Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

PHP OOP MYSQL

Domaći zadatak 1

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
|  |  |  |
| Link ka Github-u | https://github.com/simona243/iteh\_domaci\_1 | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc117932679)

[2 Implementacija 3](#_Toc117932680)

[3 Korisničko uputstvo 3](#_Toc117932681)

# Korisnički zahtev

Potrebno je kreirati aplikaciju koja će olakšati vođenje evidencije o knjigama koje su u prodaji. Potrebno je prikazati sve informacije o knjigama kroz tabelu. Za svaku knjigu se pamti naziv, autor, cena, opis, zanr. Radi lakšeg snalaženja potrebno je omogućiti pretraživanje knjiga po nazivu, kao i sortiranje po ceni rastuće. Pristup aplikaciji imaju samo ulogovani korisnici.

Slučajevi korišćenja:

1. Pregled svih knjiga
2. Pregled jedne knjige
3. Dodavanje nove knjige
4. Brisanje knjige
5. Pretraga
6. Sortiranje
7. Login

# Implementacija

CRUD operacije su implementirane pomoću AJAX-a. za svaku operaciju postoji js funkcija koja poziva php skriptu koja kreira ajax request kojim se vrši upit nad bazom. Za implementaciju ovih operacija korišćene su superglobalne varijable. Kod za dodavanje nove knjige je prilazan ispod. Na isti način su implementirane i ostale CRUD operacije.

Main.js

function dodaj(){

    var form = $('#addForm')[0];

    console.log(form);

    var formData = new FormData(form);

    event.preventDefault();

    console.log(formData);

    request = $.ajax({

        url: 'handler/add.php',

        type: 'post',

        processData: false,

        contentType: false,

        data: formData

    });

    console.log(request);

    request.done(function (response, textStatus, jqXHR) {

        console.log(textStatus);

        console.log(jqXHR);

      console.log(response);

        if (response === "Success") {

            alert("Uspesno dodato");

            location.reload(true);

        }

        else {

            console.log("Neuspesno" + response);

        }

    });

    request.fail(function (jqXHR, textStatus, errorThrown) {

        console.error('Greska: ' + textStatus, errorThrown);

    });

}

Add.php

<?php

    include '../config.php';

    include '../model/Knjiga.php';

    if ( isset($\_POST['naziv']) && isset($\_POST['opis']) && isset($\_POST['cena']) && isset($\_POST['autor'])   ) {

        $knjiga = new Knjiga(null,$\_POST['opis'],$\_POST['cena'],$\_POST['naziv'],$\_POST['zanr'],$\_POST['autor']);

        $status=Knjiga::dodaj($knjiga,$conn);

            if($status){

                echo 'Success';

            }else{

                echo $status;

                echo 'Failed';

            }

      }else{

          echo "Greska";

      }

?>

Knjiga.php

<?php

    class Knjiga{

        private $id;

        private $naziv;

        private $opis;

        private $cena;

        private $zanr;

        private $autor;

        public function \_\_construct($id=null, $opis=null, $cena=null, $naziv=null, $zanr=null,$autor=null){

            $this->naziv=$naziv;

            $this->id=$id;

            $this->opis=$opis;

            $this->cena=$cena;

            $this->zanr=$zanr;

            $this->autor=$autor;

        }

        public static function dodaj($proizvod, $conn){

            $upit = "insert into knjiga(naziv,opis,cena,autor,zanr) values('$proizvod->naziv','$proizvod->opis',$proizvod->cena,'$proizvod->autor',$proizvod->zanr)";

            return $conn->query($upit);

        }

        public static function vratiSve($conn){

            $upit = "select \* from knjiga k inner join zanr z on k.zanr=z.id\_zanra";

            return $conn->query($upit);

        }

        public static function vrati($id, $conn){

            $upit = " select \* from knjiga k inner join zanr z on k.zanr=z.id\_zanra where id=$id";

            return $conn->query($upit);

        }

        public static function obrisi($id, $conn){

            $upit = " delete from knjiga where id=$id";

            return $conn->query($upit);

        }

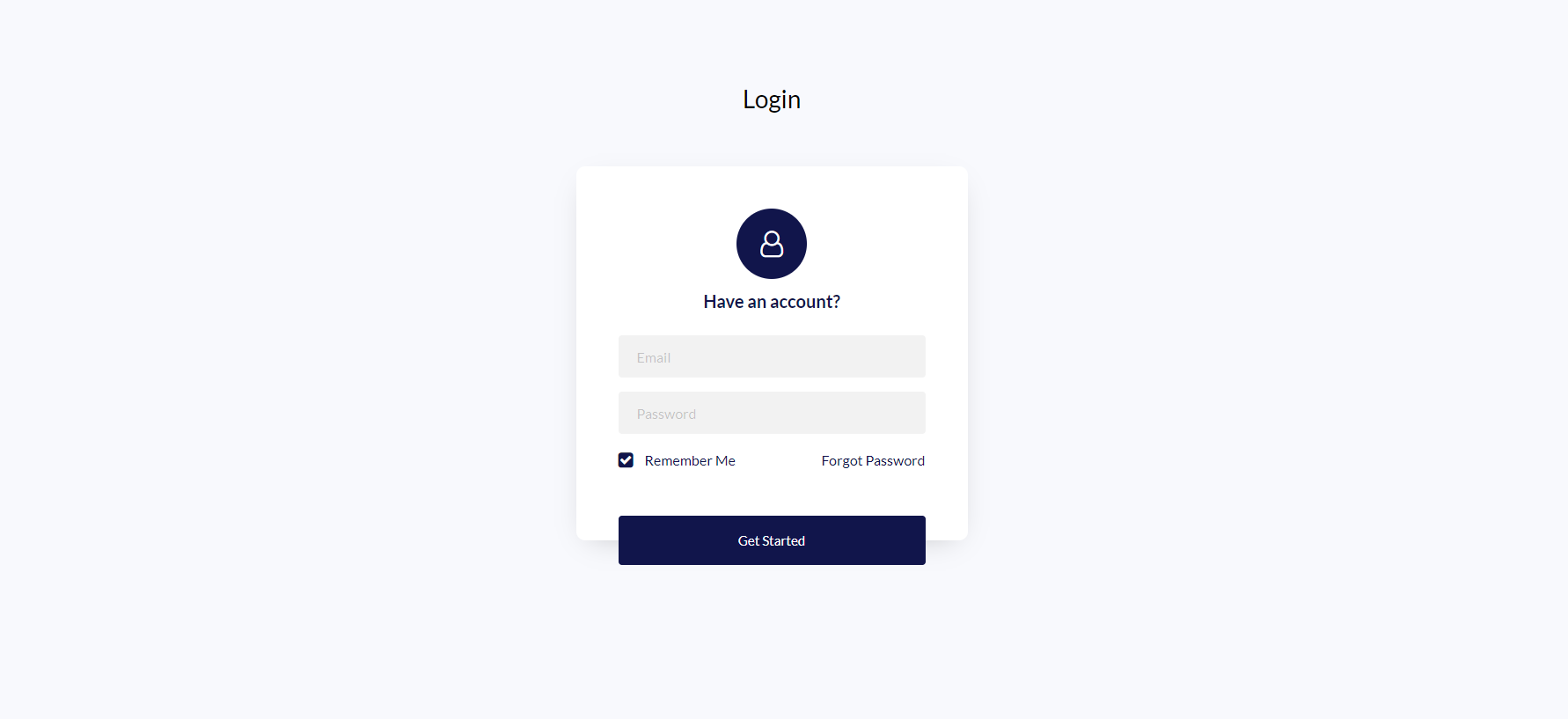
    }

?>

Algoritmi za pretragu i sortiranje su prikazani kodom ispod

# Korisničko uputstvo

Prilikom pokretanja aplikacije korisnicima se prikazuje forma za login, kao na slici ispod.



Nakon uspešnog prijavljivanja korisnicima se prikazuje početna stranica koja izgleda kao na slici ispod.

Slika na kojoj se nalazi tekst, snimak ekrana, monitor, zatvoreni prostor

Opis je automatski generisan

Klikom da dugme Detalji korisnicima se prikazuje modal u kom su prikazani detalji konkretne knjige. Ovo je prikazano na slici ispod.

Slika na kojoj se nalazi tekst, monitor, snimak ekrana, zid

Opis je automatski generisan

Klikom na dugme dodaj novu knjigu korisnicima se otvara modal sa formom preko koje mogu da unesu sve podatke o knjizi. Ovo je prikazano na slici ispod.

Slika na kojoj se nalazi tekst, snimak ekrana, monitor

Opis je automatski generisan