

WEB PROGRAMSKA APLIKACIJA KNJIGE PRIMITAKA I IZDATAKA

Nanić, Zoran

Master's thesis / Diplomski rad

2022

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:145:389890>

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

Download date / Datum preuzimanja: **2024-12-26**



Repository / Repozitorij:

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



Sveučilište Josipa Jurja Strossmayera u Osijeku
Ekonomski fakultet u Osijeku
Diplomski studij Poslovna informatika

Zoran Nanić

**WEB PROGRAMSKA APLIKACIJA KNJIGE PRIMITAKA I
IZDATAKA**

Diplomski rad

Osijek, 2022.

Sveučilište Josipa Jurja Strossmayera u Osijeku
Ekonomski fakultet u Osijeku
Diplomski studij Poslovna informatika

Zoran Nanić

**WEB PROGRAMSKA APLIKACIJA KNJIGE PRIMITAKA I
IZDATAKA**

Diplomski rad

Kolegij: Razvoj distribuiranih i Web aplikacija

JMBAG: 0010210386

email: zonanic@efos.hr

Mentor: prof. dr. sc. Branimir Dukić

Osijek, 2022.

Josip Juraj Strossmayer University of Osijek
Faculty of Economics in Osijek
Graduate Study Business Informatics

Zoran Nanić

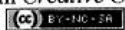
**WEB APPLICATION FOR BOOK OF INCOME AND
EXPENDITURES**

Graduate paper

Osijek, 2022.

IZJAVA

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

1. Kojom izjavljujem i svojim potpisom potvrđujem da je DIPLOMSKI
(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: ZORAN NAMIC

JMBAG: 0010210396

OIB: 71912279115

e-mail za kontakt: zoran.namic@efos.hr

Naziv studija: POSLOVNA INFORMATIKA

Naslov rada: WEB PROGRAMSKA APLIKACIJA KNJIGE PRIMITAKA I IZDATAKA

Mentor/mentorica diplomskog rada:

U Osijeku, 17.3.2022. godine

Potpis Zoran Namic

SAŽETAK

Sukladno pozitivnim zakonskim propisima obrtnici su dužni voditi evidenciju o svojim primitcima i izdatcima u knjizi primitaka i izdataka. U današnje se vrijeme za vođenje takve evidencije najčešće upotrebljavaju računalni programi. No, zbog relativno visokih troškova, posebice malim obrtima ili obrtima kada su u začetku poslovanja, skupa programska rješenja koja podrazumijevaju nabavku uz programske i skupe strojne opreme te redovito održavanje programske i strojne opreme, nisu opcija. Stoga se, kao alternativa skupim rješenjima nameću jednostavne web programske aplikacije kojima obrtnici mogu ispuniti svoje zakonske obveze vezane za vođenje poslovnih knjiga, pa tako i knjige primitaka i izdataka. S obzirom na kontinuiranu potrebu razvoja programskih rješenja za obrtničku evidenciju, posebno onih jednostavnijih i prilagodljivih potrebama malih obrta i obrta u začetku, provedeno je istraživanje radi sagledavanja mogućnosti kreiranja upotrebljivog programskog rješenja knjige primitaka i izdataka načinjene na web platformi. Tijekom istraživanja istražene su popularne i besplatne tehnologije za razvoj programskih web aplikacija kao što su PHP, HTML, CSS i MySQL. Da bi se moglo pristupiti izradi upotrebljivog programskog rješenja za knjigu primitaka i izdataka proučena je bitna zakonska regulativa vezanu za vođenje knjige primitaka i izdataka, kao i prednosti i nedostatke postojećih programskih rješenja na tržištu, kako bi se definiralo optimalno programsko rješenje. U konačnici rezultati su istraživanja pokazali kako se korištenjem mrežnih tehnologija može razviti web programska aplikacija kojom obrtnici mogu ispuniti zakonsku obvezu vezanu uz vođenje knjige primitaka i izdataka. Samo je istraživanje provedeno u dvije faze. Desk istraživanjima je sagledana zakonska regulativa i web tehnologije pogodne za razvoj programskog rješenja knjige primitaka i izdataka, dok su u okviru primarnih istraživanja sagledana raspoloživa postojeća programska rješenja namijenjena vođenju knjige primitaka i izdataka kako bi se sagledala njihova funkcionalnost i otkrile prednosti i mane, a u drugom koraku u okviru primarnih istraživanja modelirana je, a potom i kodirana te testirana programska aplikacija knjige primitaka i izdataka. U istraživanju je primijenjeno više znanstvenih metoda među kojima su svakako najznačajnije metoda apstrakcije, metoda klasifikacije, metoda generalizacije, metoda specijalizacije, metoda kompozicije, metoda abrogacije, metoda analogije, metoda kauzalnog zaključivanja, te metoda deskriptivnog modeliranja.

Ključne riječi: Knjiga primitaka i izdataka, Obrazac KPI, PHP, MySQL, web aplikacija

ABSTRACT

In accordance with positive legal regulations, craftsmen are obligated to keep records of their income and expenditures in the book of income and expenditures. Nowadays, computer programs are most often used to keep such records. However, due to relatively high cost, especially for small craft businesses or craftsmanship that are starting their business, expensive software solutions that imply the purchase of expensive equipment and regular maintenance of said equipment, are not an option. Therefore, as an alternative, simple web applications are being imposed, which can be used to fulfill legal obligations related to bookkeeping, including the book of income and expenditures. Given the continuous need for the development of software solutions for craftsmanship business, especially the simple and adaptable to meet the needs of small and starting craftsmanship businesses, research had been carried out to see the possibility to create usable software solutions for the book of income and expenditures on a web platform. During the research, popular and free technologies for the development of web applications such as PHP, HTML, CSS, and MySQL had also been researched. In order to be able to create a usable software solution for the book of income and expenditures, the important legal regulations related to keeping the book had been studied, as well as the advantages and disadvantages of existing software solutions on the market, to define the optimal software solution. In the end, the results of the research showed that by using web technologies, a web application can be developed that allows craftsmen to fulfill the legal obligation related to keeping a book of income and expenditures. The research itself was conducted in two phases. In desk research, legal regulations and web technologies suitable for the development of software solutions for the book of income and expenditures were reviewed, while in the framework of primary research, the program application of the book of income and expenditures was modeled, and then coded and tested. Several scientific methods were applied in the research, among which the most important are the abstraction method, the classification method, the generalization method, the analogy method, the casual method, and the descriptive modeling method.

Keywords: book of income and expenditures, Form KPI, PHP, MySQL, web application

SADRŽAJ

1. UVOD.....	1
2. METODOLOGIJA ISTRAŽIVANJA.....	3
3. POSLOVANJE OBRTA.....	5
3.2. Uvjeti obavljanja obrta.....	6
3.3. Posebnosti obrtničkog poslovanja.....	7
4. FINANCIJSKI I RAČUNOVODSTVENI ASPEKT OBRTNIČKOG POSVLOANJA.....	9
4.1. Primici i izdaci.....	9
4.2. Dohodak.....	10
4.3. Osobni odbitak.....	10
4.4. Porezna osnovica i porez na dohodak.....	11
4.4.1. Godišnja porezna prijava.....	12
4.4.2. Predujmovi poreza na dohodak.....	13
4.4.3. Utvrđivanje paušalnog poreza na dohodak.....	13
4.5. Knjiga primitaka i izdataka te ostale poslovne knjige.....	14
4.5.1. Popis dugotrajne imovine.....	15
4.5.2. Evidencija o tražbinama i obvezama.....	15
4.5.3. Knjiga prometa.....	15
4.5.4. Knjiga primitaka i izdataka.....	16
5. PROGRAMSKE APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA.....	18
5.1. Prednosti i nedostaci postojećih aplikacija na tržištu.....	18
5.1.1. Najčešći nedostaci.....	18
5.1.2. Najčešće prednosti.....	19
5.2. Segment tržišta koji nije pokriven postojećim programskim aplikacijama.....	19

6. ALATI KORIŠTENI ZA IZRADU WEB APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA.....	20
6.1. PHP.....	20
6.2. MySQL.....	22
6.3. SQL.....	22
6.4. HTML.....	22
6.5. CSS.....	23
6.6. Ostali alati.....	23
7. MODEL BAZE PODATAKA APLIKACIJE.....	26
8. STRUKTURA APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA.....	29
8.1. Mapa config.....	29
8.2. Mapa inc.....	31
8.3. Web aplikacije iz mape KPI i funkcije iz functions.php.....	33
8.3.1. Datoteka index.php.....	33
8.3.2. Datoteka registracija.php.....	34
8.3.3. Datoteka prijava.php.....	40
8.3.4. Datoteka profil.php.....	44
8.3.5. Datoteka unos.php.....	47
8.3.6. Datoteka izvjestaji.php.....	49
8.3.7. Datoteka KPI.php.....	50
9. MOGUĆNOSTI DALJNJEG RAZVOJA PROGRAMSKE APLIKACIJE I MOGUĆA POBOLJŠANJA.....	56
10. ZAKLJUČAK.....	57
LITERATURA.....	59

1. UVOD

Sukladno pozitivnim zakonskim propisima pod obrtom se smatra samostalno i trajno obavljanje gospodarskih djelatnosti, a te djelatnosti obavljaju fizičke osobe. Iskustva pokazuju kako se za poslovanje putem obrta obično odlučuju fizičke osobe koje žele pokrenuti poduzetnički pothvat no nisu spremne angažirati velika financijska sredstva za pokretanje poslovne aktivnosti sukladno Zakonu o trgovačkim društvima. Nerijetko se radi o pojedincima koji imaju određene stručne kvalifikacije i koji barem u startu nisu spremni upošljavati druge osobe da rade za njih. Naime, za razliku od društava kapitala, a koja se osnivaju sukladno Zakonu o trgovačkim društvima, obrti su profesionalno orijentirani subjekti koji rade u pravilu stručne poslove manjeg obujma. Prema nekim shvaćanjima, trovačka društva bi trebala biti poslovni subjekti orijentirani prema visoko serijskoj, odnosno masovnoj proizvodnji, dok bi se obrti trebali baviti zanatskim poslovima kod kojih nema podjele rada ili je ta podjela minimalna, za razliku od masovne proizvodnje. U praksi postoje izuzetci od ovog načela, no generalno gledano, obrti su mali poslovni subjekti kojima su za vođenje poslovnih knjiga i kreiranje financijskih izvješća potrebna jednostavna i jeftina programska rješenja

Jedna je od knjiga koja se koristi u obrtničkoj evidenciji knjiga primitka i izdataka. To je knjiga u koju obrtnici moraju unositi svoje primitke i izdatke sukladno Zakonu o porezu na dohodak i Pravilniku o porezu na dohodak. Knjiga primitaka i izdataka može se voditi ručno, no najpraktičnije rješenje su računalni programi namijenjeni za tu svrhu jer se time štedi na vremenu, smanjuje mogućnost pogreške i povećava razina ekonomičnosti i rentabilnosti obrade podataka te urednosti poslovnih knjiga. Osim navedenog, uobičajeno korisnici kupljenih programskih rješenja za obrtničku evidenciju ne moraju voditi računa o mogućim promjenama zakonskih regulativa koje uređuju izgled i sadržaj poslovnih knjiga i financijskih izvještaja, promjene poreznih stopa i dr. jer za njih te poslove uobičajeno obavlja onaj tko je programsko rješenje prodao ili iznajmio obrtniku.

Danas, uz sveprisutnost uređaja koji se spajaju na internet, mrežne se tehnologije nameću kao idealno rješenje za izradu programskih rješenja za obrtničku evidenciju. Prednost je upotrebe mrežnih ili web tehnologija što one korisnike ne ograničavaju na jednu platformu, već korisnici mogu koristiti bilo koji uređaj sa pristupom internetu i instaliranim internet preglednikom za pristup željenim programskim aplikacijama. Uz to upotrebom se pohrane podataka na mrežnim podatkovnim nositeljima podataka povećava sigurnost čuvanja poslovnih knjiga jer osim što ih korisnik može čuvati lokalno i u fizičkom obliku, iste se nalaze i na web poslužitelju ili cloud servisu.

S obzirom na kontinuiranu potrebu razvoja programskih rješenja za obrtničku evidenciju, posebno onih jednostavnijih i prilagodljivih potrebama malih obrta, provedeno je istraživanje radi sagledavanja mogućnosti kreiranja upotrebljivog programskog rješenja knjige primitaka i izdataka načinjene na web platformi. Tijekom istraživanja istražene su popularne i besplatne tehnologije za razvoj programskih web aplikacija kao što su PHP, HTML, CSS i MySQL. Da bi se moglo pristupiti izradi upotrebljivog programskog rješenja za knjigu primitaka i izdataka proučena je bitna zakonska regulativa vezanu za vođenje knjige primitaka, kao i prednosti i nedostatke postojećih programskih rješenja na tržištu, kako bi se definiralo optimalno programsko rješenje.

2. METODOLOGIJA ISTRAŽIVANJA

Obrtnici su sukladno pozitivnim zakonskim propisima dužni voditi evidenciju o primitcima i izdatacima u knjizi primitaka i izdataka. Za vođenje takve evidencije najčešće se upotrebljavaju računalni programi, no mnogim obrtnicima nije isplativo plaćati takva programska rješenja, te takva rješenja često zahtjevaju naprednije znanje o vođenju navedene poslovne evidencije kao i o korištenju samog programskog rješenja. Obrtnici su često ograničeni i opremom i tehnologijom koju mogu koristiti. Stoga se jednostavna web programska aplikacija nameće kao rješenja kojima bi obrtnici mogli ispuniti svoje zakonske obveze vezane za vođenje knjige primitaka i izdataka.

Temeljem uočenog problema definirana je sljedeća hipoteza:

Korištenjem mrežnih tehnologija može se razviti programska aplikacija kojom će obrtnici moći ispuniti zakonske obveze vezane za vođenje knjige primitaka i izdataka.

Temeljem izloženog problema istraživanja i definirane hipoteze postavljeni su sljedeći ciljevi istraživanja:

1. Istražiti zakonsku regulativu vezanu za vođenje knjige primitaka i izdataka
2. Istražiti i opisati prednosti i nedostatke postojećih programskih rješenja na tržištu
3. Izraditi web programsku aplikaciju knjige primitaka i izdataka

U istraživanju je korišten veći broj znanstvenih metoda. Rad je strukturiran deduktivno jer se temeljem uočenog problema i postavljene opće hipoteze istu dokazivalo kroz provedeno istraživanje te kreiran model u vidu web programske aplikacije knjige primitaka i izdataka. Istraživanje je provedeno u dvije faze. Desk istraživanjima je sagledana zakonska regulativa i web tehnologije pogodne za razvoj programskog rješenja knjige primitaka i izdataka, dok su u okviru primarnih istraživanja sagledana raspoloživa postojeća programska rješenja namijenjena vođenju knjige primitaka i izdataka kako bi se sagledala njihova funkcionalnost i otkrile prednosti i mane, a u drugom koraku u okviru primarnih istraživanja modelirana je, a potom i kodirana te testirana programska aplikacija knjige primitaka i izdataka. Najvažnije znanstvene metode koje su korištene u istraživanju su metoda apstrakcije, metoda klasifikacije, metoda generalizacije, metoda specijalizacije, metoda kompozicije, metoda abrogacije, metoda analogije, metoda kauzalnog zaključivanja, te metoda deskriptivnog modeliranja.

U okviru desk istraživanja proučena je raspoloživa znanstvena i stručna tiskana i digitalna literatura. Za potrebe provođenja primarnih istraživanja korištena je klijent-poslužiteljska mrežna infrastruktura te razvoji alati web programskih aplikacija te baza podataka PHP, HTML, CSS i MySQL. Rezultat istraživanja je web programska aplikacija knjige primitaka i izdataka.

3. POSLOVANJE OBRTA

Zakonom o obrtu (NN 143/2013, čl.1.) koji je proglašenim 27. studenog 2013. godine: „uređuju se sadržaj, način i uvjeti za obavljanje obrta, vrste obrta, prava i obveze obrtnika, obrazovanje i osposobljavanje za obavljanje vezanih obrta, institut domaće radinosti i sporednog zanimanja, pravna osoba koja obavlja obrt, organiziranost obrta, nadzor nad primjenom Zakona i druga pitanja važna za obavljanje obrta.”¹ Prema Zakon o obrtu (NN 143/2013, čl.2.st.1,2) obrt je samostalno i trajno obavljanje dopuštenih gospodarskih djelatnosti od strane fizičkih osoba sa svrhom postizanja dobiti ili dohotka koji se ostvaruje proizvodnjom, prometom ili pružanjem usluga na tržištu. Iznimno, obrt može obavljati i pravna osoba koja obavlja djelatnost vezanog obrta ako izvodi naukovanje.² U smislu predmetnog Zakona dopuštena gospodarska djelatnost je ona djelatnost koja nije zakonom zabranjena. Podzakonski akt kojim se prepoznaje dopuštenost neke gospodarske djelatnosti je Odluka o nacionalnoj klasifikaciji djelatnosti 2007. - NKD 2007. (NN 58/07). Za neke djelatnosti postoje i posebni propisi kojima su uređena područja tih istih djelatnosti kao što su primjerice Zakon o trgovini, Zakon o ugostiteljskoj djelatnosti i slično.³ Osoba koja obavlja jednu ili više djelatnosti obrta u svoje ime i za svoj račun, a pri tome se može koristiti i radom drugih osoba naziva se obrtnik.⁴

3.1. Vrste obrta

Postoje tri vrste obrta:⁵

- slobodni obrti,
- vezani oprti i
- povlašteni obrti.

Slobodni obrti su oni obrti za čije obavljanje nije potreban majstorski ispit ili ispit o stručnoj osposobljenosti, kao niti srednja stručna sprema, srednja školska sprema ili viši stupanj obrazovanja. Za obavljanje djelatnosti slobodnih obrta obrtnik mora udovoljiti jedino općim uvjetima koji su

¹ Narodne novine: Zakon o obrtu. Narodne Novine d.d., Zagreb, br. 143/13, 127/19, čl.1.

² Idem, čl.1, 2.

³ Čebetarević, V. et al., (2006). Poslovanje obrtnika i slobodnih zanimanja, 4th ed., Zagreb: RRiF-plus d.o.o., str. 6

⁴ Narodne novine: Idem, čl.4.

⁵ Amidžić-Peročević, K. et al.: Poslovanje obrtnika 2002., Hrvatska zajednica računovođa i financijskih djelatnika, Zagreb, 2002., str. 1.

propisani Zakonom o obrtu.⁶ Zakon o obrtu, kao niti podzakonski propisi, ne navode obrte koji se smatraju slobodnima, već se istima smatraju oni obrti koji nisu na popisu vezanim i povlaštenih obrta.⁷ Vezani obrti su oni obrti za čije se obavljanje, uz opće uvjete, traži i ispit o stručnoj osposobljenosti, kao i obrti koji zahtijevaju položen majstorski ispit, odnosno određena stručna sprema. Primjeri vezanih obrta su: automehaničar, stolar, frizer, slastičar, urar i sl.⁸ Povlašteni obrti su vrsta obrta za čije je obavljanje potrebno ishoditi povlasticu koju izdaje nadležno ministarstvo ili neko drugo nadležno tijelo ovisno o vrsti djelatnosti. Primjeri povlaštenih obrta su: slatkovodni ribar, morski ribar, rudarski tehničar, puškar, zaštitar-tehničar i sl.). U Pravilniku o vezanim i povlaštenim obrtima i načinu izdavanja povlastica (NN 42/08.) objavljen je popis vezanih i povlaštenih obrta te je njime utvrđen i način dobivanja povlastice.⁹ Posebnu kategoriju obrta čine tradicijski i umjetnički obrti čiji su uvjeti uređeni Pravilnikom o tradicijskim, odnosno umjetničkim obrtima (NN 112/07) kao i način za stjecanje statusa tradicijskog ili umjetničkog obrta. Tim Pravilnikom ovi obrti su opisani na sljedeći način: „Tradicijski obrti su obrti za koje je potrebno posebno poznavanje zanatskih vještina i umijeća u obavljanju djelatnosti te koji se obavljaju pretežnim udjelom ručnog rada, a koji se tehnikama proizvodnje i rada, namjenom i oblikom oslanjaju na obrasce tradicijske kulture, pa u tom smislu mogu simbolizirati lokalni, regionalni ili nacionalni identitet. Umjetnički obrti su obrti koji se odlikuju proizvodima i uslugama visoke estetske vrijednosti, dizajnom, likovnim i drugim rješenjima uz naglašenu kreativnost i individualnost majstora obrta –umjetnika.“¹⁰

3.2. Uvjeti za obavljanje obrta

Opći su uvjeti iz Zakona o obrtu koje fizička osoba mora ispuniti kako bi obavljala obrt:¹¹

1. da joj pravomoćnom sudskom presudom, rješenjem o prekršaju ili odlukom Suda časti Hrvatske obrtničke komore nije izrečena sigurnosna mjera ili zaštitna mjera zabrane obavljanja djelatnosti dok ta mjera traje
2. da ima pravo korištenja prostora ako je to potrebno za obavljanje obrta.

⁶ Ćebetarević, V. et al., Poslovanje obrtnika i slobodnih zanimanja, RRiF-plus d.o.o., Zagreb, 2006., str. 11.

⁷ Amidžić-Peročević, K. et al.: Idem, str. 2.

⁸ Amidžić-Peročević, K. et al.: Idem, str. 2.

⁹ Ćebetarević, V. et al.: Idem, str. 11.

¹⁰ Narodne novine: Pravilnik o tradicijskim, odnosno umjetničkim obrtima. Narodne Novine d.d., Zagreb, br. 112/07.

¹¹ Narodne novine: Zakon o obrtu. Narodne Novine d.d., Zagreb, br. 143/13, 127/19, čl.1.

Odabrani posebni uvjeti koje treba ispuniti obrtnik:¹²

1. Odgovarajuća stručna osposobljenost, odgovarajuće srednje strukovno obrazovanje ili položeni majstorski ispit ukoliko obrtnik obavlja vezani obrt.
2. Ukoliko fizička osoba ne ispunjava iznad navedeni uvjet, može obavljati vezani obrt ako zaposli osobu u punom radnom vremenu koja zadovoljava taj uvjet.
3. Potreba je posebna zdravstvena sposobnost za djelatnosti za koje je to propisano zakonom.
4. Za obavljanje povlaštenog obrta potreba je važeća povlastica
5. Prostor u kojem se obrt obavlja, oprema i sredstva moraju udovoljavati minimalnim tehničkim i ostalim uvjetima.
6. Za obavljanje bilo koje vrste obrta potrebno je imati obrtnicu.

3.3. Posebnosti obrtničkog poslovanja

Obrt se može obavljati i u stambenom prostoru ako taj prostor udovoljava općim i posebnim uvjetima. Djelatnosti koje se mogu obavljati u stambenom prostoru navedene su u Pravilniku o obrtima koji se mogu obavljati u stambenim prostorijama (NN 101/1995). Primjeri takvih obrta: popravak nakita i satova, prevođenje, računovodstveni poslovi, pružanje savjeta o računalnoj opremi (hardware-u), savjetovanje i pribavljanje programske opreme (software-a) i slično.¹³ S obzirom na široku mogućnost obavljanja obrta u stambenom prostoru, osim navedenih djelatnosti, moguće je obavljati i druge ako za njihovo obavljanje nije potreba izmjena postojećih uvjeta korištenja tog stambenog prostora.¹⁴ Poslovanje se može obavljati iz više odvojenih prostora, odnosno izdvojenih pogodna, koji se nalaze izvan sjedišta obrta. Svaki izdvojeni pogon u slobodnom obrtu mora imati imenovanog poslovođu, dok vezani i povlašteni obrti moraju imati stručnog poslovođu.¹⁵

Obrtniku u obavljanju djelatnosti smiju pomagati članovi obiteljskog kućanstva bez obveze zasnivanja radnog odnosa. U slučaju odsutnosti obrtnika kod vezanih obrta, član kućanstva koji pomaže mora ispunjavati uvjet stručne osposobljenosti položenog majstorskog ispita ili

¹² Hrvatska obrtnička komora: Vodič za poslovanje u obrtu, 2021., str 6.

¹³ Narodne novine: Pravilnik o obrtima koji se mogu obavljati u stambenim prostorijama. Narodne Novine d.d., Zagreb, br. 101/95.

¹⁴ Hrvatska obrtnička komora: Vodič za poslovanje u obrtu, 2021., str. 7.

¹⁵ Idem, str. 7.

odgovarajućeg srednjeg strukovnog obrazovanja. Potrebno je ispuniti i posebne zdravstvene uvjete ako je to propisano zakonom za gospodarsku djelatnost kojoj se obrtnik bavi.¹⁶

Obrt se može obavljati sezonski u trajanju ne duljem od devet mjeseci unutar kalendarske godine. Izvan trajanja rada sezonskog obrta obrtnik ne ostvaruje pravo na zdravstvenu zaštitu iz zdravstvenog osiguranja, niti mu se računa radni staž.¹⁷

Obrt može zajednički obavljati više fizičkih osoba. Ugovorom se određuju međusobni odnosi tih osoba. Obrt nastavlja poslovati i ako jedna od tih osoba istupi iz zajedničkog obrta.¹⁸

Osoba koje je u radnom odnosu može obavljati i obrt. Obrtnik radom u obrtu ostvaruje prava iz radnog odnosa (kao što su pravo na mirovinsko i zdravstveno osiguranje). Ako obrtnik uz radni odnos obavlja i obrt obveznik je doprinosa i kao obrtnik i kao zaposlenik u nekom drugom gospodarskom subjektu u kojem je zaposlen. Doprinosi u obrtu se plaćaju prema ostvarenom dohotku.¹⁹

Umirovljenici koji žele obavljati obrt mogu to činiti, no za vrijeme obavljanja obrta im se ne isplaćuje mirovina. U određenim slučajevima se obrtnicima dopušta obavljanje obrta bez prestanka isplate mirovine, a to pravo imaju ako: obavljaju domaću radinost kod kuće (izrada proizvoda osobnim radom) ili obavljaju sporedno zanimanje kod kuće ili kod naručitelja usluga. Osim toga mogu biti zaposleni u nečijem obrtu do najviše pola radnog vremena ili pomagati članu kućanstva koji je obrtnik. Korisniku invalidske mirovine se kod obavljanja obrta ne ukida isplata mirovine, već se određuje nova visina mirovine.²⁰

Kod osnivanja obrta nema troškova osnivanja niti temeljnog kapitala. No, obrtnik odgovara za obveze obrta svom svojom imovinom.²¹

¹⁶ Idem, str. 7.

¹⁷ Idem, str. 8.

¹⁸ Idem, str. 11.

¹⁹ Idem, str. 11.

²⁰ Idem, str. 12.

²¹ Idem, str. 14.

4. FINANCIJSKI I RAČUNOVODSTVENI ASPEKT OBRTNIČKOG POSLOVANJA

Kao i svi drugi oblici poslovanja tako su i obrti u obvezi voditi poslovne knjige i izvješćivati držacu o financijsko- materijalnom stanju. Vođenjem poslovnih knjiga obrta obavlja se u okviru obrtničke evidencije.

4.1. Primitci i izdatci

Primitci su sva dobra koja su poreznom obvezniku pritekla u određenom poreznom razdoblju, a to mogu biti u obliku novca, stvari, materijalnih prava, usluga i drugo.²² Izdaci su svi odljevi dobara s novčanom vrijednošću izvršeni radi ostvarivanja ili osiguranja primitaka.²³ Primitci i izdatci utvrđuju se primjenom načela blagajne.²⁴ Odnosno, primitci se bilježe nakon što su naplaćeni, a izdatci nakon što su isplaćeni. Izuzetak su izdatci temeljem amortizacije dugotrajne imovine.²⁵

Za obrtnike koji su u obveznici porezna na dodanu vrijednost, PDV sadržan u primitcima se ne smatra primitkom, a PDV sadržan u izdalcima ne smatra se izdatkom, već pretporezom. Shodno tome, PDV nema utjecaj na dohodak za obrtnike koji su porezni obveznici poreza na dodanu vrijednost. Obrtnici koji nisu u sustavu PDV-a ne prikazuju PDV u izdalcima kao pretporez, već PDV ulazi u izdatke.²⁶

Izdatci koji nisu u izravnoj svezi s obavljanjem samostalne djelatnosti ne smatraju se poslovnim izdalcima i stoga ne utječu na dohodak.²⁷ Također, izdatci koji se porezno ne priznaju prilikom utvrđivanja dohotka od samostalne djelatnosti su:²⁸

- 50% troškova reprezentacije u visini troškova nastalih iz poslovnih odnosa s poslovnim partnerom (Primjerice: ugošćenja, darovi, izdatci za odmor, i rekreaciju i sl.)
- PDV na vlastitu potrošnju

²² Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 115/16, čl.11.st.1.

²³ Idem, čl.11.st.2.

²⁴ Idem, čl.11.st.3.

²⁵ Čebetarević, V. et al.: Idem, str. 223.

²⁶ Idem.

²⁷ Narodne Novine: Idem, čl.11.st.1.

²⁸ Idem, čl.33.st.1.

- primitci poduzetnika koji obavljaju samostalnu djelatnost po osnovi naknada, potpora i nagrada, iznad propisanih iznosa
- dnevnica i troškovi službenog putovanja poduzetnika iznad propisanih iznosa i/ili
- 50% izdataka u svezi s vlastitim ili unajmljenim osobnim motornim vozilima kao i u svezi s drugim sredstvima za osobni prijevoz ako se po osnovi korištenja tih sredstava ne utvrđuje plaća ili drugi dohodak. Iznimka su izdatci za osiguranje prijevoznih sredstava koji se smatraju poslovnim izdancima.

4.2. Dohodak

Dohodak je razlika između utvrđenih primitaka i izdataka iz istog poreznog razdoblja²⁹ Dohodak od samostalne djelatnosti je dohodak od obrta i s obrtom izjednačenih djelatnosti, dohodak os slobodnih zanimanja te dohodak od poljoprivrede i šumarstva.³⁰ Dohodak se utvrđuje kao godišnji ili konačni dohodak.³¹ Godišnjim dohotkom se smatra dohodak ostvaren u primitcima od nesamostalnog rada, samostalne djelatnosti i drugim primitcima koji se ne smatraju konačnima.³² Konačnim dohotkom se smatra dohodak ostvaren u primitcima od imovine, imovinskih prava, kapitala i drugih primitaka koji se smatraju konačnima.³³

Dohodak od samostalne djelatnosti se umanjuje za iznos plaća novozaposlenih osoba, iznos države potpore za obrazovanje, potpore za izvođenje praktične nastave, iznos izdataka za istraživanje i razvoj te preneseni gubitak.³⁴

4.3. Osobni odbitak

Osobni odbitak je neoporezivi dio dohotka. Time se umanjuje porezna osnovica od koje se plaća porez. Svrha osobnog odbitka je sprječavanje oporezivanja dijela dohotka koji služi za zadovoljavanje

²⁹ Idem, čl.12.st.1.

³⁰ Idem, čl.28.

³¹ Idem, čl.12.st.2.

³² Idem, čl.12.st.3.

³³ Idem, čl.12.st.4.; Narodne novine: Zakon o izmjenama i dopunama Zakona o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 106/18, čl.5.

³⁴ Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 115/16, čl.18.st.2.

osnovnih životnih potreba (egzistencijalni minimum). Svaki porezni obveznik ima pravo na osobni odbitak za vlastito uzdržavanje, ali za uzdržavanje članova uže obitelji.³⁵ Osnovica osobnog odbitka iznosi 2.500,00 kn. Osobni odbitak i dijelovi odbitka za uzdržavane članove obitelji i djecu te invalidnost ili tjelesno oštećenje izračunavaju se primjenom osnovice osobnog odbitka i propisanih koeficijenata.³⁶ Također, osobni odbitak se uvećava i za iznose plaćene za obavezno zdravstveno osiguranje ako porezni obveznik nije drugačije osiguran, sve do visine propisanog obveznog doprinosa za zdravstveno osiguranje.³⁷ Nadalje, osobni odbitak se uvećava i za darovanja dana u tuzemstvu u novcu i u naravi, u kulturne, znanstvene, odgojno-obrazovne, zdravstvene, sportske, vjerske i humanitarne svrhe, udrugama i drugim osobama koje obavljaju te djelatnosti, do visine 2% primitaka za koje je u tekućoj godini podnesena godišnja porezna prijava i utvrđen godišnji porez na dohodak ili ako je isti utvrđen provođenjem posebnog postupka utvrđivanja visine poreza i prireza na dohodak. Iznimno se osobni odbitak može uvećati i za darovanja vrijednosti iznad propisane visine ako su darovanja dana prema odlukama nadležnih ministarstva i financiranju i provedbi posebnih programa i akcija.³⁸

4.4. Porezna osnovica i porez na dohodak

U smislu Zakona o porezu na dohodak (NN 115/2016, čl.2.), porezni obveznik je svaka fizička osoba koja ostvaruje dohodak, a u slučaju da više fizičkih osoba zajedno ostvaruje dohodak, porezni obveznik je svatko od njih zasebno za svoj udio u ostvarenom zajedničkom dohotku. Porezni obveznik je i nasljednik za porezne obveze koje proizlaze iz dohotka koje je ostavitelj ostvario do svoje smrti kao i za dohodak koji mu pritječe iz naslijeđenih izvora dohotka.³⁹

Dohodci koji se oporezuje prema Zakonu o porezu na dohodak (NN 115/2016, čl.5.st.2) i Zakonu o izmjenama i dopunama Zakona o porezu na dohodak (NN 106/2018, čl.1.) su:⁴⁰

- dohodak od nesamostalnog rada (plaća, mirovina),
- dohodak od samostalne djelatnosti,

³⁵ Amidžić-Peročević, K. et al.: Idem, str. 45.

³⁶ Narodne Novine: Idem, čl.14.st.1., 2.

³⁷ Narodne Novine: Idem, čl.15.st.1.

³⁸ Narodne Novine: Idem, čl.15.st.2., 3.

³⁹ Narodne Novine: Idem, čl.2.

⁴⁰ Narodne Novine: Idem, čl.5.st.2.; Narodne novine: Zakon o izmjenama i dopunama Zakona o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 106/18, čl.1.

- dohodak od imovine i imovinskih prava,
- dohodak od kapitala i/ili
- drugi dohodak.

Porez na dohodak utvrđuje se i plaća za jednu kalendarsku godinu, odnosno porezno razdoblje. Iznimno, porezno razdoblje može trajati kraće ako rezident u kalendarskoj godini postane nerezident ili obratno, te u slučajevima rođenja ili smrti poreznog obveznika.⁴¹ Godišnju poreznu osnovicu poreza na dohodak čini ukupan iznos dohotka od nesamostalnog rada ili samostalne djelatnosti i drugog dohotka koji se ne smatra konačnim, umanjen za osobni odbitak.⁴² Godišnji porez na dohodak plaća se po stopi od 20% na poreznu osnovicu do visine 360.000,00 kn i po stop od 30% na dio porezne osnovice iznad 360.000,00 kn.⁴³ Porezni obveznik koji obavlja samostalnu djelatnost može na vlastiti zahtjev plaćati porez na dobit umjesto poreza na dohodak ili je obvezan to činiti ako ispunjava zakonske uvjete za oporezivanje dobiti na način propisanim zakonom kojim se uređuje porez na dobit.⁴⁴

4.4.1. Godišnja porezna prijava

Godišnju poreznu prijavu je dužan podnijeti porezni obveznik koji ostvari dohodak od samostalne djelatnosti.⁴⁵ Porezni obveznik obvezan je iskazati sve ostvarene porezne dohotke.⁴⁶ Ukoliko porezni obveznik ne podnese poreznu prijavu ili ako su podatci nepotpuni ili netočni, porez na dohodak će se utvrditi procjenom.⁴⁷ Porezni obveznici dužni su poreznu prijavu podnijeti do kraja veljače za prethodno porezno razdoblje.⁴⁸ Oblik i sadržaj porezne prijave propisan je Pravilnikom o porezu na dohodak.⁴⁹

⁴¹ Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 115/16, čl.7.

⁴² Idem, čl.18.st.1.

⁴³ Idem, čl.19.st.1.; Narodne novine: Zakon o izmjenama i dopunama Zakona o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 106/18, čl.8.

⁴⁴ Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 115/16, čl.36.st.1.

⁴⁵ Idem, čl.48.st.1.

⁴⁶ Idem, čl.48.st.3.

⁴⁷ Idem, čl.48.st.4.

⁴⁸ Idem, čl.50.st.2.

⁴⁹ Idem, čl.50.st.3.

4.4.2. Predujmovi poreza na dohodak

Utvrđivanje predujma za porez na dohodak od samostalne djelatnosti utvrđuje se na temelju podataka u podnesenoj godišnjoj prijavi poreza na dohodak tako da se iznos podijeli s brojem mjeseci razdoblja u kojem se djelatnost obavljala. Ako je u poreznoj prijavi bio iskazan i dohodak koji nije od samostalne djelatnosti taj dohodak ne ulazi u obračun predujma. Predujmovi se plaćaju do posljednjeg dana u mjesecu za prethodni mjesec.⁵⁰ U prvoj godini poslovanja se ne plaćaju predujmovi, odnosno sve do podnošenja prve godišnje porezne prijave.⁵¹ Porezna uprava može promijeniti iznos predujma na temelju obavljenog nadzora i očevica ili podataka o poslovanju poreznog obveznika s kojima raspolaže, te na zahtjev poreznog obveznika.⁵²

4.4.3. Utvrđivanje paušalnog poreza na dohodak

Poreznom obvezniku koji obavlja djelatnost obrta ili djelatnost poljoprivrede i šumarstva, a nije u sustavu PDV-a i ne ostvaruje ukupni godišnji primitak veći od iznosa propisanog za obvezni ulazak u sustav PDV-a (300.000,00 kn), dohodak kao i porez na dohodak mogu se utvrđivati i u paušalnom iznosu.⁵³ Porezni obveznik u slučaju paušalnog obračuna poreza na dohodak ne mora voditi poslovne knjige, osim evidencije o prometu.⁵⁴

Prema Pravilniku o paušalnom oporezivanju samostalnih djelatnosti (NN 1/2020) i Pravilniku o izmjenama i dopuni Pravilnika o paušalnom oporezivanju samostalnih djelatnosti (NN 1/2021), godišnji paušalni porez utvrđuje se primjenom porezne stope od 10% te se uvećava za prizrez.⁵⁵ Osnovica se računa kao razlika između primitaka u unaprijed određenom iznosu i porezno priznatih izdataka u visini 85%, ako obveznik ostvari primitke do tog određenog iznosa, na način prokazan u tablici 1.⁵⁶

⁵⁰ Idem, čl.37.st.1, 5.

⁵¹ Idem, čl.33.st.4.

⁵² Idem, čl.37.st.1., 3., 6.

⁵³ Idem, čl.82.st.1.

⁵⁴ Idem, čl.82.st.4.

⁵⁵ Narodne Novine: Pravilnik o izmjenama i dopuni Pravilnika o paušalnom oporezivanju samostalnih djelatnosti. Narodne Novine d.d., Zagreb, br. 1/21, čl.1.

⁵⁶ Narodne Novine: Pravilnik o paušalnom oporezivanju samostalnih djelatnosti. Narodne Novine d.d., Zagreb, br. 1/20, čl.3.st.1.

Tablica 1: Porezna osnovica kod paušalnih obrta (u kn)

Ukupni primitci	Godišnja porezna osnovica
0,00 – 85.000,00	12.750,00
85.000,01 – 115.000,00	17.250,00
115.000,01 – 149.500,00	22.425,00
149.500,01 – 230.000,00	34.500,00
230.000,01 – 300.000,00	45.000,00

Izvor: Pravilniku o paušalnom oporezivanju samostalnih djelatnosti (NN 1/2020, čl.3.st.1)

4.5. Knjiga primitaka i izdataka te ostale poslovne knjige

Fizičke osobe koje obavljaju obrt obvezne su voditi poslovne knjige. Obveznici poreza na dohodak obvezni su voditi poslovne knjige koje su propisane Zakonom o porezu na dohodak i Pravilnikom o porezu na dohodak. Oni obrtnici koji su i obveznici poreza na dodanu vrijednost uz te evidencije moraju voditi i evidencije propisane Zakonom o porezu na dodanu vrijednost i Pravilnikom o porezu na dodanu vrijednost.⁵⁷

Poslovne knjige su osnova i preduvjet za utvrđivanje financijskog rezultata i temelj za izradu financijskih i poreznih izvješća iz kojih proizlazi i izračun porezne obveze. Stoga je važno da se poslovne knjige vode prema kriterijima urednosti, ažurnosti i temeljem vjerodostojnih isprava.⁵⁸

Poslovne knjige koje vode obrtnici su:⁵⁹

- knjiga primitaka i izdataka (Obrazac KPI),
- popis dugotrajne imovine (Obrazac DI),
- knjiga prometa (Obrazac KPR),
- evidencija o tražbinama i obvezama (Obrazac TO).

⁵⁷ Amidžić-Peročević, K. et al.: Idem, str. 157.

⁵⁸ Čebetarević, V. et al.: Idem, str. 187.

⁵⁹ Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 115/16, čl.34.st.2.

Oblik i sadržaj navedenih poslovnih knjiga opisani u Pravilniku o porezu na dohodak (NN 10/2017). Poslovne knjige mogu se voditi kao slobodni listovi, uvezene knjige ili korištenjem računalnih programa za vođenje poslovnih knjiga.⁶⁰

4.5.1. Popis dugotrajne imovine

Popis dugotrajne imovine je evidencija materijalne i nematerijalne imovine čija je nabavna vrijednost veća od 3.500,00 kn (bez PDV-a ako je obrtnik nije u sustavu PDV-a ili s PDV-om ako obrtnik je u sustavu PDV-a) i ako joj je vijek trajanja duži od jedne godine. Svrha Popisa je utvrđivanje amortizacije, odnosno otpisa vrijednosti imovine.⁶¹

4.5.2. Evidencija o tražbinama i obvezama

Evidencija o tražbinama i obvezama je evidencija svih ispostavljenih i primljenih računa, osim onih koji su naplaćeni ili plaćeni na način koji ima obilježja gotovinskog plaćanja. Obrtnici koji su u sustavu PDV-a ne moraju voditi ovu evidenciju ukoliko vode Knjigu izlaznih i ulaznih računa na način da upisuju podatke o primljenim i izdanim računima, a ne prema plaćenim, odnosno neplaćenim računima.⁶²

4.5.3. Knjiga prometa

Knjiga prometa je evidencija u koju se upisuju primitci naplaćeni u gotovini. U slučaju paušalnog oporezivanja upisuju se i primitci naplaćeni bezgotovinskim putem. Podatci se u evidenciju upisuju krajem radnog rana ili prije početka idućeg radnog dana.⁶³

⁶⁰ Hrvatska obrtnička komora: Idem, str. 43.

⁶¹ Idem.

⁶² Narodne novine: Pravilnik o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 10/17, čl.47.st.1., 2., 3.

⁶³ Idem, čl.46.st.1.; Narodne novine: Pravilnik o izmjenama i dopunama Pravilnika o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 128/17, čl.17.

4.5.4. Knjiga primitaka i izdataka

Knjiga primitaka i izdataka je kronološka evidencija o dnevnim i ukupnim poslovnim primitcima i izdatacima tijekom jedne godine, odnosno poreznog razdoblja.⁶⁴ Porezni obveznik vodi samo jednu knjigu neovisno o broju poslovnih jedinica i registriranih samostalnih djelatnosti.⁶⁵ Isto vrijedi i za zajedničku djelatnost (supoduzetništvo) bez obzira na broj supoduzetnika i poslovnih jedinica.⁶⁶ U knjigu se unose opći podatci o poreznom obvezniku, kao što su naziv, ime i prezime te osobni identifikacijski broj (OIB).⁶⁷ Stavkom 3, 44. članka Pravilnika o porezu na dohodak (NN 10/2017) definiran je oblik i sadržaj knjige primitaka i izdataka na sljedeći način:⁶⁸

1. redni broj (stupac 1)
2. nadnevak primitaka i/ili izdataka. Nadnevak se upisuje za dane u kojima je ostvaren primitak i/ili izdatak. Ako za neki od stupaca nema podataka, upisuje se crtica (stupac 2)
3. broj temeljnice ili brojevi temeljnica na osnovi kojih su izvršena knjiženja (stupac 3)
4. opis isprava (računi od – do, izvaci žiroračuna, preslike potvrdnica iz naplatnih uređaja, popisi čekova, zapisnici o izuzimanjima i ulaganjima, obračuni i drugo) na temelju kojih su sastavljene temeljnice (stupac 4)
5. iznos primitaka u gotovini i čekovima. Ako se naknadno utvrdi da primljeni ček nije naplativ, obavlja se ispravak (storno) iskazanih primitaka (stupac 5)
6. iznos primitaka preko žiroračuna (stupac 6)
7. iznos primitaka u naravi, uslugama, izuzimanjima i drugih primitaka utvrđenih Zakonom i ovim Pravilnikom (stupac 7)
8. porez na dodanu vrijednost sadržan u primicima – naplaćenim naknadama te primljenim dobrima i uslugama (stupac 8)
9. ukupni primici – zbroj iznosa iz stupaca 5, 6 i 7 umanjen za iznos poreza na dodanu vrijednost iz stupca 8 (stupac 9)
10. iznos izdataka u gotovu novcu (stupac 10)
11. iznos izdataka izvršenih putem žiroračuna (stupac 11)
12. iznos ostalih izdataka u naravi, uslugama, ulaganjima i drugih izdataka utvrđenih Zakonom i ovim Pravilnikom (stupac 12)

⁶⁴ Narodne novine: Pravilnik o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 10/17, čl.44.st.1.

⁶⁵ Idem, čl.44.st.8.

⁶⁶ Idem, čl.44.st.9.

⁶⁷ Idem, čl.44.st.2.

⁶⁸ Idem, čl.44.st.3.

13. porez na dodanu vrijednost sadržan u izdacima – plaćenim naknadama i danim dobrima i uslugama (stupac 13)
14. izdaci iz članka 33. stavka 1. točkaka 1. i 5. Zakona na koje se plaća porez na dodanu vrijednost (stupac 14) i
15. ukupni porezno dopustivi izdaci – zbroj iznosa iz stupaca 10, 11 i 12 umanjen za iznos poreza na dodanu vrijednost sadržan u izdacima iz stupca 13 i umanjen za izdatke iz članka 33. stavka 1. točkaka 1. i 5. Zakona iz stupca 14 (stupac 15).

Valja napomenuti kako se sukladno istom Pravilniku u stupce 5, 6 i 7 unose podatci o ukupnim primitcima s naplaćenim PDV-om. U stupce 10, 11 i 12 se unose podatci o ukupnim izdacima, uključujući sljedeće izdatke koji su porezno nepriznati: 50% izdataka reprezentacije te 50% izdataka u svezi s vlastitim ili unajmljenim osobnim motornim vozilima i ostalim sredstvima za osobni prijevoz obrtnika i/ili zaposlenika.⁶⁹ Porezni obrtnik koji obavlja samostalnu djelatnost dužan je u evidenciji, odnosno knjizi primitaka i izdataka, osigurati podatke o dnevnom gotovinskom prometu, osim ako vode i evidenciju i dnevnom gotovinskom prometu prema Zakonu o porezu na dodatnu vrijednost. Tada u stupac 5 knjige primitaka i izdataka mogu upisivati gotovinski promet u jednom iznosu za cijelo obračunsko razdoblje, u rokovima koji su propisani za utvrđivanje obveze poreza na dodanu vrijednost. Gotovinski promet mogu upisivati u jednom iznosu za prethodni mjesec ako nisu obveznici poreza na dodanu vrijednost.⁷⁰ Podatci se u knjigu primitaka i izdataka moraju unositi najkasnije do 15. u mjesecu za prethodni mjesec, dok obveznici poreza na dodanu vrijednost moraju unijeti podatke u roku propisanom za uplatu poreza na dodanu vrijednost.⁷¹

⁶⁹ Idem, čl.44.st.6.

⁷⁰ Idem, čl.44.st.5.

⁷¹ Idem, čl.44.st.7.

5. PROGRAMSKE APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA

Provođenjem ad hoc istraživanja, utvrđeno je da na tržištu postoje brojna programska rješenja za vođenje knjige primitaka i izdataka, no većinom su implementirana u veće sustave za financijsko računovodstvo obrtnika. Ti sustavi omogućuju vođenje svih poslovnih knjiga te njihovo povezivanje, a katkada i neke druge funkcionalnosti kao što su obračun plaća i sl.

5.1 Prednosti i nedostaci postojećih aplikacija na tržištu

S obzirom na veliki broj programskih rješenja, odnosno programskih aplikacija za financijsko računovodstvo obrtnika nije moguće generalizirati pri utvrđivanju prednosti i nedostataka takvih aplikacija općenito. Drugim riječima, određena prednost ili nedostatak nije prisutna kod svih programskih aplikacija. Stoga će biti opisani najčešći nedostaci kao i prednosti.

5.1.1. Najčešći nedostaci

Istraživanjem su uočeni sljedeći nedostaci kod programskih aplikacija za obrtničku evidenciju:

- Brojne aplikacije dizajnirane su za Microsoft Windows operativni sustav. Time se obrtnicima onemogućuje korištenje besplatnih Linux distribucija kao i korištenje mobilnih operativnih sustava kao što su Android i iOS.
- Aplikacije često nisu temeljene na web tehnologijama. Time se onemogućava nesmetano korištenje na raznim uređajima neovisno o operativnom sustavu.
- Podatci se kod nekih programskih aplikacija pohranjuju isključivo lokalno bez korištenja sigurnosnih kopija i/ili *cloud* servisa. Time se obrtnike izlaže nepotrebnom riziku trajnog gubitka podataka o poslovanju.
- Mnoga programska rješenja su suviše komplicirana za korištenje osobama koje nisu stručnjaci za financijsko računovodstvo fizičkih osoba jer su namijenjene knjigovodstvenim servisima, a ne obrtnicima koji obavljaju druge djelatnosti, a žele si sami voditi poslovne knjige.

5.1.2. Najčešće prednosti

Istraživanjem su uočeni sljedeće prednosti kod programskih aplikacija za obrtničku evidenciju:

- Programska rješenja većinom su redovno ažurirana i prate relevantne zakonske promjene.
- Većina tvrtki koje izrađuju i prodaju ovakva programska rješenja nude podršku svojim klijentima te korisnička uputstva.
- Uz programska rješenja za financijsko knjigovodstvo obrtnika nekada u ponudi budu i rješenja za druge oblike poslovanja, primjerice za društva s ograničenom odgovornošću. Time se korisnicima tih programskih rješenja omogućuje jednostavniji prelazak na ta druga rješenje ukoliko bi za time bilo potrebe.

5.2. Segment tržišta koji nije pokriven postojećim programskim aplikacijama

Obrtnicima koji ne ostvaruju visoki dohodak nije isplativo plaćati usluge knjigovodstvenog servisa za vođenje poslovnih knjiga. Neki od njih nisu u mogućnosti naučiti voditi si poslovne knjige iz više razloga, primjerice, ako su već u radnom odnosu i nemaju vremena, djelatnost kojom se bave je hobi kojim se povremeno bave ili djelatnost nije dovoljno profitabilna da im se isplati uložiti vrijeme i trud u učenje. Takvih obrtnicima je potrebno jednostavno i jeftino programsko rješenje koje mogu koristiti bez dugotrajne obuke.

6. ALATI KORIŠTENI ZA IZRADU WEB APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA

Kako bi se realizirali postavljeni ciljevi istraživanja potrebno je bilo odabrati primjerene tehnologije za razvoj web programske aplikacije namijenjene evidentiranju prihoda i rashoda u okviru obrtničke evidencije. U tu su svrhu istražene mogućnosti sljedećih web tehnologija: PHP, HTML, CSS, MySQL.

6.1. PHP

PHP je široko upotrebljavan skriptni jezik opće namjene i otvorenog koda prikladan za web razvoj. Naziv ovog jezika je rekurzivni akronim za „PHP: Hypertext Preprocessor”.⁷² Kreirao ga je Lerdorf 1994. godine pod nazivom „Personal Home Page Tools”, no od tada je prošao brojne preinake te se trenutna verzija 8.1. značajno razlikuje od inicijalne.⁷³ PHP se izvodi na strani servera što znači da se kod izvodi prije nego korisnik preuzme web stranicu na svoje računalo. Izvođenjem PHP koda se mijenja ili stvara HTML kod koji korisnik preuzima i vidi u izvornom kodu stranice.⁷⁴ PHP za output može dati i druge tipove podataka, primjerice: pdf, jpeg, png i dr. Jedna od najvažnijih značajki PHP jezika je podrška za razne baze podataka kao što su MySQL, MariaDB, PostgreSQL, Oracle, Sybase, MS-SQL, MongoDB, CouchDB i mnogi drugi.⁷⁵

W3Techs provodi istraživanja o korištenosti programskih jezika za razvoj web tehnologija. Prema njihovim istraživanjima PHP je najčešće korišten jezik na strani poslužitelja. Grafikon 1. prikazuje najčešće korištene programske jezike na strani poslužitelja.

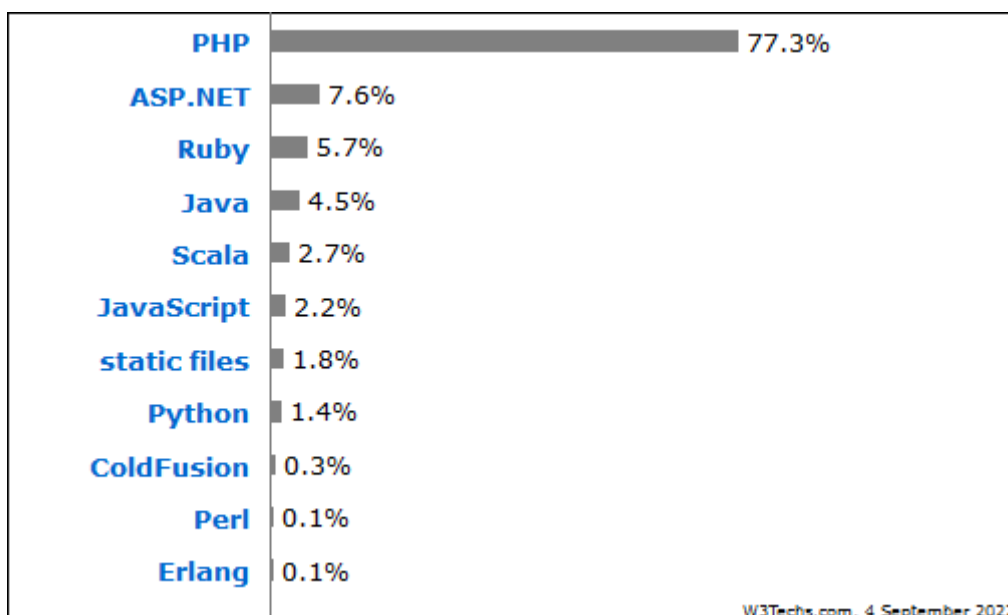
⁷² The PHP Group. PHP Manual – What is PHP?. <https://www.php.net/manual/en/intro-what-is.php>, 1.9.2022.

⁷³ Tatroe, K. i MacIntyre, P.: Programming PHP, O'Reilly Media, Sebastopol, 2020., str. 2.

⁷⁴ Beighley, L. i Morrison, M.: Head First PHP & MySQL, O'Reilly Media, Sebastopol, 2008., str. 3.

⁷⁵ Tatroe, K. i MacIntyre, P.: Idem.

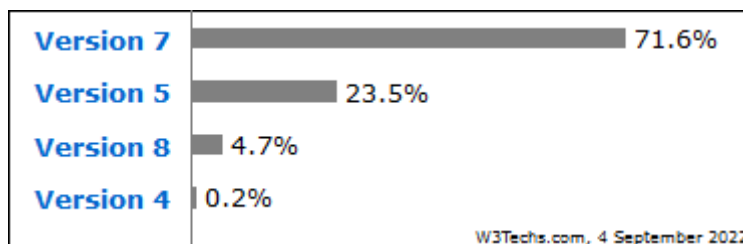
Grafikon 1. Najčešće korišteni jezici na strani poslužitelja.



Izvor: https://w3techs.com/technologies/overview/programming_language (4.9.2022.)

Na mnogim poslužiteljima se i dalje koriste starije verzije PHP-a, pa tako se verzija 5 koristi na 23,5% svih poslužitelja koji koriste PHP. Verzija 7 je po popularnosti prestigla verziju 5 u travnju 2020. godine, četiri godine nakon inicijalnog izlaska.⁷⁶ Te godine, u studenom, izašla je verzija 8.⁷⁷ Grafikon 2. prikazuje najčešće korištene verzije PHP-a

Grafikon 2. Najčešće korištene verzije PHP-a.



Izvor 1: <https://w3techs.com/technologies/details/pl-php> (4.9.2022.)

⁷⁶ Soltano, S.: Web technology fact of the day, 2020., https://w3techs.com/blog/entry/fact_20200414, 25.8.2022.

⁷⁷ The PHP Group. PHP Manual – News Archive – PHP 8.0.0. Released!, <https://www.php.net/manual/en/intro-what-is.php>, 1.9.2022.

6.2. MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka otvorenog koda nastao 1995. godine.⁷⁸ Glavni autor MySQL-a je Michael Widenius, no pravni vlasnik sustava je bila švedska kompanija MySQL AB čiji je on suvlasnik. Kompaniju je 2008. godine kupio Sun Microsystems, a dvije godine kasnije je njih preuzeo Oracle Corporation.⁷⁹ Ubrzo nakon potom mnogi razvojni programeri MySQL-a otišli su iz kompanije i započeli na razvoju novom sustavu nazvanom MariaDB. Riječ je o *forku* MySQL-a čiji je cilj ostati besplatan i otvorenog koda uz zadržavanje visoke kompatibilnosti sa MySQL sustavom.⁸⁰ MySQL omogućuje pristup, dodavanje i procesiranje podataka u bazi podataka koristeći SQL jezik. Podržava razne tipove podataka kao što su cijeli brojevi, decimalni brojevi, tekstualni podatci, datum, vrijeme i dr. Može se koristiti za upravljanje velikim bazama podataka koje sadrže više milijardi podataka.⁸¹

6.3. SQL

Edgar F. Codd osmislio je koncept relacijske baze podataka 1970. godine. Tada je osmislio i koncept jezika nazvanog DSL/Aplha za upravljanje podacima u relacijskim tablicama. Nedugo nakon toga, IBM je izradio prototip njegove ideje pod imenom SQUARE. Iz toga je proizašao jezik SEQUEL čiji je naziv kasnije skraćen na SQL.⁸²

6.4. HTML

HyperText Markup Language, odnosno HTML, je jezik za označavanje dokumenata za prikazivanje u internetskim preglednicima. Osmislio ga je Tim Berners-Lee 1990. godine, a za daljnji razvoj zadužen je bio World Wide Web Consortium (W3C) te od 2019. godine Web Hypertext Application Technology Working Group.⁸³

⁷⁸ Oracle: MySQL 8.0 Reference Manual., 2022., <https://downloads.mysql.com/docs/refman-8.0-en.a4.pdf>, 24.8.2022., str. 4.-5.

⁷⁹ Murach, J.: Murach's MySQL, Mike Murach & Associates, Inc., Fresno, 2019., str. 18.-19.

⁸⁰ Idem, str. 20.

⁸¹ Oracle: Idem, str. 5-8.

⁸² Beaulieum A.: Learning SQL, O'Reilly Media, Inc., Sebastepol, 2020., str. 8.

⁸³ WHATWG: HTML Living Standard, <https://html.spec.whatwg.org/print.pdf>, 31.8.2022., str. 25.-26.

HTML je jezik za definiranje strukture mrežnih stranica. Kada korisnik putem internetskog preglednika zatraži HTML dokument, poslužitelj mu ga pošalje, a zatim preglednik interpretira HTML oznake prema dogovorenom standardu.⁸⁴

Verzija HTML5 donijela je brojna poboljšanja i nove mogućnosti kao što su podrška za implementaciju video zapisa. U WHATWG-u je donešena odluka da se primijeniti koncept kontinuiranog razvoja HTML-a, tako da HTML neće dobivati nove brojeve inačica u budućnost.⁸⁵

6.5. CSS

S izlaskom četvrte verzije HTML-a, W3C je odlučio odvojiti prezentacijski dio od strukturnog. Tako je stvoren jezik za prezentaciju nazvan Cascading Style Sheets, odnosno CSS.⁸⁶ CSS-om se definira izgled HTML dokumenta, kao što su primjerice boje, fontovi i slično. Omogućuje responzivnost stranice, odnosno adaptaciju za različite veličine ekrana na kojima se stranica pregledava.⁸⁷ CSS kod je moguće pisati u samom HTML dokumentu ili u CSS dokumentu koji se povezuje sa HTML dokumentom.⁸⁸

6.6. Ostali alati

XAMPP je razvojno okruženje koje uključuje Apache server, MariaDB sustav za upravljanje bazom podataka (u starijim verzijama MySQL) te PHP i Perl programske jezike. Naziv je akronim gdje „X” označava višeplatformni software, a ostala slova predstavljaju ranije navedene tehnologije.⁸⁹ Za potrebe ovog rada, XAMPP je konfiguriran da koristi MySQL umjesto MariaDB sustava.

Slika 1. prikazuje izgled XAMPP sučelja.

⁸⁴ Robson, E. i Freeman, E.: Head First HTML and CSS, O'Reilly Media, Inc., Sebastopol, 2012., str. 5.

⁸⁵ WHATWG: Idem, str. 26.

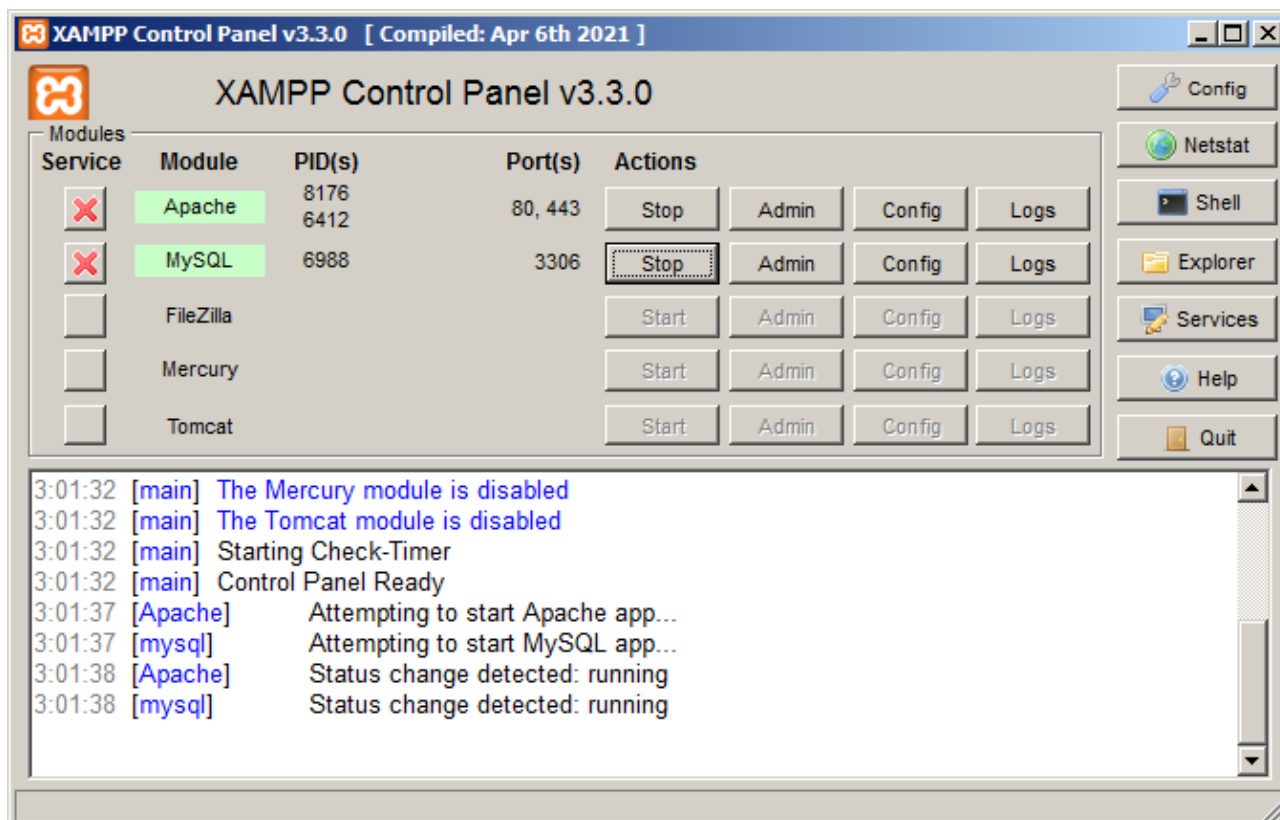
⁸⁶ Robson, E. i Freeman, E.: Idem, str. 222.

⁸⁷ W3C: HTML & CSS, <https://www.w3.org/standards/webdesign/htmlcss>, 24.8.2022.

⁸⁸ Meyer, E.: CSS Pocket Reference, O'Reilly Media, Inc., Sebastopol, 2018., str. 1.

⁸⁹ Apache Friends: XAMPP, <https://www.apachefriends.org/index.html>, 25.8.2022.

Slika 1. Izgled XAMPP sučelja.



Apache HTTP Server je besplatna platforma web poslužitelja otvorenog koda. Napravio ga je 1995. godine Rob McCool, no na projektu su kasnije sudjelovali brojni programeri. Njih osam najaktivnijih organiziralo se je u skupinu Apache Group u kojoj su nastavili raditi na projektu. 1999. godine oformili su Apache Software Foundation radi ostvarivanja bolje organizacijske, pravne i financijske podrške.⁹⁰

Atom je GitHub-ov besplatni uređivač teksta i izvornog koda dostupan za Linux, Windows i macOS operativne sustave, te je otvorenog koda. Atom je distribuirana aplikacija programirana u web tehnologijama (Electron). Podržava brojne preinake, nadogradnje i Git kontrolu verzije.⁹¹

⁹⁰ The Apache Software Foundation: About the Apache, https://httpd.apache.org/ABOUT_APACHE.html, 25.8.2022.

⁹¹ GitHub: Atom Flight Manual – Why Atom?, <https://flight-manual.atom.io/getting-started/sections/why-atom/>, 25.8.2022.

Nadalje, phpMyAdmin je brezplačni programski alat za administriranje MySQL ili MariaDB bazama podataka putem weba. Može se koristiti za kreiranje, pretraživanje, uređivanje, brisanje baze podataka i tablica u njoj. Podržava i korištenje SQL naredba.⁹²

⁹² The phpMyAdmin devel team: phpMyAdmin Documentation, docs.phpmyadmin.net/_/downloads/en/latest/pdf/, 25.8.2022., str. 3.

7. MODEL BAZE PODATAKA APLIKACIJE

Aplikacija za vođenje knjige primitaka i izdataka koristi MySQL sustav za upravljanje bazom podataka, dok se za administriranje bazom koristi alat phpMyAdmin. Shematskim prikazom na slici 2. prikazan je odnos između klijenta, aplikacije i baze podataka. Korisnik pomoću internet preglednika putem interneta šalje zahtjev prema web poslužitelju, odnosno aplikaciji koja potom izvršava SQL upit u bazu podataka. Baza podataka daje rezultat aplikaciji koja korisniku šalje traženi odgovor istim kanalima kojima je zahtjev stigao.

Slika 2. Shematski prikaz odnosa između klijenta, aplikacije i baze podataka.



Izvor: Prilagodeno prema Murach, J.: Murach's MySQL, Mike Murach & Associates, Fresno, 2019., str. 9.

Aplikacija inicijalno ima samo jednu bazu podataka za pohranjivanje podataka o registriranim korisnicima, dok se druge tablice kreiraju dinamički kroz PHP kod prilikom registracije novih korisnika. Te tablice imaju funkciju knjige primitaka i izdataka za svakog korisnika.

Tablica za pohranu podataka o korisnicima naziva se users i sadrži sljedeće stupce: user_id, naziv_poduzeca, ime_poduzetnika, oib, ulica, postanski_broj, mjesto, naziv_djelatnosti, sifra_djelatnosti, email, tel, password.

Stupac user_id je primarni ključ koji služi kao jedinstveni identifikator svakog reda u tablici.⁹³ Email i password (lozinka) služe za prijavu korisnika u aplikaciju, tel za pohranu telefonskog broja korisnika, dok se ostali podatci o korisniku, sukladno Pravilniku o porezu na dohodak (NN 10/2017, čl.44.st.2), moraju nalaziti na obrascu KPI koji aplikacija kreira.

⁹³ Murach, J.: Idem, str. 11.

Za kreiranje tablice *users* izvršava se SQL kod prikazan slikom 3.

Slika 3. Kreiranje tablice "users".


```
1 CREATE TABLE users(  
2     user_id INT PRIMARY KEY AUTO_INCREMENT,  
3     naziv_poduzeca VARCHAR(50),  
4     ime_poduzetnika VARCHAR(100),  
5     oib CHAR(11),  
6     ulica VARCHAR(30),  
7     postanski_broj VARCHAR(5),  
8     mjesto VARCHAR(30),  
9     naziv_djelatnosti VARCHAR(50),  
10    sifra_djelatnosti VARCHAR(12),  
11    email VARCHAR(50),  
12    tel VARCHAR(15),  
13    password TEXT  
14 );
```


Stupac *user_id* je primarni ključ tipa INT (*integer*), tj. cijeli broj. AUTO_INCREMENT znači da će se taj broj uvećati za jedan kod svakog novog unosa korisnika u tablicu. Stupci tipa VARCHAR mogu sadržavati sve znakove do maksimalne količine znakova koja je upisana upisane u zagradu. OIB je CHAR podatak što znači da mora imati točno određeni broj znakova, a to je 11. Password (lozinka) je tipa TEXT, te može sadržavati sve znakove, a podatak može biti maksimalno 64 kilobajta velik, što znači da može sadržavati preko 65.000 znakova. Za lozinku je potrebna veća količina memorije jer se lozinka ne pohranjuje u bazu podataka, već se pohranjuje njena *hash* vrijednost koja ima višestruku količinu znakova.

Ostale tablice kreiraju se pri registraciji korisnika, a naziv tablice se sastoji od korisnikovog OIB-a, te nastavka „_kpi”. Time svaki korisnik dobiva svoju tablicu u koju se upisuju njegovi primitci i izdatci.

Slika 4. prikazuje model baze podataka.

Slika 4. Model baze podataka.

users		
	user_id	integer
	naziv_poduzeca	varchar(50)
	ime_poduzetnika	varchar(100)
	oib	char(11)
	ulica	varchar(30)
	postanski_broj	varchar(5)
	mjesto	varchar(30)
	naziv_djelatnosti	varchar(50)
	sifra_djelatnosti	varchar(12)
	email	varchar(50)
	tel	varchar(15)
	password	text

{oib}_kpi		
	id	integer
	datum	date
	br_temeljnice	varchar(40)
	opis	varchar(50)
	naziv subjekta	varchar(50)
	tip_racuna	char(3)
	primici_gotovina	decimal(9,2)
	primici_ziro_racun	decimal(9,2)
	primici_u_naravni	decimal(9,2)
	primici_pdv	decimal(9,2)
	primici_ukupno	decimal(9,2)
	izdaci_gotovina	decimal(9,2)
	izdaci_ziro_racun	decimal(9,2)
	izdaci_u_naravni de	decimal(9,2)
	izdaci_pdv	decimal(9,2)
	izdaci_cl_33	decimal(9,2)
	izdaci_ukupno	decimal(9,2)

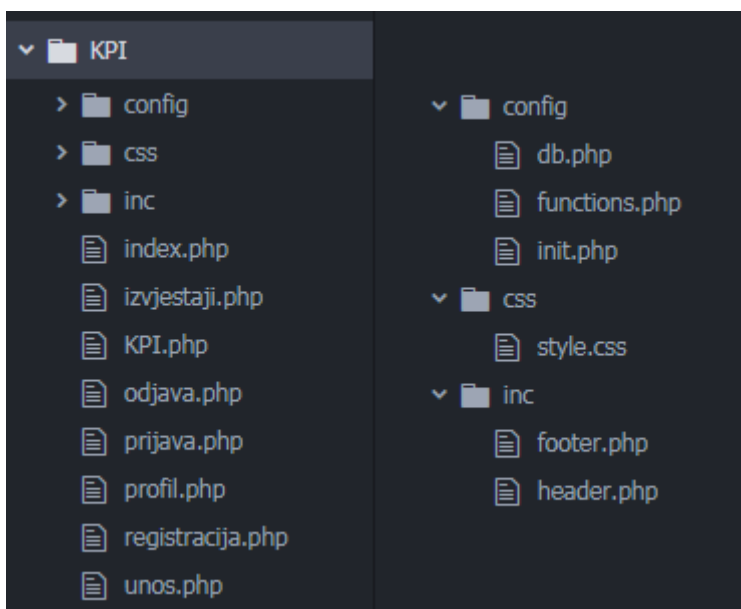
8. STRUKTURA APLIKACIJE ZA VOĐENJE KNJIGE PRIMITAKA I IZDATAKA

Kod aplikacije nalazi se u mapi KPI koja sadrži PHP datoteke koje predstavljaju web stranice kojima korisnik može pristupiti. Osim tih datoteka sadrži i tri podmape:

- „css” – u ovoj mapi se nalazi datoteka style.css koja definira izgled svih stranica aplikacije,
- „config” – sadrži .php datoteke s osnovnim funkcionalnostima aplikacije te
- „inc” – sadrži zaglavlje i podnožje aplikacije.

Slika 5. daje prikaz organizacije koda programske aplikacije.

Slika 5: Prikaz organizacije koda aplikacije.



8.1. Mapa config

Temeljne funkcionalnosti web aplikacije nalaze se u mapi config koja sadrži tri PHP datoteke. Datoteka init.php (skraćeno od *initialization*, odnosno inicijalizacija) pokreće sesiju funkcijom `session_start()`, te potom uključuje iduće dvije datoteke iz config mape – db.php i functions.php.

Slika 6. prikazuje datoteku init.php.

Slika 6. Datoteka init.php.

```
1 <?php
2
3 session_start();
4
5 include "db.php";
6 include "functions.php";
```

Sesija je način čuvanja određenih podataka te njihovog korištenja na više web stranica. Sesija se pohranjuje na poslužitelju i čuva se u vektoru `$_SESSION`.⁹⁴ U ovoj aplikaciji u sesiji se pohranjuju podatci o prijavljenom korisniku.

Datoteka `db.php` u varijabli `$conn` sadrži podatke potrebne za povezivanje s bazom podataka. To čini kroz `mysqli_connect()` funkciju koja za parametre ima ime poslužitelja, korisničko ime, lozinku i naziv baze podataka.

Slika 7. prikazuje datoteku `db.php`.

Slika 7. Datoteka db.php.

```
1 <?php
2
3 $conn= mysqli_connect(hostname:'localhost', username:'root', password:'', database:'knjige');
4
5
6 function confirm($result){
7     global $conn;
8     if(!$result){
9         die("Error: " . mysqli_error($conn));
10    }
11 }
```

Funkcija `mysqli_connect()` dio je MySQLi ekstenzije (MySQL Improved Extension) koja služi za upravljanje s MySQL i MariaDB bazama podataka, a može se koristiti na proceduralni ili objektno orijentirani način.⁹⁵ Osim navedene ugrađene funkcije sadrži i funkciju `confirm()` koja ispisuje grešku

⁹⁴ Olsson, M.: PHP 8 Quick Scripting Reference: A Pocket Guide to PHP Web Scripting, Apress Media LLC, New York, 2021., str. 147.

⁹⁵ Tatroe, K. i MacIntyre, P.: Idem, str. 217.-218.

ako do iste dođe prilikom izvršenja SQL naredbe. U datoteci functions.php iz mape functions nalaze se sve ostale funkcije potrebne za rad aplikacije.

8.2. Mapa inc

U mapi inc nalaze se dvije datoteke – header.php i footer.php, koje predstavljaju zaglavlje i podnožje. Te datoteke uključene su u svaku drugu PHP datoteku koje trebaju sadržavati zaglavlje i podnožje. Slika 8. prikazuje datoteka header.php.

Slika 8. Datoteka header.php

```
1 <?php include "config/init.php" ?>
2
3 <!DOCTYPE html>
4 <html lang="hr" dir="ltr">
5   <head>
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <link rel="stylesheet" type="text/css" href="css/style.css">
9     <title>Knjiga primitaka i izdataka</title>
10  </head>
11  <body>
12    <div class="container">
13      <nav>
14        <a href="index.php">Početna</a>
15        <?php if(!isset($_SESSION['email'])) : ?>
16          <a href="registracija.php">Registracija</a>
17          <a href="prijava.php">Prijava</a>
18        <?php else : ?>
19          <a href="unos.php">Unos računa</a>
20          <a href="izvjestaji.php">Izvještaji</a>
21          <a href="profil.php">Profil</a>
22          <a href="odjava.php">Odjava</a>
23        </nav>
24        <p class="welcome"><?php $user = get_user();
25          echo "Prijavljeni ste kao <b>" . $user['naziv_poduzeca'] . "</b>"; ?></p>
26        <?php endif; ?>
27    </div>
```

Datoteka header.php u prvoj liniji koda uključuje init.php datoteku iz mape config. Na taj način svaka stranica koja sadrži zaglavlje ima pristup kodu iz init.php datoteke, odnosno ima pristup sesiji, bazi podataka i funkcijama. Sadržaj unutar *head* oznaka se ne prikazuje, već služi za pohranu metapodataka. Ti podatci uključuju naslov mrežnog mjesta, jezik na kojem je pisan sadržaj, poveznicu za CSS datoteku i sl. Sadržaj unutar *body* oznake se prikazuje korisniku. Sadrži *div* oznaku klase container čiji je izgled definiran CSS-om. Unutar tog dijela se nalazi navigacijski dio koji neprijavljenom korisniku prikazuje poveznice za početnu stranicu, registraciju i prijavu, dok

prijavljenom korisniku prikazuje poveznice za početnu stranicu, unos računa, kreiranje izvještaja, profil (izmjena podataka o korisniku) te poveznicu za odjavu. To čini tako da provjerava nalazili se u sesiji podatak o email adresi trenutno prijavljenog korisnika. Prijavljenom korisniku će se također u desnom kutu stranice ispisati poruka da je prijavljen te s kojim računom je prijavljen. Izgled poruke je definiran CSS-om i klasom welcome.

Kao što je vidljivo iz slike 9. stranica footer.php služi isključivo za zatvaranje otvorenih HTML oznaka iz zaglavlja.

Slika 9. Datoteka footer.php

```
1     </div>
2     </body>
3 </html>
```

Kod ranije spomenute CSS datoteke koja definira izgled stranica nalazi se na slici 10.

Slika 10. Datoteka style.css (prvi dio).

```
1  body{
2    background: brown;
3  }
4
5  .container {
6    max-width: 1250px;
7    background-color: antiquewhite;
8    margin: auto;
9    padding: 20px;
10   box-shadow: 0 0 35px rgba(0,0,0,0.5);
11   min-height: 94vh;
12 }
13
14 hr {
15   margin-top: 15px;
16   margin-bottom: 15px;
17 }
18
19 nav {
20   display: inline;
21 }
22
23 nav a{
24   margin-right: 15px;
25   font-size: 20px;
26 }
27
28 p.welcome {
29   display: inline-block;
30   float: right;
31   margin: auto;
32 }
33
```

Za *body* se postavlja pozadinska boja *brown*, te se u tom dijelu nalazi *div* klase *container*. Taj *div* će biti maksimalne širine 1250 piksela, minimalne visine od 94% vertikalne dužine ekrana, boje *antiquewhite*, te će imati sjenu. Margine su postavljene posebno za svaki dio, a paragraf klase *welcome* će se uvijek prikazivati u desnom dijelu stranice.

8.3. Web stranice iz mape KPI i funkcije iz `functions.php`

Glavna mapa „KPI” sadrži PHP datoteke koje čine web stranice kojima korisnik može pristupiti. Na tim stranicama izvršavaju se funkcije iz mape „config”, odnosno datoteke `functions.php`. PHP datoteke iz mape „KPI” su: `index.php`, `registracija.php`, `prijava.php`, `unos.php`, `izvjestaji.php`, `profil.php`, te `KPI.php`.

8.3.1. Datoteka `index.php`

Početna stranica koje se otvori kada se pristupi web aplikaciji je `index.php`. Kao i ostale stranice, uključuje `header.php` i `footer.php` te `if` naredbom provjerava postoji li informacija o email-u korisniku u sesiji. Ako postoji korisnik je logiran i prikazuju mu se poruka „Prijavljeni ste i možete započeti s radom u aplikaciji.”, a ako nije ispisuje se tekst: "Za korištenje aplikacije potrebna je registracija." PHP kod je od HTML koda odvojen `php` oznakama. Slika 11. prikazuje datoteku `index.php`.

Slika 11. Datoteka `index.php`.

```
1 <?php include "inc/header.php" ?>
2
3 <br>
4
5 <?php
6 if(isset($_SESSION['email'])){
7     echo "Prijavljeni ste i možete započeti s radom u aplikaciji.";
8 }
9 else{
10     echo "Za korištenje aplikacije potrebna je registracija.";
11 }
12 ?>
13
14 <br><br>
15
16 <?php include "inc/footer.php" ?>
```

8.3.2. Datoteka registracija.php

Korisnik se mora registrirati na web aplikaciji kako mogao upravljati svojom knjigom primitaka i izdataka. Proces registracije izvršava se na stranici registracija.php. Slika 12. prikazuje datoteku registracija.php.

Slika 12. Datoteka registracija.php.

```
1 <?php include "inc/header.php"?>
2
3 <?php validate_registration(); ?>
4
5 <form method="POST">
6   <input type="text" name="naziv_poduzeca" placeholder="Naziv poduzeća" required>
7   <input type="text" name="ime_poduzetnika" placeholder="Ime poduzetnika" required>
8   <input type="text" name="oib" placeholder="OIB" required>
9   <input type="text" name="ulica" placeholder="Ulica" required>
10  <input type="text" name="postanski_broj" placeholder="Poštanski broj" required>
11  <input type="text" name="mjesto" placeholder="Grad/mjesto" required>
12  <input type="text" name="naziv_djelatnosti" placeholder="Naziv djelatnosti" required>
13  <input type="text" name="sifra_djelatnosti" placeholder="Šifra djelatnosti" required>
14  <input type="email" name="email" placeholder="Email" required>
15  <input type="text" name="tel" placeholder="Telefonski broj" required>
16  <input type="password" name="password" placeholder="Lozinka" required>
17  <input type="password" name="confirm_password" placeholder="Potvrda lozinke" required>
18  <br>
19  <input type="submit" name="registracija-submit" value="Registracija">
20 </form>
21
22 <?php include "inc/footer.php" ?>
```

Na stranici se nalazi forma za upis traženih podataka kao što su naziv obrta, ime poduzetnika, OIB, mjesto prebivališta, naziv djelatnosti i šifra djelatnosti. To su podatci koji se sukladno Pravilniku o porezu na dohodak (NN 10/2017, čl.44.st.2) moraju nalaziti na obrascu KPI. Osim toga, za kasniju prijavu korisnika potreban je i unos email adrese i lozinke.

Svi podatci osim email adrese i lozinke su tekstualnog oblika. Email adresa je oblika „email” čime se smanjuje mogućnost unos pogrešnog unosa podatka jer će korisnik dobiti povratnu informaciju ako se unese podatak koji nije email adresa. Lozinka je tipa „password”. Takav oblik daje dodatnu sigurnost pri registraciji jer u formi nije vidljiv sadržaj koji se upisuje već je isti zamijenjen točkama.

Pri registraciji je potrebno dva puta upisati lozinku kako bi se otklonila mogućnost utjecaja tipfelera na unesenu lozinku. Korištenjem metode „POST” postiže se dodatna sigurnost jer uneseni podaci neće biti vidljivi u URL-u. Klikom na dugme „Registracija” unesene podatci se spremaju u asocijativni niz \$_POST. Potom se izvršava funkcija validate_registration(). Slika 13. prikazuje funkciju validate_registration()

Slika 13. Funkcija validate_registration()

```
26 function validate_registration(){
27     $errors = [];
28     if($_SERVER['REQUEST_METHOD'] == 'POST'){
29         $naziv_poduzeca = htmlentities($_POST['naziv_poduzeca']);
30         $ime_poduzetnika = htmlentities($_POST['ime_poduzetnika']);
31         $oib = htmlentities($_POST['oib']);
32         $ulica = htmlentities($_POST['ulica']);
33         $postanski_broj = htmlentities($_POST['postanski_broj']);
34         $mjesto = htmlentities($_POST['mjesto']);
35         $naziv_djelatnosti = htmlentities($_POST['naziv_djelatnosti']);
36         $sifra_djelatnosti = htmlentities($_POST['sifra_djelatnosti']);
37         $email = htmlentities($_POST['email']);
38         $tel = htmlentities($_POST['tel']);
39         $password = htmlentities($_POST['password']);
40         $connfirm_password = htmlentities($_POST['confirm_password']);
41
42         if(strlen($oib) != 11){$errors[] = "OIB mora sadržavati 11 znamenki.";}
43         if(email_already_in_use($email)){ $errors[] = "Email se već koristi.";}
44         if(strlen($password) < 8){$errors[] = "Lozinka ne može biti kraća od 8 znakova.";}
45         if($password != $connfirm_password){$errors[] = "Lozinke se ne podudaraju.";}
46         if(!empty($errors)){
47             foreach($errors as $error){
48                 echo "<div class = 'error'>" . $error . "</div>";
49             }
50         } else{
51             create_user($naziv_poduzeca, $ime_poduzetnika, $oib, $ulica, $postanski_broj,
52                 $mjesto, $naziv_djelatnosti, $sifra_djelatnosti, $email, $tel, $password);
53         }
54     }
55 }
```

Funkcija ne poprima parametre. Kreira se niz \$errors u koji će se upisivati potencijalne pogreške nastale pri registraciji korisnika. Naredba if provjerava postoji li u asocijativnom nizu \$_SERVER tekstualni podatak „POST” pod ključem „REQUEST_METHOD”. Ako postoji to znači da su podaci iz forme uspješno uneseni te se potom nastavlja izvršavati daljnji kod. U varijable se spremaju podatci koji su uneseni putem forme, a koji se nalaze u asocijativnom nizu \$_POST. Varijable su nazvane po ključevima iz tog niza. No, na te podatke se prvo primjenjuje ugrađena PHP funkcija htmlentities().

Ta se funkcija koristi iz sigurnosnih razloga jer znakove koji se koriste u HTML tagovima pretvara u druge znakove (npr. znak „& ” pretvara u „&”) te se time smanjuje mogućnost izvršavanje PHP naredbi koje se mogu upisati u formu.

Nakon unosa podataka u varijable provjerava se ispravnost određenih podataka, pa tako OIB mora sadržavati 11 znamenki, lozinka ne smije biti kraća od osam znakova te unesena lozinka mora biti jednaka lozinki unesenoj za provjeru. Ako jedan ili više uvjeta nisu zadovoljeni unosi se tekstualni podatak za odgovarajućom porukom i sprema se u niz \$errors. Također se funkcijom email_already_in_use() provjerava postoji li već registrirani korisnik s navedenom email adresom, točnije, provjerava nalazi li se već ta email adresa u bazi podataka. Slika 14. prikazuje funkciju email_already_in_use().

Slika 14. Funkcija email_already_in_use().

```
131 function email_already_in_use($email){
132     global $conn;
133     $email = filter_var($email, filter:FILTER_SANITIZE_EMAIL);
134     $query = "SELECT user_id FROM users WHERE email = '$email'";
135     $result = mysqli_query($conn, $query);
136     $provjera = mysqli_num_rows($result);
137
138     if($provjera > 0){
139         return true;
140     }
141     else{
142         return false;
143     }
144 }
```

Ova funkcija šalje upit u bazu podataka kojim se traži *user_id* korisnika iz tablice *users* gdje je email isti poput onog unesenog pri registraciji novog korisnika i pohranjenog u \$email varijablu. Funkcijom `mysqli_query()` izvršava se upit i rezultat se sprema u varijablu \$result. Varijabla \$provjera će sadržavati broj redova u tablici gdje je upisana tražena email adresa. Taj podatak se dobiva funkcijom `mysqli_num_rows()`. Svaki red u tablici *users* predstavlja jednog korisnika. Ako je funkcija pronašla korisnika koji već koristi taj email vraća true, u suprotnom vraća false.

Potom se nastavlja daljnje izvršavanje funkcije `validate_registration()`. Ako niz \$errors nije prazan, odnosno ako sadrži neke podatke, ti podatci se ispisuju. Izgled tog ispisa definiran je klasom „error” u CSS datoteci style.css. Slika 15. prikazuje klasu style.css (klasa "error").

Slika 15. Klasa "error" (Datoteka style.css).

```
58  .error{  
59      margin-top: 5px;  
60      border: 1px solid red;  
61      background-color: lightgrey;  
62      padding: 10px;  
63      max-width: 300px;  
64  }
```

Odgovarajuća poruka ispisat će se u okviru crvene boje veličine jednog piksela i svjetlosive pozadine. Prostor iznad okvira je pet piksela, dok je maksimalna širina okvira 300 piksela. Tekst unutar okvira je od istog udaljen deset piksela sa svih strana.

U slučaju da je niz \$errors prazan, odnosno nije došlo do pogreške pri registraciji, izvršava se funkcija create_user() koja za parametre poprima varijable koje sadrže podatke o korisniku (ime poduzeća, OIB, lozinka i dr.).

Slika 16. prikazuje funkciju create_user().

Slika 16. Funkcija create_user().

```
72 function create_user($naziv_poduzeca, $ime_poduzetnika, $oib, $ulica, $postanski_broj,  
73     $mjesto, $naziv_djelatnosti, $sifra_djelatnosti, $email, $tel, $password){  
74  
75     global $conn;  
76     $naziv_poduzeca = mysqli_real_escape_string($conn, $naziv_poduzeca);  
77     $ime_poduzetnika = mysqli_real_escape_string($conn, $ime_poduzetnika);  
78     $oib = mysqli_real_escape_string($conn, $oib);  
79     $ulica = mysqli_real_escape_string($conn, $ulica);  
80     $postanski_broj = mysqli_real_escape_string($conn, $postanski_broj);  
81     $mjesto = mysqli_real_escape_string($conn, $mjesto);  
82     $naziv_djelatnosti = mysqli_real_escape_string($conn, $naziv_djelatnosti);  
83     $sifra_djelatnosti = mysqli_real_escape_string($conn, $sifra_djelatnosti);  
84     $email = mysqli_real_escape_string($conn, $email);  
85     $tel = mysqli_real_escape_string($conn, $tel);  
86     $password = password_hash($password, algo:PASSWORD_DEFAULT);  
87  
88     $sql = "INSERT INTO users(naziv_poduzeca, ime_poduzetnika, oib, ulica,  
89         postanski_broj, mjesto, naziv_djelatnosti, sifra_djelatnosti, email, tel, password) ";  
90     $sql .= "VALUES('$naziv_poduzeca', '$ime_poduzetnika', '$oib', '$ulica', '$postanski_broj',  
91         '$mjesto', '$naziv_djelatnosti', '$sifra_djelatnosti', '$email', '$tel', '$password')";  
92  
93     confirm(mysqli_query($conn, $sql));  
94     create_table($oib);  
95     set_message("Uspješno ste se registrirali. Možete se prijaviti.");  
96     redirect("prijava.php");  
97 }
```

Na sve podatke o korisniku, osim lozinke, prvo se primjenjuje `mysqli_real_escape()` funkcija. Tom funkcijom se *stringovi* prilagođavaju tako da se mogu koristiti u SQL naredbama. Ako *string* sadrži posebne znakove, kao što su apostrof, mogu izazvati pogrešku u SQL kodu jer su apostrofi dio SQL sintakse. Time se postiže i veća sigurnost aplikacije jer onemogućuje namjerno korištenje posebnih znakova za izvršavanja malicioznih SQL naredba.

Na lozinku se primjenjuje funkcija `password_hash()` uz korištenje opcije `PASSWORD_DEFAULT` s kojim se primjenjuje algoritam „bcrypt”.⁹⁶ Time se u bazu podataka ne pohranjuje lozinka već njena *hash* vrijednost iz koje nije moguće saznati koju točno lozinku koristi korisnik.

Varijabla `$sql` je tipa *string* i sadrži SQL naredbu za unos podataka u tablicu *users*. Funkcijom `mysqli_query()` aplikacije se povezuje s bazom podataka te se izvršava SQL naredba iz te varijable.

⁹⁶ The PHP Group: PHP Manual – `password_hash`, <https://www.php.net/manual/en/function.password-hash.php>, 1.9.2022.

Kod svake registracije u tablici *users* stvara se novi red s podacima o registriranom korisniku. Sljedeća funkcija `create_table()` za svakog registriranog korisnika kreira tablicu u bazi podataka koja predstavlja knjigu primitaka i izdataka tog korisnika. Slika 17. prikazuje funkcija `create_table()`.

Slika 17. Funkcija `create_table()`.

```
103 function create_table($oib){
104     global $conn;
105     $sql = "CREATE TABLE {$oib}_kpi(
106         id INT PRIMARY KEY AUTO_INCREMENT,
107         datum DATE,
108         br_temeljnice VARCHAR(40),
109         opis VARCHAR(50),
110         naziv_subjekta VARCHAR(50),
111         tip_racuna CHAR(3),
112         primici_gotovina DECIMAL(9,2),
113         primici_ziro_racun DECIMAL(9,2),
114         primici_u_naravni DECIMAL(9,2),
115         primici_pdv DECIMAL(9,2),
116         primici_ukupno DECIMAL(9,2),
117         izdaci_gotovina DECIMAL(9,2),
118         izdaci_ziro_racun DECIMAL(9,2),
119         izdaci_u_naravni DECIMAL(9,2),
120         izdaci_pdv DECIMAL(9,2),
121         izdaci_cl_33 DECIMAL(9,2),
122         izdaci_ukupno DECIMAL(9,2)
123     );";
124     confirm(mysqli_query($conn, $sql));}
```

Funkcija za parametar koristi varijablu `$oib`, a taj se podatak koristi u kreiranju naziva tablice. S obzirom na to da je OIB jedinstveni broj za svakog korisnika tablica koja se kreira moći će se povezati s tim korisnikom.

Svaki primitak i izdatak ima svoj id koji je ujedno i primarni ključ. Naredbom `AUTO_INCREMENT` se id automatski uvećava kod svakog unosa primitka i izdatka. Tekstualni podatci koji se unose u tablicu su tipa `VARCHAR`, a u zagradi se nalazi maksimalni broj znakova koji se može unijeti. Datum je tipa `DATE` koji se uobičajeno koristi pri unosu datuma, te je formata „yyyy-mm-dd”, odnosno prvo je upisana godina, a potom mjesec i dan. `DECIMAL` je decimalni broj koji u ovom slučaju može imati maksimalno devet znamenki i dva decimalna mjesta.

Potom se izvršava sljedeća linija koda iz funkcije `create_user()`, a u toj liniji se nalazi funkcija `set_message()` koja za parametar poprima tekstualni podatak, tj. *string*: "Uspješno ste se registrirali. Možete se prijaviti." Slika 18. prikazuje funkciju `set_message()`.

Slika 18. Funkcija `set_message()`.

```
3 function set_message($message){
4     if(!empty($message)){
5         $_SESSION['message'] = $message;
6     }
7     else{
8         $message = "";
9     }
10 }
```

Taj *string* se pohranjuje u varijablu `$message`. Ako ta varijabla nije prazna, odnosno ako sadrži neki podatak, on se pohranjuje u sesiju pod ključnom riječju „*message*”. Razlog korištenja ove funkcije umjesto ispisivanja poruke je taj što će se ta poruka ispisati na drugoj stranici, a ne iznad forme za registraciju korisnika, te ju je stoga potrebno pohraniti u sesiju. Nakon toga slijedi preusmjerenje korisnika na stranicu za prijavu korištenjem `redirect()` funkcije. Slika 19. prikazuje funkciju `redirect()`.

Slika 19. Funkcija `redirect()`.

```
20 function redirect($location){
21     header(header:"location: {$location}");
22     exit();
23 }
```

Nakon izvršavanja funkcije `header()` potrebno je prekinuti daljnje izvršavanje koda na toj stranici jer se može dogoditi da se dio koda izvrši prije nego li `header()` obavi svoju zadaću do kraja. Zbog toga je korišten `exit()`.

8.3.3. Datoteka `prijava.php`

Stranica `prijava.php` sadrži sljedeći kod koji omogućuje korisniku unos svojih korisničkih podataka potrebnih za prijavu u sustav. Slika 20. prikazuje datoteka `prijava.php`.

Slika 20. Datoteka prijava.php.

```
1 <?php include "inc/header.php" ?>
2
3 <?php
4 display_message();
5 login_validation();
6 ?>
7
8 <form method="POST">
9   <input type="email" name="email" placeholder="Email" required>
10  <input type="password" name="password" placeholder="Lozinka" required>
11  <input type="submit" name="prijava-submit" value="Prijavi se">
12 </form>
13
14 <?php include "inc/footer.php" ?>
```

Prvo se izvršava funkcija `display_message()` koja provjerava postoji li u sesiji podatak pod ključnom riječju „*message*”. Nakon uspješne registracije taj podatak bi treba postojati. Prvo se ispisuje, a potom se briše iz sesije kako se ubuduće ne bi prikazivala ta poruka pri prijavi tog korisnika. Slika 21. prikazuje funkciju `display_message()`.

Slika 21. Funkcija `display_message()`.

```
12 function display_message(){
13     if(isset($_SESSION['message'])){
14         echo $_SESSION['message'];
15         unset($_SESSION['message']);
16     }
17 }
```

HTML forma za prijavu od korisnika traži unos email adrese i lozinke, te se klikom na dugme „Prijavi se” ti podaci koriste u funkciji `login_validation()`.

Slika 22. prikazuje funkciju `login_validation()`.

Slika 22. Funkcija login_validation().

```
146 function login_validation(){
147
148     if($_SERVER['REQUEST_METHOD'] == "POST"){
149         $email = htmlentities($_POST['email']);
150         $password = htmlentities($_POST['password']);
151
152         if(user_login($email, $password)){
153             redirect(location:"index.php");
154         }
155         else{
156             echo '<div class="error">Podatci za prijavu nisu ispravni.';
157             echo '<br>Pokušajte ponovno.</div>';
158         }
159     }
160 }
```

Logika unosa i pohranjivanja podataka je identična onoj iz funkcije validate_registration(). Unutar te funkcije nalazi se funkcija user_login() koja za parametre poprima vrijednosti iz varijabli \$email i \$password. Slika 23. prikazuje funkciju user_login().

Slika 23. Funkcija user_login().

```
163 function user_login($email, $password){
164     global $conn;
165
166     $query = "SELECT * FROM users WHERE email='$email'";
167     $result = mysqli_query($conn, $query);
168     $provjera = mysqli_num_rows($result);
169
170     if($provjera > 0){
171         $data = mysqli_fetch_assoc($result);
172
173         if(password_verify($password, $data['password'])){
174             $_SESSION['email'] = $email;
175             return true;
176         } else{
177             return false;
178         }
179     }
180 }
```

Varijabla \$query sadži SQL naredbu za odabir svih podataka iz reda tablice users gdje je email polje identično email adresi iz varijable \$email. Funkcija iz MySQLi ekstenzije – mysqli_query() – povezuje se s bazom podataka i izvršava SQL naredbu. Podatke sprema u varijablu \$result. Ako su

podatci o korisniku pronađeni, odnosno ako u tablici postoji red gdje je email adresa jednaka email adresi koju je upisao korisnik, tada se provjerava ispravnost lozinke pomoću `password_verify()` funkcije. Ta funkcija usporedit će *hash* vrijednost upisane lozinke sa *hash-om* koji se nalazi u tom redu tablice. Ako je lozinka ispravna u sesiju se pohranjuje podatak o email adresi s kojom je korisnik prijavljen.

Nakon izvršenja `user_login()` funkcije nastavlja se izvršavanje `user_validation()` funkcije. Ako su podatci za prijavu netočni ispisuje se poruka CSS klase „*error*“: „Podatci za prijavu nisu ispravni. Pokušajte ponovno.“

No, ako su podatci ispravni korisnika se preusmjerava na stranicu `index.php` uz pomoć `redirect()` funkcije. Korisnik je tada prijavljen i može koristiti funkcionalnosti aplikacije namijenjene registriranim korisnicima, a to su unos primitaka i izdataka u knjigu primitaka i izdataka, te prikaz Obrasca KPI. Uz to je moguće promijeniti i podatke o korisniku što je potrebno napraviti ako korisnik ima novu adresu prebivališta, novu djelatnost i sl. ili ako želi promijeniti podatke za prijavu, tj. email adresu i lozinku. Za korištenje navedenih funkcionalnosti aplikacije od velike je važnosti funkcija `get_user()` koja pohranjuje podatke o korisniku u varijablu `$user`. Slika 24. prikazuje funkciju `get_user()`.

Slika 24. Funkcija `get_user()`.

```
183 function get_user(){
184     global $conn;
185     $query = "SELECT * FROM users WHERE email='" . $_SESSION['email'] . "'";
186     $result = mysqli_query($conn, $query);
187     $provjera = mysqli_num_rows($result);
188
189     if($provjera > 0){
190         return mysqli_fetch_assoc($result);
191     } else{
192         return "Korisnik nije pronađen.";
193     }
194 }
```

Funkcijom se dobivaju svi podatci o korisniku čija je email adresa pohranjena u sesiji, odnosno podatci o prijavljenom korisniku. Ako su podatci pronađeni u bazi podataka funkcijom `mysqli_fetch_assoc()` se isti pohranjuju u asocijativni niz koji funkcija vraća na stranici gdje se ista koristi. U suprotnom vraća *string* „Korisnik nije pronađen.“

8.3.4. Datoteka profil.php

Na stranici profil.php korisnik može provjeriti svoje podatke te ih izmijeniti ako za to ima potrebe. Podatci o korisniku se dobivaju pomoću `get_user()` funkcije. Slika 25. prikazuje datoteku profil.php.

Slika 25. Datoteka profil.php.

```
1 <?php
2 include "inc/header.php";
3 $user = get_user();
4 validate_and_enter_changes($user);
5 ?>
6
7 <h2>Korisnički profil</h2>
8 <div class="flex-container">
9   <div>
10    <h3>Korisnički podatci:</h3>
11    <?php
12      echo "Naziv poduzeća: " . $user['naziv_poduzeca'] . "<br>";
13      echo "Ime poduzetnika: " . $user['ime_poduzetnika'] . "<br>";
14      echo "Ulica: " . $user['ulica'] . "<br>";
15      echo "Poštanski broj: " . $user['postanski_broj'] . "<br>";
16      echo "Mjesto: " . $user['mjesto'] . "<br>";
17      echo "Naziv djelatnosti: " . $user['naziv_djelatnosti'] . "<br>";
18      echo "Šifra djelatnosti: " . $user['sifra_djelatnosti'] . "<br>";
19      echo "Email: " . $user['email'] . "<br>";
20      echo "Broj telefona: " . $user['tel'] . "<br>";
21      echo "OIB: " . $user['oib'] . "<br>";
22    ?>
23  </div>
24  <div>
25    <h3>Izmjena podataka:</h3>
26    <form method="POST">
27      <input type="text" name="naziv_poduzeca" placeholder="Naziv poduzeća">
28      <input type="text" name="ime_poduzetnika" placeholder="Ime poduzetnika">
29      <input type="text" name="ulica" placeholder="Ulica">
30      <input type="text" name="postanski_broj" placeholder="Poštanski broj">
31      <input type="text" name="mjesto" placeholder="Grad/mjesto">
32      <input type="text" name="naziv_djelatnosti" placeholder="Naziv djelatnosti">
33      <input type="text" name="sifra_djelatnosti" placeholder="Šifra djelatnosti">
34      <input type="email" name="email" placeholder="Email">
35      <input type="text" name="tel" placeholder="Telefonski broj">
36      <input type="password" name="password" placeholder="Lozinka">
37      <input type="password" name="confirm_password" placeholder="Potvrda lozinke">
38      <input type="submit" name="change_submit" value="Promjeni">
39    </form>
40  </div>
41 </div>
42
43 <?php include "inc/footer.php"?>
```

Sadržaj ove stranice sastoji se od dva dijela koji su određeni *Flexbox*-om. *CSS Flexible Box Layout* (fleksibilni rasporedni obrazac), skraćeno *Flexbox*, je način rasporeda sadržaja na web stranici. U ovom primjeru *Flexbox* omogućuje da se s lijeve strane prikazuje ispis korisničkih podataka, dok se s desne strane nalazi forma za promjenu tih podataka. Kod manjih ekrana kao što su mobilni telefoni, ti dijelovi će se prikazivati jedan ispod drugoga. Svojstva „*flex-container klase*“ definirana su CSS kodom prikazanim slikom 26.

Slika 26. Klasa "*flex-container*" (Datoteka style.css).

```
66  .flex-container {
67    display: flex;
68    flex-wrap: wrap;
69  }
70
71  .flex-container div {
72    padding: 10px;
73    text-align: left;
74    margin-right: 15px;
75    margin-bottom: 20px;
76    line-height: 40px;
77  }
```

Forma funkcionira jednako kao i prethodne, no za razliku od njih, input oznake ne sadrže argument *required* što znači da nije potrebno popuniti sva polja već samo ona koja korisnik želi izmijeniti.

Nakon unosa željenih podataka i klika na gumb „Promijeni” izvršava se funkcija `validate_and_enter_changes()` koja za parametar poprima podatke o korisniku iz varijable `$user`.

Slika 27. prikazuje funkciju `validate_and_enter_changes()`.

Slika 27. Funkcija `validate_and_enter_changes()`.

```
197 function validate_and_enter_changes($user){
198     global $conn;
199     $id = $user['user_id'];
200
201     if($_SERVER['REQUEST_METHOD'] == 'POST'){
202
203         if(!empty($_POST['password'])){
204             if($_POST['password'] == $_POST['confirm_password']){
205                 $password = password_hash(htmlentities($_POST['password']), algo:PASSWORD_DEFAULT);
206                 $sql = "UPDATE users SET password = '{$password}' WHERE user_id = {$id}";
207                 confirm(mysqli_query($conn, $sql));
208             }else{
209                 echo '<div class="error">Lozinke se ne podudaraju.</div>';
210             }
211         }
212
213         $popis_elementa_za_izmjenu = ["naziv_poduzeca", "ime_poduzetnika", "ulica",
214             "postanski_broj", "mjesto", "naziv_djelatnosti", "sifra_djelatnosti", "email", "tel"];
215
216         foreach($popis_elementa_za_izmjenu as $element){
217             if(empty($_POST["{$element}"])){
218                 continue;
219             } else{
220                 $za_unos = htmlentities($_POST[$element]);
221                 $sql = "UPDATE users SET {$element} = '{$za_unos}' WHERE user_id = {$id}";
222                 confirm(mysqli_query($conn, $sql));
223                 redirect("profil.php");
224             }
225         }
226     }
227 }
```

Nakon provjere jesu li uneseni ikakvi podatci u formi provjerava se je li unesen podatak pod ključnom riječju „*password*”. Ako su uneseni, provjerava se je li unesena lozinka, te ako je, je li jednaka lozinka unesena u polje za unos i u polje za ponovni unos lozinke. U slučaju da se lozinke ne podudaraju ispisuje se odgovarajuća poruka klase „*error*”. Ako se podudaraju, tada se unesena lozinka *hashira* istim postupkom kao i kod registracije korisnika te se novi *hash* unosi u bazu podataka. Potom je kreiran vektor s elementima iz tablice *users*. *foreach* petljom se iterira kroz elemente tog vektora te se provjerava je li u formi unesen podatak vezan za taj element. SQL naredbom se radi promjena u bazi podataka.

8.3.5. Datoteka unos.php

Unos primitaka i izdataka u knjigu primitaka i izdataka vrši se na stranici unos.php. Formom se prikupljaju potrebni podatci o primitku ili izdatku. Korisnik upisuje podatke o nazivu poslovnog subjekta, datum, broj temeljnice, opis te iznose upisuje u odgovarajuće polje. Odabire je li riječ o ulaznom ili izlaznom računu, na temelju čega će funkcija unos_racuna() te iznose unijeti kao prihod ili izdatak u bazu podataka. Slika 28. prikazuje datoteku unos.php.

Slika 28. Datoteka unos.php.

```
1  <?php include"inc/header.php"; ?>
2
3  <?php
4  $user = get_user();
5  unos_racuna($user);
6  ?>
7
8  <h2>Unos u Knjigu primitaka i izdataka</h2>
9  <form method="POST">
10     <input type="text" name="klijent" placeholder="Naziv" required>
11     <input type="date" name="datum" required>
12     <input type="text" name="br_temeljnice" placeholder="Broj temeljnice" required>
13     <input type="text" name="opis" placeholder="Opis" required>
14     <br><br>
15     <input class="radio" type="radio" id="ura" name="tip_racuna" value="URA">
16     <label for="html">Ulazni račun</label><br>
17     <input class="radio" type="radio" id="ira" name="tip_racuna" value="IRA">
18     <label for="html">Izlazni račun</label><br>
19     <br>
20     Izdaci/primici u gotovini: <input type="number" name="gotovina"><br>
21     Izdaci/primici na žiro-računu: <input type="number" name="ziro_racun"><br>
22     Izdaci/primici u naravi: <input type="number" name="narav"><br>
23     PDV u izdacima/primicima: <input type="number" name="pdv"><br>
24     Izdaci iz čl.33. st.1. t. 1. i 5.: <input type="number" name="izdaci_cl_33">
25     <input type="submit" name="unos-racuna" value="Unesi račun">
26 </form>
27
28 <?php include"inc/footer.php"; ?>
```

Slika 29. Prikazuje funkciju unos_racuna().

Slika 29. Funkcija unos_racuna().

```
228 function unos_racuna($user){
229     global $conn;
230     $oib = $user['oib'];
231
232     if($_SERVER['REQUEST_METHOD'] == 'POST'){
233         $tip_racuna = htmlentities($_POST['tip_racuna']);
234         $naziv_subjekta = htmlentities($_POST['klijent']);
235         $datum = htmlentities($_POST['datum']);
236         $br_temeljnice = htmlentities($_POST['br_temeljnice']);
237         $opis = htmlentities($_POST['opis']);
238
239         if($tip_racuna == "IRA"){
240             #PRIMICI
241             $primici_gotovina = htmlentities($_POST['gotovina']);
242             $primici_ziro_racun = htmlentities($_POST['ziro_racun']);
243             $primici_u_naravni = htmlentities($_POST['narav']);
244             $primici_pdv = htmlentities($_POST['pdv']);
245
246             #IZRAČUN
247             $primici_ukupno = floatval($primici_gotovina) + floatval($primici_ziro_racun)
248                 + floatval($primici_u_naravni) - floatval($primici_pdv);
249
250             $sql = "INSERT INTO {$oib}_kpi(datum, br_temeljnice, opis, naziv_subjekta, tip_racuna,
251                 primici_gotovina, primici_ziro_racun, primici_u_naravni, primici_pdv, primici_ukupno) ";
252             $sql .= "VALUES('$datum', '$br_temeljnice', '$opis', '$naziv_subjekta', '$tip_racuna',
253                 '$primici_gotovina', '$primici_ziro_racun', '$primici_u_naravni', '$primici_pdv',
254                 '$primici_ukupno')";
255             confirm(mysqli_query($conn, $sql));
256         }
257
258         if($tip_racuna == "URA"){
259             #IZDACI
260             $izdaci_gotovina = htmlentities($_POST['gotovina']);
261             $izdaci_ziro_racun = htmlentities($_POST['ziro_racun']);
262             $izdaci_u_naravni = htmlentities($_POST['narav']);
263             $izdaci_pdv = htmlentities($_POST['pdv']);
264             $izdaci_cl_33 = htmlentities($_POST['izdaci_cl_33']);
265
266             #IZRAČUN
267             $izdaci_ukupno = floatval($izdaci_gotovina) + floatval($izdaci_ziro_racun)
268                 + floatval($izdaci_u_naravni) - floatval($izdaci_pdv) - floatval($izdaci_cl_33);
269
270             $sql = "INSERT INTO {$oib}_kpi(datum, br_temeljnice, opis, naziv_subjekta, tip_racuna,
271                 izdaci_gotovina, izdaci_ziro_racun, izdaci_u_naravni, izdaci_pdv, izdaci_cl_33, izdaci_ukupno)";
272             $sql .= "VALUES('$datum', '$br_temeljnice', '$opis', '$naziv_subjekta', '$tip_racuna',
273                 '$izdaci_gotovina', '$izdaci_ziro_racun', '$izdaci_u_naravni', '$izdaci_pdv', '$izdaci_cl_33',
274                 '$izdaci_ukupno')";
275             confirm(mysqli_query($conn, $sql));
276         }
277     }
278 }
```

U varijablu \$oib pohranjuje se OIB prijavljenog korisnika. Ako su podatci uspješno pohranjeni u asocijativni niz \$_POST; naziv subjekta, datum, broj temeljnice, opis i tip računa (ulazni ili izlazni)

pohranjuju se u odgovarajuće varijable. Potom se provjerava tip računa. Ako je izlazni račun tada se uneseni iznosi unose kao primitak, a ako je ulazni račun kao izdatak. U oba slučaja se radi izračun ukupnih primitaka ili izdataka, no prvo se na varijable koje sadrže iznose primjenjuje `floatval()` funkcija kako bi iznosi bili tipa *float* (decimalni broj). Varijabli `$sql` se dodjeljuje SQL funkcija putem kojeg se u tablicu knjige primitaka i izdataka u novi red upisuju osnovni podatci o transakciji (datum, opis i sl), te iznosi primitaka ili izdataka. U slučaju izlaznog računa upisuju se iznosi u stupce vezane za primitke, a u stupcima koji su vezani za izdatke neće biti pohranjeni iznosi, odnosno bit će tipa `NULL`. Ugrađenom PHP funkcijom `mysqli_query()` se povezuje aplikacija sa bazom podataka i izvršava se navedena SQL naredba.

8.3.6. Datoteka `izvjestaji.php`

Za pregled knjige primitaka i izdataka, odnosno Obrasca KPI koristi se stranica `izvjestaji.php`. Slika 30. prikazuje datoteku `izvjestaji.php`.

Slika 30. Datoteka `izvjestaji.php`.

```
1  <?php include "inc/header.php";?>
2
3  <h2>Izvještaji</h2>
4  <h3>Obrazac KPI</h3>
5
6  <form action="/KPI/KPI.php" method="POST">
7    <input type="date" name="pocetak" placeholder="Početni datum">
8    <input type="date" name="kraj" placeholder="Završni Datum">
9
10   <input type="submit" name="izradi_kpi" value="Izradi Obrazac KPI">
11 </form>
12
13 <?php include "inc/footer.php" ?>
```

Ova forma sadrži samo dva polja. Klikom na njih otvara se kalendar te se na jednom polju unosi početni datum, a na drugom završni datum. Nakon odabira datuma potrebno je kliknuti na dugme „Izradi Obrazac KPI”. Za razliku od ranije spomenutih formi, ova sadrži ključnu riječ `action` koja sadrži lokaciju stranice `KPI.php`. Time se informacije unesene u formi šalju na tu stranicu, te se korisnika tamo preusmjerava.

8.3.7. Datoteka KPI.php

Za razliku od ostalih stranica, `tablica.php` ne uključuje `header.php` i `footer.php` (zaglavlje i podnožje). Na taj način ne koristi CSS datoteku `style.css` koja definira izgled ostatka stranice, već je izgled tablice definiran style oznakama na samoj stranici. Dodatno su određeni redovi uređeni i style argumentima u pripadajućim linijama koda.

Umjesto zaglavlja i podnožja uključen je `init.php` čime stranica ima pristup sesiji i datotekama `db.php` i `functions.php`. Time je omogućen pristup podacima prijavljenog korisnika koji su potrebni za kreiranje obrasca KPI.

Pomoću HTML oznaka kreirana je tablica koja sadrži osnovne podatke o poreznom obvezniku koji se dobivaju iz `$user` asocijativnog niza. Potom se u tablici ispisuju nazivi stupaca i njihovi redni brojevi.

Slika 31. prikazuje datoteku `KPI.php` (1. dio).

Slika 31. Datoteka KPI.php (1. dio).

```
1 <style>
2 table, th, td {
3   border: 1px solid black;
4 }
5 </style>
6
7 <?php
8 include "config/init.php";
9 $user = get_user();
10 ?>
11
12 <table>
13 <tr>
14   <td colspan="15"><b>I. OPĆI PODACI O POREZKOM OBVEZNIKU</b></td>
15 </tr>
16 <tr>
17   <td colspan="10">1. NAZIV DJELATNOSTI <?php echo $user['naziv_djelatnosti']?></td>
18   <td colspan="7">ŠIFRA DJELATNOSTI <?php echo $user['sifra_djelatnosti']?></td>
19 </tr>
20 <tr>
21   <td colspan="15">2. IME I PREZIME PODUZETNIKA/NOSITELJA ZAJEDNIČKE DJELATNOSTI
22     <?php echo $user['ime_poduzetnika']?></td>
23 </tr>
24 <tr>
25   <td colspan="15">3. ADRESA PREBIVALIŠTA/UOBIČAJENOG BORAVIŠTA
26     <?php echo "{$user['ulica']}, {$user['postanski_broj']} {$user['mjesto']}"?></td>
27 </tr>
28 <tr>
29   <td colspan="15">4. OIB PODUZETNIKA/NOSITELJA ZAJEDNIČKE DJELATNOSTI
30     <?php echo $user['oib']?></td>
31 </tr>
```

Slika 31. prikazuje datoteku KPI.php (2. dio).

Slika 32. Datoteka KPI.php (2. dio).

```
33 <tr style="text-align: center; vertical-align: middle;">
34   <td rowspan="2">R.BR.</td>
35   <td rowspan="2">NADNEVAK PRIMITKA/IZDATKA</td>
36   <td rowspan="2">BROJ TEMELJNICE</td>
37   <td rowspan="2">OPIS ISPRAVA</td>
38   <td colspan="3">PRIMICI</td>
39   <td rowspan="2">PDV U PRIMICIMA</td>
40   <td rowspan="2">UKUPNI PRIMICI</td>
41   <td colspan="3">IZDACI</td>
42   <td rowspan="2">PDV U IZDACIMA</td>
43   <td rowspan="2">IZDACI IZ ČL. 33. ST. 1. T. 1. I 5.</td>
44   <td rowspan="2">UKUPNO DOPUŠTENI IZDACI</td>
45 </tr>
46 <tr style="text-align: center; vertical-align: middle;">
47   <td>U GOTOVINI</td>
48   <td>NA ŽIRO-RAČUN</td>
49   <td>U NARAVNI</td>
50   <td>U GOTOVINI</td>
51   <td>NA ŽIRO-RAČUN</td>
52   <td>U NARAVNI</td>
53 </tr>
54
55 <tr style="text-align: center; vertical-align: middle;">
56   <td>1</td>
57   <td>2</td>
58   <td>3</td>
59   <td>4</td>
60   <td>5</td>
61   <td>6</td>
62   <td>7</td>
63   <td>8</td>
64   <td>9(5+6+7-8)</td>
65   <td>10</td>
66   <td>11</td>
67   <td>12</td>
68   <td>13</td>
69   <td>14</td>
70   <td>15 (10+11+12-13-14)</td>
71 </tr>
72
73 <?php create_KPI($user); ?>
74
75 </table>
```

Dinamički dio tablice ispisuje se pomoću create_KPI() funkcije čiji je kod vidljiv na slici 33.

Slika 33: Funkcija create_KPI()

```
275 function create_KPI($user){
276     global $conn;
277     $oib = $user['oib'];
278
279     if($_SERVER['REQUEST_METHOD'] == 'POST'){
280         $pocetni_datum = htmlentities($_POST['pocetak']);
281         $zavrsni_datum = htmlentities($_POST['kraj']);
282
283         $query = "SELECT * FROM {$oib}_kpi WHERE datum BETWEEN
                '$pocetni_datum' AND '$zavrsni_datum'";
284         $result = mysqli_query($conn, $query);
285         $provjera = mysqli_num_rows($result);
286         if($provjera > 0){
287             upis_u_tablicu($result);
288         }
289     }
290     else{
291         echo "Nema podataka za trazeno razdoblje.";
292     }
293 }
```

Kao i neke prethodne funkcije i ova koristi varijablu \$oib. Provjerava jesu li odabrani datumi u formi za izradu obrasca, ako jesu sprema ih u odgovarajuće varijable. U varijablu \$query pohranjuje se SQL naredba za odabir podataka iz KPI tablice prijavljenog korisnika za promatrano razdoblje. Funkcijom mysqli_query() izvršava se ta naredba. Potom se funkcijom mysqli_num_rows(), koja je dodijeljena varijabli \$provjera, provjerava jesu li pronađeni podatci za promatrano razdoblje. Ako jesu, broj redova iz tablice koji su odabrani će biti veći od nule i tada se izvršava sljedeća funkcija – upis_u_tablicu() uz parametar \$result koji sadrži odabrane podatke iz tablice KPI. Ukoliko nema podataka u promatranom razdoblju ispisat će se odgovarajuća poruka.

Slika 34. prikazuje funkciju upis_u_tablicu().

Slika 34. Funkcija upis_u_tablicu().

```
298 function upis_u_tablicu($result){
299     $redni_broj = 1;
300     while($row = mysqli_fetch_assoc($result)){
301         $datum = date('d.m.Y.', strtotime($row['datum']));
302         $br_temeljnice = $row['br_temeljnice'];
303         $opis = $row['opis'];
304         $primici_gotovina = $row['primici_gotovina'];
305         $primici_ziro_racun = $row['primici_ziro_racun'];
306         $primici_u_naravni = $row['primici_u_naravni'];
307         $primici_pdv = $row['primici_pdv'];
308         $primici_ukupno = $row['primici_ukupno'];
309         $izdaci_gotovina = $row['izdaci_gotovina'];
310         $izdaci_ziro_racun = $row['izdaci_ziro_racun'];
311         $izdaci_u_naravni = $row['izdaci_u_naravni'];
312         $izdaci_pdv = $row['izdaci_pdv'];
313         $izdaci_cl_33 = $row['izdaci_cl_33'];
314         $izdaci_ukupno = $row['izdaci_ukupno'];
315
316         echo"<tr>";
317         echo"<td> $redni_broj" . "." . "</td>";
318         echo"<td> $datum </td>";
319         echo"<td> $br_temeljnice </td>";
320         echo"<td> $opis </td>";
321         echo"<td> $primici_gotovina </td>";
322         echo"<td> $primici_ziro_racun</td>";
323         echo"<td> $primici_u_naravni</td>";
324         echo"<td> $primici_pdv</td>";
325         echo"<td> $primici_ukupno</td>";
326         echo"<td> $izdaci_gotovina</td>";
327         echo"<td> $izdaci_ziro_racun</td>";
328         echo"<td> $izdaci_u_naravni</td>";
329         echo"<td> $izdaci_pdv</td>";
330         echo"<td> $izdaci_cl_33</td>";
331         echo"<td> $izdaci_ukupno</td>";
332         echo"<tr>";
333
334         $redni_broj++;
335     }
336 }
```

Za potrebe *while* petlje na početku se iterirajućoj varijabli `$redni_broj` dodjeljuje broj 1. Iteracija kreće od broja jedan, a ne nula, kako je uobičajeno, jer se `$redni_broj` ispisuje u svrhu prikazivanja rednog broja transakcije u Obrascu KPI.

Varijabli `$row` se uz `mysqli_fetch_assoc()` funkciju dodjeljuje višedimenzionalni asocijativni niz koji sadrži podatke iz svih odabranih redova tablice. Svaka iteracija je jedan element tog niza, odnosno

jedan red iz tablice. Potom se svaki element iz tog reda dodjeljuje u odgovarajuću varijablu. Tako se u \$datum pohranjuje datum, no prvo se izvršava ugrađena funkcija date() koja datum iz baze podataka pretvara u format koji prvo prikazuje dan, potom mjesec i godinu. Na ostale podatke nije potrebno primjenjivati funkcije.

Kada sve navedene varijable poprime odgovarajuću vrijednost, iste se ispisuju u novi red tablice pomoću HTML tagova. \$redni_broj se uvećava za 1 i nastavlja se iteracija iz while petlje sve dok se ne ispušu svi podaci iz niza. Rezultat tih ispisa je Obrazac KPI koji sadrži sve zakonom propisane elemente.

9. MOGUĆNOSTI DALJNJEG RAZVOJA PROGRAMSKE APLIKACIJE I MOGUĆA POBOLJŠANJA

Izrađena programska aplikacija ima daljnje razvojne mogućnosti kojima bi se upotpunila i poboljšala usluga koju aplikacija nudi. Programska aplikacija knjige primitaka i izdataka samo po sebi nije dovoljna za ispunjavanje svih zakonskih obveza iz Zakona i Pravilnika o porezu na dohodak. Porezni obveznik, odnosno obrtnik, mora voditi i druge poslovne knjige. Programska bi se aplikacija mogla nadograditi s dodatnim modulima tako da podržava vođenje svih poslovnih knjiga koje bi međusobno bile povezane.

Za komercijalnu je upotrebu razvijene programske aplikacije potrebno implementirati sustav plaćanja prilikom registracije. Korištenjem radnog okruženja kao što su Laravel ili Symfony takva implementacija bi bila jednostavnije nego korištenjem nativnog PHP-a, a također bi se poboljšala sigurnost, autentikacija korisnika i slično.

Uporabom i programskog jezika koji se izvršava na strani klijenta kao što je JavaScript dobile bi se dodatne funkcionalnosti u razvijenoj programskoj aplikaciji koje trenutno nisu moguće. Primjerice, mogao bi biti isprogramiran kalendar za odabir datuma koji bi koristio drugi, odgovarajući format datuma. Također bi se mogao poboljšati vizualni aspekt ove web programske aplikacije .

Popratna bi dokumentacija ili upute za uporabu razvijene programske aplikacije korisnicima omogućila lakše i uspješnije korištenje programske aplikacije.

10. ZAKLJUČAK

Sukladno pozitivnim zakonskim propisima obrtnici su dužni voditi evidenciju o svojim primitcima i izdatcima u knjizi primitaka i izdataka. U današnje se vrijeme za vođenje takve evidencije najčešće upotrebljavaju računalni programi. No, zbog relativno visokih troškova, posebice malim obrtima ili obrtima kada su u začetku poslovanja, skupa programska rješenja koja podrazumijevaju nabavku uz programske i skupe strojne opreme te redovito održavanje programske i strojne opreme, nisu opcija. Stoga se, kao alternativa skupim rješenjima nameću jednostavne web programske aplikacije kojima obrtnici mogu ispuniti svoje zakonske obveze vezane za vođenje poslovnih knjiga, pa tako i knjige primitaka i izdataka. S obzirom na kontinuiranu potrebu razvija programskih rješenja za obrtničku evidenciju, posebno onih jednostavnijih i prilagodljivih potrebama malih obrta i obrta u začetku, provedeno je istraživanje radi sagledavanja mogućnosti kreiranja upotrebljivog programskog rješenja knjige primitaka i izdataka načinjene na web platformi. Tijekom istraživanja istražene su popularne i besplatne tehnologije za razvoj programskih web aplikacija kao što su PHP, HTML, CSS i MySQL. Da bi se moglo pristupiti izradi upotrebljivog programskog rješenja za knjigu primitaka i izdataka proučena je bitna zakonska regulativa vezanu za vođenje knjige primitaka i izdataka, kao i prednosti i nedostatke postojećih programskih rješenja na tržištu, kako bi se definiralo optimalno programsko rješenje. U konačnici rezultati su istraživanja pokazali kako se korištenjem mrežnih tehnologija može razviti web programska aplikacija kojom obrtnici mogu ispuniti zakonsku obvezu vezanu uz vođenje knjige primitaka i izdataka. Samo je istraživanje provedeno u dvije faze. Desk istraživanjima je sagledana zakonska regulativa i web tehnologije pogodne za razvoj programskog rješenja knjige primitaka i izdataka, dok su u okviru primarnih istraživanja sagledana raspoloživa postojeća programska rješenja namijenjena vođenju knjige primitaka i izdataka kako bi se sagledala njihova funkcionalnost i otkrile prednosti i mane, a u drugom koraku u okviru primarnih istraživanja modelirana je, a potom i kodirana te testirana programska aplikacija knjige primitaka i izdataka. U istraživanju je primijenjeno više znanstvenih metoda među kojima su svakako najznačajnije metoda apstrakcije, metoda klasifikacije, metoda generalizacije, metoda specijalizacije, metoda kompozicije, metoda abrogacije, metoda analogije, metoda kauzalnog zaključivanja, te metoda deskriptivnog modeliranja.

Izrađena programska aplikacija ima daljnje razvojne mogućnosti kojima bi se upotpunila i poboljšala usluga koju aplikacija nudi. Definirana programska aplikacija knjige primitaka i izdataka samo po sebi nije dovoljna za ispunjavanje svih zakonskih obveza iz Zakona i Pravilnika o porezu na dohodak.

Porezni obveznik, odnosno obrtnik, mora voditi i druge poslovne knjige. Stoga daljnja istraživanja mogu ići prema nadogradnji programske aplikacije dodatnim modulima tako da podržava vođenje svih poslovnih knjiga koje bi međusobno bile povezane, a koje bi obrtnicima omogućile jednostavno, učinkovito i ergonomično vođenje poslovnih knjiga.

11. LITERATURA

1. Amidžić-Peročević, K. et al.: Poslovanje obrtnika 2002., Hrvatska zajednica računovođa i financijskih djelatnika, Zagreb, 2002.
2. Beaulieu, A.: Learning SQL, O'Reilly Media, Inc., Sebastepol, 2020.
3. Beighley, L. i Morrison, M.: Head First PHP & MySQL, O'Reilly Media, Inc., Sebastepol, 2008.
4. Čebetarević, V. et al.: Poslovanje obrtnika i slobodnih zanimanja, RRiF-plus d.o.o., Zagreb, 2006.
5. Hrvatska obrtnička komora (HOK): Vodič za poslovanje u obrtu., 2021. [Online] dostupno na: https://www.hok.hr/sites/default/files/page-docs/2022-04/VODIC_za_poslovanje_u_obrtu.pdf [pristupljeno: 17.8.2022.]
6. Meyer, E.: CSS Pocket Reference, O'Reilly Media, Inc., Sebastepol, 2018.
7. Murach, J. (2019). Murach's MySQL, 3rd ed., Fresno, CA: Mike Murach & Associates, Inc.
8. Narodne novine: Odluka o Nacionalnoj klasifikaciji djelatnosti 2007. - NKD 2007. Narodne Novine d.d., Zagreb, br. 58/07.
9. Narodne novine: Pravilnik o obrtima koji se mogu obavljati u stambenim prostorijama. Narodne Novine d.d., Zagreb, br. 101/95.
10. Narodne novine: Pravilnik o paušalnom oporezivanju samostalnih djelatnosti. Narodne Novine d.d., Zagreb, br. 1/20, 1/21.
11. Narodne novine: Pravilnik o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 10/17, 128/17.
12. Narodne novine: Pravilnik o tradicijskim, odnosno umjetničkim obrtima. Narodne Novine d.d., Zagreb, br. 112/07.
13. Narodne novine: Zakon o obrtu. Narodne Novine d.d., Zagreb, br. 141/13, 127/19.
14. Narodne novine: Zakon o porezu na dohodak. Narodne Novine d.d., Zagreb, br. 118/16, 106/18, 112/19, 138/20, 32/20.
15. Olsson, M.: PHP 8 Quick Scripting Reference: A Pocket Guide to PHP Web Scripting, Apress Media LLC, New York, 2021.
16. Oracle.: MySQL 8.0 Reference Manual, 2022. [Online] dostupno na: <https://downloads.mysql.com/docs/refman-8.0-en.a4.pdf> [pristupljeno: 24.8.2022.]
17. Robson, E. i Freeman, E.: Head First HTML and CSS, O'Reilly Media, Inc., Sebastepol, 2012.
18. Tatroe, K. i MacIntyre, P.: Programming PHP, O'Reilly Media, Inc., Sebastepol, 2020.

19. The phpMyAdmin devel team: phpMyAdmin Documentation, 2022. [Online] dostupno na: https://docs.phpmyadmin.net/_downloads/en/latest/pdf/ [pristupljeno: 25.8.2022.]
20. Web Hypertext Application Technology Working Group (WHATWG): HTML Living Standard, 2022. [Online] dostupno na: <https://html.spec.whatwg.org/print.pdf> [pristupljeno: 31.8.2022.]
21. Apache Friends: XAMPP [Online] dostupno na: <https://www.apachefriends.org/index.html> [pristupljeno 25.8.2022.]
22. GitHub: Atom Flight Manual. [Online] dostupno na: <https://flight-manual.atom.io/getting-started/sections/why-atom/> [pristupljeno: 25.8.2022.]
23. The Apache Software Foundation (ASF): About the Apache [Online] dostupno na: https://httpd.apache.org/ABOUT_APACHE.html [pristupljeno: 25.8.2022.]
24. The PHP Group: New archive - PHP 8.0.0 Released!., 2020. [Online] dostupno na: <https://www.php.net/archive/2020.php#2020-11-26-3> [pristupljeno: 25.8.2022.]
25. The PHP Group: PHP Manual – password_hash. [Online] dostupno na: <https://www.php.net/manual/en/function.password-hash.php> [pristupljeno: 1.9.2022.]
26. The PHP Group: PHP Manual – What is PHP?. [Online] dostupno na: <https://www.php.net/manual/en/intro-what-is.php> [pristupljeno: 1.9.2022.]
27. Soltano, S.: Web technology fact of the day, 2020. [Online] https://w3techs.com/blog/entry/fact_20200414 [pristupljeno: 5.9.2022.]
28. W3Techs: Usage statistics of PHP for websites. [Online] <https://w3techs.com/technologies/details/pl-php> [pristupljeno 4.9.2022.]
29. W3Techs: Usage statistics of server-side programming languages for websites. [Online] https://w3techs.com/technologies/overview/programming_language [pristupljeno 4.9.2022.]
30. World Wide Web Consortium (W3C): HTML & CSS. [Online] dostupno na: <https://www.w3.org/standards/webdesign/htmlcss> [pristupljeno: 24.8.2022.]

Popis tablica

Tablica 1: Porezna osnovica kod paušalnih obrta (u kn)	14
--------------------------------------------------------------	----

Popis grafikona

Grafikon 1: Najčešće korišteni jezici na strani poslužitelja **Error! Bookmark not defined.**

Grafikon 2: Najčešće korištene verzije PHP-a **Error! Bookmark not defined.**

Popis slika

Slika 1: Izgled XAMPP sučelja **Error! Bookmark not defined.**

Slika 2: Shematski prikaz odnosa između klijenta, aplikacije i baze podataka **Error! Bookmark not defined.**

Slika 3: Kreiranje tablice "users" **Error! Bookmark not defined.**

Slika 4: Model baze podataka **Error! Bookmark not defined.**

Slika 5: Prikaz organizacije koda aplikacije..... **Error! Bookmark not defined.**

Slika 6: Datoteka init.php..... **Error! Bookmark not defined.**

Slika 7: Datoteka db.php..... 30

Slika 8: Datoteka header.php..... 31

Slika 9: Datoteka footer.php..... 32

Slika 10: Datoteka style.css (prvi dio)..... 32

Slika 11: Datoteka index.php..... 33

Slika 12: Datoteka registracija.php..... 34

Slika 13: Funkcija validate_registration()..... 35

Slika 14: Funkcija email_already_in_use()..... 36

Slika 15: Klasa "error" (datoteka style.css) 37

Slika 16: Funkcija create_user()..... 38

Slika 17: Funkcija create_table()..... 39

Slika 18: Funkcija set_message() 40

Slika 19: Funkcija redirect()..... 40

Slika 20: Datoteka prijava.php	41
Slika 21: Funkcija display_message()	41
Slika 22: Funkcija login_validation()	42
Slika 23: Funkcija user_login()	42
Slika 24: Funkcija get_user()	43
Slika 25: Datoteka profil.php	44
Slika 26: Klasa " <i>flex-container</i> " (Datoteka style.css).....	45
Slika 27: Funkcija validate_and_enter_changes().....	46
Slika 28: Datoteka unos.php.....	47
Slika 29: Funkcija unos_racuna()	48
Slika 30: Datoteka izvjestaji.php	49
Slika 31: Datoteka KPI.php (1. dio)	51
Slika 32: Datoteka KPI.php (2. dio)	52
Slika 33: Funkcija create_KPI()	53
Slika 34: Funkcija upis_u_tablicu().....	54