

# DOKUMENTACIJA

## WEB PROGRAMIRANJE 2 I PHP1

<https://nekretninebre.000webhostapp.com/>

## Catalog

UVOD .....	5
Korišćeni programski jezici .....	5
Opis funkcionalnosti .....	5
Header i Footer .....	5
allApartments .....	5
openApartment .....	5
Login .....	5
Register .....	6
Logout .....	6
userProfile .....	6
Support .....	6
Organizacija .....	7
Mapa Sajta .....	7
Slike stranica i funkcionalnosti .....	8
allApartments .....	8
allApartmentsJS .....	10
openApartment .....	12
userProfile .....	14
addApartment .....	15
Register .....	15
Login .....	16
Support .....	16
Kodovi .....	16
Index.php .....	16
.htaccess .....	17
Controller .....	17
Controller.php .....	17
addApartment.php .....	17
AllRealEstate.php .....	19
City.php .....	20
FilterAllAaprtments.php .....	21
GetAllApartments.js .....	23
Header.php .....	24
HeaderUser.php .....	24
Login.php .....	25
LogOut.php .....	27
OpenRealEstate.php .....	28
Register.php .....	29
ReturnAllApartmentsJS.php .....	31
Support.php .....	32
TypeOfRealEstate.php .....	34
UserProfile.php .....	34
Wishlist.php .....	36
GeneralFunctions .....	37
RegexValidation.php .....	37
checkIsSet.php .....	38
Model .....	39
Conn.php .....	39
Model.php .....	40
ModelAddApartment.php .....	40
ModelAddImagesApartment.php .....	43
ModelAllWishlist.php .....	46

ModelCity.php .....	47
ModelFilterAllApartments.php .....	49
ModelHeader.php .....	52
ModelListOfRE.php .....	54
ModelLogin.php .....	55
ModelRegister.php .....	57
ModelRetriveOneApartment.php .....	58
ModelRetriveOneImage.php .....	60
ModelReturnAllApartmentsJS.php .....	61
ModelSupport.php .....	63
ModelTypeOfRealEstate.php .....	65
ModelUser.php .....	67
ModelUserProfile.php .....	68
ModelWishlist.php .....	69
Router .....	71
Router.php .....	71
Routes.php .....	72



# UVOD

## Korišćeni programski jezici

- HTML
- CSS
- JS
- PHP
- XML

## Opis funkcionalnosti

### Header i Footer

- Dinamičko ispisivanje, pojavljuje se polje za korisnički profil ako je korisnik ulogovan

### allApartments

- Dinamički ispis
- Filtriranje
- Paginacija

### openApartment

- Prikaz detalja nekretnine
- Dodavanje u listu želja ako je korisnik ulogovan
- Dinamička galerija slika
- Prikaz informacija o vlasniku ako je korisnik ulogovan
- Postavljanje podataka o otvorenoj nekretnini u localStorage

### Login

- Forma za logovanje korisnika sa validacijom
- Postavljanje sesije sa id-em korisnika i njegovom ulogom

## Register

- Polja za registraciju korisnika
- Dodavanje podataka u bazu
- Cuvanje podataka u LocalStorage-u

## Logout

- Brisanje sesije za id i role polje

## userProfile

- Prikaz osnovnih podataka
- Prikaz nekretnina čiji je korisnik vlasnik
- Prikaz liste želja

## Support

- Kontakt ka administratorima
- Broj polja se razlikuje u odnosu na to da li je korisnik ulogovan

# Organizacija

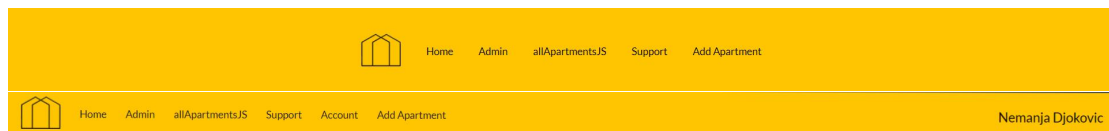
## Mapa Sajta

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
  <loc>https://nekretninebre.000webhostapp.com/</loc>
  <lastmod>2024-12-03</lastmod>
  <changefreq>daily</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>https://nekretninebre.000webhostapp.com/allApar
tmments</loc>
  <lastmod>2024-12-03</lastmod>
  <changefreq>daily</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>https://nekretninebre.000webhostapp.com/allApar
tmmentsJS</loc>
  <lastmod>2024-12-03</lastmod>
  <changefreq>daily</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>https://nekretninebre.000webhostapp.com/suppor
t</loc>
  <lastmod>2024-12-03</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.5</priority>
</url>
</urlset>
```

## Slike stranica i funkcionalnosti

### allApartments

Dinamički se ispisuje header i foote, salju GET zahtev ka serveru i dohvataju podatke iz tabele header U zavisnosti od uloge






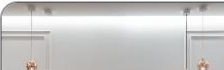
Filter na input event listener salje zahtev ka serveru sa odabranim parametrima i menja sadrzaj koji je vec bio sa novim

## All Apartments

The image shows a search form titled 'All Apartments'. It has a light gray background with rounded corners. The form contains three input fields: 'Search' with the placeholder text 'Seach list of apartments', 'Type of Real Estate' with a dropdown menu showing 'All' and a downward arrow, and 'Select city' with the placeholder text 'Select city'. Below these fields is a blue 'Submit' button.

Spisak nekretnina koje zadovoljavaju filtere se nalazi sa desne strane



	<b>Nehruova</b> Beograd Nehruova 66m <sup>2</sup>   number of rooms: 2	3485 €/m <sup>2</sup> <b>230000€</b>
	<b>Užice Kuca Nikole Pasica</b> Užice Nikole Pasica 54m <sup>2</sup>   number of rooms: 3	1037 €/m <sup>2</sup> <b>56000€</b>
	<b>Trg Uzice</b> Užice Trg Uzice 39m <sup>2</sup>   number of rooms: 2	1077 €/m <sup>2</sup> <b>42000€</b>
	<b>kuca u uzicu</b> Užice	

Paginacija se nalazi na dnu stranice



Vecina JS fajlova sadrzi obradu gresaka u vidu pojavljivanja pop up modala ili je ignorise ako nije neophodan podataka

## **allApartmentsJS**

Dohvata sve podatke sa servera o nekretninama I prikazuje ih kao u allApartments.  
Filtrira se u JavaScript-u, takodje cuva podatke o poslednje otvorenoj nekretnini  
Ostale funkcionalnosti su iste ili slične kao u allApartmentsu

## Search

Search list of apartments

## Type of Real Estate

- ☐ Flat
- ☐ Cottage
- ☐ House

## Select city

Select city

## Order by

Default



## Lastly seen

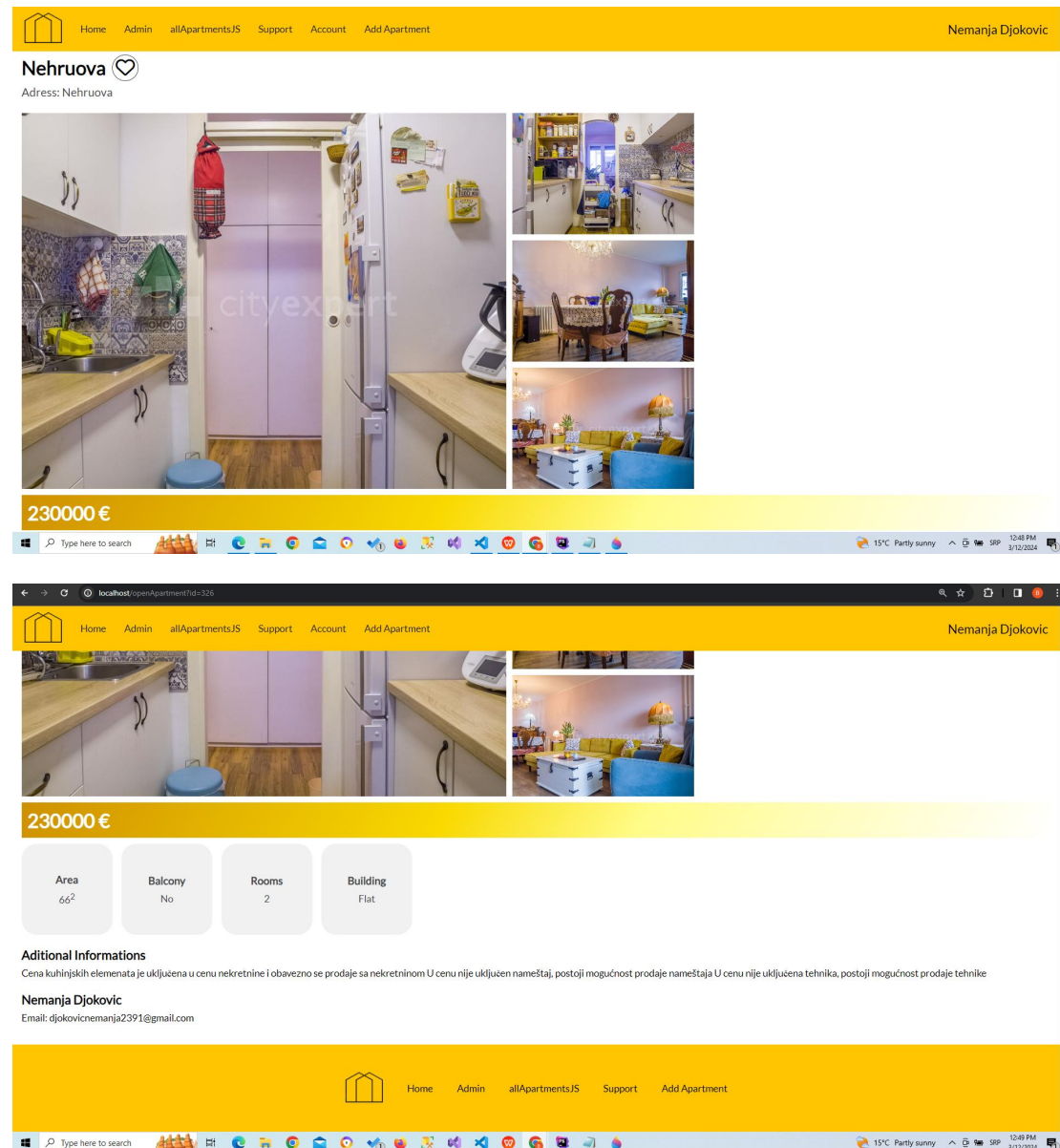


stan u Kragujevcu 90 m2

70000 €

# openApartment

Po učitavanju mogu se videti osnovni podaci o nekretnini kao i slike



U slučaju da korisnik nije ulogovan podaci u vlasniku neće biti vidljivi

Klikom na neku sliku otvara se carousel sa slikama

Klikom na srce serveru se šalje zahtev da doda u bazu podataka ovu objavu u listu želja, srce se ispunjava crvenom bojom.



## userProfile

Otvaranjem korisničkog profile prikazuju se podaci o korisniku kao i nekretnine koje su na prodaju a u njegovom vlasništvu.

U slučaju da je profil vaš prikazaće se i lista želja.

### Nemanja Djokovic

djokovicnemanja2391@gmail.com

#### List Of Real Estates

<b>Savski venac, Dedinje, Heroja Milana Tepića, 4.0</b> Address: Heroja Milana Tepića 267000€
<b>Palilula, Pančevački put, Pančevački put</b> Address: Pančevački Put 46000€
<b>Palilula, Pančevački put, Pančevački put</b> Address: Pančevački Put 46000€
<b>Karaburma, Stan</b> Address: Mije Kovacevica 7 350000€
<b>Trosoban LUX namešten stan sa garažom kod Hrama</b> Address: Branicevska 365000€
<b>namešten stan kod Hrama</b> Address: Branicevska 160000€
<b>Stan Beograd</b> Address: Branicevska 160000€
<b>Stan Beograd</b> Address: Branicevska 160000€

<b>kuca u uzicu</b> Address: Mihajla Pupina 130000€
---

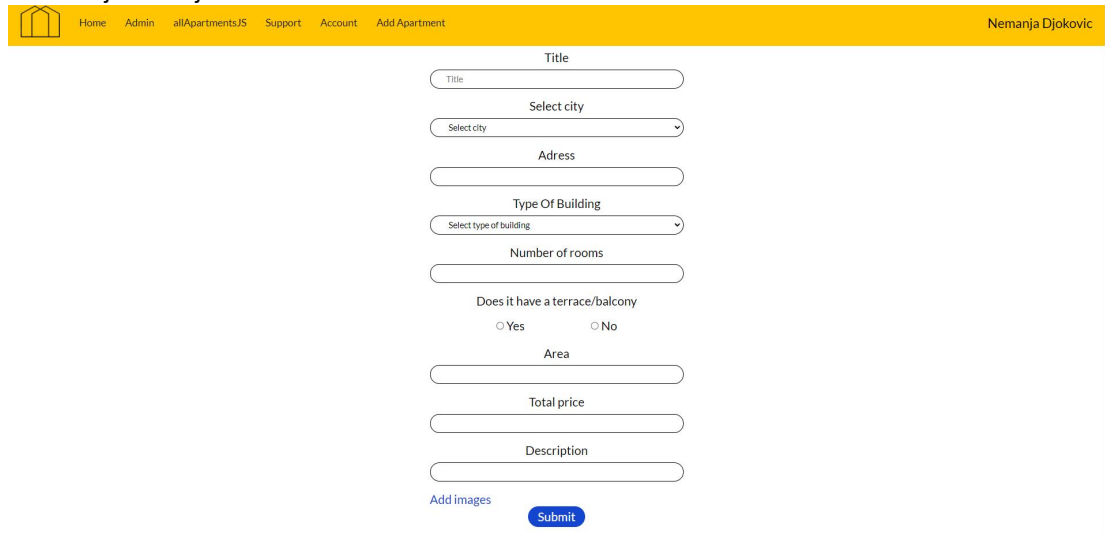
#### Wishlist

Stan Beograd
--------------

## addApartment

Sadrži polja za opisivanje nekretnina, dodavanje grada i tipa nekretnina kao i slike

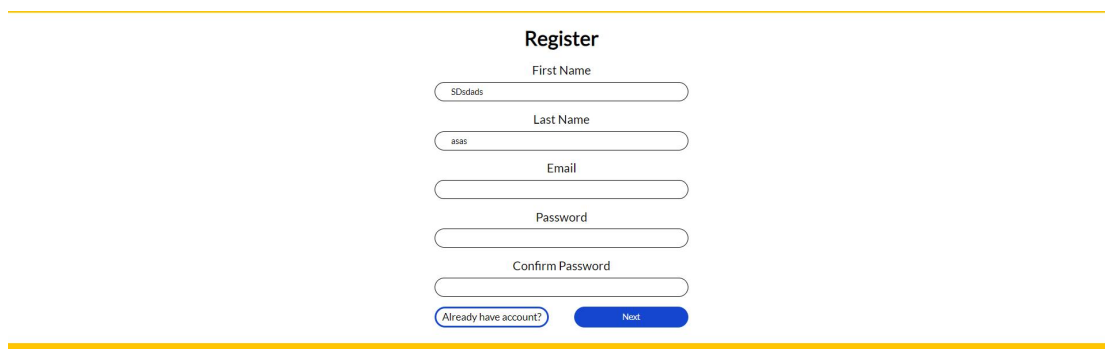
Validiranje i sa klijentske i serverske strane



The screenshot shows a web application interface for adding a new apartment. The header is yellow with a home icon, navigation links (Home, Admin, allApartmentsJS, Support, Account, Add Apartment), and the user name 'Nemanja Djokovic'. The form is centered and contains the following fields: Title, Select city (dropdown), Address, Type Of Building (dropdown), Number of rooms, Does it have a terrace/balcony (radio buttons for Yes/No), Area, Total price, and Description. There is an 'Add images' link and a 'Submit' button at the bottom.

## Register

Validira podatke na obe strane i upisuje u bazu u slucaju da su svi uslovi ispunjeni.  
Hashuje lozinku.



The screenshot shows a web application interface for registering a new user. The form is titled 'Register' and contains the following fields: First Name, Last Name, Email, Password, and Confirm Password. There is a link 'Already have account?' and a 'Next' button at the bottom. The form is set against a white background with a yellow header and footer.

## Login

Proverava validnost podataka, i vrši proveru da li postoji email sa datom lozinkom i obaveštava korisnika u uspehu akcije.

**Login**

Email

Password

Login

## Support

---

**Support**

First Name

Last Name

Email

Tittle

Content

Submit

Korisnik može da koontaktira administratore i postavi pitanje/problem, u slučaju da je korisnik ulogovan nece mu se pokazati polje za ime prezime i email.

## Kodovi

### Index.php

```
<?php
session_start();

require_once 'vendor/autoload.php';
require 'Router/Routes.php';
```



## .htaccess

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-f  
RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^([^\/]+)/? index.php?url=\$1 [L,QSA]  
RewriteRule ^(.+)/\$ /\$1 [L,R=301]

## Controller

### Controller.php

```
<?php

namespace App\Controller;
abstract class Controller
{
    public abstract static function Page();
}
```

### addApartment.php

```
<?php

namespace App\Controller;
use App\View\ViewAddApartment;
use App\Model\ModelAddApartment;
use App\Model\ModelCity;
use App\GeneralFunction\RegexValidation;
use App\Model\ModelTypeOfRealEstate;
use App\Model\ModelAddImagesApartment;

include_once 'GeneralFunctions/checkIsSet.php';
class AddApartment extends Controller
```

```
{
    public static function Page()
    {
        if(isset($_SESSION["id"])) {
            if ($_SERVER["REQUEST_METHOD"] == "GET") {
                ViewAddApartment::View();

            } else if ($_SERVER["REQUEST_METHOD"] ==
"POST") {
                var_dump($_FILES);
```

```
                $errors = 0;
                extract($_POST);
                if (!checkIfSet($city, $title,
$dodatniopis, $objectType, $adresa, $kvadratura, $cena,
$terasa)) {
                    http_response_code(403);
                    return;
                }
```

```
                if (isset($_SESSION["id"])) {
                    if (!ModelCity::checkCityInDB($city)
&& RegexValidation::minlen($title, 3) &&
RegexValidation::minlen($dodatniopis, 15) &&
ModelTypeOfRealEstate::checkType
eOfObject($objectType) &&
RegexValidation::minlen($adresa, 3) &&
RegexValidation::minnumber($kvadratura, 0) &&
RegexValidation::minnumber($cen
a, 0) && ($terasa == "Da" || $terasa == "No")) {
                        $errors++;
                    }
                }
```

```
                if (!$errors) {
```

```
                    $model = new
ModelAddApartment($title, $city, $adresa, $objectType,
```

```

$numberrooms, $terasa, $kvadratura, $cena, $dodatniopis,
$_SESSION["id"]);

$res = $model->Query();

        if ($res[0] == 1) {
            $modelimages = new
ModelAddImagesApartment((int)$res[1], $_FILES);
            $modelimages->Query();

http_response_code(200);

        } else if ($res[0] == 2 ||
$res[0] == 0) {

            http_response_code(503);
        }

    }

        } else {
            http_response_code(420);
        }

    }

}
else{
    header("Location: login");
}

}
}

```

## AllRealEstate.php

```

<?php

namespace App\Controller;
use App\View\ViewAllApartments;

```

```

class AllRealEstate extends Controller
{
    public static function Page()
    {
        if($_SERVER["REQUEST_METHOD"] == "GET"){

            ViewAllApartments::View();

        }
    }
}

```

## City.php

```

<?php

namespace App\Controller;
use App\Model\ModelCity;
class City extends Controller
{

    public static function Page()
    {
        $model = new ModelCity();

        if($res = $model->Query()){
            echo json_encode($res);

        }
        else{
            http_response_code(503);

        }
    }
}

```

## FilterAllAaprtments.php

```
<?php

namespace App\Controller;
use App\Model\ModelFilterAllApartments;
use App\Model\ModelCity;
use App\Model\ModelRetriveOneImage;
use App\Model\ModelTypeOfRealEstate;

require_once "GeneralFunctions/checkIsSet.php";
class FilterAllApartments extends Controller
{
    public static function Page()
    {
        try{
            $page = NULL;
            $city =NULL;
            $search = NULL;
            $typeObject = NULL;
            if(isset($_GET["city"]))
            {
                if(strlen($_GET["city"])>0){
                    $city = is_array($_GET["city"])
                        ? explode(",", $_GET["city"])
                        : $_GET["city"];
                    if (is_array($city)) {
                        foreach ($city as $row) {
                            if
(!ModelCity::checkCityInDB($row)) continue;
                        }
                    } else {
                        if
(!ModelCity::checkCityInDB($city)) {

                        }
                    }
                }
            }
        }
    }
}
```

```

    }
    if(isset($_GET["typeOfObject"])){
        if (
            ModelTypeOfRealEstate::checkTypeOfObject($_GET["typeOfObject"])
        ) {
            $typeObject = $_GET["typeOfObject"];
        }
    }
    if(isset($_GET["search"])){
        $search =
        htmlspecialchars($_GET["search"]);
    }
    if(isset($_GET["page"])){
        $page=$_GET["page"];
    }
    else{
        $page=0;
    }

```

```

        $model = new ModelFilterAllApartments(
            $city,
            $typeObject,
            $search, $page
        );
        $num_pages = $model->numbRows();
        $model = $model->Query();
        if ($model) {
            foreach ($model as &$amp;row) {
                $row["city_name"] =
                ModelCity::nameOfTheCity($row["city"]);
                $modelImage = new
                ModelRetriveOneImage($row["id"]);
                $row["img_url"] = $modelImage->Query();
            }
        }
    }

```

```

    http_response_code(200);

```

```

        echo(json_encode(["pages" => $model,
'num_pages'=> ceil($num_pages/20 )]));
    }
    catch(\Exception $e){
        echo "error lol";
    }

```

```

    }
}

```

## GetAllApartments.js

```
<?php
```

```

namespace App\Controller;
use App\View\ViewGetAllApartmentsJS;

```

```

class GetAllApartmentsJS extends Controller
{
    public static function Page()
    {
        try{

```

```
ViewGetAllApartmentsJS::View();
```

```

        }
        catch(\Exception $e){
            echo "error";
        }

```

```

    }
}

```

## Header.php

```
<?php

namespace App\Controller;
use App\Model\ModelHeader;

class Header extends Controller
{

    public static function Page()
    {

        $model = new ModelHeader();

        if($res = $model->Query()) {

            echo json_encode($res);
            http_response_code(200);
        }
        else{
            echo "error";
            http_response_code(400);
        }
    }
}
```

## HeaderUser.php

```
<?php

namespace App\Controller;
use App\Model\ModelUser;

class HeaderUser extends Controller
{

    public static function Page()
```



```

        {
            if($_SERVER["REQUEST_METHOD"]=="GET"){
                try {
                    if(isset($_SESSION["id"])){
                        $userModel = new
ModelUser($_SESSION["id"]);

                        http_response_code(200);
                        echo json_encode($userModel->Query());
                    }
                    else{
                        echo json_encode(["err"=>"err"]);
                        http_response_code(425);
                    }
                }catch(\Error $e) {
                    echo $e->getMessage();
                }
            }
            http_response_code(425);
        }
    }
}

```

## Login.php

```

<?php

namespace App\Controller;
use App\Model\ModelLogin;
use App\View\ViewLogin;
use App\GeneralFunction\RegexValidation;
include_once 'GeneralFunctions/checkIsSet.php';
class Login extends Controller
{
    public static function Page()
    {

```

```

        if(!isset($_SESSION["id"])) {

            if ($_SERVER["REQUEST_METHOD"] == "POST") {
                $dataReceived =
json_decode(file_get_contents('php://input'), true);

                if(!checkIfSet($dataReceived["email"],
$dataReceived["password"])){

                    echo "code:7";
                    http_response_code(403);
                    return;
                }

                $email =
htmlspecialchars($dataReceived["email"], ENT_QUOTES,
'UTF-8');

                $password =
htmlspecialchars($dataReceived["password"], ENT_QUOTES,
'UTF-8');

                if (RegexValidation::email($email)
&& RegexValidation::password($password)) {
                    $obj = new ModelLogin($email,
$password);

                    if ($session = $obj->Query()) {
                        if ( $session!=
"code:pwormailnomatch" && $session!= false ) {
                            if($session ===
"inactive") {

                                http_response_code(
401);

                                die();
                            }else{
                                $_SESSION["id"] =
$session["user_id"];

                                $_SESSION["role"] =
$session["role"];

                                echo "code:4";

```

```
                http_response_code(
200);
            }
        }
```

```
            } else if ($session ===
"code:pwormailnomatch"){
                echo $session;
                http_response_code(406);
            }
        }
        else {
            echo "code:6";
            http_response_code(503);
        }
    }
}
```

```
        }
        else {
            echo "code:wrongformat";
            http_response_code(406);
        }
    }
}
```

```
        }
        else if($_SERVER["REQUEST_METHOD"] == "GET")
{
            ViewLogin::View();
        }
    }
    else{
        header("Location: home");
    }
}
```

```
    }
}
```

## LogOut.php

```
<?php
```

```

namespace App\Controller;

class LogOut extends Controller
{
    public static function Page()
    {
        if(isset($_SESSION["id"])){
            unset($_SESSION["id"]);
        }
        header("Location: login");
    }
}

```

## OpenRealEstate.php

```

<?php

namespace App\Controller;
use App\Model\ModelRetriveOneApartment;
use App\View\ViewOpenRealEstate;
use App\Model\ModelWishlist;
use App\Model\ModelCity;
include_once "GeneralFunctions/checkIsSet.php";

```

```

class OpenRealEstate extends Controller
{
    public static function Page()
    {
        if(checkIfSet($_GET["id"])){
            $model = new
ModelRetriveOneApartment($_GET["id"]);
            $model = $model->Query();
            extract($model);
            $wishlisted = false;
            if(isset($_SESSION["id"])){
                $modelWishList = new
ModelWishlist($id, $_SESSION["id"]);

```

```
        $wishlisted = $modelWishList->isWishlisted();
    }
}
```

```
        $city = ModelCity::nameOfTheCity($city);
        if($model !== false){
            $view = new ViewOpenRealEstate($id,
            $wishlisted, $title, $city, $adress, $typeObject,
            $numberOfRooms, $terrace, $area, $price, $description,
            $owner,$typeOfObjectName, $f_name, $l_name,
            $email, $images??NULL);
        }
    }
}
```

```
        echo $view->View1();
    }
}
```

```
}
```

```
    }
}
```

## Register.php

```
<?php

namespace App\Controller;
use App\View\ViewRegister;
use App\GeneralFunction\RegexValidation;
use App\Model\ModelRegister;
include_once 'GeneralFunctions/checkIsSet.php';
```

```
class Register extends Controller
{
    public static function Page()
    {
    }
```

```

        if (isset($_SESSION["id"])) {
            header("Location: login");
        }
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $dataReceived =
json_decode(file_get_contents("php://input"), true);
            if(!checkIfSet($dataReceived["fname"],$data
Received["lname"], $dataReceived["email"],
$dataReceived["password"],$dataReceived["passwordConf"]
)){
                http_response_code(403);
                return;
            }
            $fname = htmlspecialchars(
                $dataReceived["fname"],
                ENT_QUOTES,
                "UTF-8"
            );
            $lname = htmlspecialchars(
                $dataReceived["lname"],
                ENT_QUOTES,
                "UTF-8"
            );
            $email = htmlspecialchars(
                $dataReceived["email"],
                ENT_QUOTES,
                "UTF-8"
            );
            $password = htmlspecialchars(
                $dataReceived["password"],
                ENT_QUOTES,
                "UTF-8"
            );
            $password_conf = htmlspecialchars(
                $dataReceived["passwordConf"],
                ENT_QUOTES,
                "UTF-8"
            );
        }
    }
}

```

```

        if (
            RegexValidation::email($email) &&
            RegexValidation::ime($fname) &&
            RegexValidation::ime($lname) &&
            RegexValidation::password($password,
$password_conf)
        ) {
            $obj = new ModelRegister($fname, $lname,
$email, $password);
            $res = $obj->Query();
            var_dump($res);
            if ($res === true) {
                http_response_code(201);
            } elseif ($res === false) {
                http_response_code(503);
            } else {
                http_response_code(409);
            }
        } else {
            http_response_code(406);
        }
    }
    else if ($_SERVER["REQUEST_METHOD"] == "GET"){
        ViewRegister::View();
    }
}
}

```

## ReturnAllApartmentsJS.php

```

<?php

namespace App\Controller;
use App\Model\ModelReturnAllApartmentsJS;
class ReturnAllApartmentsJS extends Controller
{
    public static function Page()

```

```

{
    if($_SERVER["REQUEST_METHOD"]=="GET"){
        $model = new ModelReturnAllApartmentsJS();
        echo json_encode($model->Query());
    }
    else{
        die();
    }
}

```

```

}

```

## Support.php

```

<?php

namespace App\Controller;
use App\View\ViewSupport;
use App\Model\ModelSupport;
require_once "GeneralFunctions/checkIsSet.php";
class Support extends Controller
{

```

```

    public static function Page()
    {
        if($_SERVER["REQUEST_METHOD"] == "GET"){

```

```

            $view = new
ViewSupport(isset($_SESSION["id"]));
            $view->View1();
        }
        else if($_SERVER["REQUEST_METHOD"] == "POST"){
            if(checkIfSetPost("titleSupport",
"contentSupport")){
                $title =
htmlspecialchars($_POST["titleSupport"]);
                $content =
htmlspecialchars($_POST["contentSupport"]);

```



```

        $model = NULL;
        if(isset($_SESSION["id"]) ){
            $logged = $_SESSION["id"];
            $model = new ModelSupport($logged,
$title, $content);

```

```

        }
        else{
            if(checkIfSetPost("fName", "lName",
"email")){
                $fname =
htmlspecialchars($_POST["fName"]);
                $lname =
htmlspecialchars($_POST["lName"]);
                $email =
htmlspecialchars($_POST["email"]);
                $model = new
ModelSupport($fname, $lname, $email, $title, $content);

```

```

            }else{
                echo "not succesful";
                die();
            }

```

```

        }
        if(
$model->Query()){
            echo "success";
        }
        else{
            echo "error";
        }
    }
    else{
        header("Location: home");
        die();
    }

```

```

    }

```

```
}  
}
```

## TypeOfRealEstate.php

```
<?php  
  
namespace App\Controller;  
use App\Model\ModelTypeOfRealEstate;  
class TypeOfRealEstate extends Controller  
{
```

```
    public static function Page()  
    {  
        $model = new ModelTypeOfRealEstate();  
        $res = $model->Query();  
        if($res){  
            http_response_code(200);  
            echo json_encode($res);  
        }  
        else{  
            http_response_code(503);  
            echo "error";  
        }  
    }  
}
```

```
}  
}
```

## UserProfile.php

```
<?php  
  
namespace App\Controller;  
use App\Model\ModelUserProfile;  
use App\Model\ModelListOfRE;  
use App\Model\ModelAllWishlist;  
use App\View\ViewUserProfile;
```

```
class UserProfile extends Controller
{
```

```
    public static function Page()
    {
        if($_SERVER["REQUEST_METHOD"]=="GET" ){
            if(!isset($_SESSION["id"])){
                header("Location: login");
            }
            $id = null;
            $model = null;
            $res = [];
            if(isset($_GET["id"])){
                $id = ($_GET["id"]);
            }
            else if(isset($_SESSION["id"])){
                $id = $_SESSION["id"];
            }
            else{
                die();
            }
            $model = new ModelUserProfile($id);
            $res["basic_info"] = $model->Query();
            $modelApartmentList = new
ModelListOfRE($id);
            $res["list_apartments"] =
$modelApartmentList->Query();
```

```
            if($_SESSION["id"] == $id){
                $modelWishlist = new
ModelAllWishlist($id);
                $res["wishlist"] = $modelWishlist-
>Query();
```

```
            }
            $view = new ViewUserProfile($res);
            $view->View1();
            /*
            echo "<pre>";
```

```
var_dump($res);  
echo "</pre>";*/
```

```
    }  
}  
}
```

## Wishlist.php

```
<?php  
  
namespace App\Controller;  
use App\Model\ModelWishlist;  
require_once "GeneralFunctions/checkIsSet.php";  
class Wishlist extends Controller  
{  
    public static function Page()  
    {
```

```
        if($_SERVER["REQUEST_METHOD"]=="POST"){
```

```
            $dataReceived =  
json_decode(file_get_contents('php://input'), true);  
            echo $_SESSION["role"];  
            if(isset($_SESSION["id"]) &&  
($_SESSION["role"] >=1) &&  
isset($dataReceived["id_post"])){
```

```
                $idpost = $dataReceived["id_post"];  
                $iduser = $_SESSION["id"];  
                $model = new ModelWishlist($idpost,  
$iduser);  
                if($model->Query()){  
                    echo "true1";  
                }  
                else{  
                    echo "false1";  
                }  
            }
```

```

    }
}
}
}

```

## GeneralFunctions

### RegexValidation.php

```

<?php
namespace App\GeneralFunction;
class RegexValidation
{
    public static function ime(string $ime):bool{

        $ime_reg = "/^[A-Z]{1}[a-z]{1,9}$/";
        return preg_match($ime_reg, $ime);
    }
    public static function email(string $mail):bool{

```

```

        $email_reg = "/[\w\d\_-\+\.\.]\@[\w\.\.]+$/";
        return preg_match($email_reg, $mail);
    }

```

```

    public static function password($pw1, $pw2 =
null):bool{

```

```

        $pw_reg1 = "/[\d]/";
        $pw_reg2 = "/[\w]/";
        $len = (strlen($pw1)>8) ? true :false;
        if($len && preg_match($pw_reg1, $pw1) &&
preg_match($pw_reg2,$pw1)){
            if($pw2 == NULL || $pw1 == $pw2) {
                return true;
            }
        }
    }

```

```
}  
return false;
```

```
}
```

```
public static function minlen($var, $len):bool{  
    if(strlen($var) >$len){  
        return true;  
    }  
    return false;  
}
```

```
public static function minnumber($var,  
$minimum,$maximum = NULL):bool{  
    if($var>$minimum) return true;  
    return false;  
}  
}
```

## checkIsSet.php

```
<?php  
  
function checkIfSetGet(){  
    $args = func_get_args();  
    foreach($args as $argument){  
        if(!isset($_GET[$argument])){  
            return false;  
        }  
    }  
}
```

```
    }  
    return true;  
}  
function checkIfSetPost(){  
    $args = func_get_args();  
    foreach($args as $argument){  
        if(!isset($_POST[$argument])){
```

```
        return false;
    }
```

```
    }
    return true;
}
function checkIfSet (){
    $args = func_get_args();
    foreach($args as $argument){
        if(!isset($argument)){
            return false;
        }
    }
}
```

```
    }
    return true;
}
```

## Model

### Conn.php

```
<?php

namespace App\Model;
use \PDO;
class Conn
{
    public $servername = "localhost";
    public $username = "korisnik";
    public $password = "korisnik";
    public $dbname = "phpapp";

    public function conn(){

        try {
            $pdo = new PDO("mysql:host=$this->servername;dbname=$this->dbname", $this->username,
            $this->password);
```

```
        $pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

```
        $pdo->  
>setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE,  
PDO::FETCH_ASSOC);
```

```
        return $pdo;  
    } catch (\PDOException $e) {
```

```
    }  
}  
}
```

## Model.php

```
<?php
```

```
namespace App\Model;
```

```
abstract class Model  
{  
    abstract function Query();
```

```
}
```

## ModelAddApartment.php

```
<?php
```

```
namespace App\Model;  
use App\Model\Conn;  
use PDO;
```

```
class ModelAddApartment extends Model  
{
```



```
private $title;  
private $city;  
private $adress;
```

```
private $typeObject;  
private $numberOfRooms;  
private $terrace;  
private $area;  
private $price;  
private $additionalDescription;  
private $owner;  
private $return = array();  
public function  
__construct($title,$city,$adress,$typeObject,$numberOfR  
ooms,$terrace,$area,$price,$additionalDescription,  
$owner){  
    $this->title = $title;  
    $this->city = $city;  
    $this->adress = $adress;  
    $this->typeObject=$typeObject;  
    $this->numberOfRooms = $numberOfRooms;  
    if($terrace == "Da"){  
        $this->terrace=true;  
    }else{  
        $this->terrace =false;  
    }  
}
```

```
    $this->area = $area;  
    $this->price=$price;  
    $this->additionalDescription =  
$additionalDescription;  
    $this->owner = $owner;  
}  
public function Query()  
{  
    $sql = "INSERT INTO realestate(title, city,  
adress, typeObject,numberOfRooms, terrace, area, price,  
description, owner)
```

## VALUES

```
(:title, :city, :adress,:typeObject,:numberOfRooms, :terrace, :area,:price,:description, :owner) ";  
$conn = new Conn();
```

```
try {  
    if ($conn = $conn->conn()) {  
        $conn->beginTransaction();  
        if ($stmt = $conn->prepare($sql)) {  
            $stmt->bindParam(':title', $this->title);  
            $stmt->bindParam(":city", $this->city);  
            $stmt->bindParam(':adress', $this->adress);  
            $stmt->bindParam(":typeObject", $this->typeObject);  
            $stmt->bindParam(":numberOfRooms", $this->numberOfRooms);  
            $stmt->bindParam(":terrace", $this->terrace);  
            $stmt->bindParam(":area", $this->area);  
            $stmt->bindParam(":price", $this->price);  
            $stmt->bindParam(":description", $this->additionalDescription);  
            $stmt->bindParam(":owner", $this->owner);  
            if ($stmt->execute()) {  
                $return[] = 1;  
                $return[] = $conn->lastInsertId();  
                $conn->commit();  
                return $return;  
            }  
            else{  
                $return[] = 2;  
                $return[] = NULL;  
            }  
        }  
    }  
}
```

```
$conn->rollBack();  
return $return;
```

```
    }  
    }  
    }  
}catch(\PDOException $e){  
    $return[]=0;  
    $return[] = NULL;  
    $conn->rollBack();  
    echo $e->getMessage();  
    $return[] = $return;  
    return $return;
```

```
    }  
    }  
}
```

## ModelAddImagesApartment.php

```
<?php  
  
namespace App\Model;  
use PDO;  
use App\Model\Conn;  
class ModelAddImagesApartment extends Model  
{
```

```
    private static $allowedTypeOfFiles  
=["image/png","image/jpeg","image/jpg"];  
    private static $imageFolder = "images/";  
    private $id;  
    private $array_images_names=[];  
    private $array_tmp_location=[];  
    private $array_tmp_extension=[];  
    private $array_errors=[];  
    public function __construct($id,  
$assoc_array_images){
```

```

        $this->id = $id;
        $count =
count($assoc_array_images["images"]["name"]);
        for($i = 0; $i<$count;$i++){
            if($assoc_array_images["images"]["error"][$i] === 0 &&
in_array($assoc_array_images["images"]["type"][$i],
ModelAddImagesApartment::$allowedTypeOfFiles)){

```

```

            $newName = "";
            for($x = 0; $x
<strlen( $assoc_array_images["images"]["name"][$i]);$x+
+ ){
                $character =
substr($assoc_array_images["images"]["name"][$i], $x,
1);
                if($character!="\n" &&
$character!="\t" && $character!=" "){
                    $newName.=$character;
                }
                else{
                    continue;
                }
            }
            $this->array_images_names[] =$newName;
            $this-
>array_tmp_location[]=$assoc_array_images["images"]["tm
p_name"][$i];

```

```

        }

```

```

    }
}
public function Query()
{
    $sql = "INSERT INTO realestate_image(location,
id_post,alt) VALUES (:name, :id_post, :alt)";

```

```

    $conn = new Conn();
    try{

```

```

        if($conn = $conn->conn()){
            for($i = 0; $i<count($this->array_images_names);$i++){
                $alt = "Image_Of_Apartment_".($i+1);
                if
(!file_exists(ModelAddImagesApartment::$imageFolder)) {
                    mkdir(ModelAddImagesApartment::$imageFolder, 0777, true);
                }
                if($stmt=$conn->prepare($sql)){
                    $newName = uniqid('', true) .
                    '_' . time() . $this->array_images_names[$i];
                    $fullPathNewName =
                    ModelAddImagesApartment::$imageFolder.$newName;
                    $stmt->bindParam(":name",
                    $fullPathNewName);
                    $stmt->bindParam(":id_post",
                    $this->id);
                    $stmt->bindParam(":alt", $alt);
                    if($stmt->execute()){
                        move_uploaded_file($this->array_tmp_location[$i], $fullPathNewName);
                    }
                    $stmt=NULL;

```

```

                }
            }
        }

```

```

    }catch ( \PDOException $e){
        echo $e;
    }

```

```

    }

```

```

}

```

## ModelAllWishlist.php

```
<?php

namespace App\Model;

class ModelAllWishlist extends Model
{
    private $iduser;

    public function __construct( $iduser){

        $this->iduser = $iduser;
    }
    public function Query()
    {
        $sql = "SELECT * FROM wishlist w INNER JOIN
realestate r ON w.realestate_id = r.id WHERE w.user_id
= :id";
        $conn = new Conn();
        try{
            if($conn = $conn->conn()){
                if($stmt=$conn->prepare($sql)){
                    $stmt->bindParam(":id", $this-
>iduser);

                    if($stmt->execute()){
                        return $stmt->fetchAll();
                    }
                }
            }
            return false;
        }
        catch(\PDOException $e){
            return false;
        }
    }
}
```

## ModelCity.php

```
<?php

namespace App\Model;
use App\Model\Conn;
use PDO;
class ModelCity extends Model
{
    public function Query()
    {
        $sql = "SELECT zip, city FROM city ";
        $conn = new Conn();
        $conn = $conn->conn();
        if($stmt = $conn->prepare($sql)){
            if($stmt->execute()){
                $return = [];
                $rows = $stmt-
>fetchAll(PDO::FETCH_ASSOC);
                foreach($rows as $row){
                    $return[] = ["zip"=>$row["zip"],
"city"=>$row["city"]];
                }
                return $return;
            }
        }
        return 0;
    }
}

public static function checkCityInDB($zip){
    $sql = "SELECT zip FROM city WHERE zip = :zip";

    $conn = new Conn();
    if( $conn = $conn->conn()){
        if($stmt = $conn->prepare($sql)){
            $stmt->bindParam(":zip", $zip);
            if($stmt->execute()){
```

```

        $row = $stmt->fetch();
        if( isset($row["zip"])
&& $row["zip"] !== NULL){
            return true;
        }
        else{
            return false;
        }
    }
}
return false;
}
public static function nameOfTheCity($zip){
    $sql = "SELECT city FROM city WHERE zip = :zip";

```

```

$conn = new Conn();
if( $conn = $conn->conn()){
    if($stmt = $conn->prepare($sql)){
        $stmt->bindParam(":zip", $zip);
        if($stmt->execute()){
            $row = $stmt->fetch();

```

```

            if( $stmt->rowCount() == 1 ){
                return $row["city"];
            }
            else{
                return false;
            }
        }
    }
}
return false;
}
}

```



## ModelFilterAllApartments.php

```
<?php
```

```
namespace App\Model;  
use App\Model\Conn;
```

```
class ModelFilterAllApartments extends Model  
{  
    private $city;  
    private $typeOfObject;  
    private $search;  
    private $page;  
    public function __construct($city =NULL,  
$typeOfObject =NULL, $search =NULL, $page = 0){  
        if(is_array($city)){  
            $this->city = implode(",", $city);  
        }  
        else if(is_string($city) && strlen($city)>1){  
            $this->city = $city;  
        }  
        else{  
            $this->city =NULL;  
        }  
        if(isset($typeOfObject)){  
            $this->typeOfObject = $typeOfObject;  
        }else{  
            $this->typeOfObject = NULL;  
        }  
    }  
}
```

```
        if(isset($search) && strlen($search)>0){  
            $this->search = $search;  
        }else{  
            $this->search =NULL;  
        }  
    }  
}
```

```
        if(isset($page)){  
            $this->page = $page * 20;  
        }  
    }  
}
```

```

    }else{
        $this->page = 20 ;
    }

```

```

    }
    public function Query()
    {
        try{
            $sql = "SELECT * FROM realestate\n";

```

```

                if(isset($this->city) && strlen($this->city)>2){
                    $sql.=" WHERE city IN ($this->city)\n";
                }
                if(isset($this->typeOfObject)){
                    if(isset($this->city) && ($this->typeOfObject<4 && $this->typeOfObject>0)){
                        $sql.= " AND typeObject = $this->typeOfObject\n";
                    }
                    else if( ($this->typeOfObject<4 && $this->typeOfObject>0)){
                        $sql.=" WHERE typeObject = $this->typeOfObject\n";
                    }

```

```

                }
                if(isset($this->search) && strlen($this->search)>0){
                    if( ((isset($this->city) && strlen($this->city)>2) || (isset($this->typeOfObject) && ($this->typeOfObject<4 && $this->typeOfObject>0) ) ) ){
                        $sql.= "AND title LIKE '%$this->search%'\n";
                    }
                    else if(isset($this->search) ){
                        $sql .= "WHERE title LIKE '%$this->search%'";
                    }

```

```

    }
    $sql.= " ORDER BY id DESC LIMIT 20 OFFSET
$this->page";

```

```

    $conn = new Conn();
    if($conn = $conn->conn()){
        if($stmt=$conn->prepare($sql)){
            if($stmt->execute()){
                $res =$stmt->fetchAll();
                return $res;
            }
        }
    }

```

```

    }
}
catch(\PDOException $e){
    echo $e->getMessage();
}

```

```

}
public function numbRows(){
    $sql = "SELECT * FROM realestate\n";

```

```

        if(isset($this->city) && strlen($this->city)>2){
            $sql.=" WHERE city IN ($this->city)\n";
        }
        if(isset($this->typeOfObject)){
            if(isset($this->city) && ($this->typeOfObject<4 && $this->typeOfObject>0)){
                $sql.= " AND typeObject = $this->typeOfObject\n";
            }
            else if( ($this->typeOfObject<4 && $this->typeOfObject>0)){
                $sql.=" WHERE typeObject = $this->typeOfObject\n";
            }

```

```
}
```

```
    }  
    if(isset($this->search) && strlen($this->search)>0){  
        if( ((isset($this->city) && strlen($this->city)>2) || (isset($this->typeOfObject) && ($this->typeOfObject<4 && $this->typeOfObject>0) ) ) ){  
            $sql.= "AND title LIKE '%$this->search%'\n";  
        }  
        else if(isset($this->search) ){  
            $sql .= "WHERE title LIKE '%$this->search%'";  
        }  
    }
```

```
    }  
    $conn = new Conn();  
    if($conn = $conn->conn()){  
        if($stmt=$conn->prepare($sql)){  
            if($stmt->execute()){
```

```
                return $stmt->rowCount();
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
}
```

## ModelHeader.php

```
<?php  
  
namespace App\Model;  
use App\Model\Conn;  
use PDO;  
class ModelHeader extends Model
```

```
{
```

```
public function Query()
{
    $sql = 'SELECT * FROM header';
    $conn = new Conn();
    $conn = $conn->conn();
    $stmt = $conn->prepare($sql);
    if($stmt->execute()){
        $array = [];
        $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
        foreach ($rows as $row) {
```

```
            if($row["parend_id"]===NULL){
                if($row["logged"] == NULL){
```

```
                    }else if( !isset($_SESSION["id"])
&& $row["logged"] == 1){
                        continue;
                    }else if(isset ($_SESSION["id"]) &&
$row["logged"]==2){
                        continue;
                    }
                    if($row["role"] == NULL){
```

```
                        }else if(isset($_SESSION["role"])
&& $row["role"] <= $_SESSION["role"]){
```

```
                            }
                            else {
                                continue;
                            }
                            $array[$row["id"]][] = ["title" =>
$row["title"], "href" => $row["href_link"]];
```

```
                        /* if(isset($_SESSION["id"])){
                            continue;
                        }
                        else {
```

```
                            }*/
```

```

        }
        else{
            if($row["logged"] == NULL){

                }else if( !isset($_SESSION["id"])
&& $row["logged"] == 1){
                    continue;
                }else if(isset ($_SESSION["id"]) &&
$row["logged"]==2){
                    continue;
                }
                if($row["role"] == NULL){

                    }else if(isset($_SESSION["role"])
&& $row["role"] <= $_SESSION["role"]){

                        }
                        else {
                            continue;
                        }
                        $array['child'.$row["parend_id"]][]
= ["title" =>$row["title"],
"href"=>$row["href_link"]];
                    }
                }
            }
            return $array;
        }
        return 0;
    }
}

```

## ModelListOfRE.php

```

<?php

namespace App\Model;

```

```

class ModelListOfRE extends Model
{
    private $id;
    public function __construct($id){
        $this->id=$id;
    }
    public function Query()
    {
        $sql = "SELECT * FROM realestate WHERE owner
= :id";
        $conn = new Conn();
        try{
            if($conn=$conn->conn()){
                if($stmt = $conn->prepare($sql)){
                    $stmt->bindParam(":id", $this->id);
                    if($stmt->execute()){
                        return $stmt->fetchAll();
                    }
                }
            }
            return false;
        }catch(\PDOException $e){
            return false;
        }
    }
}

```

## ModelLogin.php

```

<?php

namespace App\Model;
use App\Model\Conn;
use PDO;

class ModelLogin extends Model

```

```
{
    private $email;
    private $password;
    public function __construct($email, $password)
    {
        $this->email = $email;
        $this->password = $password;
    }
}
```

```
public function Query()
{
    $sql = "SELECT user_id, role, password,
is_active FROM users WHERE email = :email ";
    $conn = new Conn();
    if ($conn = $conn->conn()) {
        if($stmt = $conn->prepare($sql)){
            $stmt->bindParam(":email", $this->email,
PDO::PARAM_STR);
```

```
            if ($stmt->execute()) {
                $res = $stmt->fetch();
                if(!$res) return
"code:pwormailnomatch";
                if(!$res["is_active"]){
                    return "inactive";
                }
                if (password_verify($this->password,
$res["password"])) {
                    return array("user_id" =>
$res["user_id"], "role" =>$res["role"]);
                } else {
                    return "code:pwormailnomatch";
                }
            }
            else {
                return false;
            }
        }
    }
    else{
```



```
        return false;
    }
}
```

```
    }
    else {
        return false;
    }
}
}
```

## ModelRegister.php

```
<?php
```

```
namespace App\Model;
use \PDO;
use App\Model\Conn;
```

```
class ModelRegister extends Model
{
    private $firstName;
    private $lastName;
    private $email;
    private $password;
    public function __construct($fname,$lname,
$email,$pw){
        $this->firstName = $fname;
        $this->lastName = $lname;
        $this->email = $email;
        $this->password =
password_hash($pw,PASSWORD_ARGON2ID);
    }
    public function Query()
    {
```

```
        $sql = "INSERT INTO users(f_name,
l_name,email,password) Values
(:fname,:lname,:email,:password)";
```

```

        $conn = new Conn();
        if($conn = $conn->conn()){
            if($stmt = $conn->prepare($sql)){
                $stmt->bindParam(':fname',$this->firstName,PDO::PARAM_STR);
                $stmt->bindParam(':lname',$this->lastName,PDO::PARAM_STR);
                $stmt->bindParam(":email", $this->email,PDO::PARAM_STR);
                $stmt->bindParam(":password",$this->password, PDO::PARAM_STR);
                try{ if($stmt->execute()) {
                    return true;
                }
                else{
                    return "failure";
                }
            }
            catch(\PDOException){
                return "failure";
            }
        }
    }

```

```

    }

```

```

        return false;
    }
}

```

## ModelRetriveOneApartment.php

```

<?php

namespace App\Model;
use App\Model\Conn;
use App\Model\ModelTypeOfRealEstate;
class ModelRetriveOneApartment extends Model

```

```
{
    private $id;
    public function __construct($id){
        $this->id = $id;
    }
    public function Query()
    {
        $sql = "SELECT r.id, u.user_id, u.f_name,
u.l_name, u.email, r.title, r.city, r.adress,
r.typeObject, r.numberOfRooms, r.terrace, r.area,
r.price, r.description, r.owner FROM realestate r INNER
JOIN users u ON u.user_id = r.owner WHERE r.id=:id ";
```

```
$conn= new Conn();
if($conn = $conn->conn()){
    if($stmt=$conn->prepare($sql)){
        $stmt->bindParam(":id", $this->id);
```

```
        if($stmt->execute()){
            $result = $stmt->fetch();
            if ($result){
                $sql_images = "SELECT location,
alt FROM realestate_image WHERE id_post = :id_post";
                $conn = new Conn();
                if($conn = $conn->conn()){
                    if($stmt = $conn-
>prepare($sql_images)){
                        $stmt-
>bindParam(":id_post", $this->id);
                        if($stmt->execute()){
                            $image_res = $stmt-
>fetchAll();
                            foreach($image_res
as $row){
                                $result["images
"][] = ["location" => $row["location"],
"alt"=>$row["alt"]];
```

```
}
```

```

        if(ModelTypeOfRealEstate::checkTypeOfObject($result["typeObject"])){
            $result["typeName"] =ModelTypeOfRealEstate::checkTypeOfObject($result["typeObject"]);
        }

        return $result;
    }

    }
    }
    return false;
}

    }
    else return 0;
    }
    }
    }
    return false;
}
}
}

```

## ModelRetriveOneImage.php

```

<?php

namespace App\Model;

class ModelRetriveOneImage extends Model
{
    private $id ;
    public function __construct($id){
        $this->id = $id;
    }
    public function Query()
    {

```

```

        $sql = "SELECT location from realestate_image
WHERE id_post = :id LIMIT 1";
        try{
            $conn = new Conn();
            if($conn = $conn->conn()){
                if($stmt = $conn->prepare($sql)){
                    $stmt->bindParam(":id", $this->id);
                    if($stmt->execute()){
                        if($stmt->rowCount()){
                            $res = $stmt->fetch();
                            return $res["location"];
                        }
                        else{
                            return
'images/no_image.jpg';
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
}

```

```

    }
    return false;
}
catch(\PDOException){
    return false;
}
}

```

```

    }
}

```

## ModelReturnAllApartmentsJS.php

```

<?php

namespace App\Model;

class ModelReturnAllApartmentsJS extends Model

```

```

{
    public function Query()
    {
        $sql = "SELECT * FROM realestate ORDER BY id
DESC LIMIT 50 ";
        $conn = new Conn();
        try{
            if($conn = $conn->conn()){
                if($stmt = $conn->prepare($sql)){
                    if($stmt->execute()){
                        $res=$stmt->fetchAll();
                        $resreturn = array();
                        foreach($res as $row){
                            $model = new
ModelRetriveOneImage($row["id"]);
                            $resreturn[] = [
                                "id"=>$row["id"],
                                "title"=>$row["title"],
                                "adress"=>$row["adress"],
                                "city"=>[
                                    "citycode"
=>$row["city"],
                                    "cityname"=>ModelCi
ty::nameOfTheCity($row["city"])
                                ],
                                "typeObject"=>[
                                    "typeObject"=>$row[
"typeObject"],
                                    "typeObjectName"
=>ModelTypeOfRealEstate::checkTypeOfObject($row["typeOb
ject"])
                                ],
                                "price"=>$row["price"],
                                "numberOfRooms"=>$row["
numberOfRooms"],
                                "area"=>$row["area"],

```

```

                                "img_url"=>$model->Query()
                                ];
                                }
                                return $resreturn;
                                }
                                }
                                }
                                }
                                }catch(\PDOException $e){
                                return false;
                                }
                                }
                                }
                                }

```

## ModelSupport.php

```

<?php

namespace App\Model;

class ModelSupport extends Model
{
    private $fname;
    private $lname;
    private $email;
    private $title;
    private $content;
    private $userid;
    private $constructor;
    public function __construct(...$args){
        $count = count($args);
        match($count){
            5=> $this->notLogged(...$args),
            3=>$this->logged(...$args)

```

```
};
```

```
    }  
    private function notLogged($fname, $lname, $email,  
$title, $content){  
        $this->fname = $fname;  
        $this->lname = $lname;  
        $this->email=$email;  
        $this->title = $title;  
        $this->content = $content;  
        $this->constructor=1;  
    }  
    private function logged($userid, $title, $content){  
        $this->userid = $userid;  
        $this->title = $title;  
        $this->content = $content;  
        $this->constructor = 2;  
    }  
    public function Query()  
    {  
        $sql = null;  
        if($this->constructor==1){  
            $sql = "INSERT INTO support(fName, lName,  
email, title, content) VALUES  
(:fname, :lname, :email, :title, :content)";  
        }  
        else if($this->constructor==2){  
            $sql = "INSERT INTO support (id_user, title,  
content) values (:iduser, :title, :content)";  
        }else{  
            die();  
        }  
        try{  
            $conn = new Conn();  
            if($conn = $conn->conn()){  
                if($stmt = $conn->prepare($sql)){  
                    if($this->constructor==1){  
                        $stmt->bindParam(":fname",  
$this->fname);
```



```

        $stmt->bindParam(":lname",
$this->lname);
        $stmt->bindParam(":email",
$this->email);
        $stmt->bindParam(":title",
$this->title);
        $stmt->bindParam(":content",
$this->content);
    }
    else if($this->constructor==2){
        $stmt->bindParam(":iduser",
$this->userid);
        $stmt->bindParam(":title",
$this->title);
        $stmt->bindParam(":content",
$this->content);
    }
    if($stmt->execute()){
        return true;
    }
}
}
return false;
}catch(PDOException $e){
    echo $e->getMessage();
    return false;
}
}

```

```

        // TODO: Implement Query() method.
    }
}

```

## ModelTypeOfRealEstate.php

```
<?php
```

```

namespace App\Model;
use App\Model\Conn;
use PDO;
class ModelTypeOfRealEstate extends Model
{
    public function Query()
    {
        $sql = "select * from tip_objekta";
        $conn = new Conn();
        $conn = $conn->conn();
        if($stmt = $conn->prepare($sql)){
            if($stmt->execute()){
                $rows = $stmt-
>fetchAll(PDO::FETCH_ASSOC);
                $result =[];
                foreach($rows as $row){
                    $result[] = ["id"=> $row["id"],
"name"=>$row["naziv"]];
                }
                return $result;
            }
        }
    }
}

```

```

    }
    else{
        return 0;
    }
}

```

```

    }
    public static function checkTypeOfObject($id){
        $sql = "select * from tip_objekta WHERE id
= :id";
        $conn = new Conn();
        if($conn = $conn->conn()){
            if($stmt = $conn->prepare($sql)){
                $stmt->bindParam(":id", $id);
                if($stmt->execute()){
                    $res = $stmt->fetch();
                    if( gettype($res) !== "array" ){

```

```

        return false;
    }else if ($res["id"] ==
$id){ return $res["naziv"];}
        else return false;

```

```

    }
}
}
return false;

```

```

}
}

```

## ModelUser.php

```
<?php
```

```
namespace App\Model;
```

```

class ModelUser extends Model
{
    private $id;
    public function __construct($id)
    {
        $this->id = $id;
    }
    public function Query()
    {
        try{
            $sql = "SELECT user_id, f_name, l_name,
email FROM users WHERE user_id = :id";
            $conn = new Conn();
            if($conn = $conn->conn()) {
                if ($stmt = $conn->prepare(($sql))) {
                    $stmt->bindParam(":id", $this->id);
                    if( $stmt->execute()){
                        if($stmt->rowCount()==1){

```

```

        return $stmt->fetch();
    }
}
}
return false;

```

```

    }catch (\PDOException){
        return false;
    }

```

```

}

```

```

}

```

## ModelUserProfile.php

```

<?php

```

```

namespace App\Model;

```

```

class ModelUserProfile extends Model
{
    private $id;
    public function __construct($id)
    {
        $this->id = $id;
    }
    public function Query()
    {
        $sql = "SELECT f_name, l_name, email FROM users
WHERE user_id=:id";
        $conn = new Conn();
        try{
            if($conn=$conn->conn()){
                if($stmt=$conn->prepare($sql)){
                    $stmt->bindParam(":id", $this->id);

```

```
        if($stmt->execute()){  
            return $stmt->fetch();  
        }  
    }  
}
```

```
        }  
    }  
    return false;  
}  
catch(\PDOException $e){  
    return false;  
}
```

```
    }  
}
```

## ModelWishlist.php

```
<?php  
  
namespace App\Model;  
use App\Model\Conn;  
class ModelWishlist extends Model  
{  
    private $iduser;  
    private $idpost;  
    public function __construct($idpost, $iduser){  
        $this->idpost= $idpost;  
        $this->iduser = $iduser;  
    }  
    public function Query()  
    {  
        $sql = "INSERT INTO WISHLIST (user_id,  
realestate_id) VALUES( :iduser,:idpost) ";  
        $conn = new Conn();  
        try{  
            if($conn = $conn->conn()){  
                if($stmt= $conn->prepare($sql)){
```

```

        $stmt->bindParam(":idpost", $this->idpost);
        $stmt->bindParam(":iduser", $this->iduser);
        if($stmt->execute()){
            return array("done"=>"added");

```

```

        }else{
            $conn = NULL;
        }
    }
}
return array("done"=>"contact developers");
}
catch(\PDOException $e){
    $errorCode = $e->getCode();
    if($errorCode == "23000"){
        $sql = "DELETE FROM wishlist WHERE
user_id = :iduser AND realestate_id = :idpost";
        $conn2 = new Conn();
        if($conn2 = $conn2->conn()){
            if($stmt= $conn2->prepare($sql)){
                $stmt->bindParam(":idpost",
$this->idpost);
                $stmt->bindParam(":iduser",
$this->iduser);
                if($stmt->execute()){
                    return "Obrisan";

```

```

            }
        }
    }
    return false;
}
}

```

```

}
public function isWishlisted(){

```

```

        $sql = "SELECT * FROM wishlist WHERE
realestate_id = :idpost and user_id = :userid";
        try{
            $conn = new Conn();
            if($conn = $conn->conn()){
                if($stmt = $conn->prepare($sql)){
                    $stmt->bindParam(":idpost", $this-
>idpost);
                    $stmt->bindParam(":userid",$this-
>iduser);
                    if($stmt->execute()){
                        return $stmt->rowCount()==1;
                    }
                }
            }
        }catch(\PDOException $e){
            return false;
        }
    }
}

```

## Router

### Router.php

```

<?php

namespace App\Router;

class Router
{
    public static $validRoute = array();
}

```

```

public static function set($route, $method){
    self::$validRoute = $route;
    if(isset($_GET["url"])) {
        if ($_GET["url"] == $route) {
            $method->__invoke();

```

```

        }
    }else{
        if( $route == "allApartments"){
            $method->__invoke();

```

```

        }
    }

```

```

    }
}

```

## Routes.php

```

<?php

use App\Router\Router;
use App\Controller\Register;
use App\Controller>Login;
use App\Controller\LogOut;
use App\Controller\Header;
use App\Controller\AddApartment;
use App\Controller\City;
use App\Controller\TypeOfRealEstate;
use App\Model\ModelRetriveOneApartment;
use App\Controller\OpenRealEstate;
use App\Controller\AllRealEstate;
use App\Controller\FilterAllApartments;
use App\Controller\Wishlist;

```



```
use App\Model\ModelWishlist;
use App\Controller\Support;
use App\Controller\HeaderUser;
use App\Controller\UserProfile;
use App\Controller\GetAllApartmentsJS;
use App\Controller\ReturnAllApartmentsJS;
use App\Controller\AdminRegistration;
use App\Controller\AdminSupport;
use App\Controller\AdminRemoveUser;
USE App\Controller\AdminAddTypeObject;
Router::set("adminSupport",function() {
    AdminSupport::Page();
```

```
});
Router::set("AdminAddTypeObject",function() {
    AdminAddTypeObject::Page();
```

```
});
Router::set("adminRegistration",function() {
    AdminRegistration::Page();
```

```
});
```

```
Router::set("allApartments",function() {
    AllRealEstate::Page();
```

```
});
```

```
Router::set("userProfile", function(){
    UserProfile::Page();
});
Router::set("getApartmentsJS", function(){
    ReturnAllApartmentsJS::Page();
});
Router::set("allApartmentsJS", function(){
    GetAllApartmentsJS::Page();
});
Router::set("FilterAllApartments", function (){
    FilterAllApartments::Page();
});
```

```
Router::set("test", function (){
    $model = new ModelWishlist(189, $_SESSION["id"]);
    echo (int)$model->isWishlisted();
});
```

```
Router::set("support", function (){
    Support::Page();
});
Router::set("AdminRemoveUsers", function(){
    AdminRemoveUser::Page();
});
```

```
Router::set("loggedIcon", function (){
    HeaderUser::Page();
});
Router::set("openApartment", function(){
    OpenRealEstate::Page();
});
Router::set("wishlist", function(){
    Wishlist::Page();
});
Router::set("search", function(){
});
```

```
Router::set("typeOfRealEstate", function(){
    TypeOfRealEstate::Page();
});
Router::set("addApartment", function(){
    AddApartment::Page();
});
```

```
Router::set("city", function(){
    City::Page();
});
```

```
Router::set("home", function (){
    echo 'this is home page';
```

```
});
```

```
Router::set("register", function (){
```

```
    Register::Page();  
});
```

```
Router::set("login", function (){
```

```
    Login::Page();  
});
```

```
Router::set("logout", function(){  
    Logout::Page();  
});
```

```
Router::set("header", function(){
```

```
    Header::Page();
```

```
});
```

## View

### View.php

```
<?php
```

```
namespace App\View;
```

```
abstract class View
```

```
{  
    public abstract static function View();
```

```
}
```

### ViewAddApartment.php

```
<?php
```

```
namespace App\View;
```

```
abstract class View
{
    public abstract static function View();
}
```

## ViewAllApartments.php

```
<?php
```

```
namespace App\View;
```

```
class ViewAllApartments extends View
{
    public static function View()
    {
        require_once
__DIR__.'../public/html/allApartments.html';
    }
}
```

## ViewGetAllApartmentsJS.php

```
<?php
```

```
namespace App\View;
```

```
class ViewGetAllApartmentsJS extends View
{
    public static function View()
    {
```

```
        require_once
__DIR__."/public/html/allApartmentsJS.html";
    }
}
```

## ViewLogin.php

```
<?php

namespace App\View;

class ViewLogin extends View
{
    public static function View()
    {
        require_once __DIR__.'/public/html/login.html';
    }
}
```

## ViewOpenRealEstate.php

```
<?php

namespace App\View;

class ViewOpenRealEstate extends View
{
    private $id;
    private $wishlist;
    private $title;
    private $city;
    private $adress;
    private $typeObject;
    private $numberOfRooms;
    private $terrace;
```

```
private $area;  
private $price;
```

```
private $description;  
private $owner;  
private $images;  
private $typeOfObjectName;  
private $f_name;  
private $l_name;  
private $email;  
private $logged;
```

```
public function __construct($id,$wishlisted, $title,  
$city, $adress, $typeObject, $numberOfRooms, $terace,  
$area, $price, $description, $owner,$typeOfObjectName,  
$f_name, $l_name, $email, $images=NULL)
```

```
{  
    $this->f_name=$f_name;  
    $this->l_name = $l_name;  
    if(isset($_SESSION["id"])){  
        $this->email=$email;  
        $this->logged = true;  
    }  
    else{  
        $start = substr($email,0,3);  
        $stars = strpos($email, "@");  
        $this->email = $start."***".substr($email,  
$stars);  
        $this->logged=false;  
    }  
    if($wishlisted ){  
        $this->wishlisted = ' fa-solid active ';
```

```
}  
    else{  
        $this->wishlisted = " fa-regular ";  
    }  
}
```

```
$this->id = $id;  
$this->title = $title;
```

```
$this->city = $city;
$this->adress = $adress;
$this->typeObject = $typeObject;
$this->numberOfRooms = $numberOfRooms;
$this->typeOfObjectName=$typeOfObjectName;
```

```
$this->terrace = ($terace === 1) ? "Yes" : "No";
```

```
$this->area = $area;
$this->price = $price;
$this->description = $description;
$this->owner = $owner;
if($images){
    foreach ($images as $row) {
        $this->images[] = $row;
    }
}
```

```
}
```

```
public static function View()
{
}
```

```
public function View1()
{
    $html = '
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    ';
    $html .="<meta name='keywords' content='$this-
>adress, $this->title, $this->typeOfObjectName, $this-
>city, nekretnine' />";
    $html.=" <meta name='description'
content='$this->typeOfObjectName in $this->city, with
```

```

area of $this->area square meters, on $this->adress
street

                                Check best apartments on
Nekretnine.
    '/>";
        $html.='
            <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
            <script src="View/public/js/regex.js"
type="application/javascript"></script>
            <script src="View/public/js/lastClicked.js"
type="application/javascript"></script>
            <script
src="https://kit.fontawesome.com/104d21a00f.js"
type="application/javascript"
crossorigin="anonymous"></script>
            <script src="View/public/js/header.js"
type="application/javascript"></script>
            <script src="View/public/js/imageCarousel.js"
type="application/javascript"></script>
            <script
src="View/public/js/addToWishlist.js"
type="application/javascript"></script>
            <script></script>
            <title>' . $this->title . '</title>
</head>
<body>
<div id="realestatepost">
    <h1 id="title1">' . $this->title . ($this-
>logged?"<i id = 'wishlist' data-id = '$this->id'
class=' " . ( $this->wishlisted) . " fa-heart icon-
heart' > </i>" : "") . '</h1>
    <h2 id="adress">Adress: '.$this->adress.'</h2>
    <!-- Image block goes here -->
    <div id="images">';
        $i = 0;
        if($this->images!==null && count($this->images))
{

```



```

        foreach ($this->images as $row) {
            //$html .= '3? " d-
            none":"").' " ' . "image-id='$i'".' />';
            //$html .= "<div style='background-
            image:url(\".$row['location'].\")" . "' " . "class='".($i>3?
            " class=' d-none ' ":"") . (!$i? ' main ':'
            sidebar' . $i)."' " . "image-id='$i'". "></div>";
            $html .= "<div style='background-
            image:url(" . $row['location'] . ")' class='" . ($i >
            3 ? 'd-none' : '') . "' . ($i ? 'main' : 'sidebar' .
            $i) . "' image-id='$i'></div>";

```

```

            $i++;
        }
    }
    else{
        $html .= "<div style=\"background-image:
        url('/images/no_image.jpg')\" class='main' image-
        id='0'></div>";

```

```

    }
    $html .= '
    </div>
    <h2 id="price"><div id="textprice">' . $this-
    >price . ' &euro;</div></h2>
    <div id="basic_info">
        <span><span class="header">Area</span><span
        class="value">' . $this->area .
        '<sup>2</sup></span></span>
        <span><span class="header">Balcony</span><span
        class="value">' . $this->terrace . '</span></span>
        <span><span class="header">Rooms</span><span
        class="value">' . $this->numberOfRooms .
        '</span></span>

```

```

        <span><span class="header">Building</span><span
class="value">' . $this->typeOfObjectName .
'</span></span>
    </div>
    <div id="additionalDescription">
        <h2>Additional Informations</h2>
        <p>' . $this->description . '</p>
    </div>

    <div id="owner">
        <h1><a href="userProfile?id=' . $this->owner . '">' .
$this->f_name . ' ' . $this->l_name . '</a></h1>
        '; ($this->logged) ? $html.= ' <p>Email: <a
href="mailto:' . $this->email . '">' . $this-
>email . '</a></p>' : $html.= "<p>Email :$this-
>email<br/><a style='color:blue' href='login'>Login</a>
to see whole mail</p>". '
    </div>
</div>

```

```

</body>
</html>';

```

```

        return $html;
    }
}

```

## ViewRegister.php

```
<?php
```

```
namespace App\View;
```

```

class ViewRegister extends View
{
    public static function View()
    {
        require_once 'public/html/register.html';
    }
}

```

```
}  
}
```

## ViewSupport.php

```
<?php
```

```
namespace App\View;
```

```
class ViewSupport extends View  
{  
    private $logged;  
    public function __construct(  
        $logged  
    ){  
        $this->logged = $logged ;  
    }  
}
```

```
    public static function View()  
    {  
        require_once  
__DIR__.' /public/html/support.html';  
    }  
    public function View1(){  
        $html = '<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Support</title>  
    <link rel="stylesheet"  
href="View/public/css/style.css" type="text/css"/>  
    <script src="View/public/js/header.js"  
type="application/javascript"></script>  
    <script  
src="https://kit.fontawesome.com/104d21a00f.js"  
type="application/javascript"  
crossorigin="anonymous"></script>
```

```

</head>
<body>
<div id="support">
    <h1>Support</h1>
    <form action="support" method="POST"
id="firstForm">';
        if(!$this->logged){
            $html.= '<label for="fName">First Name</label>
<input type="text" id="fName" name="fName"/>
<label for="lName">Last Name</label>
<input type="text" id="lName" name="lName"/>
<label for="email">Email</label>
<input type="email" name="email" id="email"/>';

```

```
};
```

```

            $html.=' <label
for="titleSupport">Tittle</label>
<input type="text" name="titleSupport"
id="titleSupport"/>
<label for="contentSupport">Content</label>
<input type="text" name="contentSupport"
id="contentSupport"/>
<input type="submit" id="submit" name="submit"
class="noBorder"/>

```

```

</form></div>
        <script
src="View/public/js/regex.js" type="application/javasc
ript"></script>
        <script
src="View/public/js/support.js" type="application/java
script"></script>

```

```
</body>
```

```

</html>';
        echo $html;
    }

```

```
}
```

## ViewUserProfile.php

```
<?php
```

```
namespace App\View;
```

```
class ViewUserProfile extends View
{
    private $res;
    public function __construct($res)
    {
        $this->res = $res;
    }
    public static function View()
    {
```

```
}
```

```
    public function View1()
    {
        $html = " <!DOCTYPE html>
<html lang='en'>
<head>
    <meta charset='UTF-8'>
    <meta name='viewport' content='width=device-width,
initial-scale=1.0'>
    <title>User Profile - ". $this-
>res["basic_info"]["f_name"] ."</title>
    <script
src='https://kit.fontawesome.com/104d21a00f.js'
type='application/javascript'
crossorigin='anonymous'></script>
```

```

        <script src = 'View/public/js/header.js'
type='application/javascript'></script>
        <link rel='stylesheet'
href='View/public/css/style.css' type='text/css'>
</head>
<body>
<div id='holder'>
    <div id='basicInfo'>

";
        $html.="<h1>".$this->res["basic_info"]["f_name"]." ".$this->res["basic_info"]["l_name"]."</h1>";
        $html.="<p id='email'> ".$this->res["basic_info"]["email"]."</p></div>";
        $html.="<h2>List Of Real Estates</h2>";
        //real estate add
        $html .= "<div id='realestate'>";
        foreach($this->res["list_apartments"] as $row){
            $html.="<a
class='blockApartmentUserProfile'
href='openApartment?id=".$row["id"]."'"><h2>".$row["title"]."</h2>
            <p class='adress'>Adress:
            ".$row["adress"]."</p>
            <p class='price'>".$row["price"]."&euro;</p>

</a>";

```

```

        }
        $html.="</div>";
        if(isset($this->res["wishlist"])) ){
            $html.="<h2>Wishlist</h2>";
            $html .= "<div id='wishlist'>";
            foreach ($this->res["wishlist"] as $row) {
                $html .= "<a
class='blockApartmentUserProfile'
href='openApartment?id=" . $row["id"] . "'>".
                $row["title"] . "</a>";

```

```
    }  
}
```

```
$html.="</div></div></body></html>";  
echo $html;
```

```
    }  
}
```

## Public

### Public/css/style.css

```
header {  
    box-sizing: border-box;  
    padding: 0px 20px;  
    width: 100%;  
    position: sticky;  
    top: 0px;  
    left: 0px;  
    height: 74px;  
    background-color: #ffc500;  
    justify-content: space-between;  
    align-items: center;  
    display: flex;  
}  
header i.toggle {  
    display: none;  
}  
header a#user {  
    color: black;  
    font-size: 1.3rem;  
    text-decoration: none;  
}  
header ul {  
    list-style-type: none;  
    width: -moz-fit-content;
```

```
width: fit-content;
display: flex;
align-items: center;
justify-content: space-evenly;
}
header ul li {
  height: -moz-fit-content;
  height: fit-content;
  position: relative;
  display: flex;
  align-items: center;
  justify-content: space-evenly;
  padding: 5px 15px;
  box-sizing: border-box;
}
header ul li:first-child {
  padding-left: 0px !important;
}
header ul li a {
  color: rgb(27, 27, 27);
  text-decoration: none;
}
header ul li ul {
  left: 0px;
  top: 100%;
  background-color: aliceblue;
  border: 1px solid #c2c2c2;
  position: absolute;
  width: -moz-fit-content;
  width: fit-content;
  display: none;
  padding: 10px;
  box-sizing: border-box;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  z-index: 5;
  margin: 0px 15px;
```



```

}
header ul li ul li {
  display: block;
  text-align: left;
  width: 100%;
}
header ul li ul li:first-child {
  padding-left: 15px !important;
}
header ul li ul li a {
  display: block;
  text-align: left;
  width: 100%;
  color: rgb(27, 27, 27);
  text-decoration: none;
  white-space: nowrap;
}

footer {
  width: 100%;
  background-color: #ffc500;
  height: -moz-fit-content;
  height: fit-content;
  box-sizing: border-box;
  padding: 25px;
}
footer ul {
  margin: 0px auto;
  width: -moz-fit-content;
  width: fit-content;
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  list-style-type: none;
}
footer ul li {
  box-sizing: border-box;
  padding: 20px;
}

```

```
}
footer ul li a {
  color: black;
  text-decoration: none;
}
```

```
@font-face {
  font-family: Lato;
  src: url("../fonts/Lato-Regular.ttf");
}
* {
  margin: 0px;
  padding: 0px;
  font-family: Lato;
  box-sizing: border-box;
}
```

```
.d-flex {
  display: block;
  /*flex-direction: column;
  height: fit-content;*/
  top: 0px;
  left: 0px;
  background-color: #f1f1f1;
  width: 100%;
  position: absolute;
  z-index: 2;
}
```

```
i.fa-bars {
  position: absolute;
  top: 50%;
  left: 5%;
  transform: translate(-50%, -50%);
  z-index: 5;
}
```

```
form {
  display: block;
  width: 400px;
```

```
    box-sizing: border-box;
    margin: 0px auto;
}
form h1 {
    width: -moz-fit-content;
    width: fit-content;
    margin: 20px auto;
}
form #error {
    margin-bottom: 25px;
}
form .error {
    margin-top: 3px;
    color: red;
}
form input[type=text], form input[type=email], form
input[type=password], form select, form
input[type=number] {
    width: 100%;
    height: 30px;
    border-radius: 25px;
    border: 1px solid black;
    margin-top: 8px;
    padding: 0px 25px;
    box-sizing: border-box;
}
form input[type=file] {
    display: none;
}
form #file {
    background-color: transparent;
    border: 0px;
    color: #1446cf;
    text-align: left;
}
form label {
    width: 100%;
    display: block;
```

```

    text-align: center;
    font-size: 1.2rem;
    margin-top: 16px;
}
form #holdButtons {
    width: 100%;
    display: flex;
    margin-top: 16px;
    align-items: center;
    margin-bottom: 25px;
    justify-content: space-between;
}
form #holdButtons a, form #holdButtons
input[type=button] {
    display: flex;
    justify-content: center;
    align-items: center;
    text-decoration: none;
    width: 45%;
    border: 0px;
    height: 32px;
    box-sizing: border-box;
    border-radius: 25px;
    cursor: pointer;
}
form #holdButtons a.border, form #holdButtons
input[type=button].border {
    border: 3px solid #1446cf;
    color: black;
}
form #holdButtons a.noBorder, form #holdButtons
input[type=button].noBorder {
    background-color: #1446cf;
    color: white;
}
form div#terasa {
    display: flex;
    justify-content: space-evenly;

```

```

}
form div#terasa label {
  width: -moz-max-content;
  width: max-content;
  text-align: left;
  display: inline-block;
  margin-bottom: 5px;
}
form input#predaj {
  width: -moz-fit-content;
  width: fit-content;
  padding: 5px 15px;
  margin: 0px auto;
  background-color: #1446cf;
  color: white;
  border: 0px;
  display: block;
  border-radius: 25px;
  font-size: 1.2rem;
  margin-bottom: 20px;
}

```

```

#realestatepost {
  box-sizing: border-box;
  padding: 15px;
}
#realestatepost .icon-heart {
  display: inline-block;
  box-sizing: border-box;
  padding: 5px;
  border-radius: 50%;
  border: 1px solid #444548;
  margin-left: 10px;
}
#realestatepost .icon-heart.active {
  color: red;
}
#realestatepost #additionalDescription {
  margin: 20px 0px;
}

```

```

}
#realestatepost #additionalDescription h2 {
    font-size: 1.3rem;
    margin-bottom: 8px;
}
#realestatepost #owner {
    margin-bottom: 20px;
}
#realestatepost #owner h1 {
    font-size: 1.3rem;
    margin-bottom: 8px;
}
#realestatepost #owner h1 a {
    color: black;
    text-decoration: none;
}
#realestatepost #owner h1 a:hover {
    text-decoration: underline;
}
#realestatepost #owner p a {
    color: black;
    cursor: pointer;
    text-decoration: none;
}
#realestatepost #title {
    font-size: 3rem;
}
#realestatepost #adress {
    color: #444548;
    margin-top: 6px;
    font-size: 1.2rem;
    font-weight: normal;
}
#realestatepost #price {
    background: rgb(207, 145, 0);
    background: linear-gradient(120deg, rgb(207, 145, 0)
0%, rgb(249, 217, 0) 41%, rgba(255, 248, 0,
0.4682247899) 60%, rgba(255, 222, 0, 0) 100%);

```

```

    display: block;
    margin: 10px 0px;
}
#realestatepost #price #textprice {
    font-size: 2rem;
    color: white;
    box-sizing: border-box;
    padding: 10px 10px;
}
#realestatepost #basic_info {
    height: auto;
    box-sizing: border-box;
    display: flex;
    display: flex;
    justify-content: flex-start;
    align-items: center;
}
#realestatepost #basic_info span {
    width: 150px;
    height: 150px;
    border-radius: 25px;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    background-color: #f1f1f1;
    color: #444548;
    margin-right: 15px;
}
#realestatepost #basic_info span span.value {
    background-color: transparent;
    width: -moz-min-content;
    width: min-content;
    height: -moz-min-content;
    height: min-content;
    display: block;
    margin: 0px;
}

```

```

#realestatepost #basic_info span span.header {
  font-size: 1.1rem;
  background-color: transparent;
  width: -moz-min-content;
  width: min-content;
  height: -moz-min-content;
  height: min-content;
  font-weight: bold;
  margin-bottom: 10px;
  margin-right: 0px;
}
#realestatepost div#images {
  margin-top: 24px;
  display: grid;
  grid-template-columns: 800px 300px;
  grid-template-rows: repeat(3, 200px);
  grid-column-gap: 0px;
  grid-row-gap: 0px;
  row-gap: 10px;
  -moz-column-gap: 10px;
  column-gap: 10px;
}
#realestatepost div#images .main {
  grid-area: 1/1/4/2;
  width: 100%;
  height: 100%;
}
#realestatepost div#images .sidebar1 {
  grid-area: 1/2/2/3;
  width: 100%;
  height: 100%;
}
#realestatepost div#images .sidebar2 {
  grid-area: 2/2/3/3;
  width: 100%;
  height: 100%;
}
#realestatepost div#images .sidebar3 {

```



```

    grid-area: 3/2/4/3;
    width: 100%;
    height: 100%;
}
#realestatepost div#images .d-none {
    display: none;
}
#realestatepost div#images div {
    background-position: center center;
    background-size: cover;
    width: 100%;
    height: 100%;
    display: block;
}
#realestatepost .overlay {
    position: fixed;
    top: 0px;
    left: 0px;
    width: 100%;
    height: 100%;
    background-color: hsla(0, 0%, 20%, 0.73);
    display: flex;
    align-items: center;
    justify-content: center;
}
#realestatepost .overlay .x {
    position: absolute;
    z-index: 5;
    left: 85%;
    top: 5%;
    font-size: 46px;
}
#realestatepost .overlay .imgholder {
    display: flex;
    justify-content: center;
    align-items: center;
}
#realestatepost .overlay .imgholder img {

```

```
width: 80%;
max-height: 800px;
max-width: 80%;
height: auto;
display: block;
}
#realestatepost .overlay .imgholder .left {
position: absolute;
left: 15%;
top: 50%;
color: white;
z-index: 20;
font-size: 35px;
transform: translate(-50%, -50%);
cursor: pointer;
box-sizing: border-box;
padding: 15px;
}
#realestatepost .overlay .imgholder .right {
position: absolute;
z-index: 20;
font-size: 35px;
left: 85%;
top: 50%;
color: white;
transform: translate(-50%, -50%);
box-sizing: border-box;
padding: 15px;
cursor: pointer;
}
```

```
#title {
margin: 25px 25px 0px 25px !important;
/* position: sticky;
top: 99px;
bottom: auto */
}
```

```
#allApartments {
```

```

    box-sizing: border-box;
    padding: 25px;
    display: grid;
    grid-template-columns: 1fr 2fr;
    grid-template-rows: 1fr;
    grid-column-gap: 25px;
    grid-row-gap: 25px;
    height: 90vh;
}
#allApartments #filter {
    grid-area: 1/1/2/2;
    /* position: fixed;
       top:100px;
       bottom:100px;
       top:auto;
       height: fit-content;*/
    max-width: 350px;
    width: 100%;
    border-radius: 15px;
}
#allApartments #filter #lastClicked {
    text-decoration: none;
    color: black;
    margin-top: 15px;
    margin-bottom: 15px;
    border: 1px solid #444548;
    border-radius: 15px;
    box-sizing: border-box;
    padding: 15px;
    width: 100%;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    align-items: center;
}
#allApartments #filter #lastClicked h1, #allApartments
#filter #lastClicked h2 {
    font-size: 0.9rem;

```

```
margin-bottom: 10px;
margin-top: 10px;
}
#allApartments #filter #lastClicked p {
margin: 10px auto;
}
#allApartments #filter #lastClicked img {
border-radius: 5px;
}
#allApartments #filter #filters {
background-color: #f1f1f1;
box-sizing: border-box;
padding: 15px;
border-radius: 15px;
width: 100%;
}
#allApartments #filter #filters #checkbox input {
display: inline;
width: -moz-fit-content;
width: fit-content;
}
#allApartments #filter #filters #checkbox label {
display: inline;
margin: 0px;
margin-left: 10px;
}
#allApartments #filter #submit {
border: 0px;
color: #1446cf;
background-color: transparent;
font-weight: bold;
font-size: 1.2rem;
}
#allApartments #filter #selectedCities {
display: flex;
justify-content: flex-start;
flex-wrap: wrap;
align-items: center;
```

```

    margin-top: 12px;
}
#allApartments #filter #selectedCities div {
    display: inline-block;
    width: -moz-fit-content;
    width: fit-content;
    height: -moz-fit-content;
    height: fit-content;
    box-sizing: border-box;
    padding: 5px;
    margin-top: 5px;
    margin-left: 5px;
    background-color: #1446cf;
    color: #f1f1f1;
    border-radius: 10px;
}
#allApartments #filter #selectedCities div .icon {
    margin-left: 5px;
    padding-left: 5px;
    border-left: 1px solid #f1f1f1;
}
#allApartments #apartments {
    grid-area: 1/2/2/3;
    height: auto;
    overflow-y: auto;
    min-height: 500px;
}
#allApartments #apartments .message {
    text-align: center;
    color: #444548;
    font-style: italic;
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}
#allApartments #apartments .blockApartment {
    display: flex;

```

```

justify-content: flex-start;
margin: auto;
width: 80%;
height: -moz-fit-content;
height: fit-content;
cursor: pointer;
color: #444548;
text-decoration: none;
box-sizing: border-box;
margin-bottom: 10px;
border-radius: 15px;
border: 1px solid #444548;
}
#allApartments #apartments .blockApartment #col2 {
  overflow: hidden;
  box-sizing: border-box;
  padding: 15px;
}
#allApartments #apartments .blockApartment #col2 h1 {
  font-size: 1.1rem;
  margin-bottom: 4px;
}
#allApartments #apartments .blockApartment #col2
h1:hover {
  text-decoration: underline;
}
#allApartments #apartments .blockApartment #col2 .city
{
  font-size: 1rem;
  margin-bottom: 4px;
}
#allApartments #apartments .blockApartment
#col2 .adress {
  font-size: 0.9rem;
  display: inline-block;
}
#allApartments #apartments .blockApartment
#col2 .areaRoomDiv {

```

```

display: flex;
align-items: center;
justify-content: flex-start;
margin: 5px 0px;
}
#allApartments #apartments .blockApartment
#col2 .areaRoomDiv .numberOfRooms {
height: -moz-fit-content;
height: fit-content;
margin-left: 5px;
padding-left: 5px;
border-left: 1px solid #444548;
}
#allApartments #apartments .blockApartment #col3 {
display: flex;
align-items: end;
justify-content: end;
margin-left: auto;
flex-direction: column;
}
#allApartments #apartments .blockApartment
#col3 .priceSqm {
margin-right: 15px;
}
#allApartments #apartments .blockApartment #col3 .price
{
font-weight: bold;
font-size: 2rem;
box-sizing: border-box;
padding: 15px;
padding-top: 8px;
color: #1446cf;
}
#allApartments #apartments .blockApartment
#col3 .price:hover {
text-decoration: underline;
}
#allApartments #apartments .blockApartment #image {

```

```
width: 200px;
border-radius: 15px 0px 0px 15px;
height: 200px;
background-position: center;
background-size: cover;
}
```

```
#pagination {
  border: 1px solid #444548;
  display: flex;
  border-radius: 10px;
  overflow: hidden;
  width: -moz-fit-content;
  width: fit-content;
  border-radius: 1px solid #f1f1f1;
  margin: 20px auto;
}
#pagination span {
  box-sizing: border-box;
  padding: 10px;
  cursor: pointer;
  width: -moz-fit-content;
  width: fit-content;
  background-color: white;
}
#pagination span:nth-child(n+2) {
  border-left: 1px solid #444548;
}
#pagination span.active {
  background-color: #1446cf;
  color: white;
}
```

```
#support h1 {
  font-size: 2rem;
  text-align: center;
  margin-top: 50px;
}
#support form {
```



```
margin-bottom: 100px;
}
#support form #submit {
width: 40%;
border: 0px;
height: 32px;
box-sizing: border-box;
border-radius: 25px;
margin: 0px auto;
display: block;
cursor: pointer;
margin-top: 15px;
border: 3px solid #1446cf;
color: black;
}
```

```
#holder {
box-sizing: border-box;
padding: 10px 10%;
}
#holder #email {
margin: 10px 0px;
}
#holder #realestate, #holder #wishlist {
border: 1px solid black;
width: 100%;
border-radius: 10px;
margin: 15px auto;
height: -moz-fit-content;
height: fit-content;
max-height: 650px;
overflow-y: auto;
}
#holder #realestate a.blockApartmentUserProfile,
#holder #wishlist a.blockApartmentUserProfile {
box-sizing: border-box;
padding: 3px 10px;
display: block;
color: black;
```

```
text-decoration: none;
border: 1px solid black;
border-radius: 7px;
margin: 5px;
}
```

```
#support1 {
  width: 900px;
  margin: 10px 25px;
}
#support1 table {
  width: 100%;
}
#support1 table thead tr th {
  border: 1px solid #444548;
  box-sizing: border-box;
  padding: 10px;
  text-wrap: nowrap;
  text-align: center;
  width: auto;
  max-width: 250px;
}
#support1 table tbody tr {
  width: 100%;
}
#support1 table tbody tr td {
  border: 1px solid #444548;
  box-sizing: border-box;
  padding: 5px 10px;
  overflow: hidden;
  max-width: 250px;
  text-wrap: wrap;
  word-wrap: break-word;
}
```

```
#login {
  margin: auto;
  margin-bottom: 25px;
}
```

```
#formAdmin {
  margin: 100px auto;
  width: 300px;
}
#formAdmin input[type=submit] {
  background-color: #1446cf;
  color: white;
  padding: 10px;
  box-sizing: border-box;
  border: 0px;
  border-radius: 25px;
  display: block;
  width: -moz-fit-content;
  width: fit-content;
  margin: 20px auto;
}
```

```
#title2 {
  width: -moz-fit-content;
  width: fit-content;
  margin: 20px auto;
}
```

```
.overlay {
  position: fixed;
  top: 0px;
  left: 0px;
  width: 100%;
  height: 100%;
  background-color: hsla(0, 0%, 20%, 0.73);
  display: flex;
  align-items: center;
  justify-content: center;
}
.overlay .x {
  position: absolute;
  z-index: 5;
  left: 85%;
```

```
    top: 5%;
    font-size: 46px;
}
.overlay p {
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 2rem;
    height: 400px;
    width: 80%;
    background-color: #f1f1f1;
    color: #444548;
}
```

```
@media screen and (max-width: 1140px) {
    #allApartments #apartments .blockApartment {
        margin-left: auto;
        margin-right: 0px;
        margin-bottom: 10px;
    }
    #allApartments #filter #filters {
        width: 100%;
    }
    #allApartments #filter #filters input, #allApartments
    #filter #filters select {
        width: 100%;
    }
    #realestatepost div#images {
        width: 100%;
        display: grid;
        grid-template-columns: repeat(3, auto);
        grid-template-rows: 100px 100px 100px 200px;
        grid-column-gap: 10px;
        grid-row-gap: 10px;
    }
    #realestatepost .main {
        grid-area: 1/1/4/4 !important;
    }
    #realestatepost .sidebar1 {
```

```

        grid-area: 4/1/5/2 !important;
        height: auto;
    }
    #realestatepost .sidebar2 {
        grid-area: 4/2/5/3 !important;
        height: auto;
    }
    #realestatepost .sidebar3 {
        grid-area: 4/3/5/4 !important;
        height: auto;
    }
}
@media screen and (max-width: 768px) {
    html {
        font-size: 14px;
    }
    #title {
        margin: 15px 5px 15px 5px;
        position: static;
    }
    #allApartments {
        display: grid;
        grid-template-columns: 100%;
        grid-template-rows: repeat(21, auto);
        grid-column-gap: 0px;
        grid-row-gap: 20px;
        height: -moz-fit-content;
        height: fit-content;
    }
    #allApartments #filter {
        grid-area: 1/1/2/2;
        position: static !important;
        width: 100%;
        max-width: 100%;
    }
    #allApartments #apartments {
        grid-area: 2/1/22/2 !important;
        overflow-y: unset;
    }
}

```

```

    min-height: unset;
    height: -moz-max-content;
    height: max-content;
}
#allApartments #apartments .blockApartment {
    width: 100%;
    margin-left: 0px;
    margin-right: 0px;
    margin: 0px auto !important;
    margin-bottom: 10px !important;
}
header i.toggle {
    display: block;
}
header ul, header a#user {
    box-sizing: border-box;
    padding-top: 10px;
    display: none;
}
header ul li:first-child, header a#user li:first-
child {
    padding: 0px;
}
header ul li, header a#user li {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
header ul li ul, header a#user li ul {
    position: static;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    width: 100%;
    padding: 0px;
    color: #444548;
}

```

```

    background-color: transparent;
    border: 0px;
    text-align: center;
}
header ul li ul li, header a#user li ul li {
    text-align: center;
    display: block;
}
header ul li ul li:first-child, header a#user li ul
li:first-child {
    padding: 5px !important;
}
header ul li ul li a, header a#user li ul li a {
    text-align: center;
}
footer ul {
    flex-direction: column;
}
}
@media screen and (max-width: 480px) {
    html {
        font-size: 12px;
    }
    #allApartments {
        padding: 5px;
    }
    #allApartments #apartments .blockApartment #image {
        display: none;
    }
    #allApartments #apartments .blockApartment
#col3 .priceSqm {
        display: none;
    }
    #allApartments #apartments .blockApartment
#col3 .price {
        font-size: 1.2rem;
        padding: 0px;
        margin-bottom: 7px;
    }

```

```

        margin-right: 7px;
    }
    #realestatepost h1 {
        font-size: 1.3rem;
    }
    #realestatepost div#images {
        grid-template-rows: 100px 100px 100px
100px !important;
    }
    #realestatepost #basic_info span {
        height: auto;
        width: 22%;
        height: 60px;
        flex-shrink: 1;
    }
}/*# sourceMappingURL=style.css.map */

```

## Public/html/support.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Support</title>
    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
    <script
src="View/public/js/header.js" type="application/javas
cript"></script>
</head>
<body>
    <h1>Support</h1>
    <form action="#" id="firstForm">
        <label for="fName">First Name</label>
        <input type="text" id="fName" name="fName"/>
        <label for="lName">Last Name</label>

```



```

        <input type="text" id="lName" name="lName"/>
        <label for="email">Email</label>
        <input type="email" name="email" id="email"/>
        <label for="titleSupport">Tittle</label>
        <input type="text" name="titleSupport"
id="titleSupport"/>
        <label for="contentSupport">Content</label>
        <input type="text" name="contentSupport"
id="contentSupport"/>
        <input type="submit" id="submit"
name="submit"/>

    </form>
</body>
</html>

```

## Public/html/register.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css">
    <script
src="View/public/js/header.js" type="application/javas
cript"></script>
    <title> Register</title>
</head>
<body>
    <form action="#" id="firstForm">
        <h1>Register</h1>
        <label for="fName">First Name</label>
        <input type="text" id="fName" name="lName"/>

```

```

        <label for="lName">Last Name</label>
        <input type="text" id="lName" name="lName"/>
        <label for="email">Email</label>
        <input type="email" name="email" id="email"/>
        <label for="password">Password</label>
        <input type="password" name="password"
id="password"/>
        <label for="confPassword">Confirm
Password</label>
        <input type="password" name="confPassword"
id="confPassword"/>
        <div id="holdButtons">
            <!--<input type="button" value="Already
have account?" class="border"/>-->
            <a href="login" class="border">Already have
account?</a>
            <input type="button" value="Next"
class="noBorder" id="next"/>
        </div>

    </form>
    <script src="View/public/js/regex.js"
type="application/javascript" ></script>
    <script src="View/public/js/form.js"
type="application/javascript"></script>
</body>
</html>

```

## Public/html/login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>

```

```

    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
    <script src="View/public/js/error.js"
type="application/javascript"></script>

    <script
src="View/public/js/header.js" type="application/javas
cript"></script>
    <title>Login</title>
</head>
<body>
    <form action="">
        <h1>Login</h1>
        <label for="email">Email</label>
        <input type="email" name="email" id="email"/>
        <label for="password">Password</label>
        <input type="password" name="password"
id="password"/>
        <div id="holdButtons">
            <input type="button" value="Log in"
class="border" id="login"/>
        </div>
    </form>
    <script src="View/public/js/regex.js" type="appli
cation/javascript"></script>
    <script
src="View/public/js/login.js" type="application/javasc
ript"></script>
</body>
</html>

```

## Public/html/allApartmentsJS.html

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>All Apartments - JS</title>
    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
    <script
src="View/public/js/header.js" type="application/javas
cript"></script>

    <!--
<script src="View/public/js/cityList.js" type="applic
ation/javascript"></script>
--><script src="View/public/js/addApartment.js"
type="application/javascript"></script>
    <script src="View/public/js/getAllApartmentsJS.js"
type="application/javascript"></script>
    <script src="View/public/js/loadLastClicked.js"
type="application/javascript"></script>
    <script
src="https://kit.fontawesome.com/104d21a00f.js" type="
application/javascript"
crossorigin="anonymous"></script>
    <script src="View/public/js/error.js"
type="application/javascript"></script>

```

```

</head>
<body>
<h1 id="title">All Apartments</h1>
<div id="allApartments">

```

```

    <div id="filter">
        <form id="filters">
            <label for="search">Search</label>
            <input type="text" id="search"
name="search" placeholder="Seach list of apartments"/>

```

```

            <label for="tipObjekta">Type of Real
Estate</label>

```

```

        <!-- <select name="tipObjekta"
id="tipObjekta" required="required">
            <option value="0" >All</option>
        </select>

```

```

-->
        <div id="checkboxx">
            <input type="checkbox" name="tipObjekta"
id="tipObjekta" value="1"/>
            <label for="tipObjekta">Flat</label>
        </div>
        <div id="checkboxx">
            <input type="checkbox" name="tipObjekta"
id="tipObjekta" value="2"/>
            <label for="tipObjekta">Cottage</label>
        </div>
        <div id="checkboxx">
            <input type="checkbox" name="tipObjekta"
id="tipObjekta" value="3"/>
            <label for="tipObjekta">House</label>
        </div>
        <label for="select">Select city</label>
        <input type="text" name="cityconsole"
list="city" id="cityconsole" placeholder="Select city">
        <datalist name="city" id="city"
required="required">

```

```

</datalist>
        <div id="selectedCities">

```

```

</div>
        <label for="orderby">
            Order by
        </label>
        <select name="orderby" id="orderby">
            <option value="0">Default</option>
            <option value="1">Price ASC</option>
            <option value="2">Price Desc</option>

```

```

        <option value="3">Date Posted
Earliest</option>
        <option value="4">Date Posted
Latest</option>
    </select>
    <input type="hidden" name="hidden"
id="hidden" value="">
</form>

```

```

</div>
<div id="apartments">

```

```

    </div>
</div>
<div id="pagination"></div>

```

```

</body>
</html>

```

## Public/html/allApartments.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>All Apartments</title>
    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
    <script src="View/public/js/header.js"></script>
    <script src="View/public/js/error.js"
type="application/javascript"></script>

    <script src="View/public/js/cityList.js" type="ap
plication/javascript"></script>

```

```
<script  
src="View/public/js/addApartment.js" type="application  
/javascript"></script>
```

```
<script  
src="https://kit.fontawesome.com/104d21a00f.js" type=""  
application/javascript"  
crossorigin="anonymous"></script>
```

```
</head>  
<body>  
  <h1 id="title">All Apartments </h1>  
  <div id="allApartments">
```

```
    <div id="filter">  
      <form id="filters">  
        <label for="search">Search</label>  
        <input type="text" id="search"  
name="search" placeholder="Seach list of apartments"/>  
  
        <label for="tipObjekta">Type of Real  
Estate</label>  
        <select name="tipObjekta"  
id="tipObjekta" required="required">  
          <option value="0" >All</option>  
        </select>  
        <label for="select">Select city</label>  
        <input type="text" name="cityconsole"  
list="city" id="cityconsole" placeholder="Select city">  
        <datalist name="city" id="city"  
required="required">  
  
        </datalist>  
        <div id="selectedCities">
```

```
      </div>  
      <input type="submit" id="submit"  
disabled>
```

```
        <input type="hidden" name="hidden"
id="hidden" value="">
    </form>
```

```
</div>
<div id="apartments">
```

```
    </div>
</div>
<div id="pagination"></div>
<script src="View/public/js/filterAllApartments.js"
type="application/javascript"></script>
<script src="View/public/js/citydatadynamic.js"
type="application/javascript"></script>
<script
src="https://kit.fontawesome.com/104d21a00f.js" type="
application/javascript"
crossorigin="anonymous"></script>
```

```
</body>
</html>
```

## Public/html/addApartment.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet"
href="View/public/css/style.css" type="text/css"/>
    <script
src="https://kit.fontawesome.com/104d21a00f.js"
type="application/javascript"
crossorigin="anonymous"></script>

    <script src="View/public/js/regex.js" type="appli
cation/javascript"></script>
```



```

    <script src="View/public/js/cityList.js"
type="application/javascript"></script>
    <script src="View/public/js/addApartment.js"
type="application/javascript"></script>
    <script src="View/public/js/header.js"
type="application/javascript"></script>
    <script src="View/public/js/error.js"
type="application/javascript"></script>
    <script
src="View/public/js/addApartmentFormValidate.js"
type="application/javascript"></script>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <form id="method" enctype='multipart/form-data'
action="addApartment" method="POST" >
        <label for="title1">Title</label>
        <input type="text" id="title1" name="title"
placeholder="Title" required="required" error="Enter
valid title"/>
        <label for="select" >Select city</label>
        <select name="city" id="city"
required="required" error="Select a city" >
            <option value="0" disabled selected>Select
city</option>
        </select>
        <label for="adresa">Adress</label>
        <input type="text" id="adresa" name="adresa"
required="required" error="Enter address (ex.
Ruzveltova 12)"/>
        <label for="tipObjekta">Type Of
Building</label>
        <select name="tipObjekta" id="tipObjekta"
required="required" error="Select type of real estate">
            <option value="0" disabled selected>Select
type of building</option>
        </select>

```

```

        <label for="numberrooms">Number of
rooms</label>
        <input type="number" id="numberrooms"
error="Enter number of rooms (ex 2.5)"
name="numberrooms" min="0.5" step="0.5" max="10"
required="required"/>
        <label for="terasa">Does it have a
terrace/balcony</label>
        <div id="terasa">
            <div>
                <input type="radio"
name="terasa" id="prvi" value="Da" placeholder="Da"
data-error = "Select one of the values"/> <label
for="prvi">Yes</label>
            </div>
            <div>
                <input type="radio" name="terasa" id="drugi"
value="Ne"/> <label for="prvi">No</label>
            </div>
        </div>
    </div>

```

```

        </div>
        <label for="kvadratura" >Area</label>
        <input type="number" id="kvadratura"
error="Enter area of building" name="kvadratura"
required="required" min="1"/>
    </div>

```

```

        <label for="cena">Total price</label>
        <input type="number" id="cena" name="cena"
error="Enter price" required="required" min="0"
step="500">
        <label for="dodatniopis">Description</label>
        <input type="text" id="dodatniopis" error="Add
additional description" name="dodatniopis">
        <input type="file" multiple="multiple"
id="images" name ="images[]" accept="image/jpeg,
image/png, image/jpg"/>
        <label for="images" id="file">Add
images</label>
    </div>

```

```
        <input type="submit" id="predaj" type="button"
value="Submit"/>
    </form>
```

```
</body>
```

```
</html>
```

### Public/js/addApartment.js

```
window.addEventListener("load",function(){
    let xhr_object_type = new XMLHttpRequest();
    xhr_object_type.open("GET", "typeOfRealEstate");
    xhr_object_type.send();
    xhr_object_type.onreadystatechange = function(){
        if(xhr_object_type.readyState === 4){
            if(xhr_object_type.status ===200){
                try{

                    let select =
document.querySelector("#tipObjekta");

                    let response =
JSON.parse(xhr_object_type.response);
                    response.forEach(element => {
                        let opt =
document.createElement("option")
                        opt.value = element.id;
                        opt.innerText = element.name;
                        select.appendChild(opt);
                    });

                }catch(err){
```

```

        errorPopUp(err);
    }
}
}
}
}

}))

```

## Public/js/addApartmentFormValidation.js

```

window.addEventListener("DOMContentLoaded", function(){
    var formaz;

    async function submitForm(url, obj){
        const response = await fetch(url, {
            method: "POST",
            mode: "cors",
            cache: "no-cache",
            credentials: "same-origin",

            redirect: "follow",
            referrerPolicy: "no-referrer",
            body: obj,
        });
        return response
    }
}

```

```

    let forma = document.getElementById("method")
    formaz = new FormData(forma)

```

```

let sub =document.querySelector("#predaj")
sub.addEventListener("click", function(e){
    e.preventDefault();
    for(let x of mapa){

```

```

    }
}

```

```
if(!x){
```

```
    for(let y of mapa){
        query =
document.querySelector(`#${y[0]}`)
        let inp = new Event("input");
        let inp2 = new Event("change");
        query.dispatchEvent(inp2)
        query.dispatchEvent(inp)
        console.log(y)
    }
    return 0;
```

```
    }else{
        if(!x[1](document.querySelector(`#${x[0]
}'))){

            for(let y of mapa){
                query =
document.querySelector(`#${y[0]}`)
                let inp = new Event("input");
                query.dispatchEvent(inp)
                console.log(y)
            }
            errorPopUp("All fields are
mandatory")

            return 0;
        }
    }

}

let ter = document.getElementsByName("terasa")
if(!radioconf(ter)){
    errorPopUp("Select value for balcony")
    return 0;
}

submitForm("addApartment",formaz)
.then(async res=>{
    if(res.status==200){
```

```
window.location.href="allApartments";
```

```
    }  
    else{  
        let x = await res.text()  
        errorPopUp(x)  
    }  
}))  
  
}))  
let title = document.querySelector("#title1");  
let selectCity = document.querySelector("#city");  
let adress = document.querySelector("#adresa");  
let selectTipObjekta =  
document.querySelector("#tipObjekta");  
let numberRooms =  
document.querySelector("#numberrooms");  
let terasa = document.getElementsByName("radio")  
let brojKvadrata =  
document.querySelector("#kvadratura");  
let price = document.querySelector("#cena");  
let description =  
document.querySelector("#dodatniopis");  
let files = document.querySelector("#images");
```

```
forma.addEventListener("input", function(){
```

```
    formaz = new FormData(this)  
  
    formaz.append("city", selectCity.value)  
    formaz.append("objectType", selectTipObjekta.value)  
}))
```

```
files.addEventListener("input", function(){
    let file = files.files;
```

```
    for (let i = 0; i < file.length; i++) {
        formaz.append("images[]", file[i]);
    }
```

```
    for(let x of file){
        let fileread = new FileReader()
        let img = document.createElement("img");
```

```
        fileread.onload = function(e){
            img.src = e.target.result;
```

```
            files.insertAdjacentElement("afterend",
img);
```

```
        }
        fileread.readAsDataURL(x);
    }
```

```
    })
    let mapa = new Map();
    mapa.set(title.id ,titleconf );
    mapa.set(selectCity.id, selectValidacija );
    mapa.set(selectTipObjekta.id,selectValidacija );
    mapa.set(adress.id , adresa );
    mapa.set(numberRooms.id, sobe );
    mapa.set(brojKvadrata.id, kvadratura );
    mapa.set(price.id, priceCheck );
    mapa.set(description.id, opis );
    //mapa.set(terasa.id, radioconf)
```

```
    mapa.forEach((e,x)=>{
        let obj =document.querySelector(`#${x}`);
        obj.addEventListener("input", function(){
```

```

        if(!e(this)){
            addError(this)
        }
        else{
            removeError(this)
        }
    })
})

```

```

    })

```

## Public/js/addToWishlist.js

```

function toggleWishlist(atr,elem){
    let xhrWishlist = new XMLHttpRequest()
    xhrWishlist.open("POST", "wishlist", true)
    xhrWishlist.setRequestHeader("Content-Type",
"application/json");
    let body1;
    body1 = {"id_post": atr}

    xhrWishlist.onreadystatechange = function(){
        if(xhrWishlist.readyState == 4 &&
xhrWishlist.status== 200){
            elem.classList.toggle("fa-solid")
            elem.classList.toggle("fa-regular")

            elem.classList.toggle("active")

```



```

    }
  }
  try{
    xhrWishlist.send(JSON.stringify(body1))

  }catch(e){
    errorPopUp("error occured please try again
later")
  }
}
}

```

```

window.addEventListener("load", function(){

  let wishlist = document.querySelector("#wishlist");
  wishlist.addEventListener("click", function(){
    attr = this.getAttribute("data-id");
    toggleWishlist(attr,this)

  })

})

```

### Public/js/citydatadynamic.js

```

let datalist = document.querySelector("#city");
function addCityForSearch(element, e){
let datalist = document.querySelector("#city");
let selectedCitiesData =
document.querySelector("#selectedCities");

  if(element.value.length == 5){

```

```

    if(e.keyCode == 13 ){
      let el =
document.querySelector("option[value='" + element.value
+ "']");

```

```

        if(el){
            filter.city.push(el.value)
            let newcity =
document.createElement("div")
            let span =
document.createElement("span")
            let icon =
document.createElement("span")
            icon.classList.add("fa-solid","fa-
xmark", "icon")
            span.innerText = el.innerText;
            newcity.appendChild(span)
            newcity.appendChild(icon)
            newcity.setAttribute("data-city-id",
el.value);
            newcity.setAttribute("data-city-name",
el.innerText );
            selectedCitiesData.appendChild(newcity);
            newcity.addEventListener("click",
function(){

                let itembacktodatalist =
document.createElement("option")
                itembacktodatalist.innerText =
this.getAttribute("data-city-name");
                itembacktodatalist.value =
this.getAttribute("data-city-id");
                datalist.appendChild(itembacktodata
list)

                filter.city.splice(filter.typeOfObj
ect.indexOf(this.getAttribute("data-city-id")),1 );
                this.remove()
                filter.page=0;
                getAllApartments()

            })
            el.remove()
            element.value=""
        }

```

```
    }  
  }  
}
```

```
}  
let cityconsole =  
document.querySelector("#cityconsole");  
cityconsole.addEventListener("keydown", function(e){  
    addCityForSearch(this, e);  
})  
  
datalist.addEventListener("blur",function(e){  
    addCityForSearch(this, e);  
})  
window.addEventListener("load", getAllApartments)
```

## Public/js/cityList.js

```
window.addEventListener("DOMContentLoaded",function(){  
let xhr = new XMLHttpRequest();  
xhr.open("GET", 'city', true);  
xhr.send();  
xhr.onreadystatechange = function () {  
    if(xhr.readyState === 4){  
        try{  
            let select= document.querySelector("#city")  
            if(xhr.status === 200){  
  
                let json = JSON.parse(xhr.responseText);  
  
                json.forEach(element => {  
                    let opt =  
document.createElement("option")  
                    opt.value = element.zip;  
                    opt.innerText = element.city;
```

```

        select.appendChild(opt)
    })
}
else if(xhr.status === 503){
    window.location.reload();
}
}catch(err){
    errorPopUp("We couldnt load cities please
try again later")
}

```

```

    }
}

```

```

})

```

## Public/js/cityListCheckBox.js

```

window.addEventListener("DOMContentLoaded",function(){
    let xhr = new XMLHttpRequest();
    xhr.open("GET", 'city', true);
    xhr.send();
    xhr.onreadystatechange = function () {
        if(xhr.readyState === 4){
            try{
                let select=
document.querySelector("#city")
                if(xhr.status === 200){

                    let json =
JSON.parse(xhr.responseText);

                    json.forEach(element => {
                        let opt =
document.createElement("option")
                        opt.value = element.zip;

```

```

                                opt.innerText =
element.city;
                                select.appendChild(opt)
                                })
                                }
                                else if(xhr.status === 503){
                                    window.location.reload();
                                }
                                }catch(err){
                                    errorPopUp("We couldnt load cities
please try again later")
                                }

                                }
                                }

                                })

```

**Public/js/error.js**

```

function errorPopUp(message){
    let body=document.querySelector("body")

    let overlay=document.createElement("div")
    overlay.classList.add("overlay")
    let x = document.createElement("i")
    x.classList.add("fa-solid", "fa-xmark","x");
    let p = document.createElement("p")
    p.innerHTML=message
    x.addEventListener("click", function(){
        overlay.remove();
    })
    overlay.appendChild(p)
    overlay.appendChild(x)
    body.appendChild(overlay)
}

```

```
    //overlay.innerHTML=message;
}
```

## Public/js/filterAllApartments.js

```
const filter = {
  city:[],
  typeOfObject:"",
  search:"",
  page:0
}

let typeOfObject = document.querySelector("#tipObjekta")
typeOfObject.addEventListener("change", function(){
  filter.typeOfObject = this.value
  filter.page=0;
  getAllApartments()
})

let sub = document.querySelector("#submit")
let filtersForm =
document.querySelector("#filters");
filtersForm.addEventListener("input",function(e)
{
  e.preventDefault()
  sub.removeAttribute("disabled")

})

filtersForm.addEventListener("submit", function(e){
  e.preventDefault()
  //sub.setAttribute("disabled","disabled")
})
```

```
async function getAllApartments(){
```

```

        url = new
URL(window.location.origin+"/FilterAllApartments");
        const searchParams = new URLSearchParams();
        Object.keys(filter).forEach( e=>{
            if(Array.isArray(e)){
                e.forEach(x=>{

```

```

                    searchParams.append(e, x);
                })
            }
            else{
                searchParams.append(e, filter[e])
            }
        } )

```

```

url.search = searchParams;

```

```

        let req = new XMLHttpRequest()
        req.open("GET",url.href)

```

```

        req.send()
        req.onreadystatechange=function(){
            if(req.status==200 && req.readyState==4){
                try{
                    res = JSON.parse(req.responseText)

                    let apartmentList =
document.querySelector("#apartments")
                    apartmentList.innerHTML = "";

                    for(let i = 0; i < res["pages"].length;
i++){
                        let apartmentone =
document.createElement("a");
                        apartmentone.classList.add("blockAp
artment")
                        apartmentone.href =
`openApartment?id=${res["pages"][i].id}`;

```

```

        let col1 =
document.createElement("div");
        col1.id="image";
        col1.style.backgroundImage =
`url('${res["pages"][i].img_url}')`;

        let col2 =
document.createElement("div")
        col2.id = "col2";
        let col3 =
document.createElement("div");
        col3.id = "col3";
        //naslov
        let title =
document.createElement("h1");
        title.innerText =
res["pages"][i].title
        //adresa
        let address =
document.createElement("p")
        address.classList.add("address")
        address.innerText =
res["pages"][i].address

        //number of rooms
        let numberOfRooms =
document.createElement("p")
        numberOfRooms.classList.add("number
OfRooms")

        numberOfRooms.innerHTML = "number
of rooms: " + res["pages"][i].numberOfRooms
        let city=document.createElement("p")
        city.classList.add("city")
        city.innerHTML=res["pages"][i].city
_name

```



```

        //area

        let area =
document.createElement("p");
        area.classList.add("area")
        area.innerHTML =
res["pages"][i].area + "m<sup>2</sup>";

        //area num of rooms holder
        let areaRoomDiv =
document.createElement("div")
        areaRoomDiv.classList.add("areaRoom
Div")

        areaRoomDiv.appendChild(area)
        areaRoomDiv.appendChild(numberOfRoo
ms);

        //price
        let price=
document.createElement("p")
        price.classList.add("price")
        price.innerHTML =
` ${res["pages"][i].price}&euro; `
        //price sq/m
        let priceSQM =
document.createElement("p")
        priceSQM.classList.add("priceSqm")
        priceSQM.innerHTML =
Math.round(res["pages"][i].price / res["pages"][i].area)
+" &euro;/m<sup>2</sup>"
        col3.appendChild(priceSQM)
        col3.appendChild(price);
        col2.appendChild(title)
                                col2.appendChild(ci
ty)

        col2.appendChild(adress)

```

```

        col2.appendChild(areaRoomDiv)
        apartmentone.appendChild(col1)
        apartmentone.appendChild(col2)
        apartmentone.appendChild(col3)

        apartmentList.appendChild(apartment
one)
    }
    let pagination =
document.querySelector("#pagination");
    pagination.innerHTML=""
    if(!res["pages"].length){
        let message =
document.createElement("p")
        message.innerHTML = "We couldnt
find anything";
        message.classList.add("message");
        apartmentList.appendChild(message)
    }
    if(filter.page < 4){
        for(let i =0; i<6 &&
i<res["num_pages"] ;i++){
            let span =
document.createElement("span");
            span.innerHTML = i+1
            span.setAttribute("data-page",
i);
            if(filter.page ==
span.getAttribute("data-page")){
                span.classList.add("active")
;
            }
            span.addEventListener("click",
function(){
                filter.page =
this.getAttribute("data-page");

```

```

        getAllApartments()
    })
    pagination.appendChild(span)
}
}
else{
    for(let i =filter.page-3, j = 1;
j<7 && i<res["num_pages"] ; j++,i++){
        let span =
document.createElement("span");
        span.innerHTML = i+1
        span.setAttribute("data-page",
i);
        if(filter.page ==
span.getAttribute("data-page") ){
            span.classList.add("active")
;
        }
        span.addEventListener("click",
function(){

            filter.page =
this.getAttribute("data-page");

            getAllApartments()
        })
        pagination.appendChild(span)
    }
}
}catch(e){
    errorPopUp("Error occured please try
again later")
}
}
}
}

```

```
}
```

```
sub.addEventListener("click", function(e){  
    e.preventDefault()  
    getAllApartments();
```

```
})
```

```
let filterSearch = document.querySelector("#search")  
filterSearch.addEventListener("input", function(){  
    filter.search = this.value;  
    filter.page=0;  
    getAllApartments();  
})
```

## Public/js/form.js

```
let form = document.querySelector("#firstForm")  
let next = document.querySelector("#next")  
var mapIdFunc = new Map();  
function savedRegister(){  
    for(let [x,y] of mapIdFunc.entries()){  
        let el = document.querySelector(`#${x}`)  
  
        if(el && localStorage.getItem(x)){
```

```

        el.value = localStorage.getItem(x)

    }

}

function checkAllFields(){
    for(let [key, value] of mapIdFunc.entries()){

    }
}

function formEngine(){
    var fName = document.querySelector("#fName")
    var lName = document.querySelector("#lName")
    var email = document.querySelector("#email")
    var password = document.querySelector("#password")
    var confPassword =
document.querySelector("#confPassword")

```

```

    let arrayOfElements =
[fName, lName, email, password, confPassword]

```

```

    mapIdFunc.set(String(fName.id), confName(fName))
    mapIdFunc.set(String(lName.id), confName(lName))
    mapIdFunc.set(String(email.id), confemail(email))
    mapIdFunc.set(String(password.id), confpassword(password))
    mapIdFunc.set(String(confPassword.id),
confpassword(confPassword))

```

```

    arrayOfElements.forEach((element)=>{
        if(localStorage.getItem(element.id)) {
            element.innerHTML =
localStorage.getItem(element.id)
        }
        element.addEventListener('input', function(){
            try{
                localStorage.setItem(this.id, this.value)
            }

```

```

        catch(err){
            console.log("Turn on your localStorage")
        }
        if(!mapIdFunc.get(this.id)){
            addError(this)
        }
        else{
            removeError(this)
        }
    }

```

```

        })
    })
}
window.addEventListener("load", function(){

```

```

    formEngine()
    savedRegister()

```

```

    })

```

```

next.addEventListener("click", function(){
    for(let [key, value] of mapIdFunc.entries()){
        if(!value) break;
    }

```

```

    }
    let password = document.querySelector("#password")
    let confPassword =
document.querySelector("#confPassword")
    if(!passwordsEquals(password, confPassword)) {
        if(!document.querySelector("#password_error")){
            let error = document.createElement("p")
            error.innerText = "Passwords doesnt match"
            error.id = "password_error"
            let parent = this.parentElement
            parent.insertAdjacentElement("beforebegin",
error)

```

```

    }
  }
  else{
    if(document.querySelector("#password_error")){
      document.querySelector("#password_error").r
remove();
    }
  }

```

```

  }
  try{
    let obj = new XMLHttpRequest()
    obj.open("POST", 'register', true)
    obj.setRequestHeader("Content-type",
"application/json");
    let data = {
      fname : fName.value,
      lname : lName.value,
      email : email.value,
      password : password.value,
      passwordConf: confPassword.value
    }
  }

```

```

obj.send(JSON.stringify(data))
obj.onreadystatechange = function () {

  if (obj.readyState === XMLHttpRequest.DONE) {

    var error = document.createElement("p");
    error.id = "error"
    if(x = document.querySelector("#error")){
      x.remove()
    }
    if (obj.status === 201) {
      localStorage.clear();
      window.location.replace("login")
    }
    else if(obj.status === 503){

```

```
        error.innerText = "Our servers are  
currently down, try again in a few minutes";  
    }  
    else if(obj.status === 409){
```

```
        error.innerText = "Passwords doesn't  
match or someone is already using that mail already";  
    }  
    else if(obj.status === 403){
```

```
        error.innerText = "Don't change pre  
defined name attributes";
```

```
    }  
    else if(obj.status === 406){
```

```
        error.innerText = "Fill your fields per  
our requirements";
```

```
    }  
    form.insertAdjacentElement("beforeend",  
error);  
    }
```

```
    }  
    }catch(err){  
        console.log(err)  
    }  
})  
form.addEventListener("input", function(){  
    formEngine()  
})
```

## Public/js/getAllApartmentsJS.js

```
let retrived;  
const filter = {
```



```
city:[],
search:"",
typeOfObject:[],
orderby:0,
page:0,
max_pages:0
}
```

```
let xhrApartment = new XMLHttpRequest()
xhrApartment.open("GET", "getApartmentsJS")
xhrApartment.onreadystatechange = function(){
    if(xhrApartment.status ==200 &&
xhrApartment.readyState==4){
        try{
            var allApartments =
JSON.parse(xhrApartment.responseText);
            retrived = allApartments;
            filter.max_pages =
Math.ceil(allApartments.length/20);
```

```
addApartmentToList(allApartments);
```

```
paginationSet(retrived)
```

```
    }catch(e){
        errorPopUp(e)
    }
```

```
    }
}
xhrApartment.send();
function addApartmentToList(res){
    let apartmentList =
document.querySelector("#apartments")
    apartmentList.innerHTML="";
```

```
if(!res.length) {
```

```
        apartmentList.innerText = "We couldnt find  
anything"  
    }else{  
        filter.max_pages =  
Math.ceil( res.length / 20);
```

```
    }  
    for(let i = filter.page * 20 , j  
=filter.page*20+20; (i )<j && i<res.length ; i++){
```

```
        let apartmentone =  
document.createElement("a");  
        apartmentone.classList.add("blockApartment")  
        apartmentone.href =  
`openApartment?id=${res[i].id}`;  
        let col1 = document.createElement("div");  
        col1.id="image";  
        col1.style.backgroundImage =  
`url('${res[i].img_url}')`;
```

```
        let col2 = document.createElement("div")  
        col2.id = "col2";  
        let col3 = document.createElement("div");  
        col3.id = "col3";  
        //naslov  
        let title = document.createElement("h1");  
        title.innerText = res[i].title  
        //adresa  
        let address = document.createElement("p")  
        address.classList.add("address")  
        address.innerText = res[i].adress
```

```
        //number of rooms  
        let numberOfRooms =  
document.createElement("p")  
        numberOfRooms.classList.add("numberOfRooms")
```

```

        numberOfRooms.innerHTML = "number of rooms:
" + res[i].numberOfRooms
        let city=document.createElement("p")
        city.classList.add("city")
        city.innerHTML=res[i].city.cityname

```

```

//area

```

```

        let area = document.createElement("p");
        area.classList.add("area")
        area.innerHTML = res[i].area +
"m<sup>2</sup>";

```

```

        //area num of rooms holder
        let areaRoomDiv =
document.createElement("div")
        areaRoomDiv.classList.add("areaRoomDiv")

```

```

        areaRoomDiv.appendChild(area)
        areaRoomDiv.appendChild(numberOfRooms);

```

```

        //price
        let price= document.createElement("p")
        price.classList.add("price")
        price.innerHTML = `${res[i].price}&euro;`
        //price sq/m
        let priceSQM = document.createElement("p")
        priceSQM.classList.add("priceSqm")
        priceSQM.innerHTML =
Math.round(res[i].price / res[i].area) +
&euro;/m<sup>2</sup>
        col3.appendChild(priceSQM)
        col3.appendChild(price);
        col2.appendChild(title)
                col2.appendChild(city)
        col2.appendChild(adress)

        col2.appendChild(areaRoomDiv)
        apartmentone.appendChild(col1)

```

```
apartmentone.appendChild(col2)
apartmentone.appendChild(col3)
```

```
apartmentList.appendChild(apartmentone)
}
}
```

```
function addCityForSearch(element, e){
    let datalist = document.querySelector("#city");
    let selectedCitiesData =
document.querySelector("#selectedCities");

    if(element.value.length == 5){

        if(e.keyCode == 13 ){
            let el =
document.querySelector("option[value='" + element.value
+"']");
            if(el){
                filter.city.push(el.value)
                let newcity =
document.createElement("div")
                let span =
document.createElement("span")
                let icon =
document.createElement("span")
                icon.classList.add("fa-
solid","fa-xmark", "icon")
                span.innerText = el.innerText;
                newcity.appendChild(span)
                newcity.appendChild(icon)
                newcity.setAttribute("data-
city-id", el.value);
                newcity.setAttribute("data-
city-name", el.innerText );
                selectedCitiesData.appendChild(
newcity);
```



```
        filterAll()  
    })
```

```
        let select =  
document.getElementsByName("tipObjekta");  
        select.forEach(e=>{  
            e.addEventListener("click", function (ex) {  
                if(this.checked){  
                    filter.typeOfObject.push(this.value)  
                }  
                else{  
                    let index =  
filter.typeOfObject.indexOf(this.value);  
                    if (index !== -1) {  
                        filter.typeOfObject.splice(index, 1);  
                    }  
                }  
                filterAll();  
            })  
        })  
    })
```

```
let search = document.querySelector("#search")  
search.addEventListener("input", function(){  
    filter.search=this.value;  
    filterAll();  
})  
let orderby = document.querySelector("#orderby")  
orderby.addEventListener("change", function(){  
    filter.orderby = this.value;  
    filterAll()  
})
```

```
    })
```

```
function filterAll(){  
    var newArray = retrived;
```

```

        if(filter.city.length){
            newArray = newArray.filter((e)=>{ return
filter.city.includes(String(e.city.citycode))
            })

```

```

        }
        if(filter.typeOfObject.length){

            newArray = newArray.filter((e)=>{
                return
filter.typeOfObject.includes( String(e.typeObject.typeO
bject))
            })
        }
        if(filter.search!=""){
            let reg = new
RegExp(filter.search.toLowerCase())
            newArray = newArray.filter((e)=>{
                return reg.test(e.title.toLowerCase())})
        }
        if(Number(filter.orderby)){
            if(filter.orderby==1){
                newArray.sort((a, b) => a.price -
b.price);
            }
            else if(filter.orderby==2){
                newArray.sort((a, b) => b.price - a.price);
            }
            else if(filter.orderby==3){
                newArray.sort((a, b) => a.id - b.id);
            }
            else if(filter.orderby==4){
                newArray.sort((a, b) => b.id - a.id);
            }
        }
        filter.page=0;
        filter.max_pages = Math.ceil(newArray.length/20)

```

```
    addApartmentToList(newArray);
    paginationSet(newArray);
}
```

```
function paginationSet(res){

    let pagination =
document.querySelector("#pagination");
    pagination.innerHTML=""
```

```
    if(filter.page < 4){
        for(let i =0; i<6 && i<filter.max_pages ;i++){
            let span = document.createElement("span");
            span.innerHTML = i+1
            span.setAttribute("data-page", i);
            if(filter.page == span.getAttribute("data-
page")){
                span.classList.add("active");
            }
            span.addEventListener("click", function(){
                filter.page = this.getAttribute("data-
page");

                addApartmentToList(res)
                paginationSet(retrived)

            })
            pagination.appendChild(span)
        }
    }
    else{
        for(let i =filter.page-3, j = 1; j<7
&& i<filter.max_pages ; j++,i++){
            let span = document.createElement("span");
            span.innerHTML = i+1
            span.setAttribute("data-page", i);
            if(filter.page == span.getAttribute("data-
page")) ){
```



```

        span.classList.add("active");
    }
    span.addEventListener("click", function(){
        paginationSet(retrived)
        addApartmentToList(res);
        filter.page = this.getAttribute("data-
page");

        //getAllApartments()
    })
    pagination.appendChild(span)
}
}
}
}

```

## Public/js/header.js

```

window.addEventListener("load", function(){
    let xhr = new XMLHttpRequest();
    xhr.open("GET", 'header', true);
    xhr.setRequestHeader("Content-type",
"application/json");
    xhr.send()
    try{
        let header =
document.createElement("header");
        let footer =
document.createElement("footer");

        let ul = document.createElement("ul")
        let ul2footer =
document.createElement("ul");

        xhr.onreadystatechange = function(){

```

```

        if(xhr.readyState
=== XMLHttpRequest.DONE){
            if(xhr.status ===200){
                let result = xhr.responseText;
                if(result !== "error"){
                    result = (JSON.parse(result))

```

```

                for (let key in result) {
                    if
(result.hasOwnProperty(key)) {

                        if(!
( /^child[\d]+$/.test(key))){

                            for(let a in
result[key]){

                                let x =
document.createElement("a");

                                let li =
document.createElement("li");

                                result[key][a].title
                                x.innerHTML =
                                result[key][a].href;
                                x.href =
                                li.id ='parent'
                                +key

                                li.appendChild(x)

                                ul.appendChild(li);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        for(let key in result){
            if
(result.hasOwnProperty(key)) {

                if(!
( /^child[\d]+$/.test(key))){

```

```

                                for(let a in
result[key]){
                                    if(result[key][
a].title == "Account") continue;
                                let x =
document.createElement("a");
                                    let li =
document.createElement("li");
                                result[key][a].title
                                    x.innerHTML =
                                result[key][a].href;
                                    x.href =
                                +key
                                    li.id ='parent'

                                li.appendChild(x)
                                ul2footer.appendChi
ld(li);
                                }
                            }
                        }
                    }

                let body =
document.querySelector("body");
                body.insertAdjacentElement("aft
erbegin", header)

```

```

        body.insertAdjacentElement("beforeend", footer)

        let i =
document.createElement("i")
        i.classList.add("fa-solid", "fa-bars", "toggle")

        header.appendChild(ul)
        header.appendChild(i)
        i.addEventListener("click",
function(){
            ul.classList.toggle("d-flex")
        })
        footer.appendChild(ul2footer)

        for(let key in result){
            if(/^child[\d]+$/.test(key))
{
                let id = /[^\d]+$/.
                id = id.exec(key)

                id.forEach((element )=>
{
                    let parent =
document.querySelector(`#parent${element}`)

                    if(!parent.querySelector("ul")){
                        var ul_child =
document.createElement("ul")
                        parent.appendChild(ul_child)
                    }
                })
            }
        }
    }
}

```

```

result[key]){
    document.createElement("li");
    document.createElement("a")
    result[key][b]["href"]
    result[key][b]["title"];
    a)
    Child(li)

    for(let b in
        let li =
        let a =
        a.href =
        a.innerText =
        li.appendChild(
        ul_child.append

    }

    parent.addEventListener
        let child_ul =
        child_ul.style.

    })
    parent.addEventListener
        let child_ul =
        child_ul.style.

    })
    })

    }
    }
    let xhrLogged = new
XMLHttpRequest();
    xhrLogged.open("GET",
"loggedIcon");

```

```

        xhrLogged.setRequestHeader("Content-type", "application/json");
        xhrLogged.onreadystatechange =
function(){
            if(xhrLogged.status == 500
&& xhrLogged.readyState ==4){

                let res
= JSON.parse(xhrLogged.responseText)
                if(!res["err"]){
                    let header =
document.querySelector("header");
                    let a =
document.createElement("a")
                    a.href=
`/userProfile`
                    a.id="user";
                    a.innerHTML =
res["f_name"] + " " + res["l_name"];
                    header.appendChild(
a);
                }
            }
        }
    }
}

```

```

        }else{

        }

    }
    xhrLogged.send();

}

```

```

    }
  }
}

}catch(err){
  console.log(err)
}

})

```

## Public/js/imageCarousel.js

```

window.addEventListener("load", function(){
  let images = document.querySelectorAll("#images
div");
  let image_array = [];
  images.forEach(e=>{
    let link = getComputedStyle(e).backgroundImage;
    let imageUrl = link.replace('url(',
'').replace(')', '');
    image_array.push(imageUrl);
  })

  let images_len = image_array.length;

```

```

    let current_image;
    images.forEach((el)=>{
        el.addEventListener("click", function(){
            current_image =
Number(this.getAttribute("image-id"));
            let body = document.querySelector("body")
            body.style.overflow="hidden";
            let overlay = document.createElement("div")
            overlay.classList.add("overlay");
            let img = document.createElement("img");
            let imgholder =
document.createElement("div");
            imgholder.classList.add("imgholder")

            img.src=image_array[current_image]
            img.alt = this.src;
            imgholder.appendChild(img)

            let x = document.createElement("span");
            let left = document.createElement("span");
            left.innerText="<";
            left.classList.add("left")
            left.addEventListener("click", function(){
                current_image--;
                if(current_image<0)
                {
                    img.src = image_array[images_len-1];
                    current_image = images_len-1;
                }else{
                    img.src =
image_array[current_image];
                }
            })

            let right = document.createElement("span");
            right.addEventListener("click", function(){

```



```

        current_image++;
        if(current_image > images_len-1 ){
            current_image = 0;
            img.src =
image_array[current_image];
        }else{
            img.src = image_array[current_image];
        }

```

```

    })
    right.classList.add("right")
    right.innerText = ">"
    x.innerText = "X";
    x.classList.add("x");
    imgholder.appendChild(left);
    imgholder.appendChild(right)
    overlay.appendChild(imgholder);
    overlay.appendChild(x);

    x.addEventListener("click", function(){
        body.style.overflow="";
        ovly =
document.querySelector(".overlay");
        ovly.remove()
    })
    let realestatepost =
document.querySelector("#realestatepost")
    realestatepost.insertAdjacentElement("beforeend", overlay);

```

```

    })
    })

```

```

});

```

## Public/js/lastClicked.js

```
window.addEventListener("load",function(){
    let
title=this.document.querySelector("title").innerText;

    localStorage.setItem("titleClicked", title);
    let img = this.document.querySelector(".main")
    let img_src=
this.getComputedStyle(img).backgroundImage
    let href=this.window.location.href;

    this.localStorage.setItem("href", href);

    img_src = img_src.substring(5, img_src.length - 2)

    this.localStorage.setItem("img_srcClicked", img_src)
    let price =
this.document.querySelector("#textprice").innerText;
    this.localStorage.setItem("priceClicked", price)

})
```

## Public/js/loadLastClicked.js

```
window.addEventListener("load", function(){
    if(this.localStorage.getItem("titleClicked")
    && this.localStorage.getItem("img_srcClicked")
&&this.localStorage.getItem("titleClicked")){
        let div = this.document.createElement("a")
        div.href=this.localStorage.getItem("href")
        div.id = "lastClicked";
        let lastApartmant =
this.document.createElement("h2")
        lastApartmant.innerText="Lastly seen";
```

```

        let img = this.document.createElement("img")

        img.src=this.localStorage.getItem("img_srcClicked")

        img.alt = "lastClicked"
        img.style.width="300px"
        let title = this.document.createElement("h1")
        title.innerHTML=this.localStorage.getItem("titleClicked");
        let price = this.document.createElement("p")
        price.innerHTML =
this.localStorage.getItem("priceClicked");
        div.appendChild(lastApartment)
        div.appendChild(img)
        div.appendChild(title)
        div.appendChild(price)
        let filter =
this.document.querySelector("#filter")
        filter.appendChild(div)

    }
})

```

## Public.js/login.js

```

let form = document.querySelector("#firstForm")
let next = document.querySelector("#login")
let mail = document.querySelector("#email");
let pw = document.querySelector("#password")
mail.addEventListener("input", function(){
    if(
        !confemail(mail)){
            addError(mail)
        }
    else{

```

```

        removeError(mail)
    }
})
pw.addEventListener("input", function () {
    if(!confpassword(pw)){
        addError(pw)
    }
    else{
        removeError(pw)
    }
})

next.addEventListener("click", function(){
    if(    confemail(mail) &&    confpassword(pw)){
        let element;
        if(element =document.querySelector("#error")){
            element.remove();

```

```

        }
        let xhr = new XMLHttpRequest();
        xhr.open("POST","login", true);
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        let param={
            email : mail.value,
            password :pw.value
        }
        xhr.send(JSON.stringify(param))
        xhr.onreadystatechange = function(){
            if(xhr.readyState
=== XMLHttpRequest.DONE){
                try{

```

```

            let resp = xhr.response;

```

```

            if(xhr.status ===200){

```

```

window.location.href = "allApartments"

    }
    else if(xhr.status ===401){
        let error =
document.createElement("p")
        error.innerText = "You are
forbidden from accessing to your account, please
contact support";
        error.id="error"
        next.insertAdjacentElement("beforeb
egin", error)
    }
    else if(xhr.status===406){
        if(resp[1] === "pwormailnomatch"){
            let error =
document.createElement("p")
            error.innerText = "Mail or
Password is incorret";
            error.id="error"
            next.insertAdjacentElement("bef
orebegin", error)
        }
    }
    else if(xhr.status ===403){
        let error =
document.createElement("p")
        error.innerText = "Dont change pre
defined name attributes";
        error.id="error"
        next.insertAdjacentElement("beforeb
egin", error)
    }
    else if(xhr.status === 503){
        let error =
document.createElement("p")

```

```

        error.id="error"
        error.innerText = "Servers are down
please try again in a few minutes";
        next.insertAdjacentElement("beforeb
egin", error)
    }
}
catch(err){
    errorPopUp(err);
}
}
}
}
else{
    mail.dispatchEvent(new Event("input"));
    pw.dispatchEvent(new Event("input"))
}
})

```

## Public/js/regex.js

```

function confName(el){
    let reg = /^[A-Z]{1}[a-z]{1,9}$/

    return reg.test(el.value)
}
function confemail(e){
    let reg = /^[w\d\_-\+\.\]\@[w\.\.]+\$/
    return reg.test(e.value)
}
function confpassword(el){
    let len = (el.value.length > 8)? true:false;
    let digit = /[d]/.test(el.value);
    let letters = /[w]/.test(el.value);
    if(len && digit && letters) return true;
}

```

```
        return false;
    }
    function titleconf(el){
        if(el.value.length>2){
            return true;
        }
        return false;
    }
    function kvadratura(el){
        if(el.value >=1){
            return true;
        }
        return false;
    }
    function selectValidacija(el){
        if(el.value != 0){
            return true;
        }
        return false;
    }
    function priceCheck(el){
        if(el.value>0){
            return true;
        }
        return false;
    }
    function adresa(el){
        if(el.value.length >3){
            return true;
        }
        return false;
    }
    function radioconf(el){
        for(let x of el){
            if(x.checked){
                return true;
            }
        }
    }
}
```

```

    }

    }
    return false;
}

```

```

function opis(el, len=15){
    if(el.value.length > len){
        return true;
    }
    return false;
}

```

```

function sobe(el){
    if(el.value >=0.5 && el.value <=10){
        return true;

    }
    return false;
}

```

```

function passwordsEquals(x, y){
    return x.value == y.value

```

```

}
function addError(x){
    let name_error = x.id + '_error'
    if(document.querySelector(`#${name_error}`)){

```

```

    }
    else{
        let elementP = document.createElement("p")
        elementP.id = name_error;
        if(x.hasAttribute("data-error")){
            elementP.innerText = x.getAttribute("data-
error");
        }
        else{

```



```

        elementP.innerText = "This field is
mandatory"
    }
    elementP.classList.add("error")
    x.insertAdjacentElement('afterend',elementP);
}
}
function removeError(x){
    let name_error = x.id + '_error'

    let errorP =
document.getElementById(`${name_error}`)
    if(errorP) errorP.remove()
}

```

**Public/js/support.js**

```

let form = document.querySelector("#firstForm");
form.addEventListener("submit", function(e){

    let fname, lname, email, error=0;

    try{
        fname = document.querySelector("#fName");
        error += confName(fname)?0:1;
        lname = document.querySelector("#lName");
        error += confName(lname)?0:1;
        email = document.querySelector("#email");
        email+= confemail(email)?0:1;

    }catch(error){

    }finally{

        let title =
document.querySelector("#titleSupport");
        error+= opis(title, 5)?0:1;
    }
}

```

```

        let content =
document.querySelector("#contentSupport");
        error+= opis(title, 13)?0:1
        if(!error){

        }else{
            e.preventDefault();
            const fields =
form.querySelectorAll('input');
            fields.forEach(field => {
                field.dispatchEvent(new Event('blur'));
            });
        }
    }
}

```

```

}))

```

```

window.addEventListener("load", function(){
    try{
        let fname = document.querySelector("#fName");

        let lname = document.querySelector("#lName");
        let email = document.querySelector("#email");
        let array = [fname, lname]
        array.forEach(e=>{

            e.addEventListener("blur", function(){
                if(confName(this)){
                    removeError(this);
                }
                else{
                    addError(this)
                }
            })
        })
        email.addEventListener("blur", function(){
            if(confemail(this)){
                removeError(this);
            }
        })
    }
})

```

```

        }
        else{
            addError(this)
        }
    })

}catch(error){

}

```

```

    let title = document.querySelector("#titleSupport");
    let content =
document.querySelector("#contentSupport");
    let niz2 = [title, content]
    niz2.forEach( (e,x)=>{
        e.addEventListener("blur", function(){
            if(opis(e, (x==0)?5:13)){
                removeError(this);
            }
            else{
                addError(this)
            }
        })
    })
})

```