# Prvi Domaći Zadatak

Internet Tehnologije



PHP MySql AJAX

Dejan Čolić 260/13

Beograd 2021

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Dejan | Čolić | 0260/2013 |
| Link ka GitHub-u | https://github.com/DejanColic94/ITEH-Prvi-Domaci | |

Contents

[ITEH prvi domaći 2](#_Toc69475507)

[PHP MYSQL AJAX 2](#_Toc69475508)

[1. Pregled aplikacije 2](#_Toc69475509)

[2. Model podataka 4](#_Toc69475510)

[3. Struktura fajlova 6](#_Toc69475511)

[4. Utill 7](#_Toc69475512)

[5. Funkcionalnosti 9](#_Toc69475513)

# ITEH prvi domaći

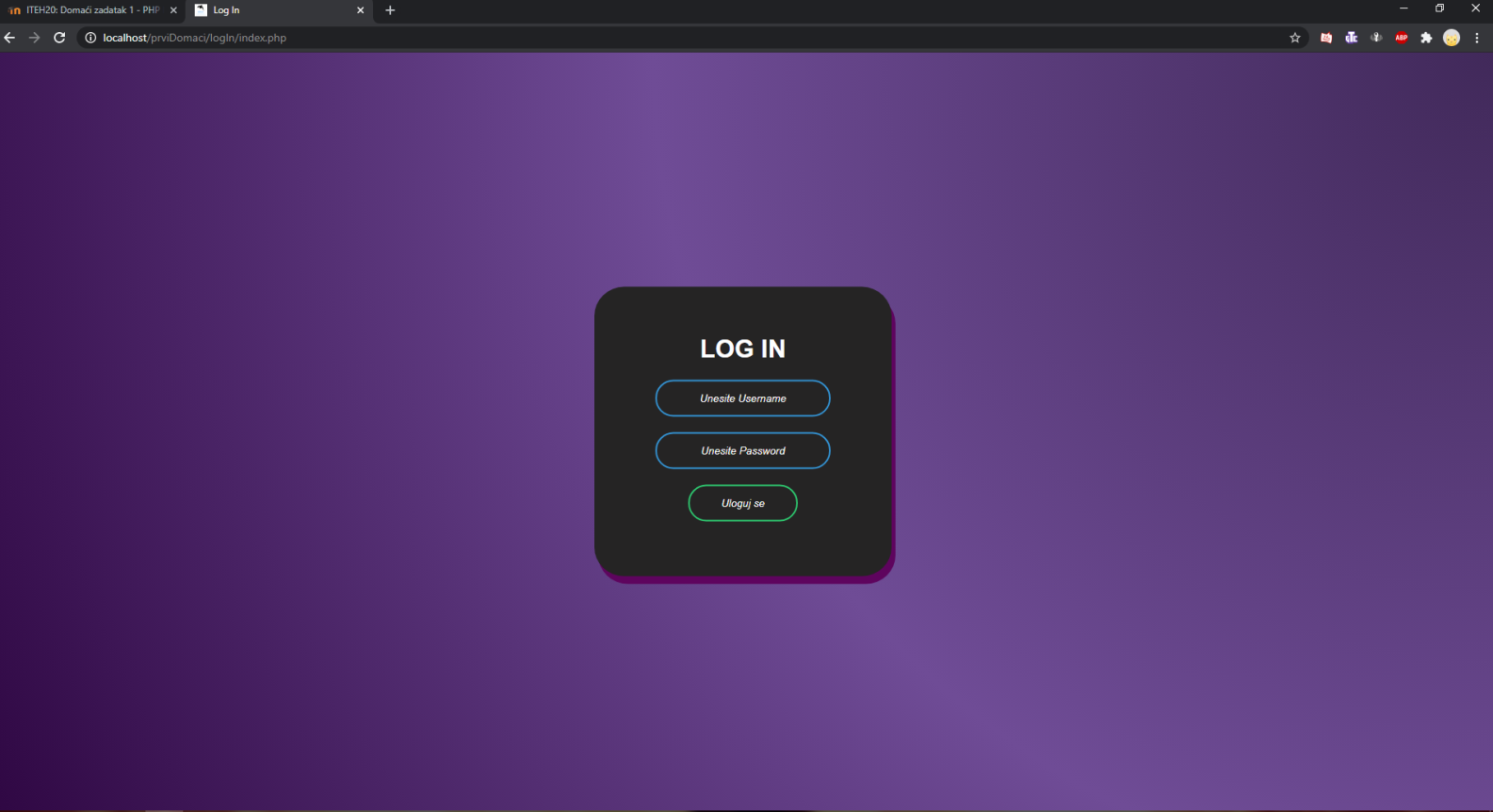
## PHP MYSQL AJAX

Cilj ove web aplikacije je da demonstrira CRUD operacije korišćenjem PHP skriptnog jezika i MySql baze podataka.

Model nad kojima se CRUD operacije izvršavaju je dosta pojednostavljen za potrebe ovog rada.

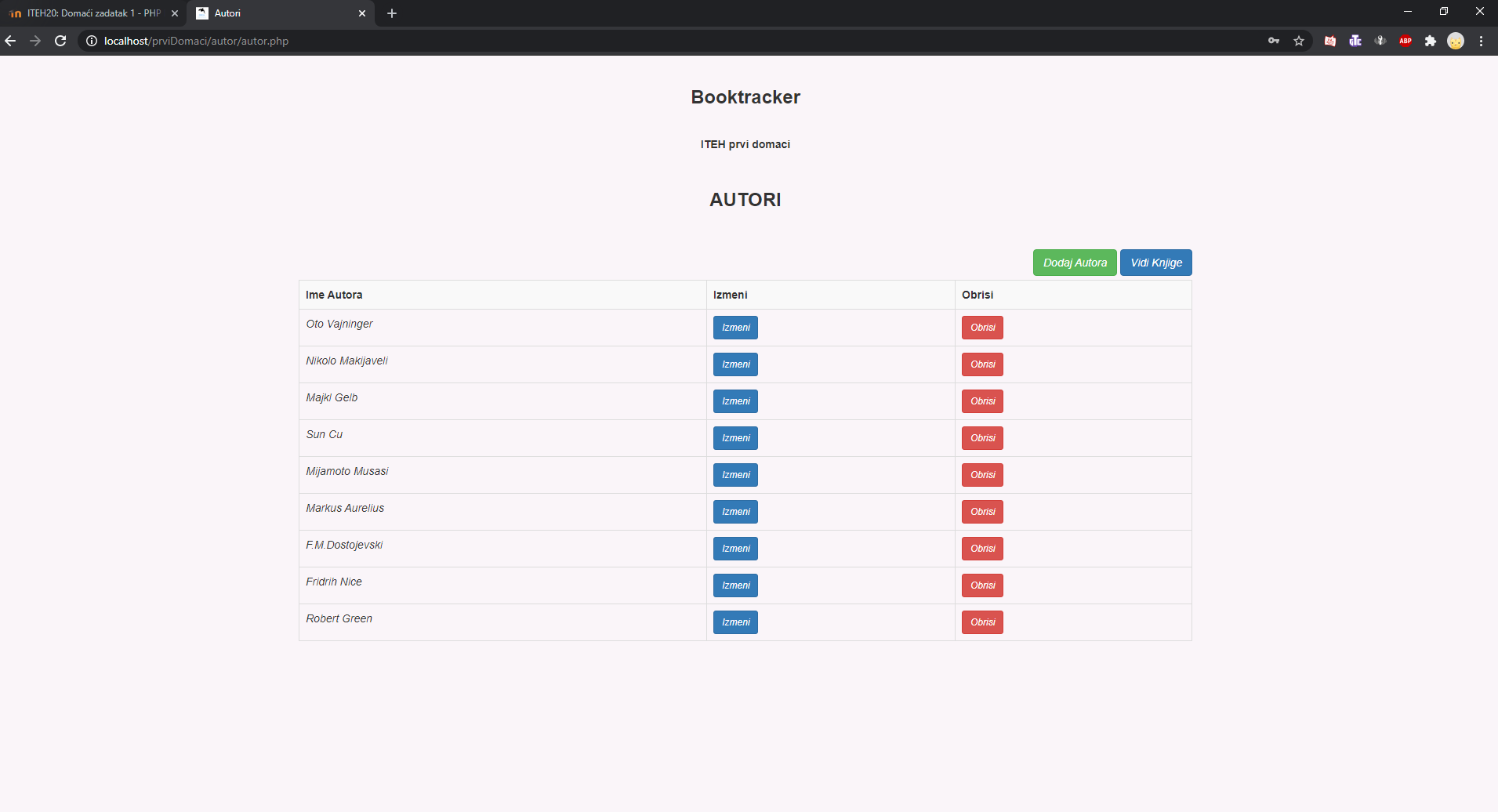
### 1. Pregled aplikacije

Prvi ekran koji vidimo je log in stranica.



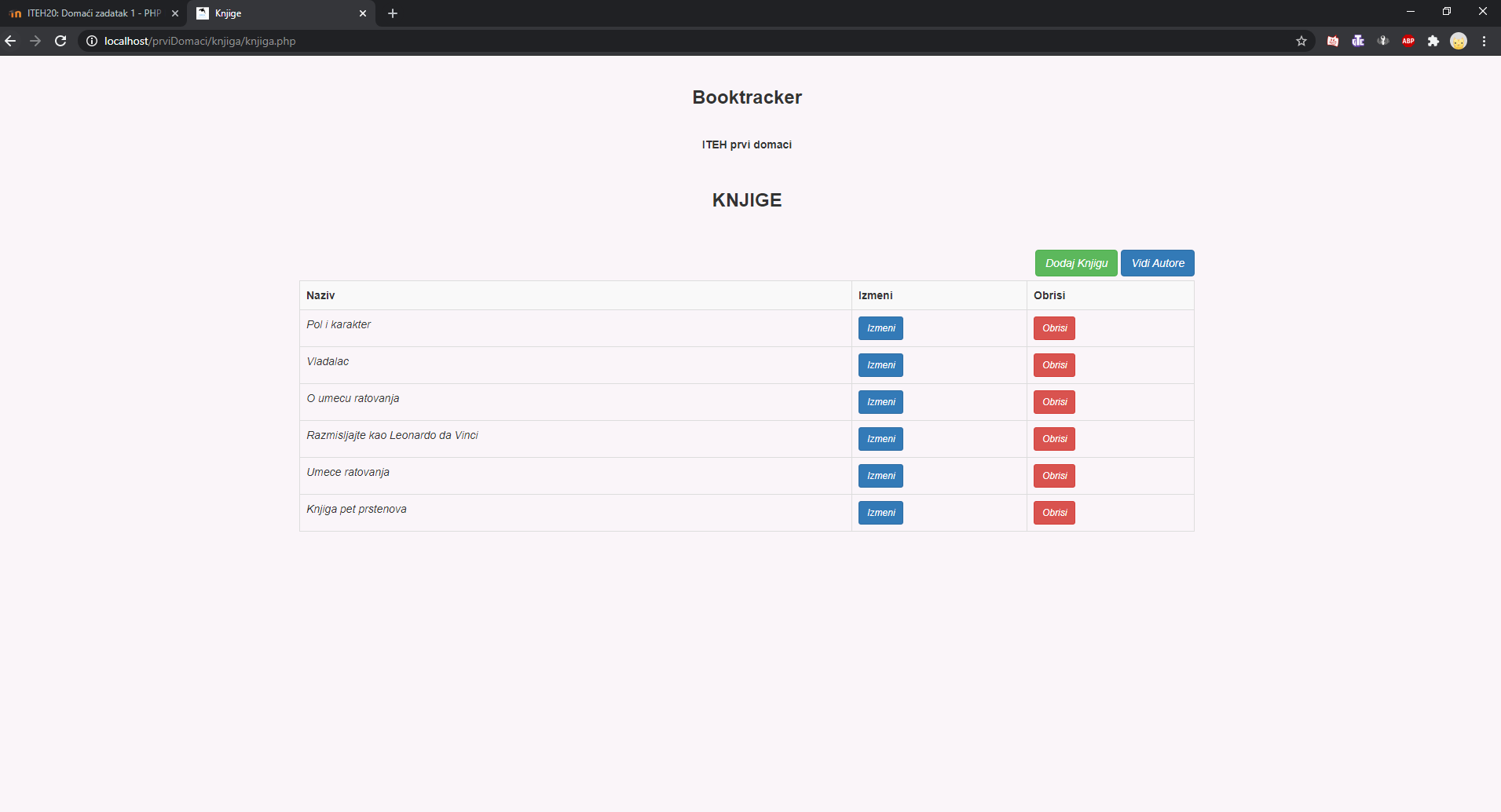
*Slika 1: Log In stranica*

Nakon uspešnog prijavljivanja otvara se stranica za rad sa autorima.



*Slika 2: Autori*

Klikom na dugme ’’ Vidi Knjige’’ odlazimo na stranicu za rad sa knjigama.

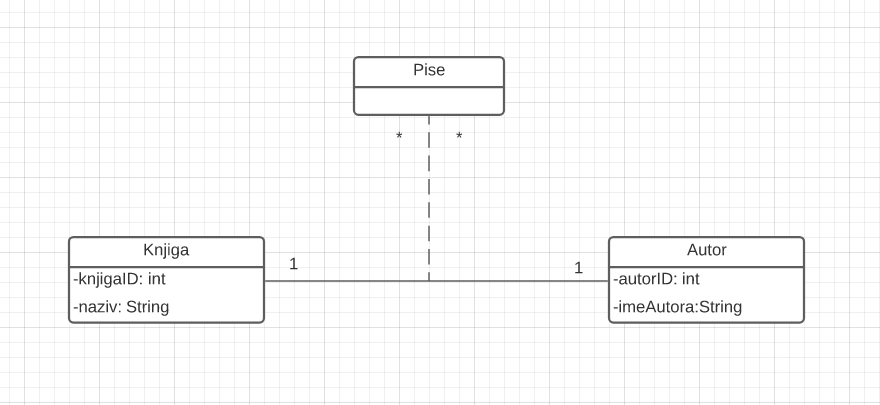


*Slika 3: Knjige*

### 2. Model podataka

Naš model se sastoji od dva entiteta : Autora i Knjige.

Idealno, konceptualni model bi izgledao ovako:



*Slika 4: Konceptualni model*

Jedan autor može da piše više knjiga, a jednu knjigu mogu da napišu više autora (grupa autora).

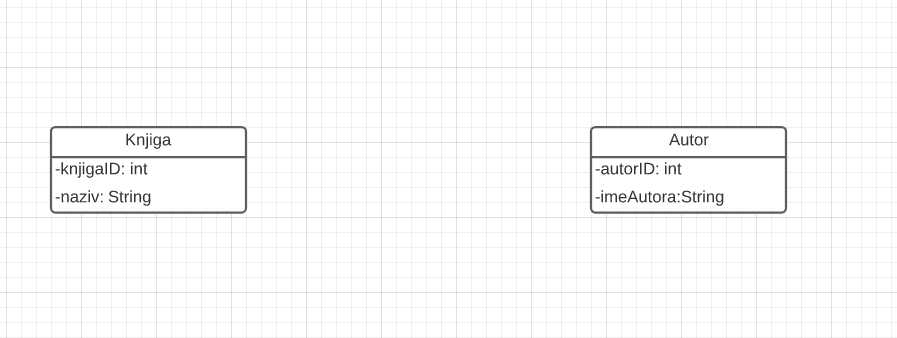
Vezu više prema više predstavljamo jednim asocijativnim entitetom (piše) čiji će identifikator biti složen ključ, sastavljen od primarnih ključeva obe tabele koje ga čine.

U bazi ovo implementiramo preko junction tabele ’’pise’’ koja bi mapirala id-eve autora i knjiga.

S obzirom da je PHP za mene tehnologija sa kojom se nisam pre susretao, i s obzirom da je tema rada demonstracija CRUD operacija i AJAX-a, maksimalno sam pojednostavio model, tako da dva objekta nemaju nikakve veze jedan sa drugim.

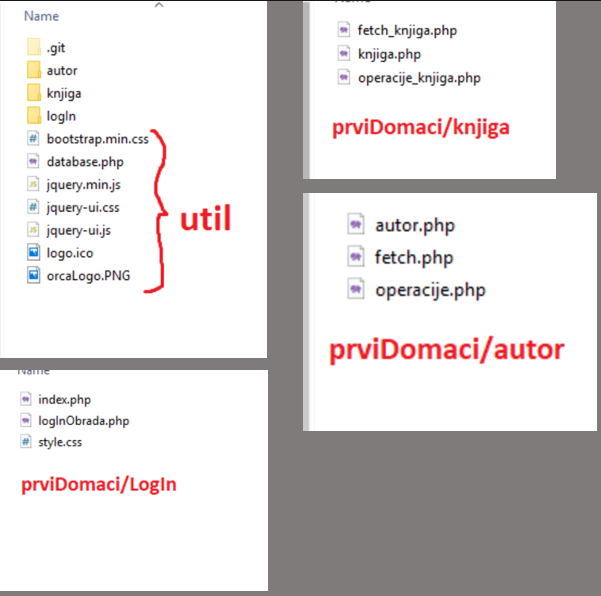
Gore prikazani ’’idealni’’ model bi zahtevao da pri ubacivanju nove knjige u bazu, postoji i ekran za ubacivanje novog autora ili barem combobox za odabir autora. Pri tome potrebno je implementirati i to da više autora može da piše jednu knjigu.

Na pojednostavljenom modelu nema potreba za tim, i možemo se fokusirati na bazične operacije nad bazom.



*Slika 5: Pojednostavljeni model*

### 3. Struktura fajlova



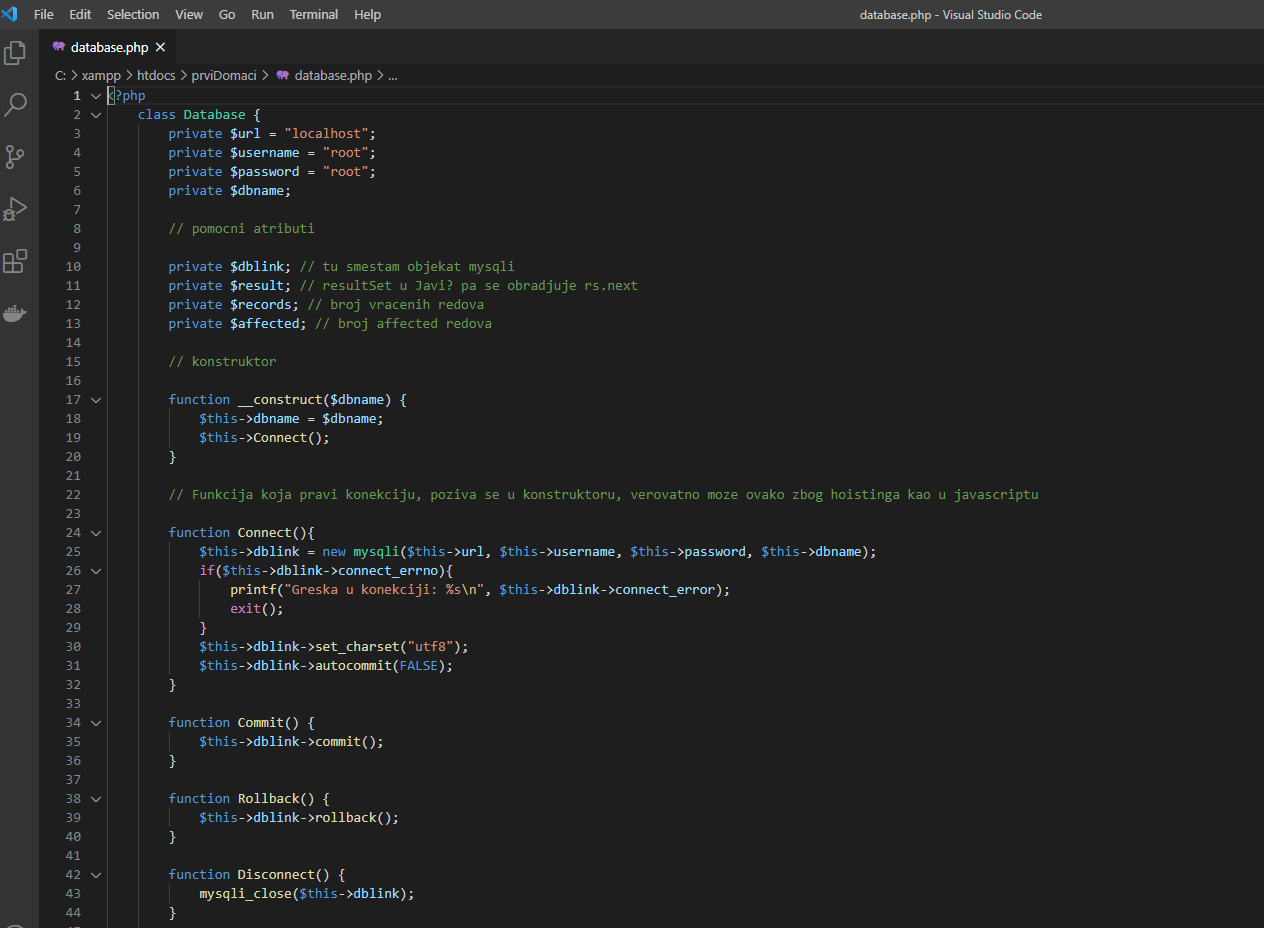
*Slika 6: Struktura fajlova*

### 4. Utill

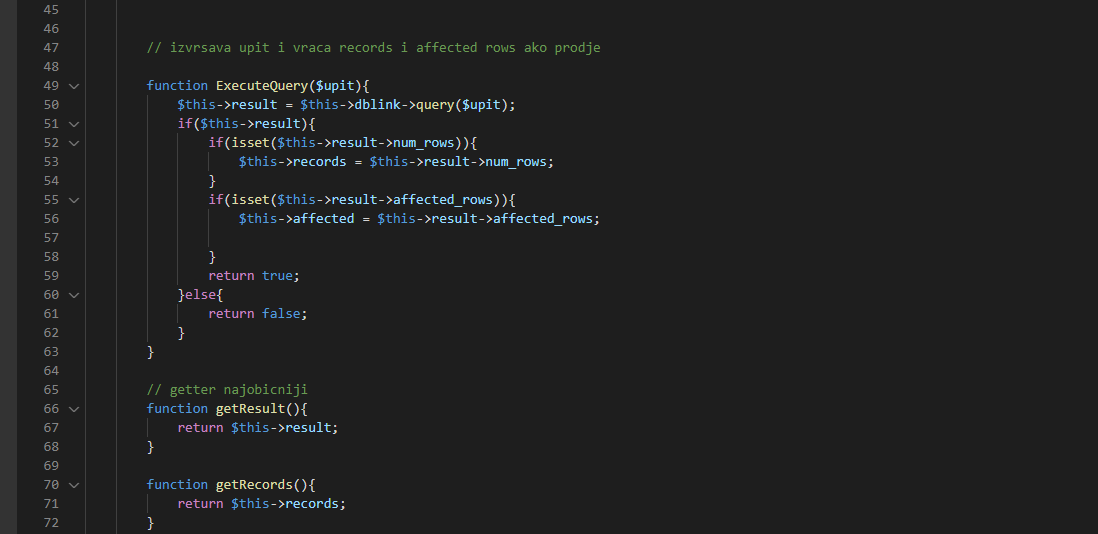
U root-u našeg foldera nalaze se jquery i bootstrap fajlovi kao i database.php, fajl za konekciju na bazu. Takođe, tu se nalazi i slika u .ico formatu koju koristimo kao favicon.

Za konekciju sa bazom koristimo MySqli i to u objektno-orijentisanom obliku.

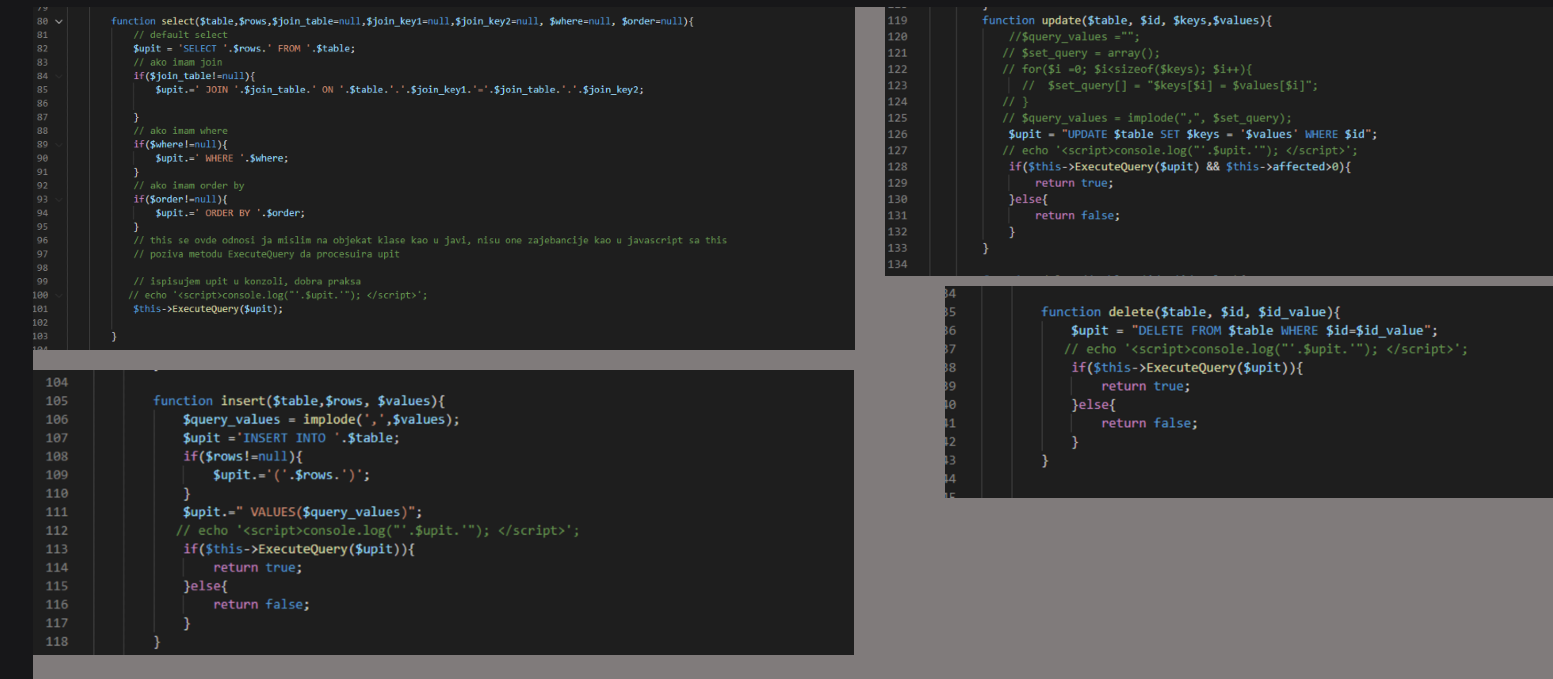
Pravimo klasu Database i u njoj potrebna polja i metode za rad sa bazom.



*Slika 7: Atributi i metode za konekciju i rad sa transakcijama*



*Slika 8: metoda koju pozivaju naše CRUD metode*

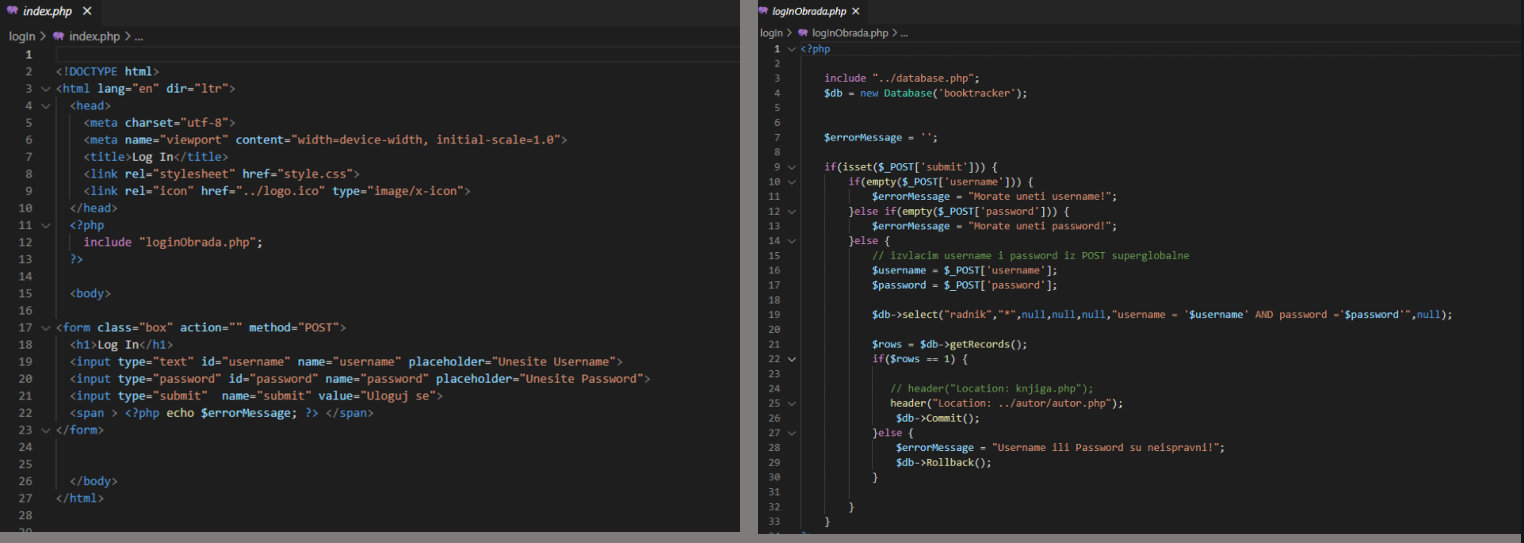
**

*Slika 9 : Crud metode*

### 5. Funkcionalnosti

#### 1. Log in

Da bi se pristupilo stranici za rad sa knjigama i autorima, prvo je potrebno ulogovati se.



*Slika 10: index.php i logInObrada.php*

#### 2. Autor

Kada pristupimo stranici za rad sa autorima, možemo da:

1. Vidimo sve autore

2. Dodamo novog autora

3. Izmenimo postojećeg autora

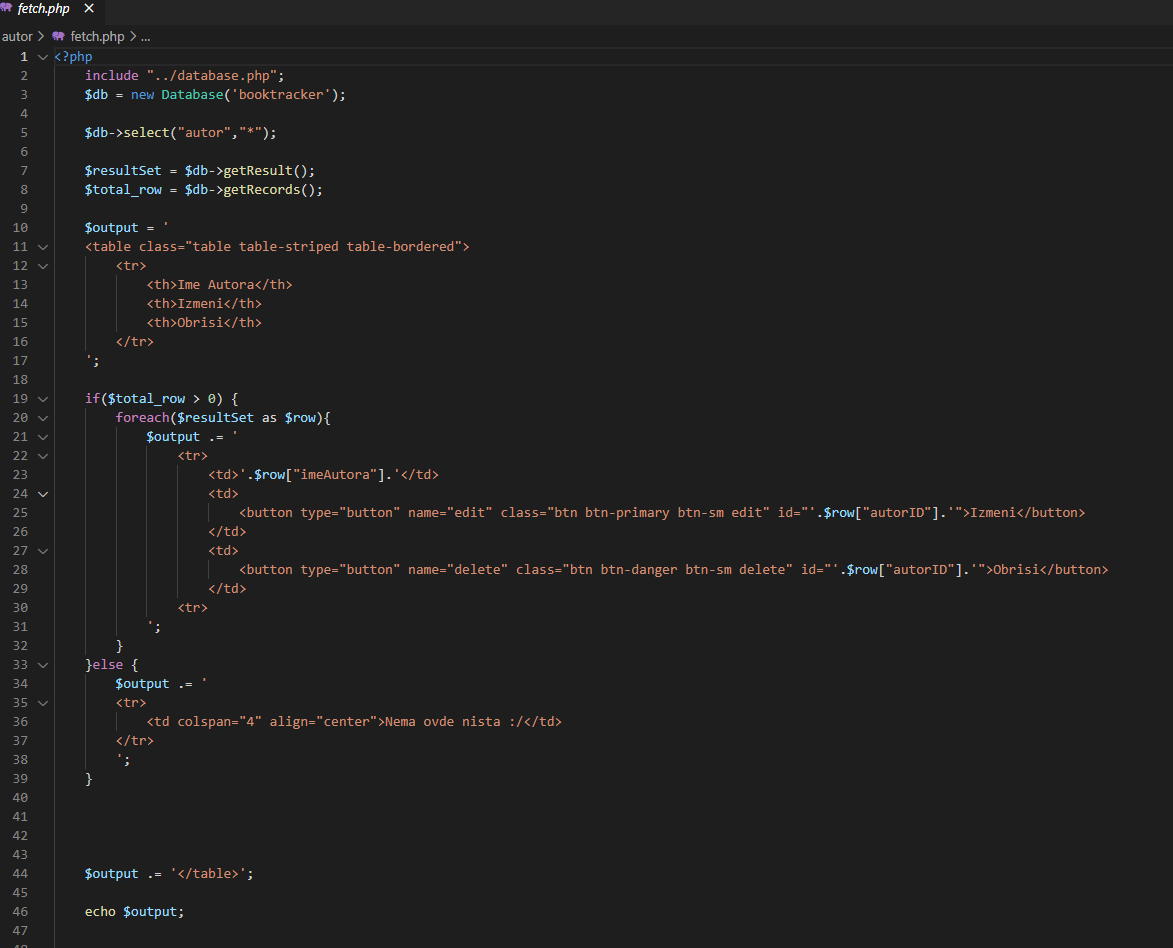
4. Obrišemo autora

Sve naše operacije na serverskoj strani se izvršavaju u fajlu operacije.php i fetch.php.

Fetch.php dinamički generiše tabelu sa sadržajem i šalje je nazad na stranicu.

Zahtev šaljemo preko AJAX-a ( i to korišćenjem JQuery biblioteke, jer olakšava pisanje AJAX-a, bez jquerija morali bi smo da pišemo xmlHttpRequest), tako da svaki put kad nam se sadržaj na stranici promeni, to se radi asihrono, u pozadini, i nema potreba za osvežavanjem cele stranice.

Sve CRUD operacije su takođe izvršene asihrono.



*Slika 11: fetch.php*

**

*Slika 12: Jedan deo skripte operacije.php*

**

*Slika 13: Slanje AJAX zahteva skripti operacije.php nakon klika na dugme ’’izmeni’’*

Za stranicu za rad sa knjigama funkcionalnosti su obezbeđene na sličan način.