Web aplikacija za izračun rate kredita te otplatni plan

Kranjčec, Karlo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:301770

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij

WEB-APLIKACIJA ZA IZRAČUN RATE KREDITA TE OTPLATNI PLAN

Završni rad

Karlo Kranjčec

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	Karlo Kranjčec
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	AR 4786, 27.07.2020.
JMBAG:	0165075007
Mentor:	Ivan Hrehorović, prof.
Sumentor:	izv. prof. dr. sc. Zdravko Krpić
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 1:	Ivan Hrehorović, prof.
Član Povjerenstva 2:	doc. dr. sc. Tomislav Rudec
Naslov završnog rada:	Web aplikacija za izračun rate kredita te otplatni plan
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada:	Izraditi aplikaciju koja će za poznate ulazne podatke izračunati ratu kredita, ispisati otplatni plan, izračunati EKS, NKS. Sumentor s FERIT-a: Zdravko, Krpić

Datum ocjene pismenog dijela završnog rada od strane mentora:	21.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Vrlo dobar (4)
Datum obrane završnog rada:	26.09.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)

Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	05.10.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 05.10.2024.

Ime i prezime Pristupnika:	Karlo Kranjčec
Studij:	Stručni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR 4786, 27.07.2020.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za izračun rate kredita te otplatni plan**

izrađen pod vodstvom mentora Ivan Hrehorović, prof.

i sumentora izv. prof. dr. sc. Zdravko Krpić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

Sadržaj

1. UVOD 1
1.1. Zadatak završnog rada 1
2. POJMOVI IZ FINANCIJSKE MATEMATIKE 2
2.1. ZAJAM 2
2.2. DEKURZIVNI OBRAČUN KAMATA 2
2.3. MODEL OTPLATE ZAJMA DEKURZIVNO JEDNAKIM ANUITETIMA 2
3. STRUKTURA WEB-APLIKACIJE 5
3.1. IZGLED KORISNIČKOG SUČELJA I UPOTREBA 5
3.1.1. Navigacijska traka 5
3.1.2. Stambeni kredit 5
3.1.3. Gotovinski kredit 7
3.2. BOOTSTRAP 10
3.3. FONT AWESOME 11
3.4. STRUKTURA HTML-A 12
3.5. STRUKTURA CSS-A 15
3.6. JAVASCRIPT 21
3.6.1. JavaScript funkcija 22
3.6.2. JavaScript za klizače 22
3.6.3. Predaja ulaznih podataka 24
3.6.4. Modalna tablica i izračun 25
4. Zaključak 31
Sažetak 32
Abstract 33
Životopis 34
Literatura 35

1. UVOD

Pojam kredita, rate i kamate često se može uočiti u svakodnevici. Prema [1, p. 78], kredit ili zajam čine: iznos odobrenog zajma, kamatna stopa, način obračuna kamata, vrijeme otplate zajma i način otplate zajma. Postoji potreba za olakšavanjem složenosti i objašnjavanjem nejasnoća oko izračuna te definicije financijske matematike koja je u pozadini svih vrsta kamata te kredita i njegovog otplatnog plana.

Teorijski dio rada primjenjen je za svu matematiku i logiku koja se nalazi iza korisničkog sučelja *web*-aplikacije koja pomaže informirati potencijalne primatelje kredita. Navedeni postupak i izračuni podijeljeni su na više koraka i načina te su pojašnjeni u drugom poglavlju. Osnovni pojmovi financijske matematike kao što su kamate, anuitet i razdoblja otplate također su objašnjeni u drugom poglavlju.

Praktični dio rada, prikazan u trećem poglavlju, sastoji se od demonstracije korištenja funkcionalnosti i korisničkog sučelja *web*-aplikacije. Navedena *web*-aplikacija na osnovu ulaznih podataka izračunava otplatni plan proizvoljnog kredita. Tehnologije pomoću kojih je aplikacija realizirana su *HTML, CSS, Javascript* i *Bootstrap library*. U konačnici je na praktičnom primjeru objašnjen proces izrade *web* sučelja, pozadinske logike te je demonstrirano korištenje aplikacije od strane krajnjeg korisnika. Posljednje je poglavlje zaključak u kojem su sažeti teorijski i praktični dio rada te potencijalna unaprijeđenja *web*-aplikacije.

1.1. Zadatak završnog rada

Zadatak ovog rada je izraditi *web*-aplikaciju koja će na osnovu poznatih ulaznih podataka izračunati ratu kredita te ispisati otplatni plan kredita.

.

2. POJMOVI IZ FINANCIJSKE MATEMATIKE

2.1. ZAJAM

Prema [1, p. 78] zajam ili kredit predstavlja imovinsko-pravni odnos između zajmodavca i zajmoprimca. Taj se odnos regulira posebnim ugovorom o zajmu tako da se u njemu definiraju:

a) iznos odobrenog zajma
b) kamatna stopa
c) način na koji će se obračunati iznos kamata
d) vrijeme otplate zajma
e) način otplate zajma.

Zajmoprimac zajam vraća zajmodavcu anuitetima. Anuiteti su novčani iznosi koji ostaju jednake vrijednosti tijekom otplatnog razdoblja i sačinjeni su od otplatnih kvota i kamata. Krajem otplatnog razdoblja, zajmoprimac je dužan vratiti posuđenu glavnicu (zajam) i iznos obračunatih, složenih kamata. Pregled otplate zajma nalazi se u otplatnoj tablici (planu) koja se sastoji od stupaca u kojima se navode: razdoblje otplate zajma, anuiteti, kamate, otplatne kvote i ostatak dugovanja.

2.2. DEKURZIVNI OBRAČUN KAMATA

Izrađena *web*-aplikacija koristi složeni dekurzivni izračun kamate. Prema [1, p. 44], dekurzivni složeni obračun kamate vrši se tako da se na kraju razdoblja ukamaćivanja izračunaju kamate na glavnicu s početka tog razdoblja te se kamate nakon obavljenog izračuna pribrajaju glavnici odnosno glavnica je promjenjiva. Drugim riječima, prema [2, p. 173], kamate računamo ne samo na glavnicu nego i na prethodno obračunate kamate. Dakle ako se posudi 100 € uz dekurzivnu godišnju kamatnu stopu od p, tada se na kraju godine mora vratiti $100 + p$ €.

2.3. MODEL OTPLATE ZAJMA DEKURZIVNO JEDNAKIM ANUITETIMA

U ovom modelu dužnik zajam otplaćuje uz složen i dekurzivan obračun kamata i jednake anuitete krajem dogovorenog vremenskog razdoblja. Anuitet je otplatni obrok koji dužnik vraća, a prema [1, p. 79], sastoji se od dijela glavnice i dijela kamate.

Prema [1, pp. 78-79] izraz za iznos jednakog mjesečnog anuiteta je:

$$\boldsymbol{a = C\ \frac{{r_{mj}}^{n}\ (r_{mj} - 1)}{{r_{mj}}^{n} - 1}}$$

Gdje je :

a – anuitet

C – iznos zajma (glavnice)

r_{mj} – dekurzivni mjesečni kamatni faktor

n – broj mjeseci.

Prema [2, p. 175] dekurzivni godišnji kamatni faktor $\boldsymbol{r}$ računa se preko izraza:

$$\boldsymbol{r = 1 + \frac{p}{100}}$$

gdje je p godišnja kamatna stopa. Ako je ukamaćivanje mjesečno, dekurzivni mjesečni kamatni faktor r_{mj} računa se prema izrazu:

$$\boldsymbol{r_{mj} = \sqrt[12]{r}}$$

Dok se mjesečna kamatna stopa p_{mj} može izračunati pomoću r_{mj} prema izrazu :

$$\boldsymbol{p_{mj} = \left(r_{mj} - 1\right) * 100}$$

Prema [1, p. 79] izraz za kamate k-tog razdoblja (k = 1, 2,....n) je:

$$\boldsymbol{I_k = \frac{C_{k-1} * p_{mj}}{100}}$$

gdje je:

I_k - kamata za k-to razdoblje

C_{k-1} – glavnica na kraju prošlog razdoblja

p_{mj} – mjesečna kamatna stopa.

Prema [1, p. 80] izraz za otplatnu kvotu glavnice k-tog razdoblja je:

$$\boldsymbol{R_k = a - I_k}$$

gdje je:

R_k – otplatna kvota glavnice za k-to razdoblje

I_k - kamata za k-to zadoblje

a – anuitet.

Prema [1, p. 80] izraz za ostatak dugovanja (glavnice) k-tog razdoblja je :

$$\boldsymbol{C_k = C_{k-1} - R_k}$$

gdje je:

C_k – ostatak dugovanja (glavnice) za k-to razdoblje

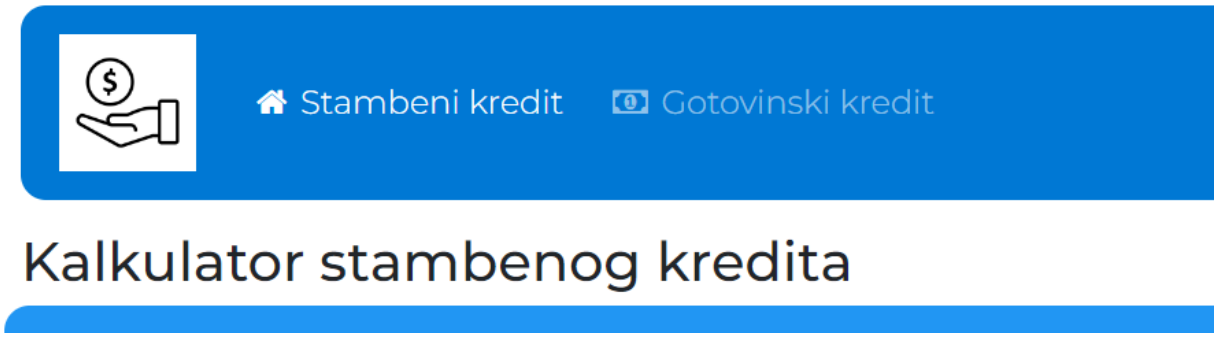
C_{k-1} – glavnica na kraju prošlog razdoblja

R_k – otplatna kvota glavnice za k-to razdoblje.

3. STRUKTURA WEB-APLIKACIJE

3.1. IZGLED KORISNIČKOG SUČELJA I UPOTREBA

3.1.1. Navigacijska traka



Slika 3.1. Navigacijska traka aplikacije

Strukturu *web*-aplikacije čine zaglavlje, tijelo i podnožje stranice. Navigacijska traka u zaglavlju stranice omogućuje odabir između dva različita korisnička sučelja *web*-aplikacije koji zasebno iz ulaznih parametara računaju otplatne planove gotovinskog ili stambenog kredita. Slika 3.1. prikazuje kako je unutar navigacijske trake stranice odabrano korisničko sučelje za stambeni kredit. Spomenuto se može uočiti iz promjene jačine bijele boje oko navigacijskog elementa s natpisom „Stambeni kredit“ i naslovom koji se nalazi ispod navigacijske trake.

3.1.2. Stambeni kredit

Tijelo se sastoji od dva dijela. Unutar lijevog navigacijskog područja korisnik pomoću klizača (*eng. slider*) odabire iznos željenog kredita i otplatnog razdoblja. Dodatno se kao parametar unosi iznos kamatne stope. Odabir korisnika ograničen je minimalnim i maksimalnim parametrima koji za iznos stambenog kredita iznose od deset tisuća eura do petsto tisuća eura. Odabir korisnika potom je ograničen minimalnom i maksimalnom vrijednošću vremena trajanja kredita koja za stambeni kredit iznosi od minimalno trideset šest do maksimalno tristo šezdeset mjeseci. Odabir kamatne stope ograničen je na raspon između jedan i dvadeset posto.

Slika 3.2. prikazuje testne podatke korisnika koji podiže kredit na otplatni rok od deset godina u iznosu sto trideset tisuća eura uz kamatnu stopu od pet cijelih devedeset dva posto. Odabirom opcije „Izračunaj“ generira se otplatni plan za testnog korisnika te se ispisuju osnovni podaci o kreditu. Na Slika 3.3. prikazan je iznos mjesečnog anuiteta, ukupni iznos otplate, ukupni iznos kamate u otplati, kamatna stopa i vrsta kamatne stope.

Slika 3.2. Odabir početnih parametara stambenog kredita

Odabirom opcije "Otvorite detaljni otplatni plan" korisniku se na idućem ekranu prikazuje tablica s mjesečnim izračunom otplatnog plana. Iako je algoritam korišten za prikaz izračuna matematički točan, konačni izračun isključivo je informativnog karaktera te bi za konkretni otplatni plan u svakom slučaju bilo preporučljivo kontaktirati banku.

Slika 3.3. Ispis osnovnih informacija o stambenom kreditu

Otplatni plan za stambeni kredit

Period	Datum dospijeća	Otplatni obrok	Glavnica	Kamata	Stanje kredita
0	20. 09. 2024.	0,00 €	0,00 €	0,00 €	250.000,00 €
1	20. 10. 2024.	1.977,51 €	935,01 €	1.042,50 €	249.064,99 €
2	20. 11. 2024.	1.977,51 €	938,90 €	1.038,60 €	248.126,09 €
3	20. 12. 2024.	1.977,51 €	942,82 €	1.034,69 €	247.183,27 €
4	20. 01. 2025.	1.977,51 €	946,75 €	1.030,75 €	246.236,52 €
5	20. 02. 2025.	1.977,51 €	950,70 €	1.026,81 €	245.285,82 €
6	20. 03. 2025.	1.977,51 €	954,66 €	1.022,84 €	244.331,16 €
7	20. 04. 2025.	1.977,51 €	958,64 €	1.018,86 €	243.372,51 €
8	20. 05. 2025.	1.977,51 €	962,64 €	1.014,86 €	242.409,87 €
9	20. 06. 2025.	1.977,51 €	966,66 €	1.010,85 €	241.443,22 €
10	20. 07. 2025.	1.977,51 €	970,69 €	1.006,82 €	240.472,53 €

Slika 3.4. Prikaz prvih deset perioda otplatnog plana stambenog kredita

Slika 3.4. prikazuje detaljni otplatni plan stambenog kredita s periodima, datumom dospijeća, otplatnim obrokom (anuitetom), glavnicom, kamatom i stanjem kredita. Valja obratiti pažnju na omjer glavnice i kamate u anuitetu prvog perioda gdje se može vidjeti kako je udio kamate veći od udjela glavnice. Udio kamate postepeno se smanjuje kroz periode, a udio glavnice povećava.

170	20. 11. 2038.	1.977,51 €	1.889,03 €	88,48 €	19.328,97 €
171	20. 12. 2038.	1.977,51 €	1.896,90 €	80,60 €	17.432,07 €
172	20. 01. 2039.	1.977,51 €	1.904,81 €	72,69 €	15.527,26 €
173	20. 02. 2039.	1.977,51 €	1.912,76 €	64,75 €	13.614,50 €
174	20. 03. 2039.	1.977,51 €	1.920,73 €	56,77 €	11.693,77 €
175	20. 04. 2039.	1.977,51 €	1.928,74 €	48,76 €	9.765,03 €
176	20. 05. 2039.	1.977,51 €	1.936,78 €	40,72 €	7.828,24 €
177	20. 06. 2039.	1.977,51 €	1.944,86 €	32,64 €	5.883,38 €
178	20. 07. 2039.	1.977,51 €	1.952,97 €	24,53 €	3.930,41 €
179	20. 08. 2039.	1.977,51 €	1.961,12 €	16,39 €	1.969,29 €
180	20. 09. 2039.	1.977,51 €	1.969,29 €	8,21 €	0,00 €

Slika 3.5. Prikaz zadnjih deset perioda otplatnog plana stambenog kredita

Slika 3.5. detaljno prikazuje odnos kamate i glavnice. Uočljivo je znatno smanjenje iznosa kamate koji se ovdje nalazi u predzadnjem redu plana otplate. Udio glavnice za to se vrijeme znatno povećao. Uzimajući prethodne zaključke u obzir, može se zaključiti da su krediti na dulji vremenski period manje isplativiji od kredita s kraćim vremenskim periodom. Taj odnos bit će vidljiv iz prikaza otplatnog plana za gotovinski kredit.

3.1.3. Gotovinski kredit

Struktura korisničkog sučelja kalkulatora za gotovinski kredit razlikuje se od strukture korisničkog sučelja za stambeni kredit u dvije ključne točke. Maksimalni mogući iznos za gotovinski kredit značajno je manji od onoga za stambeni kredit te iznosi četerdeset tisuća eura, a minimalni mogući iznos također je značajno manji u slučaju stambenog kredita te iznosi deset tisuća eura. Rok je otplate kod gotovinskog kredita u minimalnom vremenskom trajanju od trinaest mjeseci i

maksimalnom vremenskom trajanju od sto dvadeset mjeseci. Slika 3.6. prikazuje ranije spomenute razlike.

Slika 3.6. Odabir početnih parametara gotovinskog kredita

Slika 3.6. prikazuje sličnosti korisničkog sučelja za gotovinski kredit s korisničkim sučeljem za stambeni kredit. Preko klizača korisnik odabire iznos kredita i rok otplate te dodatno unosi iznos kamatne stope. Odabrani se iznosi odmah po odabiru ili promjeni prikazuju korisniku te se pri odabiru opcije „Izračunaj“ generira otplatni plan za testnog korisnika.

Slika 3.7. Ispis osnovnih informacija o gotovinskom kreditu

Slika 3.7. prikazuje osnovne podatke o kreditu nakon izračuna otplatnog plana. Podatci koji se mogu uočiti su iznos mjesečnog anuiteta, ukupni iznos otplate, ukupni iznos kamate u otplati, kamatnu stopu i vrstu kamatne stope s vrijednostima za odabrane testne podatke. Odabirom opcije „Otvorite detaljan otplatni plan“ korisniku se prikazuje detaljan prikaz otplatnog plana.

Otplatni plan za gotovinski kredit

Period	Datum dospijeća	Otplatni obrok	Glavnica	Kamata	Stanje kredita
0	20. 09. 2024.	0,00 €	0,00 €	0,00 €	20.000,00 €
1	20. 10. 2024.	397,91 €	277,91 €	120,00 €	19.722,09 €
2	20. 11. 2024.	397,91 €	279,58 €	118,33 €	19.442,50 €
3	20. 12. 2024.	397,91 €	281,26 €	116,66 €	19.161,25 €
4	20. 01. 2025.	397,91 €	282,95 €	114,97 €	18.878,30 €
5	20. 02. 2025.	397,91 €	284,64 €	113,27 €	18.593,66 €
6	20. 03. 2025.	397,91 €	286,35 €	111,56 €	18.307,30 €
7	20. 04. 2025.	397,91 €	288,07 €	109,84 €	18.019,23 €
8	20. 05. 2025.	397,91 €	289,80 €	108,12 €	17.729,43 €
9	20. 06. 2025.	397,91 €	291,54 €	106,38 €	17.437,90 €
10	20. 07. 2025.	397,91 €	293,29 €	104,63 €	17.144,61 €

Slika 3.8. Prikaz prvih deset perioda otplatnog plana za gotovinski kredit

Slika 3.8. prikazuje detaljni otplatni plan gotovinskog kredita s periodima, datumom dospijeća, otplatnim obrokom, glavnicom, kamatom i stanjem kredita. Treba obratiti posebnu pozornost na

odnos glavnice i kamate u anuitetu prvog perioda, gdje se može vidjeti kako je udio kamate manji od udjela glavnice. Do desetog perioda udio kamate značajno se smanjio u odnosu na udio glavnice koji je narastao.

50	20. 11. 2028.	397,91 €	372,57 €	25,34 €	3.850,92 €
51	20. 12. 2028.	397,91 €	374,81 €	23,11 €	3.476,11 €
52	20. 01. 2029.	397,91 €	377,06 €	20,86 €	3.099,05 €
53	20. 02. 2029.	397,91 €	379,32 €	18,59 €	2.719,73 €
54	20. 03. 2029.	397,91 €	381,60 €	16,32 €	2.338,14 €
55	20. 04. 2029.	397,91 €	383,89 €	14,03 €	1.954,25 €
56	20. 05. 2029.	397,91 €	386,19 €	11,73 €	1.568,06 €
57	20. 06. 2029.	397,91 €	388,51 €	9,41 €	1.179,56 €
58	20. 07. 2029.	397,91 €	390,84 €	7,08 €	788,72 €
59	20. 08. 2029.	397,91 €	393,18 €	4,73 €	395,54 €
60	20. 09. 2029.	397,91 €	395,54 €	2,37 €	0,00 €

Slika 3.9.Prikaz prvih deset perioda otplatnog plana za gotovinski kredit

Slika 3.9. detaljno prikazuje omjer kamate i glavnice u zadnjih deset redaka otplatne tablice. Može se uočiti znatno smanjenje iznosa kamate koji se ovdje nalazi u predzadnjem redu plana otplate. Udio glavnice za to se vrijeme znatno povećao stoga je moguće zaključiti da su krediti na kraći vremenski period isplativiji od kredita na dulji vremenski period.

3.2. BOOTSTRAP

Bootstrap je besplatan i *open source* alat dizajniran za izradu prilagodljivih *web*-stranica i *web*-aplikacija te je jedan od najpopularnijih *CSS* okvira. *Bootstrap*, kao unaprijed pripremljen *library* s širokom zajednicom i usklađenosti s različitim *web*-preglednicima, programerima omogućava lagan, brz te dosljedan razvoj *web*-sadržaja koji ima mogućnost prilagodbe na sve vrste ekrana bez suvišnog ponavljanja koda. Međutim, jedan od problema s *Bootstrap*-om je taj što se ne može omogućiti specifično specijaliziranje s obzirom da se popriličan broj njegovih elemenata mora ručno zaobići pri pokušajima prilagođavanja. Također, *Bootstrap* potiče uporabu velikog broja klasa u *HTML* elementima što kod čini nečitljivim. [3] Uzevši u obzir prednosti i nedostatke *Bootstrap library*-ja, prvenstveno je, pri izradi *web*-stranice, korišten čisti *HTML*, *CSS* i *JavaScript* uz nadopunu verzije *Bootstrap 5*.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
        rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
        crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
        crossorigin="anonymous"></script>
```

Slika 3.10. Uključivanje Bootstrap-a u dokument

Slika 3.10. prikazuje uključivanje *Bootstrap*-a u dokument preko glave *HTML* dokumenta.

Prva poveznica u dokument uključuje *Bootstrap CSS library*. Druga poveznica uključuje *Bootstrap* JavaScript *library*.

```html
<body class="montserrat-mainFont">
    <header class="myheader ">
        <nav class="navbar navbar-expand-sm navbar-dark montserrat-mainFont mycolor-bg mynav-padding mx-auto py-3 ">
            <div class="container-fluid">
                <a class="navbar-brand" href="#"><img src="images/loan-vector-icon.jpg" alt="Logo" width="100" height="auto"></a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="collapsibleNavbar">
                    <ul class="navbar-nav mynav-text">
                        <li class="nav-item">
                            <a class="nav-link active" href="index.html"><i class="fa fa-home nav-i"></i>Stambeni kredit</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="gotovinski.html"><i class="fa fa-money nav-i"></i>Gotovinski kredit</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
```

Slika 3.11. Bootstrap unutar navigacijske trake

Slika 3.11. prikazuje upotrebu *Bootstrap*-a u određenim klasama u dokumentu kako bi se postiglo lakše uvođenje navigacijske trake i osnovna prilagodljivost ekrana. Klase koje se koriste za navigacijsku traku unuatr *class* atributa sadrže ključnu riječ „*navbar*“ ili „*nav*“. Klasni atribut „container-flud“ koristi se za prilagodljivost ekrana.

3.3. FONT AWESOME

Font Awesome library je gotovih fleksibilnih vektorskih ikona na *web*-u. Prednost korištenja *Font Awesome*-a u *HTML* dokumentima, njihova je jednostavna primjena. Slika 3.12 prikazuje kako se jednostavim navođenjem veze na željeni *library* u glavu *HTML*-a, *Font Awesome library* može uključiti u dokument. Korištene su ikone iz *Font Awesome 4*.

```
<link href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="css/basicStructure.css">
<link rel="stylesheet" href="css/customStructure.css">
<link rel="stylesheet" href="css/myRangeSlider.css">
<link rel="stylesheet" href="css/media-queries.css">
<link rel="stylesheet" href="css/fullScreen.css">
<link rel="stylesheet" href="css/otplatnaTablica.css">
```

Slika 3.12. Uključivanje Font Awesome veze u dokument

Pri odabiru ikone u popisu svih ikona na službenoj stranici *Font Awesome,* odabrana ikona može se uključiti u željeno mjesto unutar dokumenta s jednom linijom koda odnosno dodatkom ključne riječi „fa“ i imena ikone u klasu *HTML* oznake „i“. Slika 3.13. prikazuje uključivanje dvije različite vektorske ikone u navigacijku traku *web*-stranice, svaka unutar jedne linije koda. Pri izradi ove *web*-stranice korištene su ikone *Font Awesome* zbog svoje jednostavnosti, funkcionalnosti i odlične mogućnosti prilagođavanja.

```
<div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav mynav-text">
        <li class="nav-item">
            <a class="nav-link active" href="index.html"><i class="fa fa-home nav-i"></i>Stambeni kredit</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="gotovinski.html"><i class="fa fa-money nav-i"></i>Gotovinski kredit</a>
        </li>
    </ul>
</div>
```

Slika 3.13.Uključivanje Font Awesome ikona za dom i novac u navigacijsku traku

3.4. STRUKTURA HTML-A

Svaki *HTML* dokument mora početi s *HTML DOCTYPE* deklaracijom. Navedena deklaracija prenosi pregledniku informacije o tipu i verziji dokumenata što pomaže pri tumačenju stranice. Najčešća *HTML* oznaka za *DOCTYPE* je *DOCTYPE HTML* koja se koristi u *HTML5* standardu i osigurava usklađenost s modernim *web* standardima. Osim *HTML DOCTYPE*, svaki *HTML* dokument mora sadržavati *HTML* oznaku koja označava izvor stranice i unutar koje se nalazi sav njen sadržaj. [4]

Većina *HTML* stranica sastoji se od *HTML* oznaka za glavu i tijelo. U glavi stranice nalazi se naslov *web* stranice koji se prikazuje u kartici *web* preglednika, skup znakova koji definira sve podržane znakove unutar postavljenog standarda te poveznica na ostale vanjske dokumente koji pomažu u prikazivanju sadržaja stranice na prozivoljne načine njenog autora. [5] U tijelu se kod većine stranica nalaze zaglavlje, glavni dio sadržaja i podnožje. U zaglavlju se najčešće nalazi

samo navigacijska traka u kojoj su navigacijski elementi koji vode na različite dijelove stranice ili pak na vanjske izvore, no u zaglavlje se može dodati bilo koji element. U glavnom dijelu sadržaja nalazi se glavni dio sadržaja stranice koji mora biti jedinstven i ne smiju postojati više od dva glavna sadržaja stranice. U podnožju stranice nalaze se informacije o autoru, autorskim pravima, veze na početak stranice ili veze na povezane dokumente. [4]

HTML obrasci (*eng. forms*) su *HTML* oznake koje se koriste za skupljanje i predaju korisničkih unosa na *web*-u. Obrasci omogućuju različite mogućnosti uzimanja korisničkih unosa i njihovo oblikovanje kroz *HTML form* oznaku. Neke od vrsta korisničkih unosa u obrascima su tekstna polja, brojevi, e-pošta, lozinke, datum i vrijeme. Obrasci reguliraju različite tipove događaja pri predaji podataka, od slanja podataka *web* poslužitelju do upisivanja podataka unutar *web* stranice. [4] Slika 3.14 prikazuje kako je pri izradi ove *web*-stranice korišten jedan obrazac za unos brojeva i spremanje tih brojeva unutar *web*-stranice pritiskom korisnika na gumb unutar obrasca.

```
<form id="stambeniForm" class="column my-form">
    <div class="column">
        <div class="slidecontainer">
            <h3>Odaberite iznos</h3>
            <input type="range" step="1000" min="10000" max="500000" value="250000" class="slider" id="myRange1">
            <div class="slider-values">
                <span id="minValue" class="width25"></span>
                <strong><span id="myRange1Output" class="width50"></span></strong>
                <span id="maxValue" class="width25"></span>
              </div>
              <input class="width50 my-input-field output" type="number" id="numberInput1" minlength="5" maxlength="6"
              min="10000" max="500000" step="1000" value="250000">
            </div>
        <br>
          <div class="slidecontainer">
            <h3>Odaberite rok otplate</h3>
            <input type="range" step="1" min="36" max="360" value="180" class="slider" id="myRange2">
            <div class="slider-values">
                <span id="minValueMonth" class="width25"></span>
                <strong><span id="myRange2Output" class="width50"></span></strong>
                <span id="maxValueMonth" class="width25"></span>
              </div>
              <input class="width50 my-input-field output" type="number" id="numberInputMonth2" minlength="2" maxlength="3"
              min="36" max="360" step="1" value="180">
            </div>
            <br>
            <div class="slidecontainer">
              <h3 class="my-h3">Odaberite kamatnu stopu</h3>
              <input type="number" id="rataStambeni-input" class="rStambeni kamata-input my-input-field output" minlength="2" maxlength="2"
              min="1" max="20" step="0.01" value="5">
              <strong><div id="rataStambeni-output" class="width80 output"></div></strong>
            </div>
    </div>
    <button class="mybtn" type="submit"><i class="fa fa-pencil-square-o btn-i"></i><strong>Izračunaj</strong></button>
    <br>
</form>
```

Slika 3.14. HTML obrazac s tri različita polja za unos brojeva

```html
<div class="slidecontainer">
    <h3>Odaberite iznos</h3>
    <input type="range" step="1000" min="10000" max="500000" value="250000" class="slider" id="myRange1">
    <div class="slider-values">
        <span id="minValue" class="width25"></span>
        <strong><span id="myRange1Output" class="width50"></span></strong>
        <span id="maxValue" class="width25"></span>
      </div>
      <input class="width50 my-input-field output" type="number" id="numberInput1" minlength="5" maxlength="6"
      min="10000" max="500000" step="1000" value="250000">
  </div>
```

Slika 3.15. Struktura HTML klizača

Slika 3.15 prikazuje *HTML* klizač i njegovu strukturu. Klizač se sastoji od tipa unosa *range* koji se definira u *HTML input* polju te mu se zadaju vrijednosti koraka, minimalna i maksimalna vrijednost te početna vrijednost. Klizači su korišteni pri izradi ove *web*-aplikacije kako bi omogućili jednostavan izbor i ograničenje izbora ulaznih parametara unutar određenih granica. Slika 3.15 prikazuje i *HTML input* polje u koje korisnik može ručno unijeti ulazne podatke bez upotrebe klizača. Polje unosa je tipa broj te je korisnički unos ograničen na iste vrijednosti kao što je to slučaj kod klizača, dakle unos klizača i unos u polje su povezani.

HTML tablica uređena je kao skup podataka sadržan u redove i stupce u obliku tablice. Tablice služe za prikazivanje tekstualnih i brojevnih podataka na uređen način. *HTML* tablice koristan su alat za brzo i jednostavno uspostavljanje veza između različitih vrsta podataka. Podatci sadržani unutar tablica mogu se oblikovati i prilagoditi na različiti broj načina. Osnovni je raspored podataka na redove i stupce unutar zaglavlja, tijela i podnožja tablice. Podnožje se često smije izostaviti iz strukture, no zaglavlje i tijelo su bitni radi opisa i uspostavljanja veza između podataka sadržanih unutar tablice. [4] Pri izradi ove *web*-aplikacije, tablica se koristi za prikaz otplatnog plana kredita radi jednostavnog ispisa i bolje preglednosti rezultata u planu otplate.

```html
<table class="my-table" id="stambeniKredit-tablica">
        <thead>
            <tr>
                <th>Period</th>
                <th>Datum dospijeća</th>
                <th>Otplatni obrok</th>
                <th>Glavnica</th>
                <th>Kamata</th>
                <th>Stanje kredita</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td data-label="Period">No data</td>
                <td data-label="Datum dospijeća">No data</td>
                <td data-label="Otplatni obrok">No data</td>
                <td data-label="Glavnica">No data</td>
                <td data-label="Kamata">No data</td>
                <td data-label="Stanje kredita">No data</td>
            </tr>
        </tbody>
    </table>
```

Slika 3.16.Struktura tablice

Slika 3.16 prikazuje strukturu tablice korištene pri izradi *web*-aplikacije. U zaglavlju tablice nalaze se opisi svakog stupca, a u tijelu tablice podaci za svaki stupac. Prije predaje ulaznih parametara na računanje, tablica je gotovo potpuno prazna i sadrži samo dva retka od kojih jedan služi za interpretiranje ispisa. Više o načinu kako se tablica popunjava prilikom izračuna bit će pojašnjeno u kasnijim poglavljima.

3.5. STRUKTURA CSS-A

CSS ili *Cascading Style Sheets* jezik je koji se koristi za poboljšavanje i stiliziranje izgleda *web*-stranica. *CSS* kontrolira kako se *HTML* elementi kao tekst, slike ili gumbi prikazuju na *web*-stranici. Pomoću *CSS*-a, mogu se prilagođavati veličine fontova, boje, dodavati pozadinske slike i mijenjati izgledi *web*-stranice pri određenim rezolucijama ekrana. [5] *CSS* omogućava rukovanje rasporedom stranice pretvarajući jednostavnu *web*-stranicu u svoju bolju inačicu, koja će biti jednostavna za rukovanje, imati poboljšano korisničko iskustvo i biti vizualno zanimljivija. Vanjski se *CSS* dokument u *HTML* dokument može uvesti preko *HTML* elementa *link* u čiji se *src* atribut unosi putanja do *CSS* dokumenta. [6]

```
html, body {
    height: 100%;
    margin: 0;
}
body {
    min-height: 50rem;
    display: flex;
    flex-direction: column;
}
main {
    justify-content: center; /*po x-osi*/
    align-items: center; /*po y osi*/
    flex: 1;
}
footer {
    color: white;
    text-align: center;
    padding: 0.625rem 0;
}
* {
  box-sizing: border-box;
}
/* Container for flexboxes */
.row {
  display: flex;
  flex-wrap: nowrap;
}
/* Create three equal columns */
.column {
  flex: 33%;
  padding: 1.25rem;
}
.mygrid-container {
  justify-content: space-between;
  grid-template-columns: 15rem 15rem 15rem; /*Make the grid smaller than the container*/
  gap: 1rem;
  background-color: #2196F3;
  padding: 1.25rem ;
  border-radius: 1.15rem;
}
```

Slika 3.17. Flexbox u CSS-u

Pri izradi *web*-aplikacije korišteno je svojstvo prikaza rastezljive kutije (eng. *flexbox display*) koje omogućava lakšu izradu prilagodljivih i rastezljivih *web*-stranica koje će imati smislen izgled na uređajima različitih dimenzija ekrana. Prednost je uporabe *flexbox*-a u jednostavnom i učinkovitom prilagođavanju dimenzija *HTML* elemenata slobodnom prostoru ekrana pomoću kojeg sprječava njihovo nekontrolirano gužvanje i pomicanje po ekranu, te u jednostavnom rješavanju problema vodoravnog i okomitog centriranja. Upravo zbog ovih prednosti, *flexbox* se koristio pri izradi ove *web*-stranice.

Slika 3.17. prikazuje upotrebu *CSS* svojstva *flexbox* u svrhu dinamičnog i prilagodljivog dizajna. *Flexbox* se u element uključuje upotrebom *CSS* svojstva *display: flex* u izborniku oznake za tijelo (*eng. body*) te svoj djeci elementima odnosno elementima sadržanim u sebi nasljeđuje svojstvo *flex*. U klasnom izborniku *row*, svojstvom *flex-direction: column* u izborniku oznake za tijelo, sadržaju *flexbox*-a omogućava se da se pri promjeni dimenzija ekrana prilagodi tim promjenama i zadrži smislen izgled sadržaja bez nepotrebnog gužvanja ili spajanja sadržaja.

Svojstvo prikaza rešetke (*eng. grid display*) korišteno je pri izradi manjih dijelova *web*-aplikacije u kombinaciji s *flexbox* svojstvom. *Grid* pruža lagano vodoravno i okomito usklađivanje elemenata, točno pozicioniranje elemenata unutar dobro definirane rešetke koristeći stupce i retke. Omogućava prikaz rasporeda koji će biti prilagodljiv ekranima svih dimenzija. [5]

```
.btn-i {
    padding-right: 0.5rem;
    padding-left: 0.5rem;
}
p.medium-font {
    font-size: medium;
}
.mid-grid {
    display: grid;
}
.mid-grid p {
    margin-bottom: 0.5rem;
}
.mid-grid-group {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding-top: 0.7rem;
    padding-bottom: 0.7rem;
}
.mid-column {
    padding-left: 3rem;
    text-align: left;
}
.mid-result {
    padding-right: 3rem;
    text-align: left;
}
```

Slika 3.18. Grid u CSS-u

Slika 3.18. prikazuje uključivanje *CSS* svojstva *grid* u svrhu dinamičnog i prilagodljivog dizajna. *Grid* se u element uključuje upotrebom *CSS* svojstva *display: grid* pri čemu se u pozadini *web*-stranice pojavljuje rešetka koja svakom elementu sadržanom unutar roditeljskog elementa nasljeđuje svojstvo *grid* te određuje točno mjesto po vodoravnoj i okomitoj osi. Pozicijom svakog elementa može se manipulirati izmjenom i dodavanjem različitih svojstava za početak, kraj te širinu redova i stupaca.

Pri izradi ove *web*-aplikacije korišteni su medijski upiti (*eng. media query*). Medijski su upiti *CSS* svojstva koja omogućavaju *web*-stranicama da pri različitim vrstama uređaja i dimenzijama ekrana prikazuju mjerodavan sadržaj bez gužvanja ili nestajanja elemenata sa *web*-stranice pri izmjeni dimenzija ekrana ili orijentacija uređaja. [5] [6]

```
@media (max-width: 480px) {
    table {
        font-size: medium;
    }
    .my-table, .my-table tbody, .my-table th, .my-table td, .my-table tr {
        display: block;
    }
    .my-table thead {
        display: none;
    }
    .my-table tr {
        margin-bottom: 15px;
    }
    .my-table tbody {
        padding-top: 1rem;
    }
    .my-table td {
        display: flex;
        justify-content: space-between;
        padding: 10px;
        border: 1px solid #ddd;
        border-radius: 5px;
        background-color: #fff;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
    .my-table td::before {
        content: attr(data-label);
        font-weight: bold;
        flex-basis: 50%;
    }
}
```

Slika 3.19. Medijski upit za tablicu

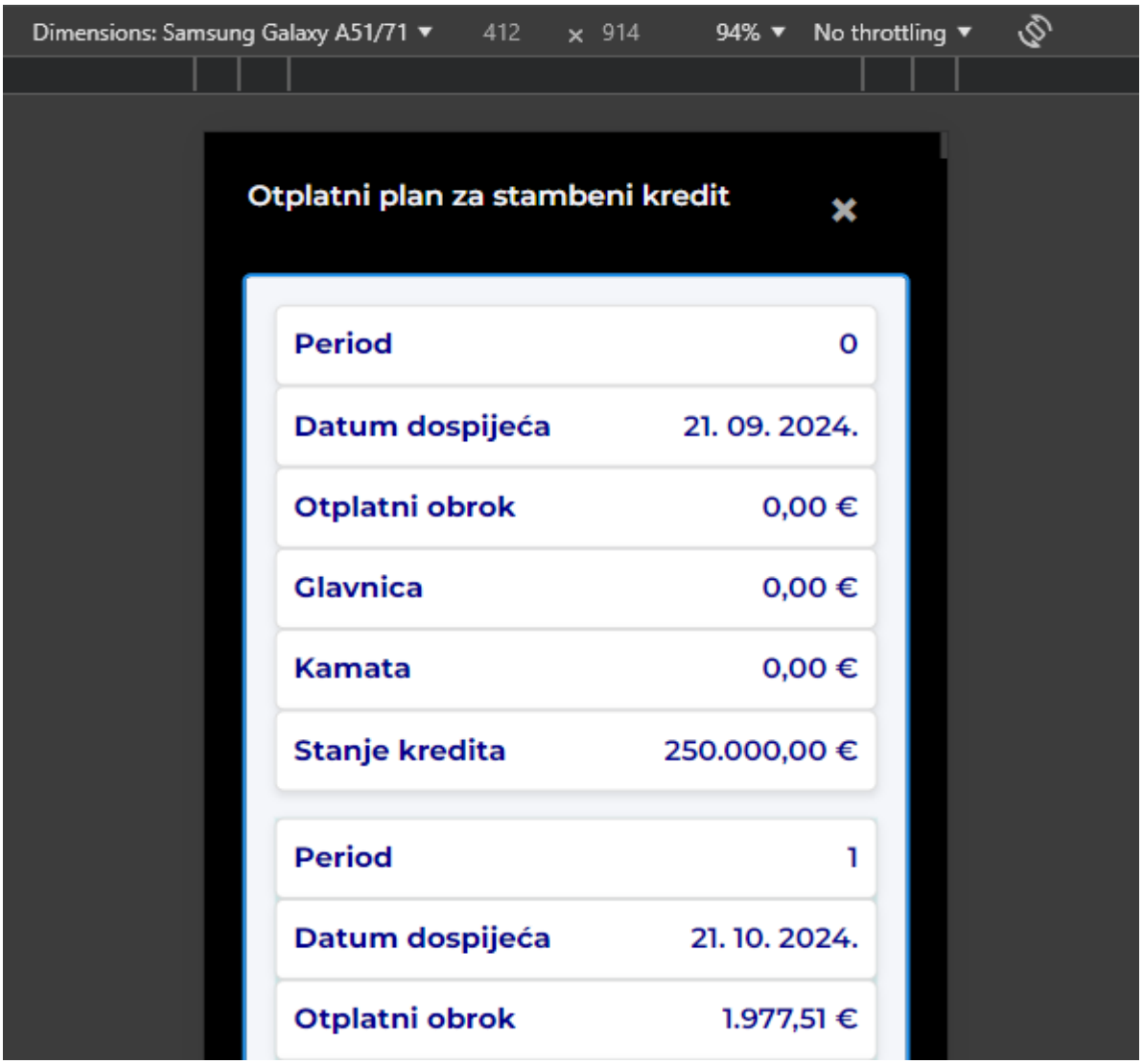
Slika 3.19. prikazuje medijski upit za uređaje s maksimalnom širinom ekrana do četristo osamdeset piksela. Medijski upit u pitanju uklanja zaglavlje tablice te od stupaca pomoću svojstva *flexbox* stvara novu strukturu tablice s razmacima između sadržaja, postavljajući svojstvo *justify-content* na vrijednost *space-between* koje će ostaviti prazan prostor između različitih elemenata. Dok selektor *td::before* prije svakog stupca ubacuje sadržaj atributa *data-label* koji će postati malo zagljavlje za svaki stupac, s obzirom da će se u tablicu ubaciti prije svakog stupca. Svojstvo *flex-basis* označava da će se elementi raširiti do pedeset posto ovisno o veličini sadržaja *data-label*.

Otplatni plan za stambeni kredit

Period	Datum dospijeća	Otplatni obrok	Glavnica	Kamata	Stanje kredita
0	21. 09. 2024.	0,00 €	0,00 €	0,00 €	250.000,00 €
1	21. 10. 2024.	1.977,51 €	935,01 €	1.042,50 €	249.064,99 €
2	21. 11. 2024.	1.977,51 €	938,90 €	1.038,60 €	248.126,09 €
3	21. 12. 2024.	1.977,51 €	942,82 €	1.034,69 €	247.183,27 €
4	21. 01. 2025.	1.977,51 €	946,75 €	1.030,75 €	246.236,52 €
5	21. 02. 2025.	1.977,51 €	950,70 €	1.026,81 €	245.285,82 €
6	21. 03. 2025.	1.977,51 €	954,66 €	1.022,84 €	244.331,16 €
7	21. 04. 2025.	1.977,51 €	958,64 €	1.018,86 €	243.372,51 €
8	21. 05. 2025.	1.977,51 €	962,64 €	1.014,86 €	242.409,87 €
9	21. 06. 2025.	1.977,51 €	966,66 €	1.010,85 €	241.443,22 €
10	21. 07. 2025.	1.977,51 €	970,69 €	1.006,82 €	240.472,53 €
11	21. 08. 2025.	1.977,51 €	974,73 €	1.002,77 €	239.497,80 €
12	21. 09. 2025.	1.977,51 €	978,80 €	998,71 €	238.519,00 €
13	21. 10. 2025.	1.977,51 €	982,88 €	994,62 €	237.536,12 €
14	21. 11. 2025.	1.977,51 €	986,98 €	990,53 €	236.549,14 €
15	21. 12. 2025.	1.977,51 €	991,10 €	986,41 €	235.558,04 €

Slika 3.20. Ispis otplatne tablice dok medijski upit nije aktivan

Slika 3.20. prikazuje kako izgleda ispis tablice otplatnog plana dok medijski upit nije aktivan. Zaglavlje je jedinstveno u tablici te svaki redak, osim zagljavlja, sadrži samo podatke o mjesečnoj uplati. Ispis u ovom formatu jako je pregledan pri velikim dimenzijama ekrana dok se pri manjim dimenzijama ekrana stupci prilagođavaju rasporedu te negativno utječu na preglednost i korisničko iskustvo, stoga je potrebno koristiti medijski upit vidljiv na Slika 3.19.



Slika 3.21. Ispis otplatne tablice kad je medijski upit aktivan

Slika 3.21. prikazuje ispis otplatne tablice kada je medijski upit prema Slika 3.19. aktivan odnosno kada je zadovoljen uvjet da je dimenzija ekrana manja od četiristo osamdeset piksela. Pri tom ispisu tablice, može se primjetiti da tablica više ne sadrži stupce i retke. Tablica je pri ovoj dimenziji ekrana poprimila oblik kartica koje sadrže elemente zaglavlja i podatke relevantne za ta zaglavlja u retcima, odvojene praznim prostorom. Takav je zapis pregledniji i bolji za korisničko iskustvo pri manjim dimenzijama ekrana.

CSS struktura i način prilagodbe klizača definirani su u zasebnom *CSS* dokumentu. Slika 3.22. prikazuje detaljno prilagođavanje klizača pomoću različitih klasa, identifikatora i pripadajućih svojstava. Jedno od svojstava *CSS* je *opacity* parametar koji utječe na prozirnost elementa. Svojstvo *opacity* i prozirnost elementa obrnuto su proporcionalni. Unutar klase *slider*, *opacity* iznosi sedamdeset posto dok se u klasi *slider* sa svojstvom *:hover* odnosno kada korisnik prijeđe

preko klizača mišem vrijednost *opacity* svojstva se poveća na sto posto i time se smanjuje prozirnost u potpunosti.

```
.slidecontainer {
    width: 100%;
  }
  .slider {
    appearance: none;
    width: 80%;
    height: 15px;
    border-radius: 5px;
    background: #5d6b83;
    outline: none;
    opacity: 0.7;
    -webkit-transition: .2s;
    transition: opacity .2s;
    margin-top: 1rem;
  }
  .slider:hover {
    opacity: 1;
  }
  .slider-values {
    display: flex;
    justify-content: space-between;
    margin-top: 1.25rem;
  }
  .slider-values span {
    text-align: center;
  }
  .width80 {
    font-size: large;
    font-weight: 900;
    width: 100%;
    padding-top: 1rem;
    text-align: center;
  }
  span.width50 {
    width: 50%;
    font-size: large;
    font-weight: 900;
  }
  span.width25 {
    width: 25%;
    font-size: medium;
    font-weight: 600;
  }
  #minValue,#maxValue,#minValueMonth,#maxValueMonth {
    font-size: medium;
  }
```

Slika 3.22. CSS struktura klizača

3.6. JAVASCRIPT

JavaScript je moćan i svestran programski jezik za programiranje *web*-sadržaja. *JavaScript* se koristi za dodavanje atraktivnog sadržaja poput animacija, interaktivnih obrazaca i prilagodljivog sadržaja na *web*-stranice. Ovaj programski jezik ima širok spektar primjene i moguće ga je koristiti na više razvojnih platformi. Postoji velik broj *JavaScript framework*-a i *library*-ja, no za razvoj ove *web*-aplikacije koristi se obični *JavaScript*. [7]

3.6.1. JavaScript funkcija

Arrow funkcija vrsta je *JavaScript* funkcije u kojoj se ključne riječi *function* i *return* izostave iz sintakse funkcije te se strelica postavlja između argumenta i zagrada na početku tijela funkcije. [8] Slika 3.23. prikazuje primjer uporabe *arrow* funkcije bez imena u *forEach* petlji. Funkcija se sastoji samo od parametara „rata“ i „index“, strelice koja upućuje na zagrade na početku tijela funkcije i tijelo funkcije s operacijama koje funkcija izvodi.

```
// Popuni tablicu podacima iz plana otplate
planOtplate.forEach((rata, index) => {
    const red = document.createElement("tr");
    const date = new Date();
    date.setMonth(date.getMonth() + index);
    const formattedDate = date.toLocaleDateString('hr-HR', {
        year: 'numeric',
        month: '2-digit',
        day: '2-digit'
    });
    red.innerHTML = `
        <td data-label='Period'>${rata.mjesec}</td>
        <td data-label='Datum dospijeća'>${formattedDate}</td>
        <td data-label='Otplatni obrok'>${formatNumber(Number(rata.anuitet))}</td>
        <td data-label='Glavnica'>${formatNumber(Number(rata.udioGlavnice))}</td>
        <td data-label='Kamata' >${formatNumber(Number(rata.udioKamate))}</td>
        <td data-label='Stanje kredita' >${formatNumber(Number(rata.stanjeKredita))}</td>
    `;
    tbody.appendChild(red);
});
```

Slika 3.23. JavaScript arrow funkcija

3.6.2. JavaScript za klizače

Slika 3.24. prikazuje dvije funkcije za postavljanje i ažuriranje vrijednosti koje pokazuju klizače i polja za unos brojeva ispod klizača. Obje funkcije primaju četiri različita identifikatora od četiri različita elemnta iz *HTML*-a. Stvaraju varijable koje koriste kao lokalne kopije nad kojima će vršiti operacije. Funkcija prvo ispisuje minimalnu i maksimalnu vrijednost granice unosa koju korisnik može unijeti kao ulazni parametar u korisničko sučelje. Zatim se postavlja ispis korisnikova odabira u polje za unos i ispis ispod klizača. Neovisno o tome koristi li se za unos polje namijenjeno za unos ili klizač, funkcija ispisuje odabrani iznos kredita ili rok otplate u elemente čiji su joj identifikatori predani kao parametri, što je prikazano na Slika 3.25. U funkciji za ispis iznosa kredita koristi se vanjska funkcija *formatNumber* kako bi ispis brojeva imao sljedeći format: tisućice.stotice,decimalni dio.

```javascript
function formatNumber(number) {

    return number.toLocaleString("hr-HR", {style:"currency", currency:"EUR"});
}

function initializeMoneySlider(sliderId, outputId, minId, maxId, numberInputId) {
    var slider = document.getElementById(sliderId);
    var output = document.getElementById(outputId);
    var minValue = document.getElementById(minId);
    var maxValue = document.getElementById(maxId);
    var numberInput = document.getElementById(numberInputId);

    minValue.innerHTML = "Min : " + formatNumber(Number(slider.min));
    maxValue.innerHTML = "Max : " + formatNumber(Number(slider.max));
    output.innerHTML = formatNumber(Number(slider.value));
    numberInput.value = slider.value;

    slider.oninput = function() {
        output.innerHTML = formatNumber(Number(this.value));
        numberInput.value = this.value;
    }
    numberInput.oninput = function() {
        slider.value = this.value;
        output.innerHTML = formatNumber(Number(this.value));
    }
}

function initializeMonthSlider(sliderId, outputId, minId, maxId, numberInputId) {
    var slider = document.getElementById(sliderId);
    var output = document.getElementById(outputId);
    var minValue = document.getElementById(minId);
    var maxValue = document.getElementById(maxId);
    var numberInput = document.getElementById(numberInputId);

    minValue.innerHTML = "Min : " + slider.min + " mj.";
    maxValue.innerHTML = "Max : " + slider.max + " mj.";
    output.innerHTML = "Vaš odabir : " + slider.value + " mj.";

    slider.oninput = function() {
        output.innerHTML = "Vaš odabir : " + this.value + " mj.";
        numberInput.value = this.value;
    }
    numberInput.oninput = function() {
        slider.value = this.value;
        output.innerHTML = "Vaš odabir : " + this.value + " mj.";
    }
}
```

Slika 3.24. Funkcije za postavljanje i ažuriranje vrijednosti klizača

Slika 3.25. Prikaz i ažuriranje klizača

3.6.3. Predaja ulaznih podataka

Predaja ulaznih podataka u *web*-aplikaciji odvija se pomoću funkcije *handleFormSubmission()* koja kao parametar zaprima događaj. Događaj je akcija poput klika pokazivača na gumb, prelazak pokazivača preko elementa ili predaja obrasca. U kreiranju *web*-aplikacije koristi se događaj *submit*. Slika 3.27. prikazuje izvođenje funkcije *handleFormSubmission()* prilikom događaja *submit*. Funkcija se poziva tako da se na element s identifikatorom *stambeniForm* postavi *EventListener* koji se okida prilikom predaje podataka iz obrasca odnosno prilkom događaja *submit*. Slika 3.26. prikazuje mogućnost odabira opcije „Izračunaj“ unutar korisničkog sučelja. Koristeći ovu funkcionalnost izračuni se vrše i šalju u korisničko sučelje.

```
            <strong><div id="rataStambeni-output" class="width80 output"></div></strong>
          </div>
    </div>
    <button class="mybtn" type="submit"><i class="fa fa-pencil-square-o btn-i"></i><strong>Izračunaj</strong></button>
    <br>
</form>
```

Slika 3.26. Opcija „Izračunaj“

```
</body>
<script src="JS/sliders.js"></script>
<script src="JS/fullscreen_overlay.js"></script>
<script src="JS/showingRates.js"></script>
<script src="JS/fillModalTable.js"></script>
<script src="JS/calculations.js"></script>
<script src="JS/handleFormSubmission.js"></script>
<script>
    document.addEventListener("DOMContentLoaded", function() {
        initializeMoneySlider("myRange1", "myRange1Output", "minValue", "maxValue", "numberInput1");
    });
    document.addEventListener("DOMContentLoaded", function() {
        initializeMonthSlider("myRange2", "myRange2Output", "minValueMonth", "maxValueMonth", "numberInputMonth2");
    });
    document.addEventListener("DOMContentLoaded", function(){
        modalOperations("myModal", "myBtn1", "close" );
    });
    document.addEventListener("DOMContentLoaded", function() {
        showRate("rataStambeni-input", "rataStambeni-output");
    });
    document.getElementById("stambeniForm").addEventListener("submit", handleFormSubmission);
</script>

</html>
```

Slika 3.27. EventListener na događaj submit

```
function handleFormSubmission(event) {
    event.preventDefault();
    // dohvati podatke iz kreditne forme
    const glavnica = parseFloat(document.getElementById("numberInput1").value);
    const godisnjaKamatnaStopa = parseFloat(document.getElementById("rataStambeni-input").value);
    const rokOtplate = parseInt(document.getElementById("numberInputMonth2").value);

    // pohrani kreditne podake u local storage
    localStorage.setItem("kreditPodaci", JSON.stringify({ glavnica, godisnjaKamatnaStopa, rokOtplate }));
    // osvjezi modalnu tablicu
    osvjeziModalnuTablicu();
  }
```

Slika 3.28. Funkcija handleFormSubmission

Uobičajeno je ponašanje pri slanju obrasca ponovno učitavanje cijele *web*-stranice odnosno *web*-aplikacije, ali se takvo ponašanje sprječava korištenjem funkcije *preventDefault()*. Slika 3.28. prikazuje sprječavanje uobičajenog ponašanja pri slanju obrasca, dohvaćanje podataka iz kreditnog obrasca te pohranjivanje u lokalnu pohranu *web*-stranice te pozivanje funkcije *osvjeziModalnuTablicu()*.

3.6.4. Modalna tablica i izračun

Modalna tablica označava prikaz koji se nalazi u pozadini *web*-aplikacije i skrivena je od korisnika sve dok korisnik ne odabere opciju „Otvorite detaljni otplatni plan“ kao što je prikazano na Slika 3.29. Otvaranje i zatvaranje vrši funkcija *modalOperations()*. Slika 3.31. i Slika 3.32 prikazuju

definiciju i upotrebu funkcije *modalOperations()* koja prima tri različita identifikatora od tri različita elementa iz *HTML*-a i ovisno o identifikatoru poziva se različita funkcija unutar funkcije *modalOperations()*. Klikom na element s identifkatorom *myBtn1*, modalna se tablica iz skrivenog prikaza pojavljuje preko korisničkog sučelja. Klikom na element s klasom *close*, modalna se tablica skriva od korisnika. Na cijeli element s identifikatorom *myModal1* odnosno na cijelu modalnu tablicu koja nije dio tablice sa sadržajem otplatnog plana, postavlja se *EventListener* koji na klik pokazivača bilo gdje izvan tablice zatvara tablični prikaz. Slika 3.30. prikazuje otvorenu modalnu tablicu bez podataka, s naznačenim elementom za izlaz iz tablice.

```
<h3 class="my-h3 my-padding">Informacije o kreditu</h3>
<button class="mybtn" id="myBtn1">
        <i class="fa fa-table btn-i"></i><strong>Otvorite detaljni otplatni plan</strong>
</button>
<h5 class="info-padding ">* Izračuni na ovom kalkulatoru su isključivo informativnog karaktera zbog čega se ne mogu koristiti u druge svrhe.</h5>
<div id="myModal" class="modal">
    <span class="otplatniTitle">Otplatni plan za stambeni kredit</span>
    <span class="close">&times;</span>
    <div class="modal-content">
    <table class="my-table" id="stambeniKredit-tablica">
            <thead>
                <tr>
                    <th>Period</th>
                    <th>Datum dospijeća</th>
                    <th>Otplatni obrok</th>
                    <th>Glavnica</th>
                    <th>Kamata</th>
                    <th>Stanje kredita</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td data-label="Period">No data</td>
                    <td data-label="Datum dospijeća">No data</td>
                    <td data-label="Otplatni obrok">No data</td>
                    <td data-label="Glavnica">No data</td>
                    <td data-label="Kamata">No data</td>
                    <td data-label="Stanje kredita">No data</td>
                </tr>
            </tbody>
        </table>
```

Slika 3.29. Modalna tablica u HTML-u

Slika 3.30. Prikaz prazne modalne tablice u korisničkom sučelju

```
function modalOperations(modalId, btnId, spanId ) {
    // Get the modal
  var modal = document.getElementById(modalId);

  // Get the button that opens the modal
  var btn = document.getElementById(btnId);

  // Get the <span> element that closes the modal
  var span = document.getElementsByClassName(spanId)[0];

  // When the user clicks the button, open the modal
  btn.onclick = function() {
    modal.style.display = "block";
  }
  // When the user clicks on <span> (x), close the modal
  span.onclick = function() {
      modal.style.display = "none";
  }
  // When the user clicks anywhere outside of the modal, close it
  window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
  }
}
```

Slika 3.31. Funkcija za otvaranje i zatvaranje modalne tablice

```
<script src="JS/sliders.js"></script>
<script src="JS/fullscreen_overlay.js"></script>
<script src="JS/showingRates.js"></script>
<script src="JS/fillModalTable.js"></script>
<script src="JS/calculations.js"></script>
<script src="JS/handleFormSubmission.js"></script>
<script>
    document.addEventListener("DOMContentLoaded", function() {
        initializeMoneySlider("myRange1", "myRange1Output", "minValue", "maxValue", "numberInput1");
    });
    document.addEventListener("DOMContentLoaded", function() {
        initializeMonthSlider("myRange2", "myRange2Output", "minValueMonth", "maxValueMonth", "numberInputMonth2");
    });
    document.addEventListener("DOMContentLoaded", function(){
        modalOperations("myModal", "myBtn1", "close" );
    });
```

Slika 3.32. modalOperations u HTML-u

```
function osvjeziModalnuTablicu() {
    const kreditPodaci = JSON.parse(localStorage.getItem("kreditPodaci"));

    // ako su podaci o kreditu dostupni, popuni tablicu
    if (kreditPodaci) {
        const tablicaOtplateKredita = document.getElementById("stambeniKredit-tablica");
        const tbody = tablicaOtplateKredita.querySelector("tbody");
        tbody.innerHTML = "";

        // Izračunaj plan otplate
        const planOtplate = izracunajPlanOtplateKredita(kreditPodaci);

        // Formatiranje brojeva koji će se prikazati u tablici da bi imali format npr. 10.000,00
        function formatNumber(number) {
            return number.toLocaleString("hr-HR", {style:"currency", currency:"EUR"});
          }

        // Popuni tablicu podacima iz plana otplate
        planOtplate.forEach((rata, index) => {
            const red = document.createElement("tr");
            const date = new Date();
            date.setMonth(date.getMonth() + index);
            const formattedDate = date.toLocaleDateString('hr-HR', {
                year: 'numeric',
                month: '2-digit',
                day: '2-digit'
            });
            red.innerHTML = `
                <td data-label='Period'>${rata.mjesec}</td>
                <td data-label='Datum dospijeća'>${formattedDate}</td>
                <td data-label='Otplatni obrok'>${formatNumber(Number(rata.anuitet))}</td>
                <td data-label='Glavnica'>${formatNumber(Number(rata.udioGlavnice))}</td>
                <td data-label='Kamata' >${formatNumber(Number(rata.udioKamate))}</td>
                <td data-label='Stanje kredita' >${formatNumber(Number(rata.stanjeKredita))}</td>
            `;
            tbody.appendChild(red);
        });
    }
}
```

Slika 3.33 prikaz funkcije osvjeziModanuTablicu

Funkcija *osvježiModalnuTablicu()* prilagođava strukturu sadržaja podataka iz lokalne pohrane i sprema ih u varijablu čija vijednost ne može biti promijenjena odnosno *kreditPodaci* tipa *const*. Zatim se vrši provjera postoje li podaci u *kreditPodaci*, kreira se tablica *const tablicaOtplateKredita* te se dohvaća tablica iz *HTML*-a s jedinstvenim identifikatorom *stambeniKredit-tablica*. Potom se kreira varijabla *const tbody* kojoj se pomoću *querySelector()* funkcije dodaje element s *HTML* oznakom *tbody* te se njegov sadržaj prazni. Ovaj je postupak nužan kako bi korisnik mogao vršiti više izračuna i otvarati više tablica zaredom tijekom korištenja *web*-aplikacije. Zatim se kreira varijabla *planOtplate* unutar koje se poziva funkcija *izracunajPlanOtplateKredita()* s parametrom *kreditPodaci*.

Prije nego što se sadržaj modalne tablice može prikazati korisniku, u pozadini se moraju obaviti izračuni unutar funkcije *izračunajOtplatniPlanKredita()*. Funkcija je podijeljenja na dva dijela. U

prvom se dijelu *kreditPodaci* predani kao parametar funkcije dohvaćaju u tri varijable tipa *const*. Varijabla *glavnica* dohvaća iznos glavnice, varijabla *godisnjaKamatnaStopa* dohvaća iznos godišnje kamatne stope dok varijabla *rokOtplate* dohvaća koliki je rok otplate korisnik unio iz podataka predanih funkciji.

Funkcija zatim vrši izračune prema Slika 3.35 i sprema ih u varijable tipa *const: mjesecnaKamatnaStopaFixed*, *dekurzivniKamatniFaktor* te u *anuitetCalculation*. Funkcija potom kreira prazno polje *planOtplate*, vrši inizijalizacije izvan *for* petlje, izračune unutar *for* petlje i pri kraju svakog izračuna sprema rezultate u polje *planOtplate* prema Slika 3.35. Slika 3.36. prikazuje izvršavanje drugog dijela funkcije u kojem se vrši zbrajanje varijable *glavnica* i konačnog rezultata računskih operacija nad varijablom *ukupniUdioKamate*. U gore spomenutoj slici može se uočiti kako funkcija *izračunajOtplatniPlanKredita()* vrši ispis sažetih informacija o gotovinskom ili stambenom kreditu u elemente *HTML*-a s određenim identifikatorima. Koje će se naredbe ispisa provesti ovisi o vrijednosti anuiteta dohvaćenog po indentifikatoru tj. koji od dohvaćenih anuiteta nije prazan. Funkcija *izračunajOtplatniPlanKredita()* kao povratnu vrijednost vraća polje *planOtplate*.

Po završetku izračuna funkcije *izračunajOtplatniPlanKredita()*, funkcija *osvježiModalnuTablicu()* nastavlja s izvršavanjem naredbi kako je prikazano na Slika 3.33. Funkcija *osvježiModalnuTablicu()* pohranjuje povratnu vrijednost funkcije *izračunajOtplatniPlanKredita()* u varijablu tipa *const*. Zatim *forEach* petlja prolazi po polju i izvršava *arrow* funkciju sadržanu u sebi. *Arrow* funkcija inicijalizira dvije varijable tipa *const*. Prva varijabla je *rata* koja stvara elemente *tr* i *date* kojem se dodjeljuje formatirana vrijednost lokalnog datuma iz funkcije *toLocaleDateString()*. Slika 3.34 prikazuje formatiranje unutarnjeg sadržaj *HTML*-a varijable *red* i njeno dodavanje u varijablu *tbody*.

Aplikacija je dostupna na poveznici :
https://kanki98.github.io/LoanCalculatorWebApp/index.html

```
red.innerHTML = `
    <td data-label='Period'>${rata.mjesec}</td>
    <td data-label='Datum dospijeća'>${formattedDate}</td>
    <td data-label='Otplatni obrok'>${formatNumber(Number(rata.anuitet))}</td>
    <td data-label='Glavnica'>${formatNumber(Number(rata.udioGlavnice))}</td>
    <td data-label='Kamata' >${formatNumber(Number(rata.udioKamate))}</td>
    <td data-label='Stanje kredita' >${formatNumber(Number(rata.stanjeKredita))}</td>
`;
tbody.appendChild(red);
```

Slika 3.34. Prikaz formatiranja reda tablice i dodavanje reda u tijelo tablice

```javascript
function izracunajPlanOtplateKredita(kreditPodaci) {
    //dohvaćamo podatke iz local storage
    const glavnica = kreditPodaci.glavnica;
    const godisnjaKamatnaStopa = kreditPodaci.godisnjaKamatnaStopa;
    const rokOtplate = kreditPodaci.rokOtplate;

    // zaokružujemo kamatnu stopu na 3 decimale
    const godisnjaKamatnaStopaFixed = godisnjaKamatnaStopa.toFixed(3);
     // računamo godišnji dekurzivni kamatni faktor
    const dekurzivniKamatniFaktor = 1 + (godisnjaKamatnaStopaFixed/100);
    console.log(dekurzivniKamatniFaktor);
    // računamo mjesečni kamatni faktor
    const dekurzivniKamatniFaktorMj = Math.pow(dekurzivniKamatniFaktor, 1/12);

    const mjesecnaKamatnaStopa = (dekurzivniKamatniFaktorMj - 1) * 100;
    // Izračunavamo anuitet (konstantnu mjesečnu otplatu)
    const anuitetCalculation =
        glavnica *  (Math.pow(dekurzivniKamatniFaktorMj, rokOtplate) * (dekurzivniKamatniFaktorMj - 1))  / (Math.pow(dekurzivniKamatniFaktorMj, rokOtplate) - 1);

    // Kreiramo niz koji će čuvati detalje svake rate
    const planOtplate = [];

    // Inicijaliziramo preostali dug, udio kamate, udio glavnice
    let stanjeKredita = glavnica;
    let udioKamate = 0;
    let udioGlavnice = 0;
    let anuitetStart  = 0;
    let ukupniUdioKamate = 0;
    let ukupniIznosOtplate = glavnica;
    // Iteriramo kroz sve rate
    for (let mjesec = 0; mjesec <= rokOtplate; mjesec++) {
        // ako je razdoblje placanja nulto tj tek je ugovoren kredit, sve osim glavnice je 0
        if(mjesec === 0) {
            udioKamate = 0;
            udioGlavnice = 0;
            stanjeKredita = glavnica;
            anuitet = anuitetStart;
        }else {
            anuitet = anuitetCalculation;
            // Izračunavamo kamatu za trenutni mjesec
            udioKamate = (stanjeKredita * mjesecnaKamatnaStopa) / 100;
            ukupniUdioKamate += udioKamate;
            // Izračunavamo otplatu glavnice za trenutni mjesec
            udioGlavnice = anuitet - udioKamate;

            // Ažuriramo preostali dug i pazimo o predznaku jer u zadnjim danima otplate zna biti negativan
            stanjeKredita = Math.abs(stanjeKredita - udioGlavnice);
        }
        // Spremamo podatke o rati u niz
        planOtplate.push({
          mjesec,
          anuitet: anuitet.toFixed(2),
          udioGlavnice: udioGlavnice.toFixed(2),
          udioKamate: udioKamate.toFixed(2),
          stanjeKredita: stanjeKredita.toFixed(2),
        });
      }
```

Slika 3.35. Funkcija izracunajPlanOtplateKredita() 1. dio

```javascript
    // izracunaj ukupni iznos otplate
    ukupniIznosOtplate = glavnica + ukupniUdioKamate;

    // popunjavanje i formatiranje osnovnih podataka o kreditu, detaljniji ispis je u drugoj tablici
    function formatNumber(number) {
      return number.toLocaleString("hr-HR", {style:"currency", currency:"EUR"});
    }
    if(document.getElementById("anuitetStambeni")) {
            document.getElementById("anuitetStambeni").innerHTML = formatNumber(Number(anuitet.toFixed(2)));
            document.getElementById("ukupniIznosStambeni").innerHTML = formatNumber(Number(ukupniIznosOtplate.toFixed(2)));
            document.getElementById("ukupniIznosKamateStambeni").innerHTML = formatNumber(Number(ukupniUdioKamate.toFixed(2)));
            document.getElementById("kamatnaStopaStambeni").innerHTML = document.getElementById("rataStambeni-input").value + " %";
            document.getElementById("tipKamatneStopeStambeni").innerHTML = "fiksna";
    } else {
            document.getElementById("anuitetGotovinski").innerHTML = formatNumber(Number(anuitet.toFixed(2)));
            document.getElementById("ukupniIznosGotovinski").innerHTML = formatNumber(Number(ukupniIznosOtplate.toFixed(2)));
            document.getElementById("ukupnaKamataGotovinski").innerHTML = formatNumber(Number(ukupniUdioKamate.toFixed(2)));
            document.getElementById("kamatnaStopaGotovinski").innerHTML = document.getElementById("rataGotovinski-input").value + " %";
            document.getElementById("tipKamatneStopeGotovinski").innerHTML = "fiksna";
    }
  return planOtplate;
}
```

Slika 3.36. Funkcija izracunajPlanOtplateKredita()2. dio

4. Zaključak

Teorijskim dijelom završnog rada objašnjeni su ekonomski pojmovi poput rate, kredita i kamate. Prikazane su matematičke funkcije korištene za izračun anuiteta kredita dekurzivnim obračunom kamate. Praktičnim primjerom stambenog kredita demonstrirani su iznosi mjesečne glavnice, kamate i ukupnog anuiteta za kredit od deset godina u iznosu od sto trideset tisuća eura i kamatnom stopom pet cijelih devedeset dva posto. Također je prikazan izračun za gotovinski kredit u trajanju od pet godina u iznosu od dvadeset tisuća eura i kamatnom stopom od sedam cijelih dva posto.

Praktični dio rada temeljio se na korištenju web tehnologija *HTML*, *CSS* i *Javascript* upotpunjenim dodatnim bibliotekama *Bootstrap* i Font *Awesome*. Pomoću *HTML*-a definirana je struktura *web*-aplikacije i njen osnovni sadržaj. Kako bi se korisniku omogućilo intuitivno korištenje sučelja, *HTML* je nadopunjen *CSS* prilagodbama. *CSS* je omogućio točan prikaz aplikacije na zaslonima različite veličine. *Web*-aplikacija također je uljepšana dodavanjem ikona iz Font *Awesome* biblioteke. Navigacija i podnožje stranice definirani su *Bootstrap* bibliotekom koja nudi mnoge unaprijed pripremljene isječke koda, ali zbog toga ju je teško prilagoditi specifičnim slučajevima. U konačnici je *Javascript* korišten za provođenje matematičkih izračuna i generiranje elemenata otplatnog plana.

Funkcionalnost *web*-aplikacije mogla bi se unaprijediti dodatnim opcijama unosa za korisnika. Aplikacija pruža podatke informativnog karaktera, ali trenutno ponuđeni izračuni bi se mogli približiti stvarnosti dodavanjem interkalarne kamate, anticipativnog izračuna, troškova solemnizacije kredita, osiguranja i troškova vođenja računa u banci. Funkcionalnost koju bi budući radovi mogli implementirati u postojeću strukturu je izračun kreditne sposobnosti korisnika. Struktura programskog koda mogla bi se prilagoditi kako bi omogućila lakše održavanje i buduće promjene. Trenutno prisutan *Javascript* kod bilo bi moguće minimizirati i time smanjiti vrijeme potrebno za učitavanje aplikacije.

Sažetak

Web-aplikacija za izračun rate kredita te otplatnog plana izrađena je koristeći *web* tehnologije *HTLM*, *CSS* i *Javascript* uputpunjene bibliotekama *Bootstrap* i Font *Awesome*. Struktura aplikacije definirana je *HTML* kodom dok je izgled prilagođen za različite veličine zaslona upotrebom *CSS* koda. *Javascript* se koristi za izračun podataka otplatnog plana i za ažuriranje korisničkog sučelja prilikom odabira parametra kredita. Teorijski su objašnjeni osnovni ekonomski pojmovi vezani za strukturu kredita kao i korištene tehnologije. Praktičnim primjerom demonstrirano je korištenje *web* sučelja na temelju informativnog izračuna stambenog i gotovinskog kredita. U konačnici su opisani ključni dijelovi programskog koda i njihov utjecaj na funkcionalnosti i izgled aplikacije.

Ključne riječi: Kredit, kamata, anuitet, kreditni kalkulator, *web* tehnologije, *HTML*, *CSS*, *Javascript*.

Abstract

The web application for calculating loan installments and repayment plans is developed using web technologies HTML, CSS, and JavaScript, complemented by the Bootstrap and Font Awesome libraries. The structure of the application is defined by HTML code, while the appearance is adapted for different screen sizes using CSS code. JavaScript is used to calculate repayment plan data and update the user interface when selecting loan parameters. The basic economic concepts related to the structure of loans and the technologies used are theoretically explained. A practical example demonstrates the use of the web interface based on an informative calculation of housing and cash loans. Finally, the key parts of the program code and their impact on the functionality and appearance of the application are described.

Keywords: Loan, interest, annuity, loan calculator, web technologies, HTML, CSS, JavaScript.

Životopis

Karlo Kranjčec, rođen je 3. srpnja 1998. godine u Zagrebu. Osnovno obrazovanje pohađa u Kutini u Osnovnoj školi Stjepana Kefelje. Godine 2017. završava srednjoškolsko obrazovanje u općoj gimnaziji u Kutini u Srednjoj školi Tina Ujevića. Godine 2020. upisuje stručni preddiplomski studij Računarstvo u trajanju od tri godine. Tijekom obrazovanja bio je član studentskog ogranka IEEE koji djeluje unutar Sveučilišta Josipa Jurja Strossmayera u Osijeku. Sudjelovao je na natjecanju IEEEXtreme. Također je bio član studentske udruge IAESTE Croatia, osječki ogranak.

Literatura

[1] A. Šegota, Financijska Matematika, Rijeka: Ekonomski fakultet Sveučilišta u Rijeci, 2012., pp. 44, 78-80.

[2] Z. Erjavec i D. Malić, »O obračunu kamata i kredita,« *Matematičko fizički list,* svez. 63, br. 251, pp. 173-178, 2013.

[3] S. Jain, »Bootstrap Tutorial - GeeksforGeeks,« Sanchhaya Education Pvt. Ltd., 19. rujan 2011. [Mrežno]. Dostupno na : https://www.geeksforgeeks.org/bootstrap/. [Pokušaj pristupa 25. kolovoz 2024.].

[4] S. Jain, »HTML Tags – A to Z List,« Sanchhaya Education Pvt. Ltd., 16. veljača 2008. [Mrežno]. Dostupno na: https://www.geeksforgeeks.org/html-tags-a-to-z-list/?ref=shm. [Pokušaj pristupa 24. kolovoz 2024.].

[5] S. Jain, »CSS Tutorial - GeeksforGeeks,« Sanchhaya Education Pvt. Ltd., 18. veljača 2008. [Mrežno]. Dostupno na: https://www.geeksforgeeks.org/css-tutorial/. [Pokušaj pristupa 24. kolovoz 2024.].

[6] H. Refsnes, S. Refsnes i E. J. Refsnes, »CSS Tutorial,« Refsnes Data, 17 srpanj 2011.. [Mrežno]. Dostupno na: https://www.w3schools.com/css/default.asp. [Pokušaj pristupa 24. kolovoz 2024.].

[7] H. Refsnes, S. Refsnes i E. J. Refsnes, »JavaScript Introduction,« Refsnes Data, 8. ožujak 2015.. [Mrežno]. Dostupno na: https://www.w3schools.com/js/js_intro.asp. [Pokušaj pristupa 28. kolovoz 2024.].

[8] T. M. Corporation, »Functions - JavaScript | MDN,« the Mozzila Foundation, 2005.. [Mrežno]. Dostupno na: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions. [Pokušaj pristupa 25. kolovoz 2024.].

[9] H. Refsnes, S. Refsnes i E. J. Refsnes, »HTML Elements,« Refsnes Data, 30. travanj 2010.. [Mrežno]. Dostupno na: https://www.w3schools.com/html/html_elements.asp. [Pokušaj pristupa 24. kolovoz 2024.].