**Dokumentacija**

**Uputstvo za korišćenje ovog projekta:**

* Korisnik može da rezerviše termin tako što će otići na odeljak Termini i cene,rezervacija se vrši tako što korisnik treba da popuni sve podatke,i odma nakon klika na dugme Rezerviši dobiće odgovor.
* Ukoliko nije popunio dobro obrazac, pojaviće se poruka koja će mu to naznačiti.
* Ukoliko je popunio dobro obrazac ali je odabrao termin koji već postoji, pojaviće se poruka koja će mu to naznačiti.
* Ukoliko je dobro popunio obrazac i ukoliko je slobodan termin, pojaviće se poruka koja će mu to naznačiti i dobiće poruku na mejl od kluba kako bi sa dogovorili oko nekih drugih stvari.
* Ukoliko korisnik želi da izrazi svoje mišljenje o klubu,to može da uradi na odeljku za utiske,gde može da ostavi svoj komentar uz propratni naslov i svoje ime,komentar ima ograničenje u broju slova.
* Što se tiče cenovnika,to može da proveri na odeljku usluge.
* Slike kluba se mogu videti na odeljku glaerija.

**Proces izrade projekta:**

U ovom projektu su se koristile sledeće tehnologije:

* HTML
* CSS
* BOOTSTRAP
* JAVASCRIPT & JQUERY
* PHP
* MYSQL

**Izrada rezervacije:**

* Napravljena je HTML strana gde su napravljena input polja u kojima treba da se upišu/odaberu ime,prezime,email,datum žurke,vreme žurk(datum početka i datum kraja žurke) i vrstu žurke.
* Putem AJAX poziva šaljemo zahtev na stranicu “struktura/rezerviši.php” metodom “POST”.
* Fajl “struktura/rezerviši.php” uključuje fajlove “baza/baza.php” i “objekti/forma.php”.
* U fajlu “baza.php” pravimo konekciju sa MySQL bazom podataka.
* U fajlu “forma.php” koji nam je neka vrsta kontrolera,imamo dve bitne funkcije a to su Upisi() i PostojiXY().
* Upisi() predstavlja funkciju kojom upisujemo rezervaciju i ona se beleži u bazi podataka
* public function Upisi(){
* $ispis = "INSERT INTO baza\_klub.podaci\_o\_zurci(ime,prezime,email,datum\_zurke,vreme\_zurke\_od,vreme\_zurke\_do,vrsta\_zurke)
* VALUES (:ime,:prezime,:email,:datum\_zurke,:vreme\_zurke\_od,:vreme\_zurke\_do,:vrsta\_zurke)";
* $upis = $this->konekcija->prepare($ispis);
* $upis->bindParam(":ime",$this->ime);
* $upis->bindParam(":prezime",$this->prezime);
* $upis->bindParam(":email",$this->email);
* $upis->bindParam(":datum\_zurke",$this->datum\_zurke);
* $upis->bindParam(":vreme\_zurke\_od",$this->vreme\_zurke\_od);
* $upis->bindParam(":vreme\_zurke\_do",$this->vreme\_zurke\_do);
* $upis->bindParam(":vrsta\_zurke",$this->vrsta\_zurke);
* $upis->execute();
* return $upis;
* }
* PostojiXY() funkcija sprečava da korisnik upadne u već rezervisani termin,odnosno sprečava unos termina koji već postoji u bazi.
* public function PostojiXY(){
* $pregled = "SELECT \* FROM  baza\_klub.podaci\_o\_zurci WHERE vreme\_zurke\_od  BETWEEN
* :vreme\_zurke\_od AND :vreme\_zurke\_do OR vreme\_zurke\_do BETWEEN  :vreme\_zurke\_od AND :vreme\_zurke\_do";
* $datum = "SELECT \* FROM  baza\_klub.podaci\_o\_zurci WHERE datum\_zurke = '".$this->datum\_zurke."'";
* $datumi = $this->konekcija->prepare($datum);
* $vreme = $this->konekcija->prepare($pregled);
* $vreme->bindParam(":vreme\_zurke\_od",$this->vreme\_zurke\_od);
* $vreme->bindParam(":vreme\_zurke\_do",$this->vreme\_zurke\_do);
* $vreme->execute();
* $datumi->execute();
* if($vreme->rowCount() > 0 && $datumi->rowCount() > 0){
* return true;
* }else{
* return false;
* }
* }
* Dolazimo do fajla “rezervisi.php” gde sve povezujemo i u jednu celinu i vršimo proveru:

<?php

require\_once "../baza/baza.php";

require\_once "../objekti/forma.php";

$mojXML = new SimpleXMLElement("<?xml version='1.0' encoding='UTF-8'?><poruke></poruke>");

//istanca klase Baza

$baza = new Baza();

//konekcija Baze

$db = $baza->PostaviKonekciju();

//istanca klase Struktura

$struktura = new Forma($db);

$struktura->ime = $\_POST["ime"];

$struktura->prezime = $\_POST["prezime"];

$struktura->email = $\_POST["email"];

$struktura->datum\_zurke = $\_POST["datum\_zurke"];

$struktura->vreme\_zurke\_od = $\_POST["vreme\_zurke\_od"];

$struktura->vreme\_zurke\_do = $\_POST["vreme\_zurke\_do"];

$struktura->vrsta\_zurke = $\_POST["vrsta\_zurke"];

//provera dal je sve popunjeno ili nije

if(!isset($struktura->ime) || empty($struktura->ime)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(!isset($struktura->prezime) || empty($struktura->prezime)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(!isset($struktura->email) || empty($struktura->email) || !filter\_var($struktura->email,FILTER\_VALIDATE\_EMAIL)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(!isset($struktura->datum\_zurke) || empty($struktura->datum\_zurke)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(!isset($struktura->vreme\_zurke\_od) || empty($struktura->vreme\_zurke\_od)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(!isset($struktura->vreme\_zurke\_do) || empty($struktura->vreme\_zurke\_do)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if(empty($struktura->vrsta\_zurke)){

    $poruka = $mojXML->addChild("poruka","Podaci nisu tacni ili nisu uneseni!");

}else if($struktura->PostojiXY()){

    $poruka = $mojXML->addChild("poruka","Postovani,postoji rezervacija za taj termin!");

}else if($struktura->Upisi()){

    $poruka = $mojXML->addChild("poruka","Uspesno ste uneli podatke,dobicete poruku na email o daljem dogovoranju.");

}//vraca XML dokumentHeader('Content-type: text/xml');echo $mojXML->asXML();?>

**Postavljanje i čitanje komentara:**

* Kao i kod rezervacije,ovde imamo fajl utisci gde putem dva AJAX poziva zovemo prvo funkciju “struktura/upisi.php” i “struktura/procitaj.php”
* Fajl “upisi.php” uključuje dva fajla “baza/baza.php” i “objekti/komentari.php”.
* U fajlu “baza.php” pravimo konekciju sa MySQL bazom podataka.
* U fajlu “komentari.php” koji nam je neka vrsta kontrolera,imamo dve bitne funkcije a to su DodajKomentar() i PrikaziKomentar ().
* DodajKomentar() je funkcija pomoću koje upisujemo komentar
* function DodajKomentar(){
* $prikazi = "INSERT INTO baza\_klub.komentari(ime\_korisnika,naslov,opis) values (:ime\_korisnika,:naslov,:opis)";
* $upis = $this->konekcija->prepare($prikazi);
* $upis->bindParam(":ime\_korisnika",$this->ime\_korisnika);
* $upis->bindParam(":naslov",$this->naslov);
* $upis->bindParam(":opis",$this->opis);
* $upis->execute();
* return $upis;
* }
* PrikaziKomentar() je funkcija pomoću koje čitamo podatke iz baze
* public function PrikaziKomentar(){
* $pogledaj = "SELECT ime\_korisnika,naslov,opis FROM baza\_klub.komentari ORDER BY ime\_korisnika";
* $prikaz = $this->konekcija->prepare($pogledaj);
* $prikaz->execute();
* return $prikaz;}
* U fajl “upisi.php” povezujemo sve u jednu celinu i vršimo proveru:
* <?php
* require\_once "../baza/baza.php";
* require\_once "../objekti/komentari.php";
* $DokumentXML = new SimpleXMLElement("<?xml version='1.0' encoding='utf-8'?><pisma></pisma>");
* //istanca baze
* $baza = new Baza();
* //istanca konekcije
* $db = $baza->PostaviKonekciju();
* //istanca Komentari
* $komentari = new Komentari($db);
* $komentari->ime\_korisnika = $\_POST["ime\_korisnika"];
* $komentari->naslov = $\_POST["naslov"];
* $komentari->opis = $\_POST["opis"];
* if(!isset($komentari->ime\_korisnika) || empty($komentari->ime\_korisnika)){
* $pismo = $DokumentXML->addChild("pismo","Neispravni podaci!");
* }elseif(!isset($komentari->naslov) || empty($komentari->naslov)){
* $pismo = $DokumentXML->addChild("pismo","Neispravni podaci!");
* }elseif(!isset($komentari->opis) || empty($komentari->opis) || strlen($komentari->opis) > 400 || strlen($komentari->opis) < 12){
* $pismo = $DokumentXML->addChild("pismo","Neispravni podaci!");
* }elseif($komentari->DodajKomentar()){
* $pismo = $DokumentXML->addChild("pismo","Uspesno ste uneli komentar.");
* }
* Header("Content-type:text/xml");
* echo $DokumentXML->asXML();
* ?>
* U fajlu “procitaj.php” povezujemo isto sve u jednu celinu i vršimo proveru,ako je sve ispravno kasnije podatke koje smo fečovali prikazujemo u tabelu i pravi mo XML format:
* <?php
* require\_once "../baza/baza.php";
* require\_once "../objekti/komentari.php";
* if(isset($\_POST["prosledi"])){
* $prosledi = $\_POST["prosledi"];
* }else{
* echo "Greska!";
* }
* //istanca baze
* $baza = new Baza();
* //konekcija baze
* $db = $baza->PostaviKonekciju();
* //istanca klase Komentari
* $prikaz = new Komentari($db);
* //zovemo funkciju
* $procitaj = $prikaz->PrikaziKomentar();
* switch($prosledi){
* case "citaj":
* $XMLfajl = new SimpleXMLElement("<?xml version='1.0' encoding='UTF-8'?><poruka></poruka>");
* if(isset($prosledi)){
* while($kom = $procitaj->fetch(PDO::FETCH\_ASSOC)){
* $poruka = $XMLfajl->addChild("poruke");
* $poruka->addChild("ime\_korisnika",$kom["ime\_korisnika"]);
* $poruka->addChild("naslov",$kom["naslov"]);
* $poruka->addChild("opis",$kom["opis"]);
* }
* }
* break;
* default:"Greska!";
* break;
* }
* Header('Content-type: text/xml');
* echo $XMLfajl->asXML();
* ?>

**Admin stranica:**

* Kod admin stranice imamo prvo interfejs gde admin treba da se uloguje sa odgovarajućom lozinkom.
* Putem AJAX poziva priključujemo “adminlog.php” fajl koji priključuje “baza\_admin.php” fajl i “objekat\_admin.php” fajl.
* Fajl “baza\_admin.php” povezuje bazu.
* Fajl “objekat\_admin.php” ima funkciju LogovanjeAdmina() koja loguje Admina,ako je rowCount() == 1 onda se poklapaju lozinka i korisničko ime
* <?php
* Class Uloguj{
* public $konekcija;
* public $ime;
* public $lozinka;
* function \_\_construct($konekcija){
* $this->konekcija = $konekcija;
* }
* function LogovanjeAdmina(){
* $prikaz = "SELECT \* FROM admin\_panel.admin\_log WHERE ime = :ime and lozinka = :lozinka";
* $pregled = $this->konekcija->prepare($prikaz);
* $pregled->bindParam(":ime",$this->ime);
* $pregled->bindParam(":lozinka",$this->lozinka);
* $pregled->execute();
* if($pregled->rowCount() == 1){
* return true;
* }else{
* return false;
* }
* }
* }
* ?>
* Fajl “admin.log” povezuje u celinu sve i pravi proveru.
* <?php
* require\_once "../admin/baza/baza\_admin.php";
* require\_once "objekat\_admin.php";
* $XmlFajl = new SimpleXMLElement("<?xml version='1.0' encoding='utf-8'?><odgovori></odgovori>");
* $baza = new BazaAdmin();
* $db = $baza->PoveziBazu();
* $ulogujAdmina = new Uloguj($db);
* $ulogujAdmina->ime = $\_POST["ime"];
* $ulogujAdmina->lozinka = $\_POST["lozinka"];
* if(!isset($ulogujAdmina->ime) || empty($ulogujAdmina->ime)){
* $odgovor = $XmlFajl->addChild("odgovor","Niste uneli tacno podatke ili ih niste uneli!");
* }elseif(!isset($ulogujAdmina->lozinka) || empty($ulogujAdmina->lozinka)){
* $odgovor = $XmlFajl->addChild("odgovor","Niste uneli tacno podatke ili ih niste uneli!");
* }elseif($ulogujAdmina->ime != "Admin" || $ulogujAdmina->lozinka != "admin123"){
* $odgovor = $XmlFajl->addChild("odgovor","Niste uneli tacno podatke ili ih niste uneli!");
* }elseif($ulogujAdmina->LogovanjeAdmina()){
* $odgovor = $XmlFajl->addChild("odgovor","Uspesno ste uneli podatke!");
* }
* //Header('Content-type: text/xml');
* echo $XmlFajl->asXML();
* ?>
* Kada se uspešno uloguje na Admin stranicu,tamo može da vidi u tabeli pregled svih rezervacija.
* Fajl “admin\_panel.php” je interfejs na kom su prikazane rezervacije.
* Funkcija Prikazi() je funkcija koja prikazuje podatke iz baze i puni ih u tabelu.
* function Prikazi(){
* $.ajax({
* url:"prikaz\_admin.php",
* type:"POST",
* data:{
* komanda:"citanje"
* },
* dataType:"xml",
* success:function(podatak){
* $(podatak).find("prikaz").each(function(){
* var idXML = $(this).find('id\_podaci').text();
* var imeXML = $(this).find("ime").text();
* var prezimeXML = $(this).find("prezime").text();
* var emailXML = $(this).find("email").text();
* var datum\_zurkeXML = $(this).find("datum\_zurke").text();
* var odXML = $(this).find("vreme\_zurke\_od").text();
* var doXML = $(this).find("vreme\_zurke\_do").text();
* var vrsta\_zurkeXML = $(this).find("vrsta\_zurke").text();
* $("#gle").append('<tr><th scope=row><i class="fas fa-user"></i></th><td>'+imeXML+'</td><td>'+prezimeXML+'</td><td>'+emailXML+'</td><td>'+datum\_zurkeXML+'</td><td>'+odXML+'</td><td>'+doXML+'</td><td>'+vrsta\_zurkeXML+'</td><td><button id='+idXML+' class="delete btn btn-danger">Izbriši</button></td></tr>');
* });
* },
* error:function(){
* alert("greska!");
* }
* });
* }
* Fajl “izaberi\_admin.php” fečuje sve iz baze i pravi XML format.
* <?php
* require\_once "../baza/baza.php";
* require\_once "../objekti/forma.php";
* if(isset($\_POST["komanda"])){
* $komanda = $\_POST["komanda"];
* }else{
* echo "Greska!";
* }
* $baza = new Baza();
* $db = $baza->PostaviKonekciju();
* $dodavanje = new Forma($db);
* $gledaj = $dodavanje->PrikaziPodatke();
* $brojac = $gledaj->rowCount();
* switch($komanda){
* case "citanje":
* $XMLdokument = new SimpleXMLElement("<?xml version='1.0' encoding='UTF-8'?><prikazi></prikazi>");
* if($brojac > 0){
* while($vidi = $gledaj->fetch(PDO::FETCH\_ASSOC)){
* $prikazi = $XMLdokument->addChild("prikaz");
* $prikazi->addChild("id\_podaci",$vidi["id\_podaci"]);
* $prikazi->addChild("ime",$vidi["ime"]);
* $prikazi->addChild("prezime",$vidi["prezime"]);
* $prikazi->addChild("email",$vidi["email"]);
* $prikazi->addChild("datum\_zurke",$vidi["datum\_zurke"]);
* $prikazi->addChild("vreme\_zurke\_od",$vidi["vreme\_zurke\_od"]);
* $prikazi->addChild("vreme\_zurke\_do",$vidi["vreme\_zurke\_do"]);
* $prikazi->addChild("vrsta\_zurke",$vidi["vrsta\_zurke"]);
* }
* }
* break;
* }
* Header('Content-type: text/xml');
* echo $XMLdokument->asXML();
* ?>
* Priključujemo već dva poznata fajla, s tim što sada u falju “forma.php” imamo funkciju PrikaziPodatke()

    public function PrikaziPodatke(){

$vidi = "SELECT id\_podaci,ime,prezime,email,datum\_zurke,vreme\_zurke\_od,vreme\_zurke\_do,vrsta\_zurke FROM baza\_klub.podaci\_o\_zurci";

        $pokazi = $this->konekcija->prepare($vidi);

        $pokazi->execute();

        return $pokazi;

    }

* Dugme Izbrisi predstavlja dugme putem koje admin stranice može da obriše rezervaciju.

$(document).on("click",".delete",function(){

            var id = $(this).attr("id");

            //alert(id);

            $.ajax({

                url:"izbrisi\_admin.php",

                type:"POST",

                data:{id:id},

                dataType:"xml",

                success:function(podaci){

                    var brisi = $(podaci).find("brisi").first().text();

                    if(brisi == "Uspesno ste obrisali!"){

$("#rez").append('<h2>Uspešno ste obrisali korisnika!</h2>').css('color','green');

                    }

                }

            });

        });

* Fajl “izbrisi\_admin.php” uključuje dva fajla “baza.php” i “forma.php” .
* Funkcija IzbrisiRezervaciju() briše podatke putem id\_podaci kolone:
* public function IzbrisiRezervaciju($id\_podaci){

$ispali = "DELETE FROM baza\_klub.podaci\_o\_zurci WHERE id\_podaci = " .$id\_podaci. "";

* $izbrisi = $this->konekcija->prepare($ispali);
* $izbrisi->execute();
* return $izbrisi;
* }
* Fajl “izbrisi\_admin.php” pravi proveru i ako je sve uredu izvršava brisanje:
* <?php
* require\_once "../baza/baza.php";
* require\_once "../objekti/forma.php";
* $XMLbrisi = new SimpleXMLElement("<?xml version='1.0' encoding='UTF-8'?><obrisi></obrisi>");
* $baza = new Baza();
* $db = $baza->PostaviKonekciju();
* $brisanje = new Forma($db);
* $brisanje->id = $\_POST["id"];
* $del = $brisanje->IzbrisiRezervaciju($brisanje->id);
* if(isset($del)){
* $brisi = $XMLbrisi->addChild("brisi","Uspesno ste obrisali!");
* }else if(!isset($brisanje->id)){
* $brisi = $XMLbrisi->addChild("brisi","Greska!");
* }
* Header('Content-type: text/xml');
* echo $XMLbrisi->asXML();
* ?>