Sadržaj

[1. UVOD 1](#_Toc197987006)

[2. TEORIJSKA PODLOGA 3](#_Toc197987007)

[2.1. HTML, CSS i JavaScript 3](#_Toc197987008)

[2.2. PHP 4](#_Toc197987009)

[2.3. XAMPP 4](#_Toc197987010)

[2.4. MySQL 4](#_Toc197987011)

[2.5. Modeli baze podataka 5](#_Toc197987012)

[2.5.1. Hijerarhijski model 5](#_Toc197987013)

[2.5.2. Mrežni model 5](#_Toc197987014)

[2.5.3. Plošni model 6](#_Toc197987015)

[2.5.4. Objektni model 6](#_Toc197987016)

[2.5.5. Relacijski model 7](#_Toc197987017)

[3. TEHNIČKO IDEJNO RJEŠENJE 9](#_Toc197987018)

[3.1. Frontend tehnologije 9](#_Toc197987019)

[3.2. Backend tehnologije 10](#_Toc197987020)

[3.3. Baza podataka 10](#_Toc197987021)

[3.4. Lokalno razvojno okruženje 11](#_Toc197987022)

[3.5. Način rada sustava 11](#_Toc197987023)

[4. ARHITEKTURA BAZE PODATAKA 12](#_Toc197987024)

[4.1. Logička razina 12](#_Toc197987025)

[4.2. Fizikalna razina 13](#_Toc197987026)

[4.3. Konceptualna razina 13](#_Toc197987027)

[4.4. Sažetak arhitekture baze podataka 14](#_Toc197987028)

[5. IZRADA BAZE PODATAKA 15](#_Toc197987029)

[5.1. Kreiranje veze 15](#_Toc197987030)

[5.2. Kreiranje baze podataka 16](#_Toc197987031)

[5.3. Kreiranje tablice 16](#_Toc197987032)

[5.4. Tipovi podataka 17](#_Toc197987033)

[5.5. Ograničenja i pravila 18](#_Toc197987034)

[5.6. Ubacivanje podataka 18](#_Toc197987035)

[5.7. Pregled podataka u bazi 18](#_Toc197987036)

[5.8. Kratki osvrt na poglavlje 19](#_Toc197987037)

[6. IZRADA WEB STRANICE 20](#_Toc197987038)

[6.1. Izrada strukture stranice (HTML) 20](#_Toc197987039)

[6.1.1. Navigacija i raspored 20](#_Toc197987040)

[6.2. Opis početne stranice – index.html 21](#_Toc197987041)

[6.3. Stranica za pretragu knjiga – pretraga.html 22](#_Toc197987042)

[6.3.1. Opis HTML tablice 23](#_Toc197987043)

[6.3.2. Filtracija knjiga po autoru 24](#_Toc197987044)

[6.4. Rezervacija knjige – rezervacija.html 25](#_Toc197987045)

[6.4.1. Osvrt na HTML dio 27](#_Toc197987046)

[6.5. Stilizacija stranice (CSS) 27](#_Toc197987047)

[6.5.1. Osnovni stil tijela stranice 27](#_Toc197987048)

[6.5.2. Navigacijska traka 28](#_Toc197987049)

[6.5.3. Sekcije i sadržaj 29](#_Toc197987050)

[6.5.4. Gumbi i forme 30](#_Toc197987051)

[6.5.5. Galerija slika i Lightbox prikaz 30](#_Toc197987052)

[6.5.6. Prilagodljiv sadržaj 32](#_Toc197987053)

[6.5.7. Sažetak CSS dijela 32](#_Toc197987054)

[6.6. Interaktivnost stranice (JavaScript) 32](#_Toc197987055)

[6.6.1. Animacijska sekcija prilikom učitavanja stranice 32](#_Toc197987056)

[6.6.2. Mekana navigacija između stranica 33](#_Toc197987057)

[6.6.3. Lightbox prikaz slika u galeriji 34](#_Toc197987058)

[6.6.4. Kratki osvrt JavaScript dijela 34](#_Toc197987059)

[7. SPAJANJE WEB STRANICE I BAZE PODATAKA 35](#_Toc197987060)

[7.1. Korišteni programi 35](#_Toc197987061)

[7.2. Stranica za pretragu (pretraga.html) 35](#_Toc197987062)

[7.3. Dohvat podataka pute AJAX-a 36](#_Toc197987063)

[7.4. PHP skripta (pretraga\_knjiga.php) 36](#_Toc197987064)

[7.5. Sažetak poglavlja 38](#_Toc197987065)

[8. ZAKLJUČAK 39](#_Toc197987066)

# UVOD

Knjižnica je ustanova koja prikuplja, čuva, obrađuje i omogućuje pristup različitim vrstama pisanih, tiskanih i elektroničkih izvora znanja. Osim knjiga, knjižnice često nude i druge medije poput časopisa, filmova i digitalnih izvora te služe kao važno mjesto za obrazovanje, istraživanje i razonodu.

U školskom okruženju knjižnica ima posebnu ulogu jer omogućuje učenicima i nastavnicima pristup dodatnim izvorima informacija koji su ključni za učenje i razvoj. Upravo zbog važnosti knjižnice kao obrazovnog centra, ključno je osigurati da njezini resursi budu lako dostupni svim korisnicima.

Tradicionalni način upravljanja knjižnicom, koji se temelji na ručnom vođenju evidencija o posudbama i dostupnosti knjiga, postaje sve neefikasnije u suvremenom digitalnom dobu. Često dolazi do problema kao što su gubitak podataka, sporost u pronalasku informacija te nemogućnost pravovremenog informiranja korisnika o stanju knjiga.

Razvojem tehnologije i sve većom prisutnošću interneta, škole sve više prepoznaju potrebu za digitalizacijom svojih usluga, uključujući i rad knjižnica. Digitalni sustavi omogućuju jednostavnije pretraživanje, rezervaciju i upravljanje knjižničnom građom, čime se štedi vrijeme i povećava dostupnost informacija.

Motivacija za izradu ovog završnog rada bila je upravo želja za modernizacijom školskog sustava i olakšavanjem pristupa knjižničnim uslugama. Sustav za upravljanje školskom knjižnicom koji je izrađen u sklopu ovog rada omogućava korisnicima da putem interneta pretraže dostupne knjige, rezerviraju željene naslove i unaprijed se informiraju o stanju knjižnice, bez potrebe za osobnim dolaskom. Time se približava suvremenim potrebama korisnika i potiče njihovu veću interakciju s dostupnim resursima.

U tradicionalnim školskim knjižnicama posudba knjiga često se obavlja ručno, pri čemu knjižničar vodi evidenciju korisnika i knjiga pomoću bilježnica ili jednostavnih obrazaca. Ovakav način rada može uzrokovati razne probleme kao što su gubitak podataka, netočno vođenje evidencije, sporost u pronalasku informacija te nemogućnost brze provjere dostupnosti knjiga.

Jedan od značajnih problema je i ograničen pristup informacija o dostupnosti knjiga za korisnike koji nisu fizički prisutni u školi. U suvremenom društvu, koje sve više teži digitalizaciji usluga, korisnici očekuju da im informacije budu dostupne jednostavno i brzo, bez obzira na njihovu lokaciju.

Cilj ovog završnog rada je izraditi sustav koji omogućuje online pregled dostupnih knjiga u školskoj knjižnici i njihovu rezervaciju putem web stranice. Na taj način korisnik može unaprijed provjeriti koje su knjige dostupne te rezervirati željeni naslov. Nakon što izvrši rezervaciju, korisnik dolazi u školu kako bi osobno preuzeo knjigu i posudio je za čitanje.

Ovim sustavom knjižnica se približava modernim standardima, poboljšava se preglednost i dostupnost informacija te se korisnicima štedi vrijeme koje bi inače potrošili dolaskom samo radi provjere dostupnosti knjiga.

# TEORIJSKA PODLOGA

Za izradu sustava za upravljanje školskom knjižnicom korišten je skup tehnologija koje zajedno omogućuju izradu dinamične, funkcionalne i vizualno privlačne web aplikacije. Svaka tehnologija ima svoju posebnu ulogu u procesu izrade sustava i omogućava ostvarivanje specifičnih funkcionalnosti, od izgradnje korisničkog sučelja do upravljanja bazom podataka.

## HTML, CSS i JavaScript

**HTML** (*eng. HyperText Markup Language*) osnovni je građevni blok svake web stranice. Njegova glavna funkcija je strukturirati sadržaj stranice, definirajući elemente poput zaglavlja, odlomka, tablica, obrazaca i poveznica. Bez HTML-a web stranice ne bi mogle sadržavati organizirani tekst, slike, videozapise ili obrasce koji omogućuju interakciju s korisnikom. HTML omogućuje značenjsko označavanje sadržaja, čineći ga čitljivim i za korisnike i za tražilice.

**CSS** (*eng. Cascading Style Sheets*) koristi za oblikovanje HTML elementa. Dok HTML daje strukturu stranici, CSS definira njezin izgled – boje, fontove, margine, veličine elemenata, animacije i raspored. Na taj način CSS omogućava vizualnu prilagodbu stranice potrebama korisnika i modernim dizajnerskim trendovima. Pomoću CSS-a, web stranica postaje responzivna, što znači d se automatski prilagođava različitim veličinama zaslona kao što su računala, tableti i pametni telefoni.

**JavaScript** je programski jezik koji omogućava dodavanje dinamičkih elemenata web stranicama. U kontekstu ovog projekta JavaScript je korišten za poboljšanje korisničkog iskustva, kao što je automatsko ažuriranje rezultata pretrage bez potrebe za ponovnim učitavanjem cijele stranice (tzv. Dinamička interakcija). JavaScript omogućava i obradu korisničkih unosa, validaciju obrazaca i rad s elementima DOM strukture stranice.

## PHP

PHP (*eng. Hypertext Preprocessor*) je skriptni jezik koji se izvršava na strani poslužitelja. Osnovna uloga PHP-a je stvaranje dinamičkog sadržaja, komunikacija s bazom podataka te obrada korisničkih zahtjeva. PHP je široko korišten zbog svoje jednostavnosti, brzine i velike kompatibilnosti s bazama podataka, osobito MySQL-om.

PHP kod se nalazi između korisnika i baze podataka. Kada korisnik unese podatke ili pošalje zahtjev (npr. pretragu), PHP ga obrađuje, komunicira s bazom podataka i vraća rezultate na pregledan način.

## XAMPP

XAMPP (*eng. Cross-Platform, Apache, MySQL, PHP, Perl*) je besplatan softverski paket koji omogućava lokalno postavljanje web poslužitelja. Ime XAMPP sastavljeno je od početnih slova glavnih komponenti: Apcache (poslužitelj), MySQL (baza podataka), PHP i Perl (programski jezici).

XAMPP simulira pravo serversko okruženje na lokalnom računalu, što programerima omogućuje da testiraju i razvijaju svoje aplikacije prije nego što ih učine dostupnima korisnicima na internetu.

## MySQL

MySQL je sustav za upravljanje relacijskom bazom podataka (*eng. RDBMS – Relational Database Menagment System*). Omogućuje pohranu velikih količina podataka u tablice koje su međusobno povezane i omogućuju brzo pretraživanje, ažuriranje i organizaciju informacija.

MySQL koristi standardni jezik SQL (eng. Structed Query Language)za komunikaciju s podatcima.

MySQL baze su popularne u svijetu zbog svoje brzine, stabilnosti i mogućnosti besplatnog korištenja u otvorenim projektima. Mnoge velike aplikacije i web stranice koriste MySQL upravo zbog njegove pouzdanosti.

## Modeli baze podataka

Model baze podataka predstavlja konceptualni okvir prema kojem se strukturiraju podatci u bazi. Odabir modela ima veliki utjecaj na funkcionalnost, brzinu i jednostavnost korištenja baze.

U nastavku ću objasniti neke najpoznatije modele baza podataka:

### Hijerarhijski model

U hijerarhijskom modelu podatci su organizirani poput stabla. Svaki roditeljski zapis može imati više djece, ali svako dijete ima samo jednog roditelja. Ovaj model danas se rijetko koristi jer je teško prilagodljiv kompleksnijim odnosima među podatcima.

Primjer hijerarhijskog modela je struktura mapa i datoteka na računalu.

Slika na kojoj se prikazuje dijagram, skeč, dizajn

Opis je automatski generiran

Slika 1 Hijerarhijski model

### Mrežni model

Mrežni model je sličan hijerarhijskom, ali omogućuje da jedan zapis ima više nadređenih i podređeni zapisa. Time je omogućeno složenije povezivanje podataka. Koristio se u ranijim fazama razvoja baze podataka, ali danas je gotovo u potpunosti zamijenjen relacijskim modelom.

Slika na kojoj se prikazuje dijagram, skeč, crta, tehničko crtanje

Opis je automatski generiran

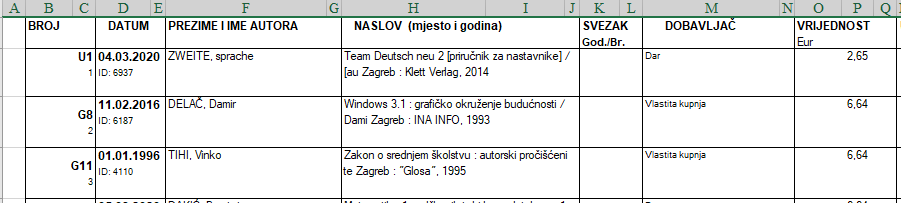
Slika 2 Mrežni model

### Plošni model

Plošni (ili jednostavni) model predstavlja podatke u obliku jedne tablice. Najčešće se koristi kod manjih baza i za demonstraciju osnovnih principa.

Primjer je Excel tablica u kojoj se svi podatci nalaze u jednom radnom listu.

U ovom projektu korištena je baza slična plošnom modelu jer se svi podatci nalaze unutar jedne tablice koja se pretražuje i filtrira.



Slika 3 Plošni model

### Objektni model

Objektni model se temelji na principima objektno-orijentiranog programiranja. Podaci se pohranjuju kao objekti koji sadrže kompleksne ili multimedijske podatke, poput slika, zvučnih zapisa, videozapisa, dokumenata te podataka koji zahtijevaju naprednu obradu i pohranu.

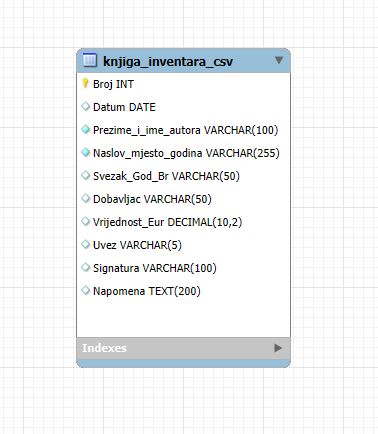
### Relacijski model

Relacijski model je danas najrašireniji i najčešće korišteni model baze podataka. Temelji se na organizaciji podataka u tablice (relacije) koje su međusobno povezane.

Svaka tablica ima redove (zapise) i stupce (atribute).Ključne značajke relacijskog modela uključuju:

* Jednostavne primarne ključeve
* Odnose među tablicama poput stranih ključeva
* Izbjegavanje ponavljanja podataka (redundancije)
* Mogućnost jednostavne nadogradnje baze

U ovom projektu korišten je relacijski model jer omogućuje pouzdanu, fleksibilnu i učinkovitu organizaciju podataka o knjigama. Kroz SQL naredbe omogućena je jednostavna manipulacija podatcima: unos, izmjena i brisanje.



Slika 4 Relacijski model

# TEHNIČKO IDEJNO RJEŠENJE

Za izradu sustava za upravljanje školskom knjižnicom korišten je skup modernih web-tehnologija i sustava za upravljanje bazom podataka. Sustav omogućuje pregled dostupnih knjiga te njihovu online rezervaciju, a tehničko rješenje obuhvaća frontend, backend, bazu podataka i lokalni razvojni server.

**Frontend tehnologije** odnose se na dio web stranice ili aplikacije koji je vidljiv korisniku. To uključuje sve što korisnik vidi i s čime može izravno komunicirati.

**Backend tehnologije** su kako bi rekli „iza kulisa“ i odgovorne su za obradu podataka, komunikaciju s bazom podataka i logiku sustava. Korisnik ovaj dio ne vidi, ali je ključan za funkcionalnost aplikacije.

## Frontend tehnologije

U korisničkom sučelju korišteni su sljedeći jezici:

**HTML** je korišten za izradu osnovne strukture web stranice – oblikovanje stranica za početni opis knjižnice i stranicu za pretraživanje knjiga. Omogućio je postavljanje elemenata kao što su zaglavlja, kartice za navigaciju, tekstualna polja za pretragu i gumbi.

**CSS** je služio za oblikovanje i vizualno uređivanje stranice te za stilizaciju. Pomoću njega definiran je izgled početne stranice, definiranje izgleda alatne trake, kartica, rezultata pretrage, pozadina, raspored elemenata i prilagodba stranice različitim uređajima.

**JavaScript** je upotrijebljen kako bi omogućio interaktivno ponašanje stranice, kao što je filtriranje rezultata pretrage i dinamički prikaz podataka bez potrebe za ponovnim učitavanjem stranice.

## Backend tehnologije

**PHP** je korišten za povezivanje web stranice s MySQL bazom podataka. Kada korisnik unese ime autora, naslov knjige ili bilo koji drugi kriterij pretrage, PHP obrađuje te podatke, šalje ih u bazu, dohvaća rezultate i prikazuje ih korisniku na stranici.

## Baza podataka

**MySQL** je korišten za izradu baze podataka koji sadrže podatke o knjigama u knjižnici. Kreirana je tablica koja sadrži informacije poput imena i prezimena autora te naslova, mjesta izdavanja i godine izdavanja knjige.

Prilikom pretraživanja korisnički upit filtrira podatke iz ove tablice i prikazuje korisniku samo tražene knjige koje su dostupne.

Baza je lokalno postavljena preko **XAMPP-a**, koji sadrži i MySQL i Apache poslužitelj.

U bazi se nalazi jedna glavna tablica pod nazivom knjiga\_inventara\_csv, koja sadrži sljedeće informacije:

* ID – jedinstveni identifikator zapisa (primarni ključ)
* Prezime i ime autora – autor knjige
* Naslov\_Mjesto\_Godina – kombinirani podatak o naslovu knjige, mjestu izdavanja i godini izdavanja
* Vrijednost knjige – financijska vrijednost knjige (euro)
* Dostupnost – prikazuje je li knjiga trenutno dostupna za posudbu
* Napomena – dodatne informacije o knjizi

Korištena je relacijska baza podataka, jer omogućuje organizaciju podataka u obliku tablica (relacija), gdje svaki zapis ima jasno definiranu strukturu.

Relacijski model odabran je zbog svoje jedinstvenosti, preglednosti i mogućnosti povezivanja podataka, iako je u ovom projektu korištena samo jedna tablica.

## Lokalno razvojno okruženje

Za lokalno izvođenje i testiranje sustava korišten je **XAMPP**, on služi za lokalno pokretanje web stranice i baze podataka. Omogućio je rad za upravljanje školskom knjižnicom na mom računalu, gdje sam mogao testirati funkcionalnosti web stranice, provjeravati pretragu knjiga, rezervaciju i komunikaciju s bazom podataka.

## Način rada sustava

Nakon što korisnik posjeti web stranicu, dočekuje ga početna stranica s opisom knjižnice i alatna traka s navigacijom. Klikom na karticu „Pretraga“, otvara se prozor u kojem korisnik unosi pojam za pretragu (ime autora, naslov, godina itd.). Nakon klika na gumb „Pretraži“, aktivira se PHP skripta koja šalje SQL upit prema bazi, filtrira rezultate i prikazuje korisniku sve odgovarajuće knjige uz podatke o dostupnosti. Ako je knjiga dostupna, korisnik može izvršiti rezervaciju online, a zatim osobno doći u školsku knjižnicu po knjigu.

Ovakvim tehničkim rješenjem omogućeno je jednostavno upravljanje knjižničnim fondom i unaprijeđeno korisničko iskustvo, čime se školsku knjižnicu približava digitalnim standardima današnjeg doba.

# ARHITEKTURA BAZE PODATAKA

Baza podataka predstavlja organizirani skup podataka koji omogućuje učinkovito upravljanje i pristup informacijama. U kontekstu ovog završnog rada, baza podataka koristi se za pohranu podataka o knjigama koje se nalaze u školskoj knjižnici, uključujući informacije o autorima, naslovima, godinama izdanja, dostupnosti i slično.

Arhitektura baze podataka može se promatrati kroz 3 osnovne razine: logičku, fizičku i konceptualnu. Ove razine omogućuju odvajanje pogleda korisnika od tehničke implementacije baze podataka.

## Logička razina

Logička razina prikazuje način na koji su podatci organizirani i međusobno povezani u samoj bazi. Na ovoj razini definiraju se tablice, njihova polja (atributi), tipovi podataka, primarni i strani ključevi te odnosi među tablicama.

U ovom projektu logička razina obuhvaća tablicu knjiga\_inventara\_csv, koja sadrži sljedeće atribute:

* Broj
* Datum
* Prezime i ime autora
* Naslov\_mjesto\_godina
* Svezak\_God\_Br
* Dobavljač
* Vrijednost (euro)
* Uvez
* Signatura
* Napomena

Svaki atribut u ovoj tablici predstavlja jednu jedinstvenu knjigu te omogućuje filtriranje i pretragu prema tim kriterijima.

## Fizikalna razina

Fizikalna razina odnosi se na način koji su podatci fizički pohranjeni na disku. U slučaju ove baze, podatci se spremaju u MySQL sustavu koji brine o strukturiranom zapisu podataka na lokalnom poslužitelju (putem XAMPP-a).

Na ovoj razini brigu vodi **DBMS** (Sustav za upravljanje bazom podataka), koji upravlja:

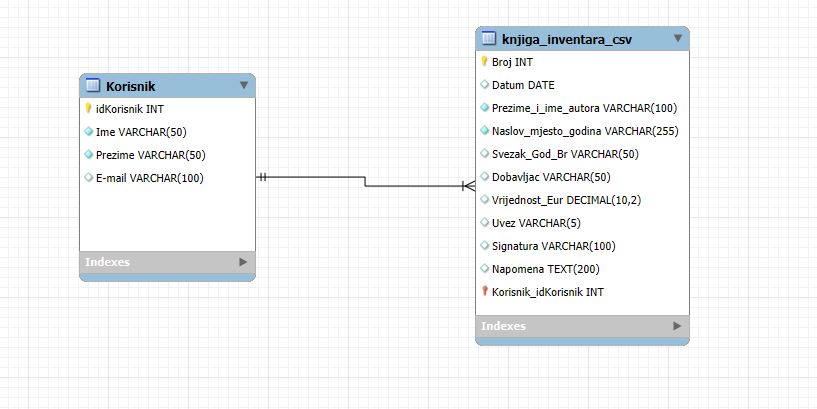
* Načinom pohrane podataka
* Korištenjem memorije
* Indeksima za bržu pretragu
* Pristupom podatcima i njegovom zaštitom

Fizikalni prikaz korisnik ne vidi direktno, ali on osigurava pouzdanu i sigurnu pohranu podataka.

## Konceptualna razina

Konceptualna razina baze podataka predstavlja apstraktan model koji opisuje koje podatke sustav treba pohranjivati, bez razmišljanja o načinu pohrane ili detaljnoj tehničkoj implementaciji. Ova razina prikazuje sustav iz perspektive korisnika i omogućuje razumijevanje veza među entitetima bez potrebe za tehničkim znanjem o strukturama baze.

U kontekstu ovog završnog rada , konceptualna razina prikazana je kroz ER dijagram (Entitet – Relacija) koji obuhvaća entitet „Knjiga“ s pripadajućim atributima. Svaki atribut u dijagramu predstavlja određeno svojstvo knjige koje se evidentira u sustavu.



Slika 5 ER Dijagram

Na slici je prikazan ER dijagram koji prikazuje dvije glavne tablice baze: Korisnik i knjiga\_inventara\_csv. Svaki korisnik može biti povezan s više knjiga, dok svaka knjiga može biti vezana za samo jednog korisnika putem stranog ključa Korisnik\_idKorisnik. Ovakva struktura omogućuje praćenje koje korisnik je rezervirao ili posudio određenu knjigu.

## Sažetak arhitekture baze podataka

Arhitektura baze podataka u ovom projektu jasno prikazuje kako su podatci o knjigama strukturirani, pohranjeni i logički povezani. Logička razina definira strukturu tablice i atribute, fizikalna opisuje način pohrane podataka u MySQL sustav putem XAMPP-a, dok konceptualna razina prikazuje odnose među entitetima pomoću ER dijagrama.

Ovakva višeslojna arhitektura osigurava jednostavno proširivanje baze, preglednost podataka i mogućnost dodatne funkcionalnosti, poput praćenja povijesti posudbi ili navođenja korisničkih računa. Struktura baze temelj je cijelog sustava i ključna komponenta u uspješnoj digitalizaciji knjižničnog poslovanja.

# IZRADA BAZE PODATAKA

U ovom poglavlju detaljno je opisana praktična izrada baze podataka za upravljanje školskom knjižnicom. Kroz ovaj odjeljak prikazani su svi koraci izrade baze podataka: od kreiranja veze, bez i tablica, definiranje tipova podataka i ograničenja, pa sve do unosa i pregleda podataka. Uz svaki korak prikazane su odgovarajuće slike i dijelovi koda kako bi se jasno prikazao tijek rada.

## Kreiranje veze

Izradu baze podataka započinjemo tako da otvorimo MySQL Workbench i povežemo se na mrežu tog programa. Na mrežu se povezujemo tako da u programu kreiramo vezu (New Connection) te ispunimo sve tražene podatke kao što su ime veze, konekcijska metoda (najčešće „Standard (TCP/IP)“), hostname (obično localhost), korisničko ime (najčešće root) i ostale potrebne stavke.

Slika na kojoj se prikazuje tekst, snimka zaslona, zaslon, softver

Opis je automatski generiran

Slika 6 Kreiranje veze u MySQL-u

Međutim, posebno moramo obratiti pažnju na portove. Port koji koristimo mora odgovarati portu na kojem se trenutno izvršava MySQL server u XAMPP-u. Po standardnim postavkama, MySQL koristi port 3306, no ako je taj port zauzet (npr. od druge baze), moguće ga je promijeniti u XAMPP-u, ali tada taj isti port moramo navesti i prilikom povezivanja u MySQL Workbenchu.

## Kreiranje baze podataka

Nakon otvaranja SQL prozora koristi se naredba za stvaranje baze podataka:

CREATE DATABASE SkolskaKnjiznica;

USE SkolskaKnjiznica;

* Naredbom CREATE DATABSE stvara se prazna baza s imenom SkolskaKnjiznica, koja će sadržavati sve tablice potrebne za rad sustava.
* Naredba USE nam služi za odabir baze podataka s kojom trenutno radimo. Pomoću ove naredbe možemo pozvati bilo koju stvorenu bazu podataka.

## Kreiranje tablice

Nakon što je baza definirana, kreira se glavna tablica koja sadrži sve informacije o našim knjigama.

Za kreiranje tablica koristimo naredbu CREATE TABLE koja nam omogućuje stvaranje tablice s jasno definiranim poljima i tipovima podataka:

CREATE TABLE knjiga\_inventara\_csv (

Broj INT AUTO\_INCREMENT PRIMARY KEY,

Datum DATE,

Prezime\_i\_ime\_autora VARCHAR(100) NOT NULL,

Naslov\_mjesto\_godina VARCHAR(255) NOT NULL,

Svezak\_God\_Br VARCHAR(50),

Dobavljac VARCHAR(50),

Vrijednost\_Eur DECIMAL(10,2),

Uvez VARCHAR(5),

Signatura VARCHAR(100),

Napomena TEXT

);

Tablica sadrži:

* BROJ – primarni ključ, predstavlja jedinstveni identifikator svake knjige unutar knjižnice
* DATUM – datu unosa knjige u knjižnicu
* PREZIME\_I\_IME\_AUTORA – ime i prezime autora knjige
* NASLOV\_mjesto\_godina – kombinirani podatak koji sadrži naslov knjige, mjesto izdavanja i godinu izdavanja
* SVEZAK\_God\_Br – dodatni podatci o svesku, godini i broju izdanja
* DOBAVLJAC – način nabave knjige
* VRIJEDNOST\_Eur – cijena knjige u eurima
* UVEZ – dostupnost knjige
* SIGNATURA – šifra knjige prema knjižničnom sustavu klasifikacije.
* NAPOMENA – slobodno polje za dodatne podatke, bilješke ili posebne oznake

## Tipovi podataka

Tipovi podataka u ovoj tablici su pažljivo odabrani:

* Tipovi podataka u ovoj tablici su pažljivo odabrani:
* VARCHAR (x) – Koristi se za spremanje kombinacije slova i brojeva
* DATE – Koristi se za pohranu podataka u formatu YYYY-MM-DD (Godina – Mjesec – Dan). Pogodan je za zapisivanje datuma unosa knjige
* TEXT – Ovaj tip podatka omogućuje pohranu većih količina teksta, stoga se koristi na mjestima gdje je potreban malo duži opis nečega
* DECIMAL (10, 2) – Ovaj tip koristi se za pohranu decimalnih brojeva (npr. cijena 6,64€). Prvi broj (10) označava ukupni broj znamenki, a drugi (2) broj znamenki iza decimalne točke

Svaki tip je odabran prema logičkoj svrsi polja.

## Ograničenja i pravila

Korištena su osnovna ograničenja:

* PRIMARY KEY osigurava jedinstvenost svake knjige
* NOT NULL jamči da polja ne mogu ostati prazna
* AUTO\_INCREMENT automatski dodjeljuje novi BROJ svakoj knjizi

## Ubacivanje podataka

Podatci o knjigama unose se pomoću naredbe INSERT INTO:

INSERT INTO knjiga\_inventara\_csv

(BROJ, DATUM, PREZIME\_I\_IME\_AUTORA, NASLOV\_mjesto\_godina, SVEZAK\_God\_Br, DOBAVLJAC, VRIJEDNOST\_Eur, UVEZ, SIGNATURA, NAPOMENA)

VALUES

('G245 133', '10.10.1998 ID: 3150', 'BRLIĆ-Mažuranić, Ivana', 'Priče iz davnine ; Čudnovate zgode šegrta Hlapića Vinkovci : Riječ', '1997',' ', 'Vlastita kupnja', '6,64', 'DA', '886.2 BRLI P', ' ');

('G550 278', '27.09.2016 ID: 6672', 'BLAŽEVIĆ, Nevenka', 'Guten appetit, njemački za III. razred ugostiteljs Zagreb : Školska knjiga, 2001',' ', 'Dar', '6,64', 'DA', '811.112.2 BLA g', 'Otpisano 04.12.2023.')

## Pregled podataka u bazi

Za dohvaćanje i prikaz podataka iz baze koristimo naredbu SELECT \* FROM:

SELECT \* FROM knjiga\_inventara\_csv;

Slika na kojoj se prikazuje tekst, izbornik, dokument, Font

Opis je automatski generiran

Slika 7 Ispis podataka

Rezultat koda prikazuje sve zapise koji su uneseni u tablicu

## Kratki osvrt na poglavlje

U ovome poglavlju prikazana je kompletna izrada baze podataka za školsku knjižnicu – od stvaranja baze, definiranja strukture tablice, tipova podataka, ograničenja, unosa sadržaja i testiranja. Tako pripremljena baza omogućuje učinkovito povezivanje s web stranicom i provedbu svih funkcionalnosti sustava.

# IZRADA WEB STRANICE

U sklopu završnog rada izrađena je moderna i funkcionalna web stranica za školsku knjižnicu, koja korisnicima omogućuje pregled knjižnice, pretragu knjiga te rezervaciju putem obrasca.

Za izradu su korištene web tehnologije: HTML (struktura), CSS (stilizacija) i JavaScript (interaktivnost).

## Izrada strukture stranice (HTML)

Za izradu korisničkog sučelja sustava korišten je HTML5 (najnovija verzija jezika HTML), temeljni jezik za oblikovanje strukture web stranice. Svaka funkcionalna cjelina sustava knjižnice implementirana je kroz vlastitu .html datoteku, koje su međusobno povezane putem navigacijske trake.

Web stranica se sastoji od sljedećih stranica:

* Index.html – početna stranica s opisom knjižnice
* Pretraga.html – Pretraga knjige iz baze podataka
* Rezervacija.html – Obrazac za rezervaciju knjige

### Navigacija i raspored

Na svim stranicama nalazi se jednostavna navigacijska traka koja omogućuje brzo kretanje kroz stranicu. Navigacija se oblikuje pomoću <nav> elementa i neuređene liste <ul>, s linkovima <a>:

<nav>

<ul>

<li><a href="index.html">Početna</a></li>

<li><a href="pretraga.html">Pretraga knjiga</a></li>

<li><a href="rezervacija.html">Rezervacija knjige</a></li>

</ul>

</nav>

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, logotip

Opis je automatski generiran

Slika 8 Navigacijska traka

## Opis početne stranice – index.html

Ova stranica prikazuje osnovne informacije o školskoj knjižnici – uključuje veliki naslov, odjeljak s tekstom za opis knjižnice te slike prostora knjižnice. Sve je strukturirano pomoću <header>, <section>, <h1>, <p>, i <img> elemenata.

<section id="o-knjiznici">

<h2>O našoj knjižnici</h2>

<p>

Naša školska knjižnica je mjesto znanja, istraživanja i inspiracije…

</p>

</section>

<section id="galerija">

<h2>Izgled naše knjižnice</h2>

<div class="galerija-slika">

<img src="Slike knjiznice/knjiznica1.jpeg" alt="Knjižnica 1" onclick="openLightbox(this.src)">

….

</div>

<div id="lightbox" class="lightbox" onclick="closeLightbox()">

<img id="lightbox-img" src="" alt="Povećana slika">

</div>

</section>

**Slika na kojoj se prikazuje tekst, snimka zaslona, u dvorani, umjetničko djelo

Opis je automatski generiran**

Slika 9 Galerija slika

## Stranica za pretragu knjiga – pretraga.html

Ova stranica omogućuje korisnicima jednostavno pretraživanje dostupnih knjiga prema imenu autora, čime im se olakšava pronalazak željene literature u knjižnici.

Sadrži:

* Input polje za unos ključne riječi
* Gumb za pretragu
* HTML tablicu koja prikazuje rezultate

<section id="pretraga">

<h2>Pretražite naš katalog knjiga</h2>

<div class="search-container">

<input type="text" id="searchInput" placeholder="Unesite ime autora ili naslov knjige">

<button id="searchButton">Pretraži</button>

</div>

<div class="table-container">

<table id="booksTable" class="display">

<thead>

<tr>

<th>AUTORA</th>

<th>NASLOV\_mjesto\_godina</th>

<th>SIGNATURA</th>

<th>DOBAVLJAČ</th>

<th>DATUM</th>

</tr>

</thead>

<tbody>

<!--Tu se učitavaju podatci iz tablice -->

</tbody>

</table>

</div>

</section>

### Opis HTML tablice

Tablica booksTable prikazuje sve rezultate koje sustav pronađe prema korisnikovom unosu.

Tablica je napravljena pomoću <table>, <thead> i <tbody> elemenata.

Svaki redak tablice (<tr>) generira se pomoću JavaScript-a na temelju podataka koje vraća PHP skripta.

Kada korisnik klikne na gumb „Pretraži“, poziva se funkcija dohvatiKnjige() (JavaScript) koja šalje zahtjev serveru, a rezultati se umeću direktno u HTML tablicu.

Slika na kojoj se prikazuje tekst, snimka zaslona, broj, softver

Opis je automatski generiran

Slika 10 HTML tablica

### Filtracija knjiga po autoru

Korisnik može unijeti dio imena autora ili prezimena u input polje. Taj pojam se prosljeđuje na server (PHP skripta search\_books.php) i koristi se u HTML upitu:

if (!empty($searchTerm)) {

$query .= " AND (`PREZIME I IME AUTORA` LIKE :searchTerm)";

}

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, broj

Opis je automatski generiran

Slika 11 Filtriranje podataka

## Rezervacija knjige – rezervacija.html

Stranica rezervacija.html ne koristi običnu formu, već sadrži ugrađeni JotForm obrazac pomoću <iframe> taga.

JotForm je besplatna online platforma za brzo i jednostavno kreiranje različitih obrazaca, kao što su kontakt forme, obrasci za prijavu, ankete, obrasci za rezervaciju i mnogi drugi. Omogućuje korisnicima da bez programerskog znanja izrade funkcionalne obrasce putem intuitivnog „drag-and-drop“ sučelja.

U ovom slučaju koristi se za rezervaciju knjiga, gdje korisnik putem jednostavnog obrasca može unijeti osnovne podatke (ime, prezime, e-mail te knjigu koju želi rezervirati) i time pokrenuti proces rezervacije.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, web-stranica

Opis je automatski generiran

Slika 12 Rezervacija knjige

Kada korisnik pritisne na gumb „Submit“ (*hrv. Pošalji*), podatci se spremaju na JotForm servis te onda, administratoru tog obrasca, dolazi poruka na e-mail sa korisnikovim ispunjenim podatcima gdje se onda može vidjeti koju knjigu je on rezervirao i time izvršiti i sačuvati njegovu rezervaciju.

Osim administratora, i korisnik prima automatsku e-mail potvrdu gdje može vidjeti svoje unesene podatke, što služi kao dokaz da je rezervacija uspješno zaprimljena.

Slika na kojoj se prikazuje tekst, softver, web-stranica, web-mjesto

Opis je automatski generiran

Slika 13 E-mail potvrde

### Osvrt na HTML dio

HTML kod čini osnovu cijelog sustava i omogućuje korisniku da vizualno i funkcionalno pristupi svim dijelovima aplikacije.

Korištenjem forme za pretragu i tablice za prikaz rezultata omogućena je jednostavna i učinkovita interakcija s bazom knjiga, dok rezervacijski obrazac (JotForm) nudi praktično rješenje za korisniče zahtjeve bez dodatnog programiranja na strani servera.

## Stilizacija stranice (CSS)

Za vizualni izgled i raspored elemenata web stranice korišten je CSS. Svi HTML dokumenti koriste istu vanjsku datoteku styles.css, što omogućuje dosljedan i moderan dizajn kroz cijeli sustav knjižnice.

### Osnovni stil tijela stranice

Stranica koristi jednostavnu i čistu pozadinu sa svijetlom temom, čime se povećava čitljivost teksta i preglednost elemenata.

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

color: #333;

}

* Font – moderan i čitljiv (Segoe UI)
* Pozadina – svijetlosiva (#f4f4f4)
* Tekst – tamnosiva za dobar kontrast (#333)

### Navigacijska traka

Navigacija je prikazana kao horizontalna traka s crvenom podlogom. Elementi unutar nje su raspoređeni pomoću flex rasporeda i sadrže hover efekte za bolji vizualni dojam pri prelasku mišem.

nav ul {

display: flex;

justify-content: center;

list-style: none;

padding: 10px 0;

background: #d32f2f;

border-radius: 10px;

}

nav ul li a {

color: white;

padding: 12px 20px;

text-decoration: none;

transition: 0.3s;

}

nav ul li a:hover {

background-color: #ff7961;

border-radius: 5px;

transform: scale(1.1);

}

* Boja trake – tamnocrvena (#d32f2f)
* Boja na hover – svjetlija crvena s efektom skaliranja (#ff7961)

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, logotip

Opis je automatski generiran

Slika 14 Navigacijska traka s hover efektom

### Sekcije i sadržaj

Sadržaj na stranicama organiziran je u <section> blokove, koji su vizualno odvojeni i imaju sjenu, padding, zaobljene rubove i laganu animaciju.

section {

background: white;

padding: 30px;

margin: 30px;

border-radius: 10px;

box-shadow: 0px 2px 5px rgba(0,0,0,0.2);

}

* Bijela pozadina – čista i profesionalna
* Sjena – stvara osjećaj dubine
* Radius – 10px za zaobljene rubove

### Gumbi i forme

Gumbi na web stranici imaju crvenu pozadinu, bijeli tekst i glatke prijelaze pri prelasku mišem. Forme su jednostavne, ali jasno stilizirane.

button {

background: #e57373;

color: white;

border: none;

padding: 10px 15px;

margin-top: 10px;

cursor: pointer;

transition: 0.3s;

border-radius: 5px;

}

button:hover {

background: #ff7961;

transform: translateY(-3px);

}

* Efekt pomaka (translateY) – pomiče gumb malo prema gore te daje dojam klikabilnosti
* Zaobljeni rubovi – čine suvremen dizajn

### Galerija slika i Lightbox prikaz

Galerija je oblikovana pomoću flex rasporeda između slika (gap) i vizualnim efektima na hover. Svaka slika ima zaobljene rubove i sjenku

.galerija-slika {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 20px;

padding: 20px 0;

}

Kada korisnik klikne na sliku, otvara se lightbox – tamna pozadina preko cijelog ekrana s povećanom slikom. CSS definira središnji prikaz slike i sakrivanje dok nije aktivno.

.lightbox {

display: none;

position: fixed;

z-index: 1000;

top: 0; left: 0;

width: 100%; height: 100%;

background-color: rgba(0,0,0,0.8);

justify-content: center;

align-items: center;

}

Slika na kojoj se prikazuje tekst, knjiga, komoda s knjigama, u dvorani

Opis je automatski generiran

Slika 15 Slika sa lightbox efektom

### Prilagodljiv sadržaj

Iako je stranica primarno napravljena za desktop pregled, raspored elemenata (npr. felx navigacija, margine i padding) omogućuju zadovoljavajuće prikazivanje i na manjim ekranima

### Sažetak CSS dijela

Datoteka styles.css omogućila je stvaranje preglednog i vizualno privlačnog korisničkog sučelja.

Dizajn je jednostavan, ali moderan, i uz pomoć animacija i interaktivnih efekata stvara ugodno korisničko iskustvo.

Zahvaljujući odvojenom CSS-u, održavanje izgleda stranice je jednostavno, a kasnije se nadogradnja može lako implementirati.

## Interaktivnost stranice (JavaScript)

Za dodavanje interaktivnosti i animacija na web stranicu korišten je JavaScript, čiji se kod nalazi u vanjskoj datoteci script.js.

JavaScript omogućuje da stranice ne budu statične – već da reagiraju na korisničke akcije kao što su klikovi, prijelazi i učitavanje stranice.

U datoteci koriste se četiri ključne funkcionalnosti:

### Animacijska sekcija prilikom učitavanja stranice

Prilikom učitavanja stranice sve sekcije (<section>) postepeno se prikazuju uz tzv. fade-in efekt. To se postiže tako da se početni stil svakog elementa postavlja na nevidljivo i pomaknuto prema dolje, a zatim kroz animaciju vraća u početni položaj i postaje vidljiv.

document.querySelectorAll("section").forEach(section => {

section.style.opacity = 0;

section.style.transform = "translateY(30px)";

setTimeout(() => {

section.style.transition = "all 1.2s ease";

section.style.opacity = 1;

section.style.transform = "translateY(0)";

}, 300);

});

Ova animacija daje stranici moderan izgled i stvara dojam uglađenog prikaza sadržaja pri svakom otvaranju stranice.

### Mekana navigacija između stranica

Umjesto trenutnog skoka na drugu stranicu, koristi se mekani prijelaz – kada korisnik klikne na link iz navigacije, tijelo stranice se polako „ugasi“ (opacity 0), a tek onda se stvori nova stranica. Time se stvara dojam glatkog prijelaza između stranica.

document.querySelectorAll("nav ul li a").forEach(link => {

link.addEventListener("click", function (event) {

if (this.getAttribute("href").includes(".html")) {

event.preventDefault();

document.body.style.opacity = 0;

setTimeout(() => {

window.location.href = this.getAttribute("href");

}, 500);

}

});

});

Ova animacija se aktivira samo za linkove koji vode na druge .html datoteke, čime se sprječava primjena na vanjske ili posebne poveznice.

### Lightbox prikaz slika u galeriji

Ova funkcionalnost omogućuje korisnicima da klikom na sliku u galeriji otvore tu sliku u velikom prikazu preko cijelog ekrana, s tamnom pozadinom (tzv. lightbox prikaz).

Ova funkcionalnost se ostvaruje s dvije JavaScript funkcije:

1. Otvaranje slike u lightboxu:

function openLightbox(src) {

const lightbox = document.getElementById("lightbox");

const lightboxImg = document.getElementById("lightbox-img");

lightboxImg.src = src;

lightbox.style.display = "flex";

}

2. Zatvaranje lightbox prikaza:

function closeLightbox() {

const lightbox = document.getElementById("lightbox");

lightbox.style.display = "none";

document.getElementById("lightbox-img").src = "";

}

### Kratki osvrt JavaScript dijela

Iako je jednostavan, JavaScript značajno doprinosi korisničkom iskustvu. Animacije i prijelazi čine aplikaciju vizualno ugodnijom, a intuitivne radnje korisnika (klik na link, otvaranje stranice) dobivaju profesionalni izgled.

# SPAJANJE WEB STRANICE I BAZE PODATAKA

Jedna od ključnih funkcionalnosti ovog sustava za upravljanje školskom knjižnicom je mogućnost pretraživanja i prikaza podataka iz baze putem web sučelja.

Kako bi se to ostvarilo, bilo je potrebno povezivanjem web stranice s MySQL bazom pomoću PHP skripti i AJAX zahtjeva, pri čemu se koristi i jQuery biblioteka te dodatak DataTables za prikaz podataka u interaktivnoj tablici.

## Korišteni programi

* MySQL – baza podataka u kojoj su pohranjene informacije o knjigama
* PHP (pretraga\_knjiga.php) – obrađuje zahtjeve s web stranice i dohvaća podatke iz baze
* jQuery + DataTables – omogućuje slanje zahtjeva, prikaz rezultata i dodatne mogućnosti filtriranja i sortiranja
* XAMPP – lokalni poslužitelj koji pokreće i PHP i MySQL servis

## Stranica za pretragu (pretraga.html)

HTML datoteka koristi DataTables za prikaz rezultata pretrage u tablici. AJAX zahtjev se šalje prema php datoteci pretraga\_knjiga.php, a pretraga se pokreće klikom na gumb „Pretraži“ ili pritiskom tipke „Enter“.

$('#searchButton').click(function() {

table.ajax.reload();

});

$('#searchInput').keypress(function(e) {

if(e.which == 13) {

table.ajax.reload();

}

});

## Dohvat podataka pute AJAX-a

Kod koristi jQuery AJAX i serverSide: true, što znači da se pretraga izvršava na serveru:

$('#booksTable').DataTable({

"dom": 'lrtip',

"processing": true,

"serverSide": true,

"ajax": {

"url": "pretraga\_knjiga.php",

"type": "POST",

"data": function(d) {

d.searchTerm = $('#searchInput').val();

}

},

"columns": [

{ "data": "PREZIME I IME AUTORA" },

{ "data": "NASLOV\_mjesto\_godina" },

{ "data": "SIGNATURA" },

{ "data": "DOBAVLJAČ" },

{ "data": "DATUM" }

],

## PHP skripta (pretraga\_knjiga.php)

Ova skripta prima pretraženi pojam, izvršava SQL upit nad bazom i vraća podatke u JSON formatu koji DataTables može prikazati. Skripta podržava server-side filtriranje, sortiranje i ograničavanje broja redova (LIMIT).

Ključni koraci u skripti:

* Povezivanje s MySQL bazom
* Primanje POST podataka iz zahtjeva:

$searchTerm = isset($\_POST['searchTerm']) ? $\_POST['searchTerm'] : '';

* Gradnja SQL upita s filtracijom:

$filteredQuery = $conn->prepare(str\_replace(

'`PREZIME I IME AUTORA`, `NASLOV\_mjesto\_godina`, `SIGNATURA`, `DOBAVLJAČ`, `DATUM`',

'COUNT(\*)',

$query

));

if (!empty($searchTerm)) {

$filteredQuery->bindValue(':searchTerm', '%' . $searchTerm . '%');

}

$filteredQuery->execute();

$filteredRecords = $filteredQuery->fetchColumn();

* Dohvat podataka i priprema za JSON izlaz:

$response = [

"draw" => $draw,

"recordsTotal" => $totalRecords,

"recordsFiltered" => $filteredRecords,

"data" => $books

];

echo json\_encode($response);

## Sažetak poglavlja

Korištenjem PHP-a, jQueryja i AJAX tehnologija omogućeno je učinkovito povezivanje web stranice s bazom podataka.

Korisnici mogu u stvarnom vremenu pretraživati dostupne knjige bez ponovnog učitavanja stranice, a sustav koristi napredne mogućnosti DataTablesa za bolju preglednost i korisničko iskustvo.

Takva arhitektura čini sustav brzim, responzivnim i spremnim za proširenja – primjerice, povezivanje s korisničkim računalima ili automatskom evidencijom posudbi.

# ZAKLJUČAK

Tijekom izrade završnog rada na temu „Sustav za upravljanje školskom knjižnicom“, uspješno je razvijen funkcionalan web sustav koji omogućuje korisnicima pregled dostupnih knjiga u knjižnici, njihovo pretraživanje prema ključnim kriterijima, kao i mogućnost online rezervacije željene knjige putem jednostavnog obrasca.

Korištenjem modernih tehnologija kao što su HTML, CSS, JavaScript, PHP i MySQL, izrađeno je pregledno i responzivno korisničko sučelje povezano s bazom podataka. Web stranica koristi AJAX pristup za dohvat podataka bez ponovnog učitavanja stranice, čime se postiže bolje korisničko iskustvo i brža interakcija.

Dodavanjem DataTables dodatka omogućeno je sortiranje, filtriranje i pregledavanje podataka u profesionalnom formatu.

Sustav je dizajniran tako da korisniku pruža:

* Pregledan prikaz informacija o knjigama
* Intuitivnu pretragu po imenu ili prezimenu autora
* Jednostavno i brzo rezerviranje knjige putem JotForm obrasca
* Potvrdu rezervacije putem e-maila

Tijekom izrade projekta, naučeno je kako povezati frontend i backend dijelove aplikacije, upravljati bazama podataka, komunicirati s njima pomoću PHP-a te koristiti JavaScript za animacije, efekte i dinamičko prikazivanje sadržaja.

Također, kroz rad se razvila bolja sposobnost planiranja, strukturiranja koda i rješavanja tehničkih problema.

**Mogućnost nadogradnje u budućnosti**

* Korisnički računi za učenike i administratore
* Prikaz povijesti posudbi i rezervacija po korisniku
* Automatska evidencija dostupnosti nakon rezervacije
* Sustav za vraćanje knjiga i slanja podsjetnika putem e-maila

U svijetu koji se ubrzano digitalizira, ovakva rješenja predstavljaju važan korak naprijed u modernizaciji školskih sustava.

Cilj rada bio je pojednostaviti pristup knjižnici i omogućiti digitalno povezivanje učenika s informacijama, a upravo to je ovim projektom ostvareno.

# LITERATURA

[Slika 1 Hijerarhijski model 5](#_Toc197480916)

[Slika 2 Mrežni model 6](#_Toc197480917)

[Slika 3 Plošni model 6](#_Toc197480918)

[Slika 4 Relacijski model 8](#_Toc197480919)

[Slika 5 ER Dijagram 14](#_Toc197480920)

[Slika 6 Kreiranje veze u MySQL-u 15](#_Toc197480921)

[Slika 7 Ispis podataka 19](#_Toc197480922)

[Slika 8 Navigacijska traka 21](#_Toc197480923)

[Slika 9 Galerija slika 22](#_Toc197480924)

[Slika 10 HTML tablica 24](#_Toc197480925)

[Slika 11 Filtriranje podataka 25](#_Toc197480926)

[Slika 12 Rezervacija knjige 26](#_Toc197480927)

[Slika 13 E-mail potvrde 27](#_Toc197480928)

[Slika 14 Navigacijska traka s hover efektom 29](#_Toc197480929)

[Slika 15 Slika sa lightbox efektom 31](#_Toc197480930)

[1] W3Schools. HTML Tutorial. Dostupno na: <https://www.w3schools.com/html/>

[2] W3Schools. CSS Tutorial. Dostupno na: <https://www.w3schools.com/css/>

[3] W3Schools. JavaScript Tutorial. Dostupno na: <https://www.w3schools.com/js/>

[4] PHP.net. mysqli\_connect – Manual. Dostupno na: <https://www.php.net/manual/en/function.mysqli-connect.php>

[5] DataTables. Manual. Dostupno na: <https://datatables.net/manual/>

[6] Mozilla Developer Network (MDN). JavaScript Guide. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

[7] Apache Friends. XAMPP Download and Documentation. Dostupno na: <https://www.apachefriends.org/>

[8] JotForm. What is Jotform? Dostupno na: <https://www.jotform.com/help/2-what-is-jotform/>

[9] MySQL. Documentation. Dostupno na: <https://dev.mysql.com/doc/>

[10] Vertabelo. Three-Level Architecture of a Database. Dostupno na: [https://www.vertabelo.com/blog](https://www.vertabelo.com/blog%20)

[11] Wikipedia. Entity–relationship model. Dostupno na: <https://en.wikipedia.org/wiki/Entity–relationship_model>

PRILOG

Web stranice za instalaciju korištenih programa:

Visual Studio Code: <https://code.visualstudio.com/download>

XAMPP: <https://www.apachefriends.org/download.html>

MySQL Workbench: <https://dev.mysql.com/downloads/workbench/>