

Planiranje i razvoj informacijskog sustava za upravljanje obiteljskim financijama

Čuljak, Niko

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:145:397284>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-22**



Repository / Repozitorij:

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Preddiplomski studij smjer Poslovna informatika

Niko Čuljak

**Planiranje i razvoj informacijskog sustava za upravljanje
obiteljskim financijama**

Završni rad

Osijek, 2021.

Sveučilište Josipa Jurja Strossmayera u Osijeku
Ekonomski fakultet u Osijeku
Preddiplomski studij smjer Poslovna informatika

Niko Čuljak

**Planiranje i razvoj informacijskog sustava za upravljanje
obiteljskim financijama**

Završni rad

kolegij: Oblikovanje i implementacija informacijskog sustava

JMBAG: 0010217527

e-adresa: nculjak@efos.hr

mentor: prof.dr.sc. Josip Mesarić

Osijek, 2021.

Josip Juraj Strossmayer University of Osijek

Faculty of Economics in Osijek

Undergraduate Study of Business informatics

Niko Čuljak


**Planning and development of information system for managing
family finances**

Final paper

Osijek, 2021

IZJAVA

O AKADEMSKOJ ČESTITOSTI, PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA, SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je završni
(navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

Ime i prezime studenta/studentice: Niko Čuljak

JMBAG: 0010217527

OIB: 16249065429


e-mail za kontakt: niko.culjak@hotmail.com

Naziv studija: Poslovna informatika

Naslov rada: Planiranje i razvoj informacijskog sustava za upravljanje obiteljskim financijama

Mentor/mentorica rada: prof.dr.sc. Josip Mesarić

U Osijeku, 7. rujna 2021. godine

Potpis 

Planiranje i razvoj informacijskog sustava za upravljanje obiteljskim financijama

SAŽETAK

U mnogim se kućanstvima ne vodi formalna evidencija i planiranje svih obiteljskih troškova, odnosno raspoređivanje kućnog budžeta. Većinakućanstava nemaju plan budžeta. Zbog toga se događaju neplanirane potrošnje, zaduživanja ili odgode plaćanja obveza. Cilj je razviti jednostavnu web-aplikaciju koja će osigurati unos podataka o budžetu i plan potrošnje i uštede financijskih sredstava kako bi kućanstva lakše pratila visinu svojih prihoda i rashoda u neodređenom razdoblju te bi na temelju unesenih informacija mogli donijeti pretpostavku o budućim troškovima. Sustav će biti skiciran pomoću UML dijagrama, dok će sam program biti izrađen pomoću HTML-a, SASS-a i JavaScript programskog jezika. Programsko rješenje će funkcionirati s pretpostavkom da će kućanstva sav nepotrošeni novac na kraju mjeseca uštedjeti. Prvo će se pomoću HTML-a napraviti „kostur“ web-stranice, zatim će se pomoću SASS-a urediti dizajn stranice kako bi stranica izgledala atraktivnije i bila preglednija te će se na kraju pomoću JavaScript-a dodati funkcionalnost stranici kako bi bile moguće računske operacije i sortiranje rezultata. Ukoliko su rashodi za promatrani mjesec veći od prihoda, pretpostavka je da će se kućanstvo zadužiti kako bi podmirili životne troškove. Uneseni podaci će se vremenski pratiti i automatski će se izračunati suma prihoda i rashoda te iznos prosječnih prihoda i rashoda.

ključne riječi: prihod, rashod, raspoloživi dohodak, UML, JavaScript

Planning and development of information system for managing family finances

ABSTRACT

Many households do not keep formal records and plan all family expenses or allocate household budgets, in other words households do not have a budget plan. As a result, unplanned spending, borrowing, or deferred payments occur. The aim of the paper is to develop a simple web application that will provide data entry on the budget and a plan of spending and financial savings so that households can more easily track the amount of their income and expenditure over an indefinite period of time and make an assumption about future costs. The system will be modelled using UML diagrams while the program itself will be created using HTML, SASS and JavaScript programming language. The application will work on the assumption that households will save all unspent money at the end of the month. If the expenses for the observed month are higher than the income, it is assumed that the household will borrow to cover the cost of living. First, the "skeleton" of the web page will be created with the help of HTML, and then the design of the pages will be edited with the help of CSS to make the page look more attractive and more visible, and finally, with the help of JavaScript, add the functionality to the page to make computational operations and sorting of results possible. The entered data will be tracked over time and automatically calculate the sum of revenues and expenditures, but also what are the average revenues and expenditures.

Keywords: Income, expense, disposable income, UML, JavaScript

Table of Contents

1. Uvod	1
2. Tipični prihodi i rashodi kućanstava i njihova dinamika	1
3. Metodologija rada.....	3
4. Prikaz programskog rješenja pomoću grafičkog jezika UMLet.....	3
5. Razvoj aplikacije pomoću skriptnih programskih jezika	7
5.1. Osnovne karakteristike skriptnih jezika (Html, SASS, JavaScript)	7
5.1.1. Html.....	7
5.1.2. CSS.....	8
5.1.2.1. SASS	9
5.1.3. JavaScript	10
5.1.3.1. Typescript.....	11
5.2. Razvoj i testiranje predloženog rješenja.....	11
5.3. Implementacija	18
6. Zaključak.....	24
7. Literatura	25

1. Uvod

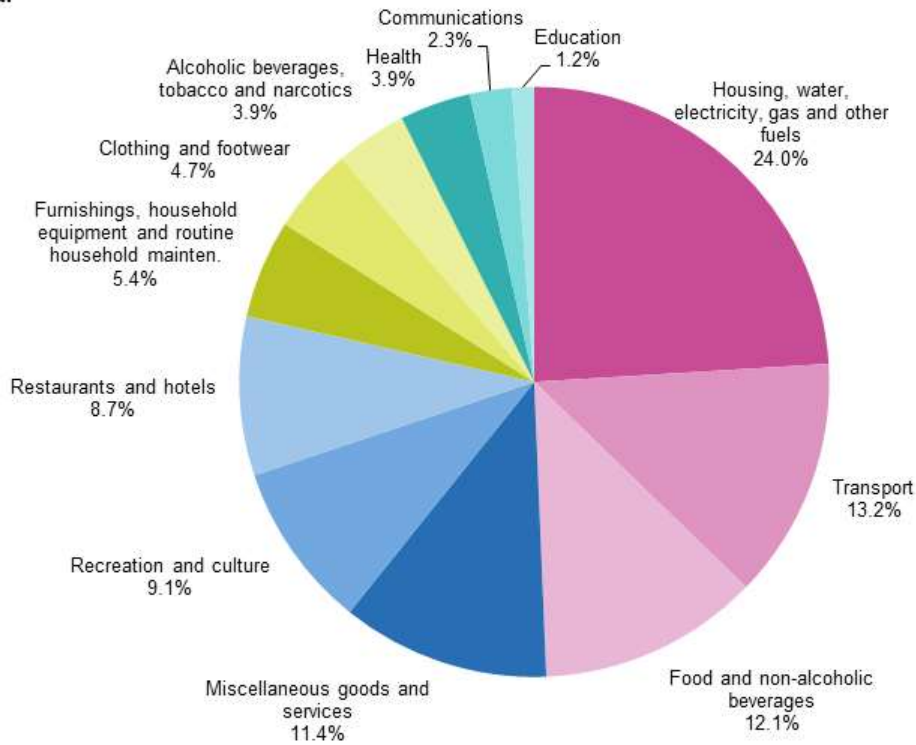
Dok je u prošlosti populacija trošila svoja novčana sredstva samo za egzistencijalne potrebe, s vremenom su ljudi stekli i potrebu za luksuznim dobrima i uslugama. Gotovo je nezamislivo da prosječni čovjek danas ne kupi novi automobil, udobniji namještaj, novo računalo ili potroši svoj dohodak na neko drugo opipljivo ili neopipljivo dobro ili uslugu. S razvojem tehnologije i pojavom novih trendova sve više raste i potrošnja na hobije ili zabavu u obliku glazbe, filmova ili videoigara. Čovjek ima tendenciju više trošiti s porastom raspoloživog dohotka što je ujedno i razlog navikavanja na viši standard i veće potražnje za luksuznim dobrima ili uslugama, no zanimljivo je da ne djeluje i suprotno. Naime, čovjek se teško odriče starih navika i s obzirom na smanjenje raspoloživog dohotka još će neko vremenom nastaviti trošiti svoj dohodak kako bi zadržao isti standard, iako mu možda ti isti životni troškovi sada prelaze granicu njegovih primitaka. Takve su situacije samo jedan od slučajeva u kojem se dolazi do zaduživanja ili odgode plaćanja obveza što je posljedica nepostojanosti financijskog plana. U ovom će se završnom radu razviti jednostavna aplikacija za praćenje primitaka i izdataka kućanstva koja će pohranjivati unose te na temelju njih izračunavati prosječne mjesečne primitke, izdatke i štednju.

2. Tipični prihodi i rashodi kućanstava i njihova dinamika

Za potpuno razumijevanje ovog završnog rada potrebno je najprije definirati sektor kućanstva i što sve unutar jednoga kućanstva može biti smatrano prihodom, odnosno rashodom. „Neko kućanstvo može sadržavati jednog ili više članova. Bitno je da članovi zajedno stanuju i troše svoj dohodak i bogatstvo za podmirenje životnih potreba“ (Borozan, 2019:20). Kućanstva primaju određenu količinu novca koja može biti zarađena (plaće) ili nezarađena (mirovine, stipendije, invalidnine, i sl.). Skup zarađenih i nezarađenih dohodaka čine osobni dohodak. Osobni dohodak nije količina novca s kojom neko kućanstvo raspolaže jer se od osobnog dohotka prvo trebaju odbiti porezi koja kućanstva plaćaju (porez na dohodak, porez na imovinu, porez na nasljedstvo i sl.). Tek se nakon odbijanja poreza dolazi do raspoloživog dohotka koju kućanstva mogu potrošiti ili uštedjeti, stoga vrijedi da je raspoloživi dohodak zbroj osobne potrošnje i privatne štednje. Prihodi unutar kućanstva sastavni su dijelovi

raspoloživog dohotka jer u programskom rješenju porezi neće biti dio rashoda kućanstva te će se prihodi gledati kao neto plaće i transferi. Rashodi se odnose na sve troškove kućanstva koji su prema COICOP-u (klasifikaciji osobne potrošnje prema namjeni) razvrstani u 12 skupina koje su prikazane na slici 1.

Household expenditure by consumption purpose - COICOP, EU-28, 2018, share of total



Source: Eurostat (online data code: nama_10_co3_p3)

eurostat 

Slika 1: *Grafički prikaz potrošnje kućanstva prema namjeni (Eurostat, 2019)*

Prema podacima Eurostata iz 2018. godine kućanstva članica Europske unije najviše svog raspoloživog dohotka potroše na stanarinu, režije i gorivo (24%), zatim na prijevoz (13,2%), hranu i bezalkoholno piće (12,1%), raznu robu i usluge (11,4%), rekreaciju i kulturu (9,1%), restorane i hotele (8,7%), opremanje stambenog prostora i rutinsko održavanje istog (5,4%), odjeću i obuću (4,7%), alkoholna pića i duhanske proizvode (3,9%), zdravstvo (3,9%), komunikacije (2,3%) te na obrazovanje (1,2%).

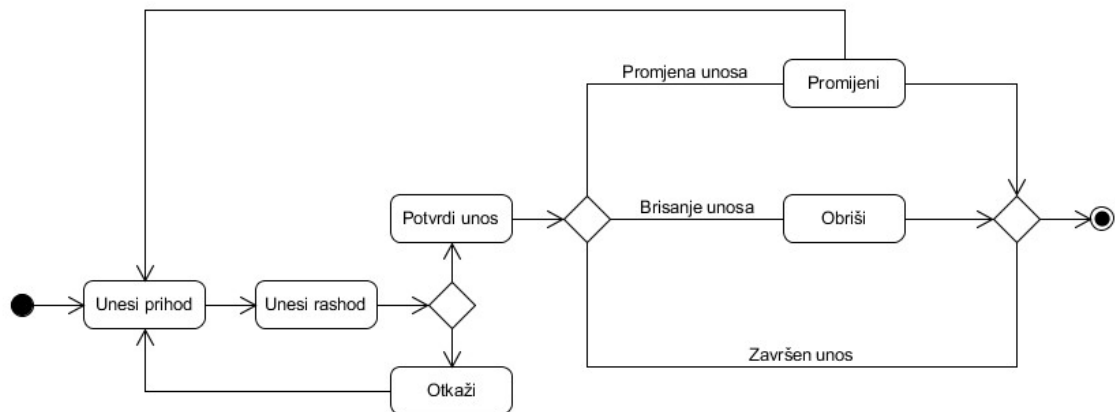
3. Metodologija rada

Cilj je ovog rada izrada pojednostavljenog prikaza odnosno modela i programskog rješenja za vođenje financija u kućanstvu. Grafički modeli koji sustav promatraju s više različitih aspekata izvest će se pomoću UML grafičkog jezika, a programsko rješenje razvit će se upotrebom HTML, SASS i JavaScript označiteljskih jezika.

Iako će program sam po sebi biti jednostavniji radi pokrivanja osnova i lakše objašnjavane funkcionalnosti, uvijek su otvorene opcije za nadogradnju samog programskog rješenja u budućnosti. Tijekom istraživanja više će se koristiti sekundarne metode prikupljanja podataka u odnosu na primarne, tako da će se u radu pretežito informacije prikupljati s internetskih izvora orijentiranih na educiranje o programskim jezicima i s platformi na kojima programeri međusobno dijele znanja i savjete o nadilaženju mogućih prepreka pri izradi programskog rješenja.

4. Prikaz programskog rješenja pomoću grafičkog jezika UMLet

Zbog kompliciranosti oblikovanja poslovnih sustava, te iste sustave je prvo potrebno modelirati. Pod modeliranjem se misli na kreiranje pojednostavljenog prikaza poslovnog sustava radi lakšeg razumijevanja i evaluacije. UML (*unified modelling language*) standardni je jezik za modeliranje sustava koji koristi grafički prikaz za izradu apstraktnog modela sustava. Dijagram slučajeva korištenja, dijagram klasa, dijagram objekata, dijagram aktivnosti i sekvencijski dijagram samo su neki od dijagrama UML-a, no u ovom radu se neće koristiti svi dijagrami. U ovom radu će se koristiti dijagram aktivnosti, sekvencijski dijagram i dijagram klasa.



Slika 2: *Dijagram aktivnosti (vlastita izrada)*

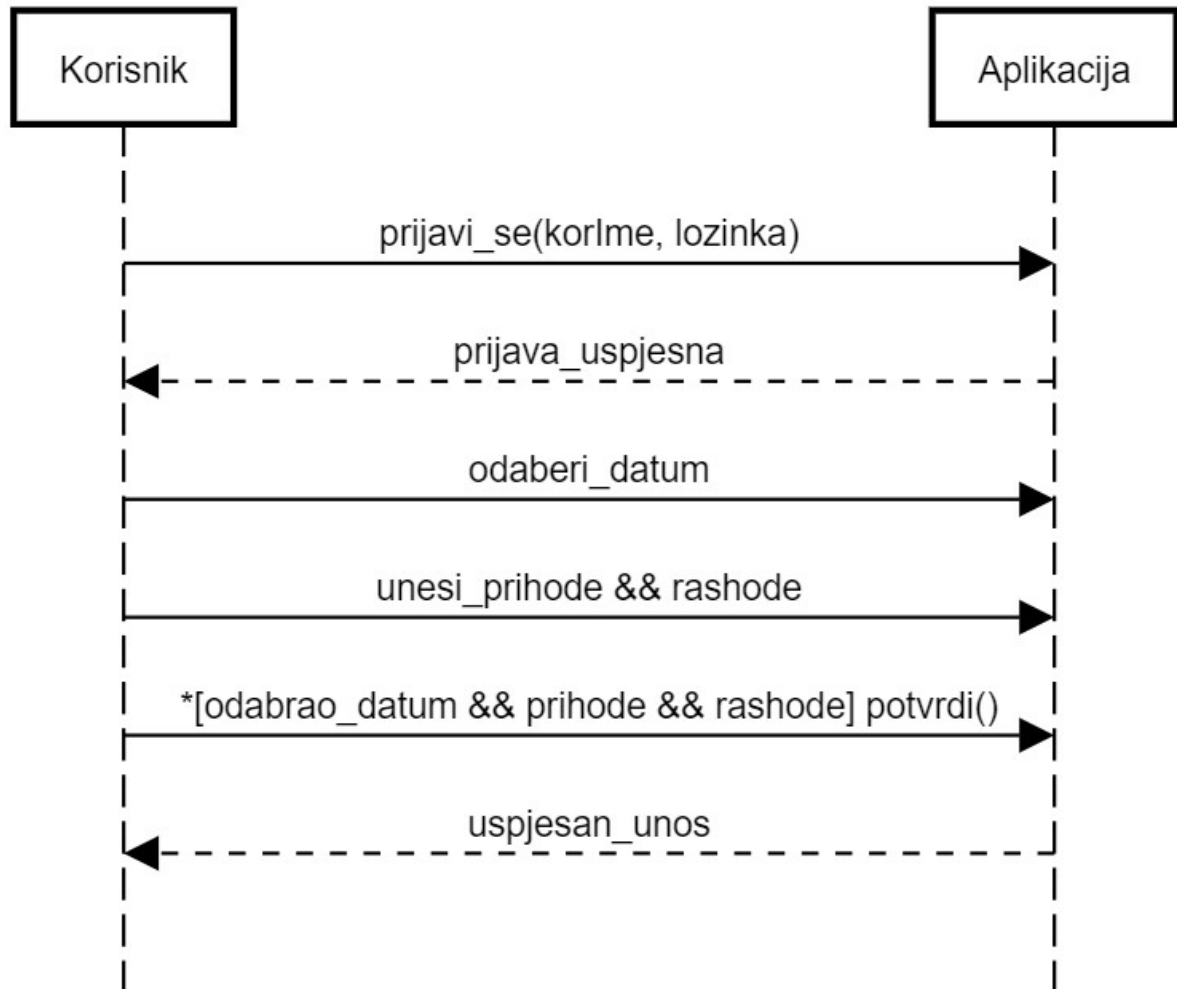
Na slici 2 prikazan je dijagram aktivnosti koji je skica za programsko rješenje koje će biti prikazano u slijedećem poglavlju. Gledajući s lijeva na desno (prateći strjelice) prikazan je prvo ispunjeni crni krug koji označava početno stanje u dijagramu, zatim prva akcija u kojoj korisnik unese svoje prihode te druga akcija u kojoj korisnik unosi svoje rashode. Nakon druge akcije na redu je odluka koja se prikazuje dijamentnim oblikom koji se ponaša kao *if*-uvjet u programskim jezicima. Odluka mora imati jedan ulazni i više izlaznih tokova. Pojavom odluke postoji alternativni slijed koji je ovisan o postavljenom uvjetu. U ovom slučaju postoje 2 uvjeta. Prvi je uvjet slučaj u kojem je unos otkazan te je korisnik vraćen na korak u kojem se unose prihodi, dok je drugi uvjet slučaj u kojem je unos potvrđen te se dolazi do još jedne odluke gdje korisnik ima izbor završiti unos, obrisati unos ili promijeniti unos koji korisnika opet vraća na prvi korak. Nakon odluke se svi razdvojeni tokovi spajaju ponovo u jedan tok koji ima više ulaznih, ali samo jedan izlazni tok koji u ovom slučaju vodi simbolu za završno stanje.



Slika 3: *Dijagram klasa (vlastita izrada)*

Slikom 3 prikazan je dijagram klasa. Pri vrhu dijagrama nalazi se naziv klase koji se piše velikim početnim slovom, nalazi se u sredini, podebljan je i piše se u jednini, u ovom slučaju naziv klase je Korisnik. Ispod naziva klase nalaze se atributi kojima se prvo zadaje vidljivost koristeći jedan od simbola za vidljivost. Vidljivost određuje tko može pristupiti atributima i operacijama neke klase. Simbol + označava mogućnost da objekti bilo koje klase mogu pristupiti informacijama. Nakon simbola za vidljivost slijedi naziv atributa koji se piše malim početnim slovom dok se ostale riječi u nazivu pišu velikim početnim slovom. Iza naziva slijedi tip atributa string koji je tekstualnog tipa i koristi se za prikaz znakova. Ispod unesenih atributa nalaze se operacije koje definiraju što neka klasa može izvršiti. Vidljivost i naziv operacije se pišu po istim pravilima kao i za attribute.

Sekvencijski dijagram



Slika 4: *Sekvencijski dijagram (vlastita izrada)*

Na slici 4 prikazan je sekvencijski dijagram koji prikazuje kronološki redoslijed događaja i operacija. Sudionici čine sekvencijski dijagram gdje je korisnik pošiljalac poruke, a aplikacija primatelj poruke. Strjelice označavaju poruke između sudionika koje predstavljaju komunikaciju između objekata. U ovom primjeru koriste se sinkrone i povratne poruke gdje se sinkrone skiciraju punom linijom, a povratne isprekidanom linijom. Sinkrone poruke se šalju kada pošiljalac očekuje povratnu poruku od primatelja i tek nakon povratne poruke se aktivnosti nastavljaju. Korisnik se prijavljuje unosom korisničkog imena i lozinke te dobiva povratnu poruku od aplikacije da je prijava uspješna. Nakon uspješne prijave korisnik odabire datum te unosi prihode i rashode, gdje se razmaci pišu donjom povlakom te se veznik „i“ piše simbolima „&&“. Simbol „*“ označava uvjet koji mora biti zadovoljen kako bi se izvršila akcija, u ovom slučaju potvrdi unos datuma, prihoda i rashoda nakon koje dolazi povratna poruka kako je unos uspješan.

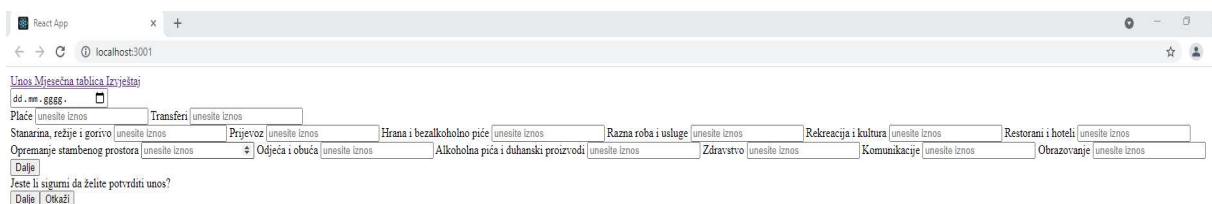
5. Razvoj aplikacije pomoću skriptnih programskih jezika

U ovom će se poglavlju prvo ukratko objasniti osnovne karakteristike korištenih skriptnih jezika i njihova uloga u samom razvojnom procesu; prikazat će se izgled aplikacije, objasniti će se na koji način aplikacija funkcionira te će se na kraju i implementirati.

5.1. Osnovne karakteristike skriptnih jezika (Html, SASS, JavaScript)

5.1.1. Html

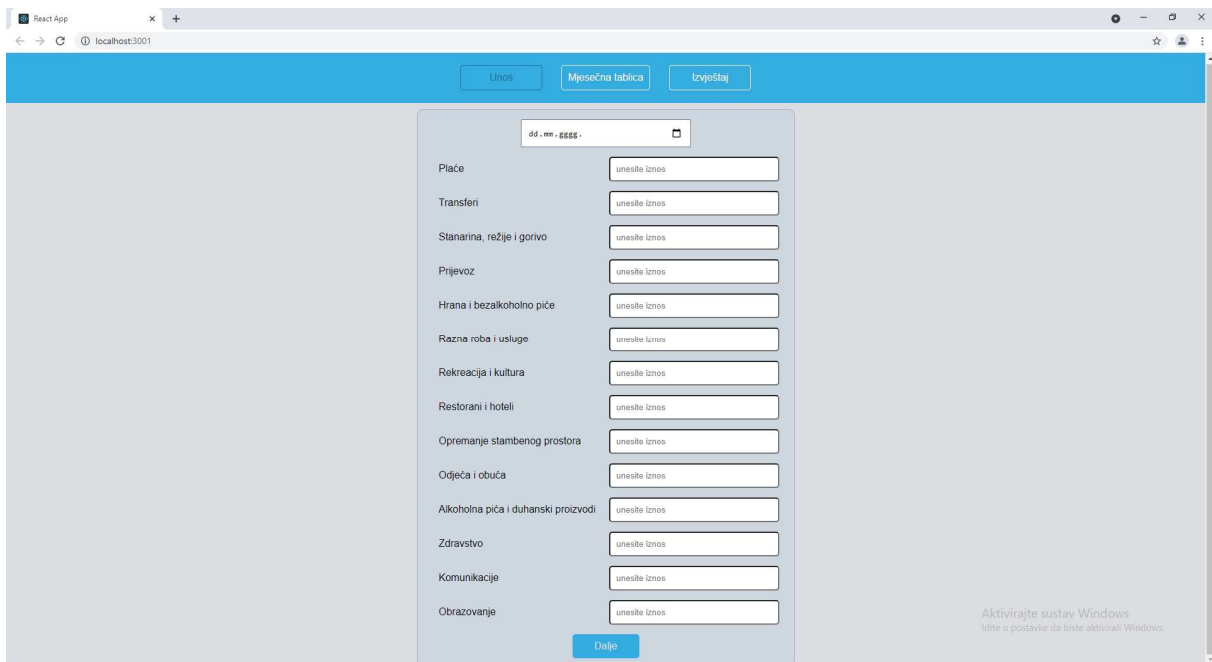
HTML (*Hyper Text Markup Language*) najosnovniji je element za izradu web-stranice. On definira sam „kostur“ web-stranice, točnije definira strukturu i elemente web-stranice. Suprotno popularnom vjerovanju, HTML nije programski jezik jer koristeći samo HTML nije moguće izvesti ni najjednostavnije računske operacije. On opisuje hipertekstualne dokumente i upućuje web-preglednik kako te dokumente prikazati. Dokumenti moraju biti prikazani isto koristeći bilo koji web-preglednik te ih je moguće povezati međusobno u uređenu hijerarhijsku strukturu. HTML element se razlikuje od drugog teksta unutar dokumenta pomoću „tagova“, koji se sastoje od imena elementa napisanog između znakova „<“ i „>“. Ime elementa unutar taga nije osjetljivo na velika i mala slova. Može se pisati velikim, malim ili kombinacijom velikih i malih slova. HTML dokument je uvijek potrebno definirati s otvarajućim i zatvarajućim <HTML> tagom, <head> tagom unutar kojeg je i <title> tag te <body> tagom. Unutar <head> taga uobičajeno se nalaze metapodatci (podatci o podacima), stil te, kao što je već spomenuto, <title> tag. <Title> tag definira naslov dokumenta. Naslov mora biti tekstualan i prikazuje se na naslovnoj traci preglednika ili na kartici stranice. <Body> tag određuje tijelo dokumenta i sadrži sve paragrafe, linkove, slike, naslove, tablice, liste i sve ostalo što čini tijelo dokumenta.



Slika 5: Prikaz HTML dokumenta bez CSS-a (vlastita izrada)

5.1.2. CSS

CSS (*Cascading Style Sheets*) je stilski jezik koji se koristi za uređivanje HTML dokumenta i koji može uštediti puno vremena jer može kontrolirati dizajn više web-stranica odjednom. CSS se koristi za definiranje stilova web-stranica, uključujući dizajn, izgled i varijacije prikaza za različite uređaje i veličine zaslona. Definicije stilova obično se spremaju u vanjske .css datoteke jer promjenom te datoteke automatski se mijenja izgled cijelog web-sjedišta.

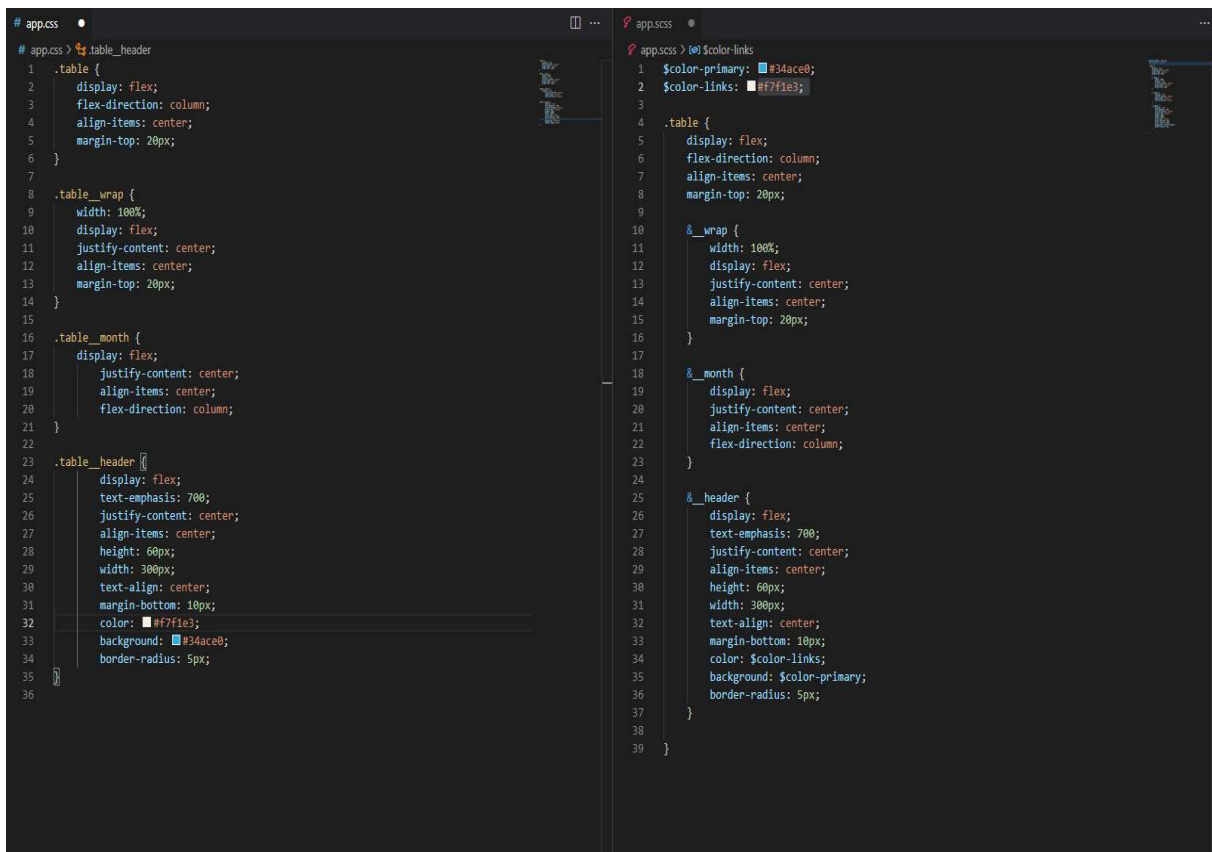


Slika 6: Prikaz HTML dokumenta sa CSS-om (vlastita izrada)

Na slici 5 prikazan je HTML dokument bez primjene .css datoteke, dok se na slici 6 vidi kako ta ista stranica izgleda s .css datotekom. Dodavanjem vanjske .css datoteke sve 3 stranice koje su vidljive na navigacijskom baru sada imaju isti dizajn. Stil dokumenta je moguće formirati i bez vanjske .css datoteke dodavanjem <style> taga unutar HTML dokumenta koji može biti vezan za npr. sve paragrafe unutar dokumenta ili za svaki paragraf posebno.

5.1.2.1. SASS

SASS (*Syntactically Awesome Style Sheets*) je skriptni jezik koji je programska ekstenzija CSS-a. Kompatibilan je sa svim verzijama CSS-a te pruža sve mogućnosti koje nudi i sam CSS uz još neke dodatne mogućnosti kao što su korištenje klasa za definiranje boja ili ugnježdivanje (eng. nesting) radi jednostavnijeg pisanja i preglednijeg izgleda dokumenta.



```
# app.css
# app.css > .table_header
1 .table {
2   display: flex;
3   flex-direction: column;
4   align-items: center;
5   margin-top: 20px;
6 }
7
8 .table_wrap {
9   width: 100%;
10  display: flex;
11  justify-content: center;
12  align-items: center;
13  margin-top: 20px;
14 }
15
16 .table_month {
17   display: flex;
18   justify-content: center;
19   align-items: center;
20   flex-direction: column;
21 }
22
23 .table_header {
24   display: flex;
25   text-emphasis: 700;
26   justify-content: center;
27   align-items: center;
28   height: 60px;
29   width: 300px;
30   text-align: center;
31   margin-bottom: 10px;
32   color: #f7f1e3;
33   background: #34ace0;
34   border-radius: 5px;
35 }
36

app.scss
app.scss > $color-links
1 $color-primary: #34ace0;
2 $color-links: #f7f1e3;
3
4 .table {
5   display: flex;
6   flex-direction: column;
7   align-items: center;
8   margin-top: 20px;
9
10  @wrap {
11    width: 100%;
12    display: flex;
13    justify-content: center;
14    align-items: center;
15    margin-top: 20px;
16  }
17
18  @month {
19    display: flex;
20    justify-content: center;
21    align-items: center;
22    flex-direction: column;
23  }
24
25  @header {
26    display: flex;
27    text-emphasis: 700;
28    justify-content: center;
29    align-items: center;
30    height: 60px;
31    width: 300px;
32    text-align: center;
33    margin-bottom: 10px;
34    color: $color-links;
35    background: $color-primary;
36    border-radius: 5px;
37  }
38
39 }
```

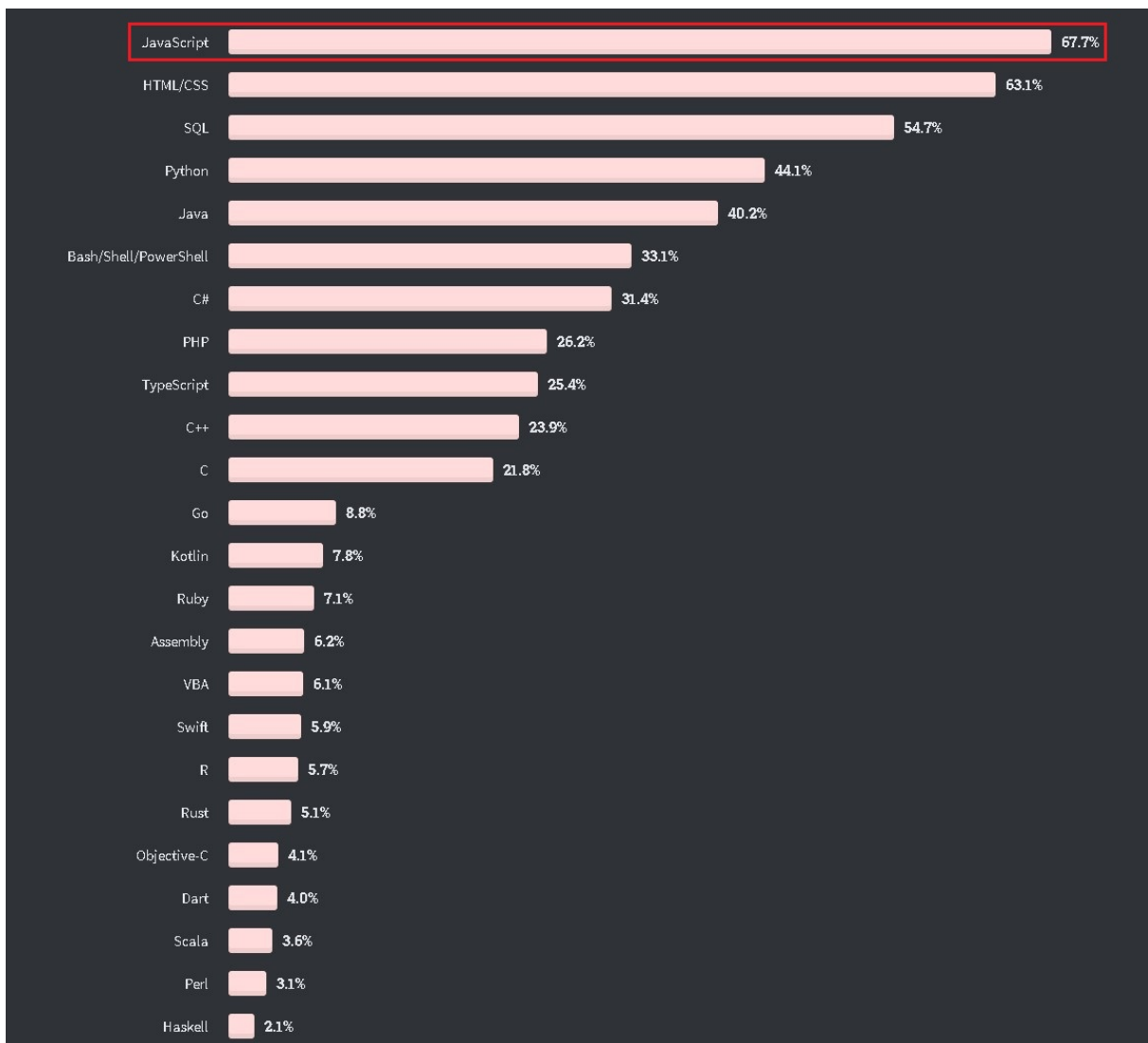
Slika 7: Prikaz CSS i SASS koda (vlastita izrada)

Slikom 7 prikazana je usporedba koda pisanog u CSS-u koji je prikazan na lijevoj strani i koda pisanog u SASS-u koji je prikazan na desnoj strani. Gledajući kod vidljive su prethodno spomenute razlike gdje je u SASS-u moguće koristiti klase za definiranje boja elemenata te su klase ugnježdene unutar drugih klasa. Važno je napomenuti kako bi stranica izgledala isto neovisno o korištenom dokumentu te se na prvi pogled možda pričinjava kako neke prevelike razlike gotovo pa i nema, ali kod pisanja više linija koda lakše je pozivati klase nego tražiti

kodove za boje i jednostavniji je pronalazak željenih elemenata ukoliko je potrebno izvršiti neku izmjenu.

5.1.3. JavaScript

JavaScript je skriptni programski jezik pomoću kojega se mogu implementirati složene značajke na web-stranice. Pomoću JavaScripta moguće je varijablama dodjeljivati vrijednosti, izvoditi računske operacije, dinamički ažurirati sadržaj, upravljati multimedijom, animirati slike, odnosno upravljati sadržajem na web-stranici. Zbog njegove jednostavnosti, lake dostupnosti i širokog izbora mogućnosti je stekao ogromnu popularnost među developerima i prema podacima ankete Stack Overflowa iz 2020. godine proglašen je najkorištenijim programskim, skriptnim ili *markup* jezikom.



Slika 8: Rezultati ankete Stackoverflow-a iz 2020. godine (Stackoverflow, 2020)

5.1.3.1. *Typescript*

Typescript je programski jezik koji je strogi sintaksnipodskup JavaScripta. Razvijen i održavan je od strane Microsofta s ciljem ranog prepoznavanja i ukazivanja na pogreške u samom kodu kako bi bila povećana učinkovitost pisanja koda. Omogućeno je statičko pisanje koda te je ponuđeno automatsko nadopunjavanje usred pisanja koda.

5.2. Razvoj i testiranje predloženog rješenja

U ovom se radu za razvoj rješenja koristio Visual Studio Code s ekstenzijom za *live server* koji služi kao privremena baza podataka za pohranjivanje unesenih informacija. Kod neće biti prikazan u cijelosti, već će biti prikazani neki od primjera za određene dijelove dok će cijeli kod biti dostupan na linku <https://github.com/NikoCuljak?tab=repositories>.

```
<div className="nav">
  <NavLink to="/" exact className="nav_link" activeClassName="active_link"> Unos </NavLink>
  <NavLink to="/monthly-table" exact className="nav_link" activeClassName="active_link"> Mjesečna tablica </NavLink>
  <NavLink to="/report" exact className="nav_link" activeClassName="active_link"> Izvještaj </NavLink>
</div>
```

Slika 9: Kod za navigacijski izbornik (vlastita izrada)

Na slici 9 prikazan je kod za navigacijski izbornik u HTML-u gdje je vidljiv div kontejner unutar kojeg se nalaze linkovi na navigacijskom izborniku. NavLink elementom stranice su međusobno povezane i klikom miša unutar web-preglednika omogućeno je prebacivanje s jedne stranice na drugu. NavLink automatski dodaje stil trenutno otvorenoj stranici te joj mijenja boju određenu u .scss datoteci kako bi se prikazalo koja je stranica trenutno otvorena.

```

.nav {
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 20px;
  background-color: $color-primary;
  &__link {
    color: $color-links;
    border: 1px solid $color-links;
    border-radius: 5px;
    margin: 0 15px;
    padding: 10px;
    min-width: 130px;
    text-align: center;
    text-decoration: none;
    cursor: pointer;

    &:hover {
      color: $color-hover;
      border-color: $color-hover;
    }
  }
}

```

Slika 10: *Stil navigacijskog izbornika (vlastita izrada)*

Slikom 10 prikazan je dio koda u .scss datoteci kojim je stiliziran navigacijski izbornik. Kao što se može uočiti na slici 9, div kontejneru i NavLink elementima dodane su klase nav i nav__link te se unutar .scss datoteke njima odredila pozicija, boja, širina, okvir, poravnanje i margina. Hover je dodan da se promijeni boja i boja okvira linka ukoliko se pokazivač miša postavi na link.

```
const [active, setActive] = useState(false);
const [date, setDate] = useState<string>(moment().format('DD/MM/YYYY'));
const [salary, setSalary] = useState<number | undefined>();
const [transfers, setTransfers] = useState<number | undefined>();
const [outcome1, setOutcome1] = useState<number | undefined>();
const [outcome2, setOutcome2] = useState<number | undefined>();
const [outcome3, setOutcome3] = useState<number | undefined>();
const [outcome4, setOutcome4] = useState<number | undefined>();
const [outcome5, setOutcome5] = useState<number | undefined>();
const [outcome6, setOutcome6] = useState<number | undefined>();
const [outcome7, setOutcome7] = useState<number | undefined>();
const [outcome8, setOutcome8] = useState<number | undefined>();
const [outcome9, setOutcome9] = useState<number | undefined>();
const [outcome10, setOutcome10] = useState<number | undefined>();
const [outcome11, setOutcome11] = useState<number | undefined>();
const [outcome12, setOutcome12] = useState<number | undefined>();
```

Slika 11: *Definiranje varijabli (vlastita izrada)*

Važan je korak definiranje varijabli kao što je prikazano slikom 11. Za definiranje varijabli u JavaScriptu koriste se dvije naredbe: `let` i `const`. Varijablama je potrebno dodijeliti naziv, tip i vrijednost. Vidljivo je kako su svim varijablama koje se odnose na prihode i rashode dodijeljen *tip number* (broj) jer će korisnici unositi numeričke varijable za iznose plaća, transfera i svih 12 skupina prema klasifikaciji osobne potrošnje prema namjeni. Nijednoj numeričkoj varijabli nije dodana vrijednost jer sami korisnici unose vrijednost preko obrasca. Date je jedina varijabla kojoj je dodijeljen tip `string` jer se ne radi o numeričkoj varijabli i dodan je format ('DD/MM/YYYY') kako bi nadnevak započeo prvo s danom u mjesecu, zatim mjesecom te u konačnici godinom.

```

<div className="form">
  <div className="form_date">
    <input type="date" required onChange={handleDateChange} value={date} />
  </div>

  <div className="form_income grid">
    <label htmlFor="place"> Plaće </label>
    <input type="number" id="place" placeholder="unesite iznos" onChange={handleSalaryChange} value={salary} />
    <label htmlFor="transferi"> Transferi </label>
    <input type="number" id="transferi" placeholder="unesite iznos" onChange={handleTransfersChange} value={transfers} />
  </div>

  <div className="form_outcome grid">
    <label htmlFor="stanarina"> Stanarina, režije i gorivo </label>
    <input type="number" id="stanarina" placeholder="unesite iznos" onChange={handleOutcome1Change} value={outcome1} />
    <label htmlFor="prijevoz"> Prijevoz </label>
    <input type="number" id="prijevoz" placeholder="unesite iznos" onChange={handleOutcome2Change} value={outcome2} />
    <label htmlFor="hrana"> Hrana i bezalkoholno piće </label>
    <input type="number" id="hrana" placeholder="unesite iznos" onChange={handleOutcome3Change} value={outcome3} />
    <label htmlFor="razno"> Razna roba i usluge </label>
    <input type="number" id="razno" placeholder="unesite iznos" onChange={handleOutcome4Change} value={outcome4} />
    <label htmlFor="rekreacija"> Rekreacija i kultura </label>
    <input type="number" id="rekreacija" placeholder="unesite iznos" onChange={handleOutcome5Change} value={outcome5} />
    <label htmlFor="restorani"> Restorani i hoteli </label>
    <input type="number" id="restorani" placeholder="unesite iznos" onChange={handleOutcome6Change} value={outcome6} />
    <label htmlFor="stan"> Opremanje stambenog prostora </label>
    <input type="number" id="stan" placeholder="unesite iznos" onChange={handleOutcome7Change} value={outcome7} />
    <label htmlFor="odjeca"> Odjeca i obuća </label>
    <input type="number" id="odjeca" placeholder="unesite iznos" onChange={handleOutcome8Change} value={outcome8} />
    <label htmlFor="alkohol"> Alkoholna pića i duhanski proizvodi </label>
    <input type="number" id="alkohol" placeholder="unesite iznos" onChange={handleOutcome9Change} value={outcome9} />
    <label htmlFor="zdravstvo"> Zdravstvo </label>
    <input type="number" id="zdravstvo" placeholder="unesite iznos" onChange={handleOutcome10Change} value={outcome10} />
    <label htmlFor="komunikacije"> Komunikacije </label>
    <input type="number" id="komunikacije" placeholder="unesite iznos" onChange={handleOutcome11Change} value={outcome11} />
    <label htmlFor="obrazovanje"> Obrazovanje </label>
    <input type="number" id="obrazovanje" placeholder="unesite iznos" onChange={handleOutcome12Change} value={outcome12} />
  </div>

```

Slika 12: Kod za obrazac (vlastita izrada)

Nakon definiranja varijabli slijedi izrada obrasca. Potrebno je napraviti ukupno 15 mjesta za unos od kojih će prvi biti unos datuma popraćen s plaćom, transferima i svim troškovima. Mjesta za unos vrijednosti definirani su tagom `<input>`. Svakom unosu se dodaje tip unosa koji će za sve biti broj osim za datum, kojem je priložen `type="date"`. Unutar svakog unosa za brojeve dodan je `placeholder="unesite iznos"` što je moguće vidjeti na slici 17. Svakom inputu dodan je id preko kojeg su povezani s odgovarajućim labelima te su im dodane prethodno definirane varijable.

Na slici 13 vidljiv je korišten stil za obrazac kojim je on početno postavljen na sredinu ekrana te su stavke stavljene unutar okvira koji se bojom razlikuje od ostatka stranice. Zaobljeni su rubovi okvira, definirane su širine unosa i rubova, stilizirano je dugme te je zadano ukoliko pokazivač miša pokaže na dugme da će dugme promijeniti svoju pozadinsku boju i boju slova kroz 0.3 sekunde.

```

.form {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 600px;
  background-color: $color-form;
  margin-top: 10px;
  border: 1px solid $color-form-border;
  border-radius: 10px;

  &__outcome {
    margin-bottom: 15px;
  }
}

.grid {
  display: grid;
  grid-template-columns: 1fr 1fr;
}

input, label {
  min-width: 270px;
  margin: 0 5px;
  margin-top: 15px;
  padding: 10px;
}

button {
  margin-bottom: 10px;
  background-color: $color-button;
  padding: 10px 30px;
  border-radius: 5px;
  transition: background-color 0.3s ease-out;
  border: none;
  font-size: 16px;
  min-width: 106px;
  color: $color-background;

  &:hover {
    color: $color-links;
    background-color: $color-button-hover;
    cursor: pointer;
  }
}

```

Slika 13: *Stil obrasca (vlastita izrada)*

Kako bi se došlo do ukupnog i prosječnog iznosa prihoda, rashoda i štednje potrebno je postaviti formule za dobivanje istih kao što je prikazano slikom 14.

```
useEffect(() => {
  incomeSum();
  outcomeSum();
  setSavings(totalIncome - totalOutcome);
  setTotalInc(totalIncome)
  setTotalOut(totalOutcome)
  setAvgIncome(totalInc / calc.length);
  setAvgOutcome(totalOut / calc.length);
  setAvgSavings(savings / calc.length);
});
```

Slika 14: Postavljanje formule za sumu i prosjek (vlastita izrada)

Za ukupne prihode i rashode potrebno je zbrojiti sve unesene prihode i rashode koje je korisnik unio kroz obrazac, dok se ukupna štednja izračunava tako da se ukupni rashodi oduzmu od ukupnih prihoda. Na kraju se ukupni iznosi podijele s `calc.length` što označava ukupan broj unosa. Ukoliko je korisnik imao pet unosa, sume svih stavki će biti podijeljene s pet. Sve sume i svi prosjeci biti će zaokruženi na dvije decimale što je kasnije definirano i vidljivo na slici 15.

```
<div className="table_wrap">
  <table className="table_month">
    <th className="table_header">Mjesečna tablica</th>
    <tr className="table_row_month">
      <td className="table_data bold"> Datum </td>
      <td className="table_data bold"> Prihodi </td>
      <td className="table_data bold"> Rashodi </td>
      <td className="table_data bold"> Štednja </td>
    </tr>
    {calc.map((x: any) => (
      <tr className="table_row_month">
        <td className="table_data">{moment(x.date).format('DD/MM/YYYY')}</td>
        <td className="table_data">{x.salary + x.transfers} KN</td>
        <td className="table_data">{x.outcome1 + x.outcome2 + x.outcome3 + x.outcome4 + x.outcome5 + x.outcome6 + x.outcome7 + x.outcome8 + x.outcome9 + x.outcome10 +
        <td className="table_data">{(x.salary + x.transfers) - (x.outcome1 + x.outcome2 + x.outcome3 + x.outcome4 + x.outcome5 + x.outcome6 + x.outcome7 + x.outcome8 +
        <td className="table_icons"><i className="icon icon--edit" onClick={() => update(x.id)}></i></td>
        <td className="table_icons"><i className="icon icon--delete" onClick={() => onDelete(x.id)}></i></td>
      </tr>
    ))}
    <tr className="table_row_month">
      <td className="table_data bold"> Ukupno </td>
      <td className="table_data bold">{totalInc.toFixed(2)} KN</td>
      <td className="table_data bold">{totalOut.toFixed(2)} KN</td>
      <td className="table_data bold">{savings.toFixed(2)} KN</td>
    </tr>
    <tr className="table_row_month">
      <td className="table_data bold"> Prosjek </td>
      <td className="table_data bold">{avgIncome.toFixed(2)} KN</td>
      <td className="table_data bold">{avgOutcome.toFixed(2)} KN</td>
      <td className="table_data bold">{avgSavings.toFixed(2)} KN</td>
    </tr>
  </table>
</div >
```


Slika 15: Kod za mjesečnu tablicu (vlastita izrada)

<Table> element se unosi za izradu tablice. <Tr> označava svaki red tablice unutar kojih se dodaju <td> tagovi koji označavaju ćelije. Na vrhu tablice biti će prikazan naslov tablice, u prvom redu će biti naslovi stupaca dok će se ostatak tablice popunjavati ukupnim prihodima i rashodima unesenih putem obrasca te će se na temelju njih izračunati i štednja. Zadnja dva reda u tablici rezervirana su za sume i prosjeke. Svakim novim unosom u obrazac će biti dodan jedan red u tablici koji izravno utječe na sumu i prosjek. Kao zadnje ćelije dodane su dvije ikonice: olovka i koš za smeće koji će klikom miša izvršavati zadane funkcije. Slikom 16 prikazan je kod za jednostavan stil tablice gdje će svaki red u tablici biti obojan drugom bojom. Svi parni redovi u tablici će biti obojani u boju zadanom klasom *color-primary* na samom početku .scss dokumenta dok će svi neparni redovi biti obojani bojom koja je zadana klasom *color-links*.

```
&__row {
  align-self: flex-start;

  &__month {
    align-self: flex-start;

    &:nth-of-type(even) {
      background: $color-primary ;
    }

    &:nth-of-type(odd) {
      background: $color-links;
    }
  }
}
```

Slika 16: Stil mjesečne tablice (vlastita izrada)

5.3. Implementacija

U ovom podpoglavlju prikazat će se stranice s obrascem, mjesečnom tablicom te detaljnom tablicom. Biti će prikazan izgled programskog rješenja, ali i njegova funkcionalnost. Kao što je vidljivo iz slike 17 navigacijski izbornik je pri vrhu stranice na kojem je označena ona stranica na kojoj se trenutno korisnik nalazi. Odmah ispod navigacijskog bara je centrirani obrazac za odabir datuma te za unose prihoda i svih skupina troškova po klasifikaciji.

Klasifikacija	unesite iznos
Plaće	unesite iznos
Transferi	unesite iznos
Stanarina, režije i gorivo	unesite iznos
Prijevoz	unesite iznos
Hrana i bezalkoholno piće	unesite iznos
Razna roba i usluge	unesite iznos
Rekreacija i kultura	unesite iznos
Restorani i hoteli	unesite iznos
Opremanje stambenog prostora	unesite iznos
Odjeća i obuća	unesite iznos
Alkoholna pića i duhanski proizvodi	unesite iznos
Zdravstvo	unesite iznos
Komunikacije	unesite iznos
Obrazovanje	unesite iznos

Slika 17: Prikaz gotove stranice unosa (vlastita izrada)

Nakon što korisnik unese sve željene stavke potrebno je kliknuti na dugme pri dnu stranice. Korisnik ne mora unijeti sve stavke kako bi se podaci uspješno poslali dalje, ukoliko je neki unos ostao prazan prihodi i rashodi se izračunavaju bez te stavke. Primjer unosa može se vidjeti na slici 18.

Unos Mjesečna tablica Izvještaj

30. 06. 2021.

Plaće	5500
Transferi	2000
Stanarina, režije i gorivo	1700
Prijevoz	300
Hrana i bezalkoholno piće	770
Razna roba i usluge	500
Rekreacija i kultura	400
Restorani i hoteli	200
Opremanje stambenog prostora	300
Odjeća i obuća	560
Alkoholna pića i duhanski proizvodi	380
Zdravstvo	0
Komunikacije	150
Obrazovanje	0

Dalje

Slika 18: Unos datuma i numeričkih vrijednosti (vlastita izrada)

Slikom 19 prikazan je modalni prozor koji se pojavljuje nakon klika na dugme na dnu stranice. Korisnik ima izbor otkazati i nastaviti s unosom u obrazac ili kliknuti na dalje. Klikom miša za nastavak se sve unesene informacije šalju u bazu podataka te se korisnika automatski premješta na stranicu „mjesečna tablica“.

The screenshot displays a web interface for entering monthly expenses. At the top, there are three buttons: 'Unos', 'Mjesečna tablica', and 'Izveštaj'. Below them is a date selector showing '30. 06. 2021.'. The main form consists of several rows, each with a category name and a corresponding input field containing a numerical value. A modal dialog box is centered over the form, asking for confirmation to proceed with the entry. The dialog has two buttons: 'Dalje' (Next) and 'Otkazi' (Cancel). At the bottom of the form, there is another 'Dalje' button.

Kategorija	Unos
Plaće	5500
Transferi	2000
Stanarina, režije i gorivo	1700
Prijevoz	300
Hrana i b	
Razna ro	
Rekreaci	
Restoran	
Opremanje stambenog prostora	300
Odjeća i obuća	560
Alkoholna pića i duhanski proizvodi	380
Zdravstvo	0
Komunikacije	150
Obrazovanje	0

Slika 19: Modalni prozor (vlastita izrada)

Na slici 20 prikazana je mjesečna tablica sa svim unosima koje je korisnik poslao u bazu podataka. Unos za datum 30.6.2021. je isti unos koji je prikazan slikama 18 i 19 te će biti korišten za demonstraciju ažuriranja i brisanja. Klikom na ikonu olovke u određenom redu korisnik ima izbor ažuriranja stavki, dok će klikom na ikonicu koša izbrisati cijeli red.

Unos Mjesečna tablica Izvještaj

Mjesečna tablica

Datum	Prihodi	Rashodi	Štednja	
30/04/2021	5000 KN	6650 KN	-1650 KN	 
31/05/2021	6000 KN	3570 KN	2430 KN	 
30/06/2021	7500 KN	5260 KN	2240 KN	 
Ukupno	18500.00 KN	15480.00 KN	3020.00 KN	
Prosjek	6166.67 KN	5160.00 KN	1006.67 KN	

Slika 20: Prikaz mjesečne tablice (vlastita izrada)

Unos
Mjesečna tablica
Izveštaj

Mjesečna tablica

Datum	Prihodi	Rashodi	Štednja	
30/04/2021	5000 KN	6650 KN	-1650 KN	
31/05/2021	6000 KN	3570 KN	2430 KN	
30/06/2021	7500 KN	5760 KN	1740 KN	
Ukupno	18500.00 KN	15980.00 KN	2520.00 KN	
Prosjek	6166.67 KN	5326.67 KN	840.00 KN	

Uspješno ažurirano! ×

Slika 21: Prikaz ažuriranja stavke (vlastita izrada)

Nakon klika na ikonicu olovke korisnik je vraćen na početnu stranicu za unos gdje ostaju zabilježeni prethodni iznosi stavki te se mogu mijenjati. To je jedini mogući način za izmjenu unesenih stavki jer svaki poslani obrazac u trenutku slanja dobije svoj unikatni id kojim je spremljen u bazu podataka, tako da novim unosom nije moguće utjecati na već postojeće unose iako se korisnik nalazi na istoj stranici kao i za unos novih obrazaca. Nakon ažuriranja pri dnu će se prikazati poruka na plavoj pozadini kako je unos uspješno ažuriran.

Unos
Mjesečna tablica
Izveštaj

Mjesečna tablica

Datum	Prihodi	Rashodi	Štednja	
30/04/2021	5000 KN	6650 KN	-1650 KN	
31/05/2021	6000 KN	3570 KN	2430 KN	
Ukupno	11000.00 KN	10220.00 KN	780.00 KN	
Prosjek	5500.00 KN	5110.00 KN	390.00 KN	

Stavka je uspješno uklonjena!
×

Slika 22: Prikaz brisanja stavke (vlastita izrada)

Klikom miša na ikonicu za brisanje uklanja stavku iz tablice uz poruku na dnu stranice kako je stavka uspješno uklonjena.

Slikom 23 prikazana je zadnja stranica u navigacijskom izborniku. Ona prikazuje detaljnu tablicu kojoj je dodijeljen sličan stil kao i tablici vidljivoj na slici 20. Unutar detaljne tablice nalaze se informacije o datumima unosa i izlistani iznosi svih unesenih stavki.

Unos Mjesečna tablica Izvještaj

Izvještaj

Datum	Plaće	Transferi	Stanarina, režije i gorivo	Prijevoz	Hrana i bezalkoholno pice	Razna roba i usluge	Rekreacija i kultura	Restorani i hoteli	Opremanje stambenog prostora	Odjeća i obuća	Alkoholna pica i duhanski proizvodi	Zdravstvo	Komunikacije	Obrazovanje
30/04/2021	5000 KN	0 KN	1560 KN	10 KN	10 KN	4000 KN	10 KN	10 KN	10 KN	10 KN	1000 KN	10 KN	10 KN	10 KN
31/05/2021	5200 KN	800 KN	1400 KN	300 KN	600 KN	400 KN	300 KN	100 KN	0 KN	0 KN	300 KN	70 KN	100 KN	0 KN

Slika 23: Prikaz svih unosa (vlastita izrada)

6. Zaključak

Cilj je ovog rada bio modelirati i napraviti web-stranicu kojom se može izračunati iznos štednje nekog kućanstva temeljena podacima o prihodima i rashodima. Programsko rješenje prikazano u ovom radu jednostavno je za uporabu i ne zahtijeva previše objašnjavanja. Riječ je o pojednostavljenom programskom rješenju i postoji još prostora za unapređenje oblikovane aplikacije. Unutar aplikacije mogla bi biti dodana obavezna registracija i prijava u sustav koji bi pratio financijska kretanja koja korisnici sami ažuriraju tijekom vremena, mogla bi se računati prosječna razina prihoda, troškova i štednje po kategorijama troškova prema namjeni i na temelju toga korisnicima dati informaciju o očekivanim budućim prihodima i rashodima. Također se može omogućiti da se uz korisničke račune može povezati njihov bankovni račun na kojem se evidentira promet pri čemu se uplate vode ili na plaće ili na druge transfere, a isplate na troškove koje vlasnik računa može staviti u neku od kategorija troškova. Ovo su samo neke od mogućnosti HTML-a, CSS-a i JavaScript programskog jezika koji su lako dostupni i s kojima je moguća izrada web-stranica na bilo kojem računalu. Moguće je testirati linije koda na *online* platformama kao što je CodePen ili „skinuti“ razne softvere za obradu koda, kao što je besplatni Microsoft softver za uređivanje izvornog koda Visual Studio Code. Ovaj bi rad mogao biti koristan onima koji žele

naučiti osnove razvijanja programskog sustava ili ih zanima po kojem načelu funkcioniraju web-stranice.

7. Literatura

1. Borozan Đ. (2019.), Makroekonomija, Ekonomski fakultet u Osijeku
2. Dukić B., Mitrović S.: Nastavni materijali za predmet Elektroničko i mobilno poslovanje, EFOS, ak. god. 2017/2018. (*on-line* verzija)
3. Mesarić J., Šebalj D.: Nastavni materijali za predmet Oblikovanje i implementacija informacijskih sustava, EFOS, ak. god. 2017/2018. (*on-line* verzija) te materijali za vježbe (modeliranje IS uz pomoć UML)
4. Html: HypertextMarkupLanguage <https://developer.mozilla.org/en-US/docs/Web/HTML>, [pristupljeno 21. travnja 2020.]
5. CSS: CascadingStyleSheets <https://developer.mozilla.org/en-US/docs/Web/CSS>, [pristupljeno 21. travnja 2020.]
6. Javascript <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, [pristupljeno 21. travnja 2020.]
7. Household consumption by purpose, https://ec.europa.eu/eurostat/statistics-explained/index.php/Household_consumption_by_purpose [pristupljeno 1. rujna 2020.]
8. Developer Survey Results 2020, <https://insights.stackoverflow.com/survey/2020> [pristupljeno 8. srpnja 2021.]
9. Typescript <https://www.typescriptlang.org> [pristupljeno 8. srpnja 2021.]
10. SASS: SyntacticallyAwesomeStyleSheets <https://sass-lang.com/> [pristupljeno 8. srpnja 2021.]

Slike

Eurostat (2019.), ec.europa.eu [mrežno] Dostupno na: https://ec.europa.eu/eurostat/statistics-explained/index.php/Household_consumption_by_purpose

Popis slika

Slika 1.*Grafički prikaz potrošnje kućanstva prema namjeni*

Slika 2.*Dijagram aktivnosti*

Slika 3.*Dijagram klasa*

Slika 4.*Sekvencijski dijagram*

Slika 5.*Prikaz HTML dokumenta bez CSS-a*

Slika 6.*Prikaz HTML dokumenta sa CSS-om*

Slika 7. *Prikaz CSS i SASS koda*

Slika 8.*Rezultati ankete Stackoverflow-a iz 2020. godine*

Slika 9.*Kod za navigacijski izbornik*

Slika 10.*Stil navigacijskog izbornika*

Slika 11.*Definiranje varijabli*

Slika 12.*Kod za obrazac*

Slika 13.*Stil obrasca*

Slika 14.*Postavljanje formule za sumu i prosjek*

Slika 15.*Kod za mjesečnu tablicu*

Slika 16.*Stil mjesečne tablice*

Slika 17.*Prikaz gotove straniceunosa*

Slika 18.*Unos datuma i numeričkih vrijednosti*

Slika 19.*Modalni prozor*

Slika 20.*Prikaz mjesečne tablice*

Slika 21.*Prikaz ažuriranja stavke*

Slika 22.*Prikaz brisanja stavke*

Slika 23.*Prikaz svih unosa*