Izrada CSS razvojnog okvira

Mikolaj, Jasmin

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:571730

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

University North Digital Repository

Sveučilište
Sjever

Završni rad br. 692/MM/2020

Izrada CSS razvojnog okvira

Jasmin Mikolaj, 2275/336

Varaždin, rujan 2020. godine

Sveučilište
Sjever

Multimedija, oblikovanje i primjena

Završni rad br. 692/MM/2020

Izrada CSS razvojnog okvira

Student

Jasmin Mikolaj, 2275/336

Mentor

mr. sc. Vladimir Stanisavljević, viši predavač

Varaždin, rujan 2020. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR-42000 Varaždin

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ preddiplomski stručni studij Multimedija, oblikovanje i primjena

PRISTUPNIK Jasmin Mikolaj | MATIČNI BROJ 2275/336

DATUM 12.09.2020. | KOLEGIJ Programski alati 3

NASLOV RADA Izrada CSS razvojnog okvira

NASLOV RADA NA ENGL. JEZIKU Design of a CSS framework

MENTOR mr.sc. Vladimir Stanisavljević | ZVANJE Viši predavač

ČLANOVI POVJERENSTVA

1. doc.dr.sc. Dean Valdec - predsjednik
2. doc.dr.sc. Andrija Bernik, pred. - član
3. mr.sc. Vladimir Stanisavljević, v.pred. - mentor
4. mr.sc. Matija Mikac, v. predavač - rezervni član
5.

Zadatak završnog rada

BROJ 692/MM/2020

OPIS

CSS je jedan od glavnih alata koji web programeri koriste kako bi bolje organizirali oblikovanje stranica i odvojili ga od programskog koda. Rad s modernim CSS-om je prilično kompleksan za početnika pa se često razvijaju male zbirke koje zovemo programski okviri a koje rješavaju određenu klasu primjena. U radu je potrebno:

* opisati osnove CSS-a, njegovu ulogu u oblikovanju Web stranica te analizirati njegove mogućnosti,
* izraditi vlastiti CSS razvojni okvir za osnovne elemente koji se ugrađuju na we stranicu,
* osmisliti i oblikovati web stranicu koja demonstrira mogućnosti CSS programske zbirke,
* ukratko usporediti postignute rezultate sa sličnim sustavima

Detaljno opisati sve korištene tehnologije i korake u radu potrebne da bi se ostvarilo traženo te opisati sva stečena iskustva i postignute rezultate.

ZADATAK URUČEN 17.09.2020. | POTPIS MENTORA

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

Sažetak

CSS razvojni okviri (eng. *CSS Frameworks*) su alati koje najčešće koriste dizajneri i programeri za razvijanje korisničkog sučelja kako bi ubrzali i olakšali posao izrade web-stranice. CSS razvojne okvire možemo još opisati i kao biblioteke koje omogućuju lakšu i bržu izradu web-stranice korištenjem CSS-a, CSS razvojni okvir je zapravo kolekcija CSS datoteka koje se sastoje od već gotovih rješenja spremnih za korištenje.

U ovom završnom radu razvijen je CSS razvojni okvir koji se sastoji od nekoliko gotovih komponenti spremnih za korištenje i izradu web-stranice. Neka od gotovih rješenja su komponente poput gumba radio gumba checkbox-a, kartica, te kontaktne forme. Također sastoji se i od gotovih rješenja spremnih za izradu sekcija web-stranice. Uz razvojni okvir, razvijena je i web-stranica kojoj je svrha pregled svih gotovih komponenti, te kopiranje i pregled izvornog koda potrebnog za implementiranje komponenti.

Ključne riječi: CSS, razvojni okvir, web-stranica, izvorni kod

Summary

CSS Frameworks are tools most commonly used by UI designers and developers to speed up and facilitate the job of creating a website. CSS development frameworks can also be described as libraries that make it easier and faster to create a web page using CSS, CSS development framework is actually a collection of CSS files that consist of ready-made solutions ready to use.

In this final project, a CSS framework has been developed that consists of several ready-made components ready to use and create a web page. Some of the ready-made solutions are components such as buttons, radio buttons, checkboxes, cards, and contact forms. It also consists of ready-made solutions ready to create sections of the website. In addition to the development of a framework, a website has been developed for the purpose of reviewing all finished components, as well as copying and reviewing the source code required for the implementation of components.

Keywords: CSS, framework, web page, source code

Sadržaj

1. Uvod....1
2. CSS Razvojni okviri....2
2.1. Prednosti korištenja CSS razvojnih okvira....5
2.2. Mane korištenja CSS razvojnih okvira....6
3. Korištene tehnologije....7
3.1. HTML....7
3.2. CSS....9
3.3. Javascript....11
3.4. Javascript biblioteka....12
3.4.1. highlight.js biblioteka....12
4. Praktični zadatak....13
4.1. Struktura web-stranice....13
4.1.2. Izrada animacije navigacijskog izbornika....16
4.2. Funckija izbornika za uređivanje komponenti....17
4.3. Kopiranje i prikaz koda uređenih komponenti....19
4.4. Komponente korisničkog sučelja CSS razvojnog okvira....22
5. Rezultat korištenja gotovog rješenja....24
6. Zaključak....25
7. Literatura....26
Popis slika....27

1. Uvod

U zadnjem desetljeću svjedočimo velikom porastu mobilnih uređaja u svrhu pristupa internetu i pregledavanju web-stranica. Povećana upotreba mobilnih uređaja ukazuje da su tijekom izrade web aplikacije dizajneri i programeri morali razmotriti rad te web-stranice na mobilnim uređajima. Web-stranica koja nije optimizirana za pregled na mobilnim uređajima rezultira frustracijom korisnika. Mobilni uređaji koriste mnogo različitih rezolucija koje se uvelike razlikuju od rezolucija računala, na koje su dizajneri i programeri morali voditi brigu tijekom razvoja aplikacije. Te tijekom razvoja web stranica i aplikacija pojavio se novi pojam, a to je responzivni dizajn. CSS okviri [1] pružaju jednostavnu implementaciju responzivnog dizajna za sve postojeće rezolucije, te uz pomoć CSS razvojnih okvira možemo izraditi optimiziranu web-stranicu za sve rezolucije.

Također, danas se web aplikacije razvijaju u različite svrhe, izrada e-trgovina, blogova, društvenih mreža itd. CSS razvojni okviri igraju jako važnu ulogu u razvoju tih web aplikacija. CSS razvojni omogućuju uštedu vremena potrebno za razvoj i dizajn web-stranice. Kod izrade statične web-stranice velik broj komponenti je jednak, te nema potrebe kod svakog razvijanja web-stranice tu komponentu razvijati ponovno od nule. Korištenje CSS razvojnih okvira uistinu pruža niz prednosti, korištenje gotovih rješenja, te unaprijed definiranih klasa koje su dio CSS razvojnog okvira ovog završnog rada značajno je ubrzalo razvoj web-stranice. Možemo zaključiti da je korištenje CSS razvojnih okvira idealno za programere koji žele brzo razviti sadržraj web-stranice, no takav način razvijanja dolazi uz nekoliko ograničenja, kao što je razvijanje kompleksnih dinamičnih stranica. Prije korištenja CSS razvojnog okvira programer treba pripaziti na nekoliko stvari. Kvalitetan CSS razvojni okvir trebao bi imati detaljnu dokumentiju kako bi korisnik lakoćom savladao njegovo korištenje. Razvojni okvir trebao bi često biti ažuriran zbog toga što se web tehnologija brzo razvija, te kako bi razvojni okvir bio upotrebljiv potrebno ga je ažurirati kako bi ostao kompatibilan s internet preglednicima i novim značajkama.

U ovom završnom radu biti će razvijen CSS razvojni okvir koji će uvelike ubrzati i olakšati razvoj web-stranice, web-stranica kojoj će uloga biti demonstrirati komponente razvojnog okvira. U radu će biti objašnjene tehnologije CSS razvojnog okvira, tehnologije koje će biti potrebne za izradu razvojnog okvira i web-stranice, prikaz rezultata, te zaključak.

2. CSS Razvojni okviri

Programeri danas više ne trebaju započeti razvijati od nule jer postoje alati koji značajno olakšavaju i ubrzavaju proces razvoja programskih rješenja i aplikacija. Razvojni okvir [2] je platforma za razvoj softverskih aplikacija. Pruža osnovnu bazu na kojoj programeri mogu graditi programe za određenu platformu. Na primjer, razvojni okvir može sadržavati unaprijed definirane klase i funkcije koje se mogu koristiti za obradu ulaznih jedinica, upravljanje hardverskim uređajima ili interakciju sa sistemskim softverom. To pojednostavljuje proces razvoja jer programeri ne moraju ispočetka programirati bazu na kojoj se temelji aplikacija. Razvojni okvir sličan je aplikacijskom programskom sučelju (eng. *Application Programming Interface*), iako se razvojni okvir tehnički nalazi unutar aplikacijskog programskog sučelja. Kao što i samo ime sugerira, razvojni okvir služi kao osnovna baza za programiranje, dok aplikacijsko programsko sučelje pruža pristup elementima koje podržava razvojni okvir. Razvojni okvir može obuhvatiti i biblioteke, kompajler i druga programska rješenja koja se koriste u procesu razvoja softvera.

Razvojni okvir [1] je skup predefiniranih koncepata, modula i standardiziranih rješenja koji razvoj web aplikacija i web-stranica olakšavaju. Pruža generalnu funkcionalnost s već gotovim modulima koji se mogu iznova koristiti i gotove standardizirano uređene komponente. Ukratko, to je softver koji se koristi za višekratnu uporabu, te omogućuje web dizajnerima i web programerima razvoj projekata s minimalno kodiranja. Smanjuje vrijeme razvoja, te omogućuje brze promjene na projektu. Postoje dvije vrste razvojnih okvira, a to su front-end razvojni okviri u koje pripadaju CSS i JavaScript razvojni okviri, te postoje back-end razvojni okviri. Back-end razvojni okviri koriste web developeri i programeri koji razvijaju aplikacije na serverskoj strani, dok se front-end razvojni okviri koriste kod razvoja korisničkog sučelja, te samog izgleda stranice implementiranjem CSS-a. Biranje idealnog razvojnog okvira najčešće se odnosi na vanjski razvojni okvir, te odabir ovisi o nekoliko čimbenika, a to je za koje potrebe će se koristiti razvojni okvir, te kvaliteta razvojnog okvira, koje su mu prednosti i koji su mu nedostaci.

Front-end razvojni [4] okviri su biblioteke koje osiguravaju strukturu i grafičku dosljednost web-stranice ili web aplikacije. Strukturu se sastoji od stupaca, tipografije (fontovi, veličina, naslovi, tekst), komponenti (gumbovi, forme, navigacijski izbornici), stanja komponenti, responzivnosti; razvojni okviri donose brojne alate za uštedu vremena u razvoju web stranice. HTML jezik strukturira podatke, a CSS stilizira te podatke kako bi bili vizualno i estetski razumljivi, a javascript omogućuje izradu animacija i interakcija te tako poboljšava korisničko iskustva. Najčešće komponente front-end razvojnog okvira [4] su:

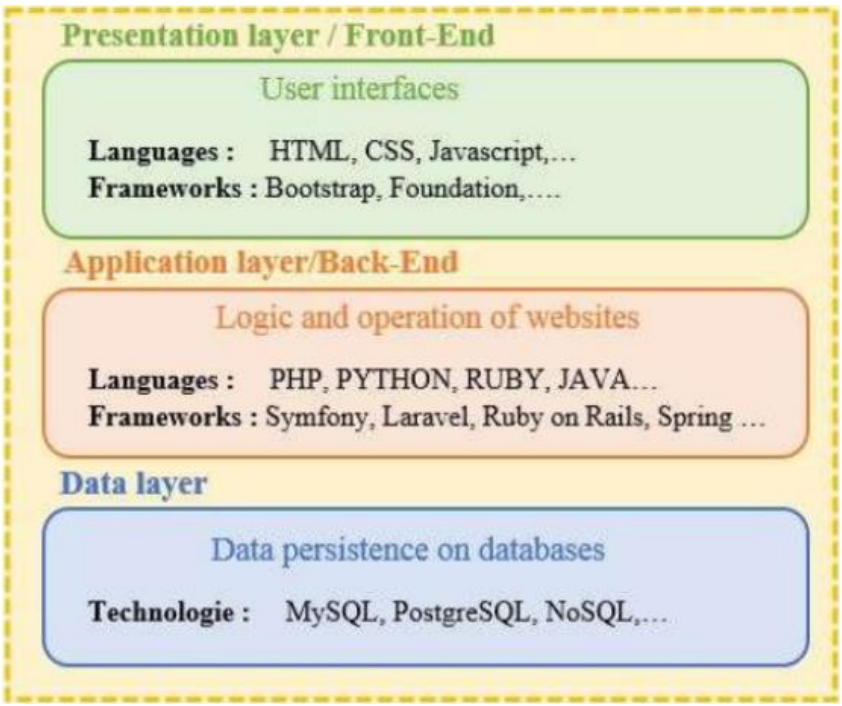
1. CSS izvorni kod za izradu mreže, to omogućava programeru da na jednostavan i svestran način pozicionira različite elemente koji čine dizajn web stranice.
2. Definirani stilovi tipografije za elemente HTML-a
3. Rješenja za slučajeve nekompatibilnosti internetskih preglednika s parametrima u CSS-u, tako da rezultat bude jednak u svakom pregledniku
4. Izrada standardiziranih CSS klasa koje mogu biti korištene u naprednim komponentama korisničkog sučelja

Front-end razvojni okviri [4] pružaju predefinirane i prilagodnjive parametre za komponente web stranice:

1. Kompletnu, prilagodljivu i lako podesivu mrežu web-stranice
2. Moduli koji pokrivaju sve aspekte i trenutne potrebe web stranice
3. Stanja komponenti
4. Višestruka interaktivna ponašanja web stranice
5. Prilagodljiv responzivan dizajn

Razlikuju se dvije vrste CSS razvojnih okvira [5] prema njihovoj složenosti, postoje jednostavni razvojni okviri i složeni razvojni okviri. To razlikovanje je subjektivno i ono ne znači da je jedna vrsta bolje od druge ili suprotno, no možemo reći da daju drugačija rješenja ovisno o složenosti ili fleksibilnosti potrebnoj kod izrade web stranice. Složeni ili cjeloviti razvojni okviri nude mogućnost konfiguracije značajki poput stiliziranja tipografije, stiliziranje obrazaca, gumbova, ikona i druge komponente za višekratnu uporabu poput navigacijskih izbornika, HTML predložaka, te prilagođene postavke. Neki od cjelovitih CSS razvojnih okvira su Bootstrap, Foundation, Skeleton, YAML 4 itd. Na tržištu postoji veliki broj različitih front-end razvojnih okvira, te se javlja pitanje kako odabrati idealan razvojni okvir za svaki određeni projekt. Odabir front-end razvojnog okvira nije lak, te je tome razlog nekoliko stvari. Svaka web stranica je različita i zahtjevati će različite karakteristike.

Kod izbora front-end razvojnog okvira treba uzeti u obzir nekoliko stvari kao što su brzina instalacije, neki razvojni okviri instaliraju se lakoćom, no neki zahtjevaju mnogo vremena i znanja za konfiguraciju. Zatim jednostavnost razumijevanja za korištenje, neki razvojni okviri biti će jednostavni za korištenje, dok će kompleksniji zahtijevati detaljnije proučavanje uputa. Neki razvojni okviri nude puno više opcija i mogućnosti u konfiguraciji elemenata, te pružaju puno više mogućnosti kod same izrade web stranice, čime ona može biti puno bolja i kvalitetnija. Kod izbora razvojnog okvira treba pripaziti i na integracije s ostalim sistemima, te imaju li dugogodišnju podršku ažuriranja, neki digitalni projekti nemaju kontinuitet u razvoju, nemaju nova ažuriranja i korisničku podršku, te je uvijek bolje odabrati projekt koji nudi trajno jamstvo podrške.



Slika 1. Vizualni prikaz tehnologije potrebne za izradu web-stranice
(Izvor: Laaziri, Benmoussa, Khoulji, Kerbeb: https://www.researchgate.net/publication/330656530_Analyzing_bootsrap_and_foundation_font-end_frameworks_a_comparative_study)

2.1. Prednosti korištenja CSS razvojnih okvira

Iskusni programeri preporučuju izbjegavanje CSS razvojnih okvira zbog njegovih potencijalnih nedostataka, te mogućih problema s zadanom strukturom web stranice, usprkos tome korištenjem CSS okvira dobivamo mnogobrojne prednosti.

CSS okviri [7] olakšavaju održavanje koda, održavanje koda je potrebno nakon što je projekt završen, te je web-stranica objavljena. Često se rade promjene u kodu te se dodaju ili mijenjaju funkcionalnosti određenih elemenata ili se dovršavaju određene sekcije web stranice. Održavanje koda je postupak kojim se ne utječe na dijelove web stranice koji trebaju biti konstantni. Budući da razvojni okviri smanjuju veličinu koda, te je kod uvijek pisan predefiniranom i konzistentnom sintaksnom, lako ga je održavati.

Još jedna prednost CSS razvojnih okvira je dosljednost i ujednačenost, kad je nešto predefinirano s posebnim pravilima i funkcijama, ta ista pravila važe za cijeli svijet. Isto vrijedi i kod CSS razvojnih okvira. Te bez obzira tko koristi neki određeni razvojni okvir, za sve vrijede ista pravila, te će ih se morati pridržavati prilkom korištenja razvojnog okvira. Takva pravila čine kod dosljednim, kod je dosljedan i ujednačen tijekom cijelog projekta, što olakšava timski rad na projektu. Dosljednost i ujednačenost pomažu programerima da razumiju kod i uštede puno vremena. Te je to jedan od glavnih razloga zašto programeri ne razvijaju vlastita rješenja, već koriste gotova programska rješenja.

U svijetu razvojnih okvira postoji veliki broj zajednica. Kako su razvojni okviri postali sve popularniji tako je i broj korisnika porastao. S većim brojem korisnika pojavljuje se i porast problema s kojima se korisnici susreću tijekom razvoja. No kako broj korisnika postaje sve veći, tako postoji i puno veća korisnička podrška, te većina problema s kojima se korisnici susreću su rješeni.

Razvijeni CSS razvojni okviri sadrže dokumentaciju koja pomaže kod učenja učinkovitog korištenja razvojnog okvira. Iako je proces izrade dokumentacije podcijenjen, on je vrlo važan i vremenski zahtjevan postupak. Jednom kad netko razvije razvojni okvir, potrebno ga je i dokumentirati kako bi ga korisnici lakše i učinkovitije koristili.

2.2. Mane korištenja CSS razvojnih okvira

CSS razvojni okviri nude mnogobrojne prednosti tijekom razvoja web stranice, no isto tako imaju i nedostatke ili ograničenja, te ponekad nisu idealni za razvoj kompleksnijih web stranica ili aplikacija.

CSS [6] razvojni okviri ograničavaju korisnika tijekom izrade web stranice, te je moguće implementirati komponente koje nudi razvojni okvir. Razvojni okvir ograničiti će način kodiranja i način rada. Određeni razvojni okvir napravljen je imajući na umu svakog programera koji će ga koristiti. Te takav način izrade razvojnog okvira ograničava način rada koji je moža potreban za izradu određene funkcije web stranice, isto vrijedi i za sam dizajn web stranice, s određenim razvojnim okvirom nije moguće izraditit zamišljeni dizajn web stranice. Za mogućnost korištenja razvojnog okvira, potrebno je koristiti već gotove klase s gotovim izgledom neke komponente, isto tako moguće je i mijenjati izvorni kod, no koja je onda svrha uopće koristiti razvojni okvir ukoliko je potrebno nešto promijeniti.

Korištenjem razvojnih okvira ograničava se način na koji se dizajniraju mali elementi u dizajnu. Korištenjem klasa razvojnog okvira dobiva se ista struktura za većinu elemenata. Budući da je veliki element kombinacija mnogih malih elemenata, veliki elementi imaju istu strukturu. Te su zbog toga web stranice razvijene CSS razvojnim okvirom lako prepoznatljive i jednolične. Za većinu programera, ali i klijenata to nije problem, no problem se javlja kad se pojavi potreba za kompleksnijim dizajnom, ili pak kompleksnijim funkcionalnostima web stranice. Ipak, prvi dojam korisnik dobiva pregledom dizajna i strukture web stranice.

Prije početka korištenja razvojnog okvira prvo je potrebno detaljno proučiti dokumentaciju i zapravo naučiti kako koristiti određeni razvojni okvir. Na tržištu postoji mnogo različitih razvojnih okvira, te svaki od njih funkcionira na drugačiji način. Problem se javlja kad jednom naučimo koristiti jedan od razvojnih okvira, no klijent potrebuje korištenje nekog drugog razvojnog okvira, to nas dovodi u situaciju gdje ili će klijent napustiti projekt ili će programer morati naučiti koristiti novi CSS razvojni okvir.

3. Korištene tehnologije

Za izradu završnog rada korišteno je nekoliko različitih tehnologija, za izradu web stranice na kojoj su prikazani svi elementi korišteni su HTML5, CSS3, JavaScript, te JavaScript biblioteka "highlight.js".

3.1. HTML

Kratica za HTML [7] *je HyperText Markup Language*. HTML je jezik kojim označujemo strukturu, sadržaj i funkciju HTML dokumenta. Označavanje je proces kojim korištenjem oznaka (eng. *tag*) stvaraju i struktruriraju elementi HTML-dokumenta. HTML pruža niz mogućnosti kao što je određivanje važnih elemenata svakog HTML-dokumenta kao što su naslov, odlomak, slika. HTML je jezik kojim se označavanjem može odrediti struktura elemenata unutar HTML-dokumenta. Osim strukture elemenata koja je definirana unutar HTML-dokumenta, CSS omogućuje da se ti elementi stilski urede. Tim postupkom web-preglednik zna kako će stranica izgledati.

HTML je razvio Tim Barners Lee početkom 90- godina. Vrlo brzo stječe popularnost i masovnu adaptaciju koja značajno utječe na razvoj interneta. Tako nagli i uspješan razvoj HTML-a rezultira izadnjem nekoliko HTML standarda koji međusobno nisu usklađeni. Na početku je HTML omogućavao izradu jednostavno strukturirane web-stranice koja se sastojala od odlomaka, zaglavlja i prijeloma redova, no nije bio moguć unos grafičkih elemenata i multimedijalnih elemenata. Na samome početku nije bio bitan izgled web-stranice niti se o tome vodila briga,

Svaki HTML element se sastoji od HTML oznake (eng. *tag*). Između dvije HTML oznake unosi se sadržaj, HTML oznake imaju funkcionalnost spremnika koji interpretira vrstu informacije koja se nalazi u tom spremniku, npr. unutar HTML-oznaka nalazit će se odlomak teksta. "Kako bi do kraja razjasnili anatomiju HTML-elementa potrebno ga je raščlaniti na osnovne dijelove. Svaka početna HTML-oznaka sastoji se od šiljastih zagrada i identifikatora. Kako bi kod bio čitljiviji, može ga se pisati na način da svaki HTML-element bude u svom redu. HTML-elementi koji nemaju sadržaj imaju samo jednu HTML-oznaku. Tako HTML-element za vodoravnu crtu, s obzirom na to de nema sadržaja ima samo jednu oznaku <hr>, dok HTML-element za prelazak u novi red ima samo oznaku <br>."

Neovisno o kompleksnosti i sadržaju svi HTML [7] dokumenti imaju jednake i zajedničke dijelove kojima se definira struktura HTML dokumenta. Elementi zajednički svim HTML dokumentima su *<html> </html>*, *<head> </head>*, *<title> </title>*, *<body> </body>*. Element obavezan na svoj stranici je html, to je prvi element unutar kojeg se nalazi sadržaj HTML dokumenta, a služi kako bi internet preglednik znao prepoznati da je riječ o HTML dokumentu. Cijeli sadržaj dokumenta nalazi se unutar elementa *<html>*, te dokument počinje i završava s tim oznakama. Na vrhu svakog HTML dokumenta prvo se navodi HTML element *<head>* u kojem se nalaze informacije of web-stranici koje se ne prikazuju kao sadržaj te web-stranice. Element *<head>* predstavlja zaglavlje stranice u koji se smješta naslov web-stranice, naslov web-stranice smješta se u element *<title>*. Naslov web-stranice se prikazuj na naslovnoj traci internet preglednika. Sadržaj koji se nalazi unutar *<body>* oznaa je sadržaj koji će se prikazati u internetskom pregledniku.

Slika 2. HTML struktura dokumenta (Izvor: CodeCracker, HTML Document Structure: https://codescracker.com/html/images/html-document-structure-example.jpg)

3.2. CSS

CSS [7] (eng. *Cascading Style Sheets*) je jednostavan jezik kojemu je svrha olakšati proces izrade prezentacije izgleda web-stranice, lakoćom se dodaju stilovi poput fontova, boja, razmak između paragrafa, veličine elemenata, pozicije elemenata, boju pozadine, te mnogobrojni efekti. Svojim dolaskom CSS je izazvao pravu revoluciju na internetu i to zahvaljujući nizu prednosti koje ima pred tabličnim layoutom (korištenje tablica za formiranje stranice). Korištenjem CSS-a postalo je moguće odvojiti prezentaciju podataka i dizajn od same strukture podataka. Korištenjem CSS-a HTML kod postaje pregledniji i manji, što znači da korisnik ima puno više kontrole i mogućnosti, te ima mogućnost uz jednostavnu primjernu parametara promijeniti izgled web-stranice. Prije postojanja CSS-a bilo je puno teže oblikovati izgled web-stranice, postojale su mnoge zapreke i ograničenja, te kod dizajniranja HTML strukture korisnik je bio ograničen na tablice, kontrolu fontova i nekoliko stilova teksta poput *bold* i *italic*. Sa stilskim obrascima moguće je kontrolirati svaki aspekt prikaza na stranici, moguće je odrediti razmak između linija, znakova, margina, te pozicije slika, primijeniti promjene na cijelu internet stranicu, osigurati dosljedan dizajn web-stranice tako da isti stilski obrzazac možemo koristiti na više web-stranica, dati internet pregledniku instrukcije kako kontrolirati izgled web-stranice, te kreirati dinamične stranice.

CSS koristi se za definiranje pravila u stilskim obrascima kojima određujemo kako da određeni sadržaj definiran HTML kodom izgleda, te da bi određeni sadržaj uredili pomoću stilskog obrasca potrebno je povezati stilska pravila i HTML kod. Pomoću CSS obrasca moguće je odrediti bilo koji element, sekciju ili komponentu web-stranice kao što je podešavanje pozadine, boju pozadine i sliku na pozadini, podesiti opcije okvira, podesiti opcije klasifikacije (kontroliraju na koji način se elementi poput slika ponašaju na stranici u odnosu na ostale elemente), uređivanje listi, uređivanje margina, kontrola pozicioniranja elemenata, kontrola veličine elemenata.

“Stilski obrasci sačinjeni su od stilskih pravila. Svako pravilo ima dva dijela, selektora koji određuje element na koji se stilsko pravilo odnosi, i deklaracija koja određuje kako izgleda sadržaj opisan CSS-om. Koristimo se sa setom interpunkcijskih i posebnih znakova kako bi definirali stilsko pravilo. Sintaksa za stilska pravila uvijek slijedi sljedeći uzorak: [7] “

```
selektor {deklaracija;}
```

“Deklaracija se dijeli na dvije stavke, a to su properties (aspekti kako da računalo prikaže tekst i grafiku), te values (podaci koji određuju kao želimo da tekst i slike izgledaju na našoj stranici). Properties se od vrijednosti (value) u deklaraciji odvaja dvotočkom, a svaka deklaracija završava s točkom-zarez: [7]”

```
selector {property: value;}
```

3.3. Javascript

Javascript [8] dinamičan je računalni programski jezik. Javascript je lagan programski jezik, što znači da ne opterećuje internet preglednike i veličina datoteka je malena. Većinom se koristi kao dio web-stranice, te njegova implementacija omogućuje interakciju skripte klijentske strane s korisnikom za izradu dinamične web-stranice. Javascript je interpretiran programski jezik s mogućnošću objektno-orijentiranog programiranja. Javascript je prvo bio poznat kao LiveScript, no Netscape je promijenio ime u JavaScript, prvi puta se pojavljuje 1995. godine.

Prednosti javascript [8] programskog jezika su:

1. Manje interakcije s serverom, što znači da korisnik prije slanja unosa neke forme može pomoću javascripta provjeriti točnost podataka prije slanja podatke iz forme serveru. Time smanjujemo opterećenost servera, te poboljšavamo kvalitetu i brzinu web-stranice.
2. Korisnik neke web-stranice više ne treba osvježiti web-stranicu kako bi provjerio ukoliko su uneseni svi potrebni podaci potrebni za određenu formu
3. Povećana interakcija korisnika na web-stranici, što znači da možemo razviti sučelja koja reagiraju i pokreću određenu radnju aktiviranu pokreto pokazivača ili pritiskom tipke
4. Bogatije korisničko sučelje, korištenjem javascripta možemo razviti komponente kao što su blokovi za postavljanje datoteka na server ili pak klizača s kojima korisnik može upravljati

Javascript se može implementirati pomoću javascript izraza koji se smješteni unutar <script> </script> oznaka na web-stranici. Oznake <script> u kojima se nalazi javascript može se smjestiti bilo gdje na web-stranici, no obično se preporučuje da se stavi unutar <head> oznaka. Oznaka <script> internet pregledniku daje upozorenje da interpretira tekst unutar tih oznaka kao skriptu. Jednostavna sintaksa javascript implementacije izgleda ovako:

```
<script>
     Javascript kod
</script>
```

3.4. Javascript biblioteka

Javascript [9] je vjerojatno jedan od najraširenijih programskih jezika na svijetu. Poakazao se kao dobra zamjena za tradicionalne platforme. Kako potreba za korištenjem javascripta raste, tako i način distribucije koda postaje sve važniji, postaje sve važnije pojednostaviti i ubrzati razvoj aplikacija, te razvijati kod koji je moguće ponovno iskoritistiti. Dakle javascript biblioteke pružaju unapijed definirani javascript kod što olakšava uobičajene ili složene zadatke. Najčešće se koriste za zadatake kao što je DOM (eng. *Document Object Model*) manipulacija, postavljanje okvira.

Da bismo u web aplikaciju uključili javascript biblioteku potrebno dodati <script> oznaku unutar <head> oznake s atributom "src" koji upućuje na URL ili putanju izvora biblioteke.

```
<script src="js/lib/library.min.js"></script>
```

3.4.1. highlight.js biblioteka

Highlight [10] je javascript biblioteka koja se koristi za označavanje i prikazivanje sintakse koda. Funkcionira u pregledniku isto kao i na serveru, radi s gotovo svim oznakama, te ne ovisi o razvojnom okviru. Podržava automatsko prepoznavanje programskog jezika, podržava 180 različitih programskih jezika i prilagođava boju i strukturu koda ovisno o kojem se programskom jeziku radi, podržava i mnogo različitih stilova za prikaz koda. Za korištenje ove biblioteke na web-stranici potrebno je povezivanje biblioteke s web-stranicom te pozivanje funkcije hljs.iniHighlightingOnLoad();

```
<link rel="stylesheet" href="/path/to/styles/default.css">
<script src="libraries/highlight.pack.js"></script>
<script>hljs.initHighlightingOnLoad();</script>
```

4. Praktični zadatak

4.1. Struktura web-stranice

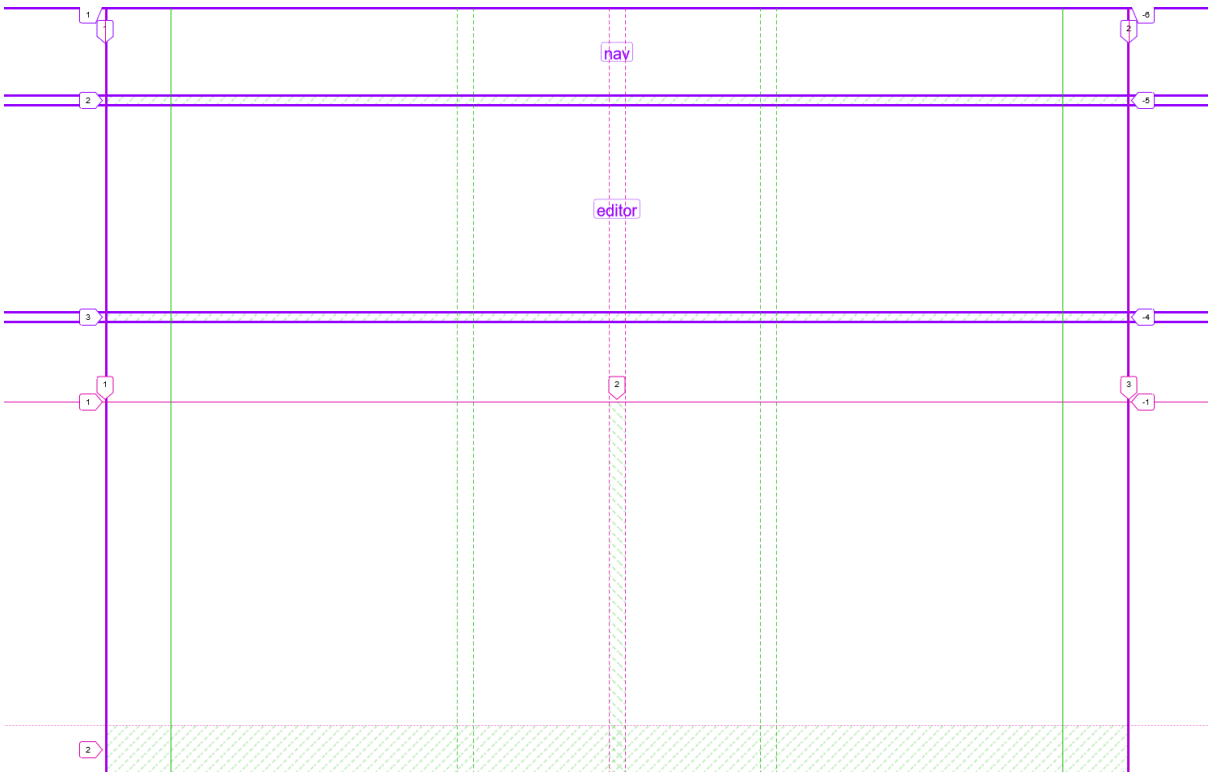
Prije izrade CSS razvojnog okvira, prvi korak je bio napraviti web-stranicu koja će se koristiti za pregled svih dostupnih komponenti u CSS razvojnom okviru. Također web-stranica će imati funkciju uređivanja implementiranih komponenti, što znači mijenjanje i odabir ponuđenih postavki mijenjat će izgled same komponente tako što će se primjenjivati predviđene klase na sve komponente. Neke od ponuđenih postavki koje korisnik može mijenjati su boja, radijus okvira ili pak stil komponente.

Dakle, struktura web-stranice sastoji se od navigacijskog izbornika, izbornika za uređivanje komponenti, glavnog dijela web-stranice koji služi za prikaz komponenti u grid sustavu, te podnožja. Struktura web-stranice izrađena je pomoću CSS grid modula za organiziranje sadržaja stranice. CSS grid modul novi je način kojim možemo jednostavno kontrolirati veličinu i poziciju blokova i sadržaja tih blokova. Grid modul je optimiziran za raspored u 2 dimenzije, što znači da se sadržaj stranice organizira i vertikalno i horizontalno. No prije uređivanja strukture u CSS grid modulu, potrebno je izraditi HTML strukturu web-stranice koja izgleda ovako:

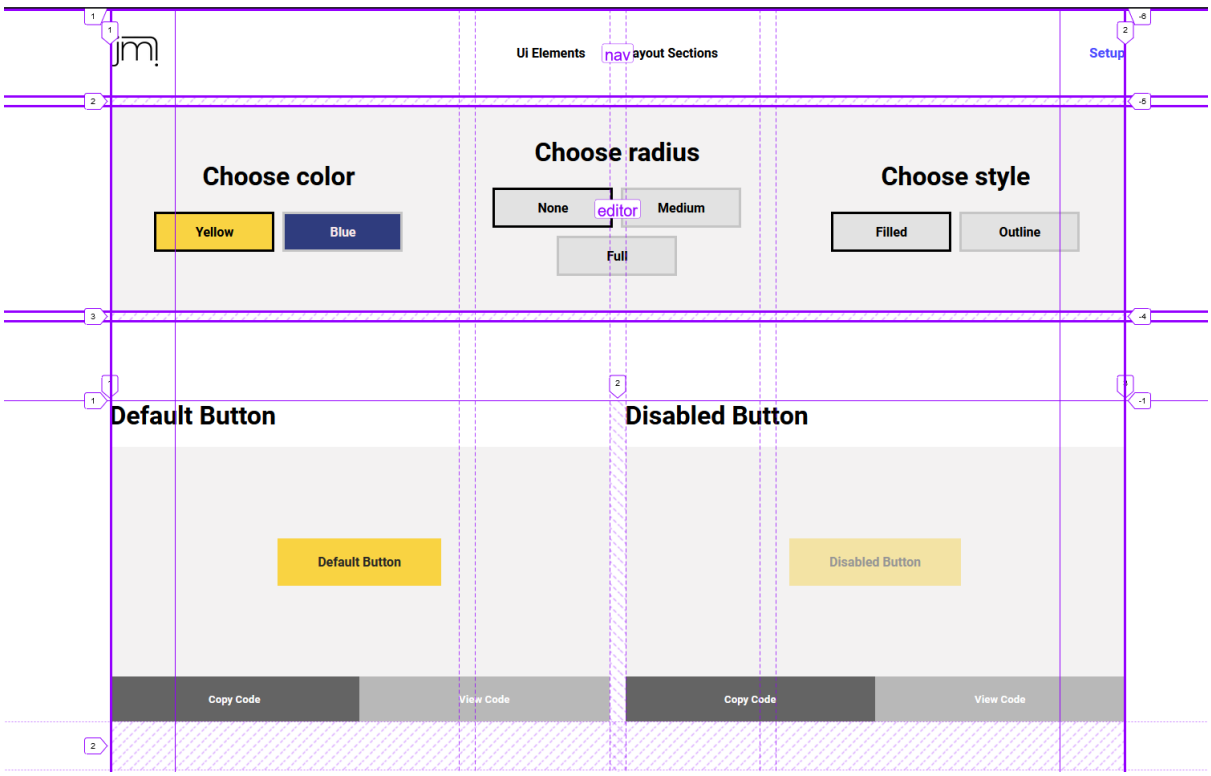
```
<html lang="en">
    <head>
        <title>Ui Elements</title>
        <link rel="stylesheet" href="css/style.css">
        <link rel="stylesheet" href="css/elementStyles.css">
    </head>
    <body>
    <div class="container">
        <div id="nav"> </div>
        <div id="editor"> </div>
        <div id="main"> </div>
        <div id="main-2"> </div>
        <div id="footer"> </div> </div>
    </body>
 </html>
```

Struktura HTML-a sastoji se od 5 sekcija kojima će se pomoću CSS grid modula odrediti veličina i pozicija na web-stranici. Pomoću CSS grid modula izrađena je mreža, te se svaki blok u mreži imenovao s parametrom grid-template-areas. U svaki blok smješta se pripadajući sadržaj korištenjem parametra grid-area, tako što se označuje HTML komponenta, te s parametrom grid-area podešava se i pripadajuće ime bloka u koji se sadržaj. Kod manjih rezolucija, blokovi mijenaju strukturu zbog responzivnog dizajna kako bi web-stranica bila pristupačna i na mobilnim uređajima.

```
.container {
        max-width: 1366px;
        margin: auto;
        padding-left: 50px;
        padding-right: 50px;
        display: grid;
        grid-template-areas:
            "nav"
            "editor"
            "content"
            "content-2"
            "footer";
        grid-template-columns:
auto;
        grid-template-rows:   110px
auto auto auto 20px;
        grid-row-gap: 10px;
        align-items: center;
        height: 100vh;
```



Slika 3. Vizualni prikaz mrežne strukture web-stranice

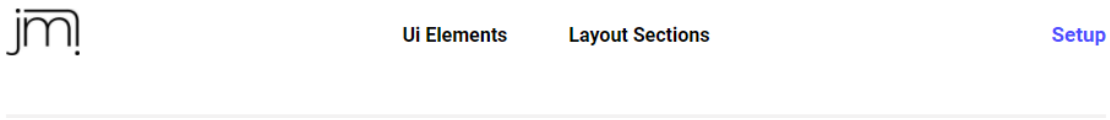


Slika 3. Vizualni prikaz mrežne strukture web-stranice s pripadajućim sadržajem

4.1.2. Izrada animacije navigacijskog izbornika

Za izradu animacije navigacijskog izbornika korišteni su javascript, te jQuery. Pritiskom na kategoriju "*Ui Elements*" ili "*Layout Sections*" web-stranica se automatski premješta na poziciju pripadajućeg sadržaja. Korištenjem medijskih upita (eng. *Media Query*) podešeno je otkrivanje hambuger gumba na rezoluciji manjoj od 1000 piksela, klikom na hambuger gumb pojavljuje se izbornik preko cijele visine i širine ekrana. Tranzicija pojavljivanja i sakrivanja mobilnog izbornika izrađena je pomoću jQuery javascript biblioteke.

```
$(document).ready(function () {
    $(".hamb-icon").click(function () {
        $(".hamb-icon").hide();
        $(".close-icon").delay(100).fadeIn("slow");
        $("#nav-mobile").fadeIn("slow");
        $(".menu-mobile").delay(300).fadeIn("slow");
    }); });
$(document).ready(function () {
    $(".close-icon").click(function () {
        $(".close-icon").hide();
        $(".close-icon").fadeOut("slow")
        $(".hamb-icon").fadeIn("slow");
        $("#nav-mobile").delay(500).fadeOut("slow");
        $(".menu-mobile").fadeOut("slow");
    }); });
```



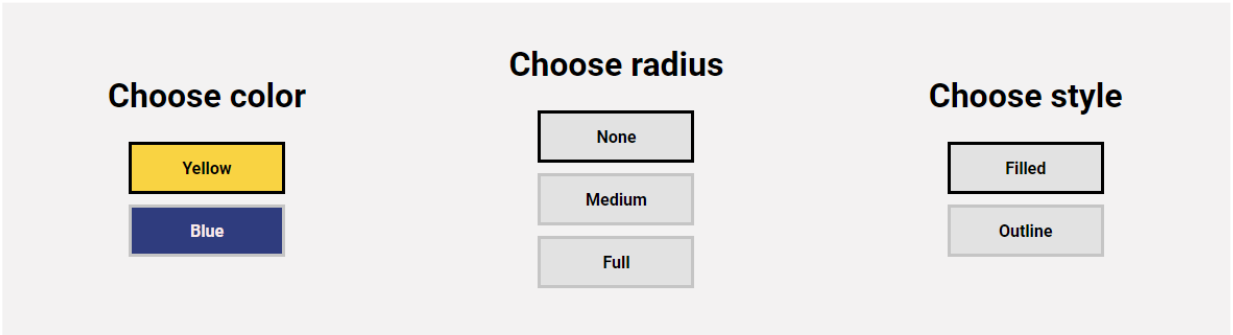
Slika 4. Navigacijski izbornik

Slika 5. Navigacijski izbornik za mobilne rezolucije

4.2. Funckija izbornika za uređivanje komponenti

Ideja izbornika za uređivanje komponenti je funkcija brzog pregleda svih mogućih varijatni te komponente koje su moguće unutar CSS razvojnog okvira. Izbornik se sastoji od tri bloka. U prvom bloku nalaze se radio gumbi koji omogućuju mijenjanje boje komponenti, odabirom na jedan od radio gumba mijenjaju se ili dodaju klase za određenu boju, u drugom bloku nalaze radio gumbi za odabir radijusa komponenti, a u trećem se nalaze gumbi za izbor stila komponenti.

```
<div id="editor">
   <div class="edit">
      <h1> Choose color</h1>
         <div class="editorRadioContainer">
            <label class="radioContainer">
  <input type="radio" name="color" value="color-c1"
onclick="changeColorClass()" checked="checked">
 <span class="checkmark yellow">Yellow</span> </label>
     <label class="radioContainer">
 <input type="radio"name="color" value="color-c2"
onclick="changeColorClass()">
  <span class="checkmark blue">Blue</span>
          </label>
        </div>
    </div>
```



Slika 6. Izbornik za uređivanje komponenti

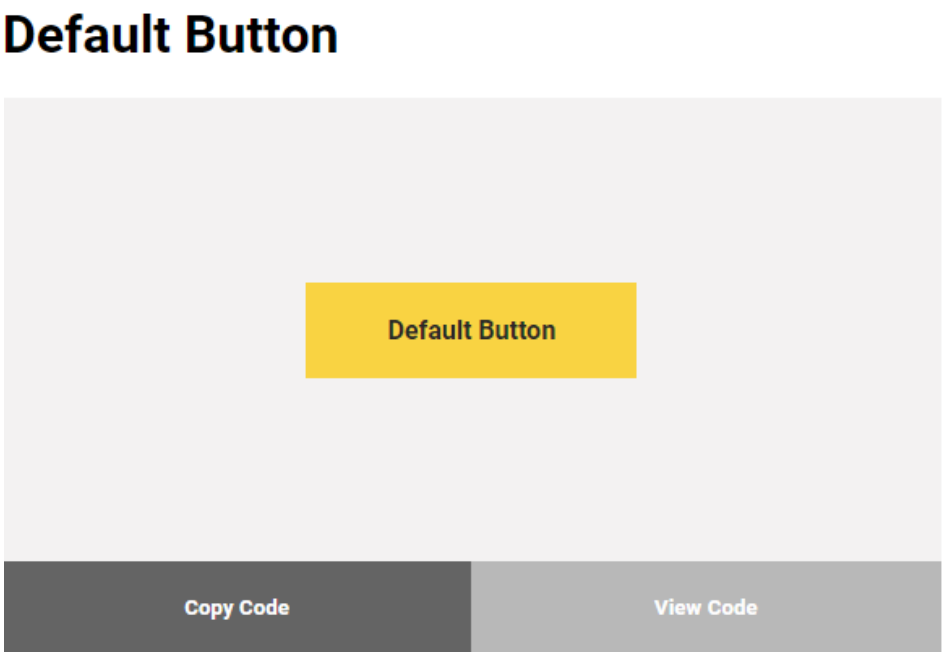
Dakle, izborom bilo koje od opcija, komponentama se dodaju ili mijenjanju pripadajuće klase koje određenim komponentama mijenaju izgled. Klase kojima se odabire boja, te koje korisnik može primijeniti na bilo koju komponentu HTML-a su “color-c1” (žuta boja), te “color-c2” (plava boja). Klase za odabir radijusa su “radius-r1” za radijus od 0 piksela, “radius-r2” za radijus od 8 piksela, te “radius-r3” za potpuni radijus. Također može se mijenjati i stil komponenti klasama *“filled”* i “*outline”,* te stanje aktivnosti komponenti klasom “*disabled*”.

Promjena ili dodavanje klase odvija se nakon klika na bilo koji od radio gumba koji se nalaze u izborniku. Nakon klika na radio gumb poziva se funckija za zamjenu klase određenog parametra, npr. “changeColorClass()” je funckija za promjenu klase “color”.

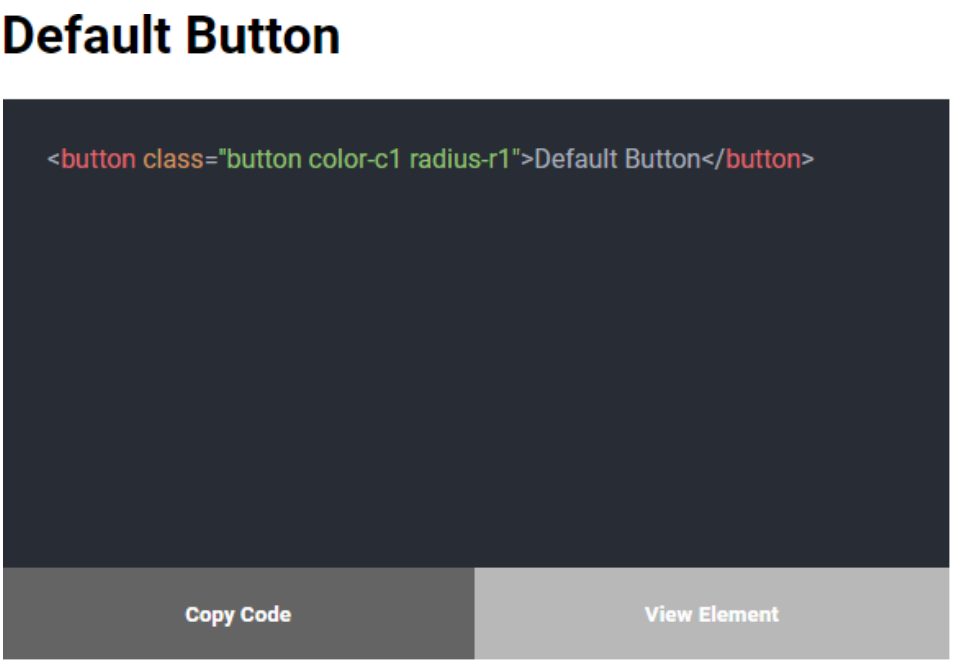
```
function changeColor(x, y) {
  for (var i = 0; i < x.length; i++) {
    if (x[i].checked) {
      for (var j = 0; j < y.length; j++) {
        if (y[j].classList.contains('color-c1')) {
     y[j].classList.replace('color-c1', x[i].value);
      } else if (y[j].classList.contains('color-c2')) {
     y[j].classList.replace('color-c2', x[i].value);
                }} }
 }}
function changeColorClass() {
    changeColor(selectedColor, buttons);
    changeColor(selectedColor, buttonsIcon);
    changeColor(selectedColor, radioButtons);
    changeColor(selectedColor, form);
    changeColor(selectedColor, sectionHero);
    changeColor(selectedColor, sectionFeature);
    changeColor(selectedColor, headingHero);
    changeColor(selectedColor, headingFeature);
    changeColor(selectedColor, paragraphHero);
    changeColor(selectedColor, card);
    convertHTML();
    updateCode();
}
```

4.3. Kopiranje i prikaz koda uređenih komponenti

Kako bi korisnik lakše naučio kako koristiti CSS razvojni okvir, na web-stranici je moguće kopirati izvorni kod svake komponente, te u izvornom kodu vidjeti koje klase se koriste za određene parametre. Isto tako kopiranjem već gotovih komponenti značajno se ubrazava razvoj web-stranice.



Slika 7. Pregled komponente na web-stranici



Slika 8. Izvorni kod komponente

Ispod svake komponente nalaze se dva gumba, pritiskom na gumb "*Copy Code*" korisnik može kopirati izvorni kod komponente, pritiskom na gumb "*View Code*" korisnik može pregledati izvorni kod, te može birati između pogleda na kod ili na komponentu.

Izvorni kod komponenti stavlja se unutar selektora <*pre*> i <*code*>, te javascript biblioteka highlight.js automatski prepoznaje o kojem se programskom jeziku radi, te prema tome prilagodi boju koda, te format i strukturu koda. Da bi kod bio formatiran, potrebno je pozvati funkciju javascript biblioteke *"hljs.highlightBlock".*

```
function updateCode() {
    document.querySelectorAll('pre code').forEach((block) => {
        hljs.highlightBlock(block);
    });
}
```

Kako bi se prikazao HTML izvorni kod komponente na web-stranici, potrebno je oko tog koda staviti <*code*> oznaku, no da bi <*code*> oznaka prikazivala HTML kod, a ne HTML komponentu renderiranu od strane internet preglednika, potrebno je HTML kod pisati pomoću specijalnih znakova, koji se prikazuju pomoću HTML-entiteta, npr. umjesto šiljaste zagrade *"<"*, koristi se znak *"<"* Upravo to radi funkcija *"convertHTML"* za svaku komponentu, tako što se selektira HTML kod komponente, te se zamijene oznake s specijalnim znakovima tako da internet preglednik ne renderira komponentu, već prikaže HTML kod.

```
function convertHTML() {
  var element =
document.getElementsByClassName("elementContent");
  var elementContent = [];
  var encodedStr = [];
  var codeContainer =
document.getElementsByClassName("codeContainer");
  for (var i = 0; i < element.length; i++) {
    elementContent[i] = element[i].innerHTML;
    encodedStr[i] = elementContent[i].replace(/[\u00A0-
\u9999<>\&]/gim, function (i) {
    return '&#' + i.charCodeAt(0) + ';'; });
    codeContainer[i].innerHTML = encodedStr[i].trim();
    }
}
```

Kopiranje izvornog koda za svaku zasebnu komponentu funkcionira tako da dodamo "EventListener" na sve gumbe "*Copy Code*". Pritiskom na gumb "*Copy Code*" poziva se funkcija "*copyHTML*" za određenu komponentu. Funkcijom "*createElement*" stvoreni je element teksualnog prostora u koji se privremeno stavlja određeni HTML kod komponente, sljedeći korak je kopiranje svog sadržaja tog tekstualnog prostora, nakon što je kopiranje obavljeno, tekstualni prostor se briše jer više nije potreban.

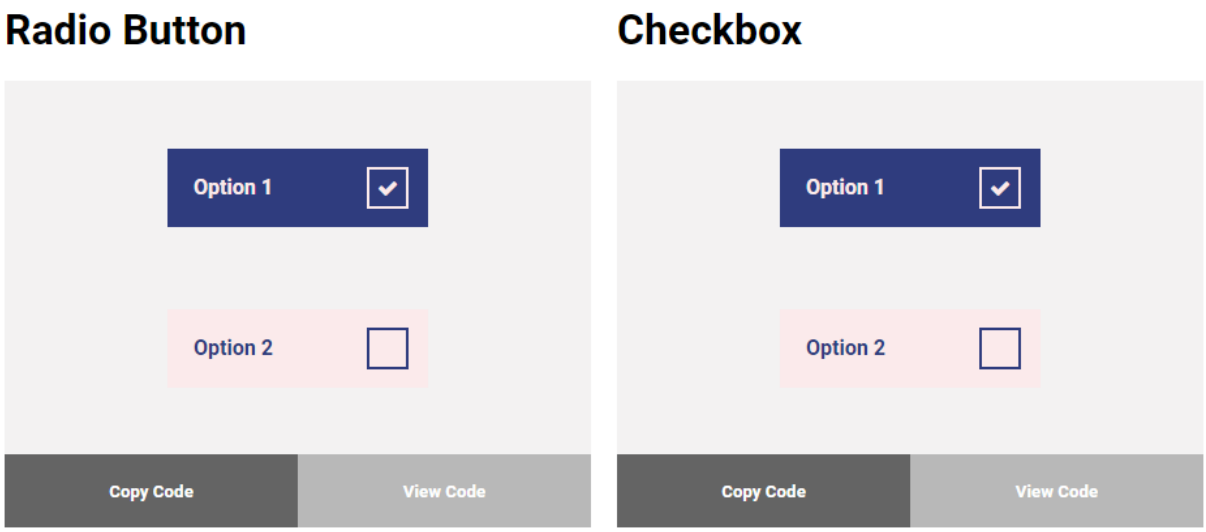
```
var copyBtn = document.getElementsByClassName("copyBtn");
for (var i = 0; i < copyBtn.length; i++) {
    copyBtn[i].addEventListener('click', copyHTML);
}
function copyHTML(e) {
    var textArea = document.createElement("textarea");
    textArea.setAttribute("id", "tempTextArea");
    textArea.value =
e.target.parentNode.parentNode.children[2].textContent;
    document.body.appendChild(textArea);

    //Remove whitespace from HTML code

 function trimTextArea() {
 var trimArea = $('#tempTextArea');
 trimArea.val($.trim(trimArea.val()).replace
(/\s*[\r\n]+\s*/g, '\n').replace(/(<[^\/][^>]*>)\s*/g, '$1')
 .replace(/\s*(<\/[^>]+>)/g, '$1')); }
    trimTextArea();

    textArea.select();
    document.execCommand("Copy");
    textArea.remove();
}
```

4.4. Komponente korisničkog sučelja CSS razvojnog okvira

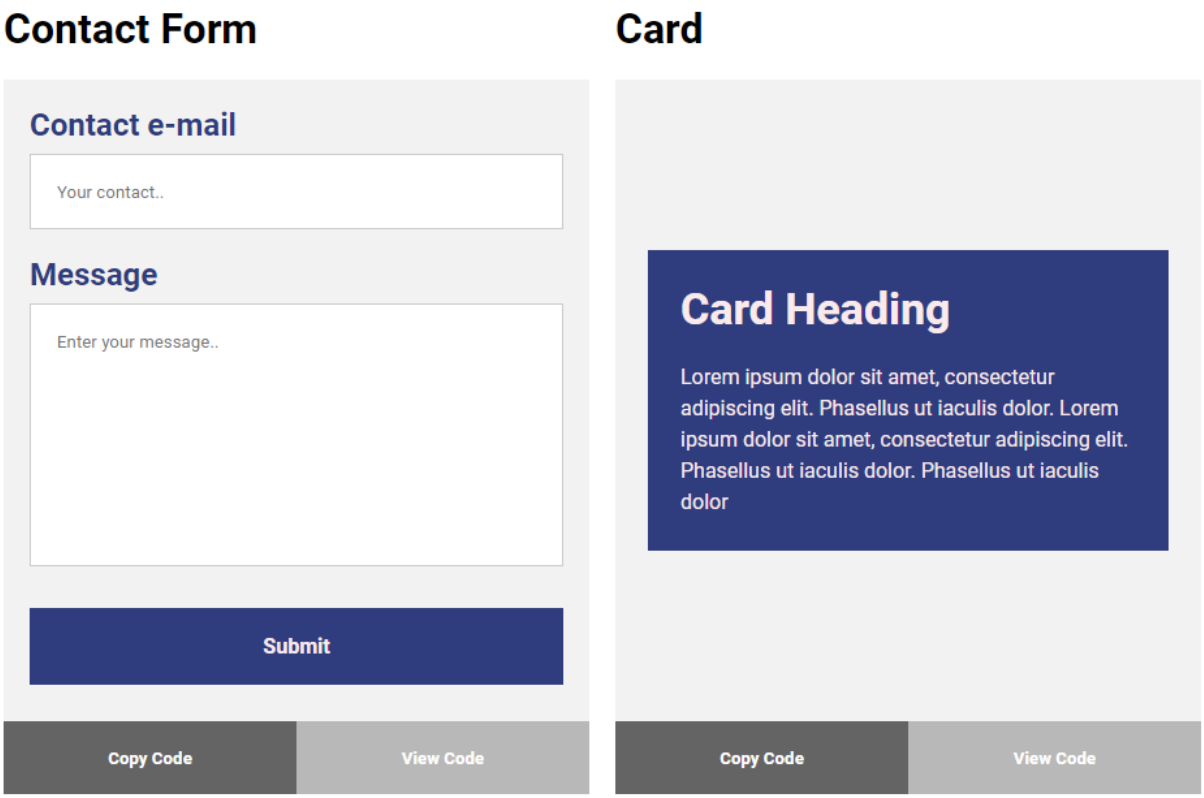
Ovaj CSS razvojni okvir sadržava nekoliko komponenti korisničkog sučelja koje će ubrzati razvoj web-stranice. Neke od tih komponenti su gumbi, gumbi se najčešće koriste za radnje poput klika ili dodira, te je obično označen pomoću teksta, ikone ili oboje.

Radio gumb [11] koji služi za odabir jedne od ponuđenih opcija. Checkbox koji korisniku omogućuju odabir jednu ili više od ponuđenih opcija. Kontaktna forma koja služi kao forma kojom korisnik može poslati svoje kontakt informacije i poruku adminu web-stranice, te kartice koje su najčešće male, te dolaze u obliku kvadrata ili pravokutnika. Kartice se najčešće sastoje od niza informacija poput slike, teksta, gumba itd. Kartice su odličan izbor za korisničko sučelje ukoliko želimo prikazati veliki broj informacija ili artikala na web-stranici.

CSS razvojni okvir sastoji se od nekoliko uređenih sekcija korisničkog sučelja kao što su navigacijski izbornik, sekcija za prikaz glavnog sadržaja, sekcija za prikaz značajki nekog objekta, predmeta, artikla itd., sekcije s mrežnom strukturom koja se dijeli u tri stupca, te broj redaka ovisi o veličini sadržaja.



Slika 9. Radio gumb i checkbox

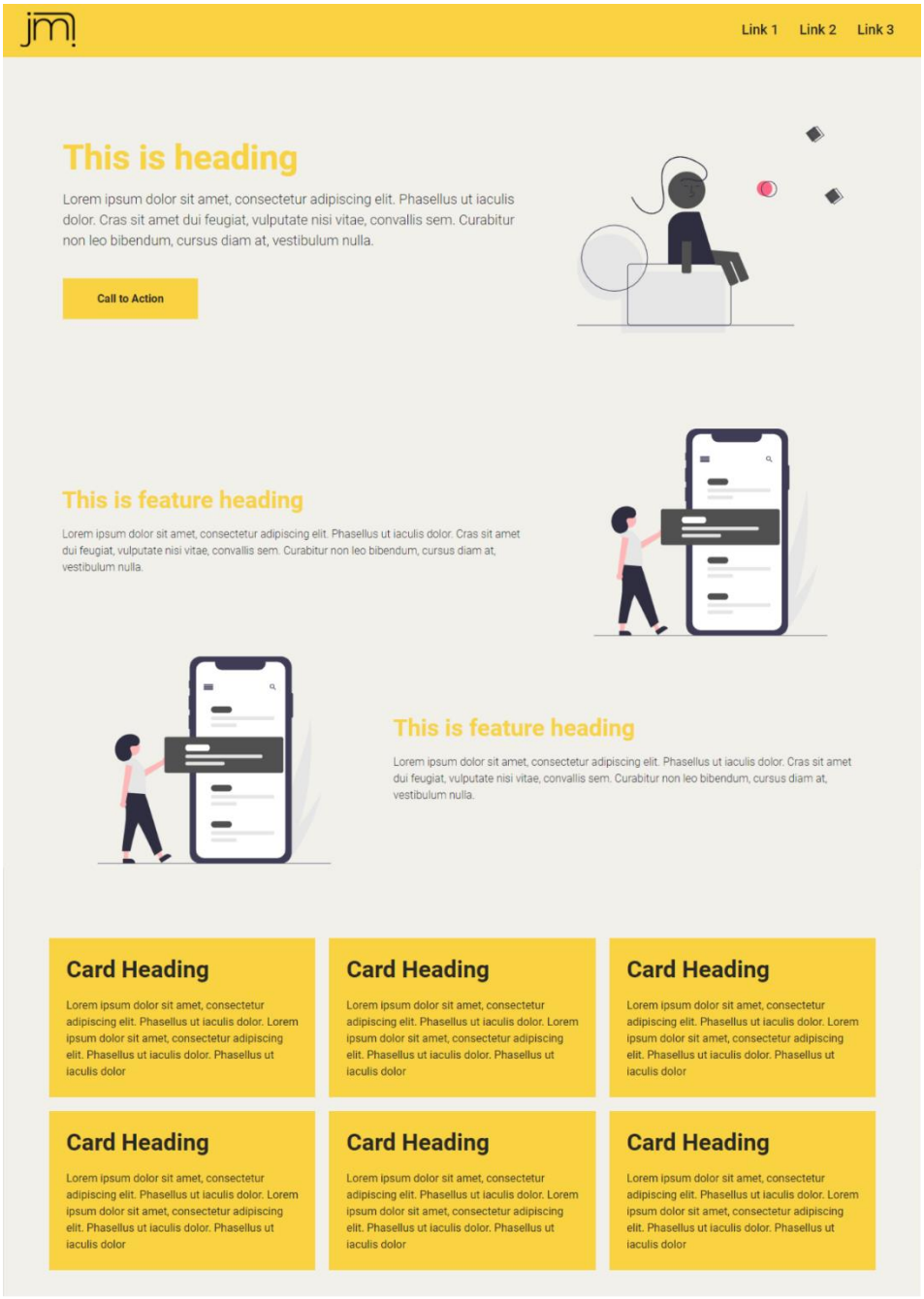


Slika 11. Kontakt forma i kartica

Slika 12. Sekcija za prikaz značajki

5. Rezultat korištenja gotovog rješenja

Korištenje CSS razvojnog okvira treba biti jednostavno i pristupačno svakom korisniku. Kako bi prikazao rezultat i mogućnosti ovog CSS razvojnog okvira izrađena je web-stranica samo korištenjem izvornog koda dostupnog na već spomenutoj web-stranici koja služi za prikaz svih komponenti. Za izradu sljedeće web-stranice potrebno je 5 minuta radnog vremena, čime je zaključeno da se korištenjem gotovih rješenja CSS razvojnih okvira znatno ubrzava i olakšava razvoj web-stranica.



Slika 10. Primjer razvijene web-stranice korištenjem komponenti CSS razvojnog okvira

6. Zaključak

U ovom završnom radu dokumentiran je proces izrade, te su opisane tehnologije i alati potrebni za izradu web-stranice i CSS razvojnog okvira.

CSS razvojni okvir uspješno je razvijen. Kako bi korisnik koristio razvojni okvir potrebno je preuzeti datoteku "*elementStyles.css*", te je implementirati u željeni HTML dokument. Također uspješno razvijena web-stranica korisniku pruža niz mogućnosti kao što je pregled svih dostupnih komponenti koje je moguće stilizirati uz pomoć razvojnog okvira. Korisnik, osim što ima pregled svih komponenti, ima opciju uređivanja tih komponenti, može promijeniti boju komponente, zaobljenost ruba komponente, te stil komponente. Promjenom izgleda komponente, mijenja se i kod svake komponente koji korisnik može kopirati, odnosno mijenja se klasa HTML komponente koja daje toj komponenti određeni izgled.

Razvojem projekta stekao sam nova znanja vezana uz CSS razvojne okvire, kako su strukturirani, na koji način funkcioniraju, kako se koriste, te koje su im mane i prednosti. Za izradu razvojnog okvira bitna stavka je bila odabrati dizajn komponenti koje će biti obuhvaćene razvojnim okvirem, zbog toga što je to jedan od čimbenika na temelju kojeg korisnik odabire razvojni okvir.

U Varaždinu, 28. rujna 2020.

_____

7. Literatura

1. Karlsson Jill, Responsive web design with CSS frameworks: http://uu.diva-portal.org/smash/get/diva2:721576/FULLTEXT01.pdf (pristupljeno 9. kolovoza 2020.)
2. Tech Terms :https://techterms.com/definition/framework (pristupljeno 9. kolovoza 2020.)
3. Shenoy Aravind, Prabhu Anirudh, CSS Framework Alternatives: https://books.google.hr/books?hl=hr&lr=&id=UJxQDwAAQBAJ&oi=fnd&pg=PP5&dq=css+framework&ots=8XJCAbq9la&sig=RVPYHmegC0alHLaaXdZ5RSasw4M&redir_esc=y#v=onepage&q=css%20framework&f=false (pristupljeno 8. kolovoza 2020.)
4. Laaziri Majida, Benmoussa, Khoulji Samira, Mohamed Larbi Kerbeb, Analyzing bootstrap and foundation front-end frameworks: a comparative study: https://www.researchgate.net/publication/330656530_Analyzing_bootsrap_and_foundation_font-end_frameworks_a_comparative_study (pristupljeno 10. kolovoza 2020.)
5. Jain Nilesh, Review of different responsive CSS front-end frameworks: http://www.rroij.com/open-access/review-of-different-responsive-css-frontend-frameworks.pdf (pristupljeno 10. kolovoza 2020.)
6. Nikhil, CSS Grid Layout vs CSS Frameworks: When To Use What?: https://www.lambdatest.com/blog/css-grid-layout-vs-css-frameworks/ (pristupljeno 11. kolovoza 2020.)
7. Kurtović Gorana, Uvod u HTML: https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c201_polaznik.pdf (pristupljeno 12. kolovoza 2020.)
8. Javascritp Language, Tutorialspoint: https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf (pristupljeno 15. kolovoza 2020.)
9. Javascript libraries, Webplatform: https://webplatform.github.io/docs/concepts/programming/javascript/libraries/ (pristupljeno 14. kolovoza)
10. Highlight.js: https://highlightjs.org/ (pristupljeno 10. kolovoza)
11. User Interface Elements, https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html (pristupljeno 15. kolovoza 2020.)

Popis slika

Slika 1. Vizualni prikaz tehnologije potrebne za izradu web-stranice....4
Slika 2. HTML struktura dokumenta....8
Slika 3. Vizualni prikaz mrežne strukture web-stranice....15
Slika 4. Vizualni prikaz mrežne strukture web-stranice s pripadajućim sadržajem....15
Slika 5. Navigacijski izbornik....16
Slika 6. Navigacijski izbornik za mobilne rezolucije....16
Slika 7. Izbornik za uređivanje komponenti....17
Slika 8. Pregled komponente na web-stranici....19
Slika 9. Izvorni kod komponente....19
Slika 10. Radio gumb i checkbox....22
Slika 11. Sekcija za prikaz značajki....23
Slika 12. Kontakt forma i kartica....23
Slika 13. Primjer razvijene web-stranice korištenjem komponenti CSS razvojnog okvira....24

Sveučilište Sjever

SVEUČILIŠTE SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi ~~tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu~~ pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

~~Ja,~~ JASMIN MIKOLAJ ~~*(ime i prezime) pod punom moralnom,*~~ materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/~~ica~~ završnog/~~diplomskog~~ (*obrisati nepotrebno*) rada pod naslovom IZRADA CSS RAZVOJNOG OKVIRA *(upisati naslov)* te da u ~~navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni~~ dijelovi tuđih radova.

Student/ica:
~~*(upisati ime i prezime)*~~
Jasmin Mikolaj
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se ~~realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.~~

Ja, JASMIN MIKOLAJ *(ime i prezime)* neopozivo izjavljujem da sam suglasan/~~na~~ s javnom objavom završnog/~~diplomskog~~ (*obrisati nepotrebno*) rada pod naslovom IZRADA CSS RAZVOJNOG OKVIRA *(upisati naslov)* čiji sam autor/~~ica.~~

Student/ica:
(upisati ime i prezime)
Jasmin Mikolaj
(vlastoručni potpis)