kada interfejs završi rad. Upravo tako Node vrši I/O operacija. Node je nastao 2009 godine, kada je Ajax već uveliko bio korišćen tako da su programeri bili spremni za rad sa asinhronim I/O.

Express.js je najpoznatiji frejmvork za razvoj web aplikacija zasnovanih na Node.js programskom jeziku i u sebi sadrži podršku za rutiranje, konfiguraciju, POST parsiranje zahteva, pristup bazama podataka i mnoge druge funkcionalnosti.

## 3.2 Klijentski deo

Klijentska strana je odrađena preko Jade templejtskog jezika koje je na prvi pogled neobičan zato što ne sadrži HTML tagove, ali umesto toga on koristi minimalistički pristup koj koristi imena tagova, razmal, i CSS-like metodu za referenciranje za definisanje HTML koda. Jedini izuzetak je div tag, koji se toliko često koristi da jade, u slučaju da je tag u templejtu preskočen, ga sam podrazumeva.

## 3.3 Baza podataka

U ovom projektu nije korišćena baza podataka.

## 3.4 Komunikacija

U fajlu index.js koji se nalazi u folderu routes renderujemo index:

router.get('/', function(req, res, next) {

res.render('index', { title: 'GraphicsMagick' });

});

Zatim dodajemo post rutu:

router.post('/', upload, function(req, res, next)

U telu post rute vrsimo prosledjivanje unetih parametara za rezoluciju, I zatim vršimo promenu rezolucije i čuvanje te slike u public/uploads fajlu, kao I proveru da li je doslo do greske:

var width = req.body.resX + '!';

var height = req.body.resY + '!';

gm(req.file.path)

.resize(width, height)

.noProfile()

.write('./public/uploads/' + req.file.fieldname + '-' + Date.now() + '.png', function(err){

if(!err){

res.redirect('/')

# 4. Rad aplikacije

## 4.1 Opis implementacije

Instaliramo NodeJS a nakon toga restartujemo kompjuter. U sledećem koraku pokrećemo Command prompt.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

Nakon instaliranja Node.js instaliramo Express application generator komandom **npm install express-generator –g**. Ovom komandom instaliramo **express-generator** globalno na računaru. Posle toga potrebno je instalirati **express-generator** i u projektu.

Nakon toga, instaliramo sve neophodne module, izvršavamo komandu **npm install**.

Ono što je potrebno je da instaliramo multer preko komande npm install --save multer, express-validator komandom: npm install express-validator –save, graphicsmagick komandom: npm install gm --save), body-parser komandom: npm install body-parser.

Nakon sšo sve završimo otvorimo browser I otkucamo localhost:3000, kako bi testirali projekat.

# 

# 5. Literatura

<https://nodejs.org>

<https://www.npmjs.com/package/express-validator>

http://www.graphicsmagick.org/GraphicsMagick.html