Izrada web aplikacije za fotografski studio Dreams

Žigant, Rea

Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Preddiplomski sveučilišni studij Informatika

Rea Žigant

Izrada web aplikacije za fotografski studio „Dreams“

Završni rad

Mentor: Doc. dr. sc. Lucia Načinović Prskalo

Rijeka, rujan 2022.

FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA
SVEUČILIŠTE U RIJECI

Rijeka, 22.4.2022

Zadatak za završni rad

Pristupnik: Rea Žigant

Naziv završnog rada: Izrada web aplikacije za fotografski studio *Dreams*

Naziv završnog rada na eng. jeziku: : Development of a web application for the photo studio *Dreams*

Sadržaj zadatka: Cilj ovog završnog rada je izrada web aplikacije za fotografski studio. U radu će se detaljno opisati svi alati koji su se koristili tijekom izrade aplikacije, njihove funkcionalnosti te prednosti i nedostaci. Također će biti prikazan izgled web aplikacije i opisane njezine funkcionalnosti.

Mentor

Doc. dr. sc. Lucia Načinović Prskalo

Načinović Prskalo

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Zadatak preuzet: 22.4.2022.

Rea Žigant

(potpis pristupnika)

Sažetak

Tema ovog završnog rada izrada je web aplikacije za fotografski studio „Dreams“. Rad se sastoji od teorijskog i praktičnog rada. U teorijskom dijelu opisani su alati, tehnologije i jezici koji su korišteni prilikom izrade praktičnog dijela rada. Danas postoji mnogo tehnologija za izradu web aplikacija, a prilikom izrade ovog rada koristili su se Vue.js, JavaSript, HTML, CSS, Quasar, Node.je te Google Firebase kao baza podataka. Navedeni alati, tehnologije i jezici jedni su od trenutno najkorištenijih na tržištu za ovo područje.

Ključne riječi: web aplikacija, Vue.js, JavaScript, HTML, CSS, Quasar, Node.je, Google Firebase, Figma

Sadržaj

1. Uvod

Web aplikacija skup je elemenata na web stranici koji obavljaju zadatke putem interneta (TechTarget, 2022). One mogu biti dizajnirane za rješavanje različitih zadataka i mogu imati različitu primjenu. Dizajnirane su za rad na web poslužiteljima i koriste web preglednike kao što su primjerice Microsoft Internet Explorer ili Chrome kao korisničko sučelje. Web aplikacije su obično klijent/poslužitelj aplikacije. Neke od najpoznatijih web aplikaciju su Facebook, Youtube i Instagram. U ovom radu upoznat ćemo se s web aplikacijom *Dreams* koja je izrađena u sklopu završnog rada.

1.1 Struktura završnog rada

U prvom dijelu rada definirane su tehnologije i jezici koji su korišteni prilikom izrade praktičnog dijela rada kao što su Figma, HTML, CSS te JavaScript. Spomenute tehnologije i jezici su zasebno ukratko opisani te su navedena mjesta i načini njihove primjene.

U radu smo se također upoznali sa alatom koji služi kao baza podataka. Baza podataka može se svrstati pod backend dio aplikacije, a predstavlja svaku zbirku podataka ili informacija koja je posebno organizirana za brzo pretraživanje i dohvaćanje pomoću računala. U našem slučaju kao bazu podataka koristimo Google Firebase.

Na kraju rada opisan je postupak izrade web aplikacije *Dreams*. Web aplikacija *Dre*ams služit će u komercijalne svrhe. Na spomenutoj web aplikacije moći ćemo naći sve najbitnije podatke vezane uz fotografski studio *Dreams*, galeriju s fotografijama i kontakt podatke.

2. Opis tehnologija koje su korištene prilikom izrade web aplikacije „Dreams“

U ovom su poglavlju opisani alati, programski jezici i baza podataka koji su se koristili prilikom izrade web aplikacije „Dreams“.

2.1. Alati i programski jezici

2.1.1. Figma

Figma je suradnička web aplikacija za dizajn sučelja s dodatnim izvanmrežnim značajkama koje omogućuju desktop aplikacije za macOS i Windows (Figma, 2022). Mobilna aplikacija Figma za Android i iOS omogućuje pregled i interakciju s Figma prototipovima u stvarnom vremenu na mobilnim i tablet uređajima. Skup značajki usredotočen je na dizajn korisničkog sučelja i korisničkog iskustva s naglaskom na suradnju u stvarnom vremenu koristeći razne alate za uređivanje vektorske grafike i izradu prototipova.

Slika 1 Logo Figma-e

Dakle, Figma je kolaborativni alat za dizajn sučelja koji osvaja svijet dizajna. U potpunosti se temelji na pregledniku i ne radi samo na Mac računalima, već i na osobnim računalima s operativnim sustavima Windows ili Linux. Nudi Web API potpuno besplatno.

Velika prednost Figme je mogućnost dijeljenja iste datoteke u stvarnom vremenu. Kada se koriste tradicionalne izvanmrežne aplikacije kao što su Sketch i Photoshop, ako dizajneri žele podijeliti svoj rad obično ga moraju izvesti u slikovnu datoteku, a zatim poslati e-poštom ili izravnom razmjenom poruka. U Figmi, umjesto izvoza statičnih slika, postoji mogućnost podjele poveznice na Figma datoteku koju klijenti i kolege mogu otvoriti u svojim preglednicima. Tako se doprinosi uštedi vremena i neugodnosti u tijeku rada dizajnera. Na taj način, kolege i klijenti mogu bolje komunicirati, biti upoznati s radom i pregledati najnoviju verziju datoteke.

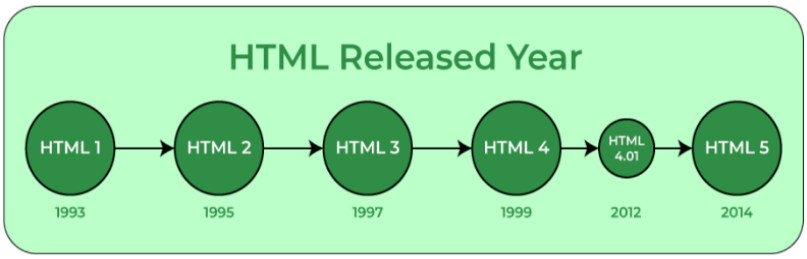
2.1.2. HTML (HyperText Markup Language)

HTML je kratica za HyperText Markup Language. Koristi se za dizajn internetskih stranica pomoću jezika za označavanje. HTML je kombinacija hiperteksta i jezika za označavanje. Hipertekst definira vezu između internetskih stranica. Za definiranje tekstualnog dokumenta unutar oznake koja definira strukturu internetskih stranica koristi se jezik za označavanje. Jezik za označavanje se koristi za označavanje teksta ili pisanje bilješki tako da ga stroj može

razumjeti i manipulirati tekstom u skladu s tim. Većina jezika za označavanje čitljiva je ljudima. Jezik koristi oznake za definiranje koje manipulacije treba učiniti na tekstu.

Slika 2 Logo HTML 5

HTML je jezik za označavanje koji preglednik koristi za rad s tekstom, slikama i drugim sadržajem kako bi ga prikazao u traženom formatu. Stvorio ga je Tim Berners – Lee 1991. godine. Prva verzija HTML – a bila je HTML 1.0, ali prva standardna verzija bila je HTML 2.0 koja je objavljena 1995. godine.



Slika 3 Lenta verzija HTML-a

HTML koristi unaprijed definirane oznake i elemente koji govore pregledniku kako pravilno prikazati sadržaj te je potrebno uključiti završne oznake. Ako se izostave završne oznake, preglednik primjenjuje učinak početne oznake do kraja stranice.

Osnovna struktura HTML internetske stranice sadrži bitne građevne elemente kao što su doctype deklaracija, HTML te elementi glave, naslova i tijela na temelju kojih se izrađuju sve internetske stranice.

Oznaka <!DOCTYPE html> je deklaracija vrste dokumenta. Deklarira dokument kao HTML dokument. Deklaracija doctype ne razlikuje velika i mala slova. Oznaka <html> je korijenski element HTML-a. Svi ostali elementi su sadržani unutar njega. Oznaka <head> sadrži pozadinske elemente internetske stranice. Elementi unutar oznake <head> nisu vidljivi na prednjem dijelu internetske stranice. HTML elementi koji se koriste unutar <head> uključuju oznake <style>, <title>, <base>, <noscript>, <script>, <meta> i <link>. Oznaka <style> omogućuje umetanje stila u internetske stranice i čini ih privlačnima za gledanje uz pomoć CSS-a. Oznaka <title> je ono što se prikazuje na vrhu preglednika kada se posjeti internetska stranica te sadrži naslov internetske stranice koja se gleda. Oznaka <base> određuje osnovni URL za sve relativne URL-ove u dokumentu. Oznaka <noscript> definira dio HTML-a koji se umeće kada je skriptiranje isključeno u korisničkom pregledniku. Nadalje, oznaka <script> se koristi za dodavanje funkcionalnosti na internetskoj stranici pomoću JavaScripta. Oznaka <meta> obuhvaća meta podatke internetske stranice koji se moraju učitati svaki put kada se

internetska stranica posjeti. Na kraju, oznaka <link> se koristi za povezivanje HTML-a, CSS-a i JavaScripta. Oznaka <body> koristi se za uključivanje cijelog vidljivog sadržaja internetske stranice. Drugim riječima, sadržaj tijela je ono što će preglednik prikazati na frontendu.

HTML dokument se može izraditi pomoću bilo kojeg uređivača teksta. Potrebno je spremiti tekstualnu datoteku koristeći nastavak .html ili .htm. Jednom spremljena datoteka kao HTML dokument može se otvoriti kao internetska stranica u pregledniku.

Značajke HTML – a su sljedeće:

- lagan za naučiti i koristiti,
- neovisan o platformi,
- slike, videozapisi i audio zapisi mogu se dodati na internetsku stranicu,
- hipertekst se može dodati tekstu,
- jezik za označavanje.

HTML je jednostavan jezik za označavanje. Implementacija navedenog jezika je vrlo jednostavna. Koristi se za izradu internetskih stranica. Pomaže u razvoju osnova internetskog programiranja. Također, pomaže i u pojačanju profesionalne karijere.

Najznačajnije prednosti HTML-a: HTML se koristi za izradu internetskih stranica, podržavaju ga svi preglednici, može se integrirati s drugim jezicima kao što su CSS i JavaScript. Najznačajniji nedostaci HTML-a: može stvarati samo statične internetske stranice, za dinamičke internetske stranice moraju se koristiti drugi jezici, za izradu jednostavne internetske stranice potrebno je napisati veliku količinu koda, sigurnosna značajka također nije dobra.

2.1.3. CSS (Cascading Style Sheets)

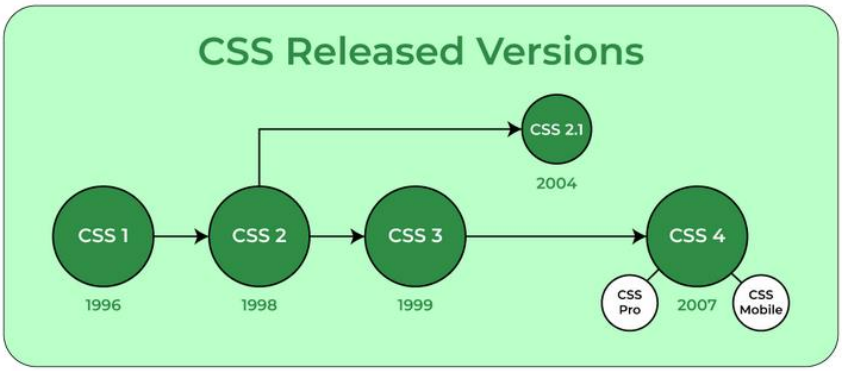
CSS je kratica za Cascading Style Sheets. CSS je jednostavno dizajniran jezik namijenjen pojednostavljenju procesa izrade internetskih stranica. Omogućuje da internetska stranica postane vidljiva neovisno o HTML–u koji čini svaku stranicu. Opisuje kako internetska stranica treba izgledati, odnosno propisuje boje, fontove, razmake i mnoge druge značajke. Ukratko, CSS može učiniti da internetska stranica izgleda kako god dizajner poželi. Programerima i dizajnerima omogućava da definiraju kako se ponaša, uključujući raspored elemenata pozicioniranih u pregledniku.

Slika 4 Logo CSS-a

Dok HTML koristi oznake, CSS koristi skup pravila. CSS je jednostavan jezik za naučiti i razumjeti, ali pruža moćnu kontrolu nad prezentacijom HTML dokumenta.

CSS štedi vrijeme. CSS se može napisati jednom i ponovno upotrijebiti isti list na više HTML stranica. Jedna od značajki je jednostavno održavanje. Za globalnu promjenu jednostavno se promijeni stil i svi elementi na svim internetskim stranicama automatski će se ažurirati. Još jedna od značajki su preglednici. CSS se smatra čistom tehnikom kodiranja, što znači da se preglednici neće morati mučiti da pročitaju njegov sadržaj. U odnosu na HTML, u CSS-u se koriste vrhunski stilovi. CSS ima puno širi niz atributa od HTML-a, tako da se HTML stranici pomoću CSS-a može dati mnogo bolji izgled u usporedbi s HTML-ovim atributima. Prednost CSS-a je i izvanmrežno pregledavanje. Naime, CSS može lokalno pohraniti internetske aplikacije uz pomoć izvanmrežne predmemorije. Pomoću toga, omogućen je pregled izvanmrežnih internetskih stranica.

CSS sadrži stilska pravila koja preglednik tumači i zatim primjenjuje na odgovarajuće elemente u dokumentu. Skup stilskih pravila sastoji se od selektora i bloka deklaracije. Selektor pokazuje na HTML element koji će se stilizirati. Deklaracijski blok sadrži jednu ili više deklaracija odvojenih točkom i zarezom. Svaka deklaracija uključuje naziv CSS svojstva i vrijednost, odvojene dvotočkom. CSS deklaracija uvijek završava točkom i zarezom, a blokovi deklaracije su okruženi vitičastim zagradama.



Slika 5 Razvoj CSS-a

CSS selektori koriste se za pronalazak ili odabir HTML elemenata na temelju njihovog naziva elementa, ID – a, klase, atributa i drugog. Postoje univerzalni, elementni, padajući, ID, klasni i grupni selektori. Umjesto odabira elemenata određene vrste, univerzalni selektor jednostavno odgovara imenu bilo koje vrste elementa. Elementni selektor odabire elemente na temelju naziva elementa. U slučaju da se treba primijeniti stilsko pravilo na određeni element samo kada se nalazi unutar određenog elementa, koristi se padajući selektor. ID selektor koristi ID atribut HTML elementa za odabir određenog elementa. ID elementa treba biti jedinstven unutar stranice, tako da se ID selektor koristi za odabir jednog jedinstvenog elementa. Klasni selektor odabire elemente s određenim atributom klase. Grupni selektori se koriste ako postoje elementi s istim definicijama stila. Tada je potrebna grupacija selektora, kako bi se kod minimizirao. Za grupiranje selektora, svaki selektor se odvaja zarezom.

2.1.4. JavaScript

JavaScript je jednostavni, višeplatformski i interpretirani kompajlirani programski jezik koji je poznat kao skriptni jezik za internetske stranice. Dobro je poznat po razvoju internetskih stranica, a koriste ga i mnoga okruženja bez preglednika. Može se koristiti za razvoj na strani klijenta i za razvoj na strani poslužitelja. JavaScript je imperativni i deklarativni tip jezika. JavaScript sadrži standardnu knjižnicu objekata kao što su Array, Date i Math te osnovni skup jezičnih elemenata, kao što su operatori, kontrolne strukture i izjave.

Slika 6 Logo JavaScript

Kada se koristi za razvoj na strani klijenta, onda JavaScript isporučuje objekte za kontrolu preglednika i Document Object Modela. Proširenja na strani klijenta omogućuju aplikaciji postavljanje elemenata na HTML obrazac i reagiranje na korisničke događaje, kao što su klikovi mišem, unos obrasca i navigacija internetskom stranicom. Korisne knjižnice za klijentsku stranu su AngularJS, ReactJS, VueJS i mnoge druge. Kada se koristi za razvoj na strani poslužitelja, onda JavaScript isporučuje objekte relevantne za pokretanje JavaScripta na poslužitelju. Primjerice, proširenja na strani poslužitelja omogućuju aplikaciji komunikaciju s bazom podataka i osiguravaju kontinuitet informacija od jednog do drugog pozivanja aplikacije ili obavljanju manipulacije nad datotekama na poslužitelju. Jedna od najkorisnijih platforma za korištenje JavaScripta je Node.js. JavaScript je imperativni jezik koji jednostavno kontrolira tijek računanja. Pristup proceduralnom programiranju je objektno orijentirani pristup koji dolazi kao asinkrono čekanje dok se razmišlja što napraviti nakon asinkronog poziva. Deklarativno programiranje zahtijeva logičko računanje. Glavni cilj je opisati željeni rezultat bez izravnog diktiranja o tome kako ga dobiti.

JavaScript se može dodati HTML datoteci kao interni JS ili kao vanjski JS. Kao interni JS, JavaScript se može dodati izravno u HTML datoteku pisanjem koda unutar oznake <script>. Oznaka <script> se može postaviti unutar oznake <head> ili oznake <body> prema zahtjevu. Kao vanjski JS, JavaScript se može zapisati u drugu datoteku koja ima ekstenziju .js i zatim povezati tu datoteku unutar oznake <head> HTML datoteke.

JavaScript je kreirao Brendan Eich 1995. godine dok je bio inženjer u Netscapeu. Izvorno se trebao zvati LiveScript ali je preimenovan. Za razliku od većine programskih jezika, JavaScript nema koncept ulaza i izlaza. Dizajniran je da radi kao skriptni jezik u glavnom okruženju, a na glavnom okruženju da osigura mehanizme za komunikaciju s vanjskim svijetom. Najčešće host okruženje je preglednik.

JavaScript je prvenstveno stvoren za manipulaciju Document Object Modela. Ranije su internetske stranice bile većinski statične, a nakon što je JS kreiran, počeo je razvoj dinamičkih

internetskih stranica. Funkcije u JavaScriptu su objekti. Oni mogu imati svojstva i metode, kao i svaki drugi objekt. Mogu se proslijediti kao argumenti u drugim funkcijama. JavaScript može rukovati također i datumom i vremenom. Može obavljati provjeru valjanosti obrazaca iako su obrasci stvoreni pomoću HTML – a. Osim toga, za JS nije potreban kompajler.

JavaScript se primjenjuje u područjima internetskog razvoja, internetskim aplikacija, poslužiteljskih aplikacija, igara, pametnih satova, umjetnosti, strojnog učenja te mobilnih aplikacija. JavaScript je kreiran za dodavanje interaktivnosti i ponašanja statičnim internetskim stranicama. S tehnologijom, preglednici su se poboljšali do te mjere da je potreban jezik za stvaranje robusnih internetskih aplikacija. JavaScript koristi API, što je kratica za Application Programming Interfaces. Uz pomoć Node.js, JavaScript je pronašao put od klijenta do poslužitelja. Također, pomaže i u stvaranju igara za slobodno vrijeme. Kombinacija JavaScripta i HTML 5 vrlo je popularna u aspektu kreiranja igara. Omogućuje knjižnicu EaseJS koja pruža rješenja za rad s bogatom grafikom. JavaScript se koristi u svim mogućim uređajima i aplikacijama. Omogućuje knjižnicu PebbleJS koja se koristi u aplikacijama za pametne satove. Ovaj okvir radi za aplikacije koje za svoj rad zahtijevaju Internet. Umjetnici i dizajneri mogu stvarati što god požele korištenjem JavaScripta za crtanje na platnu HTML 5. Zvuk čine učinkovitijim uz pomoć p5.js knjižnice. JavaScript ml5.js knjižnica može se koristiti u internetskom razvoju pomoću strojnog učenja. Na kraju, JavaScript se može koristiti i za izradu aplikacije za ne-internetske kontekste. Značajke i upotrebe JavaScripta čine ga moćnim alatom za izradu mobilnih aplikacija. Koristeći React Native mogu se izraditi mobilne aplikacije za različite operativne sustave.

Ograničenja JavaScripta su sljedeća:

- sigurnosni rizici,
- performanse,
- složenost,
- slabo rukovanje pogreškama i mogućnosti provjere tipa.

JavaScript se može koristiti za dohvaćanje podataka pomoću AJAX – a ili manipuliranjem oznaka koje učitavaju podatke, kao što su <img>, <object> i <script>. Takva vrsta napada se zove cross-site script napadi. Oni ubacuju JS koji nije dio internetskog mjesta u preglednik posjetitelja i tako dohvaćaju detalje. JavaScript ne pruža istu razinu performansi koju nude mnogi tradicionalni jezici jer bi složeni program napisan u JS-u bio relativno spor. Budući da se JavaScript koristi za izvođenje jednostavnih zadataka u pregledniku, izvedba se ne smatra velikim ograničenjem u njegovoj upotrebi. Da bi ovladali skriptnim jezikom, programeri moraju imati temeljito znanje o svim konceptima programiranja, objektima temeljnog jezika, objektima na strani klijenta i poslužitelja inače bi im bilo teško pisati napredne skripte pomoću JavaScripta. JavaScript je slabo utipkani jezik jer nema potrebe za specificiranjem tipa podataka varijable. Dakle, provjeru pogrešnog tipa se ne izvodi u kompajliranju.

JavaScript se smatra laganim programskim jezikom zbog činjenice da malo koristi CPU. Također, jednostavan je za implementaciju te ima minimalističku sintaksu. Minimalistička

sintaksa znači nepostojanje tipova podataka. Sve varijable se tretiraju kao objekti. Sintaksa JavaScripta je vrlo slična programskim jezicima C++ i Javi. JavaScript radi u pregledniku iako ima složene paradigme i logiku, što znači da koristi manje resursa od drugih jezika. Osim toga, u usporedbi s drugim programskim jezicima ima manje ugrađenih knjižnica, što isto prikazuje njegovu jednostavnost.

JavaScript se i kompajlira i interpretira. U ranijim verzijama JS – a koristio se samo tumač koji je izvršavao kod red po red i odmah prikazivao rezultat. S vremenom je izvedba postala problem jer je interpretacija prilično spora. U novije verzije JavaScripta implementiran je JIZ kompajler kako bi optimizirao izvođenje te brže prikazivao rezultat. On generira bajt kod koji je relativno lakši za kodiranje. Bajt kod predstavlja skup visoko optimiziranih uputa. V8 motor u početku koristi tumač za tumačenje koda. U daljnjim izvođenjima, motor V8 pronalazi uzorke, kao što su često izvršavane funkcije i često korištene varijable i kompajlira ih zbog poboljšanja performansi.

2.1.5. Quasar

Quasar je alat otvorenog koda za prikupljanje informacija u Githubu. To je alat za prikupljanje obavještajnih podataka i informacija temeljen na OSINT platformi. Quasar je sposoban učiniti gotovo sve što je potrebno za obavljanje izviđanja.

Slika 7 Logo Quasar-a

Prema potrebama, može lako izvršiti izviđanje. Postoje različiti moduli koji rade u ovom alatu. Pomoću različitih modula jednostavno se mogu prikupljati informacije. Informacije uključuju informacije o DNS poslužitelju, naziv organizacije, adresu, grad, poštanski broj, državu, adresu e-pošte koja se odnosi na odgovarajuću organizaciju, registre, poslužitelje za imenovanje, DNS informacije, informacije o internetskoj stranici, provjeru adrese e-pošte, informacije o telefonskom broju, provjeru spremnika za kreditne kartice, Ip lokator, Port Scanner, whois, bing, censys.io, dns, github, dnsdumpster, instagram, crt, ask i dogpile. Navedene informacije su moduli kod kojih alat koristi javno dostupne podatke kako bi dobio meta informacije. Nije potrebno puno toga učiniti jer će se dogoditi automatska konfiguracija pomoću naredbe bash. Quasar je lagani alat koji korisnicima olakšava pisanje kodaVue.js.

2.1.6. Vue.js

Vue.js je jedan od najboljih okvira za JavaScript koji je sličan ReactJS-u. Sloj korisničkog sučelja dizajniran je s Vue.js-om koji je jednostavan za naučiti bilo kojem programeru. Također, može se koristiti s drugim knjižnicama i proširenjima.

Slika 8 Logo Vue.js

Ukoliko programer želi izraditi jednostraničnu internetsku aplikaciju tada je Vue.js dobar alat za njega. Budući da postoji mnogo izazova u području razvoja koji se ne mogu riješiti korištenjem jedne knjižnice, Vue.js je interoperabilan s različitim knjižnicama tako da ga se jednostavno može koristiti. Podržavaju ga svi glavni preglednici, kao što su Chrome, Firefox, Internet Explorer i Safari.

Prednosti Vue.js su sljedeće:

- prilagodljivost,
- komponente,
- prijelaz,
- detaljna dokumentacija.

Vue.js omogućuje slobodnu migraciju, osnovnu i učinkovitu strukturu. Također, komponente Vue.js-a pomažu u stvaranju prilagođenih elemenata koji se mogu ponovno koristiti u HTML-u. U Vue.js-u su dane različite metodologije za primjenu prijelaza na HTML komponente kada su uključene ili izbačene iz Document Object Modela. Vue.js pruža jednostavnu krivulju učenja kroz dokumentaciju od točke do točke.

Instalacija Vue.js-a odrađuje se koristeći jedan od sljedeća tri načina:

- CDN file,
- npm,
- CLI.

2.1.7. Node.je

Node.js je okruženje prijelazne platforme otvorenog koda za izvršavanje JavaScript koda izvan preglednika. Potrebno je zapamtiti da Node.js nije okvir ni programski jezik. Node.js se često koristi za izgradnju pozadinskih usluga poput Application Programming Interface, kao što su internetske aplikacije ili mobilne aplikacije. U proizvodnji, također ga koriste mnoge velike tvrtke.

Postoje drugi programski jezici koji se mogu koristiti za izgradnju pozadinskih usluga, no Node.js promaknuo se kao vodeće okruženje za izgradnju pozadinskih usluga zbog svojih značajki:

- jednostavno okruženje,
- brze i visoke skalabilne usluge,
- korištenje JavaScript programskog jezika,

- čist i konzistentan izvorni kod,
- veliki ekosustav za knjižnicu otvorenog koda,
- asinkrona ili ne blokirajuća priroda.

Prednosti korištenja Node.js – a su sljedeće:

- jednostavna skalabilnost,
- internetske aplikacije u stvarnom vremenu,
- brza arhitektura,
- jednostavno učenje i kodiranje,
- predmemoriranje,
- strujanje podataka,
- hosting,
- korporativna podrška.

Programeri rado koriste Node.js, jer jednostavno skalira aplikaciju u vodoravnom i okomitom smjeru. Također, moguće je dodati resurse tijekom skalabilnosti aplikacije. Ako se gradi internetska aplikacija, može se koristiti i PHP, ali uz njega bit će potreban i Node.js. Ukoliko se izgrađuje aplikacija za chat ili za igranje, Node.js ima daleko bolja svojstva i bržu sinkronizaciju. Osim toga, petlja događaja izbjegava preopterećenje HTTP – a za razvoj Node.js-a. Node.js radi na V8 motoru koji je razvio Google. Petlja događaja obrađuje sve asinkrone operacije tako da Node.js djeluje kao brzi paket i sve se operacije mogu obaviti brzo, poput čitanja ili pisanja u bazi podataka, mrežne veze ili sustava datoteka. Node.js je jednostavan za naučiti i kodirati jer koristi programski jezik JavaScript. Omogućuje keširanje jednog modula. Kad god postoji bilo kakav zahtjev za prvi modul on se pohranjuje u memoriju aplikacije tako da se ne mora ponovno izvršavati kod. U Node.js-u HTTP zahtjev i odgovor smatraju se odvojenim događajima. Oni predstavljaju tok podataka. Kada se datoteka obrađuje u trenutku učitavanja to smanjuje ukupno vrijeme i brže se podaci prikazuju u obliku prijenosa. Također, omogućuje vrlo brzo strujanje audio i video datoteka. Paas i Heroku su hosting platforme za implementaciju Node.js aplikacije koje su jednostavne za korištenje. Većina poznatih tvrtki koriste Node.js za izradu aplikacija. Node.js koristi JavaScript tako da većina tvrtki kombinira frontend i backend timove zajedno u jednu jedinicu.

Neke od primjena Node.js – a su sljedeće:

- razgovori u stvarnom vremenu,
- složene jednostrane aplikacije,
- alati za suradnju u stvarnom vremenu,
- aplikacije za strujanje podataka,
- aplikacije temeljene na JSON API – u.

2.2. Baza podataka

2.2.1. Google Firebase

Google Firebase je softver za razvoj aplikacija uz podršku Googlea. Firebase programerima omogućuje razvoj iOS, Android i web aplikacija. Pruža alate za praćenje analitike, izvješćivanje i popravljanje padova aplikacija, stvaranje marketinški i eksperimentalnih proizvoda.

Slika 9 Logo Google Firebase-a

Google Firebase nudi niz usluga, kao što su:

- Google Analytics,
- autentifikacija,
- slanje poruka u oblaku,
- baza podataka u stvarnom vremenu,
- Firebase Crashlytics,
- performanse,
- testni laboratorij.

Google Analytics za Firebase nudi besplatno i neograničeno izvješćivanje do čak pedeset zasebnih događaja. Analytics predstavlja podatke o ponašanju korisnika u iOS i Android aplikacijama, omogućujući bolje donošenje odluka o poboljšanju performansi i marketingu aplikacija.

Firebase Authentication programerima olakšava izgradnju sigurnih sustava autentifikacije i poboljšava iskustvo prijave i uključivanja za korisnike. Nudi cjelovito rješenje identiteta, podržavajući račune e-pošte i lozinke, autentifikaciju telefona, kao i prijavu na Google, GitHub, Facebook, Twitter i ostalo.

Firebase Cloud Messaging je alat za razmjenu poruka na više platformi koji tvrtkama omogućuje pouzdano primanje i isporuku poruka na iOS-u, Androidu i web aplikacijama bez ikakvih troškova.

Firebase Realtime Database je NoSQL baza podataka smještena u oblaku koja omogućuje pohranu i sinkronizaciju podataka između korisnika u stvarnom vremenu. Podaci se sinkroniziraju na svim klijentima u stvarnom vremenu i još uvijek su dostupni kada se aplikacija isključi s mreže.

Firebase Crashlytics izvještava o rušenju aplikacije u stvarnom vremenu i pomaže razvojnim programerima u praćenju, određivanju prioriteta i rješavanju problema stabilnosti koji smanjuju kvalitetu njihovih aplikacija. Programeri tada troše manje vremena na organizaciju i rješavanje problema s rušenjima, a više vremena troše na izgradnju značajki za svoje aplikacije.

Firebase Performance Monitoring daje razvojnim programerima uvid u karakteristike performansi njihovih iOS i Android aplikacija kako bi im pomogla da odrede gdje i kada se performanse njihovih aplikacija mogu poboljšati.

Firebase Test Lab je infrastruktura za testiranje aplikacija temeljena na oblaku. Jednom radnjom programeri mogu testirati svoje iOS i Android aplikacije na različitim uređajima i konfiguracijama uređaja. Mogu vidjeti rezultate, uključujući videozapise, snimke zaslona i zapisnike, na Firebase konzoli.

3. Opis elemenata i funkcionalnosti web aplikacije

U ovom poglavlju će se opisati praktični dio završnog rada, tj. opisati će se postupak izrade web apikacije „Dreams“.

Link na web sjedište: Dreams (fotostudiodreams-f8258.web.app)

Link na GitHub projekta: https://github.com/ZigantRea/Dreams

3.1. Logo

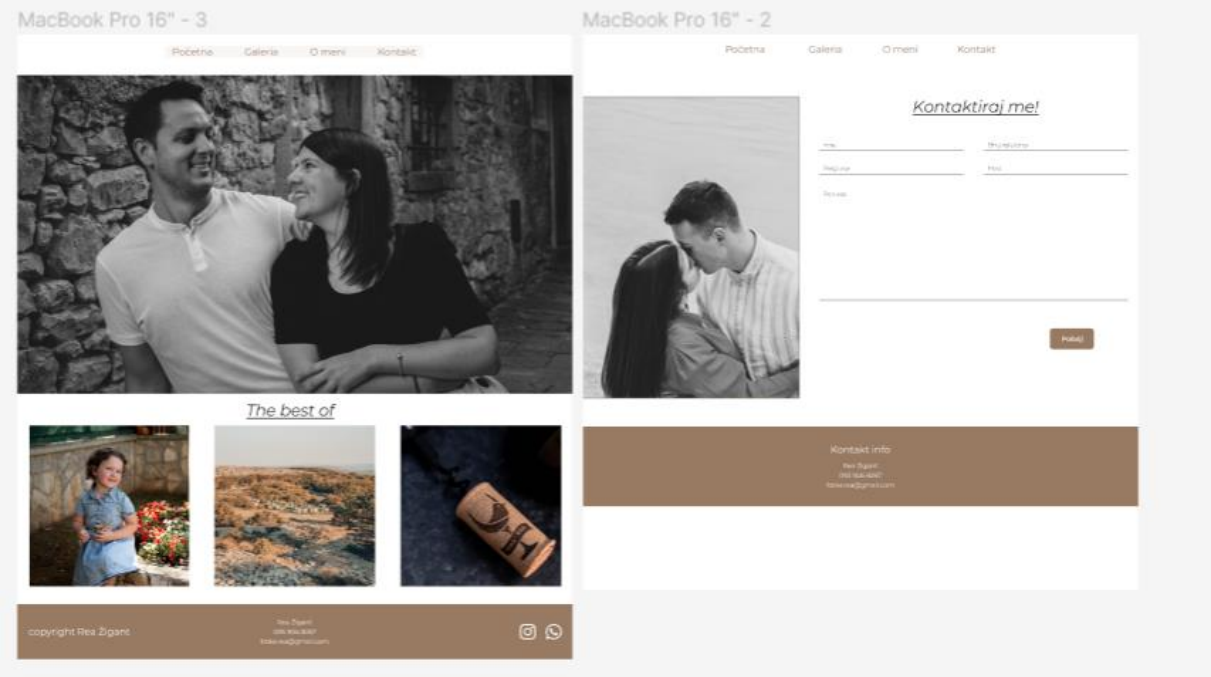
Netom nakon početne ideje o izradi web aplikacije napravljen je vizualni identitet. Logo je napravljen u Official Adobe Photoshop. Jednostavnog je dizajna te je napravljen u dvije verzije. Pri izradi su korištene dvije boje: bijela (#ffffff) i kombinacija smeđe i bež (#987a62). Font je *Amsterdam*.

Slika 10 Logo-verzija 1

Slika 11 Logo-verzija 2

3.2. Ideja dizajna

Prije samog kodiranja u programu *Figma* napravljena je ideja za dizajn web aplikacije. Skica izrađena u *Figma-i* razlikuje se od konačnog produkta, ponajviše iz razloga što se prilikom kodiranja shvatilo da se neki dijelovi mogu jednostavnije napraviti. Dijelovi koji su ostali isti su paleta boja i font. Za boje je odlučeno koristiti trenutno popularnu kombinaciju smeđe i bež u kombinaciji s bijelom i crnom. Font je *Montserrat* koji daje retro doživljaj.



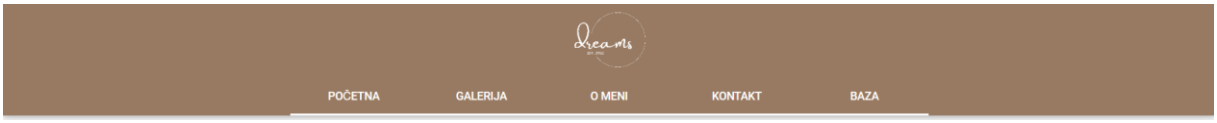
Slika 12 Skice iz Figme

Slika 13 Paleta boja

3.3. Frontend

3.3.1. Header i footer

MainLayout.vue sadrži definirani izgled i funkcionalnosti zaglavlja i podnožja. Zaglavlje se sastoji od logotipa i glavne navigacije. Navigacija sadrži *Početnu*, *Galeriju*, *O meni*, *Kontakt* i *Baza*. Kasnije u tekstu biti će objašnjena svaka od tih stranica.



Slika 14 Izgled navigacijske trake

```
<q-header elevated class="glava">
    <q-toolbar>
    <div class="col">
      <router-link to="/"><q-tab name="images"/>
      <img src="../assets/LOGO.png" width="100px">
    </router-link>
    </div>
    </q-toolbar>

    <q-tabs v-model="tab">
    <router-link to="/"><q-tab name="images" label="Početna"/></router-link>
    <router-link to="/galerija"><q-tab name="images" label="Galerija"/></router-link>
    <router-link to="/omeni"><q-tab name="images" label="O meni"/></router-link>
    <router-link to="/kontakt"><q-tab name="images" label="Kontakt"/></router-link>
    <router-link to="/novo"><q-tab name="images" label="Baza"/></router-link>
    </q-tabs>
  </q-header>
```

Slika 15 Prikaz koda za navigaciju

Logo se nalazi unutar klase *col* te mu je definirana veličina. Na logo je postavljena navigacija, a klikom na njega korisnik se vraća na početnu stranicu.

Kako bi se moglo navigirati po stranicama koristi se oznaka *router-link*, a svakoj oznaci definira se putanja, ime, te tekst koji će biti prikazan na stranici.

Osim zaglavlja u *MainLayout.vue* nalazi se i podnožje, koje sadrži samo kratki tekst i dva hiperlinka. Klikom na ikone korisnika se vodi do Instagram ili Facebook profila stranice. Kako bi se ikone prikazale na web sjedištu potrebno je u kod dodati jedan link koji sadrži ikone spomenutih društvenih mreža. Putanja se definira unutar oznake *href*.

Slika 16 Izgled podnožja

```
<q-footer reveal elevated>
  <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <q-toolbar>
      <h7>© 2022 Rea Žigant</h7>
      <p v-for="n in 145" :key="n"> </p>
      <a href="https://www.instagram.com/rea_zigant/" class="fa fa-instagram" style="font-size:100%"></a>
      <a href="https://hr-hr.facebook.com/" class="fa fa-facebook" style="font-size:100%" ></a>
    </q-toolbar>
</q-footer>
```

Slika 17 Prikaz koda podnožja

U klasama *fa fa-instagram* te *fa fa-facebook* definiran je izgled u *style* dijelu koda. Dodana je neprozirnost koja se pojavljuje ukoliko se s mišem prijeđe preko linkova.

```css
.fa {
  padding: 10px;

  text-align: left;
  text-decoration: none;
}

.fa:hover {
    opacity: 0.7;
}

.fa-instagram {
  background: #987A62;
  color: white;
}
.fa-facebook {
  background: #987A62;
  color: white;
}
```

Slika 18 Prikaz CSS koda vezanog za ikone

3.2.2. Naslovna stranica

Naslovna stranica web aplikacije sastoji se od *carousel-a*, te fotografija i teksta. *Carousel* se samostalno vrti ali postoji mogućnost ručnog pregledavanja fotografija klikom na strelice sa strane.

Forma za *carousel* preuzeta je iz *quasar framework-a* te su joj dodane fotografije.

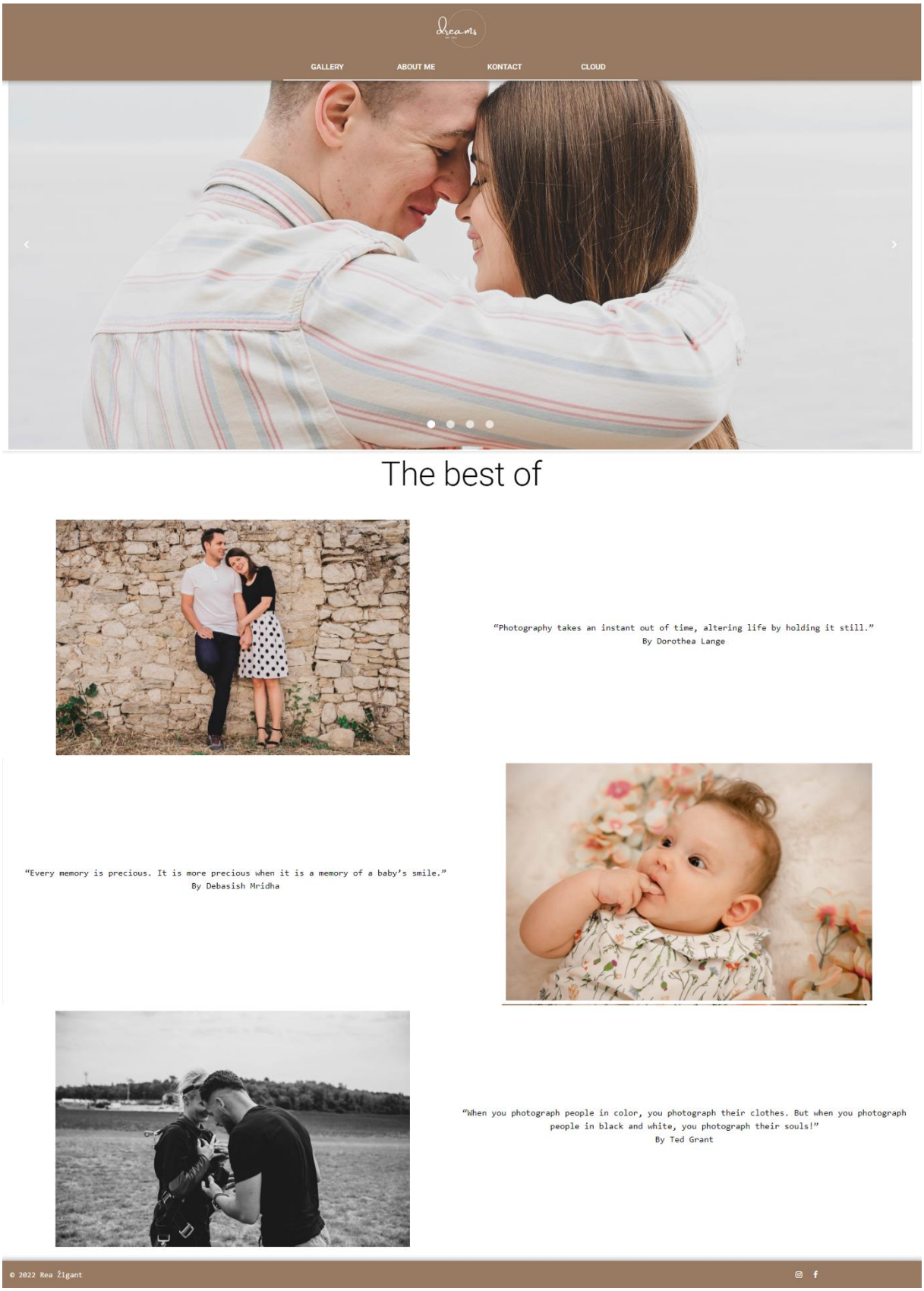
```
<q-carousel
  animated
  v-model="slide"
  navigation
  infinite
  :autoplay="autoplay"
  arrows
   height="750px"
  transition-prev="slide-right"
  transition-next="slide-left"
  @mouseenter="autoplay = false"
  @mouseleave="autoplay = true">
  <q-carousel-slide :name="1" img-src="../assets/fotke/love/DSC_5150-3.jpg" />
  <q-carousel-slide :name="2" img-src="../assets/fotke/love/DSC_7048.jpg" />
  <q-carousel-slide :name="3" img-src="../assets/fotke/kids/DSC_8312-3.jpg" />
  <q-carousel-slide :name="4" img-src="../assets/fotke/travel/DSC_8710.jpg" />
</q-carousel>
```

Slika 19 Prikaz koda vezanog u carosel

Na fotografije je dodan *overlay* u *style* dijelu koda. Prilikom prelaska mišem po fotografiji pojavljuje se neprozirnost s tekstom. Postavljena je boja pozadine s neprozirnošću, te je bilo potrebno definirati veličinu objekta s obzirom na fotografiju.

```css
.overlay {
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  background-color: #987A62;
  opacity: 80%;
  overflow: hidden;
  width: 0;
  height: 100%;
  transition: .5s ease;
}

.container:hover .overlay {
  width: 100%;
}

.text {
  color: white;
  font-size: 50px;
  position: absolute;
  overflow: hidden;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
}
```

Slika 20 Prikaz CSS koda vezanog uz dodavanje efekata fotografijama

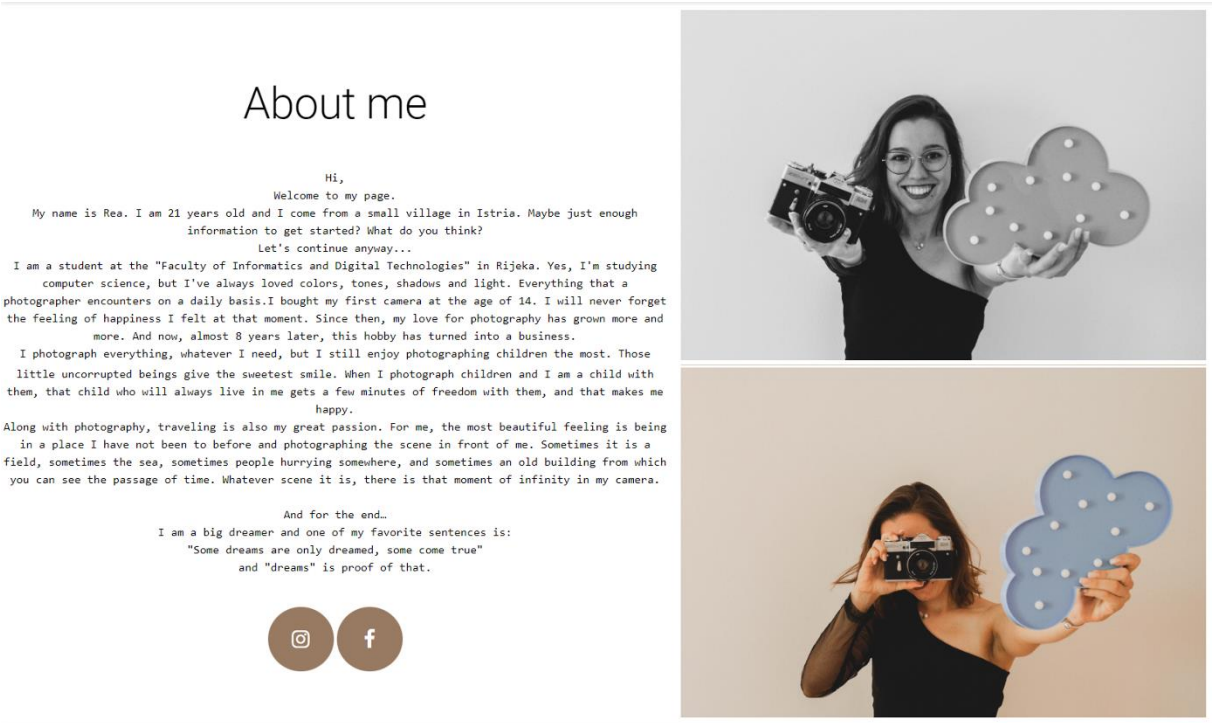


Slika 21 Izgled početne stranice

3.2.3 O meni

Stranica *O meni* sadrži tekst i dva hiperlinka. Linkovi su definirani na isti način kao što je već navedeno u poglavlju *3.2. Header i footer*.

Tekst je definiran unutar klase *row* te ima *justify* poravnanje zbog dinamičnosti.



Slika 22 Izgled stranice "About me"

3.2.4 Kontakt

Unutar *templets* odjeljka nalazi se forma za slanje e-maila. To se postiže postavljanjem *q-form-e*. Korisnik može upisati naslov poruke te napisati poruku, klikom na *send* otvara se E-mail. Postavljen je takav način slanja poruke iz razloga što je korisnik već naučen na korištenje E-maila. Klikom na *Delete* briše se napisano.

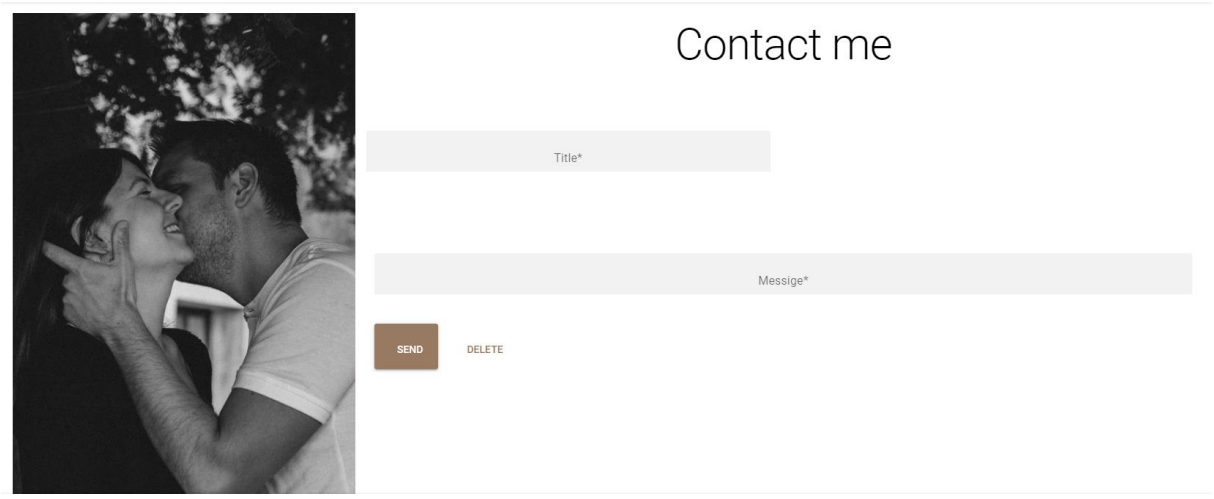
```
<q-form @submit="submit" class="q-gutter-md">
  <div class="col">
    <h1 style="font-size:60px; text-align: center;" >Contact me</h1>
    <div class="row">
        <q-input
          filled
          v-model="kontakt.naslov"
          label="Title*"
          lazy-rules
          :rules="[ val => val && val.length > 0 || 'Please type title']"
        />
        <q-input
          filled
          v-model="kontakt.poruka"
          label="Messige*"
          lazy-rules
          :rules="[ val => val && val.length > 0 || 'Please type messige']"
        />
    </div>

    <div class="col">
        <div class="row">
          <q-btn label="Send" color="primary" style="text-align: left" type="submit" />
          <q-btn label="Delete" type="reset" color="primary" flat class="q-ml-sm" />
        </div>
    </div>
  </div>
</q-form>
```

Slika 23 Prikaz koda vezanog uz forme

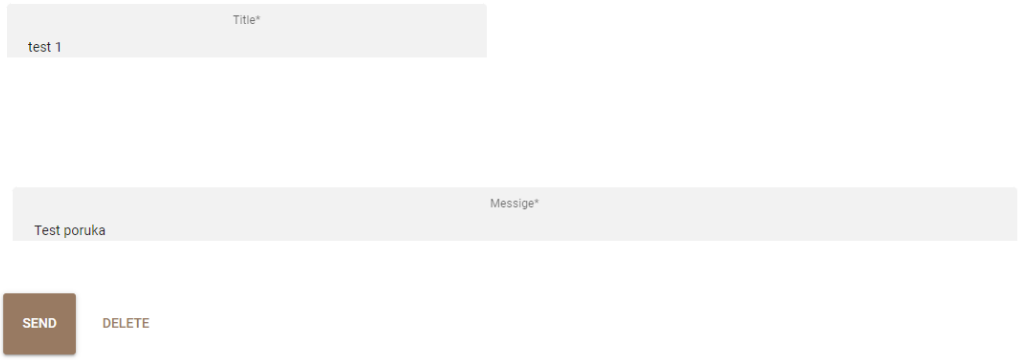
Sve poslane poruke pristižu na E-mail fotke.rea@gmail.com koji je postavljen u *script* dijelu koda.

```
methods: {
  submit () {
    console.log(this.kontakt)
    window.open('mailto:fotke.rea@gmail.com?subject=' + this.kontakt.naslov + '&body=' + this.kontakt.poruka)
  }
}
```

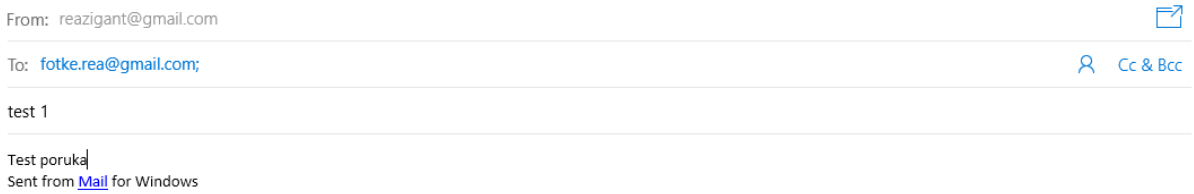
Slika 24 Prikaz JS koda vezanog za mail



Slika 25 Izgled stranice "Contact me"



Slika 26 Prikaz poruke na web sjedištu



Slika 27 Prikaz poruke na e-mail-u

3.4. Backend

U sklopu rada korišten je servis Firebase Storage koji je namijenjen skladištenju raznovrsnih podataka (tekstualne i binarne datoteke, multimedijski sadržaj i ostalo). Za uslugu Firebase, Google nudi stabilan SDK (Software Development Kit) koji omogućava jednostavno korištenje Firebase servisa. Firebase SDK korišten je sa svrhom dvosmjerne komunikacije web aplikacije i Firebase Storage servisa.

Funkcija `initializeApp` postavlja instancu klase Firebase, a kao argument prima JS objekt sa sigurnosnim informacijama potrebnim za ostvarivanje dvosmjerne komunikacije (src/plugins/firebase.js):

- apiKey: autentifikaciji ključ
- authDomain: domena dostupnih Firebase usluga
- projectId: identifikator projekta vezan uz Firebase račun
- storageBucket: naziv Firebase Storage bucketa (nešto kao Google Drive)
- messagingSenderId: identifikator korisnika Firebase usluga
- appId: identifikator aplikacije vezane uz Firebase račun

Komponente `Novo.vue` i `Galeria.vue` komuniciraju s Firebase Storage servisom

Komponente koriste `Map` instancu koja sliči switch naredbi, a služi kao referenca naziva kategorije koji se prezentira korisniku i naziv foldera koji se koristi za skladištenje u Firebase Storage-u.

Ova se stranica nalazi isključivo zbog prezentacijskog dijela završnog rada, u samoj implementaciji je neće biti jer će se dodavanje fotografija vršiti isključivo putem Firebasea.

3.4.1. Galerija

Fotografije koje se dodaju putem Firebasea ili putem stranice *Baza* biti će prikazane na ovoj stranici. Kako bi se to postiglo potrebno je definirati nazive kategorija spremljenih fotografija na *Firebase Storage.*

```
const categoryCollection = new Map();
categoryCollection.set('Kids and Family', 'kids-and-family');
categoryCollection.set('X-Mas', 'x-mas');
categoryCollection.set('Love', 'love');
categoryCollection.set('Komercijalno', 'komercijalno');
categoryCollection.set('Travel', 'travel');
```

Slika 28 Prikaz definiranih kategorija

Zatim se koristi *listAll* funkcija za dohvaćanja svih objekata spremljenih u *Firebase Storage* folderu. Funkcija iterira kroz listu svih objekata *res.items* i dohvaća javno dostupan URL *getDownloadUrl* na objekt (fotografiju). Prva 3 objekta sprema u JS polje *header*, a preostale objekte sprema u JS polje *carousel*.

```
listAll(sref(storage, categoryCollection.get(category)))
  .then(async (res) => {
    const obj = {
      name: category,
      header: [],
      carousel: []
    }

    for (const [i, itemRef] of res.items.entries()) {
      const url = await getDownloadURL(sref(storage, itemRef.fullPath))

      if (i < 3) {
        obj.header.push(url)
      } else {
        obj.carousel.push(url)
      }
    }
    if (obj.header.length > 0) { allImages.push(obj) }
  })
```

Slika 29 Prikaz koda funkcije listAll

Podaci koji se prezentiraju kroz komponentu imaju slijedeću strukturu:

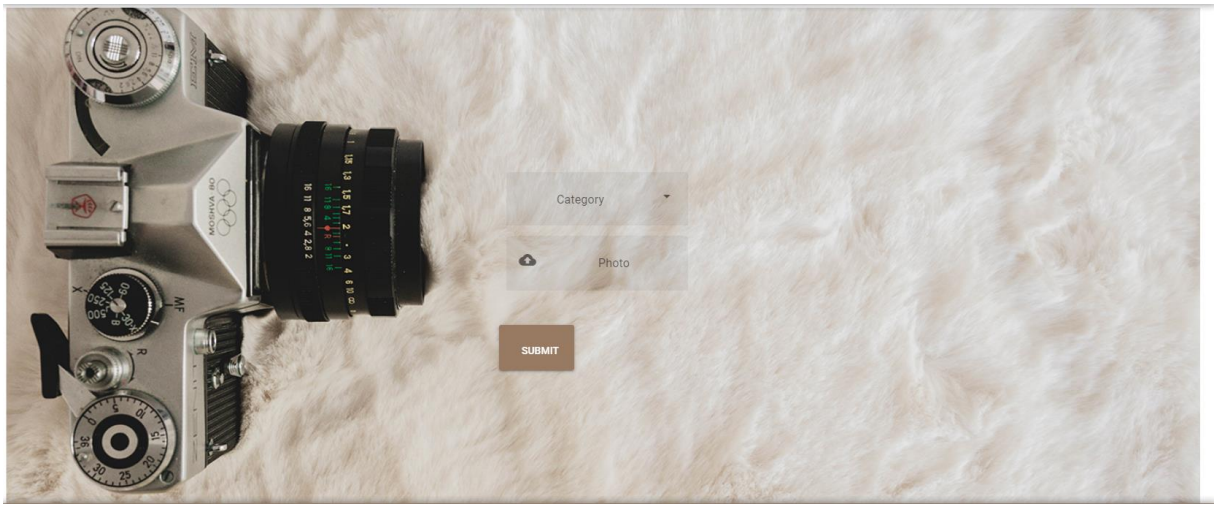
```
{
    name: "kids-and-family",
    header: [
      '{storageBucket}/kids-and-family/image-1.jpg',
      '{storageBucket}/kids-and-family/image-2.jpg',
      '{storageBucket}/kids-and-family/image-3.jpg'
    ],
    carousel: [
      '{storageBucket}/kids-and-family/image-3.jpg',
      '{storageBucket}/kids-and-family/image-4.jpg',
      '{storageBucket}/kids-and-family/image-5.jpg'
    ]
  }
```

3.4.2. Baza

Korisniku se nudi popis svih kategorija zadanih u *Map* instanci te koristi naziv datoteke kako bi mogao učitati sve odabrane fotografije unutar *Firebase Storage*. Koristi se *uploadBytes* funkcija za spremanje svakog učitanog objekta (fotografije) na odgovarajuću *Firebase Storage* datoteku. Funkcija *uploadBytes* „čeka“ dok sve fotografije ne dospiju na *Firebase Storage* prije nego se korisnika usmjeri na popis svih fotografija, tj. na modul *Galeria.vue*, gdje se zatim fotografije ponovno skidaju.

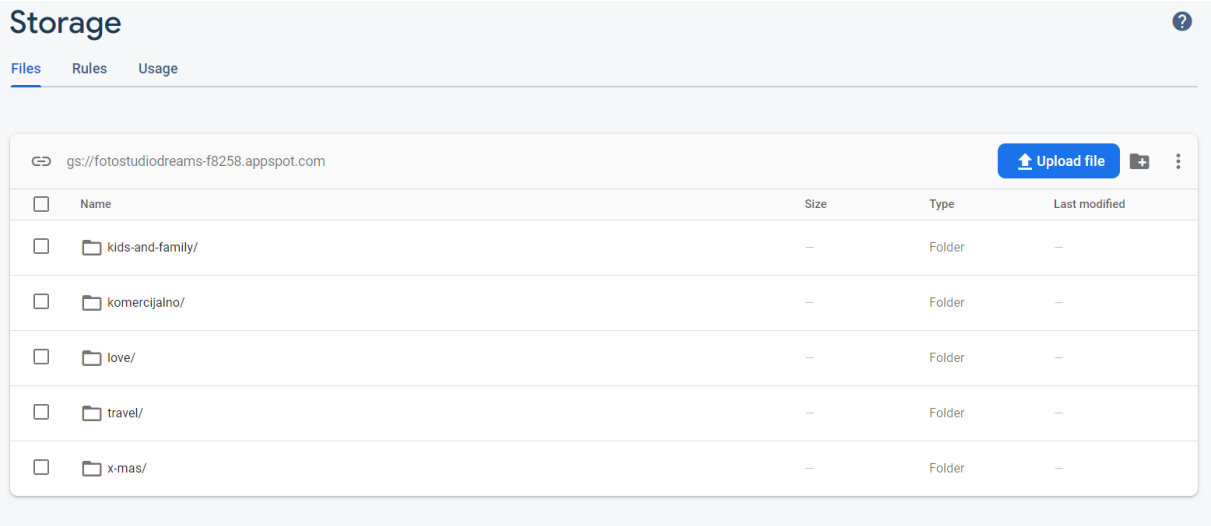
```
methods: {
  async submit () {
    try {
      for (const file of this.image.files) {
        console.log(file)
        await uploadBytes(ref(storage, categoryCollection.get(this.image.category) + '/' + file.name), file).then((snapshot) => {
          console.log('Uploaded an image!');
        });
        this.$router.push('galerija');
      }
    } catch (error) {
      console.log(error)
    }
  }
}
```

Slika 30 Prikaz koda funkcije uploadBytes



Slika 31 Izgled stranice "Cloud"

Fotografije se u galeriju mogu staviti i pomoću Firebase preglednika. U sekciji *Storage* nalaze se mape koje su definirane u stranici *Baza*. Odabirom na mape dolazi se do popisa fotografija koje se nalaze na web sjedištu. Klikom na *Upload file* dodaje se nova fotografija.



Slika 32 Prikaz mapa na Firebase-u

4. Zaključak

U ovom praktičnom završnom radu napravljena je web aplikacija *Dreams* za Foto & Video digital studio *Dreams*. Navedeni obrt se bavi visokokvalitetnim fotografiranjem i video – snimanjem raznih prigoda kao što su rođendani, vjenčanja i ostale tematske zabave. Web aplikacija *Dreams* služi kao marketinški alat za promoviranje usluga obrta. Također, pomaže poslovanju u unaprjeđenju svojih usluga pregledavanjem recenzija i narudžbi klijenata. Aplikacija *Dreams* namijenjena je tržištu te se može koristiti u komercijalne svrhe. Svakim danom broj malih, srednjih i velikih poduzeća raste te je u moru poslovanja iste tematike problematično istaknuti se. Sve više i više tvrtki ima web stranice kako bi privukli nove klijente. Kako bi se još više istaknula, poslovanja su počela raditi web aplikacije. Web aplikacijama omogućeno je spajanje web stranica i baza podataka. Aplikacijom *Dreams* poduzetnik može komunicirati s klijentima, promovirati svoje usluge te pratiti najnovije trendove u području rada. S druge strane, klijent može putem aplikacije slati narudžbe, recenzije i pritužbe na usluge poslovanja. Aplikacija *Dreams* je napravljena korištenjem Vue.js alata, koji su omogućili *user – friendly* sučelje koje je jednostavno za koristiti. Također, rezultat jednostavnog načina rada u aplikaciji je povećanje korisnika i klijenata što vodi ka povećanju posla i zarade. Kako je aplikacija napravljena na engleskom jeziku, što je najčešće govoreni jezik na svijetu, tako se otvorila mogućnost širenja poslovanja i izvan granica Republike Hrvatske i konkuriranja na svjetskom tržištu.

Bibliografija

Brown, E. (n.d.). Learning JavaScript. O'REILLY.

Duckett, J. (n.d.). HTML & Css Design and build websites. U J. Duckett. Wiley.

Figma. (10. rujan 2022). Dohvaćeno iz https://www.figma.com/

JavaSript.com. (rujan 2022). Dohvaćeno iz https://www.javascript.com/

Quasar. (rujan 2022). Dohvaćeno iz https://quasar.dev/

Syed, B. A. (n.d.). Beginning Node.js. U B. A. Syed, *Beginning Node.js.* Aspress.

TechTarget. (10. rujan 2022). Dohvaćeno iz https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app

Vue.js. (rujan 2022). Dohvaćeno iz https://vuejs.org/

w3school. (rujan 2022). Dohvaćeno iz https://www.w3schools.com/html/html_intro.asp

Popis slika