

Dizajn, razvoj i implementacija baze podataka za potrebe privatne knjižnice

Mišković, Ivana

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:131:613185>

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

Download date / Datum preuzimanja: **2024-08-28**



Sveučilište u Zagrebu
Filozofski fakultet
University of Zagreb
Faculty of Humanities
and Social Sciences

Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
SMJER BIBLIOTEKARSTVO
SMJER ISTRAŽIVAČKA INFORMATIKA
Ak. god. 2023./2024.

Ivana Mišković

**Dizajn, razvoj i implementacija baze podataka za potrebe
privatne knjižnice**

Interdisciplinarni diplomski rad

Mentori: dr. sc. Ivan Dunder, doc.; dr. sc. Tomislav Ivanjko, izv. prof.

Zagreb, ožujak 2024.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

(potpis)

Sadržaj

1. Uvod.....	1
2. Privatna knjižnica.....	3
3. Informacijska i komunikacijska tehnologija upravljanja knjižnicom.....	4
4. Aplikacije.....	7
4.1. Mobilne aplikacije.....	7
4.2. Mrežne aplikacije.....	9
5. Računalstvo u „oblaku“	13
5.1. Integrirani knjižnični sustavi u „oblaku“	14
6. Dizajn aplikacija	15
7. Primjeri knjižničnih aplikacija za organizaciju vlastitih kataloga	17
7.1. LibraryThing	17
7.2. Libib	21
7.3. CLZ Books.....	26
7.4. Book Collector	30
7.5. Calibre	32
7.6. Evaluacija knjižničnih aplikacija	34
8. Izrada, dizajn i implementacija privatne knjižnice	38
8.1. Baze podataka	38
8.1.1. SQL	40
8.2. HTML sintaksa.....	48
8.3. CSS sintaksa.....	53
8.4. PHP.....	61
8.4.1. phpMyAdmin	62

8.4.2. Spajanje i veza.....	63
8.4.3. Navigacija.....	65
8.4.4. Početna stranica.....	67
8.4.5. Katalog	69
8.4.6. Unos nove knjige.....	77
8.4.7. Rezultati pretraživanja autora.....	85
8.4.8. Prikaz rezultata pretraživanja knjige	88
8.4.9. Ispravak podataka.....	92
9. Zaključak.....	103
10. Literatura.....	104
11. Popis priloga	111
Dizajn, razvoj i implementacija baze podataka za potrebe privatne knjižnice	115
Design, development and implementation of a database for the needs of a private library	116

1. Uvod

Baze podataka, pogotovo u današnjem digitalnom dobu, čine temelj te su neizostavni dijelovi različitih područja, kao što su bankarstvo, ekonomija te medicina. Razvojem informacijske i komunikacijske tehnologije, knjižnice su također razvile te implementirale baze podataka za upravljanje knjižničnom građom. Privatne knjižnice također su zahvaćene modernizacijom kojom se korisnicima značajno pojednostavljaju procesi dohvaćanja, organiziranja, pretraživanja i korištenja vlastite zbirke. Dostupan je širok spektar aplikacija za upravljanje vlastitom zbirkom knjiga, koje nude različite automatizirane procese pomoću kojih korisnici mogu u što kraćem vremenskom periodu organizirati i pregledavati vlastite zbirke, pogotovo ako zbirka sadrži velik broj građe. Većina mobilnih aplikacija nudi mogućnost skeniranja ISBN crtičnog koda te iz baze podataka dohvaća i automatski prikazuje podatke o knjizi u kratkom vremenu. Aplikacije za organiziranje privatne knjižnice, osim oblikovanja vlastitog kataloga, također nude mogućnosti ocjenjivanja knjige, ostavljanja recenzija te dijeljenja vlastite zbirke s ostalim korisnicima ili pak pohranjivanja u „oblaku“.

U ovom diplomskom radu opisan je proces dizajna, izrade i implementacije baze podataka primjenom različitih alata, počevši od SQL-a, jezika pomoću kojeg se definiraju svojstva baze podataka, pa do HTML-a, CSS-a i PHP-a, koji služe za prikaz sadržaja te izvršavanje zadataka koje korisnik definira. Rad je podijeljen na teorijski i praktični dio. Teorijski dio obuhvaća definiranje privatne knjižnice te njezinih značajki od kojih se razlikuje od javne knjižnice, te opisivanje načina na koji je informacijska i komunikacijska tehnologija automatizirala procese u knjižnicama. U četvrtom poglavlju objašnjene su razlike između mobilnih, računalnih te mrežnih aplikacija te njihove prednosti i nedostaci. Peto poglavlje nudi pregled računalstva u „oblaku“ te njegovu sve veću primjenu u knjižničnim sustavima. Šesto poglavlje objašnjava principe dizajna aplikacija te nudi načela kvalitetnog dizajna. Nakon šestog poglavlja, opisano je i evaluirano pet knjižničnih aplikacija za organizaciju vlastite knjižnične zbirke: LibraryThing, Libib, CLZ Books, Book Collector i Calibre. Evaluacijom svojstava i funkcionalnosti tih aplikacija ponuđen je uvid u značajke koje se mogu primijeniti prilikom dizajna vlastite privatne knjižnice. Praktični dio opisuje proces izrade aplikacije za upravljanje privatnom knjižnicom, obuhvaćajući obradu i prikaz njenog sadržaja. Prvo je izgrađena SQL

baza podataka, u kojoj su definirane tablice, vrste podataka te odnosi između njih. Pomoću HTML, CSS i PHP sintakse omogućen je razvoj, prikaz te funkcija privatne knjižnice. Projekt izrade aplikacije privatne knjižnice omogućuje pregledavanje kataloga knjiga kojeg je korisnik pohranio te njegovo pretraživanje prema definiranim parametrima, kao što su pretraživanje i prikaz knjiga prema autoru, naslovu, izdavaču ili UDK oznaci te unos novih knjiga prema informacijama koje knjiga sadrži. U projektnom radu također je prikazana opcija pomoću koje se mogu izmijeniti podaci o knjigama koje su pohranjene u bazu podataka.

2. Privatna knjižnica

Privatna knjižnica, prema definiciji New World Encyclopedia predstavlja: „knjižnicu u privatnom vlasništvu, za razliku od javne ustanove, te se obično osniva samo za korištenje ograničenog broja ljudi ili čak jedne osobe“. Tadić (1994) navodi kako privatna knjižnica, kao i javna knjižnica, obuhvaća zbirku knjiga i druge građe koja može biti uređena drugačije od načela knjižničarske struke. Knjižnice koje su u vlasništvu privatnih osoba ne mogu se općenito tretirati jer se razlikuju po opsegu kojeg pojedinci mogu prikupljati (Britannica, n.d.). Privatna knjižnica, za razliku od javnih knjižnica i knjižnica u sastavu, implicira da se njezina zbirka primjenjuje za osobnu uporabu (Jagersma et al., 2023). Građa unutar privatnih knjižnice odražava interese njenog vlasnika, a sadržaj zbirke u određenom području može premašiti broj sadržaja u narodnim knjižnicama (New World Encyclopedia, n.d.). Jagersma i ostali (2023) navode kako znanstvenici pridaju važnost proučavanju privatnih knjižnica te dokumentaciji i građi koja je u njima pohranjena kako bi se proširilo razumijevanje intelektualne i kulturne povijesti. Za razliku od javnih knjižnica, privatne knjižnice omogućuju korisnicima uključiti neobjavljenu građu, kao što su bilješke, spisi i fotografije u vlastitu zbirku (New World Encyclopedia, n.d.). Cilj svakog kataloga knjižnice, tako i privatne, je pojednostaviti pronalaženje knjiga (Jagersma et al., 2023). Privatne knjižnice izgradnjom svojih zbirki uvelike pridonose ukupnom bibliografskom znanju, pogotovo ako svoje zbirke nude na raspolaganje znanstvenicima (Britannica, n.d.). Danas mnoge korporacije imaju vlastitu privatnu biblioteku koja pokriva različite djelatnosti određene kompanije čijim zaposlenicima nudi pristup (New World Encyclopedia, n.d.).

3. Informacijska i komunikacijska tehnologija upravljanja knjižnicom

Informacijska i komunikacijska tehnologija (engl. *Information & Communication technology*), skraćeno ICT, odnosi se na uporabu računala i ostalih tehnologija u knjižnicama koje omogućuju nabavu, pohranu, organizaciju te širenje informacija (Adebayo et al., 2018). Informacijska i komunikacijska je tehnologija omogućila globalizaciju cijelog svijeta upravljanjem i distribucijom informacija (Onuoha i Obialor, 2015). Knjižnice su se razvile u informacijske centre gdje su formalni alati i tehnike zamijenjene modernim tehnologijama, a baze podataka su zamijenile velike količine inventara (Onuoha i Obialor, 2015). Olakšavanjem suradnje, pripremanjem novih programa te uspostavljanjem komunikacije sa zajednicom osigurava se da knjižnica ostane relevantna i dinamična institucija u digitalnom dobu (Ashikuzzaman, 2016). Integracija informacijske i komunikacijske tehnologije promijenila je i unaprijedila različite aspekte svakodnevnog života te u tom procesu knjižnice nisu bile iznimka, pretvarajući ih u dinamična i inovativna središta informacija i znanja (Ashikuzzaman, 2016).

Choy (2010) navodi uloge i funkcije knjižnice koje su se promijenile modernizacijom:

- **Prostor.** Knjižnični prostor kojega su popunjavale rastuće zbirke tiskane građe danas zamjenjuje fokus na konfiguriranje knjižničnog prostora za korisnike uključivanjem različitih oblika i načina pristupa knjižnici izvan njezinih prostorija.
- **Pristup.** Knjižnice korisnicima pružaju informacije putem digitalnih medija te se samim time manje oslanjaju na fizički pristup.
- **Korištenje.** Pomoć korisnicima u pronalaženju i korištenju relevantnih informacija aktivnost je koja se iz tradicionalnog knjižničarskog pristupa razvila u digitalno okruženje gdje novi izazovi predstavljaju odabir, autentifikaciju te korištenje pouzdanih informacija.

Svakodnevnim razvojem informacijske i komunikacijske tehnologije te tehnologije masovne pohrane knjižnice primjenjuju i preoblikuju načine pristupa, dohvaćanja, pohrane te manipulacije usluga s dodatnom vrijednošću i pristupa sa širokim izborom digitalnih informacija koje pružaju korisnicima (Onuoha i Obialor, 2015). Gotovo sve osnovne funkcije knjižnica zahvaćene su tehnološkim inovacijama koje su imale izravan utjecaj na knjižnice (Rubin i Rubin,

2020). Pomoću informacijske i komunikacijske tehnologije moguće je ostvariti udaljeni pristup bazama podataka te istraživačkim portalima, što pojednostavljuje interdisciplinarnu suradnju i istraživanje (Ashikuzzaman, 2016). Integracija informacijskih i komunikacijskih tehnologija u knjižnicama revolucionirala je pružanje pristupa izvorima na način da su internetske baze postale sastavni dijelovi knjižnica, nudeći korisnicima raznoliku zbirku materijala (Ashikuzzaman, 2016).

Utjecaj informacijske i komunikacijske tehnologije nadilazi automatizaciju knjižničnih procesa te je preoblikovala knjižnice u dinamične i tehnološki napredne centre znanja koje poboljšavaju pristup informacijama te transformiraju korisničko iskustvo (Ashikuzzaman, 2016). Razvojem interneta kao najvećeg skladišta informacija i znanja potaknulo je stvaranje novih alata za širenje informacija, koje je rezultiralo razvojem novih i inovatnijih knjižničnih usluga (Onuoha i Obialor, 2015). Razvojem i prilagođavanjem novih informacijskih tehnologija, uloga knjižničnih stručnjaka se transformirala te knjižničari upravljaju digitalnim resursima, pružaju personaliziranu pomoć u pretraživanju te ocjenjuju relevantnost informacija na internetu (Ashikuzzaman, 2016). Knjižnice su se transformirale u kreator digitalnog sadržaja, nudeći multimedijske resurse, audiovizualne materijale i interaktivne platforme koje poboljšavaju iskustvo učenja (Ashikuzzaman, 2016). Razvojem informacijskih tehnologija digitalni izvori, baze podataka i e-knjige postali su integrirani i sastavni dijelovi knjižničnih zbirki pružajući trenutni pristup informacijama (Ashikuzzaman, 2016).

Ashikuzzaman (2016) navodi kako je informacijska i komunikacijska tehnologija pojednostavila te poboljšala učinkovitost i produktivnost knjižničnog poslovanja i usluga automatizacijom sustava upravljanja knjižnicom. Alati za analitiku prikupljaju podatke o preferencijama korisnika proučavajući resurse i trendove te omogućuju donošenje odluka o razvoju zbirke te pružanju usluga. Na taj način korisnicima se nude personalizirane preporuke na temelju njihovih preferencija, otkrivanje novih tematsko sličnih izvora te poticanje cjeloživotnog učenja (Ashikuzzaman, 2016).

Choy (2010) navodi kako se glavna funkcija knjižnica može sažeti u isporuku relevantnih i odgovarajućih informacija korisniku te se ostale funkcije mogu promatrati kao podrška, poboljšanje te maksimiziranje vrijednosti ove funkcije. Važno je naglasiti kako informacijske tehnologije te mobilni uređaji neće u potpunosti zamijeniti tradicionalne knjižnične usluge te

fizičke materijale i spremnike informacija nego se te usluge međusobno nadopunjuju (Choy, 2010).

Onuoha i Obialor (2015) navode probleme s kojim se informacijske i komunikacijske tehnologije suočavaju u knjižnicama:

- troškovi implementacije informacijskih tehnologija,
- nedostatak ili nejasno definirani standardi,
- nepodržavanje novih formata,
- autorska prava,
- nedostatak obuke osoblja

4. Aplikacije

Aplikacijski program ili aplikacijski softver predstavlja računalni softverski paket koji izvršavaju određene funkcije koje je korisnik ili neka druga aplikacija definirala. Aplikacija se može izvršavati zasebno ili unutar skupine drugih aplikacija. Za izvršavanje zadataka, aplikacije koriste operacijski sustav računala te ostale prateće programe s kojima se pristupa pomoću sučelja za programiranje (Gillis, 2021). Aplikacije se mogu podijeliti na više načina, a ovo poglavlje pruža opis o mobilnim, računalnim te mrežnim aplikacijama. Naglasak je usmjeren na mobilne aplikacije zbog njihovog dinamičnog razvoja te sve veće primjene u knjižnicama.

4.1. Mobilne aplikacije

Posljednja dva desetljeća obilježio je razvoj i napredak tehnologije koja se primjenjuje u mobilnim uređajima (Choy, 2010). Mobilne aplikacije razvijene su za komunikacijske svrhe te uređaji mobilne tehnologije kombiniraju hardver, softver, operacijske sustave te umrežavanje (Nalluri i Gaddam, 2016). Uporaba mobilnih uređaja tijekom prošlog desetljeća bila je u stalnom porastu (Rubin i Rubin, 2020). Danas mobilne aplikacije primjenjuju sve složenije softvere, gdje primjenjuju usluge pohrane u „oblaku“ za prikaz multimedijских sadržaja te pojednostavljaju njihovu interakciju (Nalluri i Gaddam, 2016). U digitalnom svijetu, knjižnice moraju redovito pratiti tehnološke napretke te promjene potreba korisnika (Choy, 2010). Prepoznajući utjecaj mobilnih tehnologija u suvremenom svijetu, knjižnice su prilagodile svoje usluge nudeći mobilne aplikacije (Ashikuzzaman, 2016). Knjižnice mogu riješiti nove zahtjeve korisnika prihvaćanjem novih mogućnosti mobilnih tehnologija te na taj način promovirati te proširiti postojeće usluge (Nalluri i Gaddam, 2016).

Choy (2010) navodi važnost mobilnih aplikacija:

- Praktičnost korištenja. Prenosivost mobilnih uređaja omogućuje pristup informacija u bilo kojem trenutku.
- Sveprisutnost. Danas gotovo svaka osoba posjeduje mobilni uređaj. Neke države imaju veći broj mobitela od svoje populacije.

- Konkurentnost na tržištu. Konkurencija na tržištu osigurava dostupnost tehnologije svim korisnicima te njihovom nadogradnjom povećava se sofisticiranost izvođenja zadataka.

Zbog naglih promjena u tehnološkom, informacijskom i obrazovnom okruženju, sve je kompleksnije održavati i prilagođavati usluge knjižničnih aplikacija (Choy, 2010). Usvajanjem mobilnih tehnologija mijenjaju se tradicionalni odnosi između knjižnica i njihovih korisnika te uvodi nova razina interakcije (Nalluri i Gaddam, 2016).

Nalluri i Gaddam (2016) navode kako su preduvjeti za implementaciju mobilne knjižničarske tehnologije:

- isporuka elektroničkih informacijskih usluga,
- informacijski proizvodi dizajnirani za e-platformu,
- digitalizirana baza podataka,
- projektiranje elektroničkih pristupnih sustava,
- odgovarajući hardver i softver,
- određivanje vrsta usluga koje će se pružiti na mobilnim uređajima.

Choy (2010) navodi kako knjižnične usluge koje inkorporiraju mobilne uređaje kao ključan element mogu se sistematizirati u tri šire kategorije:

- Pristup dokumentima
 - Funkcija pristupa. Pristup tiskanim i digitalnim dokumentima unaprijeđeno je implementacijom mobilne tehnologije. Mobilna tehnologija omogućuje pristup digitalnim dokumentima bez obzira na lokaciju i vrijeme.
 - Podrška učenju. Prilagođena podrška učenju, bez obzira na okolnosti, sve se više oslanja na mobilne uređaje za raznovrsne pristupe te korištenje resursa u trendu porasta samousmjerenog i cjeloživotnog učenja.
 - Prilagođavanje dokumenata za čitanje na mobilnim uređajima. Dokumenti koji su dostavljeni te se prikazuju putem mobilnih uređaja prilagođeni su posebom formatiranju zbog ograničenja veličine ekrana. Tehnologija je u kratkom vremenskom razdoblju napredovala te su aplikacijski softvereri u mogućnosti preoblikovati stranice neovisno o broju stupaca, transformirajući tekst u jednostavno čitljiv oblik.

- Transakcija

Primjena mobilne tehnologije u knjižničnim transakcijama koristi se prilikom rezervacije i pristupa sadržajima knjižnice, provjere dostupnosti građe te podnošenja zahtjeva za novom knjižničnom građom. Ove aktivnosti, iako se mogu izvršiti putem knjižničnih mrežnih stranica, unaprjeđuju svoju učinkovitost mobilnom implementacijom jer korisnici mogu u bilo kojem trenutku izvršiti navedene aktivnosti te se samim time uspostavlja novi sloj praktičnosti i brzine u knjižničnim procesima, prilagođavajući se suvremenim mobilnim tehnologijama. Promatrajući potrebe korisnika, mobilne aplikacije razvile su različite značajke mobilnih uređaja, uključujući kameru, sposobnost prepoznavanja lokacije, kombiniranje audio, video i multimedijских sadržaja te integrirale druge povezane tehnologije, poput očitavanja QR i crtičnog kôda. Pomoću tih kôdova, korisnik može pregledati dodatne informacije o knjizi, komentare drugih čitatelja te preporuke sličnih naslova i sadržaja.

- Usluge

Referentne usluge koje se primjenjuju u mobilnim tehnologijama, osim odgovaranja na korisničke potrebe, omogućuju efikasnije korištenje stručnosti osoblja u knjižnici, kojima je moguće prosljeđivati upite. S obzirom na porast dostupnosti informacijskih izvora digitalizacijom i otvorenim pristupima, knjižnice su razvile vlastite sustave pretraživanja kako bi korisnici identificirali pouzdane i vjerodostojne izvore.

Nalluri i Gaddam (2016) navode kako knjižnice za razvoj i funkcioniranje mobilnih aplikacija trebaju:

- obrazovati osoblje,
- istražiti partnerstva,
- financirati projekte,
- uključiti mišljenja korisnika prilikom dodavanja novih značajki aplikacije.

4.2. Mrežne aplikacije

Mrežna aplikacija (engl. *web application*) predstavlja aplikacijski program pohranjen na udaljenom poslužitelju koji se isporučuje putem sučelja internetskog preglednika (TechTarget, n.d.). Mrežne stranice dizajnirane su za interakciju omogućujući korisnicima korištenje podataka između preglednika i mrežnog poslužitelja (Codecademy, n.d.). Prilikom korištenja mrežnih aplikacija potrebna je stalna internetska povezanost, pošto se aplikacijama pristupa putem preglednika (Fitzgerald, 2021).

Prilikom dizajna mrežnih aplikacija, najčešće se primjenjuje troslojni dizajn kojeg čine prezentacijski sloj (mrežni preglednik), aplikacijski sloj (poslužitelj) te sloj za pohranu, odnosno baza podataka (Codecademy, n.d.). Pomoću mrežnog preglednika obrađuju se zahtjevi koje klijent postavlja, aplikacijski poslužitelj izvršava traženi zahtjev, dok baza podataka služi za pohranu potrebnih informacija. Mrežne aplikacije imaju kraće razvojne cikluse te manje timove od ostalih aplikacija (TechTarget, n.d.).

Fitzgerald (2021) navodi kako su prednosti mrežnih aplikacija:

- Neovisnost o platformi. Mrežne aplikacije nisu dizajnirane samo za određene platforme, već se mogu izraditi za sve platforme sve dok se izvode na mrežnom pregledniku.
- Niski troškovi izrade. Kako mrežne aplikacije nisu specifične za određenu platformu, ne zahtijevaju značajnu količinu resursa za razliku od drugih vrsta aplikacija.
- Jednostavno održavanje. Budući da mrežne aplikacije ne ovise o platformi, primjenjuju se standardni jezici za programiranje, kao što su PHP i JavaScript, koji su dostupni na više platformi te kombiniraju skripte na strani poslužitelja.
- Automatska ažuriranja. Automatsko ažuriranje aplikacije omogućuje korisnicima da uvijek primjenjuju najnoviju inačicu kada otvore mrežnu aplikaciju prilikom svakog ažuriranja.

4.3. Računalne aplikacije

Rohn (2022) objašnjava kako su računalne aplikacije (engl. *desktop application*) softverski programi koji se primjenjuju i izvode na računalnim uređajima. Za razliku od mrežnih aplikacija, računalnim aplikacijama se ne može pristupiti putem preglednika, već ih je potrebno instalirati na osobno ili prijenosno računalo. Računalne aplikacije nude širok raspon funkcija te korisnicima nude fleksibilnost te povećanje funkcionalnosti (V2 Cloud, n.d.).

Rohn (2022) navodi tri razloga zašto se daje prednost računalnim aplikacijama, pogotovo u korporacijama:

- Neovisnost o mrežnoj povezanosti. Program se može i dalje izvršavati ako nestane internetska povezanost.
- Privatnost i sigurnost. Korisnicima nudi kontrolu nad aplikacijama zato što je mnogo zahtjevnije neovlašteno pristupiti datotekama koje su spremljene na računalima.
- Izvedba. Računalne aplikacije obično u kraćem vremenskom periodu obavljaju složenije zadatke od mrežnih aplikacija.

Računalne aplikacije prilagođene su za rad na različitim operacijskim sustavima, kao što su Linux, Windows ili Mac, gdje su osigurane optimalne performanse te kompatibilnost unutar određenog okruženja. Računalne aplikacije dizajnirane su te se primjenjuju za rad u izoliranom okruženju. Za razliku od mrežnih aplikacija, računalne aplikacije iskorištavaju puni potencijal lokalnog hardvera, što rezultira izvršavanjem zadataka u kratkom vremenskom periodu (V2 Cloud, n.d.). Pomoću tablice 1 prikazane su razlike između mrežnih i računalnih aplikacija.

Tablica 1. Razlike između računalnih i mrežnih aplikacija Izvor: Rohn, 2022.

Računalne aplikacije	Mrežne aplikacije
Aplikacija se pokreće lokalno	Aplikacija se pokreće u „oblaku“
Izvanmrežni pristup	Potreban internetski pristup
Sigurnost	Rizici od <i>cyber</i> napada

Manualno ažuriranje	Automatsko ažuriranje
Potrebna instalacija	Pristup putem platforma

Kao što je vidljivo iz tablice 1, kod većine desktop aplikacija korisnicima nije potrebna internetska veza kako bi pristupili sadržaju aplikacije. Također, desktop aplikacije smatraju se sigurnijim opcijama pošto su korisnički podaci pohranjeni na jednom mjestu. No, mrežne aplikacije imaju mogućnost samostalnog ažuriranja te pristup s bilo kojeg uređaja s internetskom vezom bez potrebe za instaliranjem, što nedostaje desktop aplikacijama. Desktop aplikacije se ne ažuriraju samostalno te im korisnici moraju dati pristanak te ih ručno ažurirati na najnoviju inačicu, što može ometati poslovne procese i tijek rada (Rohn, 2022).

5. Računalstvo u „oblaku“

Računalstvo u „oblaku“ (engl. *cloud computing*) ne nudi samo „host“ softversku i hardversku infrastrukturu, već nudi pristup softveru i povezanim uslugama na mreži bez instalacije softvera ili aplikacije na računalu (Bowers i Polak, 2014). Jedno od obilježja web 3.0 je upravo računalstvo u oblaku (Xu i Yang, 2014). Rafaels (2018, str. 34) navodi kako je pohrana u oblaku „model pohrane podataka gdje se digitalnim podacima pristupa i u kojemu se podaci pohranjuju, održavaju i sigurnosno kopiraju na daljinu. Korisnici dobivaju pristup podacima putem interneta ili privatne mrežne veze“.

Rafaels (2018) navodi pet osnovnih karakteristika računalstva u „oblaku“:

- Usluga na zahtjev, koja omogućuje jednostavne računalne mogućnosti, kao što je mrežna pohrana, bez potrebe za ljudskom interakcijom sa svakim pružateljem usluga.
- Pristup mreži, pruža dostupnost raznih mogućnosti putem mreže te se njima omogućuje pristup putem standardnih mehanizama i klijentskih platformi, kao što su mobiteli, tableti i računala.
- Udruživanje resursa, nudi mogućnost dodjeljivanja različitih fizičkih i virtualnih resursa prema zahtjevima korisnika.
- Brzo širenje, koje nudi dinamičko prilagođavanje funkcija sustava kako bi se što efikasnije odgovorilo na potražnje korisnika.
- Široki raspon usluga, koji obuhvaćaju automatsku kontrolu i optimizaciju korištenja resursa.

Nadalje, Rafael (2018) opisuje četiri modela implementacije računalstva u „oblaku“:

- Privatni „oblak“. Infrastruktura predviđena je za isključivu uporabu organizacija koja se sastoji od više korisnika ili poslovnih jedinica.
- „Oblak“ zajednice. Koriste ga korisnici koji imaju zajednički cilj ili interese te može biti u vlasništvu jedne osobe ili s njime može upravljati više osoba, organizacija ili treća strana.
- Javni „oblak“. Pruža otvorenu uporabu za javnost te može biti u vlasništvu kompanije, akademske ili vladine organizacije ili njihova kombinacija.

- Hibridni „oblak“. Predstavlja sustav koji kombinira dvije ili više različitih infrastruktura koje zadržavaju jedinstvene entitete te su povezani standardiziranom tehnologijom koja omogućuje prijenos podataka.

5.1. Integrirani knjižnični sustavi u „oblaku“

Swain (2014) navodi kako s obzirom na današnje kompleksno upravljanje podacima, računalstvo u „oblaku“ nudi najdjelotvorniju podršku u dugoročnom očuvanju, pružanju usluga te pristupu podacima. Bowers i Polak (2014, str. 45) navode kako je integrirani knjižnični sustav najjednostavnije razumjeti kao „zajednički sustav u kojemu svi korisnici mogu pristupiti i doprinijeti razvoju podataka“. Računalstvo u „oblaku“ nudi nova rješenja i perspektivu pohrane, pristupa, dijeljenja i korištenja resursa digitalnih knjižnica (Xu i Yang, 2014).

Xu i Yang (2014) navode kako su prednosti računalstva u „oblaku“ za digitalne i sveučilišne knjižnice:

- Pružanje sigurnosti knjižnice. Digitalne knjižnice velik su izvor informacija te mogu biti meta zlonamjernih napada te se samim time mogu izgubiti informacije na poslužitelju. Knjižničarstvo u oblaku osigurava pouzdanost pohranjenih izvora.
- Smanjuje troškove izgradnje i implementacije digitalne knjižnice, ne zahtijeva kompliciranu mrežnu konfiguraciju i instalaciju vlastitih softverskih aplikacija. Iz „oblaka“ se također mogu primjenjivati različite vrste programskih sustava i vrste baze podataka.
- Poboljšava i ubrzava dijeljenje izvora informacija. Računalstvo u „oblaku“ temelji se na integraciji i zajedničkom pristupu informacijama u standardnom obliku. Na taj način više digitalnih knjižnica može pristupiti informacijama pohranjenima u „oblaku“ bez potrebe za ažuriranjem postojeće infrastrukture jer su pohranjeni na poslužitelju, a ne na lokalnom računalu.
- Široki raspon usluga. Ovakav način ne samo da smanjuje operativne troškove, nego također nudi i fleksibilniji prostor za razmjenu informacija između digitalnih knjižnica.

6. Dizajn aplikacija

Dizajn aplikacija ključan je faktor u kojem korisnici odlučuju o aplikacijama koje će upotrebljavati (Gomez, 2023a). Interaktivnim prikazom stranice, multimedijским izložbama te implementiranjem virtualne stvarnosti, knjižnice privlače korisnike te čine je mjestom istraživanja i učenja (Ashikuzzaman, 2016). Korisnici, prilikom odabira aplikacija i mrežnih stranica, preferiraju aplikacije s jednostavnom uporabom, modernim dizajnom te brzim učitavanjem (Gomez, 2023b). Knjižnični i informacijski stručnjaci neprestano moraju nadzirati okruženje kako bi osigurali da su knjižnične usluge prilagođene različitim uređajima (Rubin i Rubin, 2020).

Gomez (2023b) navodi ključne komponente koje treba primijeniti prilikom dizajna aplikacije:

- Dosljednost. Tijekom dizajna aplikacije, definira se određena shema boja, font, veličina teksta i grafika koja se primjenjuje kroz cijelu aplikaciju te se ne preporučuje odstupanje od te odluke.
- Kompatibilnost. Aplikacija se treba prilagoditi različitim operacijskim sustavima koji ju prikazuju. Primjerice, operacijski sustavi iOS i Android uvelike se razlikuju te programeri prilikom implementacije aplikacija moraju osigurati kompatibilnost i prikaz na tim platformama. Također je važno aplikaciju prilagoditi svakom novom ažuriranju operacijskog sustava kako bi se omogućilo nesmetano korištenje.
- Relevantnost. Aplikacija treba nuditi funkcionalnosti koje prate i odgovaraju potrebama tržišta i korisnika. Jednostavnim korištenjem korisnicima treba omogućiti intuitivno rješavanje njihovih upita.

Prilikom analize uspješnih aplikacija, najvažniju ulogu imaju dvije ključne komponente: korisničko iskustvo (engl. *User Experience*), skraćeno UX i korisničko sučelje (engl. *User Interface*), odnosno skraćeno UI (Gomez, 2023a). Korisničko iskustvo (UX) odnosi se na interakciju korisnika te njihova iskustva s funkcionalnostima aplikacije. Korisničko iskustvo predstavlja ukupan dojam koji je ostavljen na korisnika (Gomez, 2023a). Korisničko sučelje (UI) obuhvaća određene elemente dizajna, kao što su tipografija, shema boja, funkcionalnost zaslona

te ostali elementi vizualnog dizajna (Gomez, 2023a). Kombiniranjem elemenata korisničkog sustava oblikuje se cjelokupno korisničko iskustvo (Gomez, 2023a).

Objedinjavanje svih odluka o dizajnu korisničkog sučelja predstavlja korisničko iskustvo. Integracija UX/UI u cjelini oblikuje konačan dizajn (Gomez, 2023a).

Gomez (2023a) nadalje navodi aspekte koji se primjenjuju za kvalitetan dizajn mobilnih i mrežnih aplikacija:

- Logičan izbor boja. Prilikom dizajniranja mobilne i mrežne aplikacije, potrebno je razmotriti sheme boja, koje ne samo da su estetski privlačne, nego i funkcionalne. Prilikom dizajniranja, potrebno je izabrati odgovarajuće kombinacije boja koje će utjecati na udobnost promatranja kod ciljanih korisnika. Neodgovarajuće kombinacije boja mogu kod korisnika izazvati problem s razumijevanjem sadržaja.
- Usklađena struktura. Dizajniranjem definira se način postavljanja elemenata korisničkog sučelja te njihova međusobna udaljenost. Dosljedan dizajn korisničkog sučelja, koji je prilagođen na svim platformama, pruža dosljedno UX/UI iskustvo.
- Jednostavna navigacija. Navigacija predstavlja jedan od najvažnijih aspekata aplikacija te predstavlja prioritet u dizajnu aplikacije. Dizajn treba biti što jednostavniji kako bi korisnici navigaciju mogli koristiti intuitivno.
- Pregledna tipografija. Kao i svi ostali elementi korisničkog sučelja, izbor fonta treba biti dosljedan kroz cijelu aplikaciju. Prilikom dizajniranja, potrebno je odabrati jednostavan i pregledan font kojeg korisnici mogu lako iščitati.

7. Primjeri knjižničnih aplikacija za organizaciju vlastitih kataloga

U ovom poglavlju prikazane su knjižnične aplikacije pomoću kojih korisnici mogu kreirati kataloge i zbirke prema vlastitim preferencijama te opisati privatnu knjižnicu. Opisane su mobilne knjižnične aplikacije LibraryThing, Libib, CLZ Books i desktop aplikacije Book Collector i Calibre te osnovne značajke koje pružaju korisnicima prilikom izrade, organiziranja i upravljanja vlastitim kolekcijama knjiga, kao što su mogućnosti pretraživanja, praćenje pročitanih knjiga i lista želja te ostale napredne funkcije.

7.1. LibraryThing

LibraryThing besplatna je knjižnična platforma koja omogućuje praćenje čitanja te upravljanja cjelokupnom vlastitom privatnom knjižnicom. Ovom virtualnom katalogu moguće je pristupiti putem Android i iOS mobilnih operacijskih sustava te preko mrežne stranice. LibraryThing nudi mogućnost uređivanja, pretraživanja i sortiranja podataka, označivanje knjiga prema vlastitim kriterijima te korištenje različitih sustava klasifikacije, uključujući klasifikaciju Kongresne knjižnice, Deweyevu decimalnu klasifikaciju te ostale sustave za sistemsko organiziranje vlastite zbirke (LibraryThing: LibraryThing A Home For Your Books, n.d.).

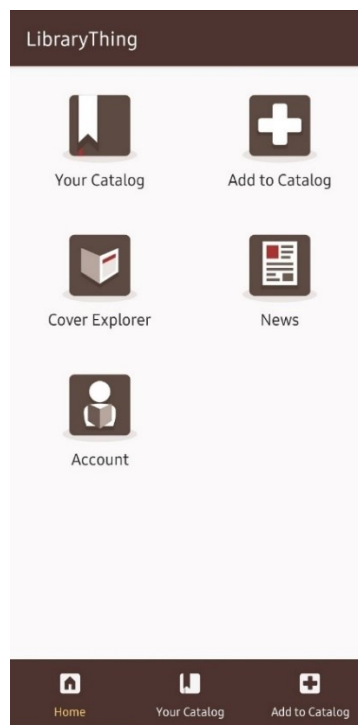
LibraryThing razvio je Tim Spalding 2005. godine. Od tad pa do 2020. godine, naplaćivala se naknada za katalogiziranje preko dvjesto knjiga, koja je iznosila 10 USD godišnje ili 25 USD za cjeloživotno članstvo. U razdoblju COVID-19 krize, LibraryThing prilagodio je svoj poslovni model te je od ožujka 2020. godine korisnicima i organizacijama omogućeno besplatno korištenje i katalogizacija neograničenog broja knjiga, reflektirajući potrebe za pristupom knjižničnim resursima te njihovom organizacijom tijekom *lockdowna* (Spalding, 2020).

LibraryThing koristi, uz Amazon, preko tisuću knjižnica za prikupljanje informacija koje pružaju otvoreni pristup svojim zbirkama s protokolom Z39.50. Na temelju pretraživanja nudi

preporuke. Na taj način LibraryThing povezuje korisnike koji dijele zajedničke čitateljske interese te predlaže slične naslove za sljedeće čitanje (LibraryThing: About LibraryThing, n.d.).

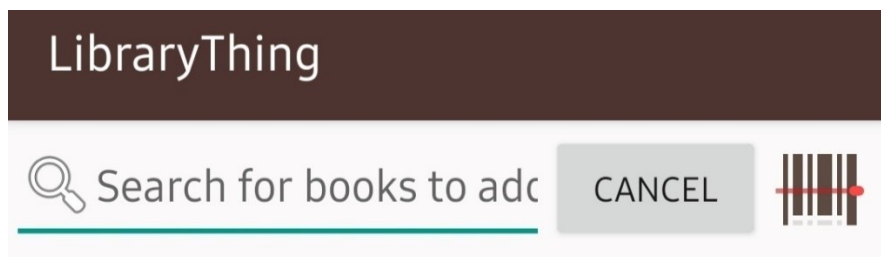
LibraryThing također je razvio i implementirao TinyCat, specijalni knjižnični katalog namijenjen malim knjižnicama. LibraryThing i TinyCat dijele istu infrastrukturu, softverske komponente te podatke. TinyCat idealno je rješenje za organizacije koje žele preoblikovati svoju knjižnicu u online katalog prilagođen korisnicima. Prilagođen je potrebama školskih knjižnica, društvenih organizacija te svih knjižnica s manje od 20 000 jedinica građe. TinyCat pruža potpuno besplatnu opciju za osobnu upotrebu, a za profesionalne organizacije postoje mjesečne pretplate (LibraryThing: TinyCat, n.d.).

Prilikom prijave na LibraryThing potrebno je unijeti korisničko ime i lozinku. Iako je opcionalno, LibraryThing preporučuje dodavanje e-maila. Slika 1 prikazuje korisničko sučelje LibraryThing aplikacije koja nudi pregled korisničkog kataloga, mogućnost dodavanja u katalog, pretraživanje i odabir naslovnica knjiga, novosti te upravljanje korisničkim računom.



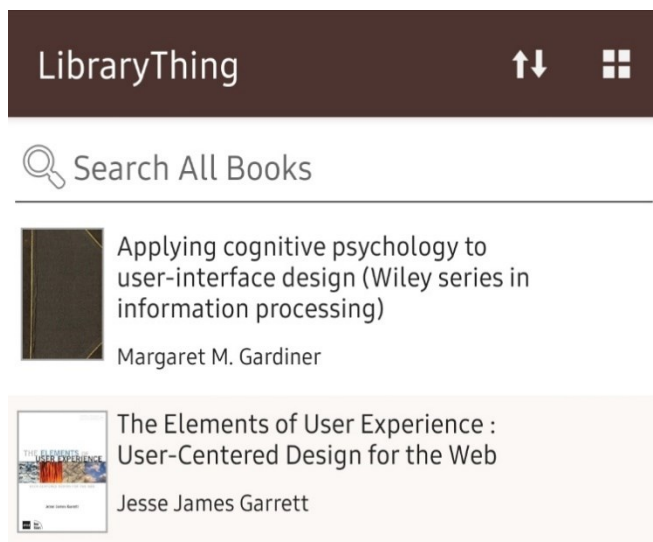
Slika 1. Korisničko sučelje LibraryThing aplikacije. Izvor: snimka vlastitog zaslona.

Opcija „*Add to Catalog*“ omogućuje dodavanje knjiga upisom u tražilicu ili skeniranja ISBN kôda na knjizi, kao što je prikazano na slici 2.

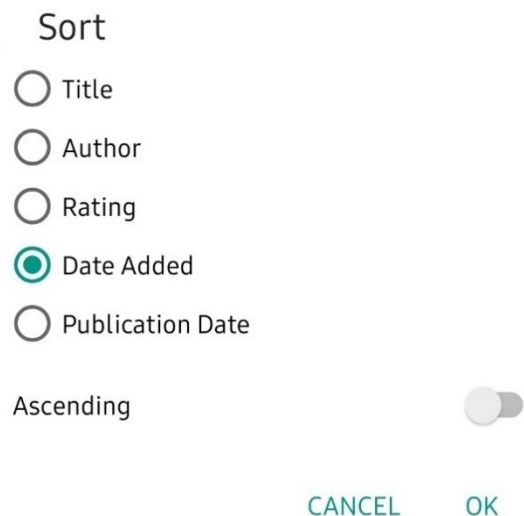


Slika 2. Mogućnosti dodavanja knjiga u aplikaciji LibraryThing. Izvor: snimka vlastitog zaslona.

U opciji „*Your Catalog*“ prikazane su knjige koje je korisnik dodao u aplikaciju te ih unutar kataloga može pretraživati te sortirati prema kriterijima, kao što su naslov, autor, rangiranje, datum dodavanja i datum izdavanja ulaznim ili silaznim redoslijedom, što je prikazano na slikama 3 i 4.

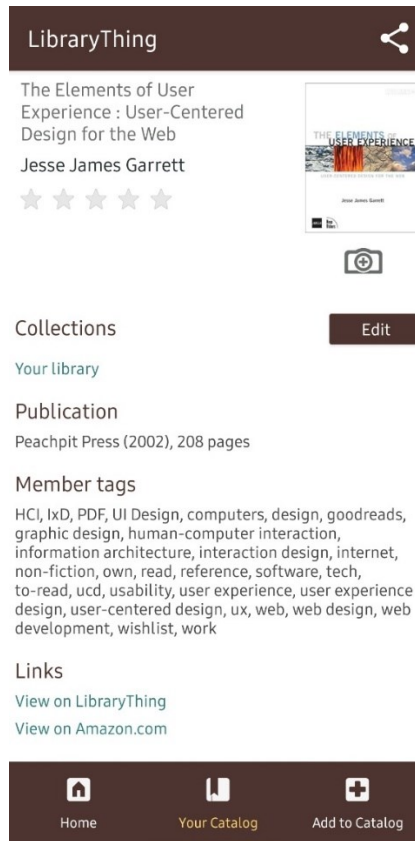


Slika 3. Prikaz kataloga LibraryThing aplikacije nakon dodavanja knjiga. Izvor: snimka vlastitog zaslona.



Slika 4. Kriteriji sortiranja u LibraryThing aplikaciji. Izvor: snimka vlastitog zaslona.

Prilikom odabira određene knjige u aplikaciji, korisnik može pregledati informacije o knjizi. Uz autora, naslov, izdavača i broj stranica, korisnik također može dati ocjenu knjizi, pregledati korisničke oznake, izmijeniti sliku naslovnice te podijeliti informacije, što je prikazano na slici 5.



Slika 5. Pregled knjige u LibraryThing aplikaciji. Izvor: snimka vlastitog zaslona.

7.2. Libib

Libib je mobilna aplikacija upravljanja knjižničnim resursima koja se primjenjuje u školama, organizacijama, knjižnicama te u privatnim kućnim katalozima. Mrežni softver nudi mogućnost stvaranja i upravljanja višestrukih zbirki, katalogiziranje knjiga, glazbe, filmova te ostalih multimedijских vrsta građe. Korisnici mogu zapisivati bilješke te dijeliti vlastite zbirke. Pomoću „*Cloud Sync-a*“ automatski se sinkroniziraju zbirke te se samim time omogućuje njihov pristup preko različitih uređaja (Libib: Cloud Cataloging, n.d.).

Libib također nudi mogućnosti:

- automatskog generiranja metapodataka,
- pružanje statistike,

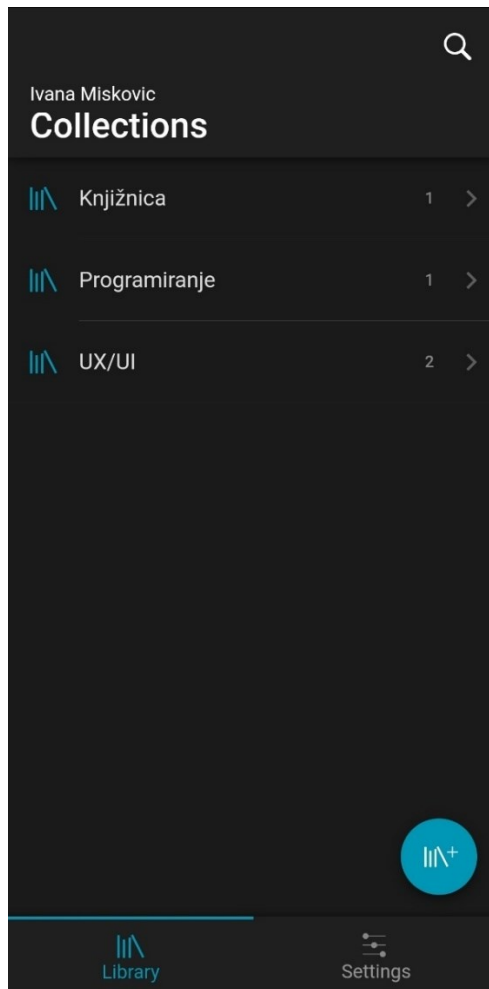
- promjene korisničkog računa,
- ostavljanja online recenzija,
- vođenja bilješki koje je naknadno moguće pretraživati,
- označivanje, grupiranje te osnovno uređivanje (Libib, n.d.).

Libib je dostupan u „Basic“ opciji, koja je besplatna za korištenje te „Pro“ opciji, koja je prilagođenija za organizaciju u školama, kompanijama te ostalim organizacijama. Glavna razlika između tih opcija je što Libib nudi mogućnost pohrane pet tisuća stavki za standardne korisničke račune te sto tisuća za „Pro“ korisničke račune (Libib: Want More? Go Pro!, n.d.).

Za razliku od „Basic“ opcije, Libib „Pro“:

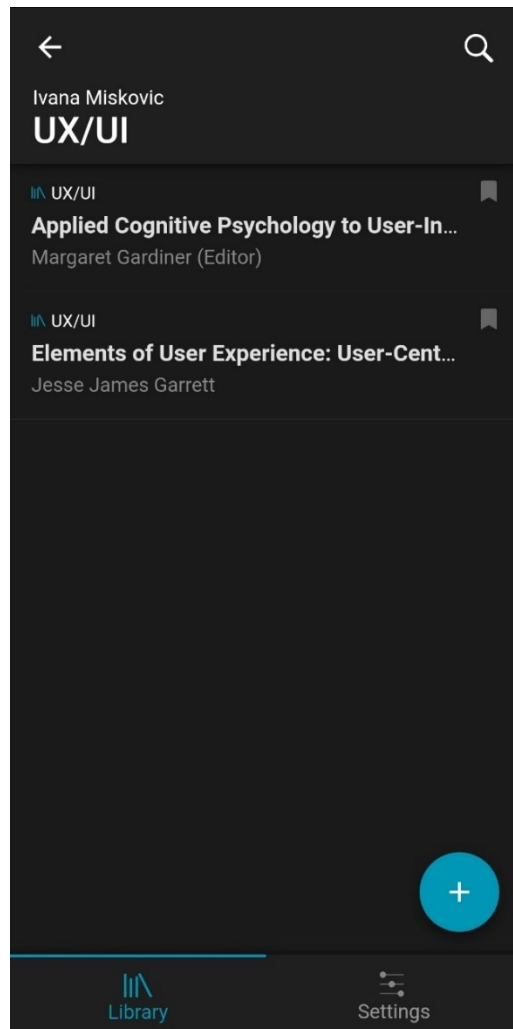
- nudi višestruku administraciju, tj. dodavanje pristupa specifičnim zbirkama na vlastitom korisničkom računu ili postavljanje drugih korisnika kao administratora s ovlastima nad zbirkama;
- nudi mogućnost odabira više stavaka unutar zbirke te ih je moguće istovremeno urediti;
- omogućuje dodavanje knjiga putem kontrolnog broja Kongresne knjižnice (LCCN);
- nudi automatsko generiranje QR kôda;
- omogućuje personalizaciju javne stranice implementiranjem logotipa, informacija te ostalih korporativnih identiteta (Libib: Want More? Go Pro!, n.d.).

Slika 6 prikazuje korisničko sučelje koje jednostavnim dizajnom prikazuje zbirku korisnika, pretraživanje zbirke te plavi gumb za dodavanje nove zbirke.



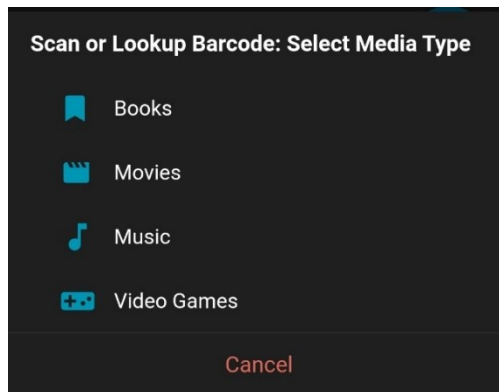
Slika 6. Korisničko sučelje aplikacije Libib. Izvor: snimka vlastitog zaslona.

Odabirom određene zbirke, aplikacija prikazuje popis knjiga te opciju „+“ koja omogućuje dodavanje novih jedinica zbirke manualno ili skeniranjem ISBN/UPC kôda, što je prikazano na slici 7.



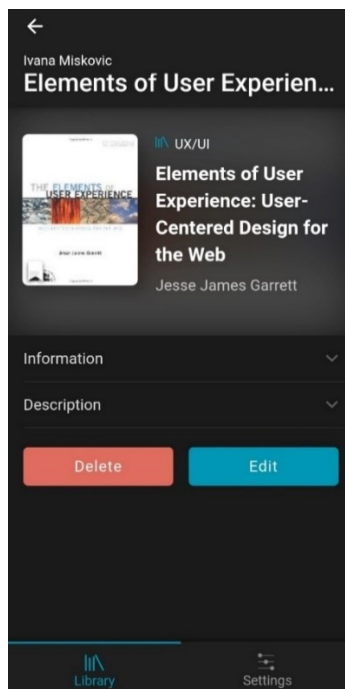
Slika 7. Prikaz kataloga aplikacije Libib. Izvor: snimka vlastitog zaslona.

Nakon odabira ISBN/UPC barkôda za unos, Libib nudi pohranu različitih medijskih formata, kao što su knjige, filmovi, glazba ili videoigre, pohranjenih zajedno ili odvojeno u zbirci, prikazano na slici 8 (Libib, n.d.).



Slika 8. Odabir medijskog formata za pohranu u katalog u aplikaciji Libib. Izvor: snimka vlastitog zaslona.

Nakon tog koraka, građa je spremljena u zbirku i spremna za pregledavanje. Slika 9 prikazuje knjigu pohranjenu u katalog gdje su dostupni naslov, autori, naslovna slika, mogućnosti uređivanja i brisanja te dodatne informacije i opis knjige, koji su prikazani na slikama 10 i 11.



Slika 9. Pregled knjige u Libib aplikaciji. Izvor: snimka vlastitog zaslona.

Information	
ISBN 13/EAN	9780735712027
ISBN 10/UPC	0735712026
Length	208
Publisher	Peachpit Press
Publish Date	2002
Copies	1

Slika 10. Informacije o knjizi u Libib aplikaciji.

Izvor: snimka vlastitog zaslona.

Description
<p>Smart organizations recognize that Web design is more than just creating clean code and sharp graphics. A site that really works fulfills your strategic objectives while meeting the needs of your users. Even the best content and the most sophisticated technology won't help you balance those goals without a cohesive, consistent user experience to support it. But creating the user experience can seem overwhelmingly complex. With so many issues involved-usability, brand identity, information architecture, interaction design-it can seem as if the only way to build a successful site is to spend a fortune on specialists who understand all the details. The Elements of User Experience cuts through the complexity of user-centered design for the Web with clear explanations and vivid illustrations that focus on ideas rather than tools or techniques. Jesse James Garrett gives readers the big picture of Web user experience development, from strategy and requirements to information architecture and visual design. This accessible introduction helps any Web development team, large or small, to create a successful user experience.</p>

Slika 11. Opis knjige unutar Libib aplikacije.

Izvor: snimka vlastitog zaslona.

7.3. CLZ Books

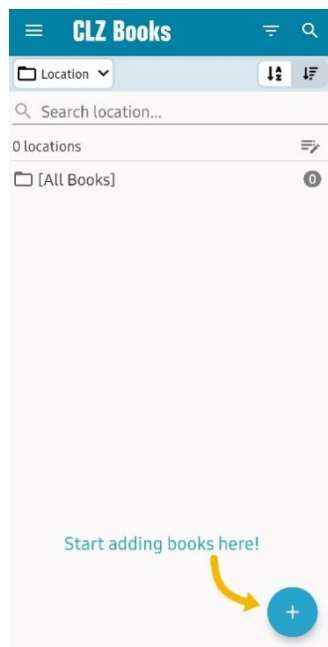
CLZ Books, kao i prethodne dvije navedene aplikacije, također nudi mogućnost upravljanja i organizacije vlastite knjižnične zbirke. CLZ Books razvila je kompanija Collectorz.com, koja je specijalizirana za izradu aplikacija za organiziranje i nadziranje kolekcionarskih aktivnosti (Collectorz.com, n.d.).

Dodatne kolekcionarske softverske aplikacije koje je ova kompanija razvila su:

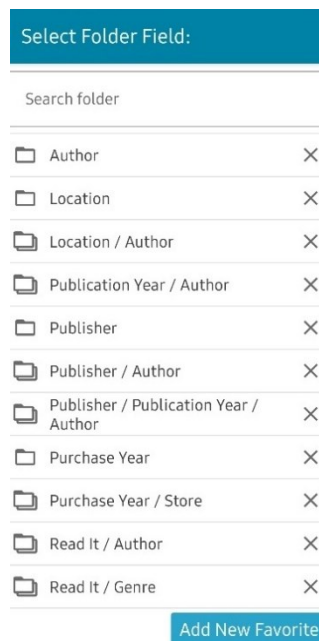
- CLZ Music, pomoću koje se organizira glazbena kolekcija,
- CLZ Movies, koja organizira filmsku kolekciju prema žanru, ocjenama filmova, glumcima, redateljima i ostalim kriterijima,

- CLZ Comics, gdje kolekcionari mogu organizirati i pratiti stripove,
- CLZ Games, gdje se prate i organiziraju informacije o videoigrama (Collector.com: Welcome to Collectorz.com, the collection database software!, n.d.).

CLZ Books dostupna je za iOS i Android operacijske sustave. Aplikacija nudi mogućnost odabira mjesečne ili godišnje pretplate. Proces dodavanja knjiga je isti kao i kod ranije navedenih aplikacija, omogućeno je skeniranje ISBN kôda knjige ili manualno unošenje informacija o knjizi, kao što su naslov i autor. Skeniranjem ISBN-a, aplikacija pronalazi i prikazuje podatke o knjizi. Moguće je urediti i prilagoditi podatke, kao što su lokacija knjige, kupovna cijena i više. Primjenjuje vlastitu „Core“ online bazu podataka knjiga koja funkcionira kao „cache“, privremeno pohranjujući podatke, čijim se povezivanjem dohvaćaju podaci iz drugih mrežnih izvora o knjigama (ISBNdb, n.d.). Slika 12 prikazuje korisničko sučelje aplikacije pomoću kojeg se dodaju nove knjige manualno ili skeniranjem ISBN kôda, a slika 13 prikazuje unaprijed definirane zbirke u kojima korisnik može organizirati svoju građu.

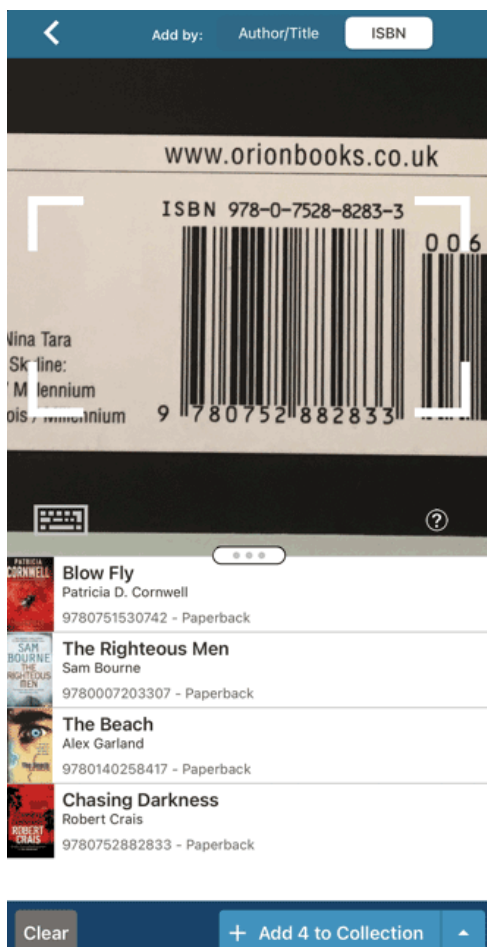


Slika 12. Korisničko sučelje CLZ Books aplikacije. Izvor: snimka vlastitog zaslona.

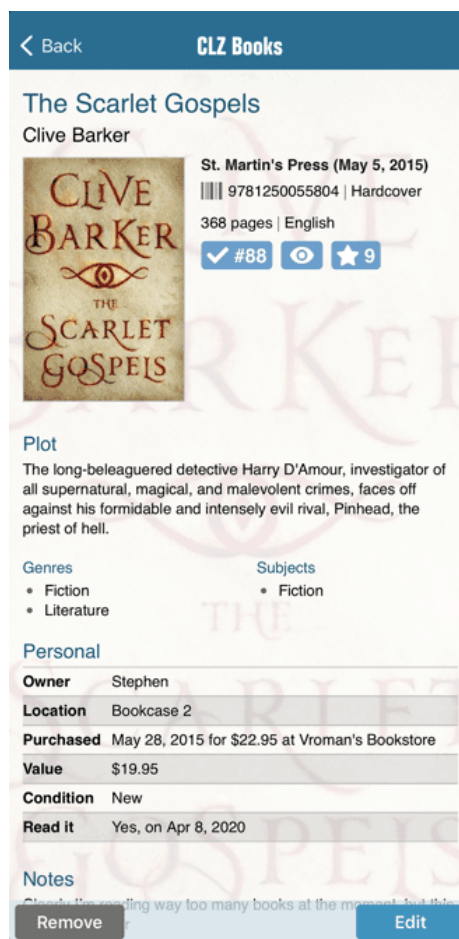


Slika 13. Zbirke unutar CLZ Books aplikacije. Izvor: snimka vlastitog zaslona

Slika 14 prikazuje postupak skeniranja i dodavanja knjiga putem ISBN kôda, dok slika 15 nudi prikaz informacija o knjizi u CLZ Books aplikaciji.



Slika 14. Dodavanje novih knjiga pomoću skeniranja ISBN kôda u aplikaciji CLZ Books. Izvor: <https://www.collectorz.com/book/clz-books> (Pristup 19.01.2024.)



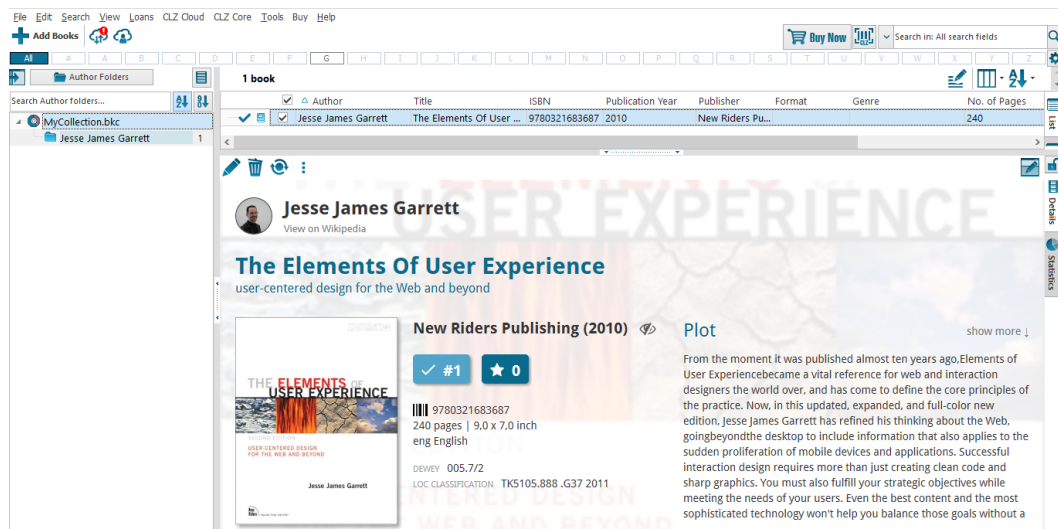
Slika 15. Detalji o knjizi unutar CLZ Books aplikacije. Izvor: <https://www.collectorz.com/book/clz-books> (Pristup 19.01.2024.)

Kompanija Collectorz.com, koja je razvila aplikaciju CLZ Books, također je razvila CLZ Cloud, koji nudi besplatnu pohranu u oblaku za mrežnu sigurnosnu kopiju baze podataka knjiga, sinkroniziranje podataka s drugim računalima i mobilnim uređajima te dijeljenje i objavljivanje vlastitog popisa knjiga. Kompanija je također razvila i desktop aplikaciju za katalogizaciju i organizaciju knjiga, Book Collector, koja omogućuje pregled i dodavanje zbirke na računalnom uređaju. Aplikacije se mogu koristiti zasebno, no ako se primjenjuju obje aplikacije, podatke o

zbirki između njih moguće je jednostavno sinkronizirati putem CLZ Cloud-a (Collectorz.com: Core, Cloud and Connect explained, n.d.).

7.4. Book Collector

Book Collector je računalni softver prilagođen za rad na Windows i macOS operacijskom sustavu. Book Collector nudi mogućnost katalogiziranja knjiga te različite mogućnosti organiziranja zbirke. Aplikacija nudi besplatan probni period i godišnju pretplatu. Pomoću Book Collector-a moguće je pretraživati, sortirati i organizirati vlastitu knjižničnu zbirku pomoću različitih prikaza. Book Collector pruža visok stupanj prilagodljivosti te omogućuje podešavanje prema vlastitim preferencijama, kao što je prikazano na slici 16 (Book Collector, n.d.).



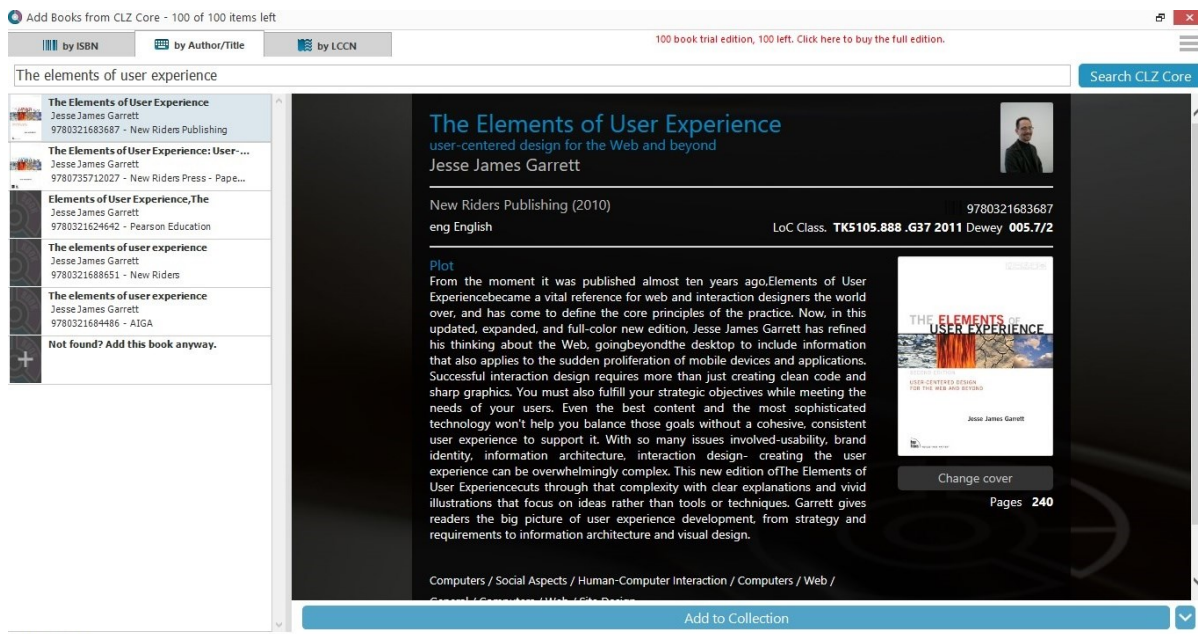
Slika 16. Korisničko sučelje Book Collector aplikacije. Izvor: snimka vlastitog zaslona.

Korisničko sučelje nudi:

- prilagodbu za različiti prikaz na glavnom zaslonu, koji uključuje mape te popise s detaljima;

- odabir opcija za prikaz popisa knjiga putem teksta, slika ili kartica prema vlastitim preferencijama;
- raznovrsne predloške za prikaz detalja o knjizi;
- odabir svijetle ili tamnije teme pozadine za prikaz na ekranu (Book Collector, n.d.).

Knjige je moguće pretraživati i dodavati putem ISBN-a te naslova ili autora, što je prikazano slikom 17.



Slika 17. Postupak dodavanja knjiga u aplikaciji Book Collector. Izvor: snimka vlastitog zaslona.

Book Collector na jednostavan način omogućuje dodavanje knjiga u vlastitu bazu podataka, također prema naslovu i autoru ili putem upisivanja ISBN-a. Baza podataka automatski pruža:

- Glavne podatke o knjizi, kao što su autor, naslov, izdavač, datum izdanja.
- Ostale informacije o knjizi, poput broja stranica, dimenzija knjige i slično.

- Sliku naslovnice, sliku autora te internetske poveznice na mrežne stranice autora (Book Collector, n.d.).

Book Collector također nudi mogućnost učitavanja vlastitih mapa i datoteka elektroničkih knjiga na računalu. Softver skenira te pronalazi datoteke elektroničkih knjiga koje nakon što se povežu u bazi podataka organizira te ih prikazuje na radnoj površini s opisima i detaljima knjige (Book Collector, n.d.).

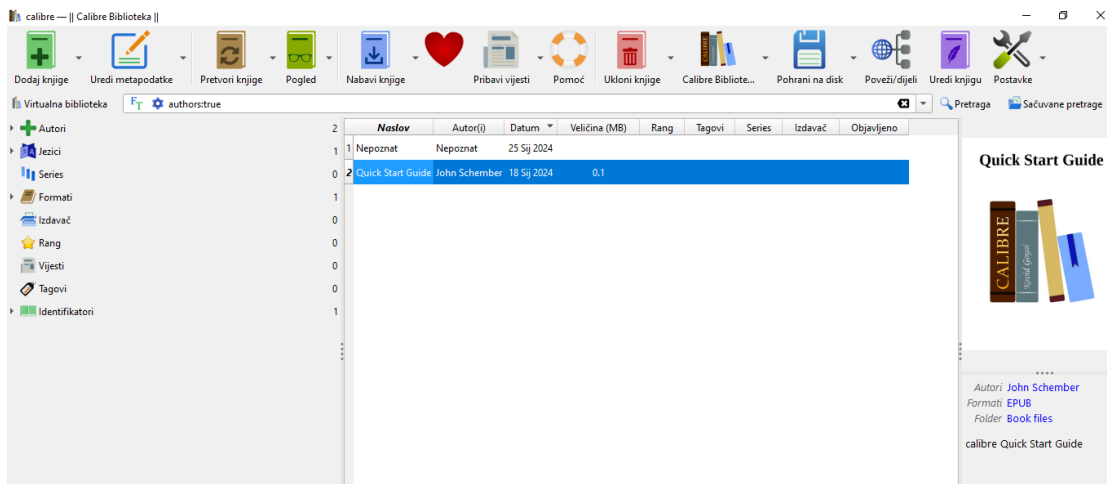
Ostale značajke i funkcije koje nudi ova desktop aplikacija su:

- Efikasna organizacija podataka pomoću zadanih postavki polja te izravnog uređivanja na popisu glavnog radnog zaslona.
- Dodavanje slika naslovnica knjiga koje nedostaju putem ugrađenog alata „*Find Image Online*“.
- Prilagođavanje baze podataka preimenovanjem postojećih polja ili kreiranjem novih definiranih polja prema vlastitim potrebama.
- Omogućuje podešavanje stupaca prema vlastitim preferencijama, pružajući fleksibilnost u organizaciji vlastitih informacija o knjigama prema željenom redoslijedu.
- Integriran sustav *Loan Manager* pruža kontrolu nad posuđivanjem i vraćanjem knjiga.
- Bazu podataka moguće je eksportirati u tekstualnu ili XML datoteku, omogućujući jednostavno dijeljenje ili izrada *backup-a* podataka (Book Collector, n.d.).

7.5. Calibre

Aplikacija Calibre, za razliku od prethodno navedenih i opisanih aplikacija koje se primjenjuju za organizaciju knjiga u fizičkom obliku, predstavlja „besplatan program otvorenog kôda za konverziju te organizaciju elektroničkih knjiga“ (Negulić, 2014). Calibre je moguće preuzeti na Windows, macOS i Linux operacijski sustav bez prethodne registracije te kao prijenosna inačica može se izravno koristiti s prijenosnog uređaja, poput USB pogona (Calibre, n.d.). Namijenjen je konverziji različitih dokumenata u formate koji su prilagođeni za uporabu na e-čitačima. Korisnici mogu, prema vlastitim preferencijama, uređivati metapodatke, umetati

naslovnice e-knjiga te kreirati tablice sadržaja. Prilikom kreiranja tablice sadržaja, generira se popis poglavlja koji pruža uvid u sadržaj e-knjige te samim time olakšava snalaženje i navigaciju (Negulić, 2014). Funkcije ove aplikacije prikazane su na slici 18.



Slika 18. Korisničko sučelje aplikacije Calibre. Izvor: snimka vlastitog zaslona.

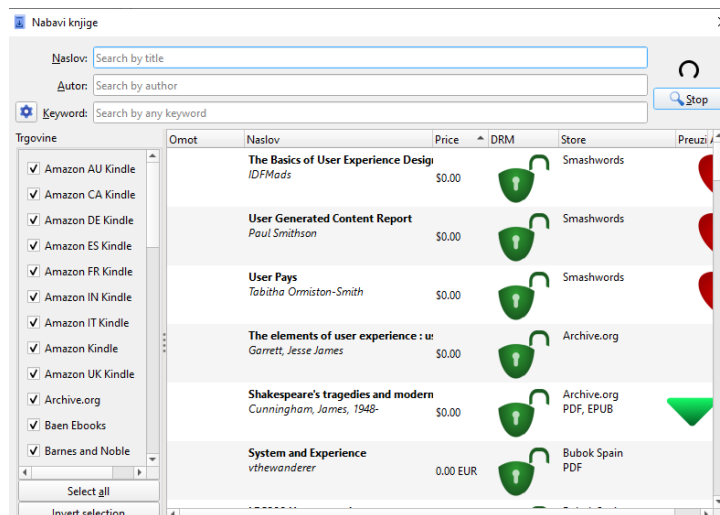
Negulić (2014) navodi smjernice kojima se dokument priprema za konverziju, inače neće biti pohranjen u originalnom sadržaju:

- Korištenje osnovnih fontova teksta.
- Formatiranje sadržaja pomoću stilova, kao što su *Normal*, *Heading 1*, *Heading 2*, *Paragraph*.
- Uklanjanje tablice sadržaja jer će se automatski generirati popis poglavlja.
- Uklanjanje zaglavlja i podnožja ako se nalaze u dokumentu.
- Korištenje slika i dijagrama u PNG formatu zato što prilikom konverzije JPG formata postoji mogućnost gubitka na kvaliteti slike.

Osim konverzije, Calibre se također primjenjuje i kao e-čitač. Navigacija ovog čitača nudi povećavanje ili smanjivanje fonta, odabir prikaza tablice, ispis e-knjige te opciju dijeljenja (Negulić, 2014).

Korisničko sučelje rezultat je neprestanog unaprjeđivanja na temelju korisničkih povratnih informacija. Nudi mogućnost preuzimanja metapodataka elektroničkih knjiga, kao što

su autor, naslov, izdavač, ISBN, omot i sažetak te stvaranje i uređivanje metapodataka u postojećim poljima ili kreirajući nova polja. Napredno pretraživanje omogućuje sortiranje rezultata prema određenim kriterijima, primjerice filtriranje rezultata tako da se prikazuju samo autori ili komentari. Calibre podržava gotovo svaki e-čitač te svakim novim ažuriranjem nudi sve veću kompatibilnost s ostalim uređajima. Aplikacija može kreirati e-knjige iz vlastitih dokumenata pomoću skripte koja generira strukturu knjige. Calibre sadrži ugrađen editor pomoću kojeg se uređuju e-knjige u poznatim formatima, kao što EPUB i Kindle, koji omogućuje jednostavno i učinkovito uređivanje. Ako Calibre ne sadrži određenu značajku koja je potrebna korisniku, može se primijeniti ugrađeni *plugin explorer* koji proširuje i implementira potrebne značajke (Calibre, n.d.). Slika 19 prikazuje mogućnost nabave knjiga za potrebe proširenja vlastite zbirke.



Slika 19. Opcija nabave knjiga unutar Calibre aplikacije. Izvor: snimka vlastitog zaslona.

7.6. Evaluacija knjižničnih aplikacija

Nakon pregleda osnovnih značajki aplikacija LibraryThing, Libib, CLZ Book, Book Collector i Calibre, aplikacije su međusobno uspoređivane na temelju zajedničkih karakteristika

kako bi se razmotrilo koje su korisničke funkcije prisutne u odnosu na ostale aplikacije. U tablici 2 prezentirani su kriteriji i funkcionalnosti koje sadrže navedene knjižnične aplikacije: kreiranje korisničkog računa, pružanje dodatnih informacija o građi, mogućnost naprednog pretraživanja, skeniranje ISBN kôda, mogućnost ocjenjivanja i recenziranja građe, dijeljenje zbirke te ugrađeni e-čitači.

Tablica 2. Pregled funkcionalnosti aplikacija LibraryThing, Libib, CLZ Books, Book Collector i Calibre.

	LibraryThing	Libib	CLZ Books	Book Collector	Calibre
Kreiranje korisničkog računa	+	+	+	+ / -	-
Dodatne informacije o građi	+	+	+	+	+
Napredno pretraživanje	+	+	+	+	+
Skeniranje ISBN kôda	+	+	+	-	-
Ocjenjivanje i recenziranje građe	+	+	+	+	-
Dijeljenje zbirke	+	+	+	+	+
Ugrađeni e-čitač	-	-	-	-	+

U tablici je uočljivo kako sve aplikacije nude dodatne informacije o građi te mogućnost naprednog pretraživanja u kojemu se mogu definirati dodatni kriteriji, poput pretraživanja po autoru, naslovu, izdavaču, godini izdavanja ili prema ISBN broju. Za korištenje aplikacija LibraryThing, Libib i CLZ Books potrebno je izraditi korisnički račun. Za registraciju kod aplikacije LibraryThing dodavanje e-mail adrese je opcionalno. Aplikaciju Book Collector moguće je koristiti bez korisničkog računa, ali za neke funkcije, poput sinkroniziranja zbirke u „oblak“ potrebno se registrirati i prijaviti na CLZ korisnički račun. Calibre aplikacija može se samostalno koristiti te ne zahtijeva kreiranje i prijavu na korisnički račun. Sve mobilne aplikacije nude skeniranje ISBN crtičnog kôda čime se značajno automatizira proces pretraživanja i dodavanja građe u vlastite zbirke. Book Collector i Calibre ne sadržavaju tu opciju pošto su desktop aplikacije, no nude mogućnost pretraživanja i dodavanja građe upisivanjem ISBN broja. LibraryThing, Libib i CLZ Books sadrže mogućnost recenziranja građe ocjenjivanjem od jedne do pet zvjezdica, dok Book Colletion nudi mogućnost ostavljanja deset zvjezdica. Calibre aplikacija ne pruža mogućnost ocjenjivanja i recenziranja. Sve aplikacije nude mogućnost dijeljenja vlastite zbirke s ostalim korisnicima. Korisnici CLZ Books i Book Collector mogu svoje zbirke učitati na zajednički CLZ Cloud, pomoću kojeg dalje mogu sinkronizirati te pristupiti podacima. Jedino Calibre aplikacija sadrži ugrađeni e-čitač, pomoću kojeg se konvertiraju različite datoteke u formate koji su prikladni za e-čitače te je aplikacija sama po sebi e-čitač u kojoj je moguće povećati ili smanjiti font te pregledavati tablicu sadržaja (Negović, 2014). Tablica 3 prikazuje koje vrste građe određena aplikacija može katalogizirati.

Tablica 3. Podržani medijski podaci za katalogiziranje u aplikacijama funkcionalnosti aplikacija LibraryThing, Libib, CLZ Books, Book Collector i Calibre.

	LibraryThing	Libib	CLZ Books	Book Collector	Calibre
Knjige	+	+	+	+	+
Filmovi	-	+	-	-	-
Glazba	-	+	-	-	-
Videoigre	-	+	-	-	-

Od svih navedenih aplikacija, jedino Libib podržava katalogiziranje knjiga, filmova, glazbe i videoigara, dok ostale katalogiziraju isključivo knjige. Libib različitu vrstu građe može pohraniti zajedno ili odvojeno u zbirci. Iako CLZ Books i Book Collector podržavaju isključivo knjige, kompanija koja je razvila te aplikacije također je i kreirala zasebne aplikacije koje korisnicima nude organizaciju isključivo filmova, glazbe, videoigara i stripova, organizirajući ih prema kriterijima koji opisuju tu građu.

Tablica 4 prikazuje na kojim operacijskim sustavima se mogu primjenjivati prethodno navedene aplikacije.

Tablica 4. Operacijski sustavi koje podržavaju aplikacije LibraryThing, Libib, CLZ Books, Book Collector i Calibre.

	LibraryThing	Libib	CLZ Books	Book Collector	Calibre
Mrežna aplikacija	+	+	+	+	-
Android	+	+	+	-	-
iOS	+	+	+	-	-
Windows	-	-	-	+	+
Mac	-	-	-	+	+
Linux	-	-	-	-	+

Aplikacijama LibraryThing, Libib i CLZ Book, iako su mobilne aplikacije, može se pristupiti i putem mrežnog preglednika. LibraryThing, Libib i CLZ Book dostupne su za Android i iOS. Korisnici također mogu pristupiti aplikacijama CLZ Books i Book Collector i putem mrežnih inačica aplikacija, koristeći vlastiti CLZ korisnički račun. Book Collector može se još koristiti na Windows i Mac operacijskim sustavima. Calibre, uz Windows i Mac, može se instalirati i koristiti na Linux-u.

8. Izrada, dizajn i implementacija privatne knjižnice

Ovo poglavlje prikazuje praktičan dio u kojemu je objašnjen postupak izrade i opisa aplikacije privatne knjižnice nakon proučavanja osnovnih svojstava aplikacija koje organiziraju i katalogiziraju građu koju korisnik posjeduje ili mu se nalazi na listi želja. Za izradu projekta koristio se SQL Server Management Studio, kako bi se dobio jasan uvid u podatke te uspostavile veze između tih podataka i tablica, zatim se primijenio HTML kako bi se izradila mrežna stranica i CSS za oblikovanje izgleda izrađene stranice te na kraju PHP za povezivanje s bazom podataka te generiranje stranice.

8.1. Baze podataka

Varga (2020) navodi kako baze podataka predstavljaju: „organizirane kolekcije međusobno povezanih podataka nekog informacijskog sustava pohranjenih na sredstvima informacijsko-komunikacijske tehnologije, kao što su poslužitelj, mreža ili „oblak““. Sustav upravljanja bazom podataka (engl. *Database Management System*), skraćeno DMS, organizira podatke te pruža okvir za brz i učinkovit pristup tim navedenim podacima (Vaswani, 2005).

Varga (2020) navodi kako su ključne komponente sustava upravljanja baze podataka:

- Funkcije za definiranje baze podataka, koje se ostvaruju naredbama za definiciju podataka jezika koji je namijenjen za rad s bazom podataka, kao što je SQL (engl. *Structured Query Language*). Definirane su sheme baze podataka koje nude logički opis baze podataka, kao što su naziv, tip, opis i odnosi među elementima podataka, te fizički opis, odnosno fizička organizacija baze podataka.
- Funkcije manipulacije baze podataka, koje su definirane naredbama jezikom za manipulaciju podataka (engl. *Data Manipulation Language*) za rad s bazom podataka.
- Upravljačke funkcije, koje se dijele na:
 - Funkcije sigurnosti baze podataka u kojima se korisnicima propisuje koje su im operacije u bazi podataka dostupne za korištenje.

- Funkcije očuvanja integriteta baze podataka, kojima se baza podataka štiti stvarajući sigurnosne kopije (engl. *backup*), a ako dođe do oštećenja primjenjuje se postupak oporavka (engl. *recovery*).
- Funkcije pomoću kojih se statistički prati rad baze podataka.

Relacijske baze podataka predstavljaju skup međusobno povezanih tablica (Shields, 2019). Sustav za upravljanje relacijskim bazama podataka (engl. *Relational Database Management System*), skraćeno RDBMS, sljedeći je korak pomoću kojega se precizno definiraju relacije, tj. odnosi između tablica u bazama podataka. Relacije omogućuju učinkovito kombiniranje podataka iz više različitih tablica u bazi podataka, pružajući mogućnosti pregleda i generiranja sadržaja te njihovo pohranjivanje, uklanjanje suvišnosti i ponavljanja, i učinkovito korištenje (Vaswani, 2005). Sustav za upravljanje relacijskim bazama podataka je poslužitelj koji pohranjuje i sprema podatke (Tatroe et al., 2015). Unutar sustava podaci su strukturirani u tablice te svaka tablica ima definirani broj stupaca s imenom i tipom podataka (Tatroe et al., 2015).

Vaswani (2005) objašnjava kako je normalizacija (engl. *normalization*) bitan postupak prilikom izrade baze podataka. Normalizacija predstavlja učinkovito organiziranje baze podataka uklanjajući vrijednosti koje se ponavljaju. Vaswani (2005, str. 131) nadalje navodi kako se „suvišne vrijednosti najčešće uklanjaju postavljanjem ponavljajućih grupa vrijednosti u zasebne tablice i njihovim povezivanjem preko vanjskih ključeva“. Tkalac (1999) opisuje normalizaciju kao metodu oblikovanja baze podataka koja je razvijena unutar relacijskog modela podataka. Također navodi kako omogućuje fleksibilnost te neometano širenje oblikovanja baze podataka. Pomoću normalizacije, baza podataka zauzima manje prostora na tvrdom disku te pojednostavljuje unošenje promjena (Vaswani, 2005). Za razliku od baze podataka koja nije normalizirana, gdje se iste vrijednosti pojavljuju u različitim tablicama te ih je potrebno manualno pronaći i zamijeniti, u normaliziranim bazama vrijednosti se pojavljuju samo jednom te je prilikom unošenja novih promjena potrebno samo provesti ažuriranje podataka (Vaswani, 2005).

Sve baze podataka sadrže metapodatke koji predstavljaju strukturu i formatiranje podataka samo po sebi, a često su opisani kao „podaci o podacima“ (Shields, 2019). Nadalje, Shields (2019, str. 23) objašnjava kako su metapodaci „podaci koji opisuju ograničenja ili specifikacije oblikovanja za druge podatke unutar baze podataka“.

8.1.1. SQL

Jedan od načina rada s relacijskom bazom podataka je primjena SQL-a (Varga, 2020). SQL, tj. *Structured Query Language*, strukturirani je jezik za upite pomoću kojeg se preuzimaju podaci (Vaswani, 2005).

Prvi korak u nastanku projektnog zadatka je izgradnja baze podataka. Za izgradnju baze podataka korišten je program Microsoft SQL Server Management Studio (MSSMS), integrirano okruženje za upravljanje SQL infrastrukturom. MSSMS omogućuje dizajn, nadzor, upravljanje te konfiguriranje podataka na računalu ili u „oblaku“. (Microsoft, 2024). Za stvaranje baze podataka primjenjuje se naredba `CREATE DATABASE` (W3Schools: SQL `CREATE DATABASE` Statement, n.d.) (Programski kôd 1).

```
CREATE DATABASE PrivatnaKnjiznica
```

Programski kôd 2. Stvaranje baze podataka „PrivatnaKnjiznica“.

U tablicama svaki zapis se identificira pomoću jedinstvenog broja, označen kao ID, te se taj jedinstveni broj naziva „primarni ključ“. U tablici je moguće samo jedno polje označiti kao primarni ključ (Vaswani, 2005). Tkalac (1999) opisuje kako se podskup atributa relacijske sheme, čije vrijednosti na jednoznačan način određuju n-torku naziva „ključ relacije“. Tkalac (1999) navodi kako primarni ključ mora ispunjavati uvjet jednoznačnosti i minimalnosti. Svaka tablica u relacijskoj bazi podataka treba sadržavati primarni ključ koji predstavlja jedinstveni identifikator (Shields, 2019). Vaswani (2005, str 128) navodi „kada je vrijednost svakog vanjskog ključa povezana s poljem u drugoj tablici, i to jedinstvenom vezom, kaže se da je sustav u stanju referencijalnog integriteta“.

Nakon definiranja baze podataka, slijedi izrada tablica te definiranje koji će se tipovi podataka nalaziti u tablici. Baza podataka `PrivatnaKnjiznica` sadrži četiri tablice:

- „Knjiga“,
- „Autor“,

- „Izdavac“ i
- „UDK“.

Tablica „Autor“ uključuje identifikator *AutorID*, koji predstavlja primarni ključ, te stupce koji uključuju informacije o autoru, kao što su *Prezime* i *Ime* (Tablica 5).

Tablica 5. Tablica „Autor“. Izvor: vlastita izrada.

Vrsta ključa	Naziv stupca	Vrsta podatka
Primaran ključ	AutorID	int
	Prezime	nvarchar(50)
	Ime	nvarchar(50)

Tablica „Izdavac“ uključuje, uz primarni ključ *IzdavacID*, općenite podatke za identificiranje izdavača, a to su naziv izdavača, mjesto izdavanja te godina izdavanja knjige (Tablica 6).

Tablica 6. Tablica „Izdavac“. Izvor: vlastita izrada.

Vrsta ključa	Naziv stupca	Vrsta podataka
Primaran ključ	IzdavacID	int
	Naziv	nvarchar(30)
	Mjesto	nvarchar(20)
	Godina	nvarchar(10)

UDK označava opću klasifikacijsku shemu koja se temelji na utvrđenim predmetnim područjima proučavanja te primjenjuje hijerarhijsku strukturu (Univerzalna Decimalna Klasifikacija, 2003). *UDKID* predstavlja identifikator ove tablice, tj. primarni ključ. UDK predstavlja oznake koje se temelje na arapskim brojevima te ostalih određenih općih simbola

(Univerzalna Decimalna Klasifikacija, 2003). *NazivSkupine* povezuje naziv skupine kojoj određena UDK oznaka pripada (Tablica 7).

Tablica 7. Tablica „UDK“. Izvor: Vlastita izrada.

Vrsta ključa	Naziv stupca	Vrsta podataka
Primaran ključ	UDKID	int
	UDK	nvarchar(20)
	NazivSkupine	nvarchar(50)

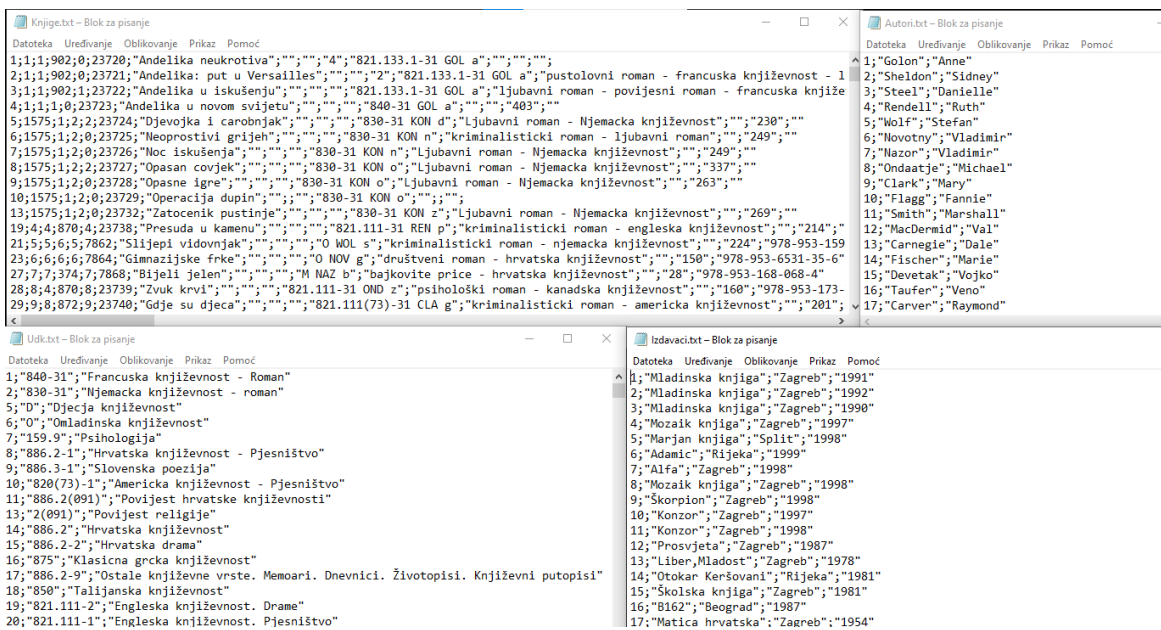
Tablica „Knjiga“ postavlja relaciju između ostalih tablica tako što ih povezuje pomoću zajedničkih polja. Ta zajednička polja koja povezuju ostale tablice, nazivaju se strani ili vanjski ključevi (Vaswani, 2005). Primarni ključ predstavlja *KnjigaID*, dok *AutorID*, *IzdavacID* te *UDKID* predstavljaju vanjske ključeve s kojima se uspostavlja odnos s ostalim tablicama. U tablici su pohranjeni osnovni podaci o knjizi, kao što su *Naslov*, *UsporedniNaslov* i *Podnaslov*. ISBN (engl. *International Standard Book Number*), tj. međunarodni standardni knjižni broj, predstavlja jedinstven identifikator knjiga te ostalih omeđenih publikacija neovisno o mediju na kojem su objavljeni (Gradska i sveučilišna knjižnica Osijek, n.d.). Ostali podaci koji dodatno opisuju knjigu su *Signatura*, *InventarniBroj*, *Svezak*, *Predmetnica*, *BrojStranica* i *Prilozi* (Tablica 8).

Tablica 8. Tablica „Knjiga“. Izvor: Vlastita izrada.

Vrsta ključa	Naziv stupca	Vrsta podataka
Primaran ključ	KnjigaID	int
Vanjski ključ	AutorID	int
Vanjski ključ	IzdavacID	int
Vanjski ključ	UDKID	int
	Naslov	nvarchar(30)

	UsporedniNaslov	nvarchar(50)
	Podnaslov	nvarchar(50)
	ISBN	nvarchar(17)
	Signatura	varchar(100)
	InventarniBroj	int
	Svezak	nvarchar(10)
	Predmetnica	nvarchar(100)
	BrojStranica	int
	Prilozi	nvarchar(100)

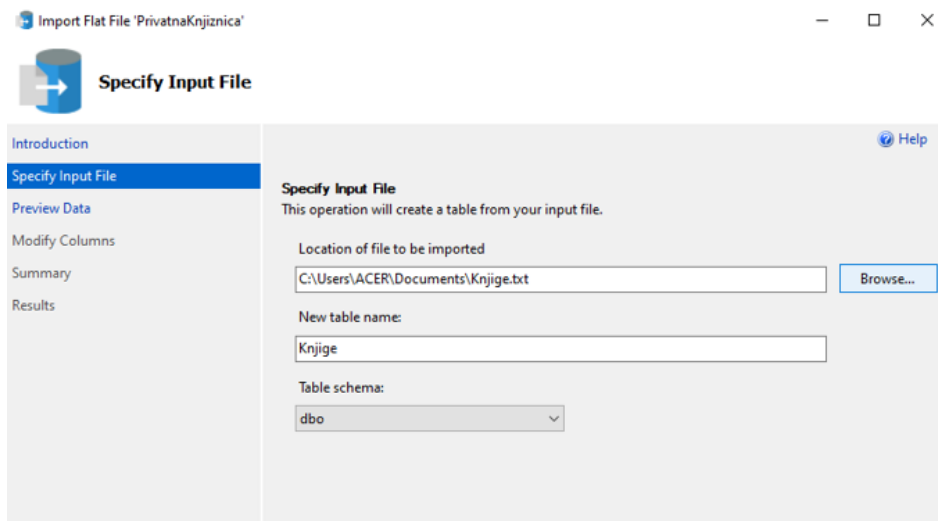
Za izradu i unos podataka u prethodno navedenim tablicama, korišteni su podaci kataloga Gradske knjižnice i čitaonice Petrinja¹.



Slika 20. Podaci za unos u tablice „Knjiga“, „Autor“, „Izdavac“ i „UDK“. Izvor: Gradska knjižnica i čitaonica Petrinja.

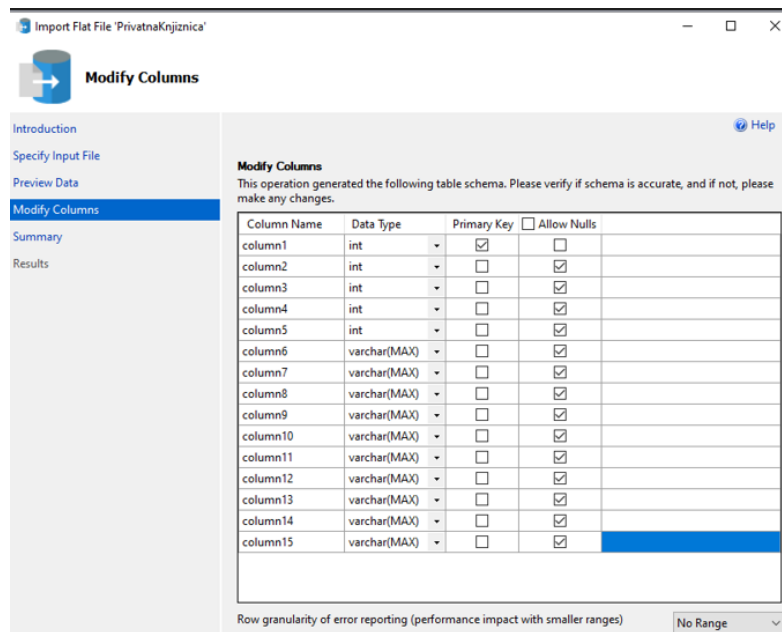
¹ Gradska knjižnica i čitaonica Petrinja, Katalog. Dostupno na: https://www.gkc-petrinja.hr/gkc_shop/katalog_products.php (Pristup 29.11.2023.).

Kako bi se stvorile tablice i podaci pohranili u bazu podataka „PrivatnaKnjiznica“, desnim klikom na tu bazu podataka odabere se opcija „Tasks“ te „Import Flat File“. Nakon toga, slikom 21 prikazan je odabir dokumenta „Knjige.txt“ koje će se pohraniti u zadanoj bazi podataka.



Slika 21. Odabir datoteke za pohranu u bazi podataka. Izvor: snimka vlastitog zaslona.

MSSMS svaki podatak koji je odvojen točka-zarezom (;) prepoznaje da pripada unutar stupca, kao što je prikazano na slici 22, te je potrebno prilagoditi imena stupaca i tip podataka s prethodno navedenom tablicom „Knjiga“, kao odabir primarnog ključa te koji stupac može sadržavati NULL vrijednost, kako bi se stvorila tablica u MSSMS-u.

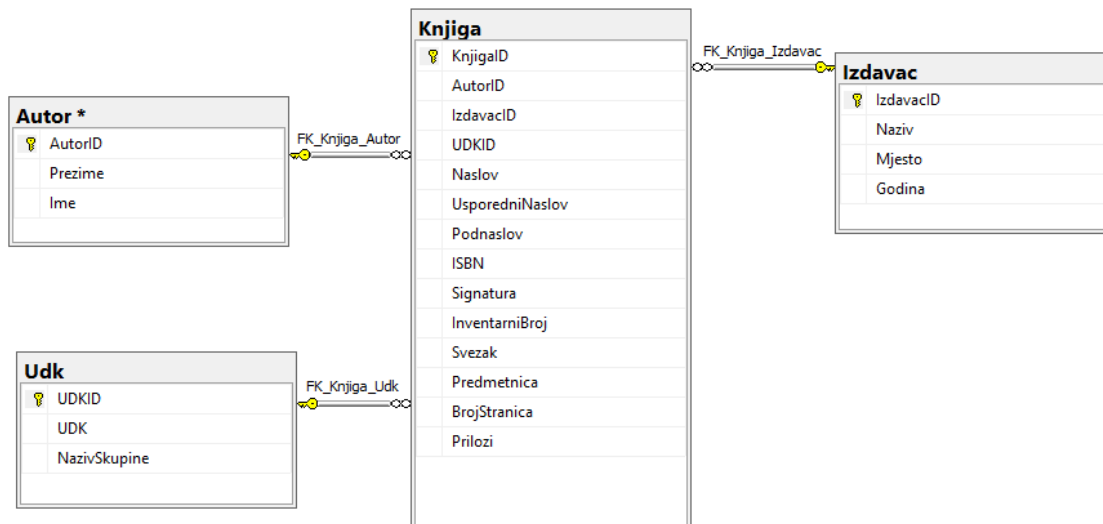


Slika 22. Prikaz učitanih stupaca, tipa podataka, primarnog ključa te NULL vrijednosti „Knjige.txt datoteke“. Izvor: snimka vlastitog zaslona.

Isti postupak je primijenjen i kod stvaranja tablica Autor, Izdavac i UDK. Shields (2019) objašnjava kako prilikom razvoja baze podataka pomoću SQL-a, za svaki se stupac mora definirati tip podataka koji mogu biti primjerice tekstualni podaci, datum i vrijeme te boolean:

- Numerički podaci su *integer*, koji predstavljaju cijele brojeve, te *decimal*.
- Tekstualni podaci mogu se konfigurirati kao znakovi fiksne ili promjenjive duljine.
- Datum može biti zabilježen u različitim formatima, kao što su YYYY-MM-DD., YYYY-MM-DD HH:MI:SS, YY-MM-DD.
- Boolean uključuje *True* ili *False* vrijednosti.

Nakon izrada tablica u MSSMS te uspostavljanja primarnih i vanjskih ključeva, dijagramom 1 prikazana je uspostavljena relacija između tablica.



Dijagram 1. Uspostavljene relacije među tablicama.

Tatroe i ostali (2015) objašnjavaju kako je sintaksa SQL-a podijeljena u dva ključna dijela:

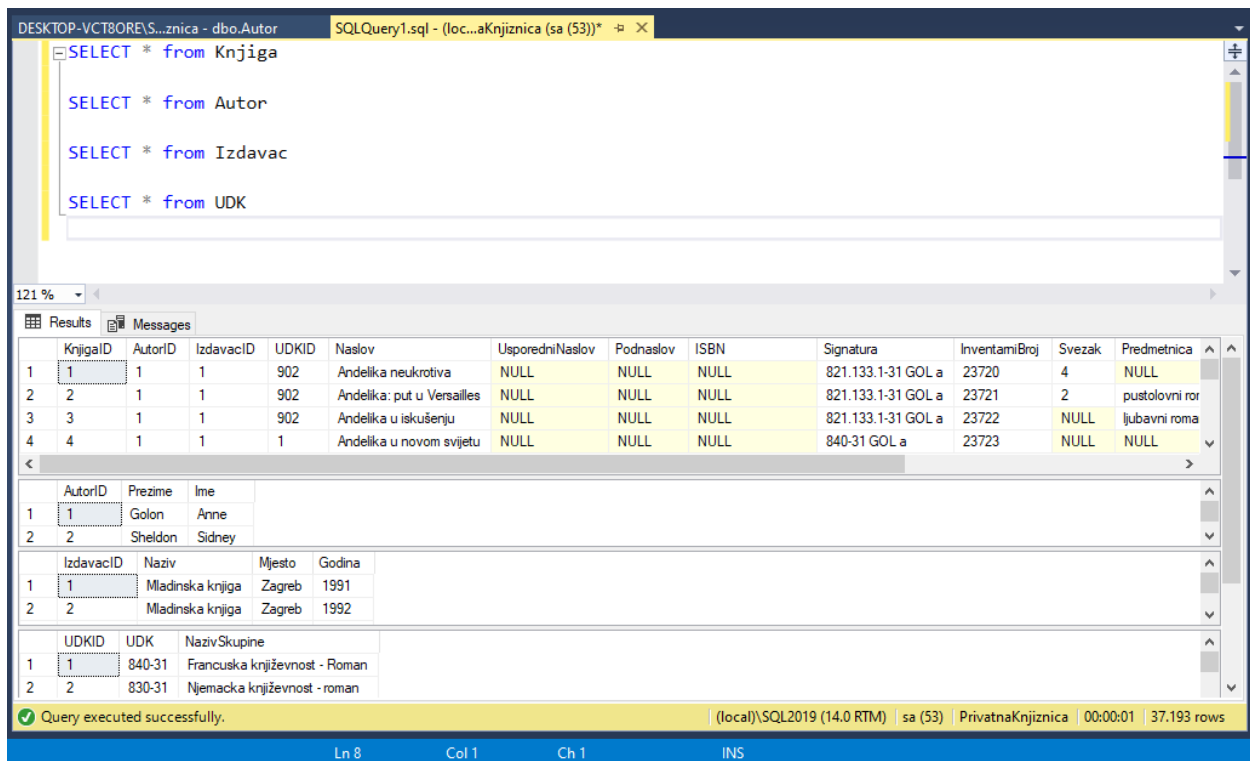
- Jezik za upravljanje podacima (engl. *Data Manipulation Language*, skraćeno DML) primjenjuje se prilikom pronalaženja i izmjene podataka u bazi podataka. Sastoji se od četiri funkcije: SELECT, INSERT, UPDATE i DELETE.
- Jezik za definiranje podataka (engl. *Data Definition Language*, skraćeno DDL) predstavlja skup SQL naredbi za stvaranje i izmjenu struktura baza podataka. Sintaksa jezika za definiranje podataka nije standardizirana na jednak način kao sintaksa jezika za upravljanje podacima. Primjerice, PHP može izvršiti SQL naredbe, no PHP primjenjuje sučelja za komunikaciju s bazom podataka umjesto izravnog podržavanja SQL-a.

Varga (2020, str. 82 - 83) navodi kako jezici za rad s relacijskom bazom podataka sadrže tri grupe naredbi:

- „Naredbe za četiri osnovne operacije nad podacima u bazi podataka:

- pronalaženje (SELECT, RETRIEVE ili GET) n-torki koje zadovoljavaju određene uvjete;
- upis (INSERT, APPEND ili STORE) novih n-torki;
- brisanje ili izbacivanje (DELETE, SUPPRESS ili ERASE) n-torki uz zadane uvjete;
- promjena (UPDATE, REPLACE ili MODIFY) podataka n-torki uz zadane uvjete;
- naredbe za definiranje relacijske sheme baze podataka, opis ograničenja u bazi podataka i definiranje pogleda na bazu podataka;
- naredbe za upravljanje bazom podataka (kao što su uspostavljanje indeksa, kontroliranje pristupa bazi i slično)*.

Kako bi se prikazali stupci u tablicama te njihovi podaci, primjenjuje se naredba SELECT *, gdje na slici 23 pruža uvid s kojim podacima i tablicama se raspolaze u bazi podataka „PrivatnaKnjiznica“.



Slika 23. Primjena naredbe `SELECT *` za prikaz tablica i njihovih podataka u bazi podataka.

Izvor: snimka vlastitog zaslona.

8.2. HTML sintaksa

HTML, tj. *HyperText Markup Language*, sadrži dvije bitne značajke: hipertekst i univerzalnost. Hipertekst predstavlja koncept koji označava stvaranje poveznica na mrežnoj stranici, čime se korisnicima nudi mogućnost navigacije i vođenja do drugih mrežnih stranica ili nekih drugih povezanih internetskih resursa. Univerzalnost podrazumijeva mogućnost pristupa HTML stranica na gotovo svim uređajima (Castro, 2007).

Informacije koje su označene oznakama, tzv. *markupom*, u HTML-u uključuju upute za oblikovanje te detalje o odnosima između različitih dijelova dokumenata (Castro, 2007).

HTML sadrži tri glavne vrste označavanja (Castro, 2007):

- elementi (engl. *elements*),

- atributi (engl. *attributes*) i
- vrijednosti (engl. *values*).

Elementi definiraju strukturu različitih dijelova mrežne stranice. Određeni elementi mogu sadržavati jedan ili više atributa koji pružaju dodatan opis i svrhu. Elementi mogu sadržavati tekst te ostale elemente definirane unutar HTML strukture. Definiraju se početnom oznakom, sadržajem, te završnom oznakom (Castro, 2007). Primjeri su prikazani programskim kôdom 2.

```
<h1>Naslov</h1>
<p>Ovo je paragraf.</p>
```

Programski kôd 3. Primjeri HTML elemenata. Izvor: W3Schools: HTML Elements.

Atributi sadrže dodatne informacije o elementima te se definiraju u početnim oznaka elementa. Vrijednosti predstavljaju informacije koje su pridružene atributima te se njima opisuju specifični parametri u HTML kôdu (W3Schools: HTML Attributes, n.d.).

HTML stranica započinje oznakom `<!DOCTYPE html>` u kojoj se definira da je dokument HTML5. Oznake `<html>` i `</html>` korijenski su elementi HTML stranice te se unutar njih nalaze svi ostali elementi (W3Schools: HTML Basic Examples, n.d.).

HTML struktura definirana je u dva glavna odjeljka, `<head></head>` i `<body></body>`. Unutar `<head></head>` elemenata definiraju se informacije o mrežnoj stranici koje su važne za internetske pretraživače (Castro, 2007). Definiranje *encoding-a* omogućuje se ispravno prikazivanje znakova na preglednicima s različitim zadanim kôdiranjem (Castro, 2007). Element `<title></title>` označava naslov HTML stranice te je prikazan na kartici mrežnog preglednika (Castro, 2007).

U projektnom zadatku izrade privatne knjižnice unutar „*head*“ elementa također je definiran Bootstrap, koji predstavlja „*front-end framework*“ pomoću kojeg je značajno pojednostavljen i ubrzan razvoj mrežnih stranica nudeći predloške dizajna za obrasce tablice, tipografiju, navigaciju te ostale elemente. Bootstrap omogućuje stvaranje mrežnih stranica koje

se automatski prilagođavaju na ekranima svih uređaja, od mobilnih uređaja do stolnih računala (W3Schools: Bootstrap Get Started, n.d.). Programskim kôdom 3 prikazano je zaglavlje početne stranice privatne knjižnice.

```
<!DOCTYPE html>
<html lang="hr">
<head>
  <meta charset="UTF-8">
  <title>Knjižnica</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <link                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/b
ootstrap.min.css">
  <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link                                rel="stylesheet"                type="text/css"
href="stilknjiznice.css">
</head>
<body>
</body>
</html>
```

Programski kôd 4. Zaglavlje početne stranice privatne knjižnice.

Unutar `<body></body>` dijela HTML dokumenta obuhvaćen je sadržaj mrežne stranice koji je korisnicima vidljiv na ekranu, uključujući tekst i grafičke medije (Castro, 2007). Naslovi u HTML strukturi definiraju se oznakama od `<h1></h1>` do `<h6></h6>`, čime h1 predstavlja najvažniji naslov, a ostali najčešće predstavljaju podnaslove (W3Schools: HTML

Headings, n.d.). Paragraf teksta prikazuje se unutar `<p></p>` oznaka, a prelazak u novi red omogućuje oznaka `
` (Castro, 2007) (Programski kôd 4).

```
<div class="main">
  <div class="knjiznica">
    <h1>Privatna knjižnica</h1>
    <h2>Sustav upravljanja privatnom knjižnicom</h2>
    <p>
      Jednostavna organizacija, pretraživanje i dodavanje
novih knjiga <br>
      te prilagođavanje čitalačkog iskustva prema
vlastitim preferencijama.
    </p>
  </div>
</div>
```

Programski kôd 5. Naslov i opis privatne knjižnice.

Oznaka `<div></div>` omogućuje primjenu stilova na određeni dio stranice grupiranjem ostalih elemenata unutar tih oznaka. Elemente je moguće identificirati u određenoj klasi primjenjivanjem oznaka `div class="ime klase"`. (Castro, 2007). Početna stranica projekta privatne knjižnice sadrži hiperlinkove koji, prilikom odabira, preusmjeravaju na nove stranice, što u ovom projektu predstavljaju stranice „Katalog“ i „Unos novih knjiga“. Sintaksa hiperlinkova prikazana je programskim kôdom 5.

```
<div class="btn">
  <a href="katalog.php">Katalog</a>
  <a href="unos.php">Unos novih knjiga</a>
</div>
```

Programski kôd 6. Hiperlinkovi unutar početne stranice privatne knjižnice.

Navigacija privatne knjižnice sastoji se od pretraživanja, odnosno polja za unos teksta, gumba za pokretanje pretraživanja, definiran elementom `button` te padajućeg izbornika, pomoću kojeg je prilikom pretraživanja moguće odabrati različite kategorije, kao što su naslov, prezime autora, izdavača ili UDK pomoću elementa `option` (Programski kôd 6).

```
<div class="navigacija">
  <div class="pretrazivanje">
    <label for="search">Pretraživanje knjižnice</label>
    <br>
    <input type="text" id="search" name="search"
placeholder="Unesite pojam za pretraživanje">
    <button type="submit"><i class="fa fa-
search"></i></button>
  </div>
  <div class="kategorija">
    <label for="category">Kategorija</label>
    <br>
    <select id="category" name="category">
      <option value="naslov">Naslov</option>
      <option value="prezime">Prezime autora</option>
      <option value="naziv">Izdavač</option>
      <option value="udk">UDK</option>
    </select>
  </div>
</div>
```

Programski kôd 7. Navigacija, pretraživanje i kategorije padajućeg izbornika početne stranice privatne knjižnice.

Na kraju `<body></body>` elementa može se definirati `<footer></footer>` koji obuhvaća osnovne informacije o stranici, kao što su kontakt te povezane stranice (W3Schools: HTML `<footer>` Tag, n.d.) (Programski kôd 7).

```
<div class="footer">
    <footer>
        <p>Sustav upravljanja privatnom knjižnicom</p>
    </footer>
</div>
```

Programski kôd 8. Footer privatne knjižnice.

8.3. CSS sintaksa

CSS (engl. *Cascading Style Sheets*) oblikuje izgled definirane HTML strukture (Castro, 2007). Opisuje i kontrolira osnovna svojstva, kao što su raspored prikazivanja elemenata, veličina elemenata, boje, izgled pozadine, veličina fonta te raspored teksta (Castro, 2007). CSS je uključen u opis HTML-a kao njegovo proširenje, te istovremeno može biti zaseban alat (Castro, 2007). U zaglavlju HTML kôda privatne knjižnice, postavljena je referenca na vanjsku CSS datoteku, *stilknjiznice.css*, pomoću koje se unutar `<link>` elementa definira prikaz stranice, objašnjena programskim kôdom 8 (W3Schools: How To Add CSS, n.d.).

```
<link                rel="stylesheet"                type="text/css"
href="stilknjiznice.css">
```

Programski kôd 9. Dodavanje CSS skripte u HTML kôd.

Sintaksa CSS-a sastoji se od dva ključna dijela: selektora i deklaracije. Selektori označavaju koji elementi će biti zahvaćeni unutar HTML sintakse, a deklaracija se sastoji od jednog ili više svojstva vrijednosti koji sadrži uputu o njihovom izvršavanju (Castro, 2007) (Slika 24).



Slika 24. Selektor i deklaracija CSS sintakse. Izvor:

https://www.W3Schoolss.com/css/css_syntax.asp (Pristup 20.11.2023.).

U ovom projektu, na početku CSS sintakse, definirana je linija kôda koja omogućuje uvoz fontova iz Google Font servisa, prikazana programskim kôdom 9, pomoću koje je definiran font koji će se prikazivati na stranicama.

```
@import  
url('https://fonts.googleapis.com/css?family=Open+Sans:300,400,6  
00,700&display=swap');
```

Programski kôd 10. Korištenje fonta pomoću unaprijed definiranih Google Font stila.

Oznaka `*` omogućuje dimenzioniranje svih HTML elemenata na stranici (W3Schools: CSS Selectors, n.d.). Funkcija `margin` označava količinu prostora između elemenata (Castro, 2007). `Padding` omogućuje stvaranje prostora oko sadržaja elemenata (W3Schools: CSS

Padding, n.d.). Pomoću `box-sizing: border-box;` elementi su predstavljeni modelom kutije, prilagođeni širinom i visinom elemenata (W3Schools: CSS Box Model, n.d.) (Programski kôd 10).

```
*{  
  margin: 0;  
  padding: 0;  
  font-family: 'Open Sans', sans-serif;  
  box-sizing: border-box;  
}
```

Programski kôd 11. CSS kôd za dimenzioniranje svih HTML elemenata na stranici.

Postavljanje pozadinske slike odvija se CSS oznakom `background` (W3Schools: CSS Backgrounds, n.d.). Pozadinska slika koja se primjenjuje u projektnom zadatku prikazana je slikom 25, a unutar CSS datoteke dodaje se sintaksom `url(knjiznica.png);`. Pomoću svojstava `width` i `height` moguće je postaviti širinu i visinu elemenata (Castro, 2007). Veličina slike prilagođava se funkcijom `background-size`, a `background-position` postavlja sliku na definiranu poziciju na stranici (W3Schools: CSS background Property, n.d.).



Slika 25. Pozadinska slika privatne knjižnice. Izvor: vlastita izrada.

Za dodatno uređivanje slike, primjenjuje se `linear-gradient()`, gdje se stvara prijelaz između dviju ili više boja duž slike (Mozilla: `linear-gradient()`, n.d.) (Programski kôd 11).

```
.main
{
    background: linear-gradient(rgba(141, 153, 174, 0.8),
    rgba(141, 153, 174, 0.5)),
    radial-gradient(ellipse at center, rgba(255, 255, 255, 0.8) 0%,
    rgba(255, 255, 255, 0) 100%),
    url(knjiznica.png) no-repeat fixed;
    background-size: cover;
    background-position: center;
    height: 100vh;
    margin-top: auto;
```

```
}
```

Programski kôd 12. CSS sintaksa za postavljanje i uređivanje pozadinske slike.

Programskim kôdom 12 prikazano je oblikovanje elemenata naslova i paragrafa unutar početne stranice privatne knjižnice. CSS omogućuje specificiranje određenih HTML elemenata nad kojim će se primjenjivati stil (W3Schools: CSS Text, n.d.). Poravnavanje teksta provodi se funkcijom `text-align`, dok `font-size` mijenja veličinu fonta teksta (Castro, 2007).

```
.knjiznica
{
    height: 94.5vh ;
    text-align: center;
    color: black;
    padding-top: 70px;
}
.knjiznica h1
{
    font-size: 60px;
}
.knjiznica h2
{
    font-size: 30px;
}
.knjiznica p
{
    font-size: 20px;
}
```

Programski kôd 13. CSS sintaksa oblikovanja naslova i paragrafa.

Sljedeći korak je oblikovanje stila hiperlinkova koji su označeni klasom „.btn“. Programskim kôdom 13 prikazano je kako su hiperlinkovi označeni ovom klasom prikazani kao blokovi uzastopno jedan pored drugoga s fiksnom širinom 150 piksela te razmacima i obrubima unutar elemenata.

```
.btn
{
    width: 100%;
    margin: 30px 0;
}
.btn a
{
    width: 150px;
    display: inline-block;
    margin: 0 10px;
    padding: 15px 0;
    color: black;
    border: .5px solid #000;
}
```

Programski kôd 14. CSS oblikovanje hiperlinkova.

Oblikovanje navigacije privatne knjižnice uključuje oblikovanje trake za pretraživanje, gumba za pretraživanje te padajućeg izbornika. CSS također omogućuje grupiranje selektora pomoću kojeg se definira isti stil elemenata (W3Schools: CSS Selectors, n.d.). Svojstvo `display: flex;` sve elemente koji se nalaze unutar toga čini fleksibilnim te pojednostavljuje upravljanje njihovim rasporedom (W3Schools: CSS Flex Items, n.d.). Svojstvo `justify-`

content pruža poravnavanje fleksibilnih elemenata (W3Schools: CSS justify-content Property, n.d.) (Programski kôd 14).

```
navigacija {
    background: linear-gradient(to right, #007c6f, #008577,
#008d80, #009688, #1a9f90, #29a899, #36b0a2);
    position: fixed;
    top: 0;
    width: 100%;
    height: 60px;
    text-align: center;
    display: flex;
    justify-content: flex-end;
    align-items: center;
}
.pretrazivanje, .kategorija {
    margin-right: 20px;
}
#search {
    width: 210px;
}
```

Programski kôd 15. CSS sintaksa za oblikovanje stila pretraživanja.

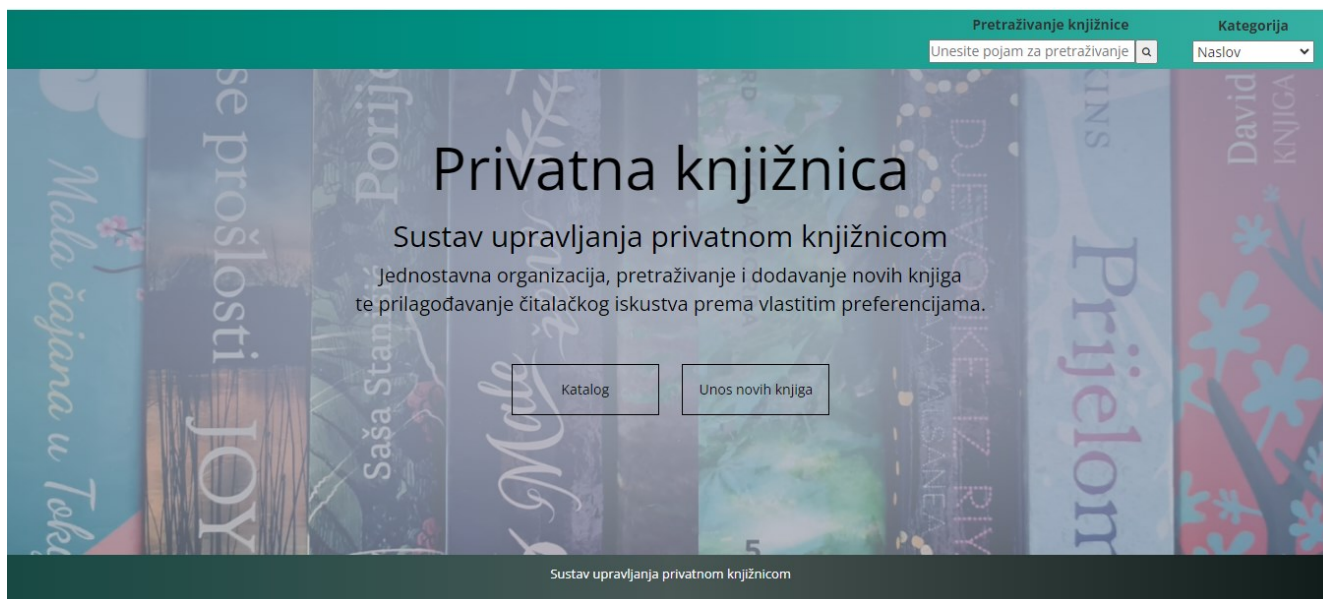
Na kraju je još potrebno i odrediti stil footer elementa, odnosno njegov raspored, veličinu, pozadinsku boju te paragrafa teksta koji se nalazi u njemu, što je prikazano programskim kôdom 15.

```
.footer
```

```
{
    background: linear-gradient(to right, #184943, #495755,
#101c1a);
    height: 60px;
    text-align: center;
    color: #fff;
    margin-top: auto;
}
.footer p
{
    text-align: center;
    padding: 10px 0;
    font-size: 13px;
    color: #fff;
}
```

Programski kôd 16. CSS sintaksa za oblikovanje footer elementa.

Nakon oblikovanja HTML i CSS sintakse, na slici 26 prikazana je početna stranica privatne knjižnice.



Slika 26. Početna stranica privatne knjižnice. Izvor: snimka vlastitog ekrana.

8.4. PHP

PHP, tj. *Hypertext Preprocessor*, korisnicima omogućuje kreiranje složenih mrežnih aplikacija za prikazivanje podataka u relativno kratkom vremenskom periodu (Vaswani, 2005). PHP nudi podršku za više od dvadeset baza podataka, uključujući besplatne komercijalne baze podataka, među kojima su i MySQL, mSQL, Oracle i Microsoft Access (Tatroe i ostali, 2015). Također podržava baze podataka koje su kompatibilne s ODBC-om (Tatroe i ostali, 2015). Prilikom otvaranja skripte u mrežnom pregledniku, mrežni poslužitelj predaje zahtjev PHP-u, gdje PHP raščlanjuje skripte i izvodi kôd između `<?php` i `?>` oznaka te ga zamjenjuje dobivenim rezultatom. Zatim se rezultat prosljeđuje mrežnom poslužitelju i klijentu te ako je kôd ispravan, preglednik može korisniku prikazati sadržaj (Vaswani, 2005).

Na PHP su snažan utjecaj imali drugi programski jezici, kao što je Perl i C (Tatroe et al., 2015). Jedna od najčešće korištenih kombinacija otvorenog kôda je kombinacija PHP-a zajedno s MySQL sustavom upravljanja bazama podataka za mrežne aplikacije temeljene na bazi podataka (Vaswani, 2005). Većina poslužiteljskih skripti usmjerena je na prikupljanje informacija iz baza podataka te njihovo pohranjivanje, pretraživanje i prikaz samih podataka (Vaswani, 2005).

Pomoću PHP programskog jezika moguće je napisati složene programe ugrađivanjem unutar HTML dokumenta za dinamičko generiranje i upravljanje mrežnim sadržajem, a mrežni poslužitelj će izvršeni kôd i rezultate vratiti pregledniku u zadanom obliku (Vaswani, 2005). PHP kôd se ne izvršava na klijentskom računalu, već na poslužitelju, a kôd se izvršava neovisno o korištenom mrežnom pregledniku te samim time korisnici ne moraju proučavati detalje specifične za te mrežne preglednike na kojima se prikazuje (Vaswani, 2005). PHP je također i interpretirani jezik, što omogućuje postepeno testiranje te uklanjanje pogrešaka koje može značajno ubrzati razvoj (Vaswani, 2005). Identifikatori u PHP-u primjenjuju se za imenovanje varijabli, funkcija i klasa te prvi znak identifikatora mora biti zabilježen ASCII velikim ili malim slovom, znakom povlake (`_`) ili bilo kojim oznakom između ASCII 0x7F i ASCII 0xFF (Tatroe et al., 2015). Nakon početnog znaka, mogu se primijeniti ti znakovi te znamenke od 0 do 9 (Tatroe et al., 2015).

PHP podržava različite tipove varijabli, kao što su logičke, cjelobrojne, decimalne varijable, objekte, polja, znakovne nizove te NULL vrijednosti, koja označava da varijabla nema vrijednost (Vaswani, 2005). Varijabla se definira simbolom \$ te imenom koji mora počinjati slovom ili znakom `_` iza kojeg dolaze slova ili brojevi (Vaswani, 2005).

Različiti tipovi i vrste podataka, objektu orijentirani mehanizam, ugrađene funkcije i podrška današnjim mrežnim tehnologijama neke su od karakteristika PHP-a (Vaswani, 2005). Jedan od važnih čimbenika programskih jezika je njegova prenosivost (engl. *portability*), tj. jednostavnost prenošenja na drugu platformu. PHP omogućuje razvoj na više platformi, kao što su Windows i macOS (Vaswani, 2005).

8.4.1. phpMyAdmin

U ovom projektu, za potrebe razvoja mrežne stranice usklađene s PHP kôdom i bazama podataka, korišten je WAMP server (WampServer, n.d.). WAMP server sadrži phpMyAdmin, gdje se izvršio „*Import*“ baze podataka, prikazano slikom 27. phpMyAdmin omogućuje korisniku HTML sučelje pomoću kojeg može izvršiti umetanje, uređivanje te brisanje podataka, izvršavanje upita, nadzor MySQL-a u stvarnom vremenu te uvoz i izvoz podataka (Vaswani,

2005). Slika 28 prikazuje uspješan uvoz podataka u phpMyAdmin kreiranu bazu podataka „PrivatnaKnjiznica“.

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: (Max: 128MiB)

Odaberi datoteku

You may also drag and drop a file on any page.

Character set of the file:

Slika 27. Import baze podataka u phpMyAdmin. Izvor: snimka vlastitog zaslona.

```
✓ 1903 rows inserted. (Query took 0.0314 seconds.)  
  
INSERT INTO `autor` (`IdAutora`, `Prezime`, `Ime`) VALUES (2095, 'Vaculik', 'Ludvik'), (2096, 'Mickiewicz', 'Adam'), (2097, 'Doblin', 'Alfred'), (2098, 'Barnes', 'Julian'), (2099, 'Davidson', 'C.'), (2100, 'Dahl', 'Robert'), (2101, 'Gardner', 'H.'), (2103, 'Hornby', 'Nick'), (2104, 'Harambašić', 'August'), (2106, 'Kovač', 'Emilija'), (2109, 'Kraus', 'Wolfgang'), (2110, 'Klain', 'Eduard'), (2112, 'Pankretić', 'Josip'), (2113, 'Skrameta', 'Antonio'), (2114, 'Weiks', 'Dudley'), (2115, 'Desnica', 'Darko'), (2116, 'Greiner i Kropilak', ''), (2117, 'Ferlinghetti', 'Lawrence'), (2118, 'Hamsun', 'Knut'), (2120, 'Lelas', 'Jasmina'), (2121, 'Mlakić', 'Josip'), (2122, 'Prodanović', 'Živko'), (2123, 'Rendić-Miočević', 'Ivo'), (2124, 'Schitzler', 'Arthur'), (2125, 'Sabolović', 'Mirko'), (2126, 'Čiča', 'Zoran'), (2128, 'Bogadi', 'Ida'), (2129, 'Smith', 'Ivo')  
[Edit]
```

```
✓ 1857 rows inserted. (Query took 0.0308 seconds.)  
  
INSERT INTO `autor` (`IdAutora`, `Prezime`, `Ime`) VALUES (4310, 'Confield', 'Jack'), (4311, 'Tirabassi', 'Becky'), (4312, 'Schaie', 'Warner'), (4313, 'Plattner', 'Gabriel'), (4314, 'Connoly', 'John'), (4315, 'Killen', 'Kari'), (4316, 'Škobić', 'Slavka'), (4317, 'Islin', 'Connie'), (4318, 'Taylor', 'Richard'), (4319, 'Robin', 'Liliane'), (4320, 'Špelić', 'Danijel'), (4322, 'Carkeet', 'David'), (4324, 'Quinn', 'Daniel'), (4325, 'Cruickhank', 'Gordon'), (4326, 'Testa', 'Caludio'), (4328, 'Littaner', 'Florence'), (4329, 'Smokvina', 'Miljenko'), (4330, 'Meyer', 'Kai'), (4331, 'Kellerman', 'Jonathan'), (4332, 'Nikolić', 'Toni'), (4334, 'Božac', 'Romano'), (4335, 'Bastašić', 'Zlatko'), (4336, 'Hudolin', 'Vladimir'), (4337, 'Marčinković', 'Juro'), (4338, 'Stroehm', 'Gustaf'), (4339, 'Davies', 'Eryl'), (4340, 'Kahn', 'James'), (4341, 'Alderton', 'Ivo')  
[Edit]
```

```
✓ 1910 rows inserted. (Query took 0.0363 seconds.)
```

Slika 28. Poruka uspješnog uvoza podataka u *phpMyAdmin* bazu podataka „PrivatnaKnjiznica“. Izvor: snimka vlastitog zaslona.

8.4.2. Spajanje i veza

Kako bi se omogućilo spajanje na poslužitelj MySQL baze podataka, potrebno je uspostaviti vezu funkcijom `mysqli_connect()` (Vaswani, 2005).

Funkcija `mysqli_connect()` sadrži četiri parametra (Vaswani, 2005):

- ime računala s MySQL poslužiteljem (`$host`),
- ime korisnika s pravom pristupa bazi podataka, (`$username`),
- lozinka (`$password`) i
- naziv baze podataka (`$db_name`).

Programskim kôdom 16 prikazano je spajanje na poslužitelj te odabranu bazu podataka. Ako je veza uspješno uspostavljena, vraća se identifikator veze koji je pohranjen u varijabli `$connection`, no ako se veza ne uspostavi, koristi die funkciju te se greška generira linijom kôda `mysqli_error($con)` (Vaswani, 2005).

```
$host = "localhost";
$username = "root";
$password = "";
$db_name = "privatnaknjiznica";

$con = mysqli_connect($host, $username, $password,
$db_name) or die(mysqli_error($con));
mysqli_set_charset($con, "utf8");
mysqli_character_set_name($con);
?>
```

Programski kôd 17. Stvaranje konekcije nad bazom podataka.

Pomoću funkcije `include()` korisnici mogu ugraditi vanjske datoteke u PHP skriptu, što predstavlja praktično i efikasno rješenje prilikom razvoja modularnih aplikacija koje su u zasebnim datotekama (Vaswani, 2005). Ugrađivanjem datoteka pomoću te naredbe omogućuje da PHP nastavi s obradom stranice čak i ako u dizajnu datoteke postoji pogreška (Tatroe et al., 2015) (Programski kôd 17).

```
<?php
    include ("navigacija.php");
?>
```

Programski kôd 18. PHP naredba za uključivanje jedne datoteke u drugu.

8.4.3. Navigacija

Programskim kôdom 18 predstavljena je navigacijska traka koja će se također funkcijom `include()` pozivati i uključivati na ostale PHP stranice, čime se značajno ubrzava proces implementacije, pošto se ne moraju ponovno ispisivati HTML i CSS elementi navigacije u svakoj datoteci.

```
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">
    <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link                                rel="stylesheet"          type="text/css"
href="stilknjiznice.css">
</head>
```

```

<body>
<form action="knjiga.php" method="GET">
<div class="navigacija">
<div class="poveznice">
    <a class="pocetna" href="index.php">Početna</a>
    <a href="katalog.php">Katalog</a>
    <a href="unos.php">Unos</a>
</div>
<div class="pretrazivanje">
    <label for="search">Pretraživanje knjižnice</label>
    <br>
    <input type="text" id="search" name="search"
placeholder="Unesite pojam za pretraživanje">
    <button type="submit"><i class="fa fa-
search"></i></button>
</div>
<div class="kategorija">
    <label for="category">Kategorija</label>
    <br>
    <select id="category" name="category">
        <option value="naslov">Naslov</option>
        <option value="prezime">Prezime autora</option>
        <option value="naziv">Izdavač</option>
        <option value="udk">UDK</option>
    </select>
</div>
</div>
</form>
</body>
</html>

```

Programski kôd 19. Struktura datoteke „navigacija.php“.

8.4.4. Početna stranica

Početna stranica privatne knjižnice prikazana je programskim kôdom 19, gdje se objedinjuju sve funkcije opisane u prethodnom poglavlju.

```
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <link                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">
  <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link                                rel="stylesheet"          type="text/css"
href="stilknjiznice.css">
</head>
<body>

<form action="knjiga.php" method="GET">
<div class="navigacija">
<div class="poveznice">
  <a class="pocetna" href="index.php">Početna</a>
  <a href="katalog.php">Katalog</a>
  <a href="unos.php">Unos</a>
</div>
```

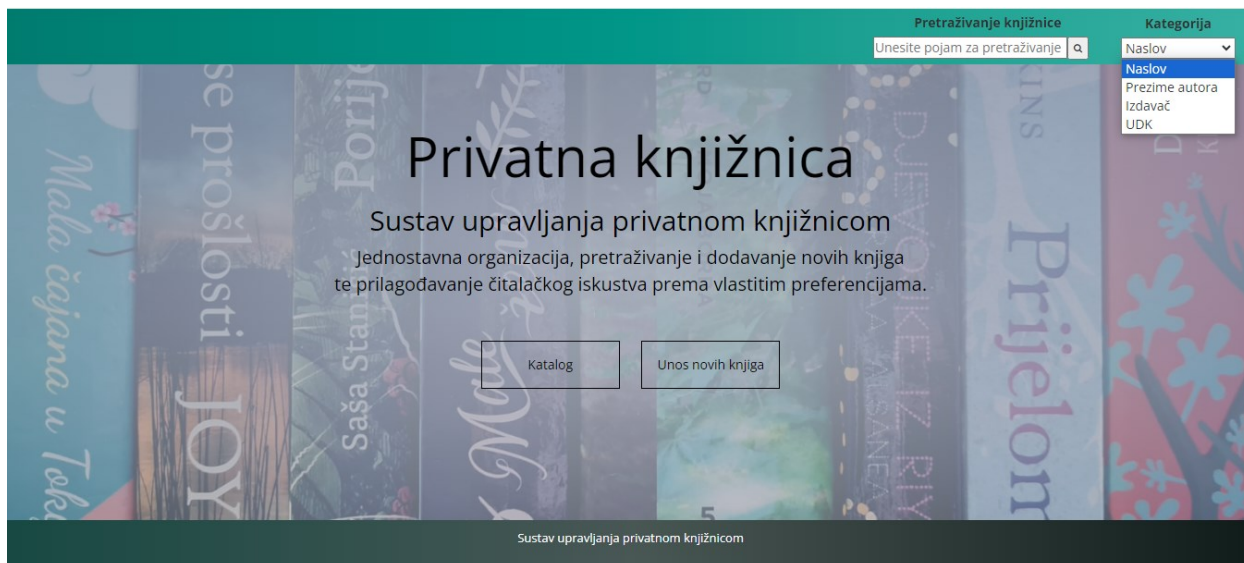
```

<div class="pretrazivanje">
    <label for="search">Pretraživanje knjižnice</label>
    <br>
    <input type="text" id="search" name="search"
placeholder="Unesite pojam za pretraživanje">
    <button type="submit"><i class="fa fa-
search"></i></button>
</div>
<div class="kategorija">
    <label for="category">Kategorija</label>
    <br>
    <select id="category" name="category">
        <option value="naslov">Naslov</option>
        <option value="prezime">Prezime autora</option>
        <option value="naziv">Izdavač</option>
        <option value="udk">UDK</option>
    </select>
</div>
</div>
</form>
</body>
</html>

```

Programski kôd 20. Struktura datoteke „index.php“.

Slika 29 pruža prikaz početne stranice privatne knjižnice na mrežnom pregledniku. Na vrhu stranice nalazi se navigacijska traka pomoću koje korisnici mogu pretraživati knjižnicu prema odabranim kategorijama dostupne u padajućem izborniku. Uz opis stranice privatne knjižnice, na središnjem dijelu stranice nalaze se linkovi pomoću kojih se može pregledati katalog korisnika ili mogućnost unosa nove knjige. Način funkcioniranja linkova opisan je u sljedećim poglavljima. Na dnu mrežne stranice prikazan je footer.



Slika 29. Početna stranica privatne knjižnice. Izvor: snimka vlastitog zaslona.

8.4.5. Katalog

Programski kôd 20 prikazuje zaglavlje PHP datoteke „katalog.php“.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Katalog</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
</head>
```

Programski kôd 21. Zaglavlje kataloga.

Navigacija stranice kataloga, za razliku od ostalih stranica, dodatno uključuje i mogućnost ispravka, koje je definirana u „header_katalog.php“ (Programski kôd 21).

```
div class="navigacija">
    <div class="poveznice">
        <a href="index.php" class="pocetna">Početna</a>
        <a href="katalog.php">Katalog</a>
        <a href="unos.php">Unos</a>
        <a href="ispravak.php">Ispravak</a>
    </div>
</div>
```

Programski kôd 22. Struktura navigacije u datoteci „header_katalog.php“.

Kako bi se prikazala navigacija na stranici, potrebno je naredbom `include` uključiti „header_katalog.php“ (Programski kôd 22).

```
<?php
    include "header_katalog.php";
?>
```

Programski kôd 23. Uključivanje navigacije u programski kôd datoteke „katalog.php“

Programski jezici, poput PHP-a, omogućuju dodavanje potprograma za izvršavanje određenih zadataka, ovisno o uvjetima, u skripte (Vaswani, 2005, str. 66). Uvjetna naredba provjerava je li uvjet koji je zadan ispunjen ili nije te će na temelju toga izvršavati različite procese (Vaswani, 2005, str. 66).

Najjednostavnija uvjetna naredba je `if ()` koja je prikazana programskim kôdom 23.

```
<php
if (uvjet)
{
    izvrsi zadatak;
}
```

Programski kôd 24. Uporaba `if ()` naredbe (Vaswani, 2005, str. 66).

Kontrolna struktura pomoću koje je moguće izvršiti i ponoviti isti skup naredbi naziva se petlja. Petlja se izvršava sve dok su naredbe točne, a kada uvjet postane netočan, petlja prestaje s izvršavanjem te se izvode naredbe koje slijede iza nje (Vaswani, 2005). Programskim kôdom 24 prikazana je struktura petlje `while ()`.

```
<?php
while (uvjet je true)
{
    ispuni zadatak;
}
?>
```

Programski kôd 25. Uporaba `while()` petlje (Vaswani, 2005, str. 73).

Definiranje elemenata stranice kataloga prikazan je programskim kôdom 25. `$_SERVER` sadržava vrijednosti poslužiteljskih varijabli, kao što su broj inačice poslužitelja, putanju do skripte koja se trenutno izvodi, IP adresu poslužitelja te korijensku mapu dokumenta (Vaswani, 2005). Polje za unos teksta za pretraživanje prikazano je kodom: `<input style="width: 50%; "type="text" id="search" name="search" class="form-control"`

placeholder="Unesite izraz">. Definiran je padajući izbornik (engl. *dropdown*) u kojemu je omogućeno pretraživanje prema kriterijima naslova knjige, prezimenu autora, nazivu izdavača, UDK kategorije ili prema svim dostupnim kategorijama.

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>"
method="GET">
<div style="margin-top: 10px; width:100%; " >
<div class="container main" style="min-height: 94vh;
width:100%; ">
<div class="form-group" style="display: flex; align-items:
center; margin-top: 70px;">
    <label style="width 9%; " for="search">Potraži
knjigu:</label>
    <input style="width: 50%; " type="text" id="search"
name="search" class="form-control" placeholder="Unesite izraz">
    <div style="margin-left: 10px;">
        <select id="category" name="category">
            <option value="naslov">Naslov</option>
            <option value="prezime">Prezime autora</option>
            <option value="naziv">Izdavač</option>
            <option value="udk">UDK</option>
            <option value="1">Sve</option>
        </select>
    </div>
</div>
<div style="width: 59%; text-align: right; ">
    <button type="submit" class="form-group" style="width:
100px;">Pretraživanje</button>
</div>
</form>
```

Programski kôd 26. Definiranje elemenata stranice kataloga.

PHP omogućuje obradu i pristup parametrima obrazaca pomoću `$_GET` i `$_POST` (Tatroe et al., 2015). Kako bi se pretraživanje kataloga izvršilo, definira se varijabla `$_GET['search']`, koja sadrži niz varijabli primljenih putem HTTP GET metode uz ključ 'search'. (W3Schools: PHP Superglobal - `$_GET`, n.d.). Varijabla `isset` vraća `true` ako je varijabla postavljena, odnosno `false` ako varijabla nije postavljena (Tatroe et al., 2015). Za vrijednosti koje se unose prilikom pretraživanja, definirane su varijable `$naslov` i `$pojam`. Nakon toga, izvodi se programski kôd definiran unutar „*if*“ uvjeta te se uključuje datoteka „konekcija.php“ za povezivanje s bazom podataka. Upravljanje MySQL upitima u PHP-u postiže se pomoću funkcije `mysqli_query()` koja prihvata znakovni niz upita i identifikator veze te ih tada šalje poslužitelju (Vaswani, 2005).

SQL omogućuje postavljanje upita na dvije ili više tablica istovremeno, te samim time prikaz kombiniranih skupova rezultata pomoću naredbe JOIN. Prikaz skupa rezultata koji sadrže samo one zapise koji zadovoljavaju upit u svim tablicama koje uključuje spajanje omogućeno je naredbom INNER JOIN (Vaswani, 2005).

Za prikaz unesenog pojma iz odabranih kategorija autora, knjige, izdavača ili UDK iz baze podataka prilikom pretraživanja primjenjuje se SQL upit prikazan programskim kôdom 26, koji na kraju izvršavanja upita sprema rezultat u varijablu `$result`. Ako je odabrana opcija „Sve“, izvršava se SQL upit definiran unutar `else()` naredbe. Linijom kôda `$count=mysqli_num_rows($result);` broji se broj redova koji odgovaraju postavljenom upitu te se spremaju u `$count` varijablu.

```
if (isset($_GET['search'])) {  
    $naslov=$_GET['category'];  
    $pojam=$_GET['search'];
```

```

include "konekcija.php";

if ($naslov != 1){

    $sql="SELECT * FROM
Izdavac INNER JOIN (UDK INNER JOIN (Autor INNER JOIN Knjiga ON
Autor.IdAutora = Knjiga.AutorID) ON UDK.idudk = Knjiga.UDKID)
ON Izdavac.IdIzdavaca = Knjiga.IzdavacID where "." $naslov ". "
like "."'%$pojam%' order by naslov asc ";
    }
else
{
    $bf=1;
    $sql="SELECT * FROM
Izdavac INNER JOIN (UDK INNER JOIN (Autor INNER JOIN Knjiga ON
Autor.IdAutora = Knjiga.AutorID) ON UDK.idudk = Knjiga.UDKID)
ON Izdavac.IdIzdavaca = Knjiga.IzdavacID ";
    }
$result = mysqli_query($con, $sql) or die(mysqli_error($con));
$count=mysqli_num_rows($result);

```

Programski kôd 27. Dohvaćanje podataka iz baze podataka prilikom postavljanja upita.

Naredba `if ($count == 0)` provjerava broj pretraga u bazi podataka, te ako je jednaka nuli, ispisuje se poruka o nepostojanju knjige u bazi podataka. Ako knjiga postoji u bazi podataka, ispisuje se tablica te se podaci iz baze podataka popunjavaju unutar definirane tablice, što je prikazano programskim kôdom 27. Kako bi se stvorio znakovni niz primjenjuje se operator učlanjivanja koji se označava točkom (.) (Vaswani, 2005).


```

if ($count==0)
{
    echo "Knjiga ne postoji u bazi podataka.";
}
else
{
    echo '<table class="katalog-tablica" border="1"
cellpadding="0" cellspacing="0" id="tablica" style="width:
100%">';
    echo '<tr style="background-color: 6db6b9e6">
<th>Naslov</th>
<th>UsporedniNaslov</th>
<th>Podnaslov</th>
<th>Autor</th>
<th>Izdavač</th>
<th>Signatura</th>
<th>Inventarni broj</th>
<th>UDK</th>
<th>Više</th>
</tr>';
while ($row=mysqli_fetch_assoc($result)){

    echo '<tr>';
    echo '<td>' . $row['Naslov'] . '</td>';
    echo '<td>' . $row['UsporedniNaslov'] . '</td>';
    echo '<td>' . $row['Podnaslov'] . '</td>';
echo '
            <td><a
                                target="_self"
href=autor.php?idautor='. $row['IdAutora'] .'>' . $row['Prezime'] . ",
" . $row['Ime'] . '</a></td>';
    echo '<td>' . $row['Naziv'] . '</td>';
    echo '<td>' . $row['Signatura'] . '</td>';

```

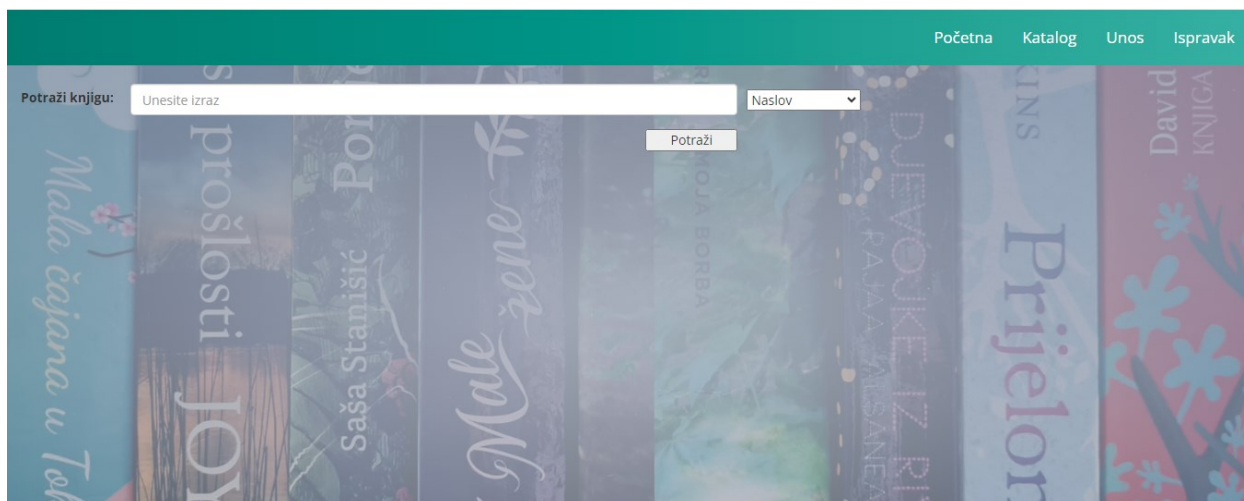
```

        echo '<td>' . $row['InventarniBroj'] . '</td>';
        echo '<td>' . $row['UDK'] . '</td>';
        echo '<td>' . $row['Predmetnica'] . '</td>';
        echo '</tr>';
    }
echo '</table>';
    }
}
echo '</table>';
}
?>
</div>
<?php include "footer.php";?>
</body>
</html>

```

Programski kôd 28. Prikaz sintakse rezultata pretraživanja kataloga.

Na kraju programskog kôda u datoteci je uključen footer. Slika 30 prikazuju stranicu kataloga te njegov prikaz nakon pretraživanja, a slika 31 prikaz stranice kataloga nakon pretraživanja izdavača „Školska knjiga“.



Slika 30. Prikaz stranice kataloga. Izvor: snimka vlastitog zaslona.

Naslov	Usporedni Naslov	Podnaslov	Autor	Izdavač	Signatura	Inventarni broj	UDK	Više
Albert Camus		Nobelova nagrada za književnost 1957. g.	Camus, Albert	Školska knjiga	821.133.1-31 CAM a	23823	821.133.1-31	roman-kronika - alegorijski roman - psihološki roman - francuska književnost
Albert Camus		Nobelova nagrada za književnost 1957. g.	Camus, Albert	Školska knjiga	821.133.1-31 CAM a	23824	821.133.1-31	roman-kronika - alegorijski roman - psihološki roman - francuska književnost
Čiča Goriot			Balzac, Honore de	Školska knjiga	821.133.1-31 BAL č	30364	821.133.1-31	realistički roman - francuska književnost - društveni roman
Evgenij Onjegin			Puškin, Aleksandar	Školska knjiga	821.161.1-31 PUŠ e	30435	821.161.1-31	roman u stihovima - ruska književnost - ljubavni roman - društveni roman
Gramatika hrvatskoga jezika		priručnik za osnovno jezično obrazovanje	Težak, Stjepko	Školska knjiga	808.62-5 TEŽ g	30121	808.62-5	Hrvatski jezik - Gramatika
Hrvatski			Babić	Školska	808.62-1 BAR			

Slika 31. Prikaz stranice kataloga nakon unosa pojma „Školska knjiga“. Izvor: snimka vlastitog zaslona.

8.4.6. Unos nove knjige

Odabirom na opciju „Unos nove knjige“ na početnoj stranici otvara se nova stranica pomoću koje korisnici imaju mogućnost unosa nove građe na temelju podataka o knjizi koja je definirana u bazi podataka, prikazana slikom 32.

Slika 32. Prikaz stranice „Unos nove knjige“. Izvor: snimka vlastitog zaslona.

Programskim kôdom 28 definirana je forma unosa podataka za dodavanje nove građe. Linijom kôda `<input type="text" name="Naslov" class="form-control" placeholder="Unesite naslov" required>` predstavlja polje za unos elemenata knjige, tj. naslova u ovom primjeru. Obuhvaća vrstu unosa, identifikator podataka te atribut `required` kojim se navodi da polje za unos mora ispunjeno prije podnošenja obrasca (W3Schools: HTML `<input>` `required` Attribute, n.d.).

```

<?php
    include "navigacija.php";
    ?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Unos knjige</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
</head>
<form class="main" action="<?php echo $_SERVER['PHP_SELF'];
?>" method="POST">
<div class="container" style="margin-top: 10px;" >
    <div class="row" style="padding-top: 80px;">
        <div class="col-md-6">
            <div class="form-group">
                <label >Naslov knjige:</label>
                <input type="text" name="Naslov" class="form-control"
placeholder="Unesite naslov" required>
            </div>
            <div class="form-group">
<label for="UsporedniNaslov" >Usporedni naslov knjige:</label>
                <input type="text" name="UsporedniNaslov" class="form-
control" placeholder="Unesite usporedni naslov">
            </div>
            <div class="form-group">
                <label for="Podnaslov">Podnaslov knjige:</label>
                <input type="text" name="Podnaslov" class="form-
control" placeholder="Unesite podnaslov">
            </div>

```

```
<div class="form-group">
  <label for="ISBN">ISBN:</label>
  <input type="text" name="ISBN" class="form-control"
placeholder="ISBN">
</div>
<div class="form-group">
  <label for="Signatura">Signatura:</label>
  <input type="text" name="Signatura" class="form-
control" placeholder="Signatura">
</div>
<div class="form-group">
  <label for="InventarniBroj">Inventarni broj:</label>
  <input type="text" name="InventarniBroj" class="form-
control" placeholder="Inventarni broj">
</div>
<div class="form-group">
  <label for="Svezak">Svezak:</label>
  <input type="text" name="Svezak" class="form-control"
placeholder="Svezak">
</div>
<div class="form-group">
  <label for="Predmetnica">Predmetnica:</label>
  <input type="text" name="Predmetnica" class="form-
control" placeholder="Predmetnica">
</div>
<div class="form-group">
  <label for="BrojStranica">Broj stranica:</label>
  <input type="text" name="BrojStranica" class="form-
control" placeholder="Broj stranica">
</div>
<div class="form-group">
```

```

        <label for="Prilozi">Prilozi:</label>
        <input type="text" name="Prilozi" class="form-control"
placeholder="Prilozi">
    </div>
</div>
<div class="col-md-6">
    <div class="form-group">
        <label for="Ime">Ime autora:</label>
        <input type="text" name="Ime" class="form-control"
placeholder="Ime autora">
    </div>
    <div class="form-group">
        <label for="Prezime">Prezime autora:</label>
        <input type="text" name="Prezime" class="form-control"
placeholder="Prezime autora">
    </div>
    <div class="form-group">
        <label for="Naziv">Naziv izdavača:</label>
        <input type="text" name="Naziv" class="form-control"
placeholder="Naziv">
    </div>
    <div class="form-group">
        <label for="Mjesto">Mjesto izdavanja:</label>
        <input type="text" name="Mjesto" class="form-control"
placeholder="Mjesto">
    </div>
    <div class="form-group">
        <label for="Godina">Godina izdavanja:</label>
        <input type="text" name="Godina" class="form-control"
placeholder="Godina">
    </div>

```

```

    <div class="form-group">
        <label for="UDK">UDK:</label>
        <input type="text" name="UDK" class="form-control"
placeholder="UDK">
    </div>
    <div class="form-group">
        <label for="NazivSkupine">Opis UDK skupine:</label>
        <input type="text" name="NazivSkupine" class="form-
control" placeholder="Naziv UDK skupine">
    </div>
    <div class="desno_poravnanje">
        <div > <button type="submit" name="submit" class="btn
btn-success btn-xm" style="width: 200px;">Spremi</button>
</div><span>
        <button id="ispravi" class="btn btn-outline-secondary
btn-xm" style="width: 100px;"
onclick="naispravak()">Ispravak</button></span>
    </div>
</div>
</div>
<div>
<?php include "footer.php";?>
</div>
<script>
    function naispravak() {
        window.location.href = "ispravak.php";
    }
</script>

```

Programski kôd 29. Definiranje unosa nove knjige u PHP datoteku „unos.php“.

Linija kôda <script> function naispravak()

{window.location.href = "ispravak.php";} </script> definira JavaScript funkciju, kojom kada se odabere opcija „Ispravak“, preusmjerava se na „ispravak.php“ (W3Schools: Window location.href, n.d.). Kada je korisnik popunio obrazac i odabrao opciju za slanje kako bi se zabilježila nova građa, izvršavaju se SQL upiti za unos knjige, autora, izdavača te UDK, prikazan programskim kôdom 29. Prilikom prvog pozivanja skripte, provjerava se postoji li ključ \$_POST['submit']. Ako obrazac nije poslan, ključ ne postoji te se zato prikazuje prazan obrazac (Vaswani, 2005). Kada se obrazac pošalje, ponovno se poziva ista skripta pa će PHP obraditi podatke iz obrasca i prikazati rezultate (Vaswani, 2005).

```
<?php
if(isset($_POST['submit']))
{
include "konekcija.php";
$sql = "INSERT INTO Autor (Ime, Prezime) VALUES
('$_POST[Ime]', '$_POST[Prezime]')";
mysqli_query($con,$sql) or die(mysqli_error($con));
$AutorID = mysqli_insert_id($con);
$sql = "INSERT INTO Izdavac (Naziv, Mjesto, Godina)
VALUES ('$_POST[Naziv]', '$_POST[Mjesto]', '$_POST[Godina]')";
mysqli_query($con,$sql) or die(mysqli_error($con));
$IzdavacID = mysqli_insert_id($con);
$sql = "INSERT INTO UDK (UDK, NazivSkupine) VALUES
('$_POST[UDK]', '$_POST[NazivSkupine]')";
mysqli_query($con,$sql) or die(mysqli_error($con));
$UDKID = mysqli_insert_id($con);
$sql = "INSERT INTO Knjiga (Naslov, UsporedniNaslov,
Podnaslov, ISBN, Signatura, InventarniBroj, Svezak,
Predmetnica, BrojStranica, Prilozi,AutorID,IzdavacID,UDKID)
```

```

VALUES
('$_POST[Naslov]',                                '$_POST[UsporedniNaslov]',
'$_POST[Podnaslov]',    '$_POST[ISBN]',    '$_POST[Signatura]',
'$_POST[InventarniBroj]',                                '$_POST[Svezak]',
'$_POST[Predmetnica]',                                '$_POST[BrojStranica]',
'$_POST[Prilozi]', $AutorID, $IzdavacID, $UDKID );
    mysqli_query($con,$sql) or die(mysqli_error($con));
    }
?>

```

Programski kôd 30. Funkcija unosa nove knjige.

Nakon definiranja programskog kôda koji omogućuje unos i pohranu, slika 33 prikazuje unos novih podataka u katalog privatne knjižnice.

Pretraživanje knjižnice		Kategorija
Početna	Katalog	Unos
Unesite pojam za pretraživanje		Naslov
Naslov knjige:	Ime autora:	
Digitalizacija	Hrvoje	
Usporedni naslov knjige:	Prezime autora:	
Unesite usporedni naslov	Stančić	
Podnaslov knjige:	Naziv izdavača:	
Unesite podnaslov	Zavod za informacijske studije	
ISBN:	Mjesto izdavanja:	
978-953-175-331-9	Zagreb	
Signatura:	Godina izdavanja:	
Signatura	2009	
Inventarni broj:	UDK:	
Inventarni broj	UDK	
Svezak:	Opis UDK skupine:	
Svezak	Naziv UDK skupine	

Slika 33. Unos novih podataka u katalog privatne knjižnice. Izvor: snimka vlastitog zaslona.

8.4.7. Rezultati pretraživanja autora

PHP sintaksom, prikazanom programskim kôdom 30, omogućuje se dohvaćanje i prikaz podataka na stranici prilikom pretrage autora. Linijom kôda `<table class="katalog-tablica" id="table" border="1" cellpadding="0" cellspacing="0" style="width: 95%">` definirano je zaglavlje tablice. Sintaksa `$id = trim($_GET['idautor']);` dohvaća ID autora te uklanja razmake i unaprijed definirane znakove s obje strane niza, dok funkcija `$res=mysqli_fetch_object($result);` dohvaća autora kao rezultat pretrage (W3Schoolss: PHP trim() Function, n.d.). Pomoću `if ($count<>0)` omogućuje se provjera je li broj redova različit od nule, čime se označava da su rezultati pretrage pronađeni (W3Schoolss: Console count(), n.d.).

```
<?php
    include ("navigacija.php");
?>
<html>
<head>
<title>Rezultati pretrage autora</title>
</head>
<body>
<div style="padding: 80px; min-height: 95vh;">
<table class="katalog-tablica" id="table" border="1"
cellpadding="0" cellspacing="0" style="width: 95%">

<?php
$id = trim($_GET['idautor']);
    include "konekcija.php";
        $sql="SELECT
naslov,podnaslov,predmetnica,inventarnibroj,autorid,isbn,usporod
```

```

ninaslov, knjigaid, udk, ime, prezime, naziv, mjesto, godina, svezak, idau-
tora, nazivskupine FROM
Izdavac INNER JOIN (UDK INNER JOIN (Autor INNER JOIN Knjiga ON
Autor.IdAutora = Knjiga.AutorID) ON UDK.idudk = Knjiga.UDKID)
ON Izdavac.IdIzdavaca = Knjiga.IzdavacID WHERE idautora= $id
GROUP BY concat(`naslov`, `podnaslov`, `prezime`, `ime`) order by
prezime ASC ";
$result = mysqli_query($con, $sql) or die(mysqli_error($con));
$count=mysqli_num_rows($result);
    if ($count<>0){
        $res=mysqli_fetch_object($result);

        echo '<h4>Naslovi od: '.$res->ime." " . $res->prezime.'</h4>';
echo '<tr align="center" >
<th><font size="3">Naslov i podnaslov</font> </th>
<th><font size="3">Svezak</font> </th>
    <th><font size="3">Usporedni naslov</font> </th>
    <th> <font size="3">Izdavač, (mjesto i godina)</font></th>

                <th> <font size="4">ISBN</font></th>
                <th> <font size="4">UDK</font></th>
    <th> <font size="4">Naziv skupine UDKk</font></th>';
        echo' </tr>';

```

Programski kôd 31. Pregled rezultata pretraživanja autora.

Prikaz preostalih rezultata pretrage prikazan je pomoću do-while petlje (Programski kôd 31). Prilikom vraćanja skupa rezultata, zauzima se određena količina memorije (Vaswani, 2005). Funkcija `mysqli_free_result()` služi za oslobađanje memorije povezane s

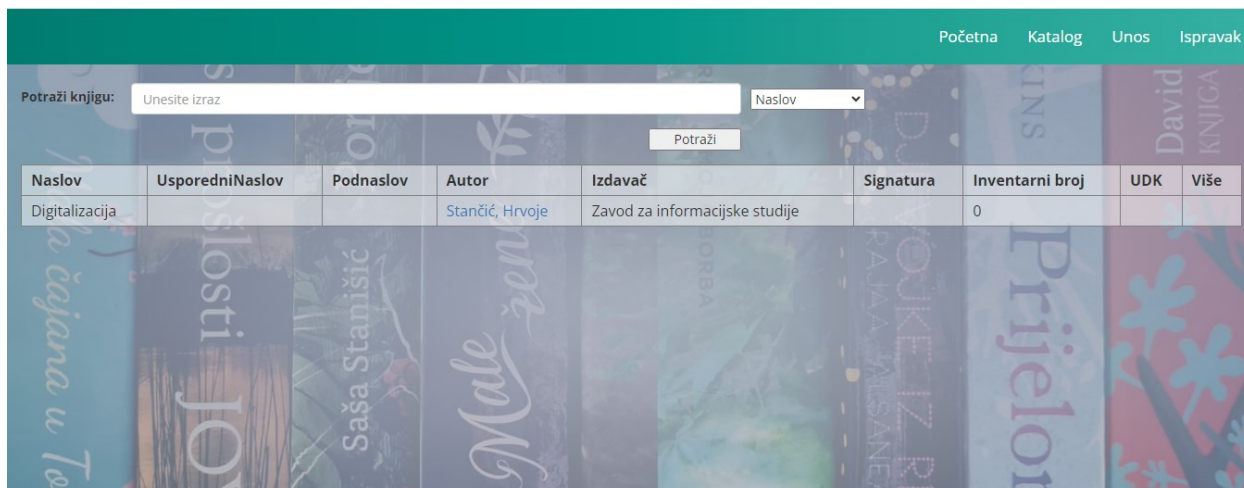
rezultatom upita (W3Schools: PHP mysqli_free_result() Function, n.d.). Prilikom završavanja upita, veza s poslužiteljem baze podataka prekida se pozivanjem funkcije `mysqli_close()`.

```
do {
    echo '<tr align="center" id="table">';
    echo '<td align="center"><font size="3" color=#800000 >'.$res->naslov.", " . $res->podnaslov.'</a></td>';
    echo '<td align="center"><font size="3" color=#800000>'.$res->svezak.'</td>';
    echo '<td align="center"><font size="3" color="800000">'.$res->usporedninaslov.'</td>';
    echo '<td align="center"><font size="3" color="800000">'.$res->mjesto." : ".$res->naziv.", ".$res->godina.'</td>';

    echo '<td><font size="3" color="800000">'.$res->isbn.'</td>';
    echo '<td><font size="3" color="800000">'.$res->udk.'</td>';
    echo '<td><font size="3" color="800000">'.$res->nazivskupine.'</td>';
    }while ($res = mysqli_fetch_object($result));
    echo '</tr>';
    echo '<tr></tr>';
}
mysqli_free_result($result);
mysqli_close($con);
?>
```

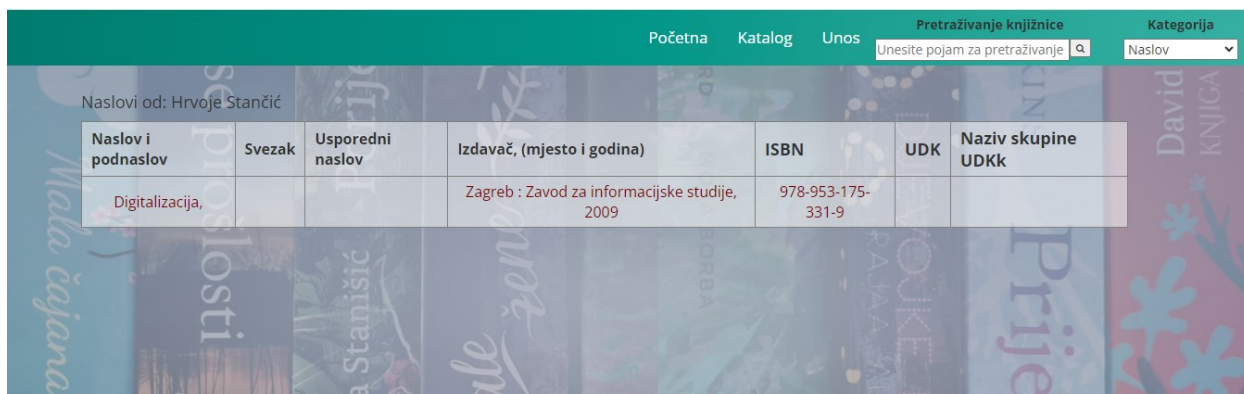
Programski kôd 32. Prikaz ostalih informacija tijekom pretraživanja autora.

Nakon unosa novih podataka prikazanih prethodnom slikom 33, na slici 34 pretražujemo autora kojeg smo dodali kako bi provjerili je li pohranjen u bazi.



Slika 34. Pretraživanje autora. Izvor: snimka vlastitog zaslona.

Prilikom odabira autora, otvara se nova stranica u kojoj su prikazana sva njegova djela pohranjena u bazi podataka (Slika 35).



Slika 35. Prikaz svih djela pojedinog autora u bazi podataka. Izvor: snimka vlastitog zaslona.

8.4.8. Prikaz rezultata pretraživanja knjige

Programski kôd 32 opisuje proces i prikaz rezultata prilikom pretraživanja određene knjige. Linija kôda `<table id="table" class="katalog-tablica"`

```
border="1" cellpadding="0" cellspacing="0" style="width: 100%">
prikazuje tablicu zaglavlja i vraćenih rezultata.
```

```
<?php
    include ("navigacija.php");
?>
<html>
<head>
    <title>Rezultati pretraživanja knjige</title>
</head>
<body>
<br>
<div class="main" style="padding: 80px; min-height: 94vh;">
<table id="table" class="katalog-tablica" border="1"
cellpadding="0" cellspacing="0" style="width: 100%">
<?php
if (isset($_GET['search'])) {
$pojam=$_GET['search'];

    if ($pojam<>"")
        $naslov=$_GET['category'];
        include ("konekcija.php");
        $sql="SELECT
naslov,podnaslov,predmetnica,inventarnibroj,autorid,isbn,uspored
ninaslov,knjigaid,udk,
ime,prezime,naziv,mjesto,godina,svezak,idautora FROM
Izdavac INNER JOIN (UDK INNER JOIN (Autor INNER JOIN Knjiga ON
Autor.IdAutora = Knjiga.AutorID) ON UDK.idudk = Knjiga.UDKID)
ON Izdavac.IdIzdavaca = Knjiga.IzdavacID where "." $naslov ". "
like      ". "%$pojam%"      ".      "      group      by
```

```

concat(`naslov`,`podnaslov`,`prezime`,`ime`) asc ";
$result = mysqli_query($con, $sql) or die(mysqli_error($con));
$count=mysqli_num_rows($result);

```

Programski kôd 33. PHP kôd za pretraživanje knjige.

Zaglavlja tablice definirana su programskim kôdom 33.

```

if ($count<>0){
echo '<tr align="center">
<th width="10%" id="table1_1"><font size="3">Autorstvo</font>
</th>
<th width="32%" id="table1_2"><font size="3">Naslov</font>
</th>
<th width="32%" id="table1_3"> <font size="3">Izdavač, (mjesto
i godina)</font></th>
<th width="20%" id="table1_4"> <font size="4">ISBN</font></th>
<th width="20%" id="table1_5"> <font size="4">UDK</font></th>'
echo' </tr>';

```

Programski kôd 34. Definiranje zaglavlje tablice.

Za prikaz rezultata, primjenjuje se programski kôd 34.

```

while ($res=mysqli_fetch_object($result)){
echo'<tr align="center" id="table">';
if ($res->prezime<>""){
echo'<td align="center"><font size="3" color=#800000 >
<a
target="_self" href=autor.php?idautor='.$res-

```



```

>autorid.'>'. $res->prezime.", " . $res->ime.'</a></td>';
}
else
    {echo'<td><font size="3" color="800000"></td>';}
if ($res->podnaslov==""){
    echo'<td align="center"><font size="3" color=#800000>'. $res->naslov.'</a></td>';
}
else
{
echo'<td><font size="3" color="800000">'. $res->naslov.'<br>'. "
: ". $res->podnaslov.'</td>';
    }
echo'<td align="center"><font size="3" color="800000">'. $res->mjesto." : ". $res->naziv.", ". $res->godina.'</td>';

<td><font size="3" color="800000">'. $res->isbn.'</td>';
echo'<td><font size="3" color="800000">'. $res->udk.'</td>';
    }
    echo'</tr>
    <tr></tr>';
}
mysqli_free_result($result);
mysqli_close($con);
}
}
?>
</table>
</div>
<div><?php include "footer.php";?></div>
</body>

```

</html>

Programski kôd 35. Prikaz rezultata pretrage.

8.4.9. Ispravak podataka

U slučaju pogrešno unesenih ili promijenjenih podataka koji se nalaze u bazi podataka, pomoću opcije „Ispravi“ moguće je ispraviti podatke na mrežnoj stranici, prikazano slikom 36. Slika 37 prikazuje kako se unosom inventarnog broja dohvaćaju svi postojeći podaci o knjizi.

Slika 36. Prikaz stranice „Ispravak“. Izvor: snimka vlastitog zaslona.

Početna		Katalog	Unos	Pretraživanje knjižnice	Kategorija
				Unesite pojam za pretraživanje	Naslov

Potraži inventarni broj:	<input type="text" value="23745"/>	<input type="button" value="Potraži"/>
Naslov knjige:	<input type="text" value="Surogat majka"/>	
Usporedni naslov knjige:	<input type="text" value="Unesite usporedni naslov"/>	
Podnaslov knjige:	<input type="text" value="Unesite podnaslov"/>	
ISBN:	<input type="text" value="978-953-6405-26-8"/>	
Signatura:	<input type="text" value="830-31 FIS s"/>	
Inventarni broj:	<input type="text" value="Inventarni broj"/>	
Ime autora:	<input type="text" value="Marie"/>	
Prezime autora:	<input type="text" value="Fischer"/>	
Naziv izdavača:	<input type="text" value="Škorpion"/>	
Mjesto izdavanja:	<input type="text" value="Zagreb"/>	
Godina izdavanja:	<input type="text" value="1998"/>	
UDK:	<input type="text" value="830-31"/>	
Opis UDK skupine:	<input type="text" value="Njemačka književnost - roman"/>	

Slika 37. Prikaz podataka pomoću pretraživanja inventarnog broja. Izvor: snimka vlastitog zaslona.

Kako bi se proveo ispravak, potrebno je dohvatiti podatke iz baze podataka te ih obraditi, što je prikazano programskim kôdom 35. Definiranjem funkcije `dohvati()`, dobiva se vrijednost elementa koja se pohranjuje u varijablu `invbrojsearch`. AJAX, tj. *Asynchronous JavaScript And XML*, predstavlja tehniku za pristup mrežnim poslužiteljima s mrežne stranice (W3Schools: AJAX Introduction, n.d.). Objekt `XMLHttpRequest` omogućuje ažuriranje dijelova mrežne stranice, bez potrebe za ponovnim učitavanjem cijele stranice (W3Schools: AJAX - The XMLHttpRequest Object, n.d.). Metoda `open()` služi za stvaranje asinkronog zahtjeva prema poslužitelju. U kôdu `xhr.open("GET", "ispravak_dohvat_podataka.php?invbrojsearch=" + invbrojsearch, true)`; definirana su tri argumenta (W3Schools: AJAX - XMLHttpRequest, n.d.):

- GET metoda;
- adresa datoteke na poslužitelju, koja može biti u bilo kojem formatu, poput `.txt` i `.xml` ili kao poslužiteljska skriptna datoteka, u formatu `.php` i `.asp` te `invbrojsearch` parametar;
- boolean vrijednost koja, ako je parametar asinkron, treba biti postavljena na `true`.

Asinkronim slanjem JavaScript ne čeka odgovor poslužitelja, već ima mogućnost izvršavanja druge skripte dok se čeka odgovor poslužitelja (W3Schools: AJAX - XMLHttpRequest, n.d.).

```
<script>
function dohvati()
{
varn                invbrojsearch                =
document.getElementById("invbrojsearch").value;
    var xhr = new XMLHttpRequest();
    xhr.open("GET",
"ispravak_dohvat_podataka.php?invbrojsearch=" + invbrojsearch,
true);

    xhr.onreadystatechange = function () {
        if (xhr.readyState == 4 && xhr.status == 200) {
            var data = JSON.parse(xhr.responseText);
document.getElementsByName("Naslov")[0].value = data.Naslov;
document.getElementsByName("UsporedniNaslov")[0].value =
data.UsporedniNaslov;
document.getElementsByName("Podnaslov")[0].value =
data.Podnaslov;
document.getElementsByName("ISBN")[0].value = data.ISBN;
document.getElementsByName("Signatura")[0].value =
data.Signatura;
document.getElementsByName("Svezak")[0].value = data.Svezak;
document.getElementsByName("Predmetnica")[0].value =
data.Predmetnica;
document.getElementsByName("BrojStranica")[0].value =
data.BrojStranica;
document.getElementsByName("Prilozi")[0].value = data.Prilozi;
```

```

document.getElementsByName("Ime")[0].value = data.Ime;
document.getElementsByName("Prezime")[0].value = data.Prezime;
document.getElementsByName("Naziv")[0].value = data.Naziv;
document.getElementsByName("Mjesto")[0].value = data.Mjesto;
document.getElementsByName("Godina")[0].value = data.Godina;
document.getElementsByName("UDK")[0].value = data.UDK;
document.getElementsByName("NazivSkupine")[0].value =
data.NazivSkupine;
    }
};
xhr.send();
}

```

Programski kôd 36. Programski kôd za dohvaćanje podataka iz baze podataka.

Nadalje, programskim kôdom 36 pripremaju se podaci za slanje.

```

function ispravi_podatke() {
    var formData = new FormData();
    formData.append("invbrojsearch",
document.getElementsByName("invbrojsearch")[0].value);
    formData.append("Naslov",
document.getElementsByName("Naslov")[0].value);
    formData.append("UsporedniNaslov",
document.getElementsByName("UsporedniNaslov")[0].value);
    formData.append("Podnaslov",
document.getElementsByName("Podnaslov")[0].value);
    formData.append("ISBN",
document.getElementsByName("ISBN")[0].value);
    formData.append("Signatura",

```

```

document.getElementsByName("Signatura")[0].value);
    formData.append("Svezak",
document.getElementsByName("Svezak")[0].value);
    formData.append("Predmetnica",
document.getElementsByName("Predmetnica")[0].value);
    formData.append("BrojStranica",
document.getElementsByName("BrojStranica")[0].value);
    formData.append("Prilozi",
document.getElementsByName("Prilozi")[0].value);
    formData.append("Ime",
document.getElementsByName("Ime")[0].value);
    formData.append("Prezime",
document.getElementsByName("Prezime")[0].value);
    formData.append("Naziv",
document.getElementsByName("Naziv")[0].value);
    formData.append("Mjesto",
document.getElementsByName("Mjesto")[0].value);
    formData.append("Godina",
document.getElementsByName("Godina")[0].value);
    formData.append("UDK",
document.getElementsByName("UDK")[0].value);
    formData.append("NazivSkupine",
document.getElementsByName("NazivSkupine")[0].value);

```

Programski kôd 37. Priprema podataka za slanje.

Funkcija `fetch` služi za asinkrono dohvaćanje. Prvi argument predstavlja URL na koji se šalje zahtjev, a drugi argument je *object* opcija. U opcijama se primjenjuje metoda POST te se dodaje argument *body* koji se implementira u *formData*, generiran u prethodnom kôdu (Mozilla: `fetch()` global function, n.d.) (Programski kôd 37).

```

fetch('update_knjige.php', {
  method: 'POST',
  body: formData
})

```

Programski kôd 38. Primjena fetch funkcije

Programski kôd 38 prikazuje programski kôd koji, ako se želi prikazati status zahtjeva, odgovor o statusu potrebno je pretvoriti u JSON format. JSON predstavlja tekstualni format za pohranu te prijenos podataka (W3Schools: JSON - Introduction, n.d.). U varijabli `jsonResponse` prikazan je JSON rezultat u obliku JavaScript objekta. Varijabla `jsonResponse.status` === 'success' predstavlja kako je objekt uspješno spremljen, a `document.getElementById("uspjeh").classList.remove('hidden');` prikazuje poruku.

```

.then((response) =>
{return response.json();
  }).then((jsonResponse) => {
  if (jsonResponse.status === 'success') {
    console.log('Uspješno spremljeno');

document.getElementById("uspjeh").classList.remove('hidden');
    setTimeout(() => {
document.getElementById("uspjeh").classList.add('hidden');
document.getElementsByName("invbrojsearch")[0].value = '';
document.getElementsByName("Naslov")[0].value = '';
document.getElementsByName("UsporedniNaslov")[0].value = '';
document.getElementsByName("Podnaslov")[0].value = '';

```

```

document.getElementsByName("ISBN")[0].value = '';
document.getElementsByName("Signatura")[0].value = '';
document.getElementsByName("Svezak")[0].value = '';
document.getElementsByName("Predmetnica")[0].value = '';
document.getElementsByName("BrojStranica")[0].value = '';
document.getElementsByName("Prilozi")[0].value = '';
document.getElementsByName("Ime")[0].value = '';
document.getElementsByName("Prezime")[0].value = '';
document.getElementsByName("Naziv")[0].value = '';
document.getElementsByName("Mjesto")[0].value = '';
document.getElementsByName("Godina")[0].value = '';
document.getElementsByName("UDK")[0].value = '';
document.getElementsByName("NazivSkupine")[0].value = '';
    }, 1 * 1000);    }

```

Programski kôd 39. Izvršavanje izmjene podataka.

Ako je pronađena pogreška u odgovoru, primjenjuje se skripta prikazana programskim kôdom 39.

```

else {
    throw(new Error(jsonResponse.message));
}
}).catch((error) => {
    console.error(error);
});
}
</script>

```

Programski kôd 40. Kôd za prikaz pogreške u odgovoru.

Programskim kôdom 40 provjerava se je li forma poslana te se SQL upitom dohvaćaju podaci iz baze podataka.

```
<?php
if (isset($_GET['invbrojsearch'])) {
    include "konekcija.php";
    $invbrojsearch = $_GET['invbrojsearch'];
    $sql="SELECT * FROM
        Izdavac INNER JOIN (UDK INNER JOIN (Autor INNER
JOIN Knjiga ON Autor.IdAutora = Knjiga.AutorID) ON UDK.idudk =
Knjiga.UDKID)
        ON Izdavac.IdIzdavaca = Knjiga.IzdavacID where
inventarnibroj=$invbrojsearch ";
    $result = mysqli_query($con, $sql) or
die(mysqli_error($con));
```

Programski kôd 41. Provjera forme i dohvaćanje podataka.

Nakon toga slijedi provjera sadrži li baza podataka rezultate (Programski kôd 41).

```
if ($row = mysqli_fetch_assoc($result)) {
    $data = array(
        'Naslov' => $row['Naslov'],
        'UsporedniNaslov'=> $row['UsporedniNaslov'],
        'Podnaslov' => $row['Podnaslov'],
        'ISBN' => $row['ISBN'],
        'Signatura' => $row['Signatura'],
```

```

        'InventarniBroj' => $row['InventarniBroj'],
        'Svezak' => $row['Svezak'],
        'Predmetnica' => $row['Predmetnica'],
        'BrojStranica' => $row['BrojStranica'],
        'Prilozi' => $row['Prilozi'],
        'Ime' => $row['Ime'],
        'Prezime' => $row['Prezime'],
        'Naziv' => $row['Naziv'],
        'Mjesto' => $row['Mjesto'],
        'Godina' => $row['Godina'],
        'UDK' => $row['UDK'],
        'NazivSkupine' => $row['NazivSkupine']
    );
    echo json_encode($data);
    }
}
?>

```

Programski kôd 42. Provjeravanje podataka u bazi.

Programskim kôdom 42 prikazana je naredba UPDATE koja prema vrijednosti ključa Autora, Knjige, Izdavaca ili UDK ažurira podatke iz baze podataka. Ako se ažuriranje provelo uspješno, šalje se uspješan status `echo json_encode(['status' => 'success']);` u protivnom prikazuje grešku `echo json_encode(['status' => 'error', 'message' => 'Greška prilikom ažuriranja podataka.']);` (W3Schools: PHP `json_encode()` Function, n.d.).

```

if ($row = mysqli_fetch_assoc($result)) {
    $sql = "update autor set Ime='$_POST[Ime]',

```

```

Prezime='$_POST[Prezime]' where IdAutora = '{$row['AutorID']}'
" ;
$updateAutor = mysql_query($con,$sql) or
die(mysql_error($con));
$sql = "UPDATE izdavac set Naziv='$_POST[Naziv]',
Mjesto='$_POST[Mjesto]', Godina='$_POST[Godina]' where
IdIzdavaca = '{$row['IzdavacID']}'";
$updateIzdavac = mysql_query($con,$sql) or
die(mysql_error($con));
$sql = "update udk set UDK='$_POST[UDK]',
NazivSkupine='$_POST[NazivSkupine]' where
idudk='{$row['UDKID']}'";
$updateUDK = mysql_query($con,$sql) or die(mysql_error($con));
$sql = "update knjiga set Naslov='$_POST[Naslov]',
UsporedniNaslov='$_POST[UsporedniNaslov]',
Podnaslov='$_POST[Podnaslov]', ISBN='$_POST[ISBN]',
    Signatura='$_POST[Signatura]',
InventarniBroj='$_POST[invbrojsearch]',
Svezak='$_POST[Svezak]', Predmetnica='$_POST[Predmetnica]',
    BrojStranica='$_POST[BrojStranica]',
Prilozi='$_POST[Prilozi]' where KnjigaID =
 '{$row['KnjigaID']}'";
$updateKnjiga = mysql_query($con,$sql) or
die(mysql_error($con));
}
if ($updateAutor && $updateIzdavac && $updateUDK &&
$updateKnjiga) {
    echo json_encode(['status' => 'success']);
} else {
    echo json_encode(['status' => 'error', 'message' => 'Greška
prilikom ažuriranja podataka.']);
}

```

```
}  
}  
?>
```

Programski kôd 43. Ažuriranje podataka tablica u bazi podataka.

9. Zaključak

Cilj ovog diplomskog rada bio je pružiti opis aplikacije za privatne knjižnice pomoću koje korisnici mogu unositi nove knjige, pregledavati i pretraživati vlastiti katalog prema određenim kriterijima te naknadno ispravljati podatke o knjizi koji su pogrešno unijeti. Projektom je opisana izrada baze podataka, tablica te njihova implementacija. Podaci sadržani u definiranim tablicama „Knjiga“, „Autor“, „Izdavac“ i „UDK“ predstavljaju temeljne elemente koji opisuju knjigu te pomoću kojih je organiziran ovaj sustav privatne knjižnice. Za izradu baze podataka korišten je Microsoft SQL Server Management Studio, a za izradu mrežne stranice primjenjivale su se sintakse HTML-a, CSS-a i PHP-a. Za lokalno izvršavanje PHP datoteka korišten je server WAMP. Evaluacijom knjižničnih aplikacija LibraryThing, Libib, CLZ Books, Book Collector i Calibre pruža se uvid napretka informacijske i komunikacijske tehnologije u području knjižničarstva. Opisane mobilne aplikacije imaju mogućnost skeniranja ISBN crtičnog koda putem kamere te prikazivanja informacija o knjizi, čime je značajno ubrzan proces dodavanja knjiga u vlastiti katalog. Prilikom izrade aplikacije privatne knjižnice, primjenjivala su se određena obilježja prethodno opisanih knjižničnih aplikacija, poput funkcionalnosti pretraživanja kataloga prema kriterijima naslova i autora. Izrada baze podataka predstavlja proces integriranja različitih aspekata, poput dizajna korisničkog sučelja, mogućnosti naprednog pretraživanja te automatiziranih procesa izvršavanja zadataka, kako bi sustav funkcionirao kao organizirana cjelina. Redovito proučavanje i analiziranje korisničkih potreba omogućuje dizajniranje značajki što kvalitetnijih aplikacija koje će korisnici primjenjivati u budućnosti. Također je potrebno pratiti trendove te novitete koji se primjenjuju u bazama podataka te u aplikacijama za upravljanje korisničkih zbirki za omogućavanje što automatiziranih procesa.

10. Literatura

1. Adebayo, O. A.; Ahmed, Y. O.; Adeniran, R. T. (2018). The Role of ICT in Provision of Library Services: A Panacea for sustainable development in Nigeria. *Library Philosophy and Practice* [online], str. 43-55. Dostupno na: <http://digitalcommons.unl.edu/libphilprac/1951> (Pristup 20.01.2024.).
2. Ashikuzzaman, MD. (2016). Impact of ICT on Modern Library. *LIS Education Network* [online]. Dostupno na: <https://www.lisedunetwork.com/impact-of-ict-on-modern-library/> (Pristup 09.01.2024.).
3. Bowers, S. K.; Polak, E. J. (2014). The Future of Cloud-Based Library Systems. In K. J. Varnum, ur. *The Top Technologies Every Librarian Needs to Know: A LITA Guide*. Chicago, IL: American Library Association, str. 43-55. Dostupno na: <https://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1082&context=libsp> (Pristup 08.01.2024.).
4. Britannica. *School Libraries*. Dostupno na: <https://www.britannica.com/topic/library/School-libraries#ref320713> (Pristup 02.11.2023.).
5. Calibre ebook management. *About calibre*. Dostupno na: <https://calibre-ebook.com/about> (Pristup 15.01.2024.).
6. Calibre ebook management *Get calibre*. Dostupno na: <https://calibre-ebook.com/download> (Pristup 15.01.2024.).
7. Castro, E. (2007). *HTML, XHTML & CSS: Visual QuickStart Guide*. Berkeley: Peachpit Press.
8. Choy, F. C. (2010). Digital library services: Towards mobile learning. In: *Seminar on E-books as Learning Resources in Chinese Libraries in Asia*. Kaohsiung, Taiwan: National Kaohsiung University of Applied Science, str. 1-10.
9. Codecademy. *What is a Web App?* Dostupno na: <https://www.codecademy.com/article/what-is-a-web-app> (Pristup 29.11.2023.).
10. Collectorz.com. *Book Collector*. Dostupno na: <https://www.collectorz.com/book/book-collector> (Pristup 19.01.2024.).

11. Collectorz.com. *Book Database Software - catalog your book Collector*. Dostupno na: <https://www.collectorz.com/book> (Pristup 19.01.2024.).
12. Collectorz.com. *CLZ Book*. Dostupno na: <https://www.collectorz.com/book/clz-books> (Pristup 19.01.2024.).
13. Collectorz.com. *Core, Cloud and Connect explained*. Dostupno na: <https://www.collectorz.com/book/book-collector/manual/windows/v15/en/topic/ccc> (Pristup 18.01.2024.).
14. Collectorz.com. *Welcome to Collectorz.com, the Collector database software!* Dostupno na: <https://www.collectorz.com/> (Pristup 18.12.2023.).
15. Fitzgerald, A. (2021). What Is a Web App? A Beginner's Guide. *HubSpot*. Dostupno na: <https://blog.hubspot.com/website/what-is-web-app> (Pristup 11.12.2023.).
16. Fonts Google. *Using web fonts from a font delivery service Google Fonts..* Dostupno na: https://fonts.google.com/knowledge/using_type/using_web_fonts_from_a_font_delivery_service (Pristup 15.12.2023.).
17. Gillis, A. S. (2021). application. *TechTarget*. Dostupno na: <https://www.techtarget.com/searchsoftwarequality/definition/application> (Pristup 21.12.2023.).
18. Gomez, J. (2023a). How to Design an App. *koombea*. Dostupno na: <https://www.koombea.com/blog/how-to-design-an-app/> (Pristup 21.11.2023.).
19. Gomez, J. (2023b). Intelligent App Design: Tips for Creating Apps with Outstanding UX in 2023. *koombea*. Dostupno na: <https://www.koombea.com/blog/intelligent-app-design-tips-for-creating-apps-with-outstanding-ux/> (Pristup 21.11.2023.).
20. Gradska i sveučilišna knjižnica Osijek. *ISBN | ISSN | ISMN*. Dostupno na: <https://www.gskos.unios.hr/index.php/isbn-issn-ismn/> (Pristup 17.12.2023.).
21. Gradska knjižnica i čitaonica Petrinja. *Katalog*. Dostupno na: https://www.gkc-petrinja.hr/gkc_shop/katalog_products.php (Pristup 29.11.2023.).
22. ISBNdb. *Book Collection Management Made Easy by Collectorz.com (CLZ)*. Dostupno na: <https://isbndb.com/blog/book-collection-management-made-easy-by-collectorz-com/> (Pristup 15.01.2024.).
23. Jagersma, R.; Blom, H.; Chayes, E.; Hansen, A-M., ur. (2023). *Private Libraries and Their Documentation, 1665-1830: Studying and Interpreting Sources (Library of the*

- Written Word / Library of the Written Word - the Handpress World, 112*). Leiden: Brill Academic Pub. Dostupno na: <https://brill.com/edcollbook-oa/title/64633> (Pristup 22.01.2024.).
24. Libib. *Cloud Cataloging*. Dostupno na: <https://www.libib.com/> (Pristup 19.01.2024.).
25. Libib. *Want More? Go Pro!* Dostupno na: <https://www.libib.com/pricing> (Pristup 29.12.2023.).
26. LibraryThing. *A Short Overview of LibraryThing*. Dostupno na: <https://www.librarything.com/quickstart.php> (Pristup 18.01.2024.).
27. LibraryThing. *About LibraryThing*. Dostupno na: <https://www.librarything.com/about> (Pristup 18.01.2024.).
28. LibraryThing. *LibraryThing: A Home For Your Books*. Dostupno na: <https://www.librarything.com/> (Pristup 18.01.2024.).
29. LibraryThing. *TinyCat*. Dostupno na: <https://www.librarycat.org/> (Pristup 18.01.2024.).
30. Microsoft. *Download SQL Server Management Studio (SSMS)*. Dostupno na: <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> (Pristup 15.01.2024.).
31. Mozilla. *fetch() global function*. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/API/fetch> (Pristup 25.01.2024.).
32. Mozilla. *linear-gradient()*. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/CSS/gradient/linear-gradient> (Pristup 21.11.2023.).
33. Nalluri, S. R.; Gaddam, B. (2016). *Mobile Library Services and Technologies: A Study*. *International Journal of Research in Library Science*, 2(2), str. 59-66. Dostupno na: <https://www.ijrils.in/wp-content/uploads/2016/08/Mobile-Library-Services-and-Technologies-A-Study.pdf> (Pristup 30.11.2023.).
34. Negulić, T. (2014). *Calibre – kreiranje e-knjiga*. e-laboratorij CARNET. Dostupno na: <https://e-laboratorij.carnet.hr/calibre-kreiranje-e-knjiga/> (Pristup 18.01.2024.).
35. New World Encyclopedia. *Private library*. Dostupno na: https://www.newworldencyclopedia.org/entry/Private_library (Pristup 02.11.2023.).
36. Onuoha, J. A.; Obialor, D. C. (2015). *The Impact of Information Technology on Modern Librarianship: A Reflective Study*. *Information and Knowledge Management*, 5(11), str. 52-58. Dostupno na: <https://core.ac.uk/reader/234672211> (Pristup 04.01.2024.).

37. Rafaels, R. (2018). *Cloud computing: From beginning to end: Cloud technology, design, and migration methodologies explained*. California: CreateSpace.
38. Rohn, S. (2022). What Is a Desktop Application? +Challenges, Use Cases. *Whatfix Blog*. Dostupno na: <https://whatfix.com/blog/desktop-application/> (Pristup 24.01.2024.).
39. Rubin, R. E., & Rubin, R. G. (2020). *Foundations of Library and Information Science*. London: Facet.
40. Shields, W. (2019). *SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data with SQL*. New York: ClydeBank Technology.
41. Spalding, Tim. (2020). *LibraryThing Is Now Free to All*. LibraryThing. Dostupno na: <https://blog.librarything.com/2020/03/librarything-goes-free/> (Pristup 18.01.2024.).
42. Swain, D. K. (2014). Cloud Computing and its Application in Library Management: A Review of Research. *e-Library Science Research Journal*. 2(4), str. 1-10. Dostupno na: https://www.researchgate.net/publication/298721580_Cloud_computing_and_its_application_in_library_management (Pristup 15.11.2023.).
43. Tadić, K. (1994). *Rad u knjižnici : priručnik za knjižničare*. Opatija: Naklada Benja.
44. TechTarget. *Web application (web app)*. Dostupno na: <https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app> (Pristup 11.12.2023.).
45. Tkalac, S. (1999). *Oblikovanje baze podataka: sinopsis predavanja*. Varaždin: Fakultet organizacije i informatike.
46. Tatroe, K., MacIntyre, P. i Lerdorf, R. (2015). *Programiranje PHP*. 3. izd. Zagreb: Dobar plan.
47. *Univerzalna decimalna klasifikacija: hrvatsko džepno izdanje*. (2003). Zagreb: Naklada Nediljko Dominović.
48. V2 Cloud. *What is a Desktop App?* Dostupno na: <https://v2cloud.com/glossary/what-is-a-desktop-app> (Pristup 24.01.2024.).
49. Varga, M. (2020). *Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka*. (2. izd.) [online]. Dostupno na: https://books.google.hr/books/about/Baze_podataka.html?id=UQPoDwAAQBAJ&redir_esc=y (Pristup 19.11.2023.).

50. Vaswani, V. (2005). *Kako upotrijebiti PHP i MySQL*. Zagreb: Mikro knjiga.
51. W3Schools. *AJAX Introduction*. Dostupno na: https://www.w3schools.com/js/js_ajax_intro.asp (Pristup 20.11.2023.).
52. W3Schools. *AJAX - The XMLHttpRequest Object*. Dostupno na: https://www.w3schools.com/js/js_ajax_http.asp (Pristup 20.11.2023.).
53. W3Schools. *AJAX - XMLHttpRequest*. Dostupno na: https://www.w3schools.com/js/js_ajax_http_send.asp (Pristup 20.11.2023.).
54. W3Schools. *Bootstrap Get Started*. Dostupno na: https://www.W3Schoolss.com/bootstrap/bootstrap_get_started.asp (Pristup 19.01.2024.).
55. W3Schools. *Console count()*. Dostupno na: https://www.w3schools.com/jsref/met_console_count.asp (Pristup 20.11.2023.).
56. W3Schools. *CSS background Property*. Dostupno na: https://www.W3Schoolss.com/cssref/css3_pr_background.php (Pristup 26.11.2023.).
57. W3Schools. *CSS Box Sizing*. Dostupno na: https://www.W3Schoolss.com/css/css3_box-sizing.asp (Pristup 20.11.2023.).
58. W3Schools. *CSS Flex Items*. Dostupno na: https://www.W3Schoolss.com/css/css3_flexbox_items.asp (Pristup 20.11.2023.).
59. W3Schools. *CSS justify-content Property*. Dostupno na: https://www.W3Schoolss.com/cssref/css3_pr_justify-content.php (Pristup 20.11.2023.).
60. W3Schools. *CSS Padding*. Dostupno na: https://www.W3Schoolss.com/css/css_padding.asp (Pristup 20.11.2023.).
61. W3Schools. *CSS Selectors*. Dostupno na: https://www.W3Schoolss.com/css/css_selectors.asp (Pristup 20.11.2023.).
62. W3Schools. *CSS Syntax*. Dostupno na: https://www.W3Schoolss.com/css/css_syntax.asp (Pristup 20.11.2023.).
63. W3Schools. *CSS Text*. Dostupno na: https://www.W3Schoolss.com/css/css_text.asp (Pristup 20.11.2023.).
64. W3Schools. *HTML <input> required Attribute*. Dostupno na: https://www.w3schools.com/tags/att_input_required.asp (Pristup 19.01.2024.).
65. W3Schools. *HTML Attributes*. Dostupno na: https://www.W3Schoolss.com/html/html_attributes.asp (Pristup 17.11.2023.).

66. W3Schools. *HTML Basic Examples.* Dostupno na: https://www.W3Schoolss.com/html/html_basic.asp (Pristup 17.11.2023.).
67. W3Schools. *HTML Elements.* Dostupno na: https://www.W3Schoolss.com/html/html_elements.asp (Pristup 15.11.2023.).
68. W3Schools. *HTML Headings.* Dostupno na: https://www.W3Schoolss.com/html/html_headings.asp (Pristup 15.11.2023.).
69. W3Schools. *HTML Introduction.* Dostupno na: https://www.W3Schoolss.com/html/html_intro.asp (Pristup 15.11.2023.).
70. W3Schools. *HTML <footer> Tag.* Dostupno na: https://www.W3Schoolss.com/tags/tag_footer.asp (Pristup 16.11.2023.).
71. W3Schools. *How To Add CSS.* Dostupno na: https://www.W3Schoolss.com/css/css_howto.asp (Pristup 21.11.2023.).
72. W3Schools. *JavaScript JSON.* Dostupno na: https://www.W3Schoolss.com/js/js_json.asp (Pristup 27.12.2023.).
73. W3Schools. *JSON - Introduction.* Dostupno na: https://www.w3schools.com/js/js_json_intro.asp (Pristup 27.12.2023.).
74. W3Schools. *PHP json_encode() Function.* Dostupno na: https://www.w3schools.com/php/func_json_encode.asp (Pristup 27.12.2023.).
75. W3Schools. *PHP mysqli_free_result() Function.* Dostupno na: https://www.w3schools.com/php/func_mysqli_free_result.asp (Pristup 20.11.2023.).
76. W3Schools. *PHP Superglobal - \$_GET.* Dostupno na: https://www.W3Schoolss.com/php/php_superglobals_get.asp (Pristup 03.01.2024.).
77. W3Schools. *PHP trim() Function.* Dostupno na: https://www.w3schools.com/php/func_string_trim.asp (Pristup 03.01.2024.).
78. W3Schools. *SQL CREATE DATABASE Statement.* Dostupno na: https://www.W3Schoolss.com/sql/sql_create_db.asp (Pristup 20.11.2023.).
79. W3Schools. *Window location.href.* Dostupno na: https://www.w3schools.com/jsref/prop_loc_href.asp (Pristup 20.11.2023.).
80. WampServer. *WampServer, a Windows web development environment.* Dostupno na: <https://www.wampserver.com/en/> (Pristup 29.12.2023.).

81. Xu, D.; Yang, L. (2014). Digital Library and Cloud Computing Technology. U: Liu, D. L., Zhu, X. B., Xu, K. L. i Fang, D. M., ur. *Applied Science, Materials Science and Information Technologies in Industry*. Zurich: Trans Tech Publications, vol. 513-517, str. 2170-2173.

11. Popis priloga

Popis slika

Slika 1. Korisničko sučelje LibraryThing aplikacije. Izvor: snimka vlastitog zaslona.	18
Slika 2. Mogućnosti dodavanja knjiga u aplikaciji LibraryThing. Izvor: snimka vlastitog zaslona.	19
Slika 3. Prikaz kataloga LibraryThing aplikacije nakon dodavanja knjiga. Izvor: snimka vlastitog zaslona.....	20
Slika 4. Kriteriji sortiranja u LibraryThing aplikaciji. Izvor: snimka vlastitog zaslona.	20
Slika 5. Pregled knjige u LibraryThing aplikaciji. Izvor: snimka vlastitog zaslona.....	21
Slika 6. Korisničko sučelje aplikacije Libib. Izvor: snimka vlastitog zaslona.	23
Slika 7. Prikaz kataloga aplikacije Libib.Izvor: snimka vlastitog zaslona.	24
Slika 8. Odabir medijskog formata za pohranu u katalog u aplikaciji Libib. Izvor: snimka vlastitog zaslona.....	25
Slika 9. Pregled knjige u Libib aplikaciji. Izvor: snimka vlastitog zaslona.....	25
Slika 10. Informacije o knjizi u Libib aplikaciji. Izvor: snimka vlastitog zaslona.	26
Slika 11. Opis knjige unutar Libib aplikacije. Izvor: snimka vlastitog zaslona.	26
Slika 12. Korisničko sučelje CLZ Books aplikacije. Izvor: snimka vlastitog zaslona.	28
Slika 13. Zbirke unutar CLZ Books aplikacije. Izvor: snimka vlastitog zaslona	28
Slika 14. Dodavanje novih knjiga pomoću skeniranja ISBN kôda u aplikaciji CLZ Books. Izvor: https://www.collectorz.com/book/clz-books (Pristup 19.01.2024.)	29
Slika 15. Detalji o knjizi unutar CLZ Books aplikacije. Izvor: https://www.collectorz.com/book/clz-books (Pristup 19.01.2024.)	29
Slika 16. Korisničko sučelje Book Collector aplikacije. Izvor: snimka vlastitog zaslona.	30
Slika 17. Postupak dodavanja knjiga u aplikaciji Book Collector. Izvor: snimka vlastitog zaslona.	31
Slika 18. Korisničko sučelje aplikacije Calibre. Izvor: snimka vlastitog zaslona.	33
Slika 19. Opcija nabave knjiga unutar Calibre aplikacije. Izvor: snimka vlastitog zaslona.	34
Slika 20. Podaci za unos u tablice „Knjiga“, „Autor“, „Izdavac“ i „UDK“. Izvor: Gradska knjižnica i čitaonica Petrinja.....	43

Slika 21. Odabir datoteke za pohranu u bazi podataka. Izvor: snimka vlastitog zaslona.	44
Slika 22. Prikaz učitanih stupaca, tipa podataka, primarnog ključa te NULL vrijednosti „Knjige.txt datoteke“. Izvor: snimka vlastitog zaslona.	45
Slika 23. Primjena naredbe SELECT * za prikaz tablica i njihovih podataka u bazi podataka. Izvor: snimka vlastitog zaslona.....	48
Slika 24. Selektor i deklaracija CSS sintakse. Izvor: https://www.W3Schoolss.com/css/css_syntax.asp (Pristup 20.11.2023.).	54
Slika 25. Pozadinska slika privatne knjižnice. Izvor: vlastita izrada.	56
Slika 26. Početna stranica privatne knjižnice. Izvor: snimka vlastitog ekrana.	61
Slika 27. Import baze podataka u phpMyAdmin. Izvor: snimka vlastitog zaslona.	63
Slika 28. Poruka uspješnog uvoza podataka u phpMyAdmin bazu podataka „PrivatnaKnjiznica“. Izvor: snimka vlastitog zaslona.....	63
Slika 29. Početna stranica privatne knjižnice. Izvor: snimka vlastitog zaslona.....	69
Slika 30. Prikaz stranice kataloga. Izvor: snimka vlastitog zaslona.	77
Slika 31. Prikaz stranice kataloga nakon unosa pojma „Školska knjiga“. Izvor: snimka vlastitog zaslona.....	77
Slika 32. Prikaz stranice „Unos nove knjige“. Izvor: snimka vlastitog zaslona.	78
Slika 33. Unos novih podataka u katalog privatne knjižnice. Izvor: snimka vlastitog zaslona. ...	84
Slika 34. Pretraživanje autora. Izvor: snimka vlastitog zaslona.	88
Slika 35. Prikaz svih djela pojedinog autora u bazi podataka. Izvor: snimka vlastitog zaslona...	88
Slika 36. Prikaz stranice „Ispravak“. Izvor: snimka vlastitog zaslona.	92
Slika 37. Prikaz podataka pomoću pretraživanja inventarnog broja. Izvor: snimka vlastitog zaslona.....	93

Popis tablica

Tablica 1. Razlike između mrežnih i desktop aplikacija Izvor: Rohn, 2022.	11
Tablica 2. Pregled funkcionalnosti aplikacija LibraryThing, Libib, CLZ Books, Book Collector i Calibre.....	35
Tablica 3. Podržani medijski podaci za katalogiziranje u aplikacijama funkcionalnosti aplikacija LibraryThing, Libib, CLZ Books, Book Collector i Calibre.	36

Tablica 4. Operacijski sustavi koje podržavaju aplikacije LibraryThing, Libib, CLZ Books, Book Collector i Calibre.....	37
Tablica 5. Tablica „Autor“. Izvor: vlastita izrada.....	41
Tablica 6. Tablica „Izdavac“. Izvor: vlastita izrada.....	41
Tablica 7. Tablica „UDK“. Izvor: Vlastita izrada.....	42
Tablica 8. Tablica „Knjiga“. Izvor: Vlastita izrada.....	42

Popis programskih kôdova

Programski kôd 1. Stvaranje baze podataka „PrivatnaKnjiznica“.....	40
Programski kôd 2. Primjeri HTML elemenata. Izvor: W3Schools: HTML Elements.....	49
Programski kôd 3. Zaglavlje početne stranice privatne knjižnice.....	50
Programski kôd 4. Naslov i opis privatne knjižnice.....	51
Programski kôd 5. Hiperlinkovi unutar početne stranice privatne knjižnice.....	52
Programski kôd 6. Navigacija, pretraživanje i kategorije padajućeg izbornika početne stranice privatne knjižnice.....	53
Programski kôd 7. Footer privatne knjižnice.....	53
Programski kôd 8. Dodavanje CSS skripte u HTML kôd.....	54
Programski kôd 9. Korištenje fonta pomoću unaprijed definiranih Google Font stila.....	54
Programski kôd 10. CSS kôd za dimenzioniranje svih HTML elemenata na stranici.....	55
Programski kôd 11. CSS sintaksa za postavljanje i uređivanje pozadinske slike.....	57
Programski kôd 12. CSS sintaksa oblikovanja naslova i paragrafa.....	57
Programski kôd 13. CSS oblikovanje hiperlinkova.....	58
Programski kôd 14. CSS sintaksa za oblikovanje stila pretraživanja.....	59
Programski kôd 15. CSS sintaksa za oblikovanje footer elementa.....	60
Programski kôd 16. Stvaranje konekcije nad bazom podataka.....	64
Programski kôd 17. PHP naredba za uključivanje jedne datoteke u drugu.....	65
Programski kôd 18. Struktura datoteke „navigacija.php“.....	66
Programski kôd 19. Struktura datoteke „index.php“.....	68
Programski kôd 20. Zaglavlje kataloga.....	69
Programski kôd 21. Struktura navigacije u datoteci „header_katalog.php“.....	70

Programski kôd 22. Uključivanje navigacije u programski kôd datoteke „katalog.php“	70
Programski kôd 23. Uporaba if() naredbe (Vaswani, 2005, str. 66).	71
Programski kôd 24. Uporaba while() petlje (Vaswani, 2005, str. 73).	71
Programski kôd 25. Definiranje elemenata stranice kataloga.	73
Programski kôd 26. Dohvaćanje podataka iz baze podataka prilikom postavljanja upita.	74
Programski kôd 27. Prikaz sintakse rezultata pretraživanja kataloga.	76
Programski kôd 28. Definiranje unosa nove knjige u PHP datoteku „unos.php“	82
Programski kôd 29. Funkcija unosa nove knjige.	84
Programski kôd 30. Pregled rezultata pretraživanja autora.	86
Programski kôd 31. Prikaz ostalih informacija tijekom pretraživanja autora.	87
Programski kôd 32. PHP kôd za pretraživanje knjige.	90
Programski kôd 33. Definiranje zaglavlje tablice.	90
Programski kôd 34. Prikaz rezultata pretrage.	92
Programski kôd 35. Programski kôd za dohvaćanje podataka iz baze podataka.	95
Programski kôd 36. Priprema podataka za slanje.	96
Programski kôd 37. Primjena fetch funkcije.	97
Programski kôd 38. Izvršavanje izmjene podataka.	98
Programski kôd 39. Kôd za prikaz pogreške u odgovoru.	98
Programski kôd 40. Provjera forme i dohvaćanje podataka.	99
Programski kôd 41. Provjeravanje podataka u bazi.	100
Programski kôd 42. Ažuriranje podataka tablica u bazi podataka.	102

Popis dijagrama

Dijagram 1. Uspostavljene relacije među tablicama.	46
---	----

Dizajn, razvoj i implementacija baze podataka za potrebe privatne knjižnice

Ovaj diplomski rad pruža pregled procesa izrade, dizajna te implementacije baze podataka za privatnu knjižnicu. Definirane su osnovne postavke definicije privatne knjižnice te je opisan utjecaj razvoja informacijske i komunikacijske tehnologije na procese automatizacije prilikom upravljanja knjižnicom. Objasnjene su osnovne teorijske postavke mobilnih, računalnih i web aplikacija koje se primjenjuju za organiziranje vlastitih korisničkih zbirki. Opisano i evaluirano je pet knjižničnih aplikacija za organizaciju vlastitih zbirki: LibraryThing, Libib, CLZ Books, Book Collector i Calibre. Pregledavanjem njihovih osnovnih funkcionalnosti, pružen je uvid za izgradnju sustava privatne knjižnice. Za izradu baze podataka privatne knjižnice korišten je Microsoft SQL Server Management Studio, a korisničko sučelje je implementirano pomoću HTML, CSS i PHP sintaksi. Privatna knjižnica organizirana je na način da sadrži početnu stranicu, koja dalje pruža mogućnost odabira kataloga, kojeg je moguće pretraživati po različitim kategorijama, opciju unosa nove knjige pomoću koje korisnici mogu unijeti novu građu upisivanjem navedenih podataka u obrascu te opciju ispravka, kojom je moguće izmijeniti podatke o građi.

Ključne riječi: privatna knjižnica, knjižnične aplikacije, baze podataka, programiranje, SQL, HTML, CSS, PHP.

Design, development and implementation of a database for the needs of a private library

This master thesis provides an overview of the processes of creating, designing and implementing a database for a private library. The basic settings of the definition of a private library are defined, and the impact of the development of information and communication technology on automation processes during library management is described. The basic theoretical settings of mobile, computer and web applications that are used to organize one's own user Collectors are explained. Five library applications for organizing one's own Collectors are described and evaluated: LibraryThing, Libib, CLZ Books, Book Collector and Calibre. By reviewing their basic functionalities, insight was provided for building a private library system. Microsoft SQL Server Management Studio was used to create the database of the private library, and the user interface was implemented using HTML, CSS and PHP syntax. The private library is organized in such a way that it contains a home page, which further provides the option of selecting a catalog, which can be searched by different categories, the option of entering a new book, with which users can enter new material by entering the specified data in the form, and the option of correction, which can be used to change material data.

Keywords: private library, library applications, databases, programming, SQL, HTML, CSS, PHP.