

DOKUMENTACIJA



Sofija Vitorovic 171/17

Sajt: <http://hotelsofija610.epizy.com/>

Visoka škola strukovnih studija za informacione i
komunikacione tehnologije

Uvod

Za izradu sajta koriscene su sledece tehnologije:

- HTML
- CSS
- BOOTSTRAP
- PHP
- AJAX
- jQuery
- JavaScript

Funkcionalnosti

Admin panel

Na admin panelu postoji mogucnost dodavanja i brisanja korisnika i soba, kao i ispis poruka koje su korisnici sajta poslali.

Gallery strana

Slike automatski učitane iz baze.

Events strana

Eventovi takodje učitani iz baze, na pocetnoj strani se prikazuju posladnja tri dodata eventa.

Reservation

Strana na kojoj se nalazi glavna logika sajta, mogu joj pristupiti samo ulogovani korisnici. Strana je napravljena za potrebe hotela koji ima odredjene tipove soba. Postoje jednokrevetne, dvokrevetne i trokrevetne, klikom na Check availability, proverava se dostupnost za odredjenu vrstu sobe. Soba sa odredjenim imenom ima tacno odredjen broj ljudi koji su za nju namenjeni. Postoji vise soba istog naziva. Klikom na book now, dole se dinamicki učitava tacno kliknuta soba i pojavljuje se forma za check in i check out, klikom na Confirm booking, upisuje se idKorisnika, idSobe, check in i out u tabelu rezervacija.

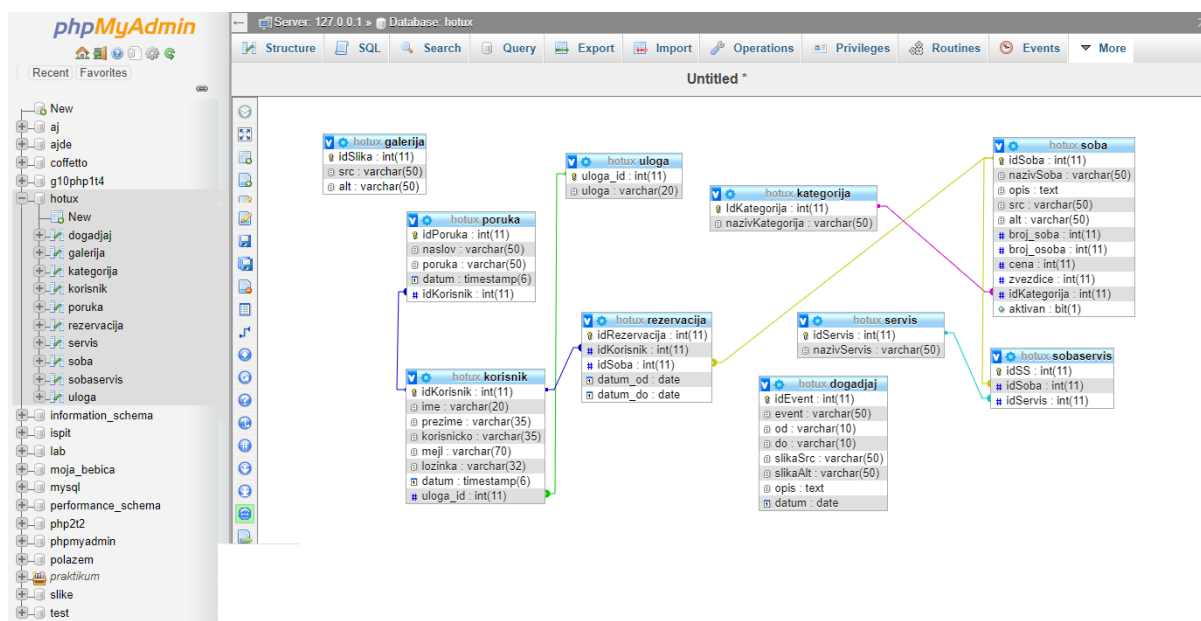
Contact us

Stranica dostupna samo ulogovanim korisnicima, ostavljanje poruke i upis u bazu.

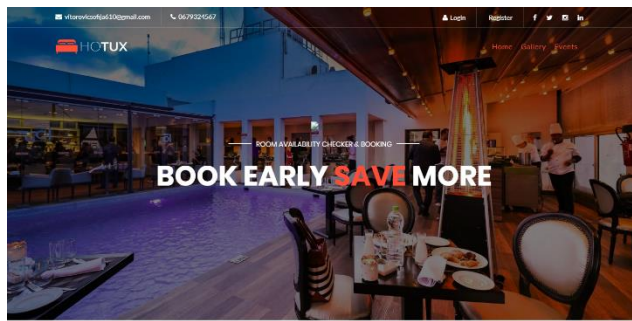
Registracija i logovanje

Na sajtu postoji registracija i logovanje korisnika. Provera podataka prilikom registracije i logovanja je urađena na serverskoj i klijentskoj strani

Baza



SLIKE SAJTA



COMFORT AND JOY ARE PERFECTLY COMBINED HERE

This charming hotel, has been complicitly made with care & passion.



EXPLORE SERVICES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ex neque, sodales accumsan sagittis et, auctor vulputate quam. In consequat, consectetur turpis.



GYM & EXERCISE

Excepteur sint occaecat cupidatat non proident, sunt culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis natus error inventore veritatis et quasi architecto beatae vitae dicta.

BEACH SIGHT

Excepteur sint occaecat cupidatat non proident, sunt culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis natus error inventore veritatis et quasi architecto beatae vitae dicta.



MASSAGE PALOUR

Excepteur sint occaecat cupidatat non proident, sunt culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis natus error inventore veritatis et quasi architecto beatae vitae dicta.



UPCOMING EVENTS OF HOTUX



2020-02-07

HOTELXX

18:00 pm - 03:00 am

Work week is over, so let's party!



2020-02-08

BBQ BEERDAY

19:00 pm - 10:00 am

Beer and BBQ, just a perfect combination!



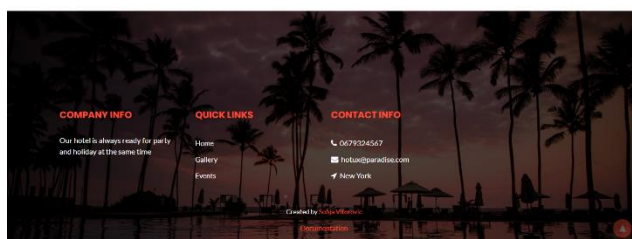
2020-01-25

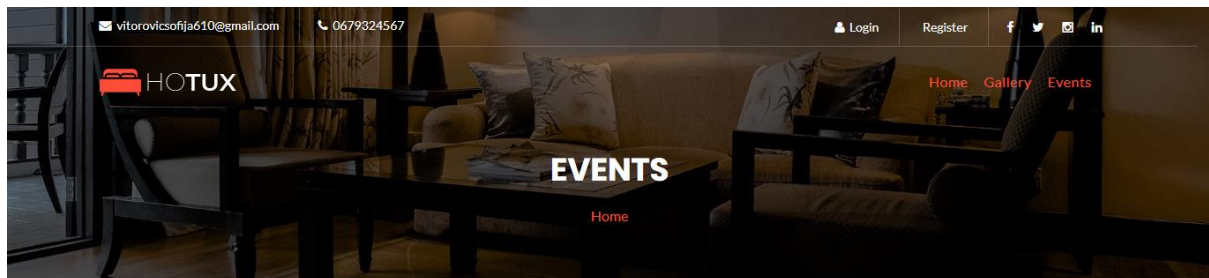
NEW YEAR DAY





19:00 pm - 1:00:00 am

Come and be prepared for night!

HOTUX






	WEDDING PARTY 8:00 pm - 4:00 am 2019-01-06 Let wedding be incredibly!
	HOTELIXX PARTY 8:00 pm - 4:00 am 2020-02-07 Work week is over, so let's party!
	NEW YEAR DAY PARTY 9:00 pm - 15:00 am 2020-01-25 Come and be prepared for night!
	BBQ BEEERDAY PARTY 9:00 pm - 18:00 am 2020-02-06 Beer and BBQ, just a perfect combination!

COMPANY INFO Our hotel is always ready for party and holiday at the same time	QUICK LINKS Home Gallery Events	CONTACT INFO ☎ 0679324567 ✉ hotux@paradise.com 📍 New York
Created by Sofija Vitorovic Documentation		



Make Your Reservation


GUESTS



STANDARD SUITE
₹ 900/night
Perfect pleasure with amazing view

4 Bedrooms Quick Service


[BOOK NOW](#)



DELUXE SUITE
₹ 800/night
Room Deluxe will give you perfect comfort

4 Bedrooms Quick Service


[BOOK NOW](#)



SUPERIOR SUITE
₹ 700/night
Superior room guaranty quality sleep and relax

3 Double rooms Quick Service


[BOOK NOW](#)



NORMAL SUITE
₹ 550/night
All you want after long weekend

2 Bedrooms Quick Service


[BOOK NOW](#)



ROYAL SUITE
₹ 1200/night
Room for real pleasure and luxury

4 Bedrooms Quick Service

[BOOK NOW](#)



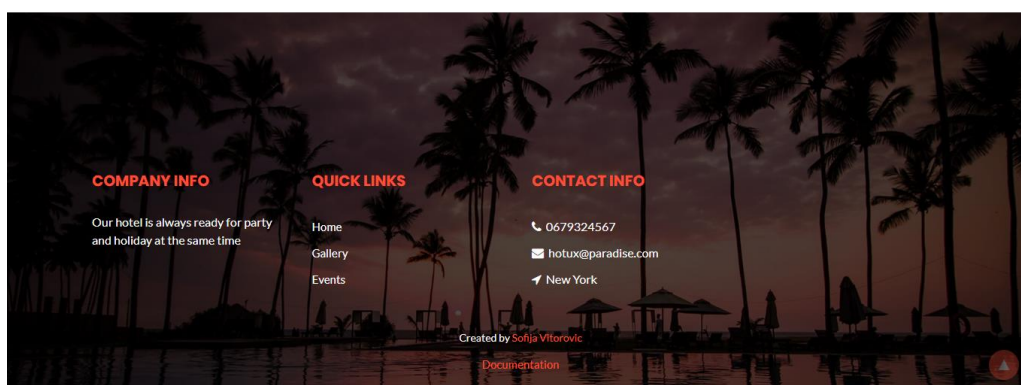
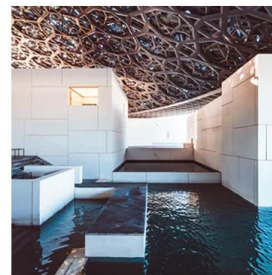
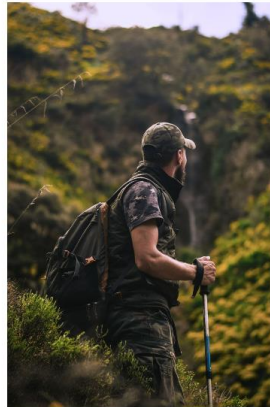
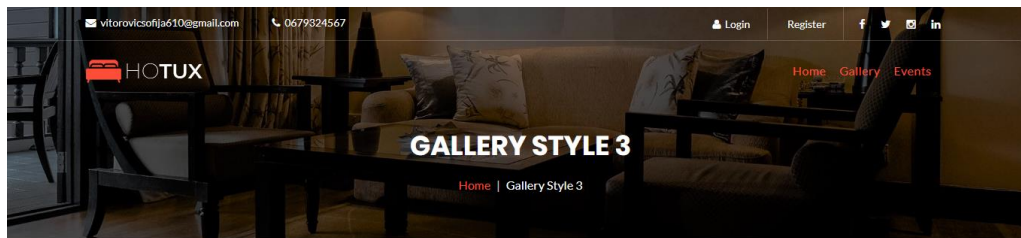
ROYAL SUITE
★★★★★

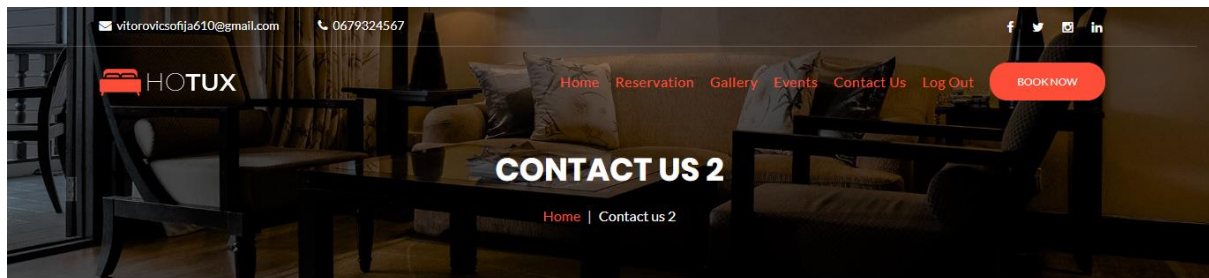
1200/night

YOUR RESERVATION

CHECK IN	CHECK OUT
dd/mm/yyyy	dd/mm/yyyy

CONFIRM BOOKING





445 Mount Eden Road, Mt Eden
Basundhara Chakrapath

977-444-666-888
977-444-222-000

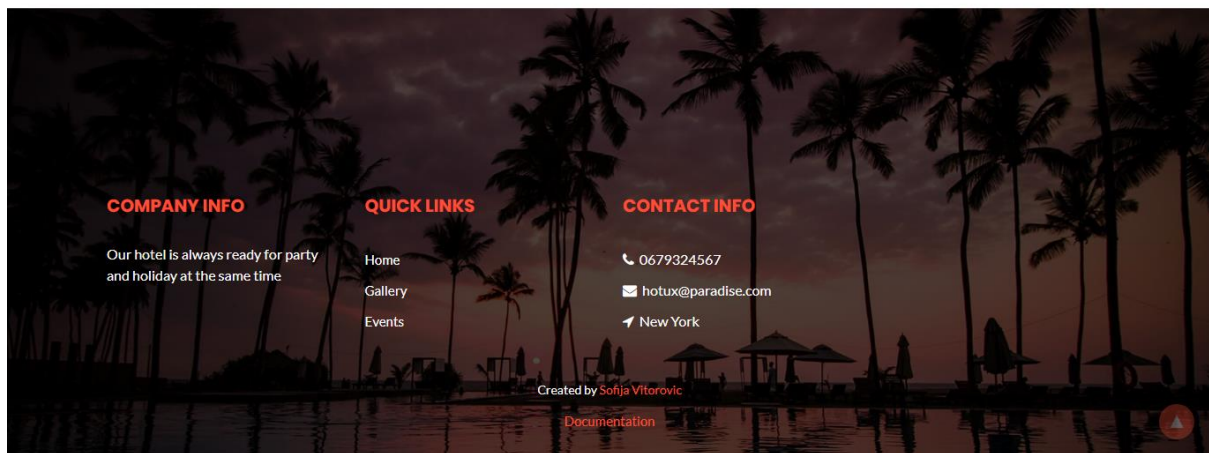
info@hotux.com.np
help@hotux.com.np

LEAVE US A MESSAGE

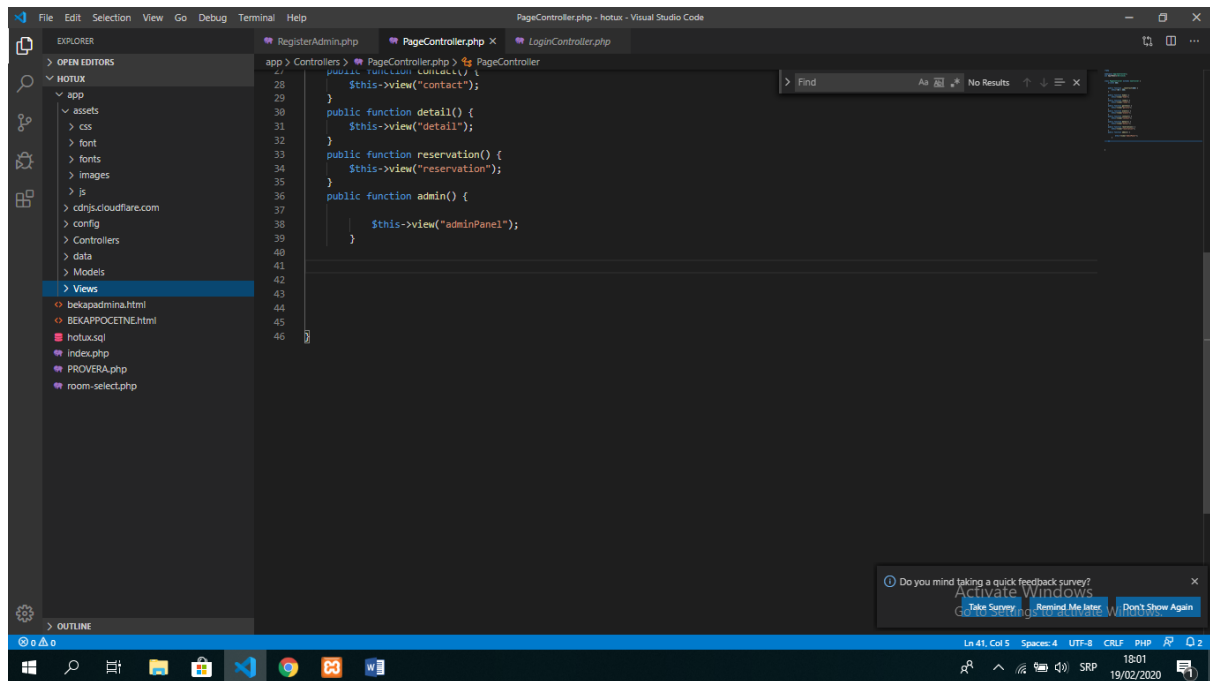
Caption

Enter a message

Singup



Izgled u editor



Kodovi

Index.php

```
<?php

require_once "App/config/autoload.php";
require_once "App/config/database.php";

use app\Models\DB;
use app\Controllers\PageController;
use app\Controllers\RegisterController;
use app\Controllers>LoginController;
use app\Controllers\EventController;
use app\Controllers>ContactController;
use app\Controllers\SobaController;
use app\Controllers\RegisterAdmin;
use app\Controllers\AdminController;
use app\Controllers\InsertRoomController;
use app\Controllers\IspisiController;

$db = new DB(SERVER, DATABASE, USERNAME, PASSWORD);

$pageController = new PageController($db);
$registerController = new RegisterController($db);
$loginController = new LoginController ($db);
$eventController = new EventController ($db);
$contactController=new ContactController($db);
$sobaController=new SobaController($db);
$registerAdminController=new RegisterAdmin($db);
$adminController=new AdminController($db);
$insertRoom=new InsertRoomController($db);
$ispisiController=new IspisiController($db);

if(isset($_GET['page'])){

    switch($_GET['page']){
        case "home":
            $pageController->home();//prikaz home page
            break;

        case "rooms":
```

```
        $pageController->rooms();
        break;
    case "gallery":
        $pageController->gallery();//prikaz galerije
        break;
    case "events":
        $pageController->events();
        //prikaz eventova
        break;
    case "contact":
        $pageController->contact();//prikaz contact strane
        break;
    case "admin":
        $pageController->admin();//prikaz ADMINA koji moram da sredim
        break;
        exit;
        case "detail":
            $pageController->
>detail();//prikaz detalj sobe,MORAM DA SREDIM, IYBACITI PO POTREBI
            break;
            exit;
    case "reservation":
        $pageController->reservation();//strana rezervacija
        break;
        exit;
    case "registerUser":
        $registerController->registerUser();//registruje Korisnika
        break;
        exit;
    case "do-login":
        $loginController->login();// logovanje korisnika
        exit;
        break;
    case "logout":
        $loginController->logout();//logautuje

        break;
    case "prikazEventovaSvih"://dohvatam eventove iz baze sve
        $eventController->eventoviSvi();
        exit;
        break;
    case "prikazSoba":
        $sobaController->sveSobe();
        exit;
        break;
    case "prikazK":
        $sobaController->sveKategorije();
        exit;
```

```
break;
case "probajTop3"://dohvatam top 3 eventa iz baze
    $eventController->eventoviTop3();
    exit;
break;
case "probajSlike":
    $ispisController->slikeSve();
    exit;
break;
case "contactPoruka":
    $contactController->unesiPoruku();
break;
exit;
case "ajde":
    $sobaController->sviServisi();
break;
exit;
//ADMINPANEL
case "registerUserA":
    $registerAdminController->
>registerUserA();//registruje Korisnika
break;
case "probajKategorije"://ovde dohvatam kategorije
    $adminController->kategorijeSve();
break;
case "probajKorisnike"://ovde dohvatam korisnike
    $adminController->korisniciSvi();
break;
case "probajPoruke"://ovde dohvatam poruke
    $adminController->porukeSve();
break;
case "obrisiKorisnika":
    $adminController->obrisiKorisnika();
break;
case "obrisiSobu":
    $adminController->obrisiSobu();
break;
case "insertRoom":
    $insertRoom->insertRoom();
break;
case "probajCheck":
    $sobaController->filtrirajPoKat();
break;
case "available":
    $sobaController->available();
break;
case "dohvatiSobuJednu":
    $sobaController->dohvatiSobu();
```



```
        break;
        case "posaljiR":
            $sobaController->unesiRezeraciju();
            break;

    }
} else {
    $pageController->home();
}
```

Config

Autoload.php

```
<?php
    session_start();

spl_autoload_register(function($classname){

    $classname = lcfirst($classname);
    $classname = str_replace("\\", DIRECTORY_SEPARATOR, $classname);
    $classname .= ".php";

    require_once $classname;
});
```

Database.php

```
<?php

define("SERVER", "localhost");
define("DATABASE", "hotux");
define("USERNAME", "root");
define("PASSWORD", "");
```

APP

Svi javaskriptovi, slike , fontovi ,css...

Hoteli.js

```

$(document).ready(function(){

    $("#sendId").click(registrationUser);
    $("#sendIdA").click(registrationUserA);
    $("#sendMessage").click(messagesFromContactUs);
    getEvents();
    getRooms();
    getCategories();
    getServices();
    getUsers();
    getMessages();
    getGallery();
    getEventsTop3();
    $(document).on("click",".deleteK",function(e){
        e.preventDefault();
        let id=$(this).data('id');

        $.ajax({
            url:'index.php?page=obrisiKorisnika',
            method:'GET',
            data:{
                id:id
            },
            success:function(){
                window.location.replace("http://localhost/hotux/index.php?page
=admin");
            },
            error:function(xhr,greska,status){
                alert(greska);
            }
        })
    })
    $(document).on("click",".deletes",function(e){
        e.preventDefault();
        let id=$(this).data('id');

        $.ajax({
            url:'index.php?page=obrisiSobu',
            method:'GET',
            data:{
                id:id
            },
            success:function(data){
                window.location.replace("http://localhost/hotux/index.php?page
=admin");
            },
            error:function(xhr,greska,status){

```

```

        alert(greska);
    }
    })
})
})

function registrationUser() {

    let fname = $("#fname").val();
    let lname = $("#lname").val();
    let uname = $("#uname").val();
    let mail = $("#mail").val();
    let pass = $("#pass").val();
    let cpass = $("#cpass").val();

    $pFirstname=/^[A-Z]{1}[a-z]{2,25}$/;
    $pLastname=/^[A-Z]{1}[a-z]{2,25}$/;
    $pUsername=/^[a-zA-Z][a-zA-Z0-9-_{3,32}$/; // a-z A-Z - I _
    $PPassword=/^(?=.*\d){6,}$/; // brojevi, donje, slova
    $pEmail=/^[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?$/;

    //PROVERE
    $marker=0; //ako je 1 IMA GRESAKA
    if(fname.match($pFirstname)){
        $('#fG').html('');
    }
    else if(fname==""){
        $('#fG').html( 'Firstname is required field');
        $marker=1;
    }
    else{
        $('#fG').html("Firstname is not in good format");
        $marker=1;
    }
    //llastname
    if(lname.match($pLastname)){
        $('#lG').html( '');
    }
    else if(lname==""){
        $('#lG').html( 'Lastname is required field');
        $marker=1;
    }
    else{
        $('#lG').html("Lastname is not in good format");
        $marker=1;
    }
    //username

```

```

    if(uname.match($pUsername)){
        $('#uG').html( '');
    }
    else if(uname==""){
        $('#uG').html( 'Username is required field');
        $marker=1;
    }
    else{
        $('#uG').html("Username is not in good format");
        $marker=1;
    }
    //PASSWORD
    if(pass.match($PPassword)){
        $('#pG').html( '');
    }
    else if(pass==""){
        $('#pG').html( 'Password is required field');
        $marker=1;
    }
    else{
        $('#pG').html("Password is not in good format");
        $marker=1;
    }
    //REPASS
    if(cpass!=pass){
        $('#rPG').html("Passwords must match");
        $marker=1;
    }
    if(cpass==''){
        $('#rPG').html('');
    }
    else if(cpass==""){
        $('#rPG').html('You must repeat password');
    }
    if(mail.match($pEmail)){
        $('#eG').html( '');
    }
    else if(mail==""){
        $('#eG').html( 'Email is required field');
        $marker=1;
    }
    else{
        $('#eG').html("Email is not in good format");
        $marker=1;
    }
    console.log($marker);
    if($marker==0){

```



```

$.ajax({
  url: 'index.php?page=registerUser',
  method: 'POST',
  data: {
    fname: fname,
    lname: lname,
    uname: uname,
    mail: mail,
    pass: pass,
    kliknuto: true
  },
  success: function() {
    console.log("bravo");
    $("#feed").html("<h2>Successful registration!</h2>");
    $("#sendId").hide();
  },
  error: function(xhr, status, error) {
    console.log(error);
  }
})
}
} //end registracija

//contact us, slanje poruka
function messagesFromContactUs() {
  let naslov = $("#naslovId").val();
  let poruka = $("#poruka").val();
  let id = $("#hidn").val();

  $Renaslov = /^[A-Za-z]{4,25}$/;
  $Reporuka = /^[ a-zA-Z0-9,#.?-]+/;

  $marker = 0;
  if (!naslov.match($Renaslov) || naslov == "" || poruka == '' || !poruka.match($Reporuka)) {
    $marker = 1;
  }

  if ($marker == 0) {
    $.ajax({
      url: 'index.php?page=contactPoruka',
      method: 'POST',
      data: {
        naslov: naslov,

```

```

        poruka:poruka,
        id:id,
        kl:true
    },
    success:function(){
        console.log("bravo");

    },
    error:function(xhr, status, error){
        console.log(error);
    }
})
}

}

function getEvents() {

    $.ajax({
        url: "index.php?page=prikazEventovaSvih",
        method: "GET",
        dataType: "json",
        success: function (data) {
            printEvents(data);
        },
        error: function (xhr, status, error) {

        }

    });
}

function getEventsTop3() {

    $.ajax({
        url: "index.php?page=probajTop3",
        method: "GET",
        dataType: "json",
        success: function (data) {
            printEventsTop3(data);
        },
        error: function (xhr, status, error) {

        }

    });
}

function printEventsTop3(data) {

```



```

                <h4><a href="events-
detail.html"><span>${e.event}</span> Party</a></h4>
                <div class="meta mar-bottom-15">
                    <span class="time mar-right-
10"> <i class="fas fa-clock"></i> ${e.od} pm - ${e.do} am</span>

                </div>
                <p class="mar-0">${e.opis}</p>
            </div>
        </div>
        <div class="col-md-2 col-sm-2 col-xs-12">
            <div class="time-from text-center">
                <span >${e.datum }</span>

            </div>
        </div>
    </div>
</div>
`
    });

    $("#eventovi").html(html);
}
function getGallery() {

    $.ajax({
        url: "index.php?page=probajSlike",
        method: "GET",
        dataType: "json",
        success: function (data) {
            printPhotos(data);
        },
        error: function (xhr, status, error) {

        }

    });
}

function printPhotos(data) {
    let html = "";

    data.forEach(p =>{

        html += `
        <div class="brick">
        <div class="gallery-item">
            <div class="gallery-image">

```

```

        
    </div>

    </div>
</div>
`;
});

$("#tuSlike").html(html);
}
function getRooms() {

$.ajax({
    url: "index.php?page=prikazSoba",
    method: "GET",
    dataType: "json",
    success: function (sobe) {
        printRooms(sobe);
        printRoomsForAdmin(sobe);

    },
    error: function (xhr, status, error){

    }

})
};
function printRoomsForAdmin(sobe) {
    let html = "";

    sobe.forEach(s =>{

        html += `
        <tr>
            <td>${s.idSoba}</td>
            <td>${s.nazivSoba}</td>
            <td>${s.opis}</td>
            <td></td>
            <td>${s.broj_soba}</td>
            <td>${s.broj_osoba}</td>
            <td>${s.cena}</td>
            <td>${s.zvezdice}</td>

            <td><a class= 'deletes' data-id=${s.idSoba}>Delete</a></td>
        </tr>
        `;
    });
}

```



```

        <div class="reservation-detail">
        <input type='hidden' id='hdnSoba' data-
id='${j.idSoba}'>

                <div class="rd-top">

                        <div class="rd-box">
                                <label>Check in</label>
                                <input type='date' id='in' />
                        </div>
                        <div class="rd-box">
                                <label>Check out</label>
                                <input type='date' id='out' />
                        </div>
                </div>
                <div class="rd-top">

                </div>
        </div>
        <table class="reservation-table table">
                <tbody>

                </tbody>

        </table>
    </div>

    </div>
    <div class="card-btn">
    <button id='klik' name='confirm'> <a href="#" class="btn btn-
orange">CONFIRM BOOKING</a></button>
    <div id='res'></div>
</div>
,
    }
    $('#upisi').html(html);
    $('#klik').click(function(){
        let idKor= $('#hidnId').data('id');
        let idSobe= $('#hdnSoba').data('id');
        let od=$('#in').val();
        let out= $('#out').val();
        $.ajax({
            url:'index.php?page=posaljiR',
            method:'POST',
            data:{

```

```

        idKor:idKor,
        idSobe:idSobe,
        od:od,
        out:out,
        kliknutaRez:true
    },
    success:function(){
        console.log("bravo");
        $('#res').html('<h1>Booked!</h1>');
        $('#klik').hide();
    },
    error:function(xhr, status, error){
        console.log(error);
    }
})
})
},
error: function (xhr, status, error) {

}

});
// function printRoomONE(d){

//     html += `    <div class="booking-image">
// 
// <div class="booking-title">
//     <div class="title-left">
//         <h3>Luxury King Room</h3>
//         <div class="rating">
//             <span class="fa fa-star checked"></span>
//             <span class="fa fa-star checked"></span>
//             <span class="fa fa-star checked"></span>
//             <span class="fa fa-star checked"></span>
//             <span class="fa fa-star checked"></span>
//         </div>
//     </div>
//     <div class="title-right pull-right">
//         <div class="title-price">
//             <h4 class="pad-top-
15"><a href="#">$1200<span>/Night</span></a></h4>
//         </div>
//     </div>
// </div>
// </div>`;

```

```

        // }
        // $('#probica').html(html);

    })

}

function getCategories() {
    $.ajax({
        url: "index.php?page=prikazK",
        method: "GET",
        dataType: "json",
        success: function (kategorije) {
            printCat(kategorije);
            printCatForAdmin(kategorije);

        },
        error: function (xhr, status, error){

        }

    })
};

function printCat(kategorije) {
    let html = "";

    kategorije.forEach(k =>{

        html += `
        <div class="pretty p-default p-thick p-pulse mar-bottom-15">
            <input type="checkbox" value="${k.IdKategorija}" name="kategorije"//>
            <div data-id='${k.IdKategorija}' class="state p-warning-o" id='kat'>
                <label>${k.nazivKategorija}</label>
            </div>
        </div>
        `;
    });

    $("#kat").html(html);
}

function getServices() {
    $.ajax({
        url: "index.php?page=ajde",
    })
}

```



```

        method: "GET",
        dataType: "json",
        success: function (servisi) {
            printService(servisi);
        },
        error: function (xhr, status, error){

        }

    })
};
function printService(servisi ) {
    let html = "";

    servisi.forEach(s =>{

        html += `
        <div class="pretty p-default p-thick p-pulse mar-bottom-15" id='ser'>
            <input type="checkbox" />
            <div data id='${s.idServis}'class="state p
-warning-o">

                <label>${s.nazivServis}</label>
            </div>
        </div>
        `;
    });

    $("#ser").html(html);
}
function registrationUserA() {

    let fnameA = $("#fnameA").val();
    let lnameA = $("#lnameA").val();
    let unameA = $("#unameA").val();
    let mailA = $("#mailA").val();
    let passA = $("#passA").val();
    let cpassA = $("#cpassA").val();

    $pFirstname=/^[A-Z]{1}[a-z]{2,25}$/;
    $pLastname=/^[A-Z]{1}[a-z]{2,25}$/;
    $pUsername=/^[a-zA-Z][a-zA-Z0-9-_{3,32}]$/; // a-z A-Z - I _
    $PPassword=/^(?=.*\d).{6,}$/; // brojevi, donje, slova
    $pEmail=/^[a-z0-9!#$%&'*/+=?^_`{|}~- ]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~- ]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?$/;

    //PROVERE
    $marker=0; //ako je 1 IMA GRESAKA

```

```
if(fnameA.match($pFirstname)){
    $('#fGA').html('');
}
else if(fnameA==""){
    $('#fGA').html( 'Firstname is required field');
    $marker=1;
}
else{
    $("#fGA").html("Firstname is not in good format");
    $marker=1;
}
//llastname
if(lnameA.match($pLastname)){
    $('#lGA').html( '');
}
else if(lnameA==""){
    $('#lGA').html( 'Lastname is required field');
    $marker=1;
}
else{
    $("#lGA").html("Lastname is not in good format");
    $marker=1;
}
//username
if(unameA.match($pUsername)){
    $('#uGA').html( '');
}
else if(unameA==""){
    $('#uGA').html( 'Username is required field');
    $marker=1;
}
else{
    $("#uGA").html("Username is not in good format");
    $marker=1;
}
//PASSWORD
if(passA.match($PPassword)){
    $('#pGA').html( '');
}
else if(passA==""){
    $('#pGA').html( 'Password is required field');
    $marker=1;
}
else{
    $("#pGA").html("Password is not in good format");
    $marker=1;
}
//REPASS
```

```

        if(cpassA!=passA){
            $("#rPGA").html("Passwords must match");
            $marker=1;
        }
        if(cpassA==''){
            $("#rPGA").html('');
        }
        else if(cpassA==""){
            $("#rPGA").html('You must repeat password');
        }
        if(mailA.match($pEmail)){
            $('#eGA').html(' ');
        }
        else if(mailA==""){
            $('#eGA').html('Email is required field');
            $marker=1;
        }
        else{
            $("#eGA").html("Email is not in good format");
            $marker=1;
        }
        console.log($marker);
    if($marker==0){

        $.ajax({
            url: 'index.php?page=registerUserA',
            method: 'POST',
            data:{
                fnameA:fnameA,
                lnameA:lnameA,
                unameA:unameA,
                mailA:mailA,
                passA:passA,
                kliknutoA:true
            },
            success:function(){
                console.log(BRAVO);
                $("#feedA").html("<h2>New user added!</h2>");
                $("#sendIdA").hide();

            },
            error:function(xhr, status, error){
                console.log(error);
            }
        })
    }
}
}

```

```

function printCatForAdmin(kategorije) {
    let html = "";

    kategorije.forEach(ka =>{

        html += `

        <option name='katS' value="${ka.IdKategorija}">${ka.nazivKategorija}</option
    >

        `;
    });

    $("#kategorijaa").html(html);
}
function getUsers() {

    $.ajax({
        url: "index.php?page=probajKorisnike",
        method: "GET",
        dataType: "json",
        success: function (data) {
            printUsers(data);
        },
        error: function (xhr, status, error) {

        }

    });
}

function printUsers(data) {
    let html = "";

    data.forEach(u =>{

        html += `
    <tr>
        <td>${u.idKorisnik}</td>
        <td>${u.ime}</td>
        <td>${u.prezime}</td>
        <td>${u.uloga}</td>

        <td><a class= 'deleteK' data-id="${u.idKorisnik}">Delete</a></td></tr>

```

```

        `;
    });

    $("#korisnici").html(html);
}
function getMessages() {

    $.ajax({
        url: "index.php?page=probajPoruke",
        method: "GET",
        dataType: "json",
        success: function (data) {
            printPoruke(data);
        },
        error: function (xhr, status, error) {

        }

    });
}

function printPoruke(data) {
    let html = "";

    data.forEach(p =>{

        html += `

        <h3>${p.poruka}</h3><p> user ${p.ime} ${p.prezime} , with id ${p.idKorisnik}</p>

        `;
    });

    $("#poruke").html(html);
}

$("#kat").change(function(){//filter checkboxova
let kategorije = $("input[name='kategorije']:checked");
    let selectedCategories = [];

    for(let kategorija of kategorije){
        selectedCategories.push(kategorija.value);
    }
    console.log(selectedCategories);
    $.ajax({

```

```

        url: 'index.php?page=probajCheck',
        method: 'POST',
        data: {

            kategorije: selectedCategories,
            send: true
        },
        success: function(data, status, xhr){
            console.warn("Bravo");

        },
        error: function(xhr, status, statusText){
            console.error('----> ERROR <----');
            console.log(xhr);
        }
    })
} );
// $("#rangeCena").mousemove(function(){

//     let min = $("#span#min.min-value").val();
//     let max = $("#max").val();

//     console.log(min ,max);
//     $.ajax({
//         url: 'index.php?page=probajRange',
//         method: 'POST',
//         data: {

//             min:min,
//             max:max,
//             okej:true
//         },
//         success: function(data, status, xhr){
//             console.warn("Bravo");

//         },
//         error: function(xhr, status, statusText){
//             console.error('----> ERROR <----');
//             console.log(xhr);
//         }
//     })
// } );
$('#filtriraj').click(function(){
    let brojGostiju=$('#guests').val();

    $.ajax({
        url:'index.php?page=available',
        method:'POST',

```

```

        dataType: 'json',
        data: {
            brojGostiju: brojGostiju,

            poslato: true
        },
        success: function(data) {
            printRooms(data);
        },
        error: function(xhr, greska, status) {
            alert(greska);
        }
    })
})

```

Controllers

ADMINCONTROLLER.PHP

```

<?php

namespace App\Controllers;
use App\Models\Admin;

class AdminController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function kategorijeSve(){
        $adminModel = new Admin($this->db);

        $kategorije = $adminModel->getAllCategories();
        echo \json_encode($kategorije);
    }

    public function korisniciSvi(){
        $adminModel = new Admin($this->db);

        $korisnici = $adminModel->getAllUsers();
        echo \json_encode($korisnici);
    }

    public function porukeSve(){
        $adminModel = new Admin($this->db);
    }
}

```

```

        $poruke = $adminModel->getAllMesagess();
        echo \json_encode($poruke);
    }

    public function obrisiKorisnika(){
        if(isset($_GET['id'])){

            $id = $_GET['id'];

            try {
                $model = new Admin($this->db);
                $model->deleteUser($id);
                http_response_code(204);
            }
            catch (\PDOException $ex) {
                echo json_encode(['message'=> $ex->getMessage()]);
                http_response_code(500);
            }
        }
    }

    public function obrisiSobu(){
        if(isset($_GET['id'])){

            $id = $_GET['id'];

            try {
                $model = new Admin($this->db);
                $model->deleteSoba($id);
                http_response_code(204);
            }
            catch (\PDOException $ex) {
                echo json_encode(['message'=> $ex->getMessage()]);
                http_response_code(500);
            }
        }
    }
}

```


CONTROLLER

```
<?php

namespace App\Controllers;

class Controller {

    protected function view($fileName, $data = []){

        // extract($data);
        include "app/Views/fixed/top.php";
        include "app/Views/fixed/head.php";
        include "app/Views/fixed/meni.php";
        include "app/Views/pages/".$fileName.".php";
        /* include "app/Views/fixed/register.php";
        include "app/Views/fixed/login.php"; */
        include "app/Views/fixed/futer.php";

    }

    protected function redirect($page) {
        header("Location: " . $page);
    }

    protected function json($data = null, $statusCode = 200) {
        header("content-type: application/json");
        http_response_code($statusCode);
        echo json_encode($data);
    }

}
```

ContactController.php

```
<?php

namespace App\Controllers;
use App\Models\ContaktPoruka;

class ContactController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function unesiPoruku(){
        if(isset($_POST['kl'])){
```



```

namespace App\Controllers;
use App\Models\Event;

class EventController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function eventoviSvi(){
        $eventModel = new Event($this->db);

        $eventovi = $eventModel->getAllEvents();
        echo \json_encode($eventovi);
    }
    public function eventoviTop3(){
        $eventModel = new Event($this->db);
        $top3=$eventModel->getTop3Events();
        echo \json_encode($top3);
    }
}

```

InsertRoomController.php

```

<?php

namespace App\Controllers;
use App\Models\Soba;

class InsertRoomController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function insertRoom(){

        if(isset($_POST['sobaSend'])){

            $soba = $_POST['nazivSobe'];
            $opis = $_POST['opis'];
            $slika = $_FILES['slika'];
            $cena=$_POST['cena'];
            $brojOsoba=$_POST['brojOsoba'];
            $brojSoba = $_POST['brojSoba'];
            $zvezdice = $_POST['zvezdice'];

```

```

$idKategorijeS=$_POST['kategorijaaS'];
$ime_fajla = $slika['name'];
$tip_fajla = $slika['type'];
$velicina_fajla = $slika['size'];
$tmp_putanja = $slika['tmp_name'];
$errorsRoom = [];
    $dozvoljeni_formati = array("image/jpg", "image/jpeg",
        "image/png");
        if(!in_array($tip_fajla, $dozvoljeni_formati)){
            $errorsRoom[] = "Type is not good!";
        }
        if($velicina_fajla > 3000000){
            $errorsRoom[] = "File must be less than 3MB";
        }
        if(empty($soba)){
            $errorsRoom[] = "Room is required field.";
        }
        if(empty($opis)){
            $errorsRoom[] = "Caption is required field.";
        }
        if(empty($scena)){
            $errorsRoom[] = "Price is required field.";
        }
        if(empty($brojOsoba)){
            $errorsRoom[] = "Number of people is required field.";
        }
        if(empty($brojSoba)){
            $errorsRoom[] = "Number of rooms is required field.";
        }
        if(empty($zvezdice)){
            $errorsRoom[]="Stars required.";
        }
        if(empty($idKategorijeS)){
            $errorsRoom[]="Category required";
        }
        if(count($errorsRoom)!=0){
            $status = 400;

        }
        else{
            $novNaziv = $ime_fajla;
            $nova_putanja= "app/assets/images/room-
list/.$novNaziv";
            if(move_uploaded_file($tmp_putanja, $nova_putanja)){

```

```

        try {
            $model = new Soba($this->db);
            $model->
>DodajSobu($soba,$opis,$nova_putanja,$brojSoba,$brojOsoba,$scena,$zvezdice,$idKategorijeS);
            http_response_code(201);

        }

        catch (\PDOException $ex) {
            echo json_encode(['message'=> $ex->getMessage()]);
            http_response_code(500);
        }
    }
}
} }}

```

IspisiController.php

```

<?php

namespace App\Controllers;
use App\Models\Ispis;

class IspisiController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function slikeSve(){
        $model = new Ispis($this->db);

        $slike = $model->getAllPhotos();
        echo \json_encode($slike);
    }
}

```

LoginController.php

```

<?php

namespace App\Controllers;
use App\Models\Korisnik;

```

```

class LoginController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function login() {

        $errors = [];
        if(isset($_POST['sendL'])){
            if(!isset($_POST["passwordL"])) {
                array_push($errors, "Password is required.");
            }

            if(!isset($_POST["usernameL"])) {
                array_push($errors, "Username is required.");
            }
            if(count($errors)) {
                $this->view('reservation', [
                    "errors" => $errors
                ]);
                exit;
            }

            try {
                $model = new Korisnik($this->db);
                $user = $model-
>findByUsernameAndPassword($_POST["usernameL"],$_POST["passwordL"]);

                if($user != null) {
                    $_SESSION["user"] = $user;
                    $_SESSION["uloga"] = $user->uloga_id;
                }
                // $this->view('events', [
                //     "user" => $user
                // ]);
                if($_SESSION['uloga']!=1){
                    $this->redirect("index.php");
                }else{
                    $this->redirect("index.php?page=admin");
                }

            }catch(PDOException $ex) {
                $this->redirect("index.php");
            }
        }
    }
}

```

```

        echo json_encode(['message'=> $ex->getMessage()]);
        http_response_code(500);
    }

}

// if(!$erros) {
//     array_push($errors, "Invalid username and password combination.
");
// }

}

public function logout() {
    /* $_SESSION["user"] = null;
    $_SESSION["uloga"] = null;*/
    session_destroy();
    $this->redirect("index.php");
}

}

}

```

PageController.php

```

<?php

namespace App\Controllers;
use App\Models\Korisnik;

class PageController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function home() {
        $this->view("home");
    }
    public function rooms() {
        $this->view("rooms");
    }
    public function gallery() {
        $this->view("gallery");
    }
    public function events() {
        $this->view("events");
    }
}

```

```

    }
    public function contact() {
        $this->view("contact");
    }
    public function detail() {
        $this->view("detail");
    }
    public function reservation() {
        $this->view("reservation");
    }
    public function admin() {

        $this->view("adminPanel");
    }
}

```

RegisterConrtoller.php

```

<?php

namespace App\Controllers;
use App\Models\Korisnik;

class RegisterController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function registerUser(){
        if(isset($_POST['kliknuto'])){

            $firstname=$_POST['fname'];
            $lastname=$_POST['lname'];
            $username=$_POST['uname'];
            $email=$_POST['mail'];
            $password=$_POST['pass'];

            //var_dump($_POST);
            $errors=[];

            $reFirstLast="/^[A-Z]{1}[a-z]{2,25}$/";
            $rePassword="/^(?=.*\d).{6,}$/";

```



```

use App\Models\Soba;

class SobaController extends Controller {
    private $db;

    public function __construct($db) {
        $this->db = $db;
    }

    public function sveSobe(){
        $sobeModel = new Soba($this->db);

        $sobe = $sobeModel->getAllAvailableRooms();
        echo \json_encode($sobe);
    }

    public function sveKategorije(){
        $sobeModel = new Soba($this->db);

        $kategorije = $sobeModel->getAllCategories();
        echo \json_encode($kategorije);
    }

    public function sviServisi(){
        $sobeModel = new Soba($this->db);

        $servisi = $sobeModel->getServices();
        echo \json_encode($servisi);
    }

    public function filtrirajPoKat(){
        if(isset($_POST['kategorije'])){
            $kategorije=$_POST['kategorije'];
            try{
                $model = new Soba($this->db);
                $model->FiltrirajSobu ($kategorije);
                http_response_code(204);

            }
            catch (\PDOException $ex) {
                echo json_encode(['message'=> $ex->getMessage()]);
                http_response_code(500);
            }
        }
    }

    public function available(){
        if(isset($_POST['poslato'])){
            $brojGostiju=$_POST['brojGostiju'];

```

```

        try{
            $model = new Soba($this->db);
            $sobeA=$model-> getRoomsAvailable($brojGostiju);
            echo \json_encode($sobeA);
        }
        catch (\PDOException $ex) {
            echo json_encode(['message'=> $ex->getMessage()]);
            http_response_code(500);
        }
    }
}

public function dohvatiSobu(){
    if(isset($_POST['idS'])){
        $idS=$_POST['idS'];

        try{
            $model = new Soba($this->db);
            $sobA=$model-> getRoom($idS);
            echo \json_encode($sobA);
        }
        catch (\PDOException $ex) {
            echo json_encode(['message'=> $ex->getMessage()]);
            http_response_code(500);
        }
    }
}

public function unesiRezeraciju(){
    if(isset($_POST['kliknutaRez'])){
        $idKor=$_POST['idKor'];
        $idSobe=$_POST['idSobe'];
        $od=$_POST['od'];
        $out=$_POST['out'];

        try {
            $model = new Soba($this->db);
            $model->upisiRez($idKor,$idSobe,$od,$out);
            http_response_code(201);
        }
        catch (\PDOException $ex) {
            echo json_encode(['message'=> $ex->getMessage()]);
            http_response_code(500);
        }
    }
}

```

```

    }
}
}

```

Models

Admin.php

```

<?php
namespace App\Models;

class Admin {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }
    public function getAllCategories(){
        return $this->db->executeQuery("SELECT * FROM kategorija");
    }
    /* public function DodajKorisnika ($firstname, $lastname, $username, $email
    , $password) {

        $params = [$firstname, $lastname, $username, $email, $password];

        $query = "INSERT INTO korisnik values(NULL, ?, ?, ?, ?, ?,CURRENT_TIME
STAMP, 2)";

        $this->db->executeInsert($query, $params);
    }*/
    public function getAllUsers(){
        return $this->db-
>executeQuery("SELECT * FROM korisnik k inner join uloga u on u.uloga_id=k.ulo
ga_id");
    }
    public function getAllMesagess(){
        return $this->db-
>executeQuery("SELECT p.idKorisnik,k.ime,k.prezime,p.naslov,p.poruka FROM poru
ka p inner join korisnik k ON p.idKorisnik=k.idKorisnik ");
    }

    // public function obrisiKorisnika(){
    //
    // }
}

```

```

public function deleteUser($id) {
    $params = [$id];

    $query = "DELETE FROM korisnik WHERE idKorisnik = ?";

    return $this->db->executeInsert($query, $params);
}
public function deleteSoba($id) {
    $params = [$id];

    $query = "DELETE FROM soba WHERE idSoba = ?";

    return $this->db->executeInsert($query, $params);
}
}

```

ContactPoruka.php

```

<?php
namespace App\Models;

class KontaktPoruka {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }
    public function upisiPoruke($naslov,$poruka,$idKorisnik){
        $params = [$naslov, $poruka, $idKorisnik];

        $query = "INSERT INTO poruka values(NULL, ?, ?,CURRENT_TIMESTAMP, ?)";

        $this->db->executeInsert($query, $params);
    }
}

```

DB.php

```

<?php

namespace App\Models;

class DB {
    private $server;
    private $database;
    private $username;

```

```

private $password;

public $conn;

public function __construct($server, $database, $username, $password){
    $this->server = $server;
    $this->database = $database;
    $this->username = $username;
    $this->password = $password;

    $this->connect();
}

private function connect(){
    $this->conn = new \PDO("mysql:host={$this->server};dbname={$this->database};charset=utf8", $this->username, $this->password);

    $this->conn->setAttribute(\PDO::ATTR_DEFAULT_FETCH_MODE, \PDO::FETCH_OBJ);

    $this->conn->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
}

public function executeQuery($query){
    // bilo koji SELECT upit bez WHERE
    return $this->conn->query($query)->fetchAll();
}

public function executeGet($query) {
    return $this->conn->query($query)->fetchAll();
}

public function executeQueryWithParams($query, $params){
    $prepare = $this->conn->prepare($query); // SELECT * WHERE naziv = ? OR opis = ?
    // $params = ['dads', 'dasd']
    $prepare->execute($params);
    return $prepare->fetchAll();
}

public function executeInsert($query, $params){
    $prepare = $this->conn->prepare($query);
    $prepare->execute($params);
}

}

```

Event.php

```
<?php
namespace App\Models;

class Event {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }

    public function getAllEvents(){
        return $this->db->executeQuery("SELECT * FROM dogadjaj");
    }

    public function getTop3Events(){
        return $this->db->executeQuery("select * FROM dogadjaj ORDER by datum DESC limit 3");
    }

    public function datumSamo($datum){

        $timestamp = strtotime($datum);
        $yearmonthday = date('Y m d', $timestamp);
        echo $yearmonthday;

    }

}
?>
```

Ispis.php

```
<?php
namespace App\Models;

class Ispis {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }

    public function getAllPhotos(){
        return $this->db->executeQuery("SELECT * FROM galerija");
    }

}
?>
```

Korisnik.php

```
<?php
namespace App\Models;

class Korisnik {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }

    public function DodajKorisnika ($firstname, $lastname, $username, $email, $password) {

        $params = [$firstname, $lastname, $username, $email, $password];

        $query = "INSERT INTO korisnik values(NULL, ?, ?, ?, ?, ?,CURRENT_TIME
STAMP, 2)";

        $this->db->executeInsert($query, $params);
    }
    public function find($id) {
        $data = $this->db->executeGet("select * from korisnik where idKorisnik = " . $id);

        if(!count($data)) {
            return null;
        }

        return $data[0];
    }

    public function findByUsernameAndPassword($username, $password) {
        $data = $this->db->executeGet("select * from korisnik where
        korisnicko = '" . $username . "' AND lozinka = '" . md5($password
) . "'");
        // lozinka = '" . md5($password) . "' AND korisnicko_ime = '" . $u
sername . "'");

        if(!count($data)) {
            return null;
        }

        return $data[0];
    }
}
```


Soba.php

```
<?php
namespace App\Models;

class Soba {
    private $db;

    public function __construct(DB $db){
        $this->db = $db;
    }

    public function getAllEavaileableRooms(){
        return $this->db->executeGet("SELECT * FROM soba");
    }

    public function getAllCategories(){
        return $this->db->executeGet("SELECT * FROM kategorija");
    }

    public function getServices(){
        return $this->db->executeGet("SELECT * FROM servis");
    }

    public function getRoomsAvailable($brojGostiju){
        return $this->db->executeQueryWithParams("SELECT * FROM soba WHERE broj_osoba = ?",[$brojGostiju]);
    }

    public function DodajSobu ($soba,$opis,$nova_putanja,$brojSoba,$brojOsoba,$cena,$zvezdice,$idKategorijeS) {

        $params = [$soba,$opis,$nova_putanja,$brojSoba,$brojOsoba,$cena,$zvezdice,$idKategorijeS];

        $query = "INSERT INTO soba(idSoba,nazivSoba,opis,src,broj_soba,broj_osoba,cena,zvezdice,idKategorija) values ( NULL,?, ?, ?, ?, ?, ?, ?, ?)";

        $this->db->executeInsert($query, $params);
    }

    public function FiltrirajSobu ($kategorije) {
        $k=implode( " ", $kategorije );

        $query = "SELECT * FROM soba WHERE idKategorija = ? OR idKategorija= ? OR idKategorija=?";
        $params = [$k];
        $this->db->executeQueryWithParams($query, $params);
    }
}
```

```
public function getRoom($idS){
    return $this->db->executeQueryWithParams("SELECT * FROM soba WHERE idSoba = ?",[$idS]);
}
public function upisiRez($idKor,$idSobe,$od,$out){
    $params = [$idKor,$idSobe,$od,$out];

    $query = "INSERT INTO rezervacija values(NULL,? ,?, ?, ?)";

    $this->db->executeInsert($query, $params);
}
?>
```

Views

Tu se nalaze fiksni delovi strane, koji su u folderu `fixed` I `pages`.