VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

Dokumentacija

https://fidelioperfumesphp.000webhostapp.com/index.php

https://fidelioperfumes.netlify.app/

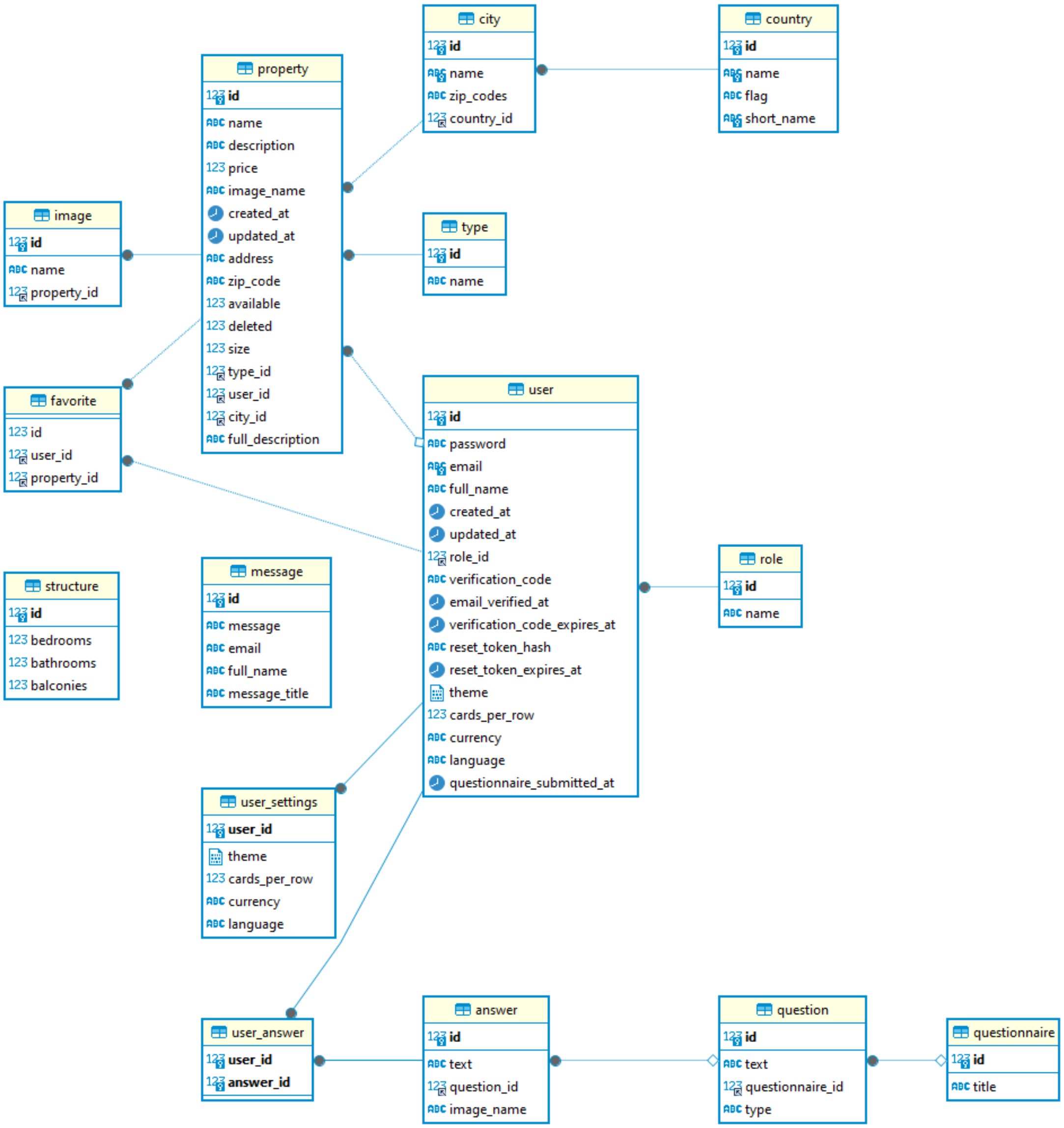
Veselin Ševo 23/22 9.03.2023

Korisceni programski jezici

- HTML

- TAILWIND(CSS)
- JS
- PHP

Slika seme baze podataka



Opis funkcionalnosti

Dinamcki Header

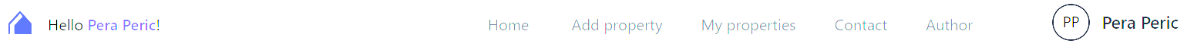
- Dinamicko ispisivanje sadrzaja headera

Ukoliko korisnik nije ulogovan:

- U headeru se pojavljuje Login i Register button

Ukoliko korisnik jeste ulogovan:

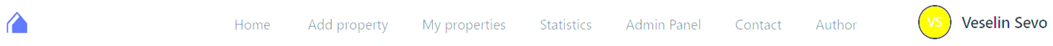
- U headeru se pojavljuje Korisnicka slika sa linkovima ka profilu
- Add properties link
- My properties link
- Ukoliko se korisnik upravo ulogovao poruka: Hello (ime korisnika). Nakon sto se prikaze vise se ne prikazuje dok se korisnik opet ne uloguje.
-



Hello Pera Peric!

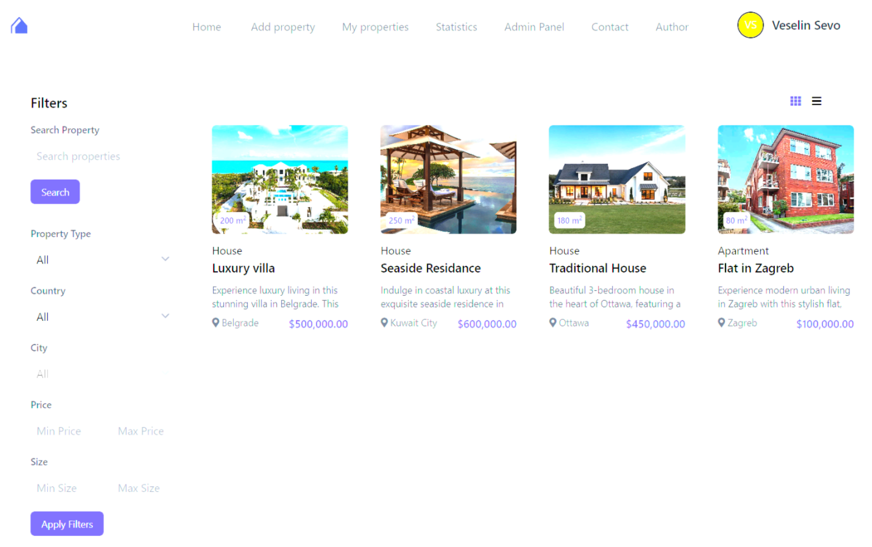
Ukoliko korisnik jeste ulogovan kao Admin:

- Prikazuje se sve isto kao kod obicnog ulogovanog korisnika +
- Statistics link za uvid u statistiku odgovora iz ankete
- Admin Panel



Home Page

- Dinamicko ispisivanje sadrzaja
- Search bar za pretragu nekretnina po imenu
- Filteri ciji se sadrzaj dobija dinamicki - iz baze
  - Filter za grad se generise i moze selektovati tek nakon selekcije drzave prvo
- Ikonice za promenu pogleda - dal prikazivati 3 ili 4 nekretnine po redu



Filters

Search Property: Search properties

Property Type: All

Country: All

City: All

Price: Min Price, Max Price

Size: Min Size, Max Size

House
Luxury villa
Experience luxury living in this stunning villa in Belgrade. This
Belgrade $500,000.00

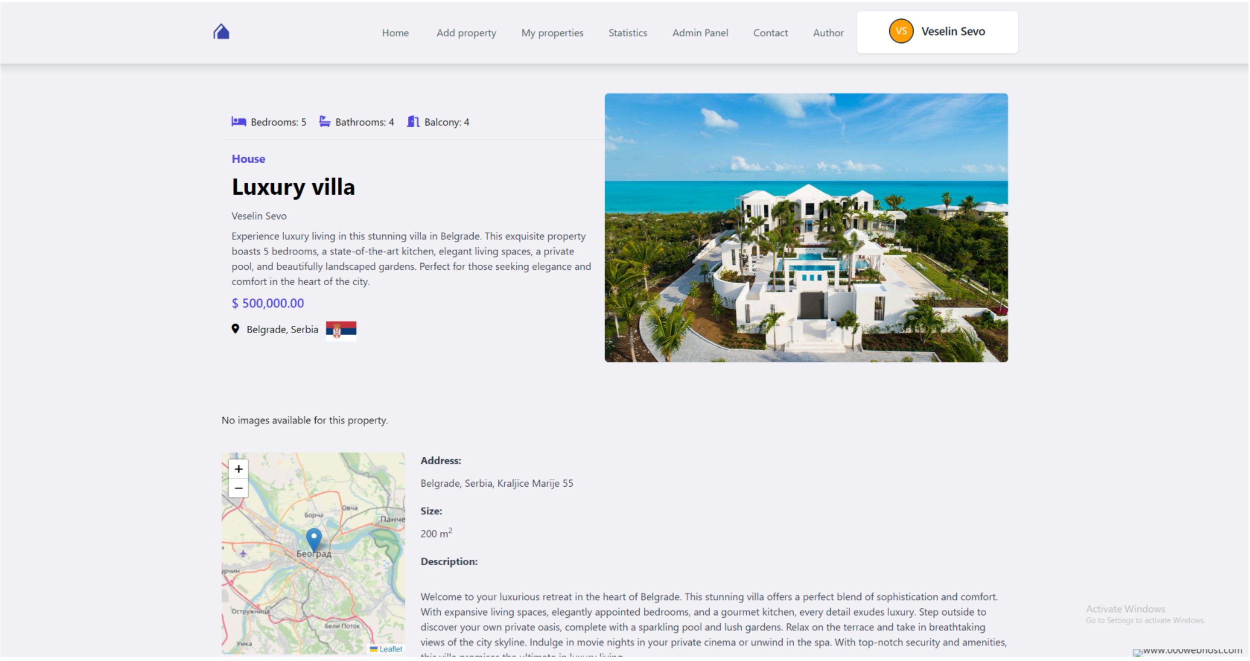
House
Seaside Residance
Indulge in coastal luxury at this exquisite seaside residence in
Kuwait City $600,000.00

House
Traditional House
Beautiful 3-bedroom house in the heart of Ottawa, featuring a
Ottawa $450,000.00

Apartment
Flat in Zagreb
Experience modern urban living in Zagreb with this stylish flat.
Zagreb $100,000.00

Property Page

- Prikaz detalja nekretnine(struktura nekretnine, broj soba, detaljan opis koji korisnik unosi u editoru)
- Dinamicka galerija slika
- Prikaz imena vlasnika koji je okacio nekretninu
- Prikaz grada nekretnine na mapi i zastave zemlje u kojoj se nalazi nekretnina



Bedrooms: 5 Bathrooms: 4 Balcony: 4

House

Luxury villa

Veselin Sevo

Experience luxury living in this stunning villa in Belgrade. This exquisite property boasts 5 bedrooms, a state-of-the-art kitchen, elegant living spaces, a private pool, and beautifully landscaped gardens. Perfect for those seeking elegance and comfort in the heart of the city.

$ 500,000.00

Belgrade, Serbia

No images available for this property.

Address:

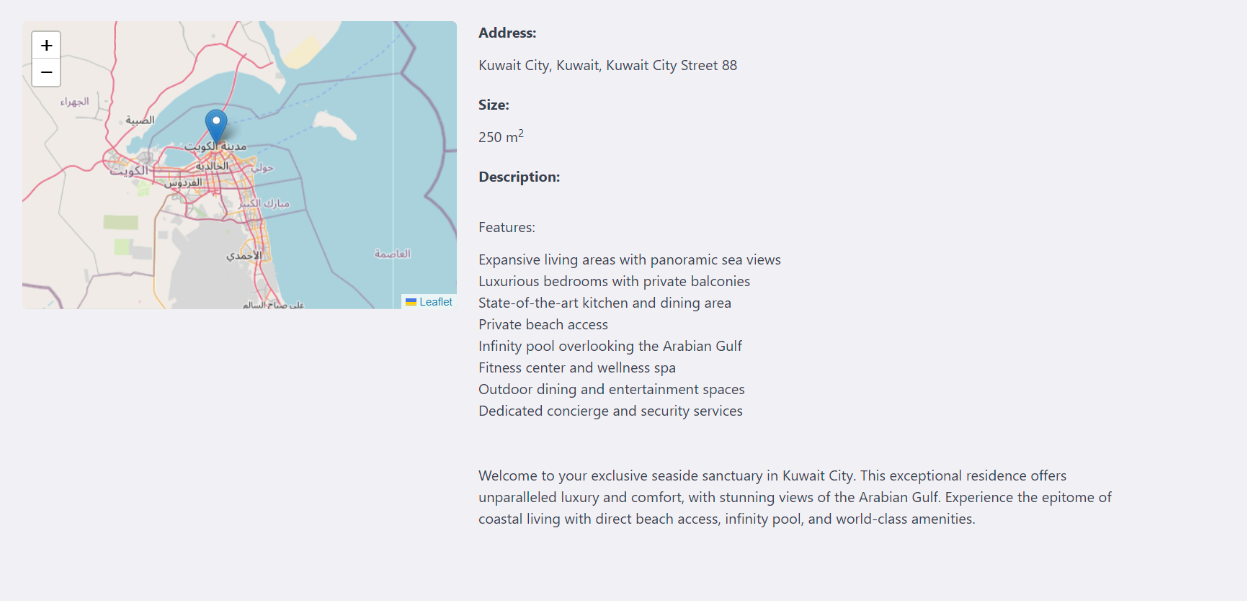
Belgrade, Serbia, Kraljice Marije 55

Size:

200 m^2

Description:

Welcome to your luxurious retreat in the heart of Belgrade. This stunning villa offers a perfect blend of sophistication and comfort. With expansive living spaces, elegantly appointed bedrooms, and a gourmet kitchen, every detail exudes luxury. Step outside to discover your own private oasis, complete with a sparkling pool and lush gardens. Relax on the terrace and take in breathtaking views of the city skyline. Indulge in movie nights in your private cinema or unwind in the spa. With top-notch security and amenities, this villa promises the ultimate in luxury living.



Address:

Kuwait City, Kuwait, Kuwait City Street 88

Size:

250 m^2

Description:

Features:

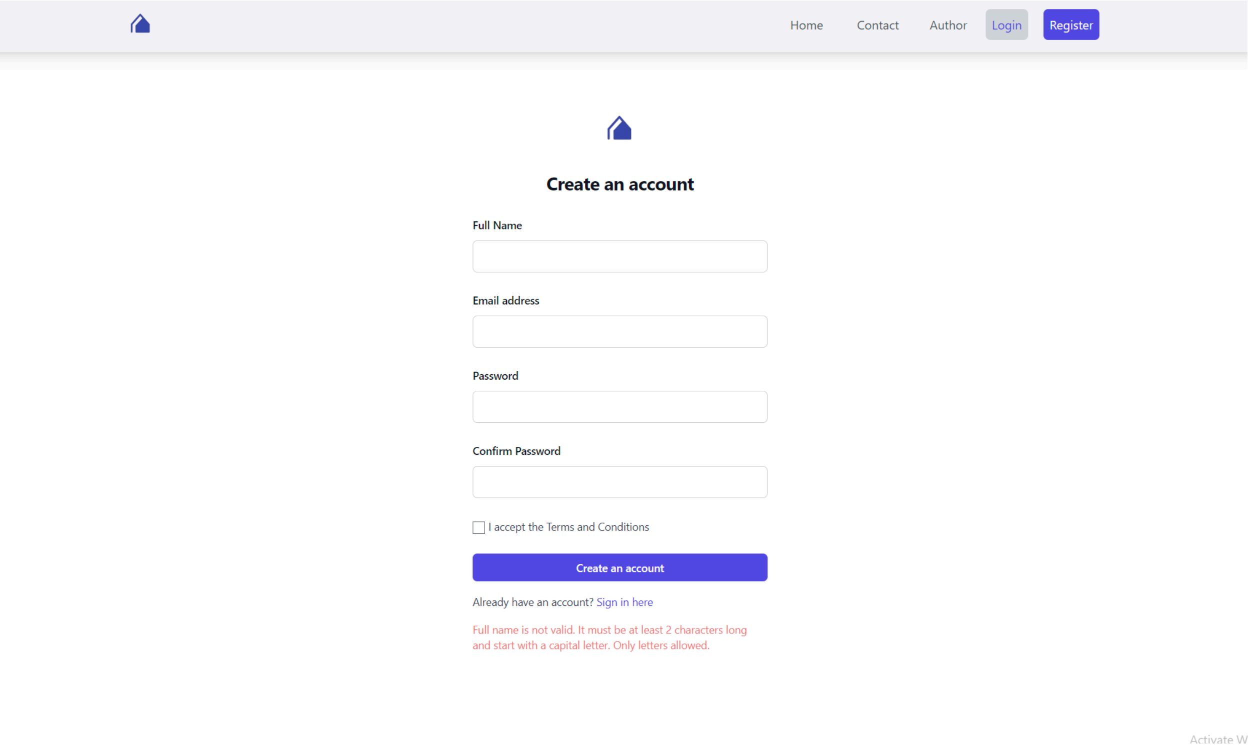
Expansive living areas with panoramic sea views
Luxurious bedrooms with private balconies
State-of-the-art kitchen and dining area
Private beach access
Infinity pool overlooking the Arabian Gulf
Fitness center and wellness spa
Outdoor dining and entertainment spaces
Dedicated concierge and security services

Welcome to your exclusive seaside sanctuary in Kuwait City. This exceptional residence offers unparalleled luxury and comfort, with stunning views of the Arabian Gulf. Experience the epitome of coastal living with direct beach access, infinity pool, and world-class amenities.

Proces i stranice registracije i logovanja

Register Page

- Forma za registraciju korisnika
- Validacija podataka koju je poslao korisnik (front/back)
- Provera da li se slaze sa Terms and Conditions
- Nakon toga slanje mejla korisniku sa kodom za verifikaciju



Create an account

Full Name

Email address

Password

Confirm Password

□ I accept the Terms and Conditions

Already have an account? Sign in here

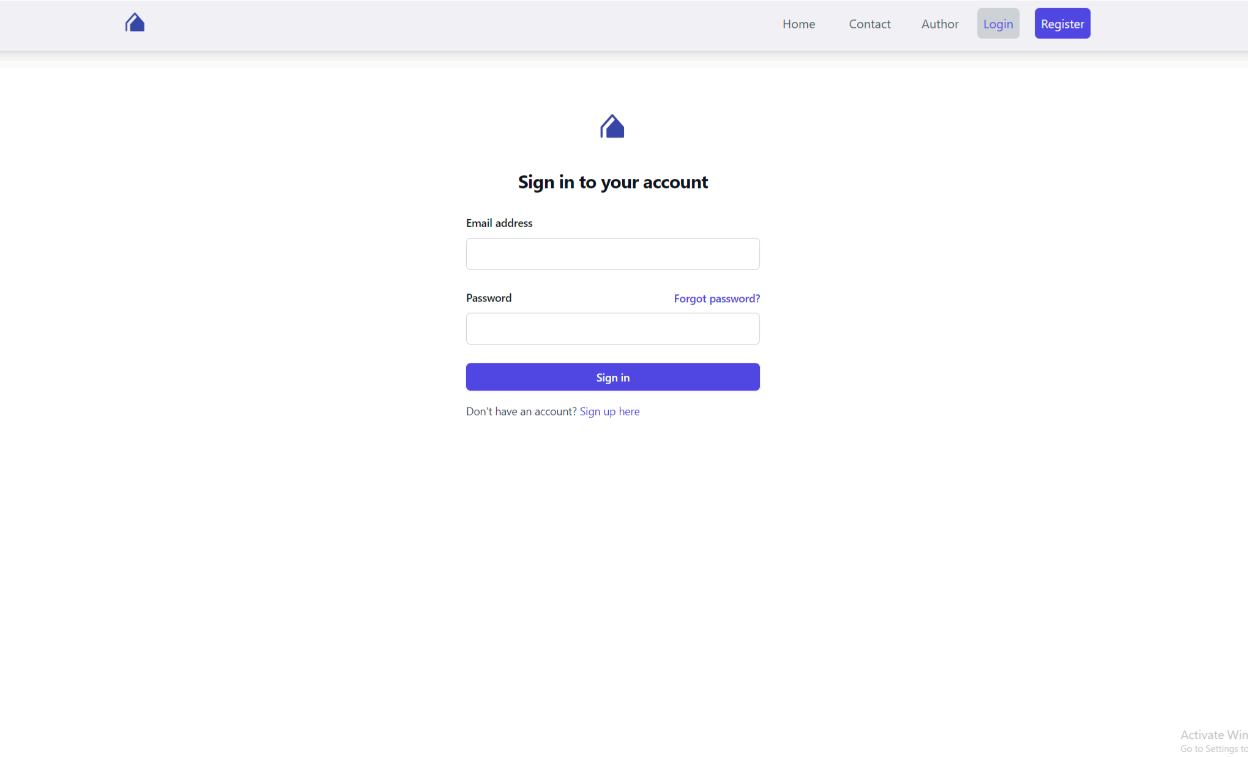
Full name is not valid. It must be at least 2 characters long and start with a capital letter. Only letters allowed.

Verification Code Page

- Polje u koje korisnik mora da unese kod koji mu je stigao na mejl i koji vazi 20 minuta
- Dugme za ponovno slanje koda
- Polje u kojoj se ispisuje poruka o izvrsetku naredbe(Da li je uspesno ponovno slanje koda, ili poruka o pogresnom unosu koda od strane korisnika) pomocu FlashMessage-a, koje na reload stranice nestaje.
- Nakon uspesnog slanja koda redirekcija na Login gde, sada, korisnik moze uspesno da se uloguje

Login Page

- Forma za logovanje korisnika
- Postavljanje informacija o korisniku u sesiju
- Polje u kojoj se ispisuje poruka ukoliko logovanje nije uspesno



Sign in to your account

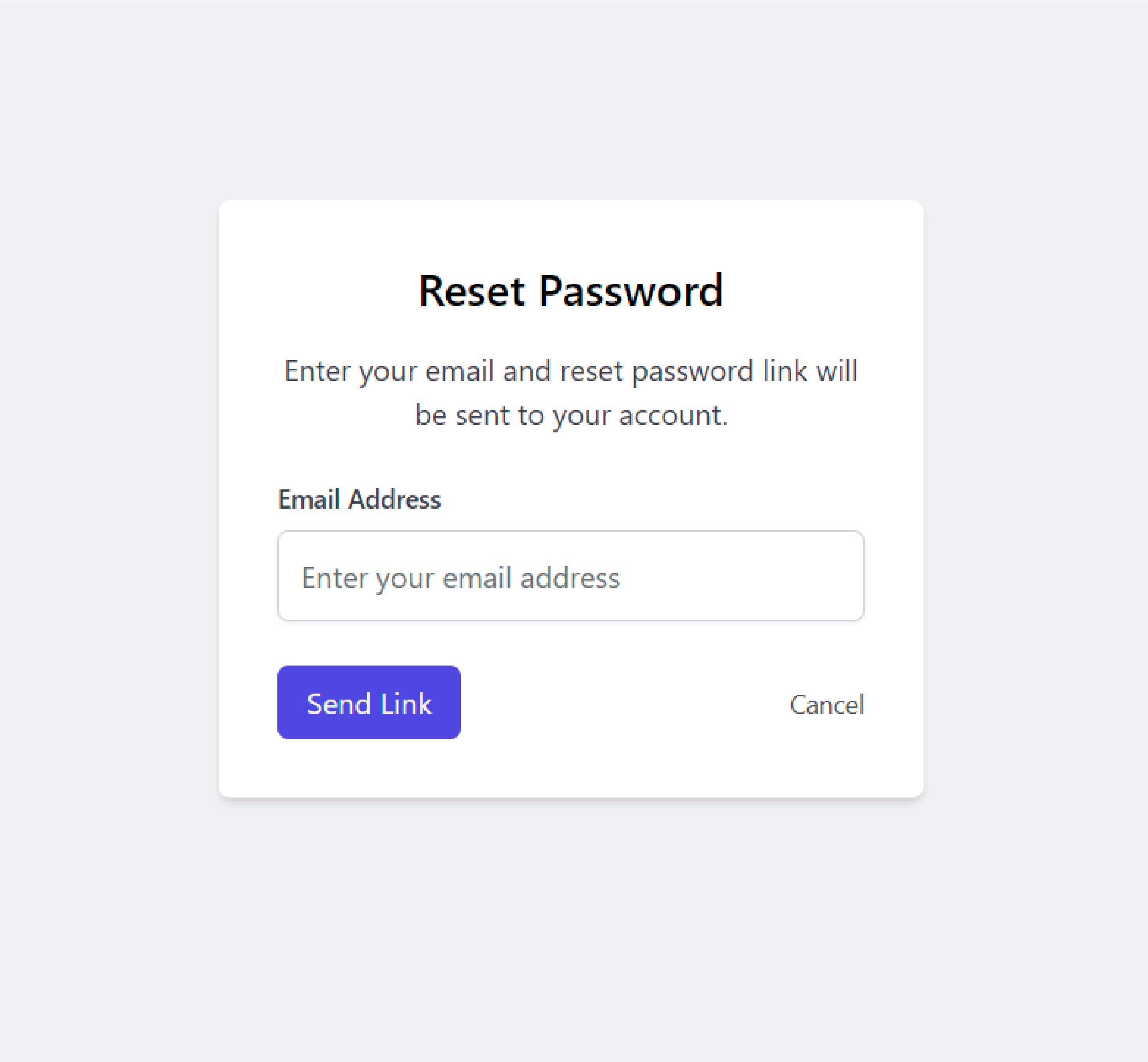
Email address

Password

Don't have an account? Sign up here

Reset Password Page I - Prvi korak, slanje mejla

- Polje u koje korisnik mora da unese mejl adresu na koji ce mu se poslati link za promenu lozinke (koji takodje ima odredjeno vreme trajanja i vazi samo za tog korisnika)
- Dva dugmeta: Send link and Cancel, jedno za slanje mejla na tu adresu, a drugo za povratak nazad.
- Poruka o uspesnosti ili neuspesnosti slanja mejla, pomocu FlashMessage-a, koja na reload stranice nestaje.



Reset Password

Enter your email and reset password link will be sent to your account.

Email Address

Enter your email address

Reset Password Page II - Drugi korak, unos nove sifre

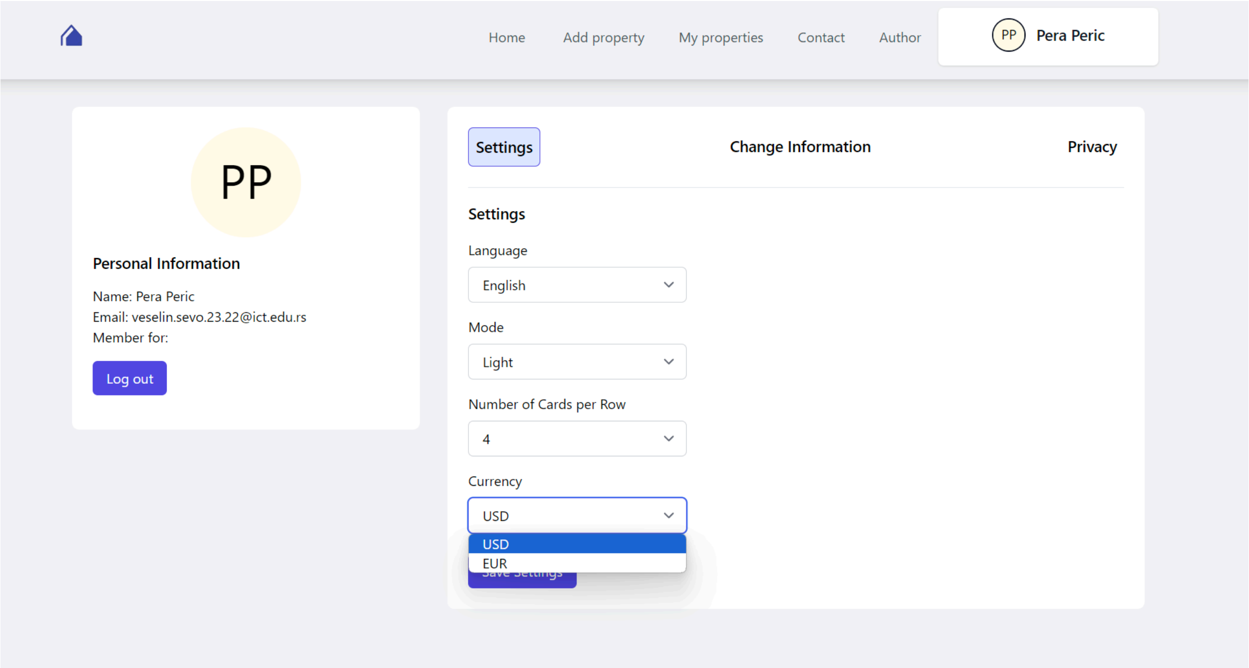
- Polje u koje korisnik mora da unese mejl adresu na koji ce mu se poslati link za promenu lozinke (koji takodje ima odredjeno vreme trajanja i vazi samo za tog korisnika)
- Dva dugmeta: Send link and Cancel, jedno za slanje mejla na tu adresu, a drugo za povratak nazad.
- Poruka o uspesnosti ili neuspesnosti slanja mejla, pomocu FlashMessage-a, koja na reload stranice nestaje.
-

Reset Password Page III - Treci korak, uspesno promenjena sifra

- Poruka o uspesno promenjenoj sifri
- Dugme za odlazak na login stranicu

User Profile Page

- Nalaze se podaci o profilu
- Defaultna slika sa inicijalima iz imena unetog pri registraciji
- Settings deo gde korisnik moze za podesi valutu u kojoj zeli da mu se prikazuju nekretnine (mora nakon toga kliknuti Save Settings dugme)



Personal Information

Name: Pera Peric
Email: veselin.sevo.23.22@ict.edu.rs
Member for:

Settings

Language: English

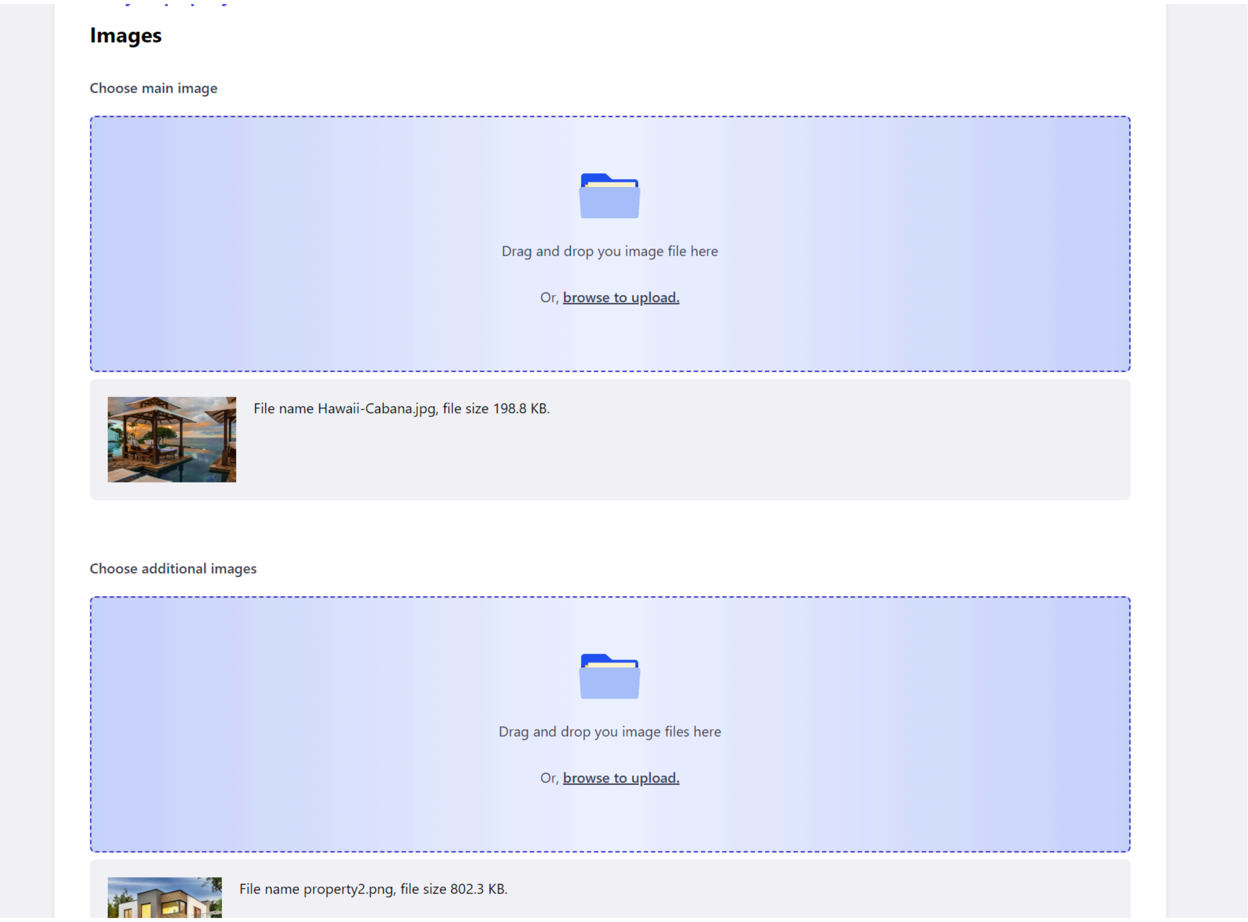
Mode: Light

Number of Cards per Row: 4

Currency: USD

Add Property Page

- Stranica na kojoj korisnik moze da doda nekretninu
- U formi step-ova, unosi deo po deo podataka
- Inputi koji prihvataju privlacenje slike/slika i prikazuju detalje slike
- Dropdown-i cije se opcije dinamciki popunjavanju, gradovi se mogu selektovati tek nakon selektovane drzave



Images

Choose main image

Drag and drop you image file here

Or, browse to upload.

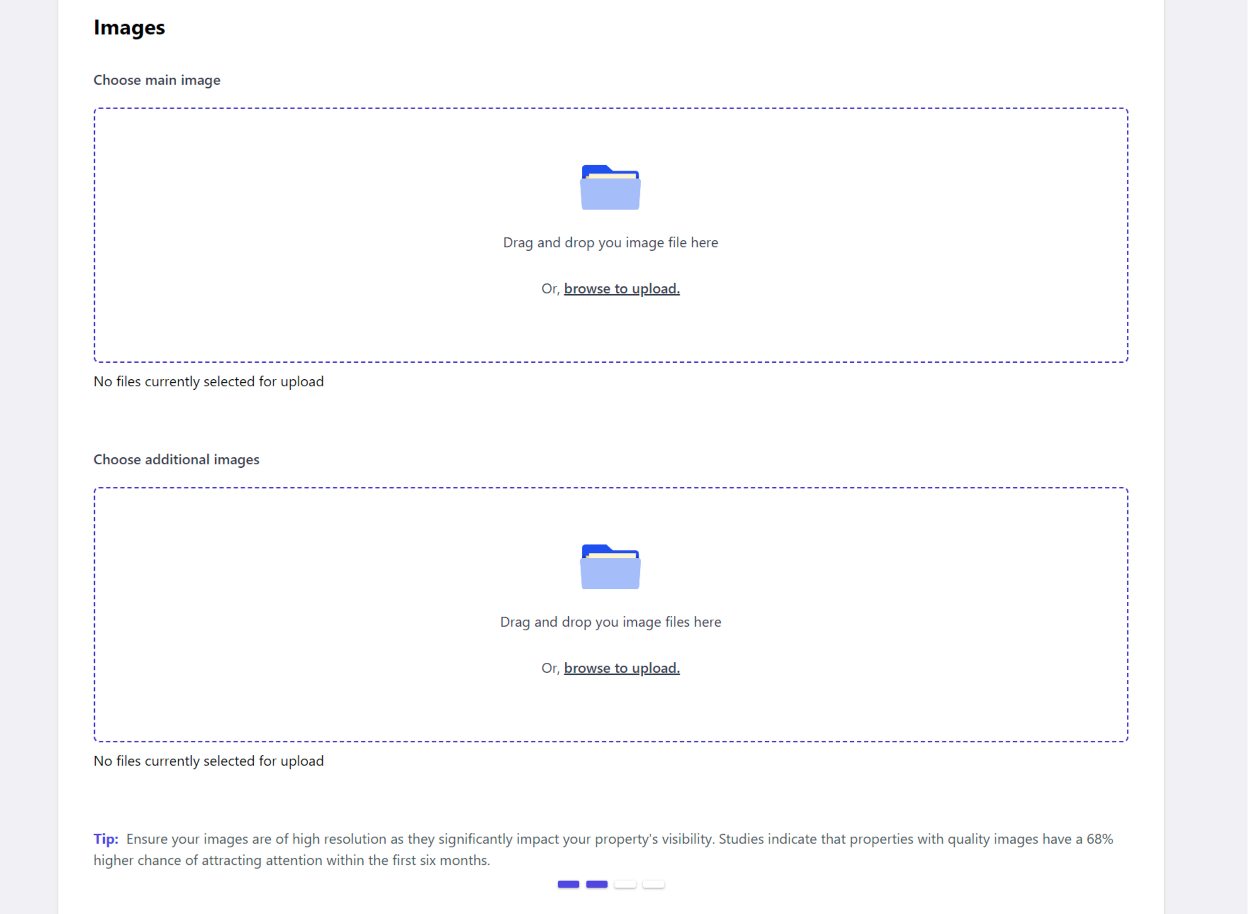
File name Hawaii-Cabana.jpg, file size 198.8 KB.

Choose additional images

Drag and drop you image files here

Or, browse to upload.

File name property2.png, file size 802.3 KB.



Images

Choose main image

Drag and drop you image file here
Or, browse to upload.

No files currently selected for upload

Choose additional images

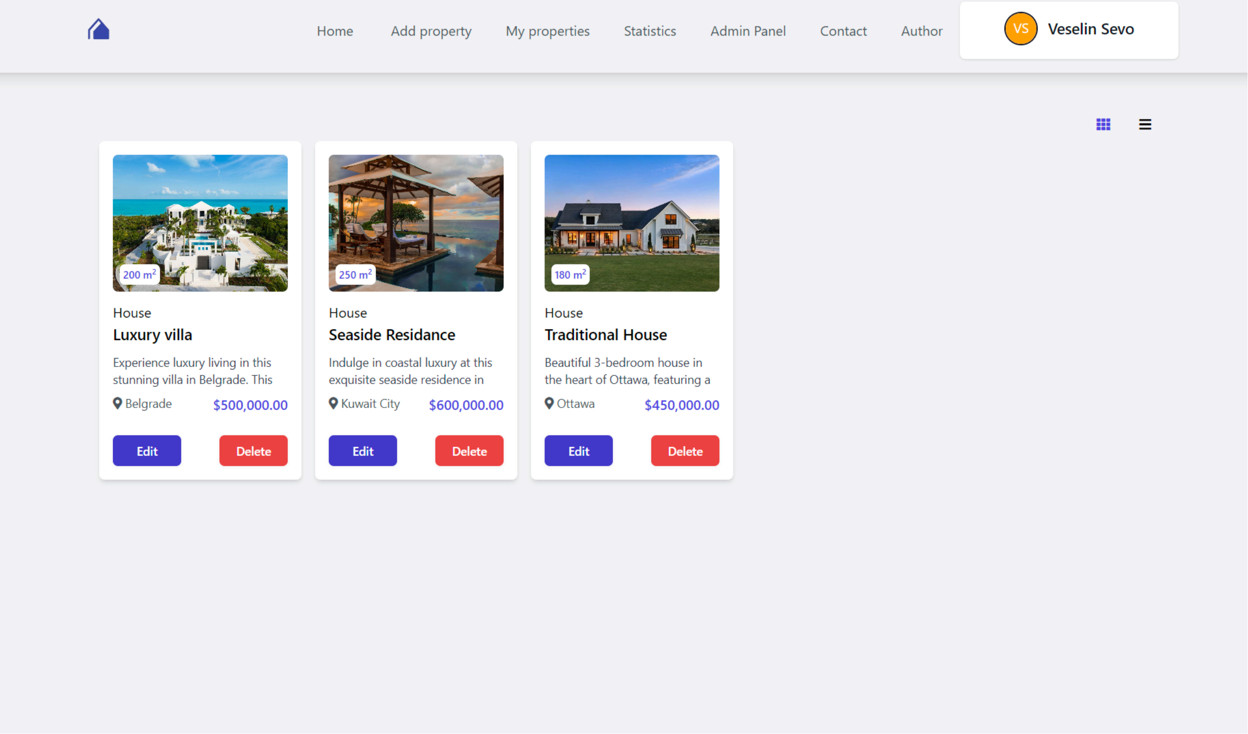
Drag and drop you image files here
Or, browse to upload.

No files currently selected for upload

Tip: Ensure your images are of high resolution as they significantly impact your property's visibility. Studies indicate that properties with quality images have a 68% higher chance of attracting attention within the first six months.

My Properties Page

- Stranica na kojoj korisnik moze da vidi svoje nekretnine
- Za svaku nekretninu ima opciju Delete i Edit
- Inputi koji prihvataju prevlacenje slike/slika i prikazuju detalje slike
- Dropdown-i cije se opcije dinamciki popunjavanju, gradovi se mogu selektovati tek nakon selektovane drzave



200 m²

House

Luxury villa

Experience luxury living in this stunning villa in Belgrade. This

Belgrade $500,000.00

250 m²

House

Seaside Residance

Indulge in coastal luxury at this exquisite seaside residence in

Kuwait City $600,000.00

180 m²

House

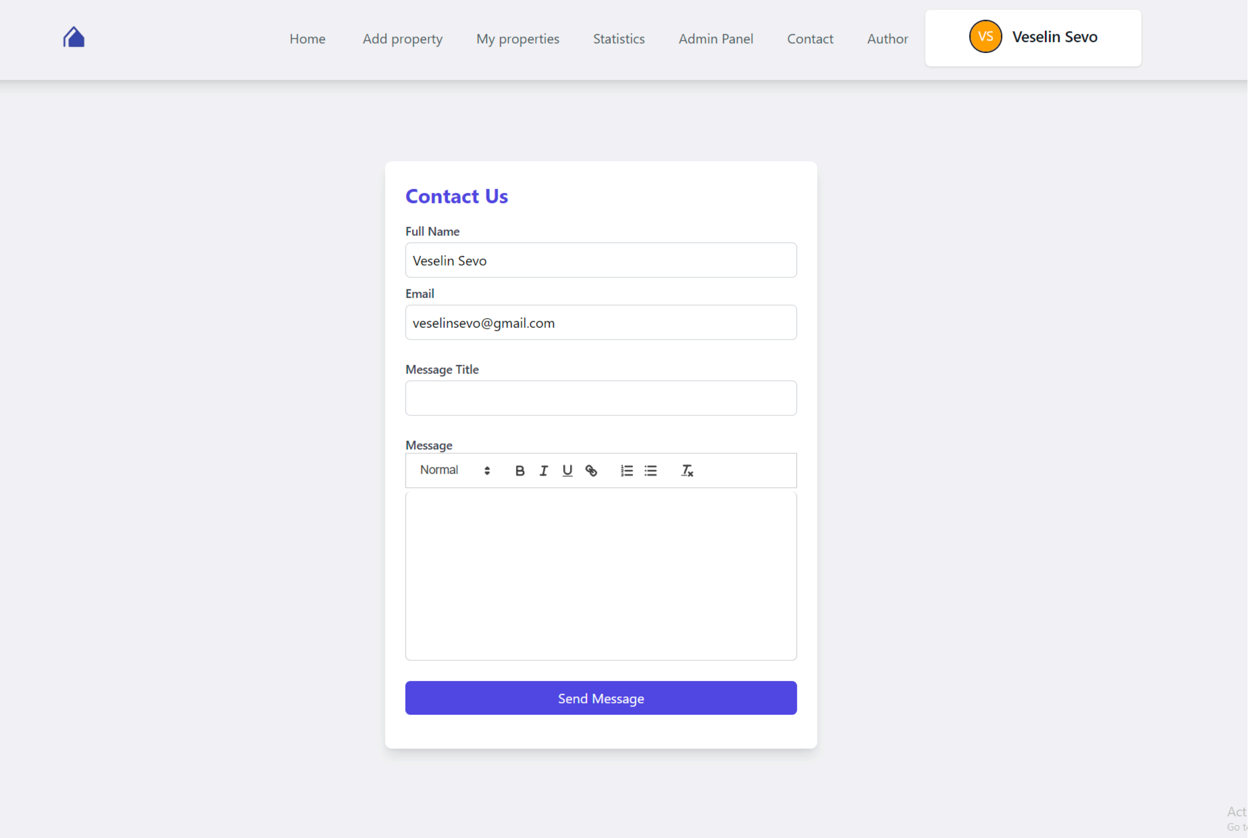
Traditional House

Beautiful 3-bedroom house in the heart of Ottawa, featuring a

Ottawa $450,000.00

Contact Page

- Stranica na kojoj se adminima sajta mogu poslati poruke
- Ako je korisnik ulogovan automatski mu se popunjava ime i email
- Editor u koji se upisuje sadrzaj poruke



Contact Us

Full Name: Veselin Sevo

Email: veselinsevo@gmail.com

Message Title:

Message:

Anketa

- Sadrzaj ankete dolazi iz baze podataka(pitanja, ponudjene opcije…)
- Prikazuje se samo ako je korisnik vec nije popunio, pri uspesnom logovanju korisnika
- Sadrzi dva pitanja na koje se moze izabrati jedna opcija, i jedno gde moze vise opcija

Statistics Page

- Stranica na kojoj se vidi kolicina odgovora za svako pitanje i opciju popunjene ankete
- Tabelarni i Bar Chart prikaz

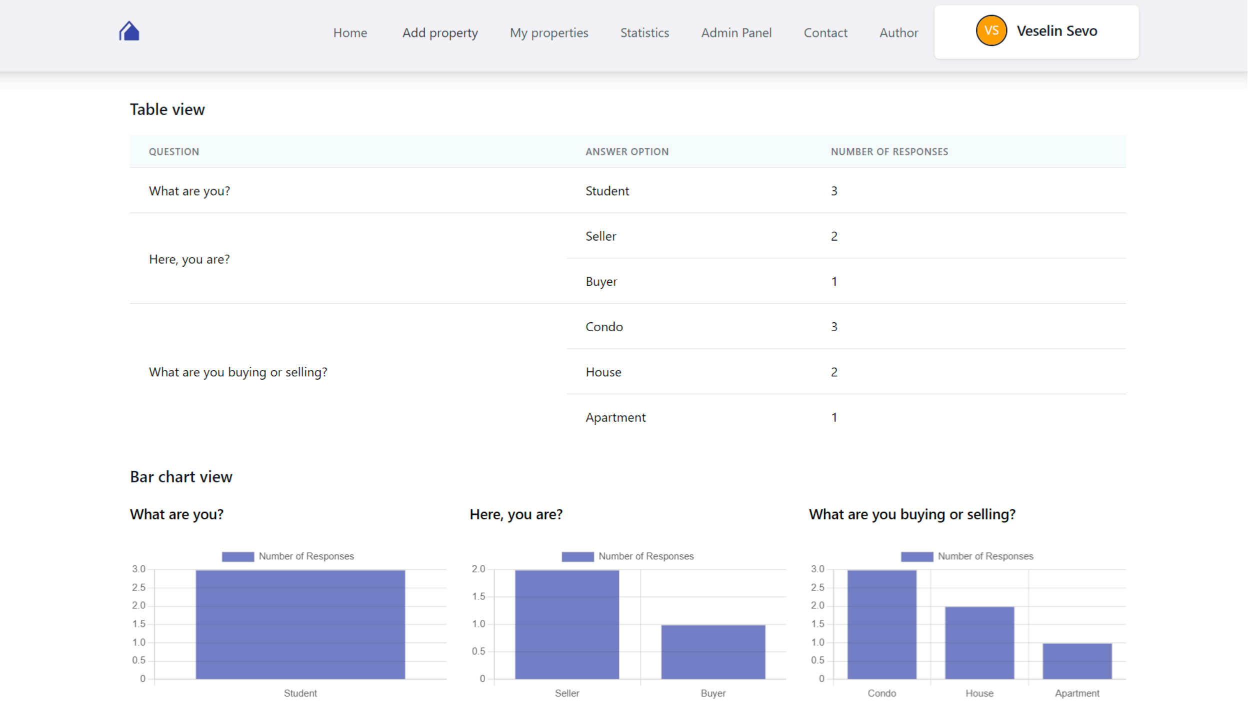


Table view

QUESTION	ANSWER OPTION	NUMBER OF RESPONSES
What are you?	Student	3
Here, you are?	Seller	2
	Buyer	1
What are you buying or selling?	Condo	3
	House	2
	Apartment	1

Bar chart view

What are you?

Here, you are?

What are you buying or selling?

Admin Panel

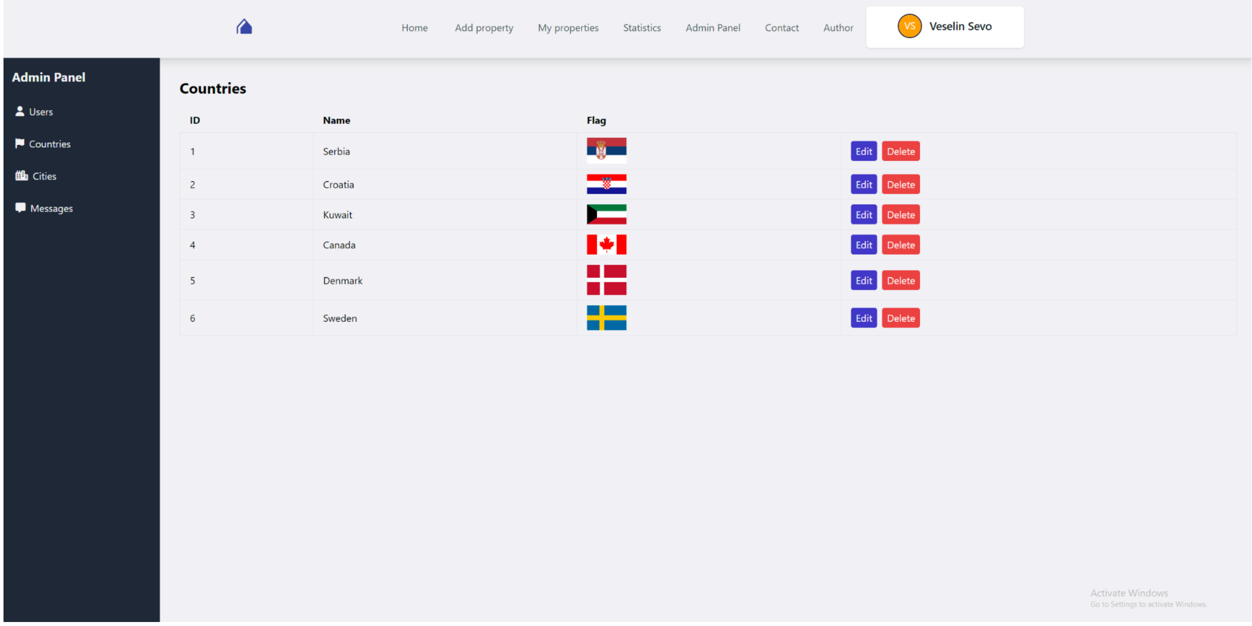
- Sekcija za admine sajta
- Levo sidebar, desno content

Users -> Show Users

- Prikazuju se svi korisnici

Countries -> Show Countries

- Prikazuju se sve drzave
- Svaka ima opciju Edit i Delete
- Po izvrsenom brisanju prikazuje se poruka o uspesnosti/neuspesnosti brisanja



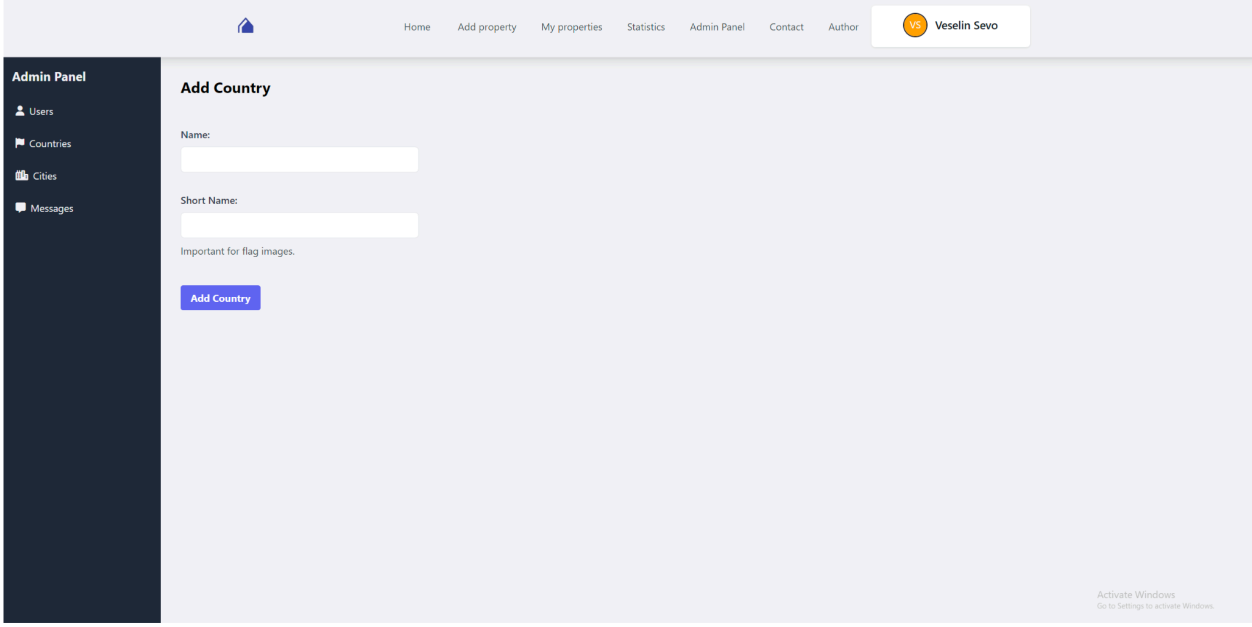
Countries

ID	Name	Flag	
1	Serbia		Edit Delete
2	Croatia		Edit Delete
3	Kuwait		Edit Delete
4	Canada		Edit Delete
5	Denmark		Edit Delete
6	Sweden		Edit Delete

Countries -> Edit Country

- Izmena drzave sa vec popunjenim podacima

- Countries -> Add Country
- Dodavanje nove drzave, sa poljem Short Name koji ce sluziti da odredi koja je zastava za tu drzavu, ne moze se dodati drzava koja vec postoji
- Po izvrsenom brisanju prikazuje se poruka o uspesnosti/neuspesnosti dodavanja



Add Country

Name:

Short Name:

Important for flag images.

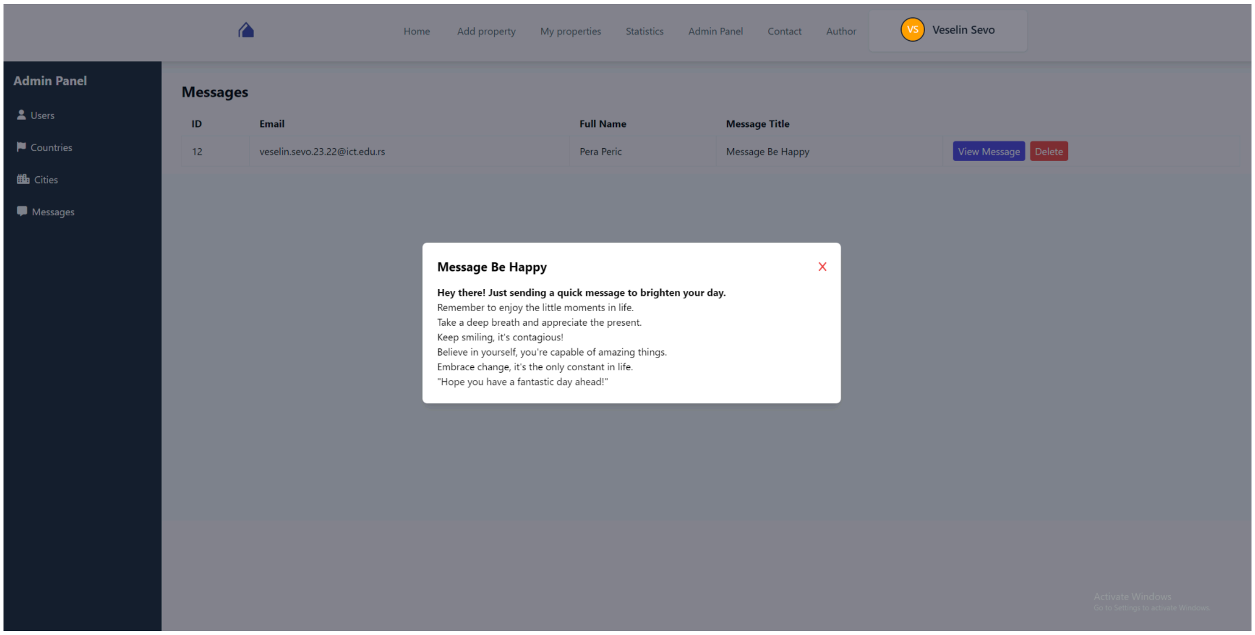
Cities -> Show Cities

- Prikazuju se svi gradovi
- Svaka ima opciju Edit i Delete
- Po izvrsenom brisanju prikazuje se poruka o uspesnosti/neuspesnosti brisanja

- Cities -> Edit City
- Izmena grada sa vec popunjenim podacima
- Dropdown sa dostupnim drzavama

- Cities -> Add City
- Dodavanje novog grada
- Po izvrsenom brisanju prikazuje se poruka o uspesnosti/neuspesnosti dodavanja

- Messages -> Show Messages
- Prikaz svih poruka sa Contact Page-a
- Button View Message koji prikazuje celu poruku koju je neko poslao
- Svaka ima opciju Delete
- Po izvrsenom brisanju prikazuje se poruka o uspesnosti/neuspesnosti brisanja



Messages

ID	Email	Full Name	Message Title	
12	veselin.sevo.23.22@ict.edu.rs	Pera Peric	Message Be Happy	View Message Delete

Message Be Happy

Hey there! Just sending a quick message to brighten your day.
Remember to enjoy the little moments in life.
Take a deep breath and appreciate the present.
Keep smiling, it's contagious!
Believe in yourself, you're capable of amazing things.
Embrace change, it's the only constant in life.
"Hope you have a fantastic day ahead!"

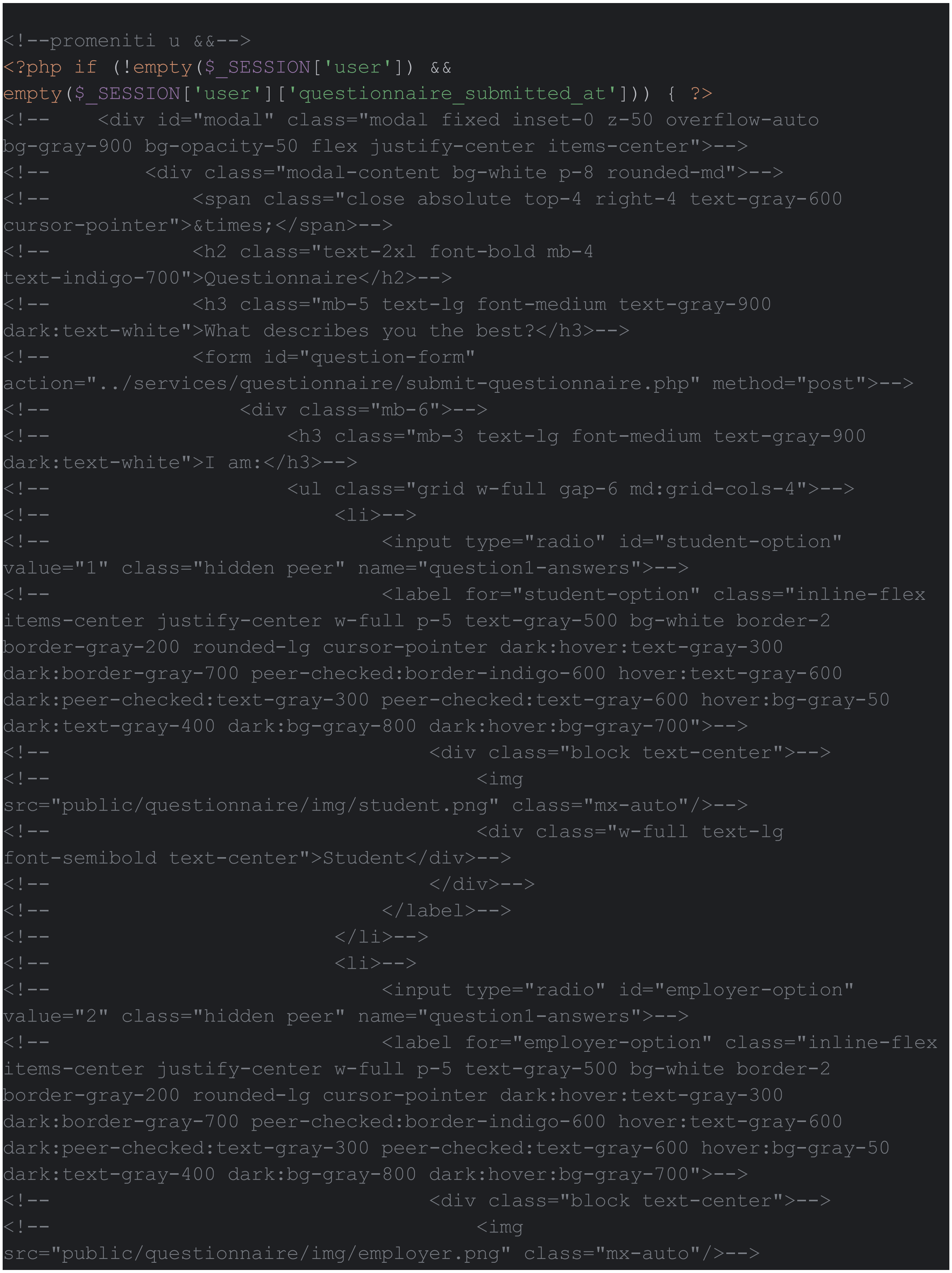
Kodovi

Index.phtml

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

<?php require "public/head.phtml"; ?>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/numeral.js/2.0.6/numeral.min.js"></
script>


<body class="">

<!-- Navbar -->
<?php include "navigation.phtml"; ?>

<!-- Main Content -->
<main class="container mx-auto md:max-w-screen-xl mt-8 px-4 mb-10">
   <div class="flex flex-col md:flex-row justify-between">
       <!-- Toggle Filters Button (Icon) -->
       <div class="flex justify-end">
           <button id="toggleFiltersBtn" class="block md:hidden">
               <i class="fas fa-filter"></i> <!-- Font Awesome icon for filter
-->
           </button>
       </div>
       <!-- Filters -->
       <div class="w-3/4 p-4 mx-auto bg-white rounded-md md:mr-4 hidden
md:block md:w-1/4" id="filtersSection">
           <h2 class="text-xl font-semibold mb-4">Filters</h2>
           <div class="mb-4">
               <form id="searchForm" action="" method="GET">
                   <label for="search" class="block text-sm font-medium
text-gray-700 mb-2">Search Property</label>
                   <input type="text" name="search" id="search"
placeholder="Search properties" value="<?php echo
htmlspecialchars($_GET['search'] ?? '', ENT_QUOTES); ?>" class="w-full h-10
border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500 focus:ring
focus:ring-indigo-200 focus:ring-opacity-50">
                   <button type="submit" class="mt-4 text-sm md:text-sm
bg-indigo-600 text-white py-2 px-4 rounded-md">Search</button>
               </form>
           </div>
           <hr class="mb-4">
           <div class="mb-4">
```

```
                <label for="typeFilter" class="block text-sm font-medium text-gray-700 mb-2">Property Type</label>
                <select id="typeFilter" class="w-full h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500 focus:ring focus:ring-indigo-200 focus:ring-opacity-50">
                    <option value="">All</option>
                    <option value="House">House</option>
                    <option value="Apartment">Apartment</option>
                    <option value="Condo">Condo</option>
                </select>
            </div>
            <div class="mb-4">
                <label for="countryFilter" class="block text-sm font-medium text-gray-700 mb-2">Country</label>
                <select id="countryFilter" class="w-full h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500 focus:ring focus:ring-indigo-200 focus:ring-opacity-50">

                </select>
            </div>
            <div class="mb-4">
                <label for="cityFilter" class="block text-sm font-medium text-gray-700 mb-2">City</label>
                <select id="cityFilter" class="w-full h-10 border-gray-300 p-2 disabled rounded-md shadow-sm focus:border-indigo-500 focus:ring focus:ring-indigo-200 focus:ring-opacity-50">
                    <option value="">All</option>
                    <option value="New York">New York</option>
                    <option value="Los Angeles">Los Angeles</option>
                    <option value="Chicago">Chicago</option>
                    <!-- Add more cities if needed -->
                </select>
            </div>
            <div class="mb-4">
                <label class="block text-sm font-medium text-gray-700 mb-2">Price</label>
                <div class="mb-4 flex gap-5">
                    <div>
                        <input type="number" id="minPrice" min="0" class="w-full h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500 focus:ring focus:ring-indigo-200 focus:ring-opacity-50" placeholder="Min Price">
                    </div>
                    <div>
                        <input type="number" id="maxPrice" min="0" class="w-full h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500 focus:ring focus:ring-indigo-200 focus:ring-opacity-50" placeholder="Max Price">
                    </div>
```

```
               </div>
           </div>
           <div class="mb-4">
               <label class="block text-sm font-medium text-gray-700
mb-2">Size</label>
               <div class="mb-4 flex gap-5">
                   <div>
                       <input type="number" id="minSize" min="0" class="w-full
h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500
focus:ring focus:ring-indigo-200 focus:ring-opacity-50" placeholder="Min Size">
                   </div>
                   <div>
                       <input type="number" id="maxSize" min="0" class="w-full
h-10 border-gray-300 p-2 rounded-md shadow-sm focus:border-indigo-500
focus:ring focus:ring-indigo-200 focus:ring-opacity-50" placeholder="Max Size">
                   </div>
               </div>
           </div>
           <div id="errorMsg" class="hidden text-red-500 text-sm mb-2"></div>
           <button class="text-sm md:text-sm bg-indigo-600 text-white py-2 px-4
rounded-md">Apply Filters</button>
       </div>

       <!-- Property Grid -->
       <div class="w-full relative pt-6 md:pt-12">
           <!-- Icon for displaying 4 properties in a row -->
           <div class="absolute top-4 right-16 flex gap-4 z-0">
               <button id="displayFourBtn" class="hidden md:block
text-indigo-600 hover:text-indigo-600">
                   <i class="fas fa-th"></i> <!-- Font Awesome icon for grid
view -->
               </button>

               <!-- Icon for displaying 3 property in a row -->
               <button id="displayThreeBtn" class="hidden md:block
hover:text-indigo-600">
                   <i class="fas fa-bars"></i> <!-- Font Awesome icon for list
view -->
               </button>
           </div>

           <div class="grid mx-auto grid-cols-1 md:grid-cols-4 lg:grid-cols-4
gap-4 items-start" id="properties-container">

           </div>

       </div>
   </div>
</main>
```

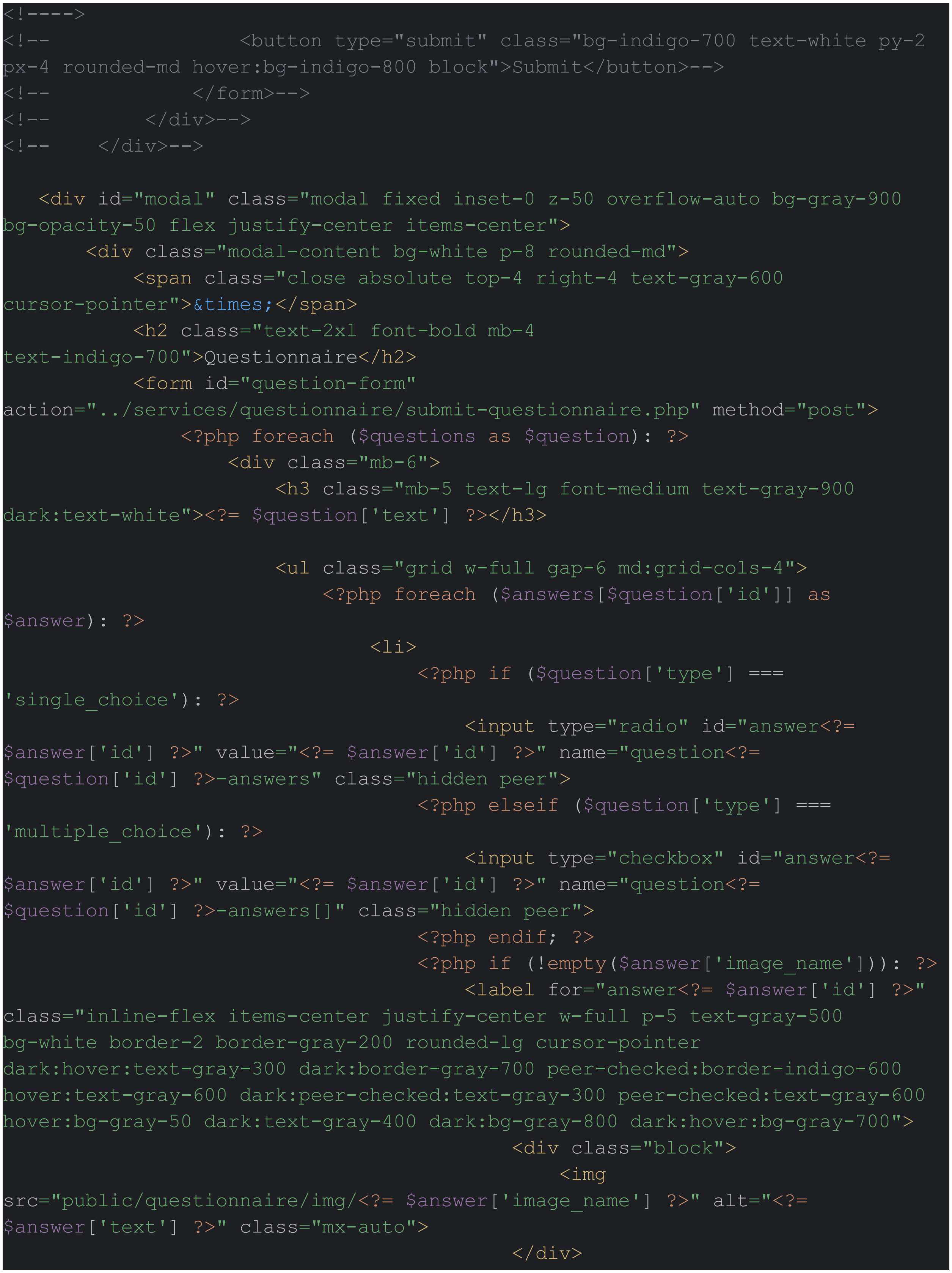


```
<!--promeniti u &&-->
<?php if (!empty($_SESSION['user']) &&
empty($_SESSION['user']['questionnaire_submitted_at'])) { ?>
<!--    <div id="modal" class="modal fixed inset-0 z-50 overflow-auto
bg-gray-900 bg-opacity-50 flex justify-center items-center">-->
<!--        <div class="modal-content bg-white p-8 rounded-md">-->
<!--            <span class="close absolute top-4 right-4 text-gray-600
cursor-pointer">&times;</span>-->
<!--            <h2 class="text-2xl font-bold mb-4
text-indigo-700">Questionnaire</h2>-->
<!--            <h3 class="mb-5 text-lg font-medium text-gray-900
dark:text-white">What describes you the best?</h3>-->
<!--            <form id="question-form"
action="../services/questionnaire/submit-questionnaire.php" method="post">-->
<!--                <div class="mb-6">-->
<!--                    <h3 class="mb-3 text-lg font-medium text-gray-900
dark:text-white">I am:</h3>-->
<!--                    <ul class="grid w-full gap-6 md:grid-cols-4">-->
<!--                        <li>-->
<!--                            <input type="radio" id="student-option"
value="1" class="hidden peer" name="question1-answers">-->
<!--                            <label for="student-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block text-center">-->
<!--                                    <img
src="public/questionnaire/img/student.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Student</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="radio" id="employer-option"
value="2" class="hidden peer" name="question1-answers">-->
<!--                            <label for="employer-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block text-center">-->
<!--                                    <img
src="public/questionnaire/img/employer.png" class="mx-auto"/>-->
```

```
<!--                                    <div class="w-full text-lg
font-semibold text-center">Employer</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="radio" id="employee-option"
value="3" class="hidden peer" name="question1-answers">-->
<!--                            <label for="employee-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block text-center">-->
<!--                                    <img
src="public/questionnaire/img/employee.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Employee</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="radio" id="none-option" value="4"
class="hidden peer" name="question1-answers">-->
<!--                            <label for="none-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block text-center">-->
<!--                                    <img
src="public/questionnaire/img/none.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">None</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                    </ul>-->
<!--                </div>-->
<!---->
<!---->
<!--                <div class="mb-6">-->
<!--                    <h3 class="mb-3 text-lg font-medium text-gray-900
dark:text-white">I am:</h3>-->
<!--                    <ul class="grid w-full gap-6 md:grid-cols-4">-->
<!--                        <li>-->
```

```
<!--                            <input type="radio" id="buyer-option" value="5"
class="hidden peer" name="question2-answers">-->
<!--                            <label for="buyer-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block">-->
<!--                                    <img
src="public/questionnaire/img/buyer.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Buyer</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="radio" id="seller-option"
value="6" class="hidden peer" name="question2-answers">-->
<!--                            <label for="seller-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block">-->
<!--                                    <img
src="public/questionnaire/img/seller.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Seller</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                    </ul>-->
<!--                </div>-->
<!---->
<!---->
<!---->
<!--                <div class="mb-6">-->
<!--                    <h3 class="mb-3 text-lg font-medium text-gray-900
dark:text-white">I am looking for / selling:</h3>-->
<!--                    <ul class="grid w-full gap-6 md:grid-cols-4">-->
<!--                        <li>-->
<!--                            <input type="checkbox" id="apartment-option"
value="7" class="hidden peer" name="question3-answers[]">-->
<!--                            <label for="apartment-option"
class="inline-flex items-center justify-center w-full p-5 text-gray-500
bg-white border-2 border-gray-200 rounded-lg cursor-pointer
dark:hover:text-gray-300 dark:border-gray-700 peer-checked:border-indigo-600
```

```
hover:text-gray-600 dark:peer-checked:text-gray-300 peer-checked:text-gray-600
hover:bg-gray-50 dark:text-gray-400 dark:bg-gray-800
dark:hover:bg-gray-700">-->
<!--                                <div class="block">-->
<!--                                    <img
src="public/questionnaire/img/apartment.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Apartment</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="checkbox" id="house-option"
value="8" class="hidden peer" name="question3-answers[]">-->
<!--                            <label for="house-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block">-->
<!--                                    <img
src="public/questionnaire/img/house.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">House</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                        <li>-->
<!--                            <input type="checkbox" id="condo-option"
value="9" class="hidden peer" name="question3-answers[]">-->
<!--                            <label for="condo-option" class="inline-flex
items-center justify-center w-full p-5 text-gray-500 bg-white border-2
border-gray-200 rounded-lg cursor-pointer dark:hover:text-gray-300
dark:border-gray-700 peer-checked:border-indigo-600 hover:text-gray-600
dark:peer-checked:text-gray-300 peer-checked:text-gray-600 hover:bg-gray-50
dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">-->
<!--                                <div class="block">-->
<!--                                    <img
src="public/questionnaire/img/condo.png" class="mx-auto"/>-->
<!--                                    <div class="w-full text-lg
font-semibold text-center">Condo</div>-->
<!--                                </div>-->
<!--                            </label>-->
<!--                        </li>-->
<!--                    </ul>-->
<!--                </div>-->
<!---->
<!---->
```



```
<!---->
<!--                <button type="submit" class="bg-indigo-700 text-white py-2
px-4 rounded-md hover:bg-indigo-800 block">Submit</button>-->
<!--            </form>-->
<!--        </div>-->
<!--    </div>-->

   <div id="modal" class="modal fixed inset-0 z-50 overflow-auto bg-gray-900
bg-opacity-50 flex justify-center items-center">
       <div class="modal-content bg-white p-8 rounded-md">
           <span class="close absolute top-4 right-4 text-gray-600
cursor-pointer">&times;</span>
           <h2 class="text-2xl font-bold mb-4
text-indigo-700">Questionnaire</h2>
           <form id="question-form"
action="../services/questionnaire/submit-questionnaire.php" method="post">
               <?php foreach ($questions as $question): ?>
                   <div class="mb-6">
                       <h3 class="mb-5 text-lg font-medium text-gray-900
dark:text-white"><?= $question['text'] ?></h3>

                       <ul class="grid w-full gap-6 md:grid-cols-4">
                           <?php foreach ($answers[$question['id']] as
$answer): ?>
                               <li>
                                   <?php if ($question['type'] ===
'single_choice'): ?>
                                       <input type="radio" id="answer<?=
$answer['id'] ?>" value="<?= $answer['id'] ?>" name="question<?=
$question['id'] ?>-answers" class="hidden peer">
                                   <?php elseif ($question['type'] ===
'multiple_choice'): ?>
                                       <input type="checkbox" id="answer<?=
$answer['id'] ?>" value="<?= $answer['id'] ?>" name="question<?=
$question['id'] ?>-answers[]" class="hidden peer">
                                   <?php endif; ?>
                                   <?php if (!empty($answer['image_name'])): ?>
                                       <label for="answer<?= $answer['id'] ?>"
class="inline-flex items-center justify-center w-full p-5 text-gray-500
bg-white border-2 border-gray-200 rounded-lg cursor-pointer
dark:hover:text-gray-300 dark:border-gray-700 peer-checked:border-indigo-600
hover:text-gray-600 dark:peer-checked:text-gray-300 peer-checked:text-gray-600
hover:bg-gray-50 dark:text-gray-400 dark:bg-gray-800 dark:hover:bg-gray-700">
                                           <div class="block">
                                               <img
src="public/questionnaire/img/<?= $answer['image_name'] ?>" alt="<?=
$answer['text'] ?>" class="mx-auto">
                                           </div>
```

```
                                            <div class="w-full text-lg
font-semibold text-center"><?= $answer['text'] ?></div>
                                        </label>
                                    <?php endif; ?>
                                    <br>
                                </li>
                            <?php endforeach; ?>
                        </ul>
                    </div>
                <?php endforeach; ?>
                <button type="submit" class="bg-indigo-700 text-white py-2 px-4
rounded-md hover:bg-indigo-800 block">Submit</button>

            </form>
        </div>
    </div>
<?php } ?>


<script>
    // Display the modal
    var modal = document.getElementById('modal');
    function showModal() {
        modal.style.display = 'flex';
    }

    // Close the modal
    function closeModal() {
        modal.style.display = 'none';
    }

    // When the user clicks anywhere outside of the modal, close it
    window.onclick = function(e) {
        e.stopPropagation()
        console.log(e.target)
        if (e.target == modal) {
            closeModal()
        }
    }

    if(modal) {
        showModal()
    }

    let selectedCurrency = '<?php echo $_SESSION['user-settings']['currency'];
?>';
    let selectedCardsPerRow = '<?php echo
$_SESSION['user-settings']['cards_per_row']; ?>';
</script>
```

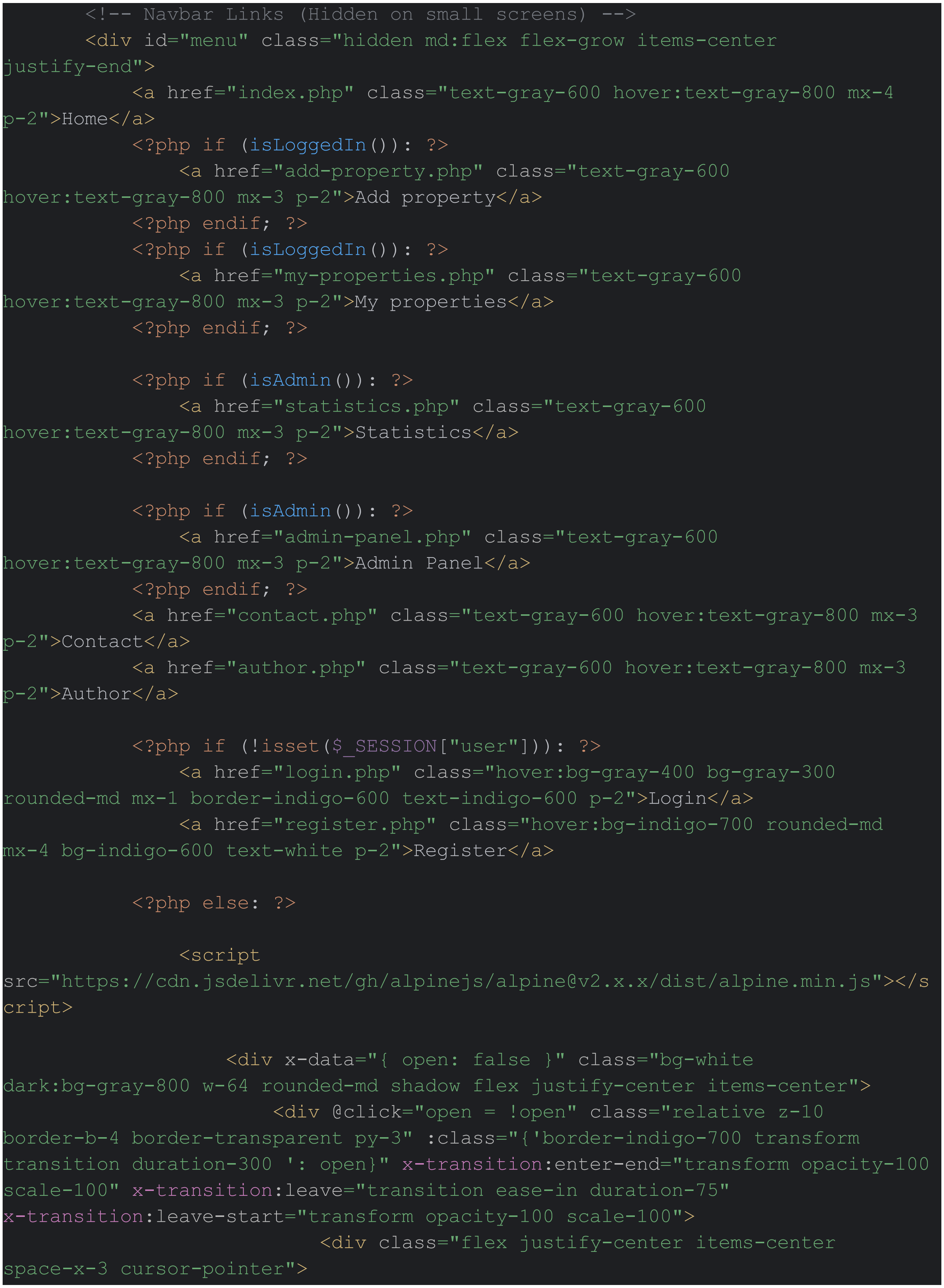
```
<script type="module" src="public/js/index.js"></script>
</body>

</html>
```

Index.php

```
<?php
session_start();

error_reporting(error_reporting() & ~E_NOTICE);

require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";


if (!empty($_SESSION['user']) &&
empty($_SESSION['user']['questionnaire_submitted_at'])) {
   require 'services/questionnaire/get-questionnaire.php';
}

require 'public/index.phtml';
```

Navigation.phtml

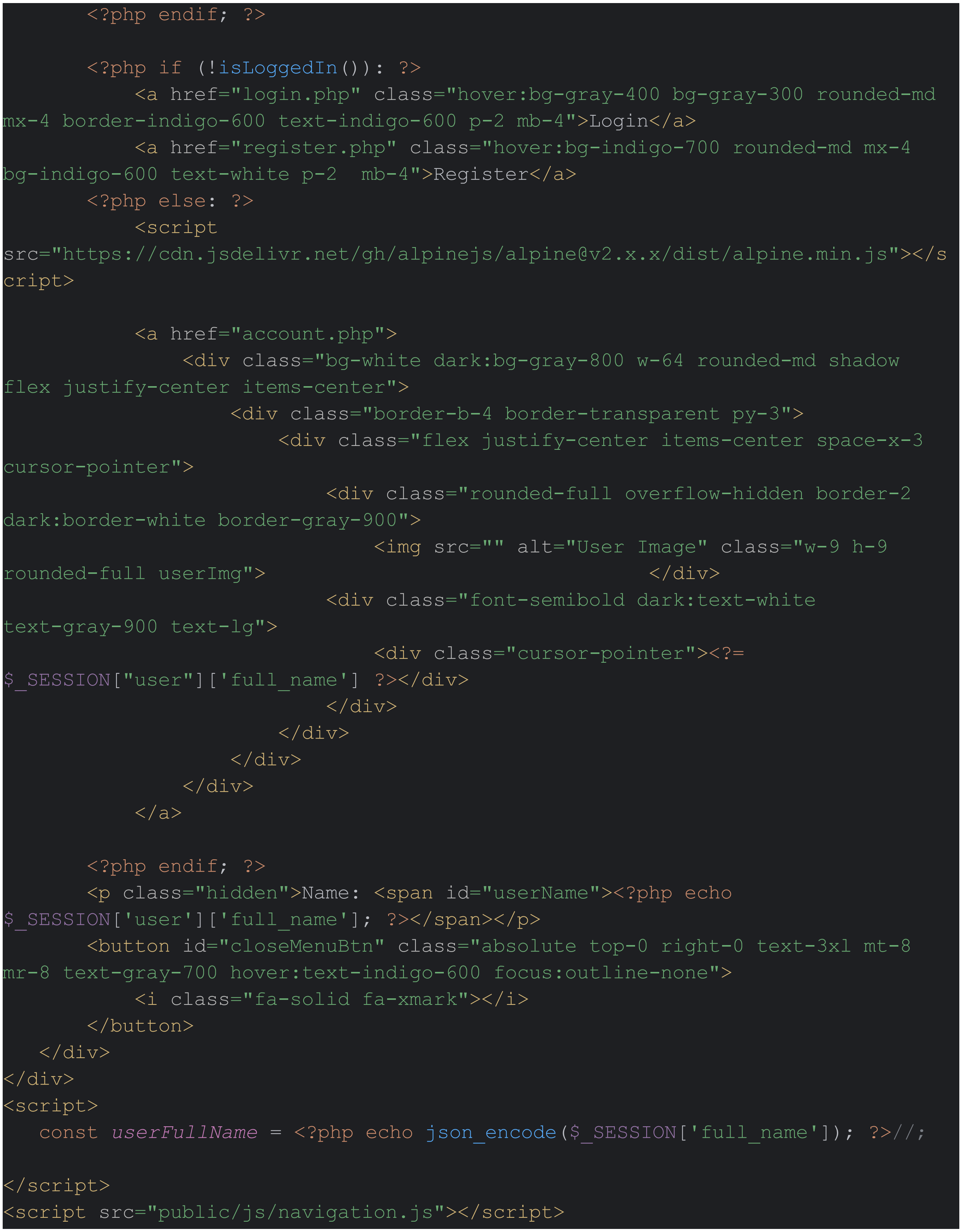
```
<nav class="bg-gray-100 p-4 shadow-lg">
   <div class="container md:max-w-screen-xl mx-auto flex justify-between
items-center">
       <!-- Logo -->
       <a href="index.php" class="text-xl font-semibold text-gray-800"><img
class="w-8 h-auto" src="public/img/realestate-logo.png"></a>

       <?php if (isset($_SESSION["fullname"])): ?>
           <!-- Display the "Hello [name]!" message -->
           <p class="md:ml-3">Hello <span class="font-medium
text-indigo-600"><?php echo getFlashMessage("fullname") ?></span>!</p>
       <?php endif; ?>
       <!-- Responsive Navbar Toggler -->
       <button id="toggleMenuBtn" class="block md:hidden focus:outline-none">
           <svg class="h-6 w-6" fill="none" stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" viewBox="0 0 24 24"
stroke="currentColor">
               <path d="M4 6h16M4 12h16M4 18h16"></path>
           </svg>
       </button>

```



```
       <!-- Navbar Links (Hidden on small screens) -->
       <div id="menu" class="hidden md:flex flex-grow items-center
justify-end">
           <a href="index.php" class="text-gray-600 hover:text-gray-800 mx-4
p-2">Home</a>
           <?php if (isLoggedIn()): ?>
               <a href="add-property.php" class="text-gray-600
hover:text-gray-800 mx-3 p-2">Add property</a>
           <?php endif; ?>
           <?php if (isLoggedIn()): ?>
               <a href="my-properties.php" class="text-gray-600
hover:text-gray-800 mx-3 p-2">My properties</a>
           <?php endif; ?>

           <?php if (isAdmin()): ?>
               <a href="statistics.php" class="text-gray-600
hover:text-gray-800 mx-3 p-2">Statistics</a>
           <?php endif; ?>

           <?php if (isAdmin()): ?>
               <a href="admin-panel.php" class="text-gray-600
hover:text-gray-800 mx-3 p-2">Admin Panel</a>
           <?php endif; ?>
           <a href="contact.php" class="text-gray-600 hover:text-gray-800 mx-3
p-2">Contact</a>
           <a href="author.php" class="text-gray-600 hover:text-gray-800 mx-3
p-2">Author</a>

           <?php if (!isset($_SESSION["user"])): ?>
               <a href="login.php" class="hover:bg-gray-400 bg-gray-300
rounded-md mx-1 border-indigo-600 text-indigo-600 p-2">Login</a>
               <a href="register.php" class="hover:bg-indigo-700 rounded-md
mx-4 bg-indigo-600 text-white p-2">Register</a>

           <?php else: ?>

               <script
src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.x.x/dist/alpine.min.js"></s
cript>

                   <div x-data="{ open: false }" class="bg-white
dark:bg-gray-800 w-64 rounded-md shadow flex justify-center items-center">
                       <div @click="open = !open" class="relative z-10
border-b-4 border-transparent py-3" :class="{'border-indigo-700 transform
transition duration-300 ': open}" x-transition:enter-end="transform opacity-100
scale-100" x-transition:leave="transition ease-in duration-75"
x-transition:leave-start="transform opacity-100 scale-100">
                           <div class="flex justify-center items-center
space-x-3 cursor-pointer">
```

```
                               <div class="rounded-full overflow-hidden
border-2 dark:border-white border-gray-900">
                                   <img src="" alt="User Image" class="w-9 h-9
rounded-full userImg">                                    </div>
                               <div class="font-semibold dark:text-white
text-gray-900 text-lg">
                                   <div class="cursor-pointer"><?=
$_SESSION["user"]['full_name'] ?></div>
                               </div>
                           </div>
                           <div x-show="open" class="absolute w-52 px-5 py-3
dark:bg-gray-800 bg-white rounded-lg shadow border dark:border-transparent mt-5
z-10">
                               <ul class="space-y-3 dark:text-white">
                                   <li class="font-medium">
                                       <a href="account.php" class="flex
items-center transform transition-colors duration-200 border-r-4
border-transparent hover:border-indigo-700">
                                           <div class="mr-3">
                                               <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"
xmlns="http://www.w3.org/2000/svg"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12
14a7 7 0 00-7 7h14a7 7 0 00-7-7z"></path></svg>
                                           </div>
                                           Account
                                       </a>
                                   </li>
                                   <li class="font-medium">
                                       <a href="account.php" class="flex
items-center transform transition-colors duration-200 border-r-4
border-transparent hover:border-indigo-700">
                                           <div class="mr-3">
                                               <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"
xmlns="http://www.w3.org/2000/svg"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M10.325 4.317c.426-1.756
2.924-1.756 3.35 0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37
2.37a1.724 1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724 1.724 0
00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0 00-2.572 1.065c-.426
1.756-2.924 1.756-3.35 0a1.724 1.724 0
00-2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0
00-1.065-2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0
001.066-2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07
2.572-1.065z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"></path></svg>
                                           </div>
                                           Setting
                                       </a>
```

```
                                    </li>
                                    <hr class="dark:border-gray-700">
                                    <li class="font-medium">
                                        <a href="services/user/logout.php"
class="flex items-center transform transition-colors duration-200 border-r-4
border-transparent hover:border-red-600">
                                            <div class="mr-3 text-red-600">
                                                <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"
xmlns="http://www.w3.org/2000/svg"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3
0 01-3 3H6a3 3 0 01-3-3V7a3 3 0 013-3h4a3 3 0 013 3v1"></path></svg>
                                            </div>
                                            Logout
                                        </a>
                                    </li>
                                </ul>
                            </div>
                        </div>
                    </div>
            <?php endif; ?>
        </div>
    </div>
</nav>

<!-- Mobile Menu -->
<div id="mobileMenu" class="md:hidden fixed inset-0 bg-gray-100 z-50 hidden">
    <div class="flex flex-col h-full justify-center items-center over">
        <a href="index.php" class="text-lg text-gray-600 hover:text-gray-800
mb-4">Home</a>
        <a href="contact.php" class="text-lg text-gray-600 hover:text-gray-800
mb-4 p-2">Contact</a>
        <a href="author.php" class="text-lg text-gray-600 hover:text-gray-800
mb-4 p-2">Author</a>
        <?php if (isLoggedIn()): ?>
            <a href="add-property.php" class="text-lg text-gray-600
hover:text-gray-800 mb-4 p-2">Add property</a>
        <?php endif; ?>
        <?php if (isLoggedIn()): ?>
            <a href="my-properties.php" class="text-lg text-gray-600
hover:text-gray-800 mb-4 p-2">My properties</a>
        <?php endif; ?>
        <?php if (isAdmin()): ?>
            <a href="statistics.php" class="text-lg text-gray-600
hover:text-gray-800 mb-4 p-2">Statistics</a>
        <?php endif; ?>
        <?php if (isAdmin()): ?>
            <a href="admin-panel.php" class="text-lg text-gray-600
hover:text-gray-800 mb-4 p-2">Admin Panel</a>
```



```
       <?php endif; ?>

       <?php if (!isLoggedIn()): ?>
           <a href="login.php" class="hover:bg-gray-400 bg-gray-300 rounded-md
mx-4 border-indigo-600 text-indigo-600 p-2 mb-4">Login</a>
           <a href="register.php" class="hover:bg-indigo-700 rounded-md mx-4
bg-indigo-600 text-white p-2  mb-4">Register</a>
       <?php else: ?>
           <script
src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.x.x/dist/alpine.min.js"></s
cript>

           <a href="account.php">
               <div class="bg-white dark:bg-gray-800 w-64 rounded-md shadow
flex justify-center items-center">
                   <div class="border-b-4 border-transparent py-3">
                       <div class="flex justify-center items-center space-x-3
cursor-pointer">
                           <div class="rounded-full overflow-hidden border-2
dark:border-white border-gray-900">
                               <img src="" alt="User Image" class="w-9 h-9
rounded-full userImg">                                </div>
                           <div class="font-semibold dark:text-white
text-gray-900 text-lg">
                               <div class="cursor-pointer"><?=
$_SESSION["user"]['full_name'] ?></div>
                           </div>
                       </div>
                   </div>
               </div>
           </a>

       <?php endif; ?>
       <p class="hidden">Name: <span id="userName"><?php echo
$_SESSION['user']['full_name']; ?></span></p>
       <button id="closeMenuBtn" class="absolute top-0 right-0 text-3xl mt-8
mr-8 text-gray-700 hover:text-indigo-600 focus:outline-none">
           <i class="fa-solid fa-xmark"></i>
       </button>
   </div>
</div>
<script>
   const userFullName = <?php echo json_encode($_SESSION['full_name']); ?>//;

</script>
<script src="public/js/navigation.js"></script>
```

Author.phtml

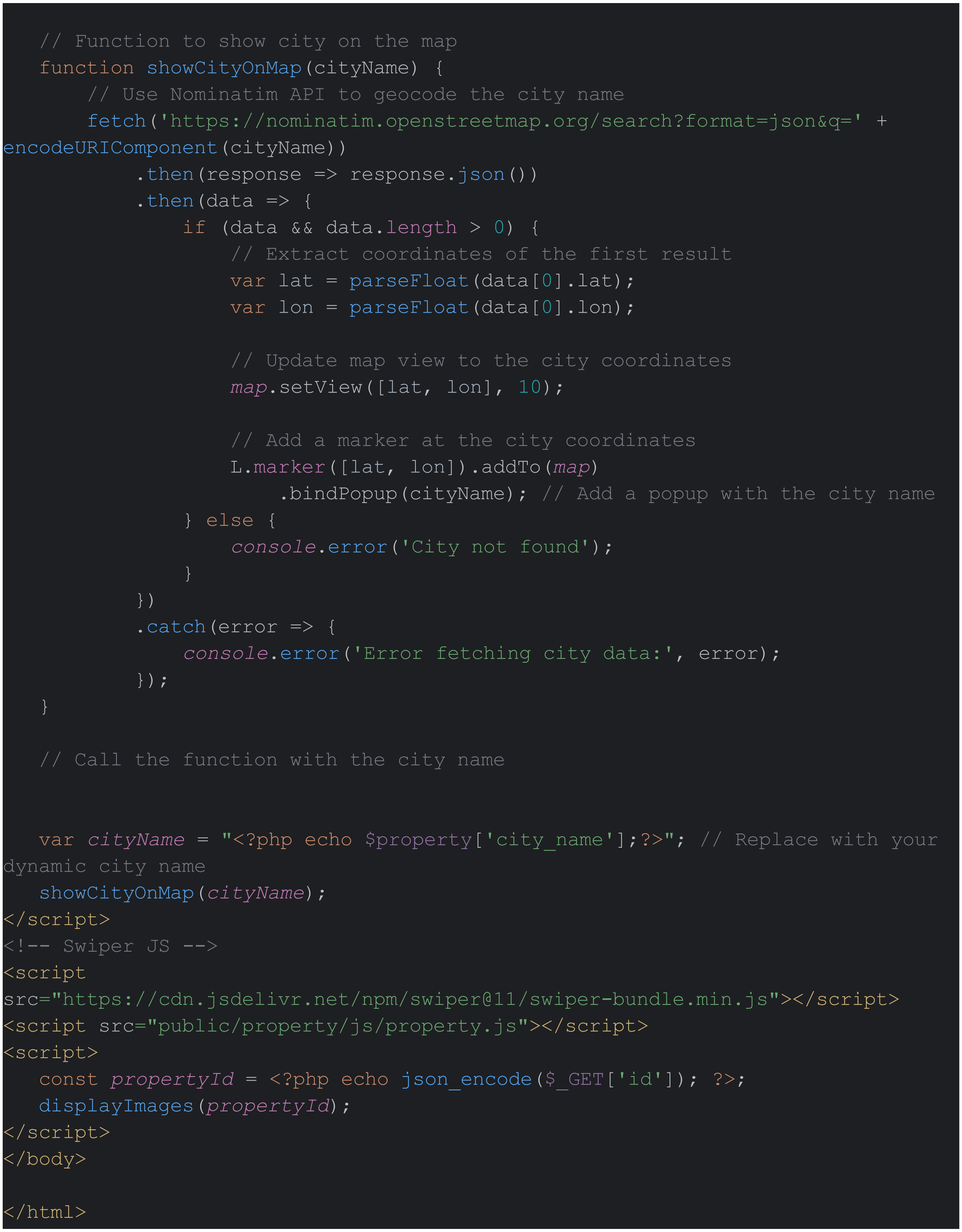
```
<!DOCTYPE html>
<html lang="en">

<?php require "public/head.phtml"; ?>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/numeral.js/2.0.6/numeral.min.js"></
script>


<body class="">

<!-- Navbar -->
<?php include "navigation.phtml"; ?>

<!-- Main Content -->
<main class="container mx-auto md:max-w-screen-xl mt-8 px-4">
   <div class=" m-auto">
       <h2 class="text-md md:text-lg font-bold mb-3">Veselin Sevo 23/22</h2>
       <img class="w-64" src="../../public/img/my-image.jpg" alt="Author
image">
   </div>

</main>
</body>

</html>
```

Property.phtml

```
<?php include "util/format-price.php"?>

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Property Page</title>
   <link
href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css"
rel="stylesheet">
   <script src="https://kit.fontawesome.com/a279dcf9f7.js"
crossorigin="anonymous"></script>

   <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
integrity="sha256-p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY=" crossorigin=""
/>
```

```
   <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo="
crossorigin=""></script>

   <!-- Swiper CSS -->
   <link
           rel="stylesheet"
           href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.css"
   />


   <link rel="apple-touch-icon" sizes="76x76" href="/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
   <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
   <link rel="manifest" href="/site.webmanifest">
   <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5">
   <meta name="msapplication-TileColor" content="#da532c">
   <meta name="theme-color" content="#ffffff">
</head>

<body class="bg-gray-100 dark:bg-amber-800">
<!-- Navbar -->
<?php include "public/navigation.phtml"; ?>

<main class="container mx-auto md:max-w-screen-xl mt-8 px-4">
   <div class="flex flex-col md:flex-row items-center md:justify-between">
       <!-- Property Details -->
       <div class="w-full md:w-1/2 py-2 md:py-8">
           <div class="max-w-4xl mx-auto">
               <!-- Structure Information -->
               <div class="p-4">

                   <div class="flex items-center space-x-5">
                       <!-- Bedroom Card -->
                       <div class="flex items-center">
                           <i class="fas fa-bed text-indigo-600 text-lg"></i>
                           <span class="text-sm lg:text-base ml-3
md:ml-2"><span class="hidden md:inline">Bedrooms:</span> <?php echo
ucfirst($property['bedrooms']); ?></span>
                       </div>
                       <!-- Bathroom Card -->
                       <div class="flex items-center">
                           <i class="fas fa-bath text-indigo-600 text-lg"></i>
                           <span class="text-sm lg:text-base ml-3
md:ml-2"><span class="hidden md:inline">Bathrooms:</span> <?php echo
ucfirst($property['bathrooms']); ?></span>
                       </div>
                       <!-- Balcony Card -->
                       <div class="flex items-center">
```

```
                           <i class="fas fa-door-open text-indigo-600
text-lg"></i>
                           <span class="text-sm lg:text-base ml-3
md:ml-2"><span class="hidden md:inline">Balcony:</span> <?php echo
ucfirst($property['balconies']); ?></span>
                       </div>
                       <!-- Add more structure details as needed -->
                   </div>
               </div>

               <!-- Property Category and Title -->
               <div class="p-4 border-t border-gray-200">
                   <h3 class="text-lg text-indigo-600 font-bold mb-2"><?php
echo ucfirst($property['type_name']); ?></h3>
                   <h1 class="text-3xl md:text-4xl font-bold mb-4"><?php echo
$property['name']; ?></h1>

                   <!-- Description, Price, and Location -->
                   <p class="text-gray-700 mb-2"><?php echo
$property['full_name']; ?></p>
                   <p class="text-gray-700 mb-2"><?php echo
$property['description']; ?></p>
                   <p class="text-xl text-indigo-600 font-semibold mb-4"><?php
echo formatPrice($property['price'], $_SESSION['user-settings']['currency']);
?></p>
                   <p class="text-gray-600">
                       <div class="flex gap-3">
                           <i class="fas fa-map-marker-alt mt-1"></i>
                           <?php echo $property['city_name'];?>,
                           <?php echo $property['country_name']; ?>
                           <img class="w-12"
src="https://flagcdn.com/w320/<?php echo $property['short_name']; ?>.png"
alt="<?php echo $property['name']; ?>"/>
                       </div>
                   </p>








               </div>
           </div>
       </div>

       <img class="hidden md:block rounded-md max-w-screen-sm"
src="store/property/main-images/<?php echo $property['image_name']; ?>" />
   </div>
   <div class="w-full md:w-full mx-auto py-2 md:py-16 relative
overflow-hidden">
       <div class="w-full mx-auto">
           <div class="swiper-container w-full">
               <div class="swiper-wrapper">

               <!-- Navigation arrows -->
<!--                <div class="swiper-button-prev text-indigo-600"></div>-->
<!--                <div class="swiper-button-next text-indigo-600"></div>-->
           </div>
       </div>
   </div>

       <div class="flex flex-col md:flex-row mt-10 gap-6">
           <div id="map" class="w-full !md:w-96 h-80 rounded-md z-0"></div>
           <div>
               <p class="text-gray-600">
               <div class="flex flex-col gap-3 text-gray-700">
                   <p class="font-bold">Address: </p>
                   <p class=" mb-2"><?php echo $property['city_name'];?>, <?php
echo $property['country_name']; ?>, <?php echo $property['address']; ?></p>
                   <p class="font-bold">Size: </p>
                   <!--                    Description to be implemented-->
                   <p class=" mb-2"><?php echo intval($property['size']);?>
m<sup>2</sup></p>
                   <p class="font-bold">Description: </p>
<!--                    Description to be implemented-->
                   <p class=" mb-2"><?php echo
$property['full_description'];?></p>
               </div>
               </p>
           </div>
       </div>



</main>

<script>
   // Initialize Leaflet map
   var map = L.map('map').setView([0, 0], 2); // Initial view centered on the
world

   // Add the base OpenStreetMap layer
   L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
       // attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
   }).addTo(map);
```



```
    // Function to show city on the map
    function showCityOnMap(cityName) {
        // Use Nominatim API to geocode the city name
        fetch('https://nominatim.openstreetmap.org/search?format=json&q=' +
encodeURIComponent(cityName))
            .then(response => response.json())
            .then(data => {
                if (data && data.length > 0) {
                    // Extract coordinates of the first result
                    var lat = parseFloat(data[0].lat);
                    var lon = parseFloat(data[0].lon);

                    // Update map view to the city coordinates
                    map.setView([lat, lon], 10);

                    // Add a marker at the city coordinates
                    L.marker([lat, lon]).addTo(map)
                        .bindPopup(cityName); // Add a popup with the city name
                } else {
                    console.error('City not found');
                }
            })
            .catch(error => {
                console.error('Error fetching city data:', error);
            });
    }

    // Call the function with the city name


    var cityName = "<?php echo $property['city_name'];?>"; // Replace with your
dynamic city name
    showCityOnMap(cityName);
</script>
<!-- Swiper JS -->
<script
src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.js"></script>
<script src="public/property/js/property.js"></script>
<script>
    const propertyId = <?php echo json_encode($_GET['id']); ?>;
    displayImages(propertyId);
</script>
</body>

</html>
```

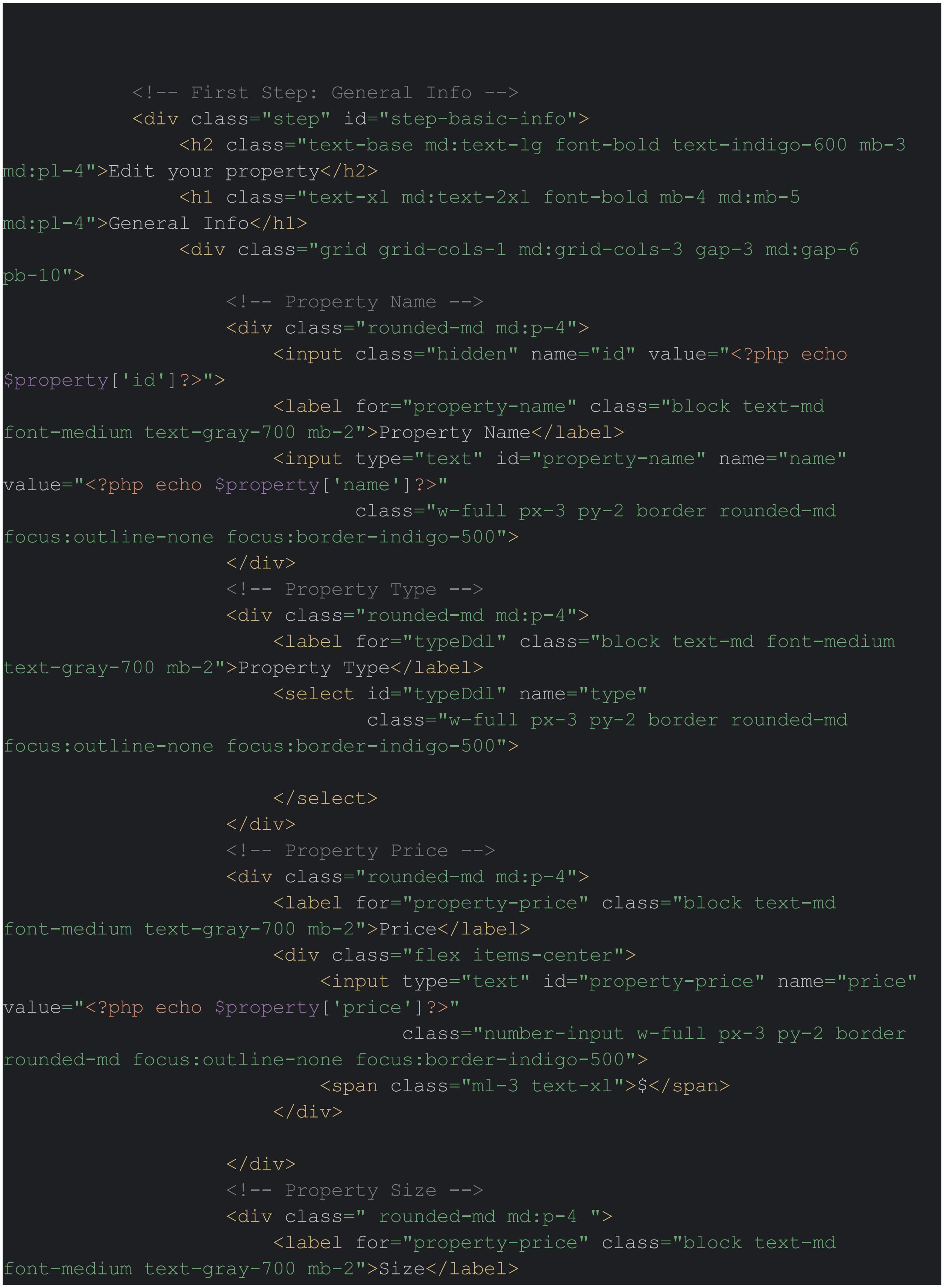
My-properties.phtml

```
<!DOCTYPE html>
<html lang="en">
<?php require "public/head.phtml"; ?>
<script src="https://cdnjs.cloudflare.com/ajax/libs/numeral.js/2.0.6/numeral.min.js"></script>

<body class="bg-gray-100">

<!-- Navbar -->
<?php require "public/navigation.phtml"; ?>

<main class="container mx-auto md:max-w-screen-xl mt-8 px-4">
   <div class="flex flex-col md:flex-row justify-between">
       <!-- Property Grid -->
       <div class="w-full relative pt-6 md:pt-12">
           <!-- Icon for displaying 4 properties in a row -->
           <button id="displayFiveBtn" class="absolute top-4 right-16 hidden md:block text-indigo-600 hover:text-indigo-600">
               <i class="fas fa-th"></i> <!-- Font Awesome icon for grid view -->
           </button>

           <!-- Icon for displaying 3 property in a row -->
           <button id="displayFourBtn" class="absolute top-4 right-4 hidden md:block hover:text-indigo-600">
               <i class="fas fa-bars"></i> <!-- Font Awesome icon for list view -->
           </button>
           <div class="grid mx-auto grid-cols-1 md:grid-cols-5 lg:grid-cols-5 gap-4 items-start" id="properties-container">

           </div>
           <div id="no-properties-message" class="flex flex-col align-middle gap-3">
               <h2 class="text-center md:text-left text-lg md:text-2xl font-bold mb-3 md:pl-4">You don't have any properties, yet.</h2>
               <a href="add-property.php" class="text-sm text-center md:text-base w-36 m-2 md:m-4 px-4 py-2 rounded-md bg-indigo-600 leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600">Add property</a>
           </div>

       </div>
   </div>
</main>

<script>
```

```
   let selectedCurrency = '<?php echo $_SESSION['user-settings']['currency'];
?>';
   let selectedCardsPerRow = <?php echo
$_SESSION['user-settings']['cards_per_row']; ?>;
   let userId = <?php echo $_SESSION['user']['id']; ?>;
</script>
<script type="module" src="public/property/js/my-properties.js"></script>
<script src="https://cdn.tailwindcss.com/"></script>
</body>
</html>
```

Edit-property.phtml

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Property Page</title>
   <link
href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css"
rel="stylesheet">
   <link href="https://cdn.jsdelivr.net/npm/quill@2.0.2/dist/quill.snow.css"
rel="stylesheet" />
   <script src="https://kit.fontawesome.com/a279dcf9f7.js"
crossorigin="anonymous"></script>

   <link rel="apple-touch-icon" sizes="76x76" href="/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
   <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
   <link rel="manifest" href="/site.webmanifest">
   <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5">
   <meta name="msapplication-TileColor" content="#da532c">
   <meta name="theme-color" content="#ffffff">
</head>
<body class="bg-gray-100">

<!-- Navbar -->
<?php require "public/navigation.phtml"; ?>

<main class="container mx-auto md:max-w-screen-xl mt-8 md:mt-24 px-4 mb-8
md:mb-24">
   <form id="property-form" action="services/property/edit-property.php"
method="POST" enctype="multipart/form-data">
       <input name="property_id" class="hidden" value="<?php echo
$property['id']?>"/>
       <div class="flex flex-col bg-white rounded-md shadow-md align-baseline
p-6 md:p-6">
```



```
           <!-- First Step: General Info -->
           <div class="step" id="step-basic-info">
               <h2 class="text-base md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Edit your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4 md:mb-5
md:pl-4">General Info</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Property Name -->
                   <div class="rounded-md md:p-4">
                       <input class="hidden" name="id" value="<?php echo
$property['id']?>">
                       <label for="property-name" class="block text-md
font-medium text-gray-700 mb-2">Property Name</label>
                       <input type="text" id="property-name" name="name"
value="<?php echo $property['name']?>"
                              class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                   </div>
                   <!-- Property Type -->
                   <div class="rounded-md md:p-4">
                       <label for="typeDdl" class="block text-md font-medium
text-gray-700 mb-2">Property Type</label>
                       <select id="typeDdl" name="type"
                               class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">

                       </select>
                   </div>
                   <!-- Property Price -->
                   <div class="rounded-md md:p-4">
                       <label for="property-price" class="block text-md
font-medium text-gray-700 mb-2">Price</label>
                       <div class="flex items-center">
                           <input type="text" id="property-price" name="price"
value="<?php echo $property['price']?>"
                                  class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                           <span class="ml-3 text-xl">$</span>
                       </div>

                   </div>
                   <!-- Property Size -->
                   <div class=" rounded-md md:p-4 ">
                       <label for="property-price" class="block text-md
font-medium text-gray-700 mb-2">Size</label>
```

```
                       <div class="flex items-center">
                           <input type="text" id="property-price" name="size"
value="<?php echo $property['size']?>"
                                  class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                           <span class="ml-3 text-xl">m<sup>2</span>
                       </div>
                   </div>
               </div>
           </div>

           <!-- Second Step: Location -->
           <div class="step hidden" id="step-location">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Edit your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Location</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Location fields (Country, City, Address) -->
                   <div class="rounded-md md:p-4 md:col-span-2">
                       <label for="countryDdl" class="block text-md font-medium
text-gray-700 mb-2">Country</label>
                       <select id="countryDdl" name="country" class="w-full
px-3 py-2 border rounded-md
                       focus:outline-none focus:border-indigo-500"">
                           <option value="USA">USA</option>
                           <option value="Canada">Canada</option>
                           <option value="UK">UK</option>
                           <!-- Add more options as needed -->
                       </select>
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="cityDdl" class="block text-md font-medium
text-gray-700 mb-2">City</label>
                       <select id="cityDdl" name="city_id" class="w-full px-3
py-2 border rounded-md
                       focus:outline-none focus:border-indigo-500">
                           <!-- Add more options as needed -->
                       </select>
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="address" class="block text-md font-medium
text-gray-700 mb-2">Address</label>
                       <input type="text" id="address" name="address"
value="<?php echo $property['address']?>"
                              class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                   </div>
```

```
                    <!-- Zip Code -->
                    <div class="rounded-md md:p-4">
                        <label for="zip" class="block text-md font-medium
text-gray-700 mb-2">Zip Code</label>
                        <input type="text" id="zip" name="zip" value="<?php echo
$property['zip_code']?>"
                               class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                    </div>
                </div>
            </div>

            <!-- Third Step: Images -->
            <div class="step hidden" id="stepImages">
                <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
                <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Images</h1>
                <div class="grid grid-cols-1 md:grid-cols-1 gap-3 md:gap-6
pb-10">


                    <div class="rounded-md md:p-4">
                        <p class="block text-md font-medium text-gray-700
mb-2">Choose main image</p>
                        <div id="main-image-drop-container"
class="drop-container mt-5 flex flex-col px-3 py-2 bg-transparent
text-indigo-700 rounded-md transition duration-300 border-2 border-dashed
border-indigo-700 w-full h-72 items-center justify-center" draggable="true">

                            <svg xmlns="http://www.w3.org/2000/svg" width="69"
height="51" viewBox="0 0 69 51" class="dropzone-content__image tw-mx-auto
tw-mb-2"><defs><linearGradient id="icon-drop-folder_svg__a" x1="34.5" x2="34.5"
y1="48.67" y2="0.19" gradientUnits="userSpaceOnUse"><stop offset="0.67"
stop-color="#112caf"></stop><stop offset="0.77"
stop-color="#2250f4"></stop></linearGradient></defs><path
fill="url(#icon-drop-folder_svg__a)" d="M62.93 6.8H37.11L29.84.19H6.07a3.53
3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0 0 3.53 3.53h56.86a3.53 3.53 0 0 0
3.53-3.53V10.33a3.53 3.53 0 0 0-3.53-3.53"></path><path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path><path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path><path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                                <path fill="url(#icon-drop-folder_svg__a)"
d="M62.93 6.8H37.11L29.84.19H6.07a3.53 3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0
0 3.53 3.53h56.86a3.53 3.53 0 0 0 3.53-3.53V10.33a3.53 3.53 0 0
0-3.53-3.53"></path>
                                <path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path>
```

```
                                <path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path>
                                <path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                            </svg>
                            <input type="file" id="main-image-input"
name="main-image" class="hidden file-input" accept=".jpg, .jpeg, .png">


                            <p class="block mt-6 text-md text-gray-700
mb-2">Drag and drop you image file here</p>
                            <p class="block mt-5 text-md text-gray-700 mb-2">Or,
                                <label for="main-image-input" data-file-name=""
class="file-input cursor-pointer font-medium text-sm md:text-base
underline">browse to upload.</label>
                            </p>


                        </div>
                        <div id="main-image-preview" class="w-full py-2">
                            No files currently selected for upload
                        </div>
                    </div>


                    <div class="rounded-md md:p-4">
                        <p class="block text-md font-medium text-gray-700
mb-2">Choose additional images</p>
                        <div id="additional-images-drop-container"
class="drop-container mt-5 flex flex-col px-3 py-2 bg-transparent
text-indigo-700 rounded-md transition duration-300 border-2 border-dashed
border-indigo-700 w-full h-72 items-center justify-center">
                            <svg xmlns="http://www.w3.org/2000/svg" width="69"
height="51" viewBox="0 0 69 51" class="dropzone-content__image tw-mx-auto
tw-mb-2"><defs><linearGradient id="icon-drop-folder_svg__a" x1="34.5" x2="34.5"
y1="48.67" y2="0.19" gradientUnits="userSpaceOnUse"><stop offset="0.67"
stop-color="#112caf"></stop><stop offset="0.77"
stop-color="#2250f4"></stop></linearGradient></defs><path
fill="url(#icon-drop-folder_svg__a)" d="M62.93 6.8H37.11L29.84.19H6.07a3.53
3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0 0 3.53 3.53h56.86a3.53 3.53 0 0 0
3.53-3.53V10.33a3.53 3.53 0 0 0-3.53-3.53"></path><path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path><path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path><path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                                <path fill="url(#icon-drop-folder_svg__a)"
d="M62.93 6.8H37.11L29.84.19H6.07a3.53 3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0
```

```
0 3.53 3.53h56.86a3.53 3.53 0 0 0 3.53-3.53V10.33a3.53 3.53 0 0
0-3.53-3.53"></path>
                               <path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path>
                               <path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path>
                               <path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                           </svg>

                           <input type="file" id="additional-images-input"
name="additional-images[]" multiple class="hidden file-input" accept=".jpg,
.jpeg, .png">
                           <p class="block mt-6 text-md text-gray-700
mb-2">Drag and drop you image files here</p>
                           <p class="block mt-5 text-md text-gray-700 mb-2">Or,
                               <label for="additional-images-input"
data-file-name="" class="file-input cursor-pointer font-medium text-sm
md:text-base underline">browse to upload.</label>
                           </p>


                       </div>
                       <div id="additional-images-preview" class="py-2">
                           No files currently selected for upload
                       </div>
                   </div>


               </div>
               <h2 class="text-sm md:text-base mb-3 md:pl-4
text-gray-600"><span class="font-bold text-indigo-600">Tip: &nbsp</span>Ensure
your images are of high resolution as they significantly impact your property's
visibility. Studies indicate that properties with quality images have a 68%
higher chance of attracting attention within the first six months.</h2>
           </div>

           <!-- Fourth Step: Description -->
           <div class="step hidden" id="step-description">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Description</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-2">
                   <!-- Property Description -->
                   <div class="rounded-md md:p-6 md:col-span-2">
```

```
                       <label for="description" class="block text-md
font-medium text-gray-700 mb-2">Description</label>
                       <textarea id="description" name="description" rows="6"
                                 class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500"><?php echo
$property['description']?></textarea>
                   </div>
               </div>

               <div class="rounded-md md:p-6 md:pb-8">
                   <label for="full_description" class="block text-md
font-medium text-gray-700 mb-2">Full Description</label>
                   <div id="editor">
                       <p>Hello World!</p>
                       <p>Some initial <strong>bold</strong> text</p>
                       <p><br /></p>
                   </div>
               </div>

               <!-- Hidden input field to store Quill editor content -->
               <input type="hidden" id="full_description"
name="full_description">
               <h2 class="text-sm md:text-base mb-3 md:pl-4
text-gray-600"><span class="font-bold text-indigo-600">Tip: &nbsp</span>When
crafting your property description, focus on highlighting unique features and
amenities to captivate potential buyers or renters and increase interest in
your listing.</h2>

           </div>

           <!-- Fifth Step: Structure -->
           <div class="step hidden" id="step-structure">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Edit your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Structure</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Structure fields (bedrooms, bathrooms, balconies) -->
                   <div class="rounded-md md:p-4">
                       <label for="bedrooms" class="block text-md font-medium
text-gray-700 mb-2">Number of Bedrooms</label>
                       <input type="text" id="bedrooms" name="bedrooms"
value="<?php echo $property['bedrooms']?>"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
                   <div class="rounded-md md:p-4">
```

```
                       <label for="bathrooms" class="block text-md font-medium
text-gray-700 mb-2">Number of Bathrooms</label>
                       <input type="text" id="bathrooms" name="bathrooms"
value="<?php echo $property['bathrooms']?>"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="balconies" class="block text-md font-medium
text-gray-700 mb-2">Number of Balconies</label>
                       <input type="text" id="balconies" name="balconies"
value="<?php echo $property['balconies']?>"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
               </div>
           </div>

           <!-- Step Indicator -->
           <!--            <div id="step-indicator" class="text-center
mb-8"></div>-->
           <div id="step-indicator" class="flex justify-center mb-6">
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
           </div>
       </div>
       <div class="flex justify-between mt-4 md:mt-6">
           <button type="button" id="prev-step" class="prev-step-button text-sm
md:text-base hidden w-28 m-2 md:m-4 px-4 py-2 bg-gray-500 text-white rounded-md
hover:bg-gray-400 focus:outline-none focus:bg-gray-400">Previous</button>
           <button type="button" id="next-step" class="next-step-button text-sm
md:text-base w-28 m-2 md:m-4 px-4 py-2 rounded-md bg-indigo-600 leading-6
text-white shadow-sm hover:bg-indigo-500 focus-visible:outline
focus-visible:outline-2 focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Next</button>
           <button type="submit" id="submit-property" class="next-step-button
text-base md:text-lg hidden w-40 m-2 md:m-4 px-4 py-2 rounded-md bg-indigo-600
leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline
```

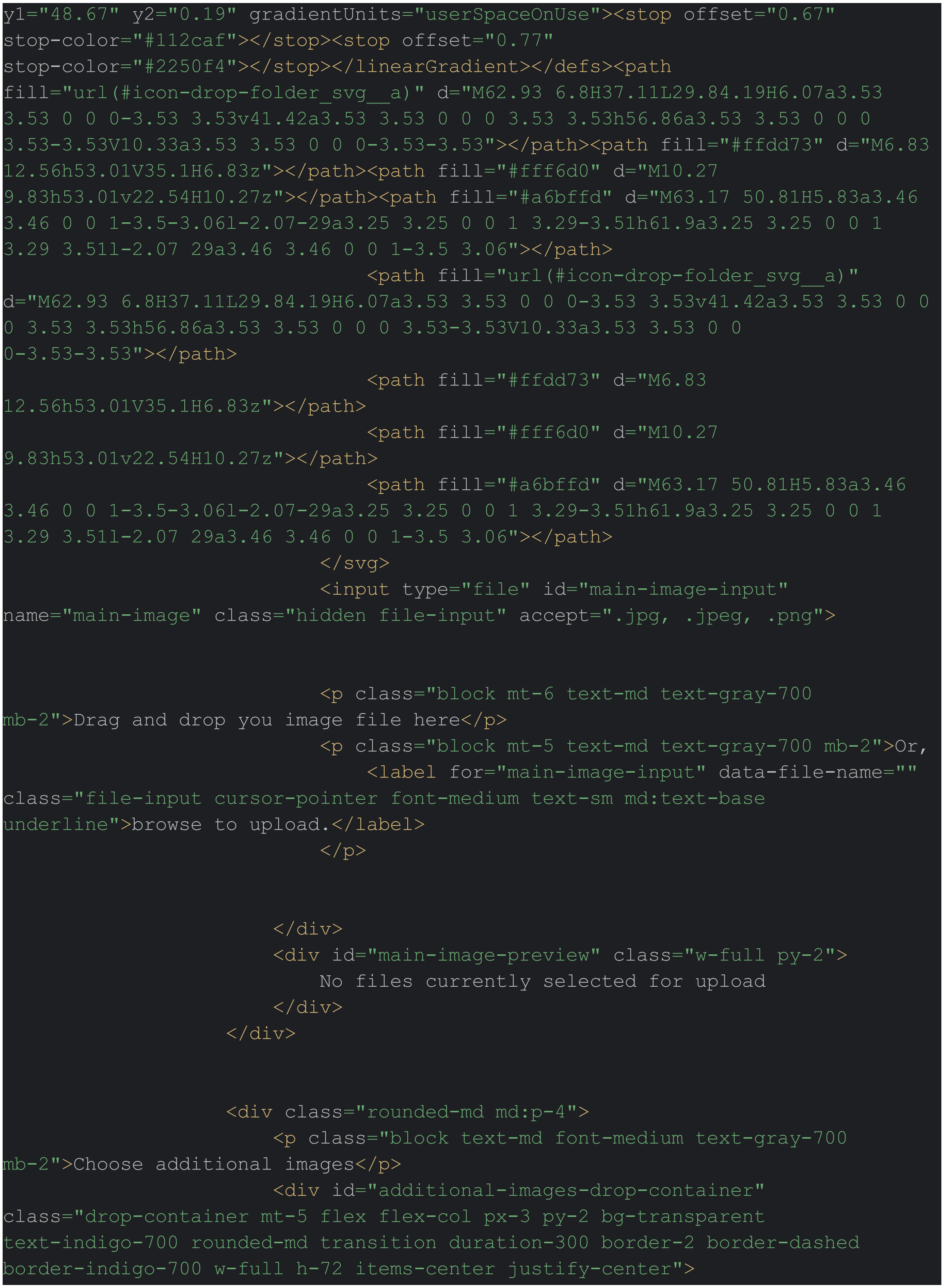
```
focus-visible:outline-2 focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Edit Property</button>
       </div>
   </form>
</main>
<script>
   const preselectCountryId = "<?php echo $property['country_id']; ?>";
   const preselectCityId = "<?php echo $property['city_id']; ?>";
   const propertyId = "<?php echo $property['id']; ?>"
</script>
<script src="https://cdn.jsdelivr.net/npm/quill@2.0.2/dist/quill.js"></script>
<script>
   const quill = new Quill('#editor', {
       theme: 'snow'
   });


   const description = "<?php echo $property['full_description']; ?>";
   quill.root.innerHTML = description;


   document.querySelector('#full_description').value = description;

   document.getElementById('property-form').addEventListener('submit',
function(event) {

       var fullDescription = quill.root.innerHTML;

       // Set the HTML content as the value of the hidden input field
       document.querySelector('#full_description').value = fullDescription;
   });
</script>
<script src="https://cdn.tailwindcss.com/"></script>
<script type="module" src="public/property/js/edit-property.js"></script>
</body>
</html>
```

Add-property.phtml

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Property Page</title>
   <link
href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css"
rel="stylesheet">
```

```
   <script src="https://kit.fontawesome.com/a279dcf9f7.js"
crossorigin="anonymous"></script>
   <link href="https://cdn.jsdelivr.net/npm/quill@2.0.2/dist/quill.snow.css"
rel="stylesheet" />

   <link rel="apple-touch-icon" sizes="76x76" href="/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
   <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
   <link rel="manifest" href="/site.webmanifest">
   <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5">
   <meta name="msapplication-TileColor" content="#da532c">
   <meta name="theme-color" content="#ffffff">
</head>
<body class="bg-gray-100">

<!-- Navbar -->
<?php require "public/navigation.phtml"; ?>

<main class="container mx-auto md:max-w-screen-xl mt-8 md:mt-24 px-4 mb-8
md:mb-24">
   <form id="property-form" action="services/property/add-property.php"
method="POST" enctype="multipart/form-data">
       <div class="flex flex-col bg-white rounded-md shadow-md align-baseline
p-6 md:p-6">



           <!-- First Step: General Info -->
           <div class="step" id="step-basic-info">
               <h2 class="text-base md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4 md:mb-5
md:pl-4">General Info</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Property Name -->
                   <div class="rounded-md md:p-4">
                       <label for="property-name" class="block text-md
font-medium text-gray-700 mb-2">Property Name</label>
                       <input type="text" id="property-name" name="name"
                              class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                   </div>
                   <!-- Property Type -->
                   <div class="rounded-md md:p-4">
                       <label for="typeDdl" class="block text-md font-medium
text-gray-700 mb-2">Property Type</label>
                       <select id="typeDdl" name="type"
```

```
                               class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">

                       </select>
                   </div>
                   <!-- Property Price -->
                   <div class="rounded-md md:p-4">
                       <label for="property-price" class="block text-md
font-medium text-gray-700 mb-2">Price</label>
                       <div class="flex items-center">
                           <input type="text" id="property-price" name="price"
                                  class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                           <span class="ml-3 text-xl">$</span>
                       </div>

                   </div>
                   <!-- Property Size -->
                   <div class=" rounded-md md:p-4 ">
                       <label for="property-price" class="block text-md
font-medium text-gray-700 mb-2">Size</label>
                       <div class="flex items-center">
                       <input type="text" id="property-price" name="size"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                           <span class="ml-3 text-xl">m<sup>2</span>
                       </div>
                   </div>
               </div>
           </div>

           <!-- Second Step: Location -->
           <div class="step hidden" id="step-location">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Location</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Location fields (Country, City, Address) -->
                   <div class="rounded-md md:p-4 md:col-span-2">
                       <label for="countryDdl" class="block text-md font-medium
text-gray-700 mb-2">Country</label>
                       <select id="countryDdl" name="country" class="w-full
px-3 py-2 border rounded-md focus:outline-none focus:border-indigo-500">
                           <option value="USA">USA</option>
                           <option value="Canada">Canada</option>
                           <option value="UK">UK</option>
                           <!-- Add more options as needed -->
                       </select>
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="cityDdl" class="block text-md font-medium
text-gray-700 mb-2">City</label>
                       <select id="cityDdl" name="city_id" class="w-full px-3
py-2 border rounded-md focus:outline-none focus:border-indigo-500">
                           <!-- Add more options as needed -->
                       </select>
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="address" class="block text-md font-medium
text-gray-700 mb-2">Address</label>
                       <input type="text" id="address" name="address"
                              class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                   </div>
                   <!-- Zip Code -->
                   <div class="rounded-md md:p-4">
                       <label for="zip" class="block text-md font-medium
text-gray-700 mb-2">Zip Code</label>
                       <input type="text" id="zip" name="zip"
                              class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500">
                   </div>
               </div>
           </div>

           <!-- Third Step: Images -->
           <div class="step hidden" id="stepImages">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Images</h1>
               <div class="grid grid-cols-1 md:grid-cols-1 gap-3 md:gap-6
pb-10">


                   <div class="rounded-md md:p-4">
                       <p class="block text-md font-medium text-gray-700
mb-2">Choose main image</p>
                       <div id="main-image-drop-container"
class="drop-container mt-5 flex flex-col px-3 py-2 bg-transparent
text-indigo-700 rounded-md transition duration-300 border-2 border-dashed
border-indigo-700 w-full h-72 items-center justify-center" draggable="true">

                           <svg xmlns="http://www.w3.org/2000/svg" width="69"
height="51" viewBox="0 0 69 51" class="dropzone-content__image tw-mx-auto
tw-mb-2"><defs><linearGradient id="icon-drop-folder_svg__a" x1="34.5" x2="34.5"
```

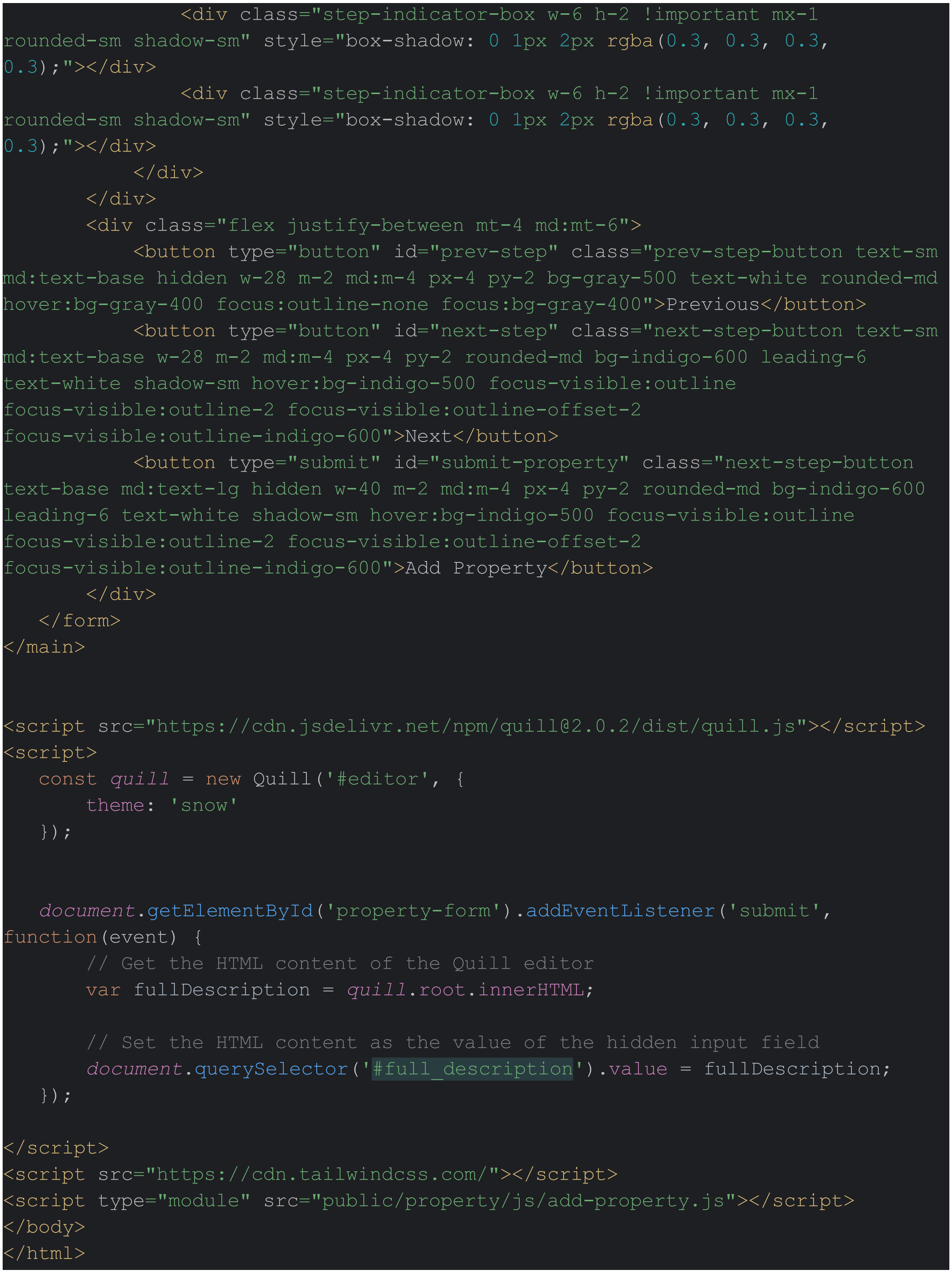


```
y1="48.67" y2="0.19" gradientUnits="userSpaceOnUse"><stop offset="0.67"
stop-color="#112caf"></stop><stop offset="0.77"
stop-color="#2250f4"></stop></linearGradient></defs><path
fill="url(#icon-drop-folder_svg__a)" d="M62.93 6.8H37.11L29.84.19H6.07a3.53
3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0 0 3.53 3.53h56.86a3.53 3.53 0 0 0
3.53-3.53V10.33a3.53 3.53 0 0 0-3.53-3.53"></path><path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path><path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path><path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                               <path fill="url(#icon-drop-folder_svg__a)"
d="M62.93 6.8H37.11L29.84.19H6.07a3.53 3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0
0 3.53 3.53h56.86a3.53 3.53 0 0 0 3.53-3.53V10.33a3.53 3.53 0 0
0-3.53-3.53"></path>
                               <path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path>
                               <path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path>
                               <path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                           </svg>
                           <input type="file" id="main-image-input"
name="main-image" class="hidden file-input" accept=".jpg, .jpeg, .png">


                           <p class="block mt-6 text-md text-gray-700
mb-2">Drag and drop you image file here</p>
                           <p class="block mt-5 text-md text-gray-700 mb-2">Or,
                               <label for="main-image-input" data-file-name=""
class="file-input cursor-pointer font-medium text-sm md:text-base
underline">browse to upload.</label>
                           </p>



                       </div>
                       <div id="main-image-preview" class="w-full py-2">
                           No files currently selected for upload
                       </div>
                   </div>



                   <div class="rounded-md md:p-4">
                       <p class="block text-md font-medium text-gray-700
mb-2">Choose additional images</p>
                       <div id="additional-images-drop-container"
class="drop-container mt-5 flex flex-col px-3 py-2 bg-transparent
text-indigo-700 rounded-md transition duration-300 border-2 border-dashed
border-indigo-700 w-full h-72 items-center justify-center">
```

```
                            <svg xmlns="http://www.w3.org/2000/svg" width="69"
height="51" viewBox="0 0 69 51" class="dropzone-content__image tw-mx-auto
tw-mb-2"><defs><linearGradient id="icon-drop-folder_svg__a" x1="34.5" x2="34.5"
y1="48.67" y2="0.19" gradientUnits="userSpaceOnUse"><stop offset="0.67"
stop-color="#112caf"></stop><stop offset="0.77"
stop-color="#2250f4"></stop></linearGradient></defs><path
fill="url(#icon-drop-folder_svg__a)" d="M62.93 6.8H37.11L29.84.19H6.07a3.53
3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0 0 3.53 3.53h56.86a3.53 3.53 0 0 0
3.53-3.53V10.33a3.53 3.53 0 0 0-3.53-3.53"></path><path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path><path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path><path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                                <path fill="url(#icon-drop-folder_svg__a)"
d="M62.93 6.8H37.11L29.84.19H6.07a3.53 3.53 0 0 0-3.53 3.53v41.42a3.53 3.53 0 0
0 3.53 3.53h56.86a3.53 3.53 0 0 0 3.53-3.53V10.33a3.53 3.53 0 0
0-3.53-3.53"></path>
                                <path fill="#ffdd73" d="M6.83
12.56h53.01V35.1H6.83z"></path>
                                <path fill="#fff6d0" d="M10.27
9.83h53.01v22.54H10.27z"></path>
                                <path fill="#a6bffd" d="M63.17 50.81H5.83a3.46
3.46 0 0 1-3.5-3.06l-2.07-29a3.25 3.25 0 0 1 3.29-3.51h61.9a3.25 3.25 0 0 1
3.29 3.51l-2.07 29a3.46 3.46 0 0 1-3.5 3.06"></path>
                            </svg>

                            <input type="file" id="additional-images-input"
name="additional-images[]" multiple class="hidden file-input" accept=".jpg,
.jpeg, .png">
                            <p class="block mt-6 text-md text-gray-700
mb-2">Drag and drop you image files here</p>
                            <p class="block mt-5 text-md text-gray-700 mb-2">Or,
                                <label for="additional-images-input"
data-file-name="" class="file-input cursor-pointer font-medium text-sm
md:text-base underline">browse to upload.</label>
                            </p>


                        </div>
                        <div id="additional-images-preview" class="py-2">
                            No files currently selected for upload
                        </div>
                    </div>


                </div>
                <h2 class="text-sm md:text-base mb-3 md:pl-4
text-gray-600"><span class="font-bold text-indigo-600">Tip: &nbsp</span>Ensure
your images are of high resolution as they significantly impact your property's
```

```
visibility. Studies indicate that properties with quality images have a 68%
higher chance of attracting attention within the first six months.</h2>
           </div>

           <!-- Fourth Step: Description -->
           <div class="step hidden" id="step-description">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Description</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-2">
                   <!-- Property Description -->
                   <div class="rounded-md md:p-6 md:col-span-2">
                       <label for="description" class="block text-md
font-medium text-gray-700 mb-2">Description</label>
                       <textarea id="description" name="description" rows="6"
                                 class="w-full px-3 py-2 border rounded-md
focus:outline-none focus:border-indigo-500"></textarea>
                   </div>

               </div>

               <div class="rounded-md md:p-6 md:pb-8">
                   <label for="full_description" class="block text-md
font-medium text-gray-700 mb-2">Full Description</label>
                   <div id="editor">
                       <p>Here goes full description of your property</p>
                       <p>Describe your property <strong>the best</strong> you
can!</p>
                       <p><br /></p>
                   </div>
               </div>

               <!-- Hidden input field to store Quill editor content -->
               <input type="hidden" id="full_description"
name="full_description">

               <!-- Create the editor container -->

               <!-- Property Description -->





               <h2 class="text-sm md:text-base mb-3 md:pl-4
text-gray-600"><span class="font-bold text-indigo-600">Tip: &nbsp</span>When
crafting your property description, focus on highlighting unique features and
```

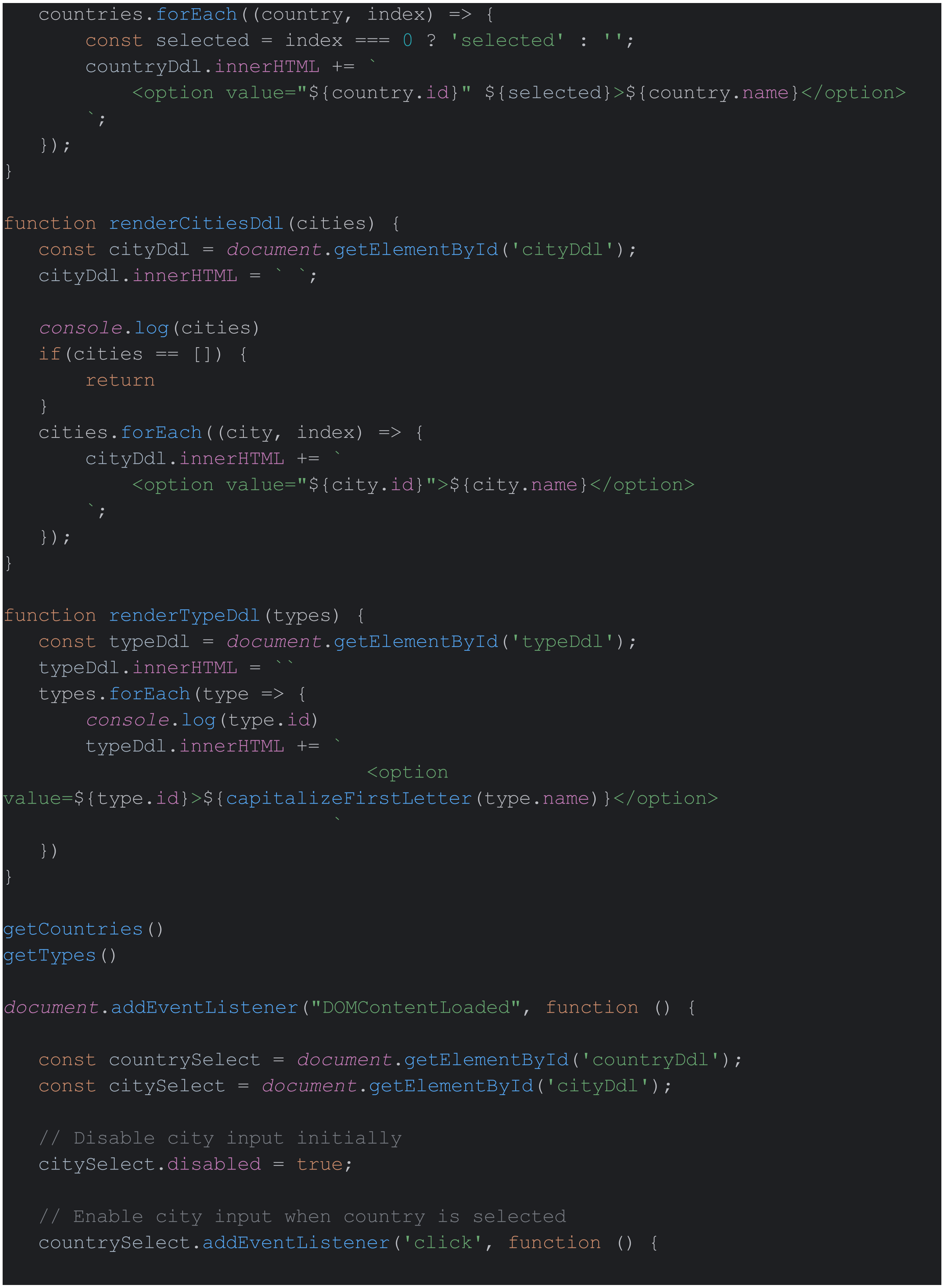
```
amenities to captivate potential buyers or renters and increase interest in
your listing.</h2>


           </div>

           <!-- Fifth Step: Structure -->
           <div class="step hidden" id="step-structure">
               <h2 class="text-md md:text-lg font-bold text-indigo-600 mb-3
md:pl-4">Add your property</h2>
               <h1 class="text-xl md:text-2xl font-bold mb-4
md:pl-4">Structure</h1>
               <div class="grid grid-cols-1 md:grid-cols-3 gap-3 md:gap-6
pb-10">
                   <!-- Structure fields (bedrooms, bathrooms, balconies) -->
                   <div class="rounded-md md:p-4">
                       <label for="bedrooms" class="block text-md font-medium
text-gray-700 mb-2">Number of Bedrooms</label>
                       <input type="text" id="bedrooms" name="bedrooms"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="bathrooms" class="block text-md font-medium
text-gray-700 mb-2">Number of Bathrooms</label>
                       <input type="text" id="bathrooms" name="bathrooms"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
                   <div class="rounded-md md:p-4">
                       <label for="balconies" class="block text-md font-medium
text-gray-700 mb-2">Number of Balconies</label>
                       <input type="text" id="balconies" name="balconies"
                              class="number-input w-full px-3 py-2 border
rounded-md focus:outline-none focus:border-indigo-500">
                   </div>
               </div>
           </div>

           <!-- Step Indicator -->
           <!--            <div id="step-indicator" class="text-center
mb-8"></div>-->
           <div id="step-indicator" class="flex justify-center mb-6">
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
               <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
```



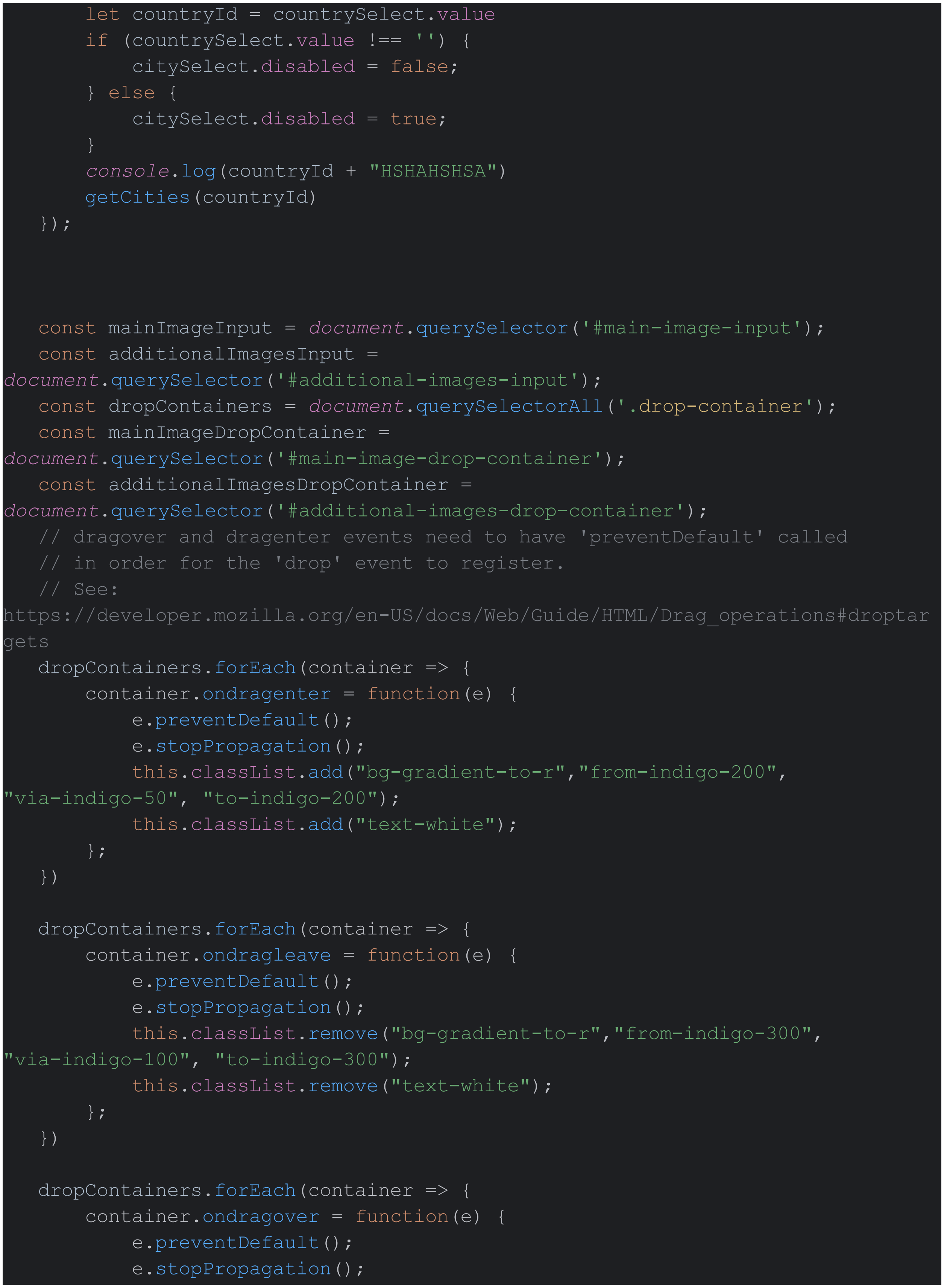
```
                <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
                <div class="step-indicator-box w-6 h-2 !important mx-1
rounded-sm shadow-sm" style="box-shadow: 0 1px 2px rgba(0.3, 0.3, 0.3,
0.3);"></div>
            </div>
        </div>
        <div class="flex justify-between mt-4 md:mt-6">
            <button type="button" id="prev-step" class="prev-step-button text-sm
md:text-base hidden w-28 m-2 md:m-4 px-4 py-2 bg-gray-500 text-white rounded-md
hover:bg-gray-400 focus:outline-none focus:bg-gray-400">Previous</button>
            <button type="button" id="next-step" class="next-step-button text-sm
md:text-base w-28 m-2 md:m-4 px-4 py-2 rounded-md bg-indigo-600 leading-6
text-white shadow-sm hover:bg-indigo-500 focus-visible:outline
focus-visible:outline-2 focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Next</button>
            <button type="submit" id="submit-property" class="next-step-button
text-base md:text-lg hidden w-40 m-2 md:m-4 px-4 py-2 rounded-md bg-indigo-600
leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline
focus-visible:outline-2 focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Add Property</button>
        </div>
    </form>
</main>


<script src="https://cdn.jsdelivr.net/npm/quill@2.0.2/dist/quill.js"></script>
<script>
    const quill = new Quill('#editor', {
        theme: 'snow'
    });


    document.getElementById('property-form').addEventListener('submit',
function(event) {
        // Get the HTML content of the Quill editor
        var fullDescription = quill.root.innerHTML;

        // Set the HTML content as the value of the hidden input field
        document.querySelector('#full_description').value = fullDescription;
    });

</script>
<script src="https://cdn.tailwindcss.com/"></script>
<script type="module" src="public/property/js/add-property.js"></script>
</body>
</html>
```

add-property.js

```
import { capitalizeFirstLetter } from "../../../util/formatting-functions.js";

async function getCountries() {
    try {
        const response = await fetch('services/property/get-countries.php');
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const countries = await response.json();
        renderCountriesDdl(countries);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

async function getCities(countryId) {
    console.log(countryId)
    try {
        const response = await
fetch(`services/property/get-cities.php?country_id=${countryId}`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const cities = await response.json();
        console.log(cities)
        renderCitiesDdl(cities);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

async function getTypes() {
    try {
        const response = await fetch(`services/property/get-types.php`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const cities = await response.json();
        renderTypeDdl(cities);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

function renderCountriesDdl(countries) {
    var countryDdl = document.getElementById('countryDdl');
    countryDdl.innerHTML = ``;
```



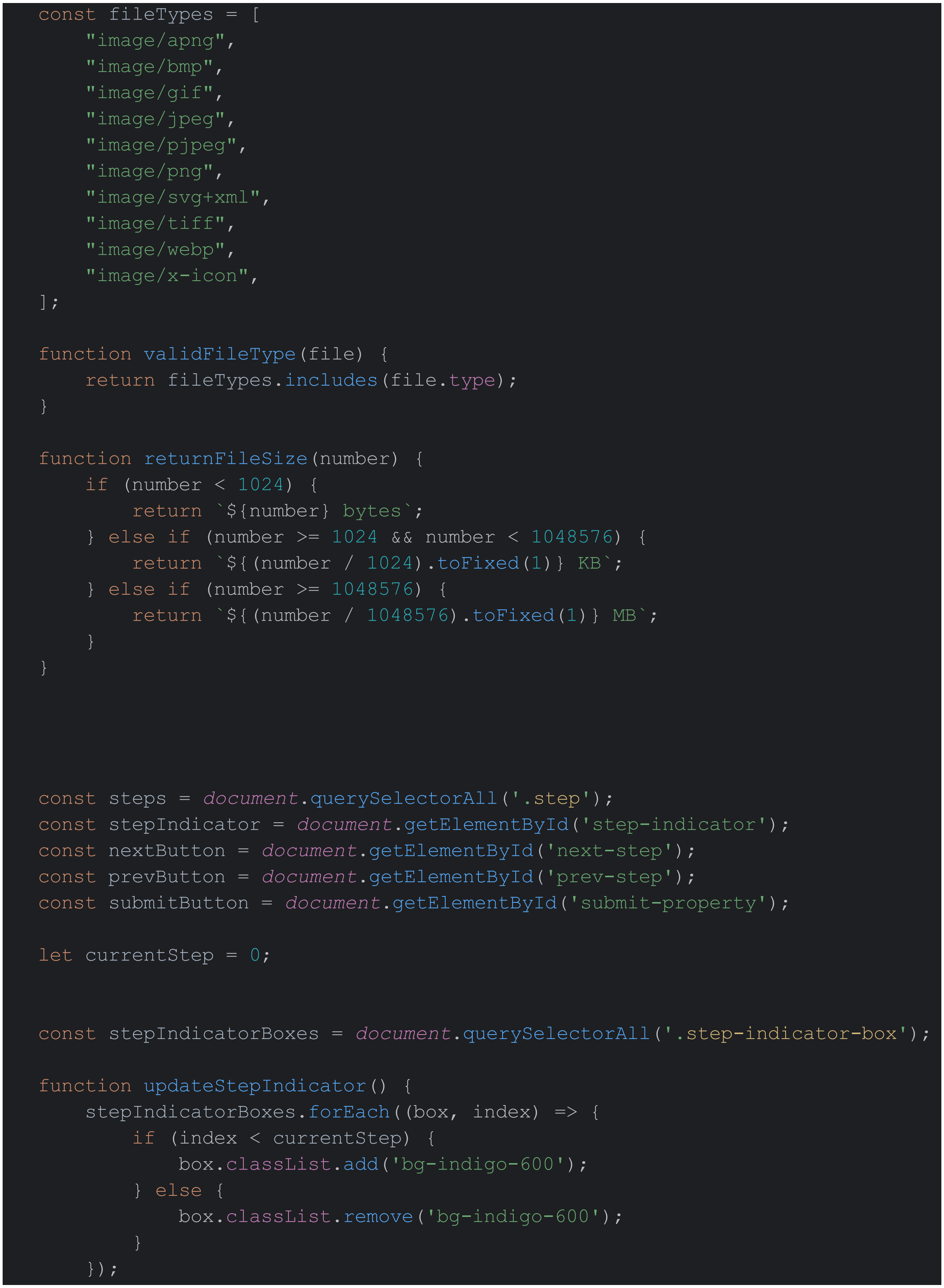
```
    countries.forEach((country, index) => {
        const selected = index === 0 ? 'selected' : '';
        countryDdl.innerHTML += `
            <option value="${country.id}" ${selected}>${country.name}</option>
        `;
    });
}

function renderCitiesDdl(cities) {
    const cityDdl = document.getElementById('cityDdl');
    cityDdl.innerHTML = ` `;

    console.log(cities)
    if(cities == []) {
        return
    }
    cities.forEach((city, index) => {
        cityDdl.innerHTML += `
            <option value="${city.id}">${city.name}</option>
        `;
    });
}

function renderTypeDdl(types) {
    const typeDdl = document.getElementById('typeDdl');
    typeDdl.innerHTML = ``
    types.forEach(type => {
        console.log(type.id)
        typeDdl.innerHTML += `
                                <option
value=${type.id}>${capitalizeFirstLetter(type.name)}</option>
                            `
    })
}

getCountries()
getTypes()

document.addEventListener("DOMContentLoaded", function () {

    const countrySelect = document.getElementById('countryDdl');
    const citySelect = document.getElementById('cityDdl');

    // Disable city input initially
    citySelect.disabled = true;

    // Enable city input when country is selected
    countrySelect.addEventListener('click', function () {
```



```
       let countryId = countrySelect.value
       if (countrySelect.value !== '') {
           citySelect.disabled = false;
       } else {
           citySelect.disabled = true;
       }
       console.log(countryId + "HSHAHSHSA")
       getCities(countryId)
   });



   const mainImageInput = document.querySelector('#main-image-input');
   const additionalImagesInput =
document.querySelector('#additional-images-input');
   const dropContainers = document.querySelectorAll('.drop-container');
   const mainImageDropContainer =
document.querySelector('#main-image-drop-container');
   const additionalImagesDropContainer =
document.querySelector('#additional-images-drop-container');
   // dragover and dragenter events need to have 'preventDefault' called
   // in order for the 'drop' event to register.
   // See:
https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#droptar
gets
   dropContainers.forEach(container => {
       container.ondragenter = function(e) {
           e.preventDefault();
           e.stopPropagation();
           this.classList.add("bg-gradient-to-r","from-indigo-200",
"via-indigo-50", "to-indigo-200");
           this.classList.add("text-white");
       };
   })

   dropContainers.forEach(container => {
       container.ondragleave = function(e) {
           e.preventDefault();
           e.stopPropagation();
           this.classList.remove("bg-gradient-to-r","from-indigo-300",
"via-indigo-100", "to-indigo-300");
           this.classList.remove("text-white");
       };
   })

   dropContainers.forEach(container => {
       container.ondragover = function(e) {
           e.preventDefault();
           e.stopPropagation();
```

```
        };
    })


    mainImageDropContainer.ondrop = function(e) {
        e.stopPropagation();
        e.preventDefault();
        console.log('Dropped in main image')
        mainImageInput.files = e.dataTransfer.files;

        const dT = new DataTransfer();
        dT.items.add(e.dataTransfer.files[0]);
        console.log(e.dataTransfer.files[0])
        mainImageInput.files = dT.files;
        updateImageDisplay(mainImageInput, mainImagePreview,
mainImageDropContainer);
    };

    additionalImagesDropContainer.ondrop = function(e) {
        e.stopPropagation();
        e.preventDefault();
        console.log('Dropped in additional images')
        additionalImagesInput.files = e.dataTransfer.files;

        const dT = new DataTransfer();


        for(const file of e.dataTransfer.files) {
            console.log(file)
            dT.items.add(file)
        }

        console.log(dT.files)
        additionalImagesInput.files = dT.files;
        updateImageDisplay(additionalImagesInput, additionalImagesPreview,
additionalImagesDropContainer)
    };


    const mainImagePreview = document.querySelector("#main-image-preview")
    const additionalImagesPreview =
document.querySelector("#additional-images-preview")



    mainImageInput.addEventListener('change', () =>
updateImageDisplay(mainImageInput, mainImagePreview, mainImageDropContainer))
```

```
   additionalImagesInput.addEventListener('change', () =>
updateImageDisplay(additionalImagesInput, additionalImagesPreview,
additionalImagesDropContainer))

   function updateImageDisplay(input, preview, dropContainer) {
       dropContainer.classList.add("bg-gradient-to-r","from-indigo-200",
"via-indigo-50", "to-indigo-200");
       while (preview.firstChild) {
           preview.removeChild(preview.firstChild);
       }

       const curFiles = input.files;
       if (curFiles.length === 0) {
           const para = document.createElement("p");
           para.textContent = "No files currently selected for upload";
           preview.appendChild(para);
       } else {
           const list = document.createElement("ol");
           list.classList.add("flex", "flex-col", "w-full");
           preview.appendChild(list);

           for (const file of curFiles) {
               const listItem = document.createElement("li");
               listItem.classList.add("flex", "gap-5", "bg-gray-100", "p-5",
"rounded-md")
               const para = document.createElement("p");
               if (validFileType(file)) {
                   para.textContent = `File name ${file.name}, file size
${returnFileSize(
                       file.size,
                   )}.`;
                   const image = document.createElement("img");
                   image.classList.add("h-24")
                   image.src = URL.createObjectURL(file);
                   image.alt = image.title = file.name;

                   listItem.appendChild(image);
                   listItem.appendChild(para);
               } else {
                   para.textContent = `File name ${file.name}: Not a valid file
type. Update your selection.`;
                   listItem.appendChild(para);
               }

               list.appendChild(listItem);
           }
       }
   }
```



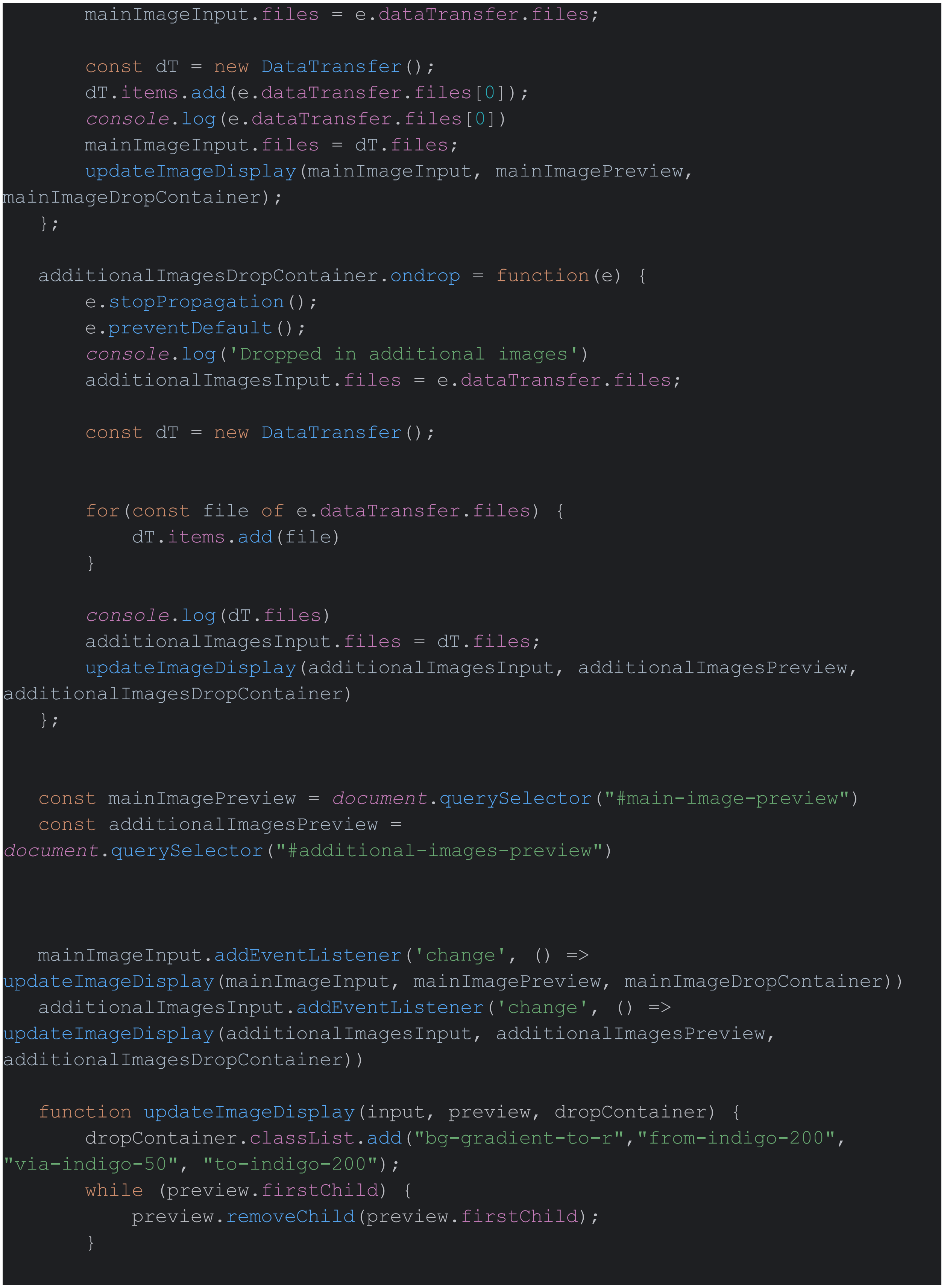
```
const fileTypes = [
    "image/apng",
    "image/bmp",
    "image/gif",
    "image/jpeg",
    "image/pjpeg",
    "image/png",
    "image/svg+xml",
    "image/tiff",
    "image/webp",
    "image/x-icon",
];

function validFileType(file) {
    return fileTypes.includes(file.type);
}

function returnFileSize(number) {
    if (number < 1024) {
        return `${number} bytes`;
    } else if (number >= 1024 && number < 1048576) {
        return `${(number / 1024).toFixed(1)} KB`;
    } else if (number >= 1048576) {
        return `${(number / 1048576).toFixed(1)} MB`;
    }
}




const steps = document.querySelectorAll('.step');
const stepIndicator = document.getElementById('step-indicator');
const nextButton = document.getElementById('next-step');
const prevButton = document.getElementById('prev-step');
const submitButton = document.getElementById('submit-property');

let currentStep = 0;


const stepIndicatorBoxes = document.querySelectorAll('.step-indicator-box');

function updateStepIndicator() {
    stepIndicatorBoxes.forEach((box, index) => {
        if (index < currentStep) {
            box.classList.add('bg-indigo-600');
        } else {
            box.classList.remove('bg-indigo-600');
        }
    });
    }

    function showStep(stepIndex) {
        updateStepIndicator();
        steps.forEach((step, index) => {
            if (index === stepIndex) {
                step.classList.remove('hidden');
            } else {
                step.classList.add('hidden');
            }
        });

        if(stepIndex == 0) {
            prevButton.classList.add("hidden")
        } else {
            prevButton.classList.remove("hidden")
        }


        if(stepIndex == steps.length - 1)  {
            submitButton.classList.remove('hidden')
            nextButton.classList.add('hidden')
        } else {
            submitButton.classList.add('hidden')
            nextButton.classList.remove('hidden')
        }

        updateStepIndicator();
    }

    function goToNextStep() {
        if (currentStep < steps.length - 1) {
            currentStep++;
            showStep(currentStep);

        } else if (currentStep === steps.length - 1) {

            // document.getElementById('add-property-form').submit(); // Submit
the form on the last step
        }
    }

    function goToPrevStep() {
        if (currentStep > 0) {
            currentStep--;
            showStep(currentStep);
        }
    }
    nextButton.addEventListener('click', goToNextStep);

    if (prevButton) { // Check if the previous button exists
        prevButton.addEventListener('click', goToPrevStep);
    }

    // Initial setup
    showStep(currentStep);


    const numberInputs = document.querySelectorAll('.number-input');

    // Add input event listener for each number input field
    numberInputs.forEach(input => {
        validateNumericInput(input);
    });

    showStep(currentStep);
});

function validateNumericInput(input) {
    console.log(123)
    input.addEventListener('input', function(event) {
        const value = event.target.value;
        const valid = /^\d*\.?\d*$/.test(value);
        if (!valid) {
            event.target.value = value.slice(0, -1); // Remove the last character if it's not a digit or dot
        }
    });
}
```

edit-property.js

```
import { capitalizeFirstLetter } from "../../../util/formatting-functions.js";

async function getCountries(preselectCountryId = null) {
    try {
        const response = await fetch('services/property/get-countries.php');
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const countries = await response.json();
        renderCountriesDdl(countries, preselectCountryId);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

```

```
async function getCities(countryId, preselectCity = null) {
    try {
        const response = await
fetch(`services/property/get-cities.php?country_id=${countryId}`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const cities = await response.json();
        renderCitiesDdl(cities, preselectCity);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

async function getTypes() {
    try {
        const response = await fetch(`services/property/get-types.php`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const cities = await response.json();
        renderTypeDdl(cities);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

function renderCountriesDdl(countries, preselectCountryId = null) {
    var countryDdl = document.getElementById('countryDdl');
    console.log('preselectCountryId')
    countryDdl.innerHTML = ``;
    countries.forEach((country, index) => {
        const selected = country.id === preselectCountryId ? 'selected' : '';

        countryDdl.innerHTML += `
            <option value="${country.id}" ${selected}>${country.name}</option>
        `;
    });
}

function renderCitiesDdl(cities, preselectCityId = null) {
    console.log(preselectCityId)
    const cityDdl = document.getElementById('cityDdl');
    cityDdl.innerHTML = ` `;
    if(cities) {

    }
    cities.forEach((city, index) => {
        const selected = city.id === preselectCityId ? 'selected' : '';
        cityDdl.innerHTML += `
            <option value="${city.id}" ${selected}>${city.name}</option>
        `;
    });
}

function renderTypeDdl(types) {
    const typeDdl = document.getElementById('typeDdl');
    typeDdl.innerHTML = ``
    types.forEach(type => {
        console.log(type.id)
        typeDdl.innerHTML += `
                               <option
value=${type.id}>${capitalizeFirstLetter(type.name)}</option>
                            `
    })
}

console.log(preselectCountryId)
getCountries(preselectCountryId);
getTypes()

document.addEventListener("DOMContentLoaded", function () {

    const countrySelect = document.getElementById('countryDdl');
    const citySelect = document.getElementById('cityDdl');


    // Enable city input when country is selected
    countrySelect.addEventListener('click', function () {
        let countryId = countrySelect.value
        if (countrySelect.value !== '') {
            citySelect.disabled = false;
        } else {
            citySelect.disabled = true;
        }
        console.log(countryId + "HSHAHSHSA")
        getCities(countryId)
    });


    // Preselect country and city if they are provided as data
    if (preselectCountryId) {
        countrySelect.value = preselectCountryId;
        getCities(preselectCountryId); // Fetch cities for the selected country
    }
    if (preselectCityId) {
        citySelect.value = preselectCityId;
    }
```

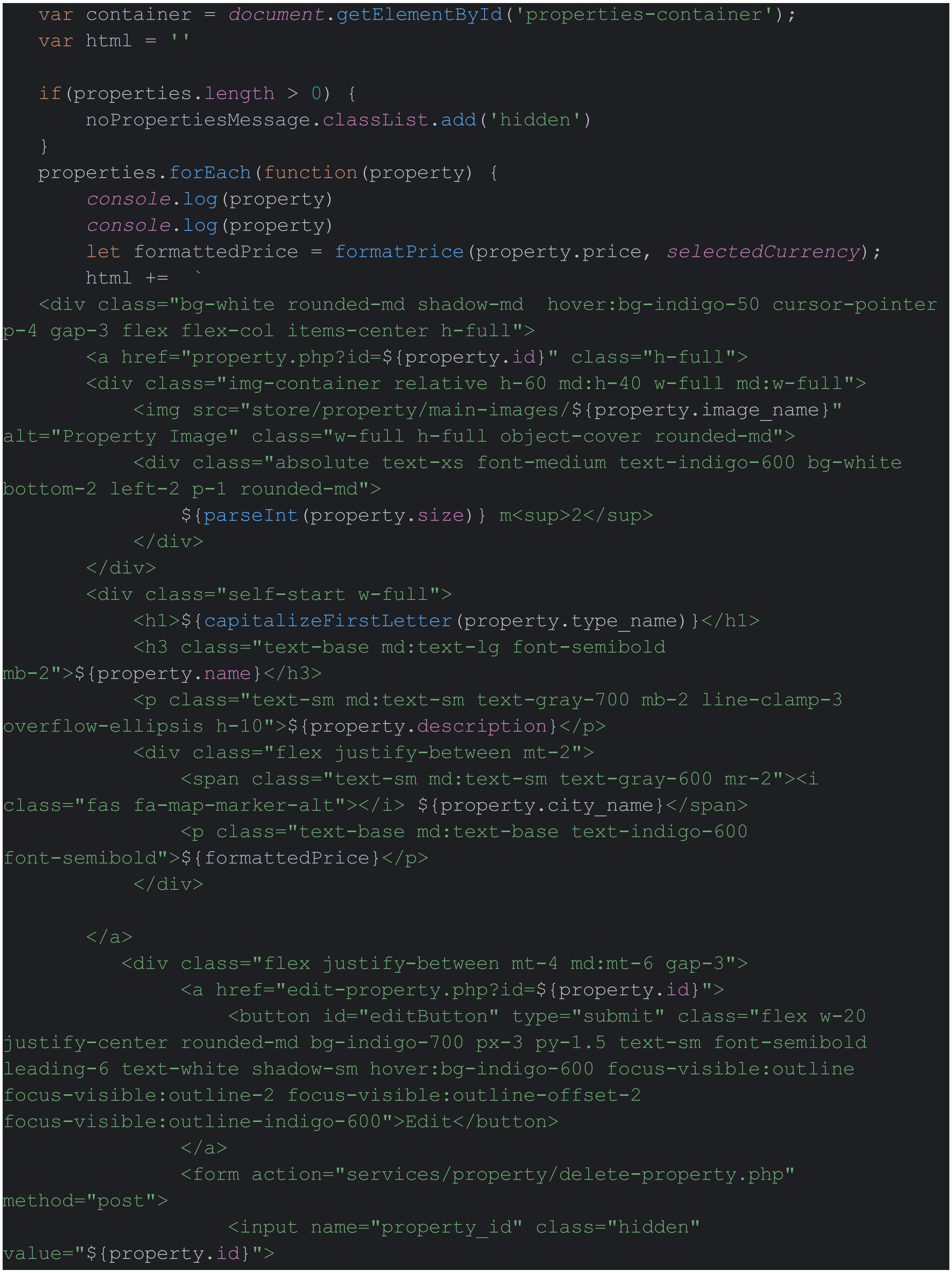
```
   const mainImageInput = document.querySelector('#main-image-input');
   const additionalImagesInput =
document.querySelector('#additional-images-input');
   const dropContainers = document.querySelectorAll('.drop-container');
   const mainImageDropContainer =
document.querySelector('#main-image-drop-container');
   const additionalImagesDropContainer =
document.querySelector('#additional-images-drop-container');
   // dragover and dragenter events need to have 'preventDefault' called
   // in order for the 'drop' event to register.
   // See:
https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_operations#droptar
gets
   dropContainers.forEach(container => {
       container.ondragenter = function(e) {
           e.preventDefault();
           e.stopPropagation();
           this.classList.add("bg-gradient-to-r","from-indigo-200",
"via-indigo-50", "to-indigo-200");
           this.classList.add("text-white");
       };
   })

   dropContainers.forEach(container => {
       container.ondragleave = function(e) {
           e.preventDefault();
           e.stopPropagation();
           this.classList.remove("bg-gradient-to-r","from-indigo-300",
"via-indigo-100", "to-indigo-300");
           this.classList.remove("text-white");
       };
   })

   dropContainers.forEach(container => {
       container.ondragover = function(e) {
           e.preventDefault();
           e.stopPropagation();
       };
   })


   mainImageDropContainer.ondrop = function(e) {
       e.stopPropagation();
       e.preventDefault();
       console.log('Dropped in main image')
```



```
        mainImageInput.files = e.dataTransfer.files;

        const dT = new DataTransfer();
        dT.items.add(e.dataTransfer.files[0]);
        console.log(e.dataTransfer.files[0])
        mainImageInput.files = dT.files;
        updateImageDisplay(mainImageInput, mainImagePreview,
mainImageDropContainer);
    };

    additionalImagesDropContainer.ondrop = function(e) {
        e.stopPropagation();
        e.preventDefault();
        console.log('Dropped in additional images')
        additionalImagesInput.files = e.dataTransfer.files;

        const dT = new DataTransfer();


        for(const file of e.dataTransfer.files) {
            dT.items.add(file)
        }

        console.log(dT.files)
        additionalImagesInput.files = dT.files;
        updateImageDisplay(additionalImagesInput, additionalImagesPreview,
additionalImagesDropContainer)
    };


    const mainImagePreview = document.querySelector("#main-image-preview")
    const additionalImagesPreview =
document.querySelector("#additional-images-preview")



    mainImageInput.addEventListener('change', () =>
updateImageDisplay(mainImageInput, mainImagePreview, mainImageDropContainer))
    additionalImagesInput.addEventListener('change', () =>
updateImageDisplay(additionalImagesInput, additionalImagesPreview,
additionalImagesDropContainer))

    function updateImageDisplay(input, preview, dropContainer) {
        dropContainer.classList.add("bg-gradient-to-r","from-indigo-200",
"via-indigo-50", "to-indigo-200");
        while (preview.firstChild) {
            preview.removeChild(preview.firstChild);
        }
        const curFiles = input.files;
        if (curFiles.length === 0) {
            const para = document.createElement("p");
            para.textContent = "No files currently selected for upload";
            preview.appendChild(para);
        } else {
            const list = document.createElement("ol");
            list.classList.add("flex", "flex-col", "w-full");
            preview.appendChild(list);

            for (const file of curFiles) {
                const listItem = document.createElement("li");
                listItem.classList.add("flex", "gap-5", "bg-gray-100", "p-5", "rounded-md")
                const para = document.createElement("p");
                if (validFileType(file)) {
                    para.textContent = `File name ${file.name}, file size ${returnFileSize(
                        file.size,
                    )}.`;
                    const image = document.createElement("img");
                    image.classList.add("h-24")
                    image.src = URL.createObjectURL(file);
                    image.alt = image.title = file.name;

                    listItem.appendChild(image);
                    listItem.appendChild(para);
                } else {
                    para.textContent = `File name ${file.name}: Not a valid file type. Update your selection.`;
                    listItem.appendChild(para);
                }

                list.appendChild(listItem);
            }
        }
    }

    const fileTypes = [
        "image/apng",
        "image/bmp",
        "image/gif",
        "image/jpeg",
        "image/pjpeg",
        "image/png",
        "image/svg+xml",
        "image/tiff",
        "image/webp",
        "image/x-icon",
    ];

    function validFileType(file) {
        return fileTypes.includes(file.type);
    }

    function returnFileSize(number) {
        if (number < 1024) {
            return `${number} bytes`;
        } else if (number >= 1024 && number < 1048576) {
            return `${(number / 1024).toFixed(1)} KB`;
        } else if (number >= 1048576) {
            return `${(number / 1048576).toFixed(1)} MB`;
        }
    }


    //TO BE IMPLEMENTED

    function prepopulateMainImageInput() {

    }

    function prepopulateAdditionalImageInput(fileNames) {
        // const dT = new DataTransfer();
        //
        // for(const file of fileNames) {
        //     dT.items.add(file)
        // }

    }

    async function fetchImages(propertyId) {
        try {
            const response = await
fetch(`services/property/get-images.php?id=${propertyId}`);
            if (!response.ok) {
                throw new Error(`Error fetching images:
${response.statusText}`);
            }
            const imageNames = await response.json();
            console.log(imageNames)
            if (imageNames.error) {
                throw new Error(`Server error: ${imageNames.error}`);
            }
            return imageNames;
        } catch (error) {
            console.error(error);
            return [];
        }
    }

    // Function to render images dynamically
    async function renderImages() {
        // const container =
document.querySelector('#additional-images-preview-container');
        // container.innerHTML = '';

        let imageUrls = await fetchImages(propertyId);

        prepopulateAdditionalImageInput(imageUrls)
        // Create img elements for each image URL and append them to the
container
        // imageUrls.forEach(imageUrl => {
        //
        // });
    }

    renderImages()

    //END



    const steps = document.querySelectorAll('.step');
    const stepIndicator = document.getElementById('step-indicator');
    const nextButton = document.getElementById('next-step');
    const prevButton = document.getElementById('prev-step');
    const submitButton = document.getElementById('submit-property');

    let currentStep = 0;


    const stepIndicatorBoxes = document.querySelectorAll('.step-indicator-box');

    function updateStepIndicator() {
        stepIndicatorBoxes.forEach((box, index) => {
            if (index < currentStep) {
                box.classList.add('bg-indigo-600');
            } else {
                box.classList.remove('bg-indigo-600');
            }
        });
    }

    function showStep(stepIndex) {
        updateStepIndicator();
        steps.forEach((step, index) => {
            if (index === stepIndex) {
                step.classList.remove('hidden');
            } else {
                step.classList.add('hidden');
            }
        });

        if(stepIndex == 0) {
            prevButton.classList.add("hidden")
        } else {
            prevButton.classList.remove("hidden")
        }


        if(stepIndex == steps.length - 1)  {
            submitButton.classList.remove('hidden')
            nextButton.classList.add('hidden')
        } else {
            submitButton.classList.add('hidden')
            nextButton.classList.remove('hidden')
        }

        updateStepIndicator();
    }

    function goToNextStep() {
        if (currentStep < steps.length - 1) {
            currentStep++;
            showStep(currentStep);

        } else if (currentStep === steps.length - 1) {

            // document.getElementById('add-property-form').submit(); // Submit
the form on the last step
        }
    }

    function goToPrevStep() {
        if (currentStep > 0) {
            currentStep--;
            showStep(currentStep);
        }
    }

    nextButton.addEventListener('click', goToNextStep);

    if (prevButton) { // Check if the previous button exists
        prevButton.addEventListener('click', goToPrevStep);
    }
    // Initial setup
    showStep(currentStep);


    const numberInputs = document.querySelectorAll('.number-input');

    // Add input event listener for each number input field
    numberInputs.forEach(input => {
        validateNumericInput(input);
    });

    showStep(currentStep);
});

function validateNumericInput(input) {
    console.log(123)
    input.addEventListener('input', function(event) {
        const value = event.target.value;
        const valid = /^\d*\.?\d*$/.test(value);
        if (!valid) {
            event.target.value = value.slice(0, -1); // Remove the last
character if it's not a digit or dot
        }
    });
}
```

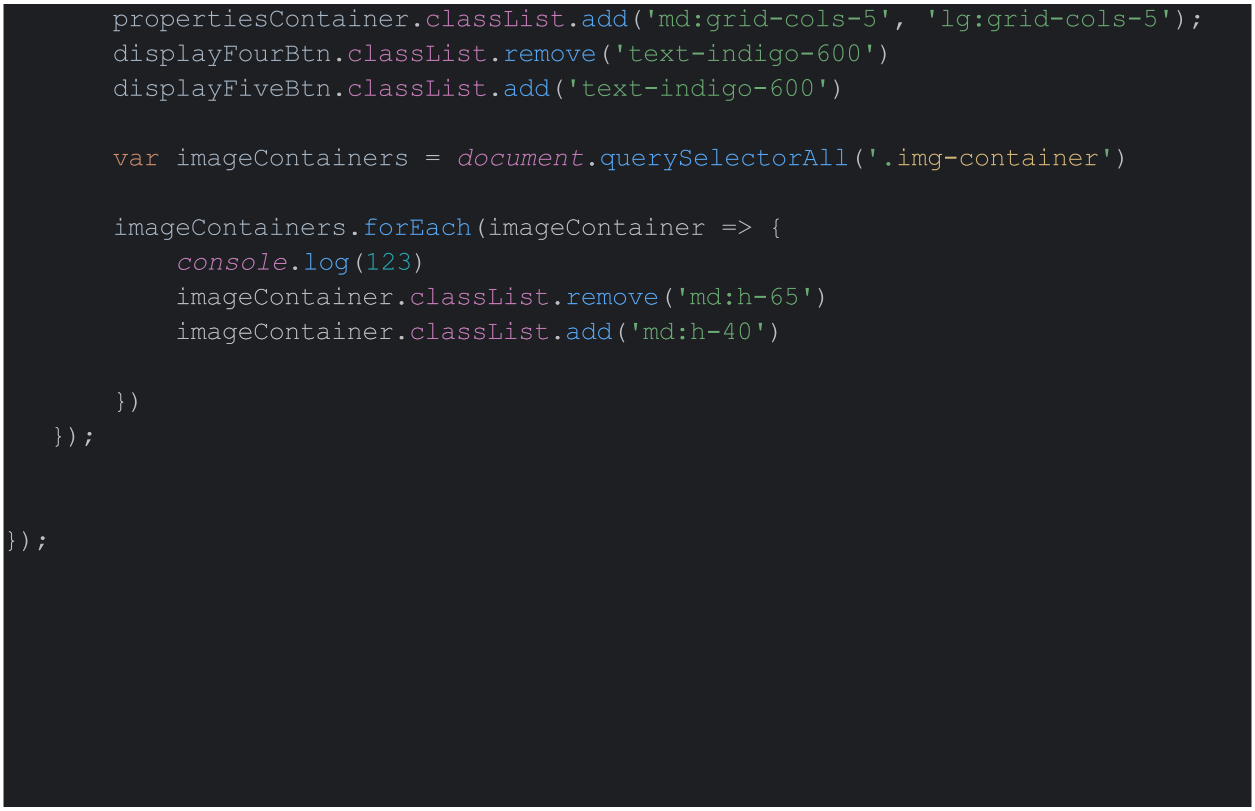
my-properties.js

```
import {formatPrice, capitalizeFirstLetter} from
"../../../util/formatting-functions.js"

async function getProperties() {
    try {
        const response = await
fetch(`services/property/get-my-properties.php?id=${userId}`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const properties = await response.json();
        renderProperties(properties);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

const noPropertiesMessage = document.querySelector('#no-properties-message')
function renderProperties(properties) {
```



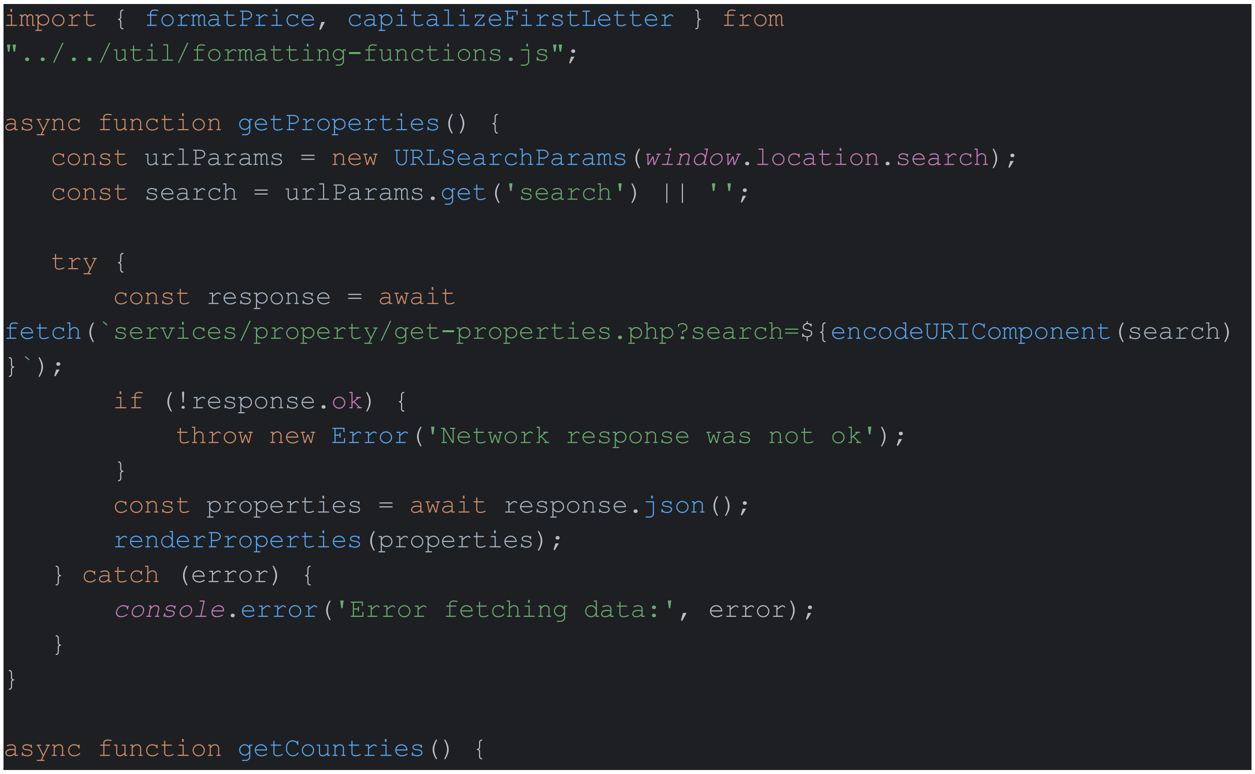
```
   var container = document.getElementById('properties-container');
   var html = ''

   if(properties.length > 0) {
       noPropertiesMessage.classList.add('hidden')
   }
   properties.forEach(function(property) {
       console.log(property)
       console.log(property)
       let formattedPrice = formatPrice(property.price, selectedCurrency);
       html +=  `
   <div class="bg-white rounded-md shadow-md  hover:bg-indigo-50 cursor-pointer
p-4 gap-3 flex flex-col items-center h-full">
       <a href="property.php?id=${property.id}" class="h-full">
       <div class="img-container relative h-60 md:h-40 w-full md:w-full">
           <img src="store/property/main-images/${property.image_name}"
alt="Property Image" class="w-full h-full object-cover rounded-md">
           <div class="absolute text-xs font-medium text-indigo-600 bg-white
bottom-2 left-2 p-1 rounded-md">
               ${parseInt(property.size)} m<sup>2</sup>
           </div>
       </div>
       <div class="self-start w-full">
           <h1>${capitalizeFirstLetter(property.type_name)}</h1>
           <h3 class="text-base md:text-lg font-semibold
mb-2">${property.name}</h3>
           <p class="text-sm md:text-sm text-gray-700 mb-2 line-clamp-3
overflow-ellipsis h-10">${property.description}</p>
           <div class="flex justify-between mt-2">
               <span class="text-sm md:text-sm text-gray-600 mr-2"><i
class="fas fa-map-marker-alt"></i> ${property.city_name}</span>
               <p class="text-base md:text-base text-indigo-600
font-semibold">${formattedPrice}</p>
           </div>

       </a>
          <div class="flex justify-between mt-4 md:mt-6 gap-3">
               <a href="edit-property.php?id=${property.id}">
                   <button id="editButton" type="submit" class="flex w-20
justify-center rounded-md bg-indigo-700 px-3 py-1.5 text-sm font-semibold
leading-6 text-white shadow-sm hover:bg-indigo-600 focus-visible:outline
focus-visible:outline-2 focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Edit</button>
               </a>
               <form action="services/property/delete-property.php"
method="post">
                   <input name="property_id" class="hidden"
value="${property.id}">
```

```
                    <button type="submit" class="flex w-20 justify-center
rounded-md bg-red-500 px-3 py-1.5 text-sm font-semibold leading-6 text-white
shadow-sm hover:bg-red-600 focus-visible:outline focus-visible:outline-2
focus-visible:outline-offset-2
focus-visible:outline-indigo-600">Delete</button>
                </form>
            </div>
        </div>
    </div>




`;
    });
    container.innerHTML = html;
}


getProperties()

document.addEventListener("DOMContentLoaded", function () {
    var displayFourBtn = document.getElementById('displayFourBtn');
    var displayFiveBtn = document.getElementById('displayFiveBtn');
    var propertiesContainer = document.getElementById('properties-container');


    displayFourBtn.addEventListener('click', function() {
        propertiesContainer.classList.remove('md:grid-cols-5',
'lg:grid-cols-5');
        propertiesContainer.classList.add('md:grid-cols-4', 'lg:grid-cols-4');
        displayFiveBtn.classList.remove('text-indigo-600')
        displayFourBtn.classList.add('text-indigo-600')


        var imageContainers = document.querySelectorAll('.img-container')
        console.log(imageContainers)
        imageContainers.forEach(imageContainer => {
            console.log(123)
            imageContainer.classList.remove('md:h-40')
            imageContainer.classList.add('md:h-65')

        })

    });

    displayFiveBtn.addEventListener('click', function() {
        propertiesContainer.classList.remove('md:grid-cols-4',
'lg:grid-cols-4');
```

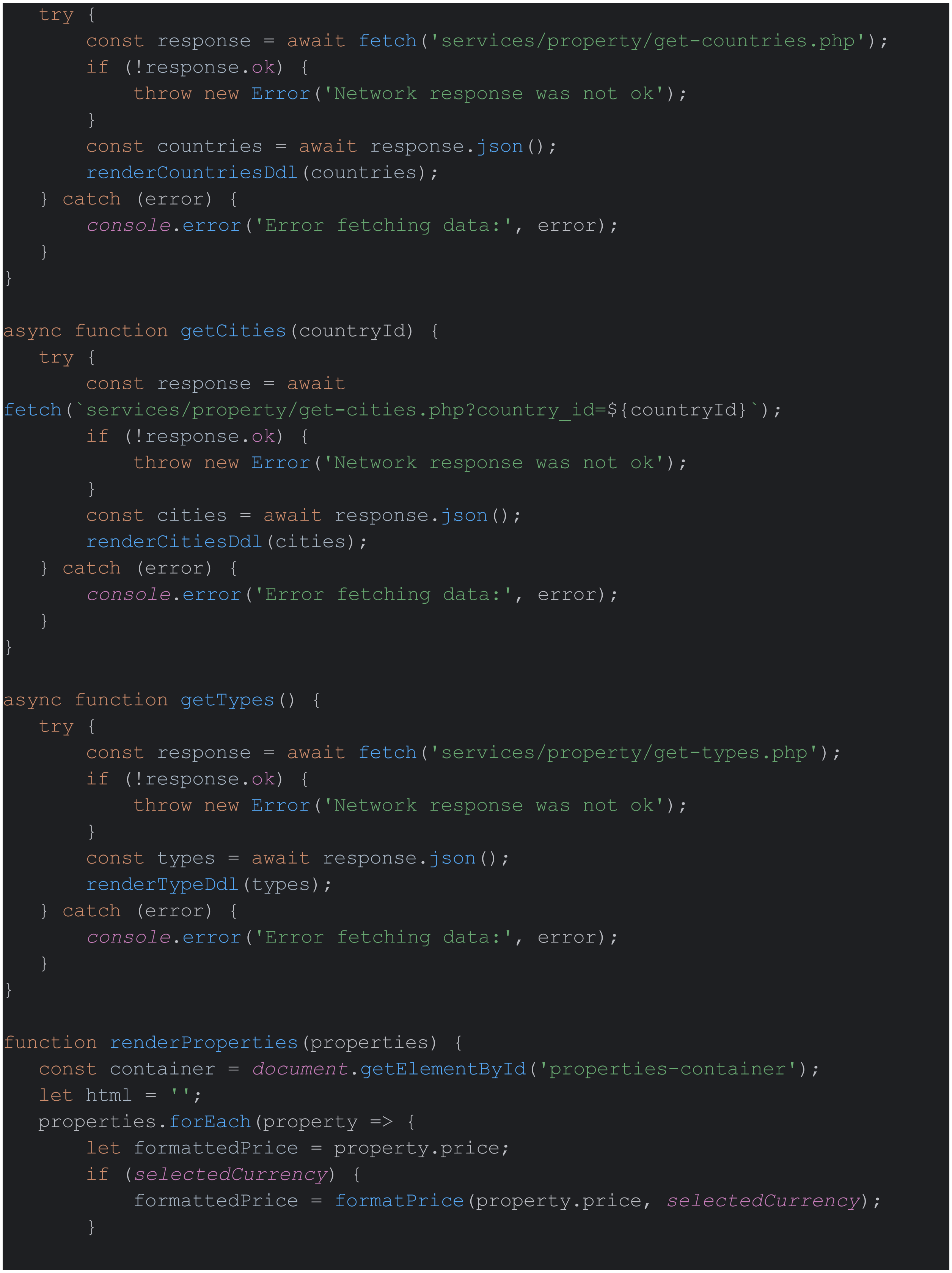


```
       propertiesContainer.classList.add('md:grid-cols-5', 'lg:grid-cols-5');
       displayFourBtn.classList.remove('text-indigo-600')
       displayFiveBtn.classList.add('text-indigo-600')

       var imageContainers = document.querySelectorAll('.img-container')

       imageContainers.forEach(imageContainer => {
           console.log(123)
           imageContainer.classList.remove('md:h-65')
           imageContainer.classList.add('md:h-40')

       })
   });


});
```

Index.js



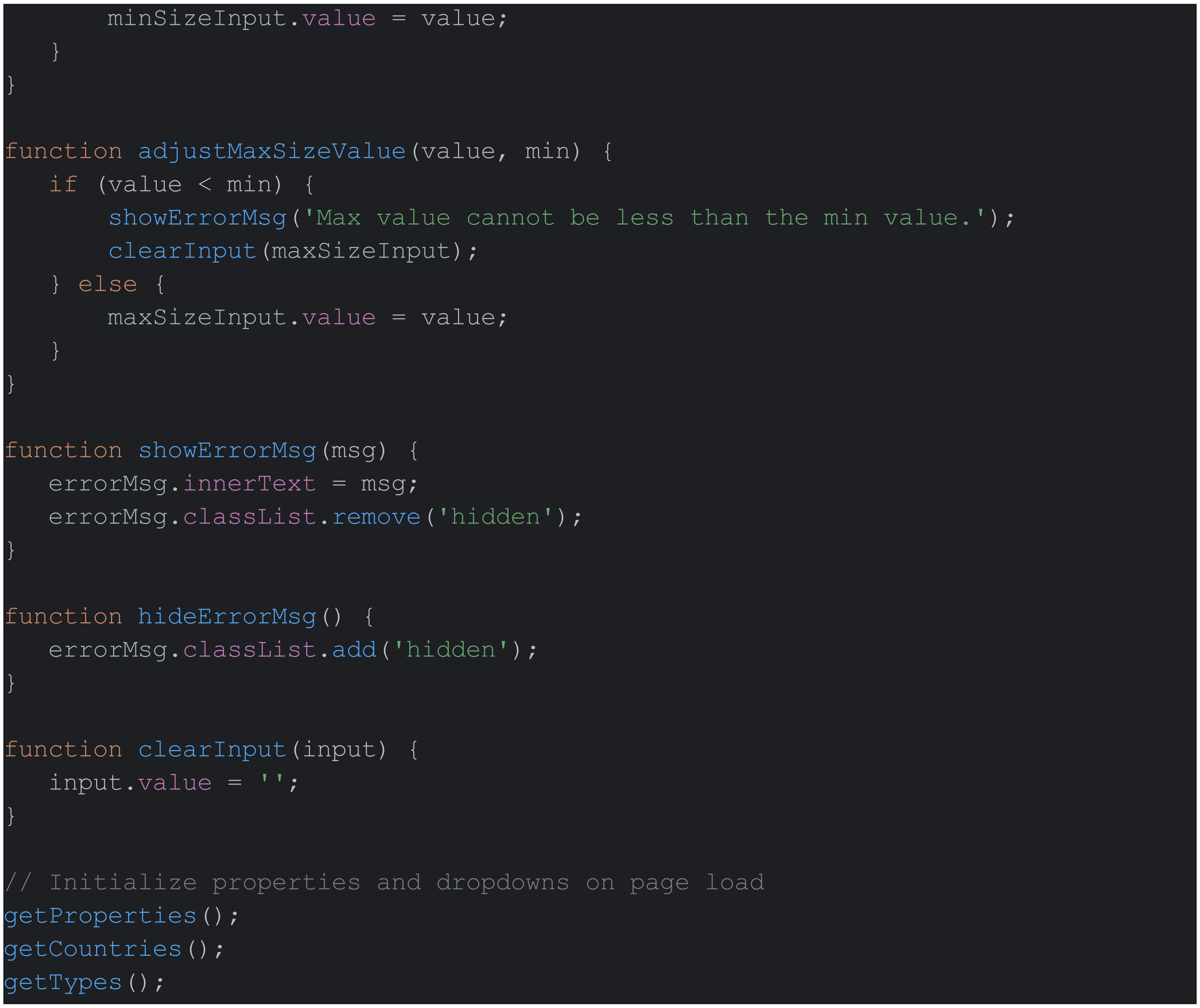
```
import { formatPrice, capitalizeFirstLetter } from
"../../util/formatting-functions.js";

async function getProperties() {
   const urlParams = new URLSearchParams(window.location.search);
   const search = urlParams.get('search') || '';

   try {
       const response = await
fetch(`services/property/get-properties.php?search=${encodeURIComponent(search)
}`);
       if (!response.ok) {
           throw new Error('Network response was not ok');
       }
       const properties = await response.json();
       renderProperties(properties);
   } catch (error) {
       console.error('Error fetching data:', error);
   }
}

async function getCountries() {
```



```
    try {
        const response = await fetch('services/property/get-countries.php');
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const countries = await response.json();
        renderCountriesDdl(countries);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

async function getCities(countryId) {
    try {
        const response = await
fetch(`services/property/get-cities.php?country_id=${countryId}`);
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const cities = await response.json();
        renderCitiesDdl(cities);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

async function getTypes() {
    try {
        const response = await fetch('services/property/get-types.php');
        if (!response.ok) {
            throw new Error('Network response was not ok');
        }
        const types = await response.json();
        renderTypeDdl(types);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
}

function renderProperties(properties) {
    const container = document.getElementById('properties-container');
    let html = '';
    properties.forEach(property => {
        let formattedPrice = property.price;
        if (selectedCurrency) {
            formattedPrice = formatPrice(property.price, selectedCurrency);
        }
```

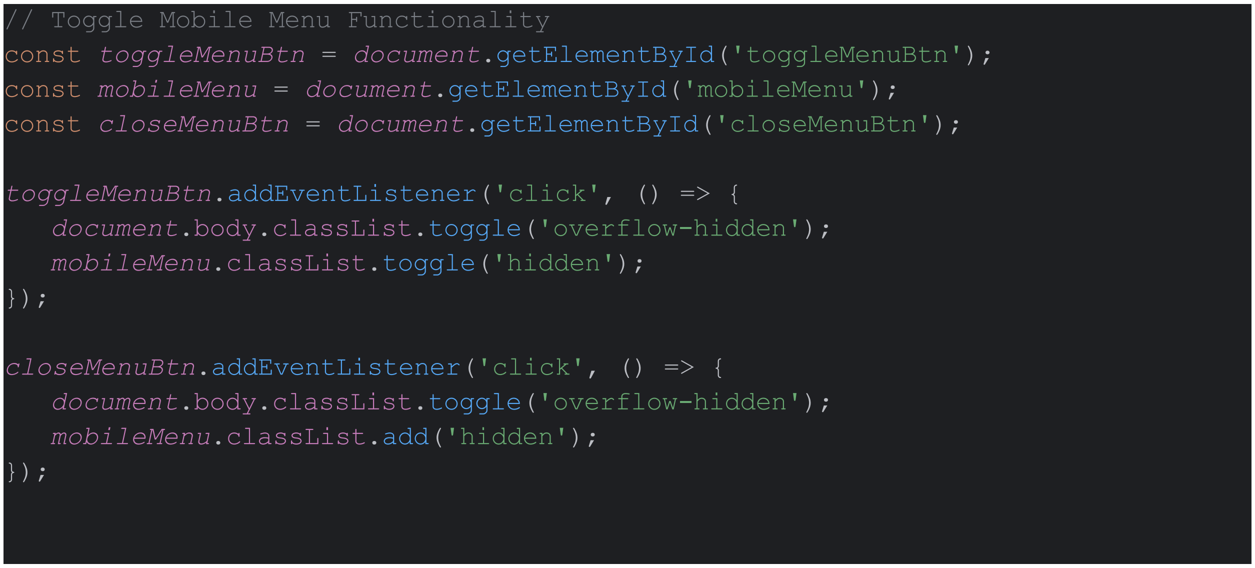
```
       const imageSrcString =
`https://flagcdn.com/w320/${property.country_short_name}.png`;

       html += `
           <a href="property.php?id=${property.id}" class="h-full">
               <div class="bg-white rounded-md shadow-md hover:bg-indigo-50
cursor-pointer p-4 gap-3 flex flex-col items-center h-full">
                   <div class="relative img-container h-60 md:h-40 w-full
md:w-full">
                       <img
src="store/property/main-images/${property.image_name}" alt="Property Image"
class="w-full h-full object-cover rounded-md">
                       <div class="absolute text-xs font-medium text-indigo-600
bg-white bottom-2 left-2 p-1 rounded-md">
                           ${parseInt(property.size)} m<sup>2</sup>
                       </div>
                   </div>
                   <div class="self-start w-full">
                       <h1>${capitalizeFirstLetter(property.type_name)}</h1>
                       <h3 class="text-base md:text-lg font-semibold
mb-2">${property.name}</h3>
                       <p class="text-sm md:text-sm text-gray-700 mb-2
line-clamp-3 overflow-ellipsis h-10">${property.description}</p>
                       <div class="flex justify-between mt-2">
                           <span class="text-sm md:text-sm text-gray-600
mr-2"><i class="fas fa-map-marker-alt"></i> ${property.city_name}</span>
                           <p class="text-base md:text-base text-indigo-600
font-semibold">${formattedPrice}</p>
                       </div>
                   </div>
               </div>
           </a>
       `;
   });

   container.innerHTML = html;
}

function renderCountriesDdl(countries) {
   const countryDdl = document.getElementById('countryFilter');
   countryDdl.innerHTML = `<option value="">All</option>`;
   countries.forEach(country => {
       countryDdl.innerHTML += `<option
value=${country.id}>${country.name}</option>`;
   });
}

function renderCitiesDdl(cities) {
   const cityDdl = document.getElementById('cityFilter');
```

```
    cityDdl.innerHTML = `<option value="">All</option>`;
    cities.forEach(city => {
        cityDdl.innerHTML += `<option value=${city.id}>${city.name}</option>`;
    });
}

function renderTypeDdl(types) {
    const typeDdl = document.getElementById('typeFilter');
    typeDdl.innerHTML = `<option value="">All</option>`;
    types.forEach(type => {
        typeDdl.innerHTML += `<option
value=${type.id}>${capitalizeFirstLetter(type.name)}</option>`;
    });
}

document.getElementById('searchForm').addEventListener('submit',
function(event) {
    event.preventDefault();
    const searchInput = document.getElementById('search').value;
    const url = new URL(window.location);
    url.searchParams.set('search', searchInput);
    window.history.pushState({}, '', url);
    getProperties();
});

document.addEventListener("DOMContentLoaded", function () {
    var toggleFiltersButton = document.getElementById('toggleFiltersBtn');
    var filterSection = document.getElementById('filtersSection');
    var displayThreeBtn = document.getElementById('displayThreeBtn');
    var displayFourBtn = document.getElementById('displayFourBtn');
    var propertiesContainer = document.getElementById('properties-container');

    toggleFiltersButton.addEventListener('click', function() {
        filterSection.classList.toggle('hidden');
        toggleFiltersButton.classList.toggle('text-indigo-600');
    });

    displayThreeBtn.addEventListener('click', function() {
        var imageContainers = document.querySelectorAll('.img-container');

        imageContainers.forEach(imageContainer => {
            imageContainer.classList.remove('md:h-40');
            imageContainer.classList.add('md:h-65');
        });

        propertiesContainer.classList.remove('md:grid-cols-4',
'lg:grid-cols-4');
        propertiesContainer.classList.add('md:grid-cols-3', 'lg:grid-cols-3');
        displayFourBtn.classList.remove('text-indigo-600');
        displayThreeBtn.classList.add('text-indigo-600');
    });

    displayFourBtn.addEventListener('click', function() {
        var imageContainers = document.querySelectorAll('.img-container');

        imageContainers.forEach(imageContainer => {
            imageContainer.classList.remove('md:h-65');
            imageContainer.classList.add('md:h-40');
        });

        propertiesContainer.classList.remove('md:grid-cols-3',
'lg:grid-cols-3');
        propertiesContainer.classList.add('md:grid-cols-4', 'lg:grid-cols-4');
        displayThreeBtn.classList.remove('text-indigo-600');
        displayFourBtn.classList.add('text-indigo-600');
    });

    const countrySelect = document.getElementById('countryFilter');
    const citySelect = document.getElementById('cityFilter');

    citySelect.disabled = true;

    countrySelect.addEventListener('click', function() {
        let countryId = countrySelect.value;
        if (countryId !== '') {
            citySelect.disabled = false;
        } else {
            citySelect.disabled = true;
        }
        getCities(countryId);
    });
});

const minPriceInput = document.getElementById('minPrice');
const maxPriceInput = document.getElementById('maxPrice');
const minSizeInput = document.getElementById('minSize');
const maxSizeInput = document.getElementById('maxSize');
const errorMsg = document.getElementById('errorMsg');

minPriceInput.addEventListener('change', restrictNegativeNumberAndMinMax);
maxPriceInput.addEventListener('change', restrictNegativeNumberAndMinMax);
minSizeInput.addEventListener('change', restrictNegativeNumberAndMinMax);
maxSizeInput.addEventListener('change', restrictNegativeNumberAndMinMax);

function restrictNegativeNumberAndMinMax(event) {
    const input = event.target;
    const value = parseFloat(input.value);
    const minPrice = parseFloat(minPriceInput.value);
    const maxPrice = parseFloat(maxPriceInput.value);
    const minSize = parseFloat(minSizeInput.value);
    const maxSize = parseFloat(maxSizeInput.value);

    if (value < 0) {
        showErrorMsg('You cannot enter a negative number.');
        clearInput(input);
    } else {
        hideErrorMsg();
        adjustMinMaxValues(input, value, minPrice, maxPrice, minSize, maxSize);
    }
}

function adjustMinMaxValues(input, value, minPrice, maxPrice, minSize, maxSize)
{
    if (input.id === 'minPrice') {
        adjustMinPriceValue(value, maxPrice);
    } else if (input.id === 'maxPrice') {
        adjustMaxPriceValue(value, minPrice);
    } else if (input.id === 'minSize') {
        adjustMinSizeValue(value, maxSize);
    } else if (input.id === 'maxSize') {
        adjustMaxSizeValue(value, minSize);
    }
}

function adjustMinPriceValue(value, max) {
    if (value > max) {
        showErrorMsg('Min value cannot exceed the max value.');
        clearInput(minPriceInput);
    } else {
        minPriceInput.value = value;
    }
}

function adjustMaxPriceValue(value, min) {
    if (value < min) {
        showErrorMsg('Max value cannot be less than the min value.');
        clearInput(maxPriceInput);
    } else {
        maxPriceInput.value = value;
    }
}

function adjustMinSizeValue(value, max) {
    if (value > max) {
        showErrorMsg('Min value cannot exceed the max value.');
        clearInput(minSizeInput);
    } else {
```



```
       minSizeInput.value = value;
   }
}

function adjustMaxSizeValue(value, min) {
   if (value < min) {
       showErrorMsg('Max value cannot be less than the min value.');
       clearInput(maxSizeInput);
   } else {
       maxSizeInput.value = value;
   }
}

function showErrorMsg(msg) {
   errorMsg.innerText = msg;
   errorMsg.classList.remove('hidden');
}

function hideErrorMsg() {
   errorMsg.classList.add('hidden');
}

function clearInput(input) {
   input.value = '';
}

// Initialize properties and dropdowns on page load
getProperties();
getCountries();
getTypes();
```

navigation.js



```
// Toggle Mobile Menu Functionality
const toggleMenuBtn = document.getElementById('toggleMenuBtn');
const mobileMenu = document.getElementById('mobileMenu');
const closeMenuBtn = document.getElementById('closeMenuBtn');

toggleMenuBtn.addEventListener('click', () => {
   document.body.classList.toggle('overflow-hidden');
   mobileMenu.classList.toggle('hidden');
});

closeMenuBtn.addEventListener('click', () => {
   document.body.classList.toggle('overflow-hidden');
   mobileMenu.classList.add('hidden');
});
```

```
function initializeAccount(userFullName) {
   // Use userFullName in your script

   // Example usage
   const userImgs = document.querySelectorAll(".userImg");
   userImgs.forEach(img => {
       img.src = getAvatarImage(userFullName);
   })
}

function getAvatarImage(fullName) {
   // Split the full name into first name and last name
   console.log(fullName)
   const [firstName, lastName] = fullName.split(" ");

   // Construct the URL to fetch the avatar image
   const apiUrl =
`https://ui-avatars.com/api/?background=random&color=fff&name=${firstName}+${lastName}`;

   // Return the URL
   return apiUrl;
}

// Call the function when the DOM content is loaded
document.addEventListener("DOMContentLoaded", () => {
   const fullNameElement = document.getElementById("userName");
   const userFullName = fullNameElement.textContent.trim();
   initializeAccount(userFullName);
});
```

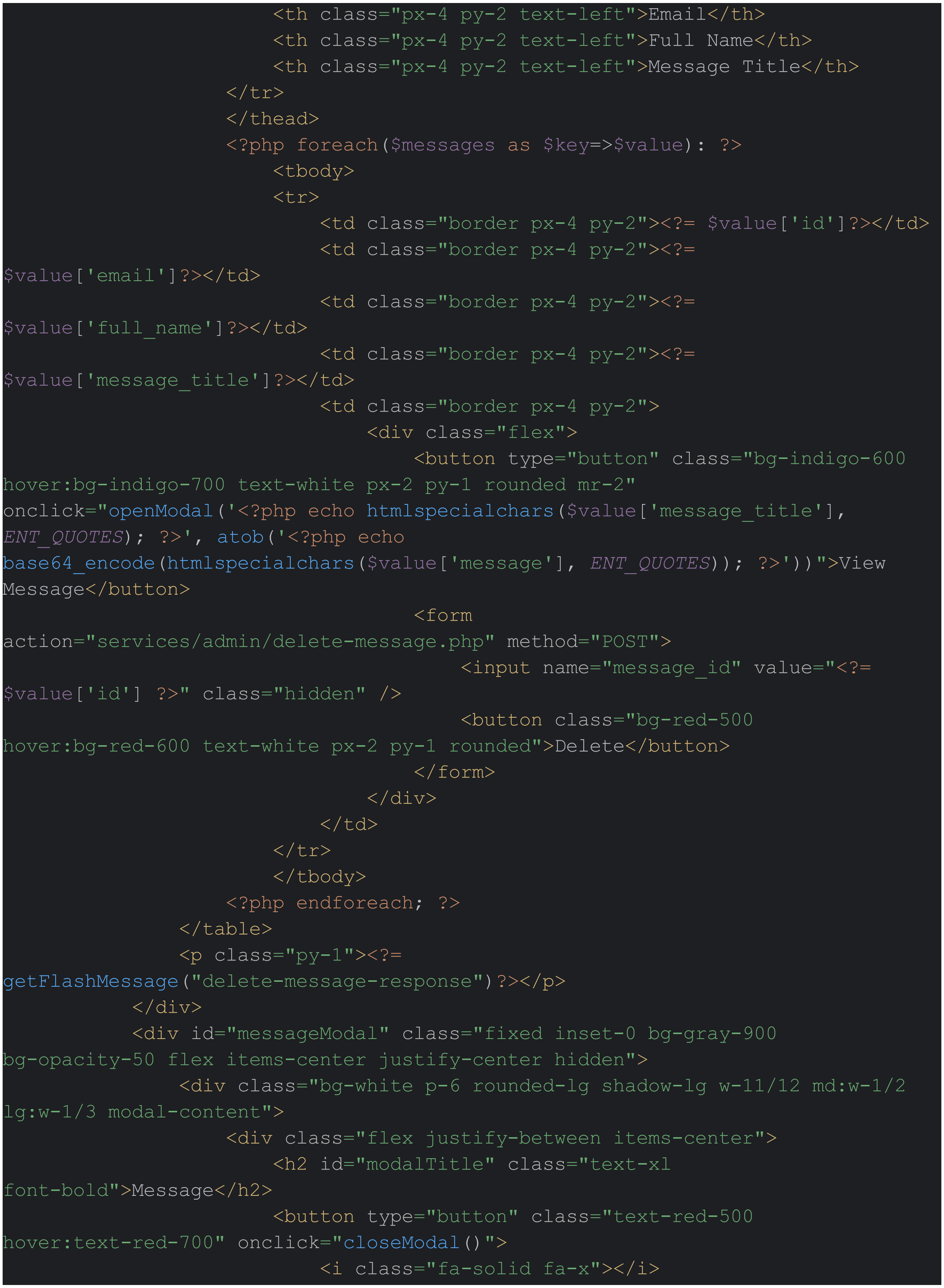
Admin-panel.phtml

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

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Property Page</title>
   <link
href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css"
rel="stylesheet">
   <script src="https://kit.fontawesome.com/a279dcf9f7.js"
crossorigin="anonymous"></script>

   <link rel="apple-touch-icon" sizes="76x76" href="/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
   <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
   <link rel="manifest" href="/site.webmanifest">
   <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5">
   <meta name="msapplication-TileColor" content="#da532c">
   <meta name="theme-color" content="#ffffff">

</head>

<?php require "public/navigation.phtml"; ?>

<body class="bg-gray-100">


<div class="block lg:hidden">
   <!-- Navbar -->
   <div class="flex justify-between bg-gray-800 p-4 text-white">
       <h1 class="text-xl font-bold">Admin Panel</h1>
       <button id="toggleSidebar" class="lg:hidden">
           <i class="fas fa-bars"></i>
       </button>
   </div>
</div>


<div class="flex lg:flex lg:flex-row" id="container">
   <!-- Sidebar -->
   <div  id="sidebar" class="bg-gray-800 text-gray-100 w-64 hidden lg:flex
h-screen lg:w-64 lg:h-screen">
       <div class="p-2 lg:p-4 -mt-3 lg:mt-0">
           <h1 class="text-xl font-bold hidden lg:block">Admin Panel</h1>
           <ul class="lg:mt-4">
               <li class="py-2">
                   <a href="#" class="flex items-center toggle-dropdown p-1.5
rounded-md" data-target="user-dropdown">
                       <i class="fa-solid fa-user mr-2"></i>
                       Users
                   </a>
                   <ul class="p-2 ml-4 hidden" id="user-dropdown">
<!--                        <li><a href="admin-panel.php?source=add_user"
class="block p-1.5 hover:bg-gray-700 rounded-md">Add User</a></li>-->
                       <li><a href="admin-panel.php?source=show_users"
class="block p-1.5 hover:bg-gray-700 rounded-md">Show Users</a></li>
                   </ul>
               </li>
               <li class="py-2">
                   <a href="#" class="flex items-center toggle-dropdown p-1.5
rounded-md" data-target="country-dropdown">
                       <i class="fa-solid fa-flag mr-2"></i>
                       Countries
                   </a>
                   <ul class="p-2 ml-4 hidden" id="country-dropdown">
                       <li><a href="admin-panel.php?source=add_country"
class="block p-1.5 hover:bg-gray-700 rounded-md">Add Country</a></li>
                       <li><a href="admin-panel.php?source=show_countries"
class="block p-1.5 hover:bg-gray-700 rounded-md">Show Countries</a></li>
                   </ul>
               </li>
               <li class="py-2">
                   <a href="#" class="flex items-center toggle-dropdown p-1.5
rounded-md" data-target="city-dropdown">
                       <i class="fa-solid fa-city mr-2"></i>
                       Cities
                   </a>
                   <ul class="p-2 ml-4 hidden" id="city-dropdown">
                       <li><a href="admin-panel.php?source=add_city"
class="block p-1.5 hover:bg-gray-700 rounded-md">Add City</a></li>
                       <li><a href="admin-panel.php?source=show_cities"
class="block p-1.5 hover:bg-gray-700 rounded-md">Show Cities</a></li>
                   </ul>
               </li>
               <li class="py-2">
                   <a href="#" class="flex items-center toggle-dropdown p-1.5
rounded-md" data-target="message-dropdown">
                       <i class="fa-solid fa-message mr-2"></i>
                       Messages
                   </a>
                   <ul class="p-2 ml-4 hidden" id="message-dropdown">
                       <li><a href="admin-panel.php?source=show_messages"
class="block p-1.5 hover:bg-gray-700 rounded-md">Show Messages</a></li>
                   </ul>
               </li>
           </ul>
       </div>
   </div>

   <!-- Content -->
   <div class="flex-grow p-8">
       <?php if(empty($source)) : ?>
           <h2 class="text-2xl font-bold mb-4">Welcome to Admin Panel</h2>

       <?php endif; ?>

       <?php if($source == 'show_users') : ?>
           <h2 class="text-2xl font-bold mb-4">Users</h2>
           <!-- Table for Users -->
           <div class="overflow-x-auto">
               <table class="table-auto min-w-full">
                   <thead>
                   <tr>
```

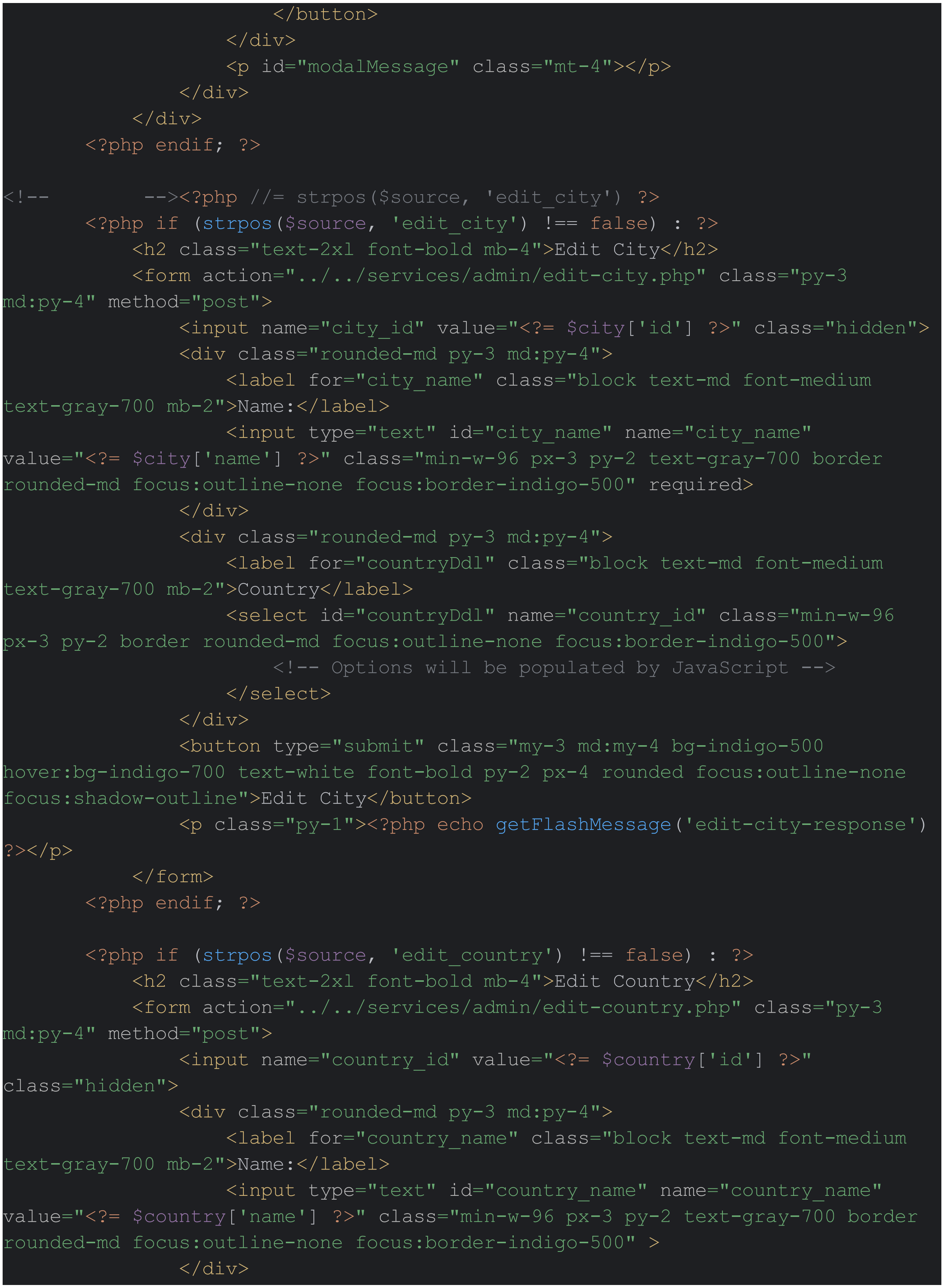
```
                       <th class="px-4 py-2 text-left">ID</th>
                       <th class="px-4 py-2 text-left">Name</th>
                       <th class="px-4 py-2 text-left">Email</th>
                   </tr>
                   </thead>
                   <?php foreach($users as $key=>$value): ?>
                       <tbody>
                       <!-- Sample User Row (Replace with dynamic data) -->
                       <tr>
                           <td class="border px-4 py-2"><?= $value['id']?></td>
                           <td class="border px-4 py-2"><?=
$value['full_name']?></td>
                           <td class="border px-4 py-2"><?=
$value['email']?></td>
<!--                            <td class="border px-4 py-2">-->
<!--                                <button class="bg-indigo-700
hover:bg-indigo-600 text-white px-2 py-1 rounded mr-2">Edit</button>-->
<!--                                <button class="bg-red-500 hover:bg-red-600
text-white px-2 py-1 rounded">Delete</button>-->
<!--                            </td>-->
                       </tr>
                       </tbody>
                   <?php endforeach; ?>
               </table>
           </div>
       <?php endif; ?>

       <?php if($source == 'show_countries') : ?>
           <h2 class="text-2xl font-bold mb-4">Countries</h2>
           <!-- Table for Users -->
           <div class="overflow-x-auto">
               <table class="table-auto min-w-full">
                   <thead>
                   <tr>
                       <th class="px-4 py-2 text-left">ID</th>
                       <th class="px-4 py-2 text-left">Name</th>
                       <th class="px-4 py-2 text-left">Flag</th>
                   </tr>
                   </thead>
                   <?php foreach($countries as $key=>$value): ?>
                       <tbody>
                       <!-- Sample User Row (Replace with dynamic data) -->
                       <tr>
                           <td class="border px-4 py-2"><?= $value['id']?></td>
                           <td class="border px-4 py-2"><?=
$value['name']?></td>
                           <td class="border px-4 py-2">
```

```
                                <img class="w-16"
src="https://flagcdn.com/w320/<?php echo $value['short_name']; ?>.png"
alt="<?php echo $property['name']; ?>"/>
                            </td>
                            <td class="border px-4 py-2 ">
                                <div class="flex">
                                    <a
href="admin-panel.php?source=edit_country&id=<?= $value['id']?>"
class="bg-indigo-700 hover:bg-indigo-600 text-white px-2 py-1 rounded
mr-2">Edit</a>
                                    <form
action="services/admin/delete-country.php" method="POST">
                                        <input name="country_id" value="<?=
$value['id']?>" class="hidden"/>
                                        <button class="bg-red-500
hover:bg-red-600 text-white px-2 py-1 rounded">Delete</button>
                                    </form>
                                </div>
                            </td>
                        </tr>
                        </tbody>
                    <?php endforeach; ?>
                </table>
                <p class="py-1"><?=
getFlashMessage("delete-country-response")?></p>
            </div>
        <?php endif; ?>


        <?php if($source == 'show_cities') : ?>
            <h2 class="text-2xl font-bold mb-4">Cities</h2>
            <!-- Table for Users -->
            <div class="overflow-x-auto">
                <table class="table-auto min-w-full">
                    <thead>
                    <tr>
                        <th class="px-4 py-2 text-left">ID</th>
                        <th class="px-4 py-2 text-left">Name</th>
                        <th class="px-4 py-2 text-left">Country Name</th>
                    </tr>
                    </thead>
                    <?php foreach($cities as $key=>$value): ?>
                        <tbody>
                        <!-- Sample User Row (Replace with dynamic data) -->
                        <tr>
                            <td class="border px-4 py-2"><?=
$value['city_id']?></td>
                            <td class="border px-4 py-2"><?=
$value['city_name']?></td>
```

```
                           <td class="border px-4 py-2"><?=
$value['country_name']?></td>
                           <td class="border px-4 py-2">
                               <div class="flex">
                                   <a
href="admin-panel.php?source=edit_city&id=<?= $value['city_id']?>"
class="bg-indigo-700 hover:bg-indigo-600 text-white px-2 py-1 rounded
mr-2">Edit</a>
                                   <form
action="services/admin/delete-city.php" method="POST">
                                       <input name="city_id" value="<?=
$value['city_id']?>" class="hidden"/>
                                       <button class="bg-red-500
hover:bg-red-600 text-white px-2 py-1 rounded">Delete</button>
                                   </form>
                               </div>
                           </td>
                       </tr>
                       </tbody>
                   <?php endforeach; ?>
               </table>
               <p class="py-1"><?=
getFlashMessage("delete-city-response")?></p>
           </div>
       <?php endif; ?>


       <?php if($source == 'add_country') : ?>
           <h2 class="text-2xl font-bold mb-4">Add Country</h2>
           <form action="../../services/admin/add-country.php" class="py-3 py-3
md:py-4" method="post" enctype="multipart/form-data">
               <div class="rounded-md py-3 py-3 md:py-4">
                   <label for="country_name" class="block text-md font-medium
text-gray-700 mb-2">Name:</label>
                   <input type="text" id="country_name" name="country_name"
class="min-w-96 px-3 py-2 text-gray-700 border rounded-md focus:outline-none
focus:border-indigo-500" required>
               </div>
               <div class="rounded-md py-3 py-3 md:py-4">
                   <label for="country_short_name" class="block text-md
font-medium text-gray-700 mb-2">Short Name:</label>
                   <input type="text" id="country_short_name"
name="country_short_name" class="min-w-96 px-3 py-2 text-gray-700 border
rounded-md focus:outline-none focus:border-indigo-500" required>
                   <p class="text-sm md:text-base mb-3 mt-2
text-gray-600">Important for flag images.</p>
               </div>
               <button type="submit" class="my-3 md:my-4 bg-indigo-500
hover:bg-indigo-700 text-white font-bold py-2 px-4 rounded focus:outline-none
focus:shadow-outline">Add Country</button>
               <p class="py-1"><?php echo
getFlashMessage('country-add-response')?></p>
           </form>
       <?php endif; ?>


       <?php if($source == 'add_city') : ?>
           <h2 class="text-2xl font-bold mb-4">Add City</h2>
           <form action="../../services/admin/add-city.php" class="py-3 py-3
md:py-4" method="post" enctype="multipart/form-data">
               <div class="rounded-md py-3 py-3 md:py-4">
                   <label for="city_name" class="block text-md font-medium
text-gray-700 mb-2">Name:</label>
                   <input type="text" id="city_name" name="city_name"
class="min-w-96 px-3 py-2 text-gray-700 border rounded-md focus:outline-none
focus:border-indigo-500" required>
               </div>
               <div class="rounded-md py-3 py-3 md:py-4">
                   <label for="countryDdl" class="block text-md font-medium
text-gray-700 mb-2">Country</label>

                   <select id="countryDdl" name="country_id" class="min-w-96
px-3 py-2 border rounded-md focus:outline-none focus:border-indigo-500">
                       <option value="USA">USA</option>
                       <option value="Canada">Canada</option>
                       <option value="UK">UK</option>
                       <!-- Add more options as needed -->
                   </select>
               </div>
               <button type="submit" class="my-3 md:my-4 bg-indigo-500
hover:bg-indigo-700 text-white font-bold py-2 px-4 rounded focus:outline-none
focus:shadow-outline">Add City</button>
               <p class="py-1"><?php echo
getFlashMessage('city-add-response')?></p>

           </form>
       <?php endif; ?>

       <?php if($source == 'show_messages') : ?>
           <h2 class="text-2xl font-bold mb-4">Messages</h2>
           <!-- Table for Messages -->
           <div class="overflow-x-auto">
               <table class="table-auto min-w-full">
                   <thead>
                   <tr>
                       <th class="px-4 py-2 text-left">ID</th>
```



```
                        <th class="px-4 py-2 text-left">Email</th>
                        <th class="px-4 py-2 text-left">Full Name</th>
                        <th class="px-4 py-2 text-left">Message Title</th>
                    </tr>
                    </thead>
                    <?php foreach($messages as $key=>$value): ?>
                        <tbody>
                        <tr>
                            <td class="border px-4 py-2"><?= $value['id']?></td>
                            <td class="border px-4 py-2"><?=
$value['email']?></td>
                            <td class="border px-4 py-2"><?=
$value['full_name']?></td>
                            <td class="border px-4 py-2"><?=
$value['message_title']?></td>
                            <td class="border px-4 py-2">
                                <div class="flex">
                                    <button type="button" class="bg-indigo-600
hover:bg-indigo-700 text-white px-2 py-1 rounded mr-2"
onclick="openModal('<?php echo htmlspecialchars($value['message_title'],
ENT_QUOTES); ?>', atob('<?php echo
base64_encode(htmlspecialchars($value['message'], ENT_QUOTES)); ?>'))">View
Message</button>
                                    <form
action="services/admin/delete-message.php" method="POST">
                                        <input name="message_id" value="<?=
$value['id'] ?>" class="hidden" />
                                        <button class="bg-red-500
hover:bg-red-600 text-white px-2 py-1 rounded">Delete</button>
                                    </form>
                                </div>
                            </td>
                        </tr>
                        </tbody>
                    <?php endforeach; ?>
                </table>
                <p class="py-1"><?=
getFlashMessage("delete-message-response")?></p>
            </div>
            <div id="messageModal" class="fixed inset-0 bg-gray-900
bg-opacity-50 flex items-center justify-center hidden">
                <div class="bg-white p-6 rounded-lg shadow-lg w-11/12 md:w-1/2
lg:w-1/3 modal-content">
                    <div class="flex justify-between items-center">
                        <h2 id="modalTitle" class="text-xl
font-bold">Message</h2>
                        <button type="button" class="text-red-500
hover:text-red-700" onclick="closeModal()">
                            <i class="fa-solid fa-x"></i>
```



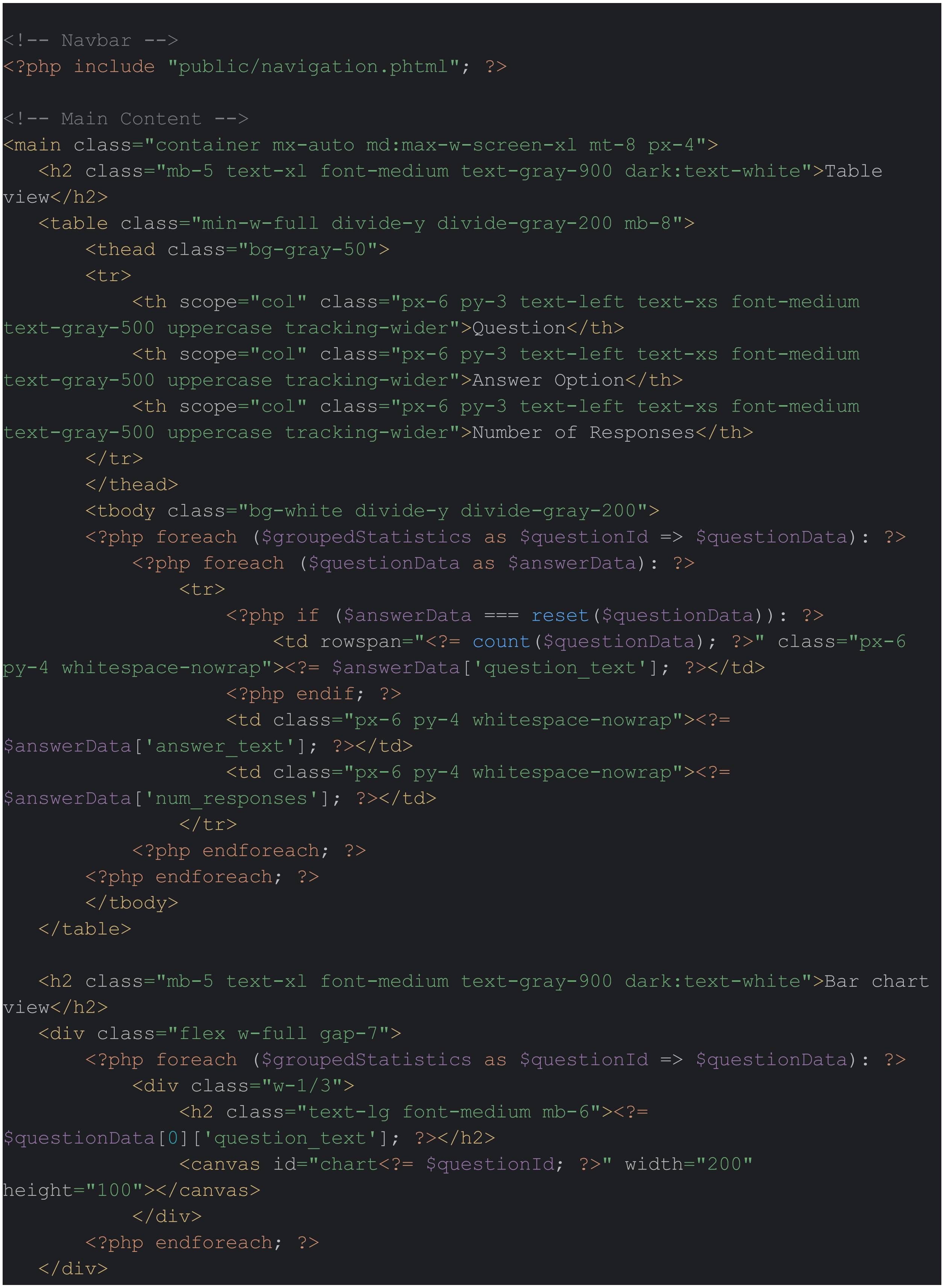
```
                       </button>
                   </div>
                   <p id="modalMessage" class="mt-4"></p>
               </div>
           </div>
       <?php endif; ?>

<!--        --><?php //= strpos($source, 'edit_city') ?>
       <?php if (strpos($source, 'edit_city') !== false) : ?>
           <h2 class="text-2xl font-bold mb-4">Edit City</h2>
           <form action="../../services/admin/edit-city.php" class="py-3
md:py-4" method="post">
               <input name="city_id" value="<?= $city['id'] ?>" class="hidden">
               <div class="rounded-md py-3 md:py-4">
                   <label for="city_name" class="block text-md font-medium
text-gray-700 mb-2">Name:</label>
                   <input type="text" id="city_name" name="city_name"
value="<?= $city['name'] ?>" class="min-w-96 px-3 py-2 text-gray-700 border
rounded-md focus:outline-none focus:border-indigo-500" required>
               </div>
               <div class="rounded-md py-3 md:py-4">
                   <label for="countryDdl" class="block text-md font-medium
text-gray-700 mb-2">Country</label>
                   <select id="countryDdl" name="country_id" class="min-w-96
px-3 py-2 border rounded-md focus:outline-none focus:border-indigo-500">
                       <!-- Options will be populated by JavaScript -->
                   </select>
               </div>
               <button type="submit" class="my-3 md:my-4 bg-indigo-500
hover:bg-indigo-700 text-white font-bold py-2 px-4 rounded focus:outline-none
focus:shadow-outline">Edit City</button>
               <p class="py-1"><?php echo getFlashMessage('edit-city-response')
?></p>
           </form>
       <?php endif; ?>

       <?php if (strpos($source, 'edit_country') !== false) : ?>
           <h2 class="text-2xl font-bold mb-4">Edit Country</h2>
           <form action="../../services/admin/edit-country.php" class="py-3
md:py-4" method="post">
               <input name="country_id" value="<?= $country['id'] ?>"
class="hidden">
               <div class="rounded-md py-3 md:py-4">
                   <label for="country_name" class="block text-md font-medium
text-gray-700 mb-2">Name:</label>
                   <input type="text" id="country_name" name="country_name"
value="<?= $country['name'] ?>" class="min-w-96 px-3 py-2 text-gray-700 border
rounded-md focus:outline-none focus:border-indigo-500" >
               </div>
```

```
                <div class="rounded-md py-3 md:py-4">
                    <label for="country_short_name" class="block text-md
font-medium text-gray-700 mb-2">Short Name:</label>
                    <input type="text" id="country_short_name"
name="country_short_name" value="<?= $country['short_name'] ?>" class="min-w-96
px-3 py-2 text-gray-700 border rounded-md focus:outline-none
focus:border-indigo-500" >
                    <p class="text-sm md:text-base mb-3 mt-2
text-gray-600">Important for flag images.</p>
                </div>

                <button type="submit" class="my-3 md:my-4 bg-indigo-500
hover:bg-indigo-700 text-white font-bold py-2 px-4 rounded focus:outline-none
focus:shadow-outline">
                    Edit Country</button>
                <p class="py-1"><?php echo
getFlashMessage('edit-country-response')?></p>
            </form>
        <?php endif; ?>



    </div>
</div>

<script>
    // JavaScript to toggle dropdowns
    document.addEventListener('DOMContentLoaded', function() {
        const toggleButtons = document.querySelectorAll('.toggle-dropdown');
        toggleButtons.forEach(button => {
            button.addEventListener('click', function() {
                if(button.classList.contains('bg-gray-700')) {
                    button.classList.remove('bg-gray-700');
                } else {
                    button.classList.add('bg-gray-700');
                }

                const targetId = this.getAttribute('data-target');
                const targetDropdown = document.getElementById(targetId);
                targetDropdown.classList.toggle('hidden');
            });

        });
    });
</script>
<script>
    // JavaScript to toggle sidebar on small screens
    document.addEventListener('DOMContentLoaded', function() {
        const container = document.getElementById('container')
        const toggleSidebar = document.getElementById('toggleSidebar');
        const sidebar = document.getElementById('sidebar');

        toggleSidebar.addEventListener('click', function() {
            sidebar.classList.toggle('hidden');
            container.classList.toggle('flex')
            sidebar.classList.toggle('h-screen');
            sidebar.classList.toggle('w-full');
        });
    });
</script>

<script>
    async function getCountries() {
        try {
            const response = await
fetch('../../services/property/get-countries.php');
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            const countries = await response.json();
            renderCountriesDdl(countries);
        } catch (error) {
            console.error('Error fetching data:', error);
        }
    }

    function renderCountriesDdl(countries) {
        const countryDdl = document.getElementById('countryDdl');
        countryDdl.innerHTML = '';
        const currentCountryId = '<?= $city['country_id'] ?>';

        countries.forEach((country) => {
            const selected = country.id == currentCountryId ? 'selected' : '';
            countryDdl.innerHTML += `<option value="${country.id}"
${selected}>${country.name}</option>`;
        });
    }

    if('<?php echo ($source == 'add_city') ?>' || '<?php echo ($source ==
'edit_city') ?>') {
        getCountries()
    }

    function decodeHtml(html) {
        var txt = document.createElement("textarea");
        txt.innerHTML = html;
        return txt.value;
    }
    if ('<?php echo ($source == 'show_messages') ?>') {
        function openModal(messageTitle, messageContent) {
            console.log(messageContent);
            const decodedMessage = decodeHtml(messageContent);
            const modal = document.getElementById('messageModal');
            const modalTitle = document.getElementById('modalTitle');
            const modalMessage = document.getElementById('modalMessage');

            // Update the modal title with the provided message title
            modalTitle.textContent = messageTitle;

            // Set the decoded message as the inner HTML of the modal message
            modalMessage.innerHTML = decodedMessage;

            // Remove the 'hidden' class to display the modal
            modal.classList.remove('hidden');

            // Add event listener to close modal when clicking outside the modal
content
            setTimeout(() => {
                window.addEventListener('click', outsideClickListener);
            }, 0.1);
        }

        function closeModal() {
            const modal = document.getElementById('messageModal');
            modal.classList.add('hidden');

            // Remove event listener when modal is closed
            window.removeEventListener('click', outsideClickListener);
        }

        function outsideClickListener(event) {
            const modal = document.getElementById('messageModal');
            const modalContent = document.querySelector('.modal-content'); //
Modal content element
            if (!modalContent.contains(event.target)) {
                closeModal();
            }
        }

        // Optional: Add a close button inside the modal content to close the
modal
        document.querySelector('.fa-x').addEventListener('click', closeModal);
    }


</script>
```

```
<script src="https://cdn.tailwindcss.com/"></script>

</body>

</html>
```

Statistics.phtml

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

<?php require "public/head.phtml"; ?>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<?php
// Assuming $statistics contains the fetched data from your database query

// Group the statistics by question_id for easier processing
$groupedStatistics = [];
foreach ($statistics as $row) {
   $questionId = $row['question_id'];
   if (!isset($groupedStatistics[$questionId])) {
       $groupedStatistics[$questionId] = [];
   }
   $groupedStatistics[$questionId][] = $row;
}


// Prepare arrays to hold labels, data, and colors for the chart
$labels = [];
$data = [];
$colors = [];

// Iterate through grouped statistics to extract data
foreach ($groupedStatistics as $questionId => $questionData) {
   $questionText = $questionData[0]['question_text']; // Assuming all answers
in a group have the same question text
   $labels[] = $questionText;

   foreach ($questionData as $answerData) {
       $data[] = $answerData['num_responses'];
       // You can choose different colors for each answer option if needed
       $colors[] = 'rgb(57, 73, 171, 0.7)'; // Example color
   }
}
?>

<body class="mb-8">
```



```
<!-- Navbar -->
<?php include "public/navigation.phtml"; ?>

<!-- Main Content -->
<main class="container mx-auto md:max-w-screen-xl mt-8 px-4">
   <h2 class="mb-5 text-xl font-medium text-gray-900 dark:text-white">Table view</h2>
   <table class="min-w-full divide-y divide-gray-200 mb-8">
       <thead class="bg-gray-50">
       <tr>
           <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Question</th>
           <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Answer Option</th>
           <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Number of Responses</th>
       </tr>
       </thead>
       <tbody class="bg-white divide-y divide-gray-200">
       <?php foreach ($groupedStatistics as $questionId => $questionData): ?>
           <?php foreach ($questionData as $answerData): ?>
               <tr>
                   <?php if ($answerData === reset($questionData)): ?>
                       <td rowspan="<?= count($questionData); ?>" class="px-6 py-4 whitespace-nowrap"><?= $answerData['question_text']; ?></td>
                   <?php endif; ?>
                   <td class="px-6 py-4 whitespace-nowrap"><?= $answerData['answer_text']; ?></td>
                   <td class="px-6 py-4 whitespace-nowrap"><?= $answerData['num_responses']; ?></td>
               </tr>
           <?php endforeach; ?>
       <?php endforeach; ?>
       </tbody>
   </table>

   <h2 class="mb-5 text-xl font-medium text-gray-900 dark:text-white">Bar chart view</h2>
   <div class="flex w-full gap-7">
       <?php foreach ($groupedStatistics as $questionId => $questionData): ?>
           <div class="w-1/3">
               <h2 class="text-lg font-medium mb-6"><?= $questionData[0]['question_text']; ?></h2>
               <canvas id="chart<?= $questionId; ?>" width="200" height="100"></canvas>
           </div>
       <?php endforeach; ?>
   </div>
```

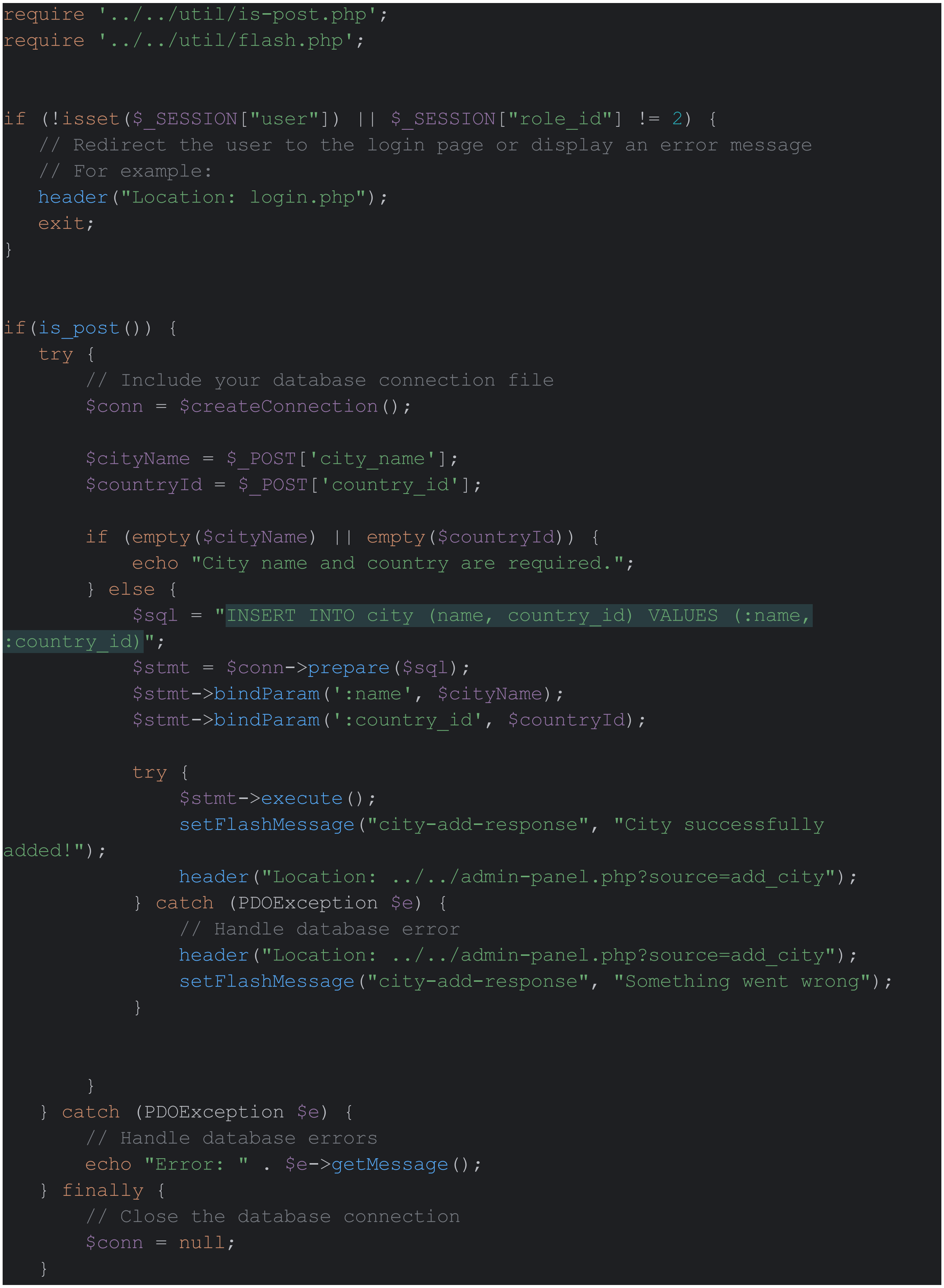
```
</main>
</body>

<script>
   // JavaScript code to create the chart using Chart.js
   document.addEventListener('DOMContentLoaded', function() {
       <?php foreach ($groupedStatistics as $questionId => $questionData): ?>
       var ctx<?= $questionId; ?> = document.getElementById('chart<?=
$questionId; ?>').getContext('2d');
       var myChart<?= $questionId; ?> = new Chart(ctx<?= $questionId; ?>, {
           type: 'bar',
           data: {
               labels: <?= json_encode(array_column($questionData,
'answer_text')); ?>,
               datasets: [{
                   label: 'Number of Responses',
                   data: <?= json_encode(array_column($questionData,
'num_responses')); ?>,
                   backgroundColor: <?= json_encode($colors); ?>,
                   borderWidth: 1
               }]
           },
           options: {
               scales: {
                   yAxes: [{
                       ticks: {
                           beginAtZero: true
                       }
                   }]
               }
           }
       });
       <?php endforeach; ?>
   });
</script>
</html>
```

Conn.php

```
<?php

//class Connection {
//    private static $conn = null;
//
//    public $connection = null;
//    private function __construct() {
//        // Database credentials
//        $servername = "localhost"; // Change this to your database server address
//        $username = "root"; // Change this to your database username
//        $password = ""; // Change this to your database password
//        $database = "sevorealestate"; // Change this to your database name
//
//        try {
//            // Create connection
//            $conn = new PDO("mysql:host=$servername;dbname=$database", $username, $password);
//            // Set PDO to throw exceptions on error
//            $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
//
//            // Additional PDO configurations
//            $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
//
//            // Echo a message if connected successfully
//            // echo "Connected successfully";
//        } catch(PDOException $e) {
//            // Handle connection error
//            die("Connection failed: " . $e->getMessage());
//        }
//
//        $this->connection = $conn;
//    }
//
//    public static function getConnection() {
//        if (is_null(self::$conn)) {
//            self::$conn = new self();
//        }
//
//        return self::$conn;
//    }
//
//    public function __call($method, $args) {
//        if (method_exists($this->connection, $method)) {
//            return call_user_func_array(array($this->connection, $method), $args);
//        } else {
//            return call_user_func(array($this, $method), $args);
//        }
//    }
//
//    public function __get($property) {
//        if (property_exists($this->connection, $property)) {
//            return $this->connection->$property;
//         } else {
//             return $this->$property;
//         }
//     }
//}
//
//$conn = Connection::getConnection();

$_conn = null;

$createConnection = function() use ($_conn) {
    if ($_conn) {
        return $_conn;
    }
    // Database credentials
    $servername = "db"; // Change this to your database server address
    $username = "root"; // Change this to your database username
    $password = "root"; // Change this to your database password
    $database = "sevorealestate"; // Change this to your database name

    $conn = null;
    try {
        // Create connection
        $conn = new PDO("mysql:host=$servername;dbname=$database", $username,
$password);
        // Set PDO to throw exceptions on error
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        // Additional PDO configurations
        $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);

        // Echo a message if connected successfully
        // echo "Connected successfully";
    } catch(PDOException $e) {
        // Handle connection error
        die("Connection failed: " . $e->getMessage());
    }

    $_conn = $conn;
    return $_conn;
}
?>
```

Add-city.php

```
<?php

session_start();
require '../../config/conn.php';
```



```
require '../../util/is-post.php';
require '../../util/flash.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}


if(is_post()) {
   try {
       // Include your database connection file
       $conn = $createConnection();

       $cityName = $_POST['city_name'];
       $countryId = $_POST['country_id'];

       if (empty($cityName) || empty($countryId)) {
           echo "City name and country are required.";
       } else {
           $sql = "INSERT INTO city (name, country_id) VALUES (:name,
:country_id)";
           $stmt = $conn->prepare($sql);
           $stmt->bindParam(':name', $cityName);
           $stmt->bindParam(':country_id', $countryId);

           try {
               $stmt->execute();
               setFlashMessage("city-add-response", "City successfully
added!");
               header("Location: ../../admin-panel.php?source=add_city");
           } catch (PDOException $e) {
               // Handle database error
               header("Location: ../../admin-panel.php?source=add_city");
               setFlashMessage("city-add-response", "Something went wrong");
           }


       }
   } catch (PDOException $e) {
       // Handle database errors
       echo "Error: " . $e->getMessage();
   } finally {
       // Close the database connection
       $conn = null;
   }
}
```

Add-country.php

```
<?php

session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}


if(is_post()) {
   try {
       // Include your database connection file
       $conn = $createConnection();

       $countryName = $_POST['country_name'];
       $shortName = $_POST['country_short_name'];

       if (empty($countryName) || empty($shortName)) {
           echo "Country name and short name are required.";
       } else {
           $sql = "INSERT INTO country (name, short_name) VALUES (:name,
:short_name)";
           $stmt = $conn->prepare($sql);
           $stmt->bindParam(':name', $countryName);
           $stmt->bindParam(':short_name', $shortName);

           try {
               $stmt->execute();
               setFlashMessage("country-add-response", "Country successfully
added!");
               header("Location: ../../admin-panel.php?source=add_country");
           } catch (PDOException $e) {
               // Handle database error
               header("Location: ../../admin-panel.php?source=add_country");
               setFlashMessage("country-add-response", "Something went wrong");
           }
```

```
        }
    } catch (PDOException $e) {
        // Handle database errors
        echo "Error: " . $e->getMessage();
    } finally {
        // Close the database connection
        $conn = null;
    }
}
```

Delete-country.php

```
<?php

session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';

if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    // Redirect the user to the login page or display an error message
    // For example:
    header("Location: login.php");
    exit;
}

if (is_post()) {
    if (isset($_POST['city_id'])) {
        $cityId = $_POST['city_id'];

        try {
            $conn = $createConnection();

            // Prepare and execute SQL statement to delete the city
            $stmt = $conn->prepare("DELETE FROM city WHERE id = :city_id");
            $stmt->bindParam(':city_id', $cityId);
            $stmt->execute();

            // Set success flash message
            setFlashMessage('delete-city-response', 'City deleted
successfully.');

            // Redirect to a page indicating successful deletion
            header("Location: ../../admin-panel.php?source=show_cities");
            exit;
        } catch (PDOException $e) {
            // Handle database errors
            setFlashMessage('error', 'Error deleting city: ' .
$e->getMessage());
```

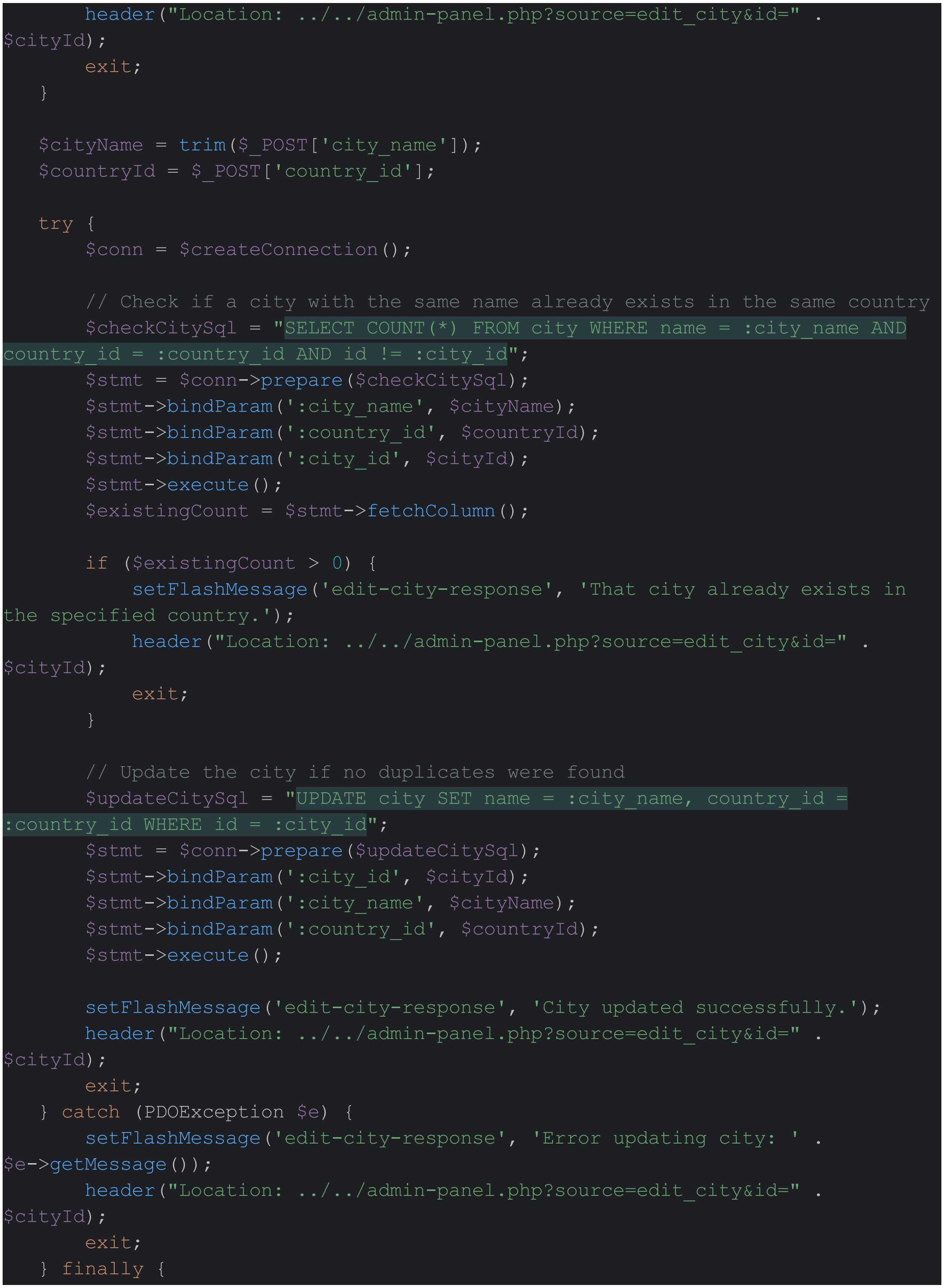
```php
            header("Location: ../../admin-panel.php?source=show_cities");
            exit;
        } finally {
            // Close the database connection
            $conn = null;
        }
    } else {
        // If city_id is not provided, redirect with an error message
        setFlashMessage('delete-city-response', 'City ID not provided.');
        header("Location: ../../admin-panel.php?source=show_cities");
        exit;
    }
} else {
    // If it's not a POST request, redirect with an error message
    setFlashMessage('delete-city-response', 'Invalid request.');
    header("Location: ../../admin-panel.php?source=show_cities");
    exit;
}
```

Delete-message.php

```php
<?php

session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';

if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    // Redirect the user to the login page or display an error message
    // For example:
    header("Location: login.php");
    exit;
}

if (is_post()) {
    if (isset($_POST['message_id'])) {
        $messageId = $_POST['message_id'];

        try {
            $conn = $createConnection();

            // Prepare and execute SQL statement to delete the city
            $stmt = $conn->prepare("DELETE FROM message WHERE id = :message_id");
            $stmt->bindParam(':message_id', $messageId);
            $stmt->execute();

            // Set success flash message
            setFlashMessage('delete-message-response', 'Message deleted
successfully.');

            // Redirect to a page indicating successful deletion
            header("Location: ../../admin-panel.php?source=show_messages");
            exit;
        } catch (PDOException $e) {
            // Handle database errors
            setFlashMessage('error', 'Error deleting messages: ' .
$e->getMessage());
            header("Location: ../../admin-panel.php?source=show_messages");
            exit;
        } finally {
            // Close the database connection
            $conn = null;
        }
    } else {
        // If city_id is not provided, redirect with an error message
        setFlashMessage('delete-message-response', 'Message ID not provided.');
        header("Location: ../../admin-panel.php?source=show_message");
        exit;
    }
} else {
    // If it's not a POST request, redirect with an error message
    setFlashMessage('delete-message-response', 'Invalid request.');
    header("Location: ../../admin-panel.php?source=show_message");
    exit;
}
```

Edit-city.php

```
<?php

session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';

if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    header("Location: login.php");
    exit;
}

$cityId = $_POST['city_id'];

if (is_post()) {
    if (empty($_POST['city_id']) || empty($_POST['city_name']) ||
empty($_POST['country_id'])) {
        setFlashMessage('edit-city-response', 'All fields are required.');
```



```
        header("Location: ../../admin-panel.php?source=edit_city&id=" .
$cityId);
        exit;
    }

    $cityName = trim($_POST['city_name']);
    $countryId = $_POST['country_id'];

    try {
        $conn = $createConnection();

        // Check if a city with the same name already exists in the same country
        $checkCitySql = "SELECT COUNT(*) FROM city WHERE name = :city_name AND
country_id = :country_id AND id != :city_id";
        $stmt = $conn->prepare($checkCitySql);
        $stmt->bindParam(':city_name', $cityName);
        $stmt->bindParam(':country_id', $countryId);
        $stmt->bindParam(':city_id', $cityId);
        $stmt->execute();
        $existingCount = $stmt->fetchColumn();

        if ($existingCount > 0) {
            setFlashMessage('edit-city-response', 'That city already exists in
the specified country.');
            header("Location: ../../admin-panel.php?source=edit_city&id=" .
$cityId);
            exit;
        }

        // Update the city if no duplicates were found
        $updateCitySql = "UPDATE city SET name = :city_name, country_id =
:country_id WHERE id = :city_id";
        $stmt = $conn->prepare($updateCitySql);
        $stmt->bindParam(':city_id', $cityId);
        $stmt->bindParam(':city_name', $cityName);
        $stmt->bindParam(':country_id', $countryId);
        $stmt->execute();

        setFlashMessage('edit-city-response', 'City updated successfully.');
        header("Location: ../../admin-panel.php?source=edit_city&id=" .
$cityId);
        exit;
    } catch (PDOException $e) {
        setFlashMessage('edit-city-response', 'Error updating city: ' .
$e->getMessage());
        header("Location: ../../admin-panel.php?source=edit_city&id=" .
$cityId);
        exit;
    } finally {
        $conn = null;
    }
} else {
    setFlashMessage('edit-city-response', 'Invalid request.');
    header("Location: ../../admin-panel.php?source=edit_city&id=" . $cityId);
    exit;
}

?>
```

Edit-country.php

```
<?php

session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';

if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   header("Location: login.php");
   exit;
}

$countryId = $_POST['country_id'];

if (is_post()) {
   if (empty($_POST['country_id']) || empty($_POST['country_short_name']) ||
empty($_POST['country_name'])) {
       setFlashMessage('edit-country-response', 'All fields are required.');
       header("Location: ../../admin-panel.php?source=edit_country&id=" .
$countryId);
       exit;
   }

   $countryName = trim($_POST['country_name']);
   $countryShortName = trim($_POST['country_short_name']);
   $countryId = $_POST['country_id'];

   try {
       $conn = $createConnection();

       // Check if a country with the same name already exists
       $checkCountrySql = "SELECT COUNT(*) FROM country WHERE name =
:country_name AND id != :country_id";
       $stmt = $conn->prepare($checkCountrySql);
       $stmt->bindParam(':country_name', $countryName);
       $stmt->bindParam(':country_id', $countryId);
       $stmt->execute();
```

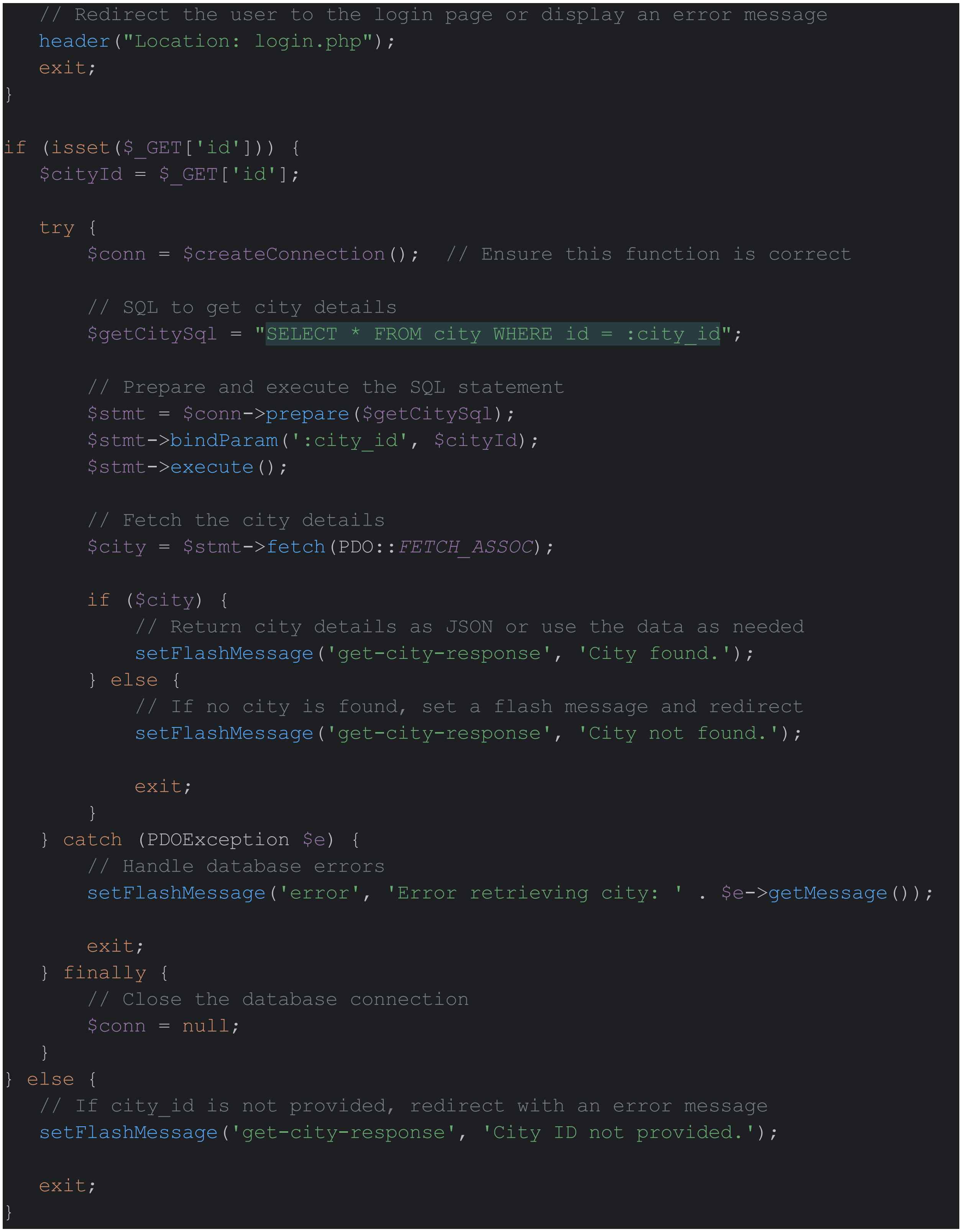
```
        $existingCount = $stmt->fetchColumn();

        if ($existingCount > 0) {
            setFlashMessage('edit-country-response', 'That country already
exists.');
            header("Location: ../../admin-panel.php?source=edit_country&id=" .
$countryId);
            exit;
        }

        // Update the country if no duplicates were found
        $updateCountrySql = "UPDATE country SET name = :country_name, short_name
= :country_short_name WHERE id = :country_id";
        $stmt = $conn->prepare($updateCountrySql);
        $stmt->bindParam(':country_id', $countryId);
        $stmt->bindParam(':country_name', $countryName);
        $stmt->bindParam(':country_short_name', $countryShortName);
        $stmt->execute();

        setFlashMessage('edit-country-response', 'Country updated
successfully.');
        header("Location: ../../admin-panel.php?source=edit_country&id=" .
$countryId);
        exit;
    } catch (PDOException $e) {
        setFlashMessage('edit-country-response', 'Error updating country: ' .
$e->getMessage());
        header("Location: ../../admin-panel.php?source=edit_country&id=" .
$countryId);
        exit;
    } finally {
        $conn = null;
    }
} else {
    setFlashMessage('edit-country-response', 'Invalid request.');
    header("Location: ../../admin-panel.php?source=edit_country&id=" .
$countryId);
    exit;
}

?>
```

Get-cities.php

```
<?php

session_start();
require 'config/conn.php';
require 'util/is-get.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}

try {
   // Include your database connection file
   $conn = $createConnection();

   // Prepare SQL statement to select user settings
   $sql = "SELECT c.id AS city_id, c.name  AS city_name, co.name AS
country_name FROM city c INNER JOIN country co ON c.country_id = co.id";

   // Prepare the statement
   $stmt = $conn->prepare($sql);

   // Bind parameter
   //$stmt->bindParam(":user_id", $userId);

   // Execute the statement
   $stmt->execute();

   // Fetch the user settings
   $cities = $stmt->fetchAll(PDO::FETCH_ASSOC);


//    var_dump($statistics);
} catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
} finally {
   // Close the database connection
   $conn = null;
}
```

Get-city.php

```
<?php

session_start();
require 'config/conn.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
```



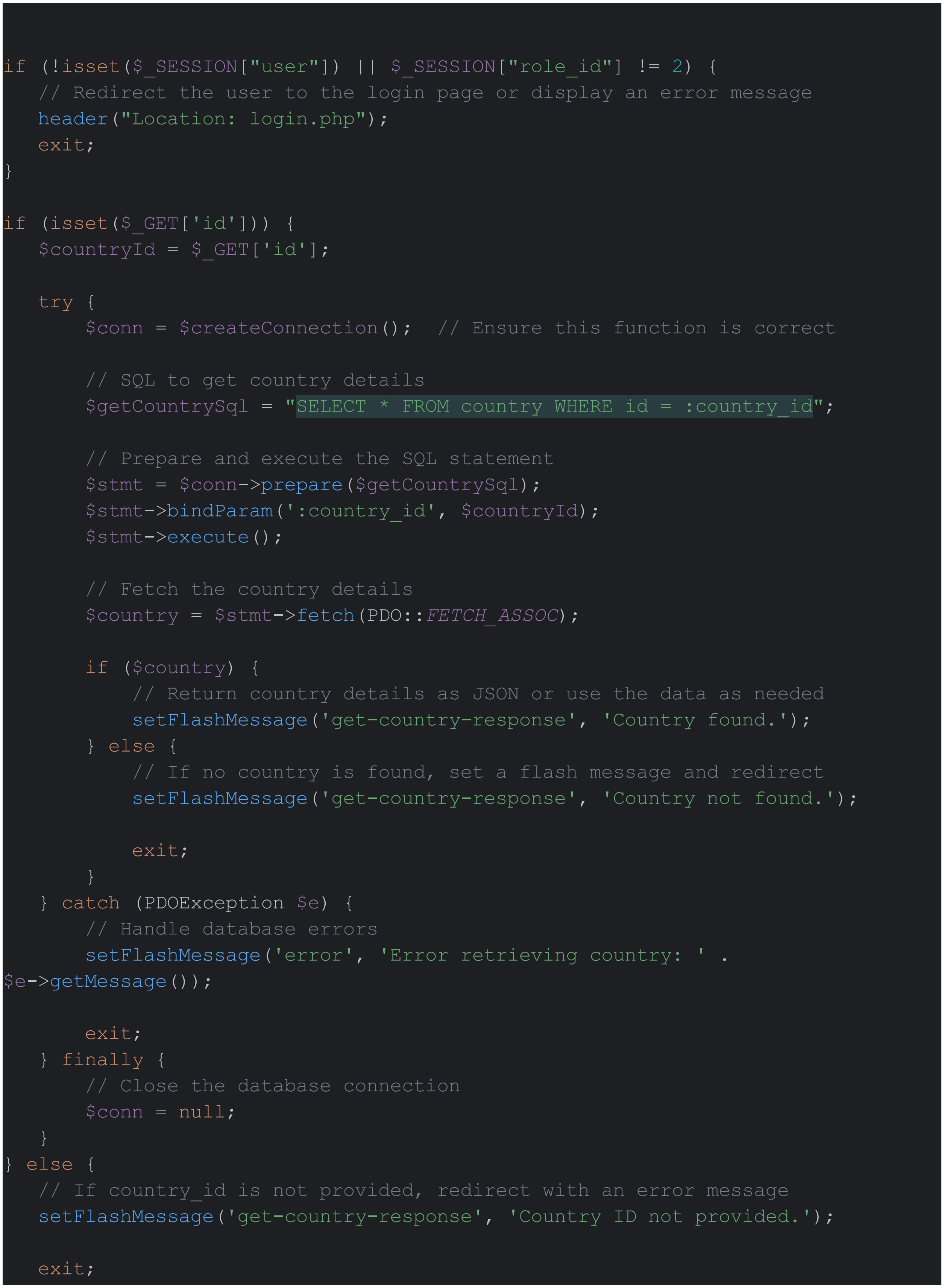
```php
    // Redirect the user to the login page or display an error message
    header("Location: login.php");
    exit;
}

if (isset($_GET['id'])) {
    $cityId = $_GET['id'];

    try {
        $conn = $createConnection();  // Ensure this function is correct

        // SQL to get city details
        $getCitySql = "SELECT * FROM city WHERE id = :city_id";

        // Prepare and execute the SQL statement
        $stmt = $conn->prepare($getCitySql);
        $stmt->bindParam(':city_id', $cityId);
        $stmt->execute();

        // Fetch the city details
        $city = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($city) {
            // Return city details as JSON or use the data as needed
            setFlashMessage('get-city-response', 'City found.');
        } else {
            // If no city is found, set a flash message and redirect
            setFlashMessage('get-city-response', 'City not found.');

            exit;
        }
    } catch (PDOException $e) {
        // Handle database errors
        setFlashMessage('error', 'Error retrieving city: ' . $e->getMessage());

        exit;
    } finally {
        // Close the database connection
        $conn = null;
    }
} else {
    // If city_id is not provided, redirect with an error message
    setFlashMessage('get-city-response', 'City ID not provided.');

    exit;
}
```

Get-countries.php

```php
<?php

session_start();
require 'config/conn.php';
require 'util/is-get.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    // Redirect the user to the login page or display an error message
    // For example:
    header("Location: login.php");
    exit;
}

try {
    // Include your database connection file
    $conn = $createConnection();

    // Prepare SQL statement to select user settings
    $sql = "SELECT * FROM country";

    // Prepare the statement
    $stmt = $conn->prepare($sql);

    // Bind parameter
    //$stmt->bindParam(":user_id", $userId);

    // Execute the statement
    $stmt->execute();

    // Fetch the user settings
    $countries = $stmt->fetchAll(PDO::FETCH_ASSOC);


//    var_dump($statistics);
} catch (PDOException $e) {
    // Handle database errors
    echo "Error: " . $e->getMessage();
} finally {
    // Close the database connection
    $conn = null;
}
```

Get-country.php

```php
<?php

session_start();
require 'config/conn.php';
```



```
if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    // Redirect the user to the login page or display an error message
    header("Location: login.php");
    exit;
}

if (isset($_GET['id'])) {
    $countryId = $_GET['id'];

    try {
        $conn = $createConnection();  // Ensure this function is correct

        // SQL to get country details
        $getCountrySql = "SELECT * FROM country WHERE id = :country_id";

        // Prepare and execute the SQL statement
        $stmt = $conn->prepare($getCountrySql);
        $stmt->bindParam(':country_id', $countryId);
        $stmt->execute();

        // Fetch the country details
        $country = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($country) {
            // Return country details as JSON or use the data as needed
            setFlashMessage('get-country-response', 'Country found.');
        } else {
            // If no country is found, set a flash message and redirect
            setFlashMessage('get-country-response', 'Country not found.');

            exit;
        }
    } catch (PDOException $e) {
        // Handle database errors
        setFlashMessage('error', 'Error retrieving country: ' .
$e->getMessage());

        exit;
    } finally {
        // Close the database connection
        $conn = null;
    }
} else {
    // If country_id is not provided, redirect with an error message
    setFlashMessage('get-country-response', 'Country ID not provided.');

    exit;
}
```

Get-messages.php

```
<?php

session_start();
require 'config/conn.php';
require 'util/is-get.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}

try {
   // Include your database connection file
   $conn = $createConnection();

   // Prepare SQL statement to select user settings
   $sql = "SELECT * FROM message";

   // Prepare the statement
   $stmt = $conn->prepare($sql);

   // Bind parameter
   //$stmt->bindParam(":user_id", $userId);

   // Execute the statement
   $stmt->execute();

   // Fetch the user settings
   $messages = $stmt->fetchAll(PDO::FETCH_ASSOC);


//    var_dump($statistics);
} catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
} finally {
   // Close the database connection
   $conn = null;
}
```

Get-statistics.php

```
<?php

session_start();
require 'config/conn.php';
require 'util/is-get.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}

try {
   // Include your database connection file
   $conn = $createConnection();

   // Prepare SQL statement to select user settings
   $sql = "SELECT * FROM message";

   // Prepare the statement
   $stmt = $conn->prepare($sql);

   // Bind parameter
   //$stmt->bindParam(":user_id", $userId);

   // Execute the statement
   $stmt->execute();

   // Fetch the user settings
   $messages = $stmt->fetchAll(PDO::FETCH_ASSOC);


//    var_dump($statistics);
} catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
} finally {
   // Close the database connection
   $conn = null;
}
```

Get-users.php

```
<?php

session_start();
```

```
require 'config/conn.php';
require 'util/is-get.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
   // Redirect the user to the login page or display an error message
   // For example:
   header("Location: login.php");
   exit;
}

try {
   // Include your database connection file
   $conn = $createConnection();

   // Prepare SQL statement to select user settings
   $sql = "SELECT * FROM user";

   // Prepare the statement
   $stmt = $conn->prepare($sql);

   // Bind parameter
   //$stmt->bindParam(":user_id", $userId);

   // Execute the statement
   $stmt->execute();

   // Fetch the user settings
   $users = $stmt->fetchAll(PDO::FETCH_ASSOC);


//    var_dump($statistics);
} catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
} finally {
   // Close the database connection
   $conn = null;
}
```

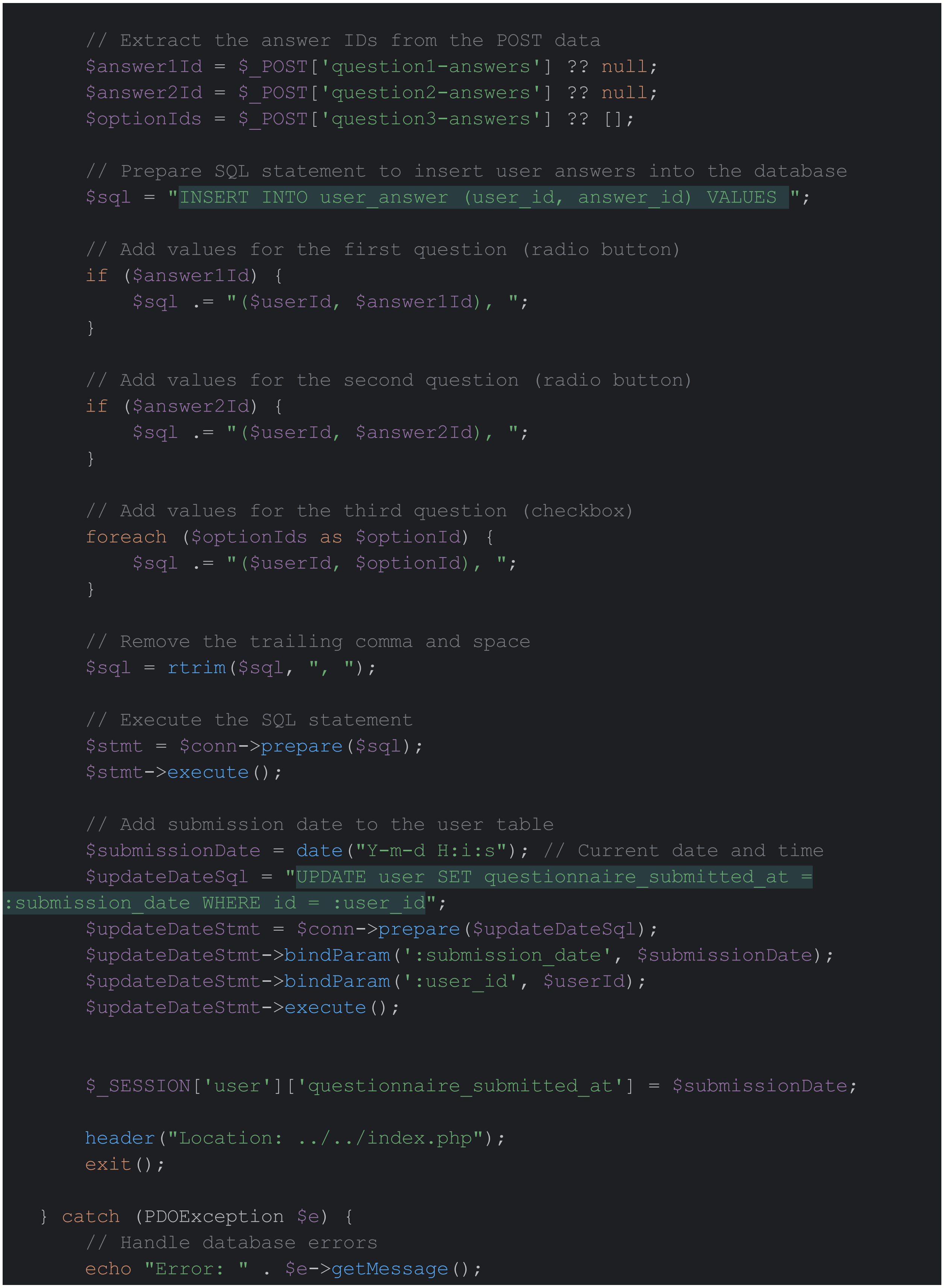
Get-questionaire.php

```
<?php
require 'config/conn.php';
session_start();
// Assuming you have a database connection established already


   $conn = null;
```

```
    try {
        // Connect to your database
        $conn = $createConnection();
        // Fetch question data
        $queryQuestions = "SELECT id, text, type FROM question";
        $stmtQuestions = $conn->prepare($queryQuestions);
        $stmtQuestions->execute();
        $questions = $stmtQuestions->fetchAll(PDO::FETCH_ASSOC);

        // Fetch answers data with image names
        $answers = [];
        foreach ($questions as $question) {
            $queryAnswers = "SELECT id, text, image_name FROM answer WHERE question_id = :question_id";
            $stmtAnswers = $conn->prepare($queryAnswers);
            $stmtAnswers->bindParam(':question_id', $question['id']);
            $stmtAnswers->execute();
            $answers[$question['id']] = $stmtAnswers->fetchAll(PDO::FETCH_ASSOC);
        }

        // Now $questions contains all questions, and $answers is an associative array where keys are question IDs
        // and values are arrays of answers associated with each question, including image names if available.

    } catch (PDOException $e) {
        // Handle database errors
        echo "Error: " . $e->getMessage();
    }

?>
```

Submit-questionaire.php

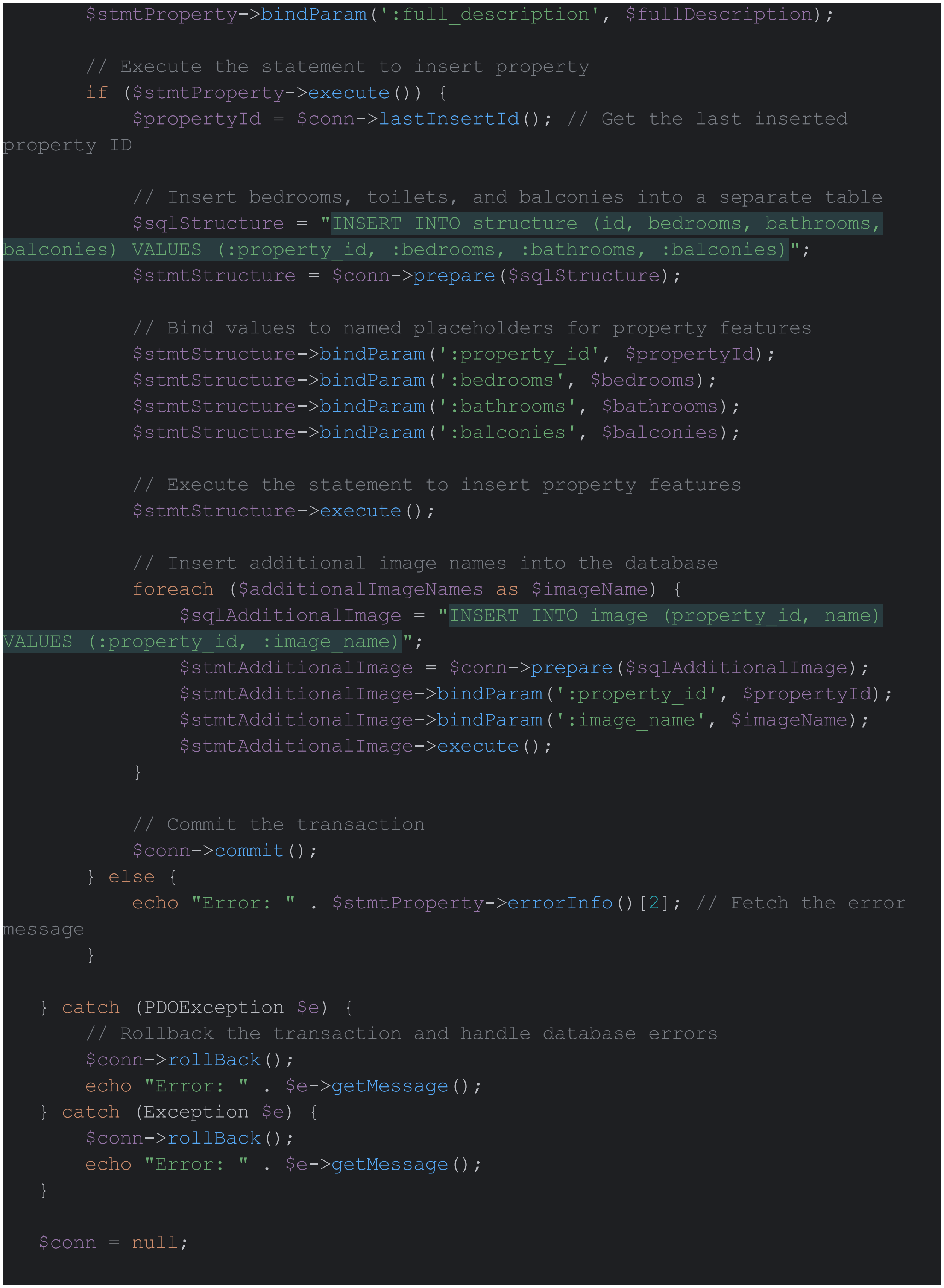
```
<?php
require '../../config/conn.php';
require '../../util/is-post.php';
session_start();

if (is_post()) {
    $conn = null;

    try {
        // Connect to your database
        $conn = $createConnection();
        $userId = $_SESSION['user']['id'];
```



```
        // Extract the answer IDs from the POST data
        $answer1Id = $_POST['question1-answers'] ?? null;
        $answer2Id = $_POST['question2-answers'] ?? null;
        $optionIds = $_POST['question3-answers'] ?? [];

        // Prepare SQL statement to insert user answers into the database
        $sql = "INSERT INTO user_answer (user_id, answer_id) VALUES ";

        // Add values for the first question (radio button)
        if ($answer1Id) {
            $sql .= "($userId, $answer1Id), ";
        }

        // Add values for the second question (radio button)
        if ($answer2Id) {
            $sql .= "($userId, $answer2Id), ";
        }

        // Add values for the third question (checkbox)
        foreach ($optionIds as $optionId) {
            $sql .= "($userId, $optionId), ";
        }

        // Remove the trailing comma and space
        $sql = rtrim($sql, ", ");

        // Execute the SQL statement
        $stmt = $conn->prepare($sql);
        $stmt->execute();

        // Add submission date to the user table
        $submissionDate = date("Y-m-d H:i:s"); // Current date and time
        $updateDateSql = "UPDATE user SET questionnaire_submitted_at =
:submission_date WHERE id = :user_id";
        $updateDateStmt = $conn->prepare($updateDateSql);
        $updateDateStmt->bindParam(':submission_date', $submissionDate);
        $updateDateStmt->bindParam(':user_id', $userId);
        $updateDateStmt->execute();


        $_SESSION['user']['questionnaire_submitted_at'] = $submissionDate;

        header("Location: ../../index.php");
        exit();

    } catch (PDOException $e) {
        // Handle database errors
        echo "Error: " . $e->getMessage();
    }
}
?>
```

services/property/addproperty.php

```
<?php
require '../../config/conn.php';
require '../../util/is-post.php';
session_start();

if (is_post()) {
    $conn = null;
    try {
        $name = $_POST["name"];
        $type_id = $_POST["type"];
        $price = $_POST["price"];
        $size = $_POST["size"];
        $description = $_POST["description"];
        $bedrooms = $_POST["bedrooms"];
        $bathrooms = $_POST["bathrooms"];
        $balconies = $_POST["balconies"];
        $city_id = $_POST["city_id"]; // Using city_id instead of country and city
        $address = $_POST["address"];
        $zip = $_POST["zip"];


        $fullDescription = $_POST["full_description"];

        $userId = $_SESSION['user']['id'];

        // Connect to your database
        $conn = $createConnection();

        // Start a transaction
        $conn->beginTransaction();

        // Handle file upload for main image
        $mainImageName = null;
        if (isset($_FILES['main-image'])) {
            $targetDirectory = "../../store/property/main-images/"; // Specify your folder path

            $extension = pathinfo($_FILES["main-image"]["name"], PATHINFO_EXTENSION);
            $mainImageName = $_SESSION['user']['id'] . "_" . generateRandomString() . "." . $extension;

            $targetFilePath = $targetDirectory . $mainImageName;
            if (move_uploaded_file($_FILES["main-image"]["tmp_name"],
$targetFilePath)) {
                // File uploaded successfully
            } else {
                throw new Exception("Image uploading failed");
            }
        }

        // Handle file upload for additional images
        $additionalImageNames = [];
        if (isset($_FILES['additional-images'])) {
            $targetDirectory = "../../store/property/images/"; // Specify your
folder path
            $additionalImages = $_FILES['additional-images'];
            foreach ($additionalImages['tmp_name'] as $key => $tmpName) {
                $extension = pathinfo($additionalImages["name"][$key],
PATHINFO_EXTENSION);
                $imageName = $_SESSION['user']['id'] . "_" .
generateRandomString() . "." . $extension;
                $targetFilePath = $targetDirectory . $imageName;
                if (move_uploaded_file($tmpName, $targetFilePath)) {
                    $additionalImageNames[] = $imageName;
                } else {
                    throw new Exception("Image uploading failed");
                }
            }
        }

        // Prepare SQL statement to insert property into the database
        $sqlProperty = "INSERT INTO property (name, type_id, price, description,
image_name, city_id, address, zip_code, size, user_id, full_description) VALUES
(:name, :type_id, :price, :description, :image_name, :city_id, :address, :zip,
:size, :user_id, :full_description)";
        $stmtProperty = $conn->prepare($sqlProperty);

        // Bind values to named placeholders
        $stmtProperty->bindParam(':name', $name);
        $stmtProperty->bindParam(':type_id', $type_id);
        $stmtProperty->bindParam(':price', $price);
        $stmtProperty->bindParam(':description', $description);
        $stmtProperty->bindParam(':image_name', $mainImageName); // Placeholder
for the main image name
        $stmtProperty->bindParam(':city_id', $city_id); // Bind city_id instead
of country and city
        $stmtProperty->bindParam(':address', $address);
        $stmtProperty->bindParam(':zip', $zip);
        $stmtProperty->bindParam(':size', $size);
        $stmtProperty->bindParam(':user_id', $userId);
```

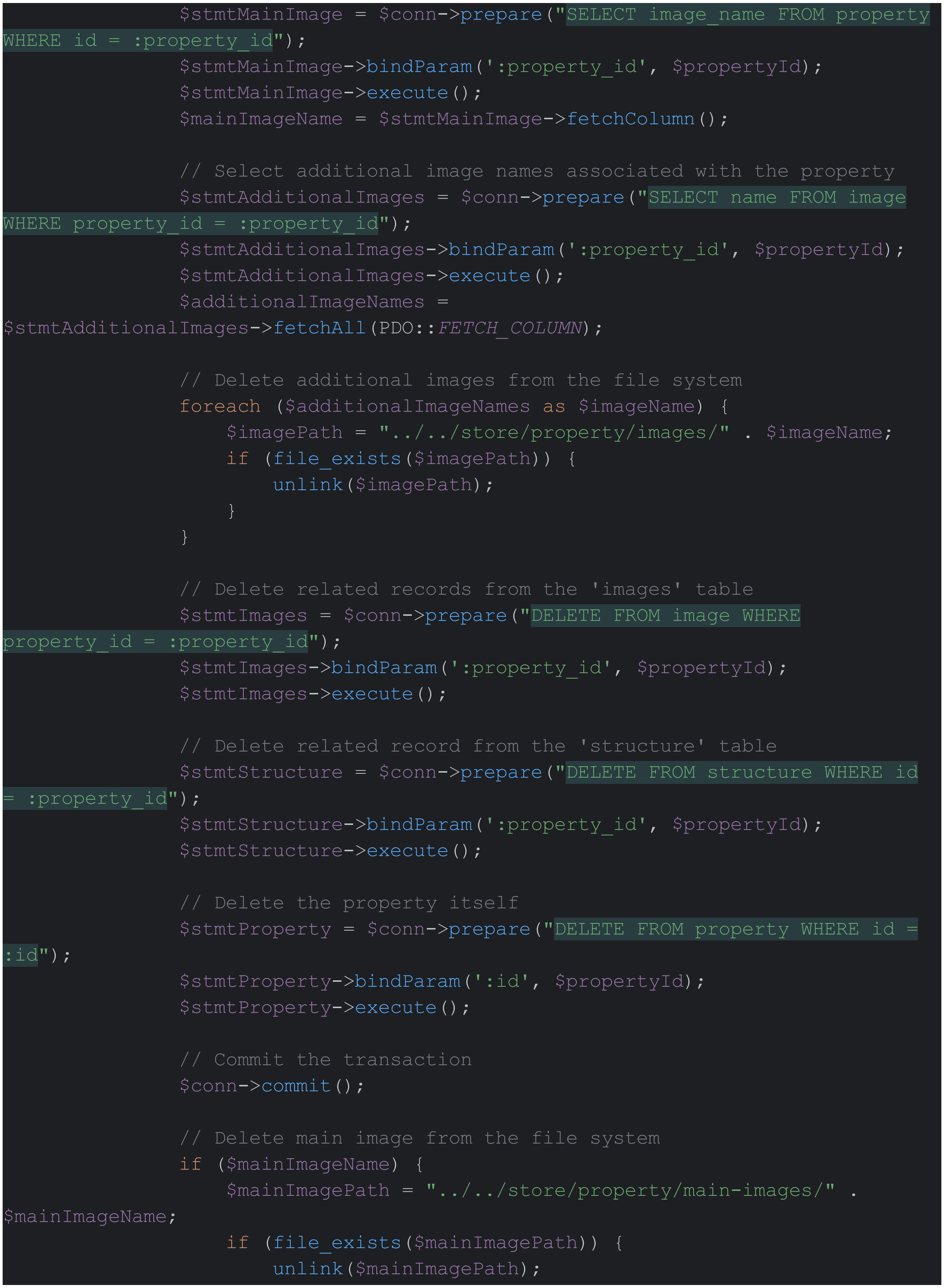


```
        $stmtProperty->bindParam(':full_description', $fullDescription);

        // Execute the statement to insert property
        if ($stmtProperty->execute()) {
            $propertyId = $conn->lastInsertId(); // Get the last inserted
property ID

            // Insert bedrooms, toilets, and balconies into a separate table
            $sqlStructure = "INSERT INTO structure (id, bedrooms, bathrooms,
balconies) VALUES (:property_id, :bedrooms, :bathrooms, :balconies)";
            $stmtStructure = $conn->prepare($sqlStructure);

            // Bind values to named placeholders for property features
            $stmtStructure->bindParam(':property_id', $propertyId);
            $stmtStructure->bindParam(':bedrooms', $bedrooms);
            $stmtStructure->bindParam(':bathrooms', $bathrooms);
            $stmtStructure->bindParam(':balconies', $balconies);

            // Execute the statement to insert property features
            $stmtStructure->execute();

            // Insert additional image names into the database
            foreach ($additionalImageNames as $imageName) {
                $sqlAdditionalImage = "INSERT INTO image (property_id, name)
VALUES (:property_id, :image_name)";
                $stmtAdditionalImage = $conn->prepare($sqlAdditionalImage);
                $stmtAdditionalImage->bindParam(':property_id', $propertyId);
                $stmtAdditionalImage->bindParam(':image_name', $imageName);
                $stmtAdditionalImage->execute();
            }

            // Commit the transaction
            $conn->commit();
        } else {
            echo "Error: " . $stmtProperty->errorInfo()[2]; // Fetch the error
message
        }

    } catch (PDOException $e) {
        // Rollback the transaction and handle database errors
        $conn->rollBack();
        echo "Error: " . $e->getMessage();
    } catch (Exception $e) {
        $conn->rollBack();
        echo "Error: " . $e->getMessage();
    }

    $conn = null;
```

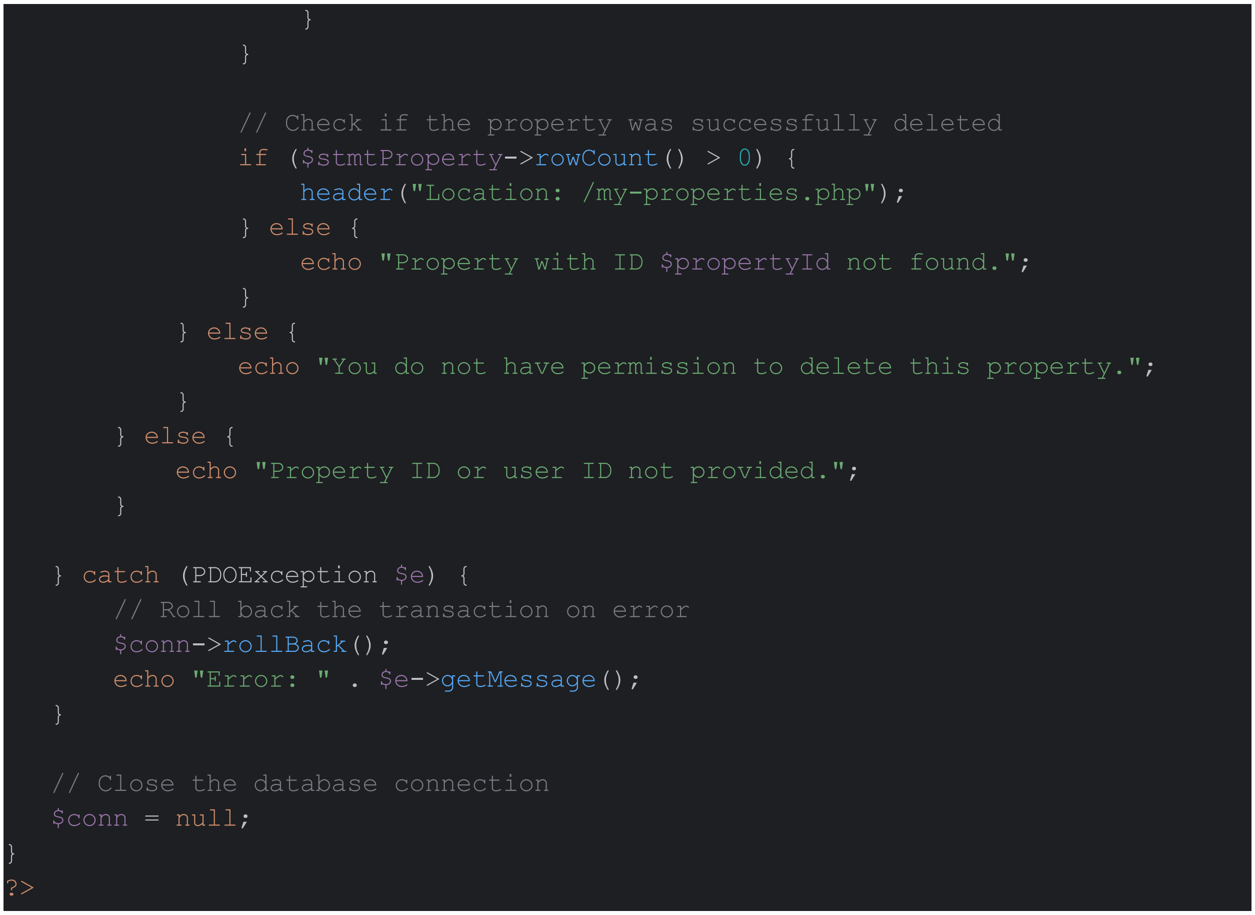
```php
   // Redirect to my properties
   header("Location: ../../my-properties.php");
}

// Function to generate a random string
function generateRandomString($length = 10) {
   $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
   $randomString = '';
   for ($i = 0; $i < $length; $i++) {
       $randomString .= $characters[rand(0, strlen($characters) - 1)];
   }
   return $randomString;
}
?>
```

services/delete-property.php

```php
<?php
require '../../config/conn.php';
require '../../util/is-post.php';
session_start();

if (is_post()) {
   $conn = null;
   try {
       // Check if the property ID and user ID are provided via POST
       if(isset($_POST['property_id']) && isset($_SESSION['user']['id'])) {
           // Retrieve the property ID and user ID from the POST data and
session
           $propertyId = $_POST['property_id'];
           $userId = $_SESSION['user']['id'];

           // Connect to the database
           $conn = $createConnection();

           // Start a transaction
           $conn->beginTransaction();

           // Check if the property belongs to the current user
           $stmtCheckOwnership = $conn->prepare("SELECT COUNT(*) FROM property
WHERE id = :property_id AND user_id = :user_id");
           $stmtCheckOwnership->bindParam(':property_id', $propertyId);
           $stmtCheckOwnership->bindParam(':user_id', $userId);
           $stmtCheckOwnership->execute();
           $propertyCount = $stmtCheckOwnership->fetchColumn();

           if ($propertyCount > 0) {
               // Select main image name associated with the property
```

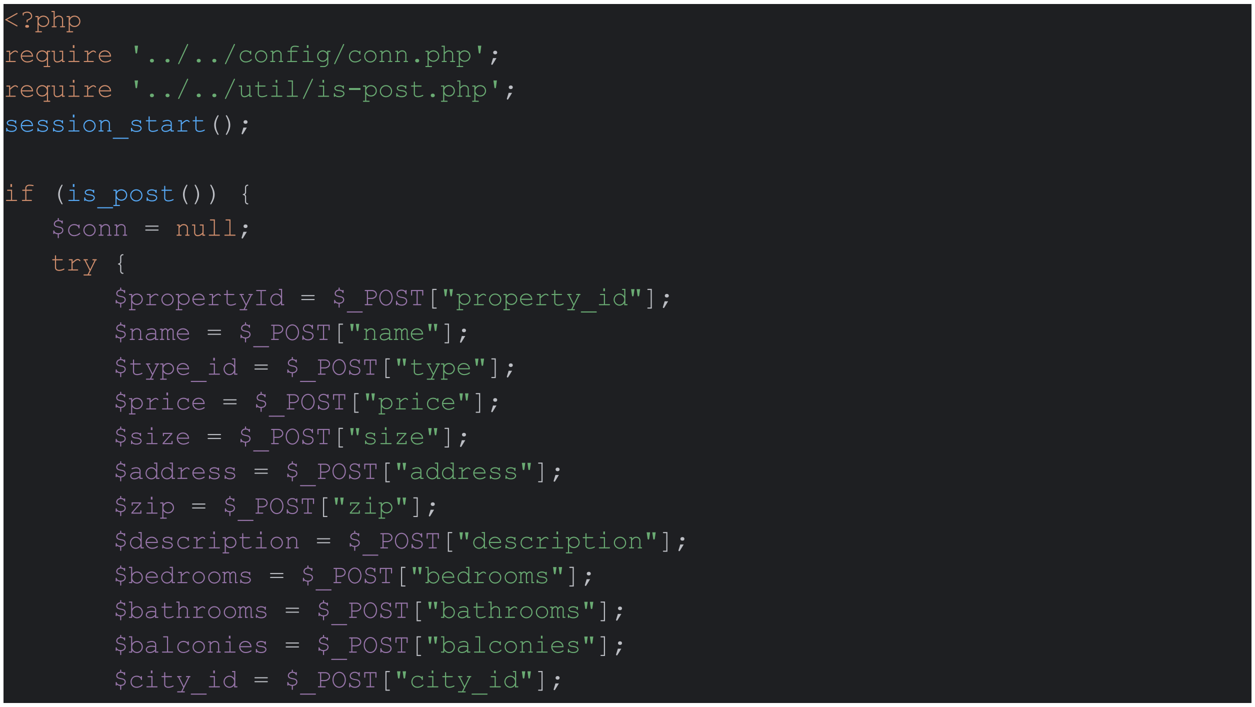


```
                $stmtMainImage = $conn->prepare("SELECT image_name FROM property
WHERE id = :property_id");
                $stmtMainImage->bindParam(':property_id', $propertyId);
                $stmtMainImage->execute();
                $mainImageName = $stmtMainImage->fetchColumn();

                // Select additional image names associated with the property
                $stmtAdditionalImages = $conn->prepare("SELECT name FROM image
WHERE property_id = :property_id");
                $stmtAdditionalImages->bindParam(':property_id', $propertyId);
                $stmtAdditionalImages->execute();
                $additionalImageNames =
$stmtAdditionalImages->fetchAll(PDO::FETCH_COLUMN);

                // Delete additional images from the file system
                foreach ($additionalImageNames as $imageName) {
                    $imagePath = "../../store/property/images/" . $imageName;
                    if (file_exists($imagePath)) {
                        unlink($imagePath);
                    }
                }

                // Delete related records from the 'images' table
                $stmtImages = $conn->prepare("DELETE FROM image WHERE
property_id = :property_id");
                $stmtImages->bindParam(':property_id', $propertyId);
                $stmtImages->execute();

                // Delete related record from the 'structure' table
                $stmtStructure = $conn->prepare("DELETE FROM structure WHERE id
= :property_id");
                $stmtStructure->bindParam(':property_id', $propertyId);
                $stmtStructure->execute();

                // Delete the property itself
                $stmtProperty = $conn->prepare("DELETE FROM property WHERE id =
:id");
                $stmtProperty->bindParam(':id', $propertyId);
                $stmtProperty->execute();

                // Commit the transaction
                $conn->commit();

                // Delete main image from the file system
                if ($mainImageName) {
                    $mainImagePath = "../../store/property/main-images/" .
$mainImageName;
                    if (file_exists($mainImagePath)) {
                        unlink($mainImagePath);
```

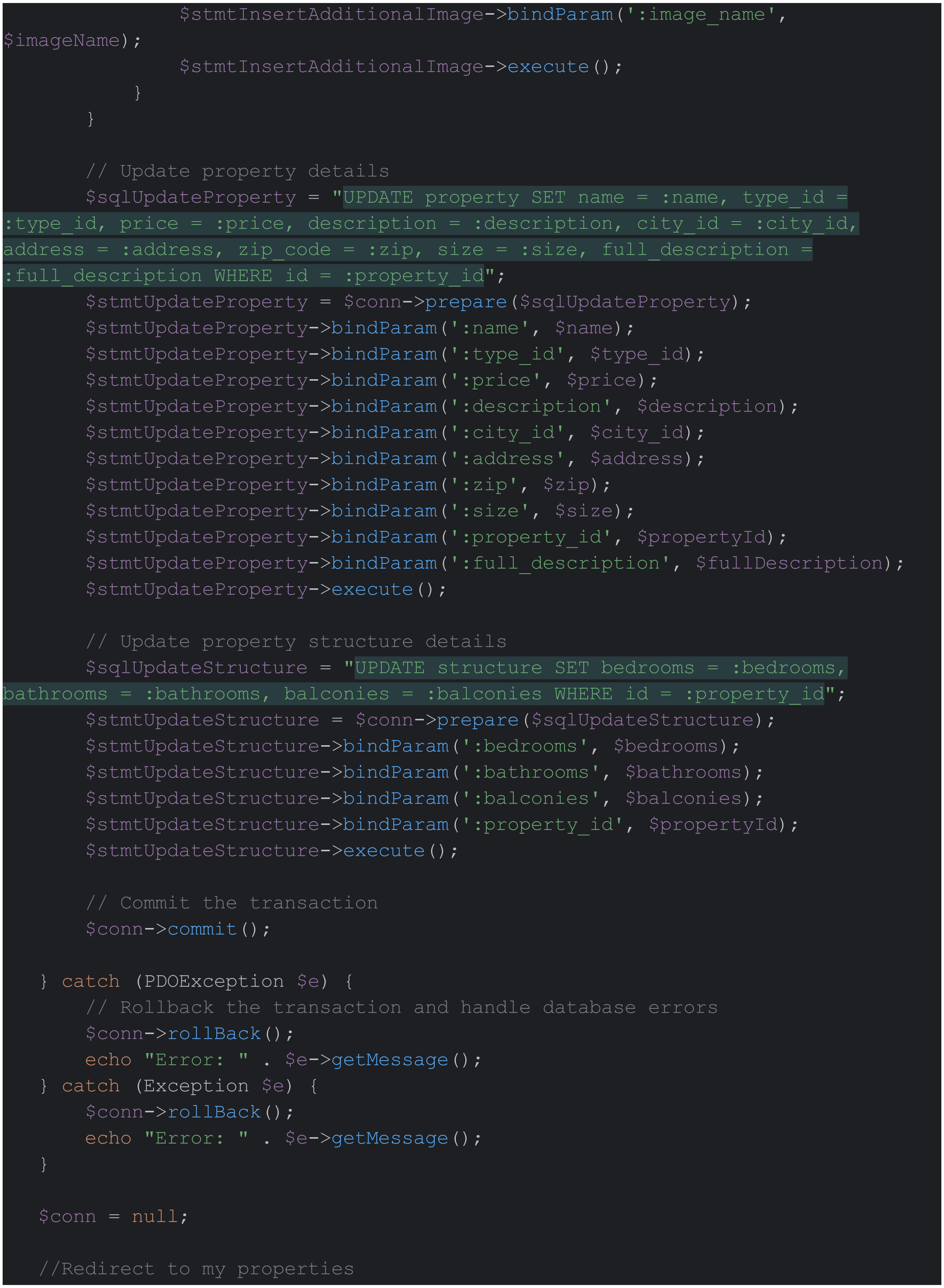


```
                    }
                }

                // Check if the property was successfully deleted
                if ($stmtProperty->rowCount() > 0) {
                    header("Location: /my-properties.php");
                } else {
                    echo "Property with ID $propertyId not found.";
                }
            } else {
                echo "You do not have permission to delete this property.";
            }
        } else {
            echo "Property ID or user ID not provided.";
        }

    } catch (PDOException $e) {
        // Roll back the transaction on error
        $conn->rollBack();
        echo "Error: " . $e->getMessage();
    }

    // Close the database connection
    $conn = null;
}
?>
```

services/property/edit-property.php



```
<?php
require '../../config/conn.php';
require '../../util/is-post.php';
session_start();

if (is_post()) {
    $conn = null;
    try {
        $propertyId = $_POST["property_id"];
        $name = $_POST["name"];
        $type_id = $_POST["type"];
        $price = $_POST["price"];
        $size = $_POST["size"];
        $address = $_POST["address"];
        $zip = $_POST["zip"];
        $description = $_POST["description"];
        $bedrooms = $_POST["bedrooms"];
        $bathrooms = $_POST["bathrooms"];
        $balconies = $_POST["balconies"];
        $city_id = $_POST["city_id"];
        $fullDescription = $_POST["full_description"];

        // Connect to your database
        $conn = $createConnection();

        // Start a transaction
        $conn->beginTransaction();

        // Update main image if a new one is provided
        if (isset($_FILES['main-image']) && $_FILES['main-image']['size'] > 0) {
            // Delete old main image from database and file system
            $sqlSelectOldMainImage = "SELECT image_name FROM property WHERE id =
:property_id";
            $stmtSelectOldMainImage = $conn->prepare($sqlSelectOldMainImage);
            $stmtSelectOldMainImage->bindParam(':property_id', $propertyId);
            $stmtSelectOldMainImage->execute();
            $oldMainImageName = $stmtSelectOldMainImage->fetchColumn();
            if ($oldMainImageName) {
                unlink("../../store/property/main-images/" . $oldMainImageName);
                // Delete old main image record from database
                $sqlDeleteOldMainImage = "UPDATE property SET image_name = NULL
WHERE id = :property_id";
                $stmtDeleteOldMainImage =
$conn->prepare($sqlDeleteOldMainImage);
                $stmtDeleteOldMainImage->bindParam(':property_id', $propertyId);
                $stmtDeleteOldMainImage->execute();
            }

            // Upload and update new main image
            $targetDirectory = "../../store/property/main-images/";
            $extension = pathinfo($_FILES["main-image"]["name"],
PATHINFO_EXTENSION);
            $mainImageName = $_SESSION['user']['id'] . "_" .
generateRandomString() . "." . $extension;

            $targetFilePath = $targetDirectory . $mainImageName;
            if (!move_uploaded_file($_FILES["main-image"]["tmp_name"],
$targetFilePath)) {
                throw new Exception("Main image uploading failed");
            }

            // Update the database with the new main image name
            $sqlUpdateMainImage = "UPDATE property SET image_name = :image_name
WHERE id = :property_id";
            $stmtUpdateMainImage = $conn->prepare($sqlUpdateMainImage);
            $stmtUpdateMainImage->bindParam(':image_name', $mainImageName);
            $stmtUpdateMainImage->bindParam(':property_id', $propertyId);
            $stmtUpdateMainImage->execute();
        }

        // Handle file upload for additional images
        if (isset($_FILES['additional-images'])) {
            $targetDirectory = "../../store/property/images/";

            // Delete old additional images from database and file system
            $sqlSelectOldAdditionalImages = "SELECT name FROM image WHERE
property_id = :property_id";
            $stmtSelectOldAdditionalImages =
$conn->prepare($sqlSelectOldAdditionalImages);
            $stmtSelectOldAdditionalImages->bindParam(':property_id',
$propertyId);
            $stmtSelectOldAdditionalImages->execute();
            $oldAdditionalImages =
$stmtSelectOldAdditionalImages->fetchAll(PDO::FETCH_COLUMN);

            foreach ($oldAdditionalImages as $oldImageName) {
                unlink($targetDirectory . $oldImageName);
                // Delete old additional image record from database
                $sqlDeleteOldAdditionalImage = "DELETE FROM image WHERE name =
:image_name";
                $stmtDeleteOldAdditionalImage =
$conn->prepare($sqlDeleteOldAdditionalImage);
                $stmtDeleteOldAdditionalImage->bindParam(':image_name',
$oldImageName);
                $stmtDeleteOldAdditionalImage->execute();
            }

            // Upload new additional images
            $additionalImages = $_FILES['additional-images'];
            foreach ($additionalImages['tmp_name'] as $key => $tmpName) {
                $extension = pathinfo($additionalImages["name"][$key],
PATHINFO_EXTENSION);
                $imageName = $_SESSION['user']['id'] . "_" .
generateRandomString() . "." . $extension;

                $targetFilePath = $targetDirectory . $imageName;
                if (!move_uploaded_file($tmpName, $targetFilePath)) {
                    throw new Exception("Additional image uploading failed");
                }

                // Insert the new additional image into the database
                $sqlInsertAdditionalImage = "INSERT INTO image (property_id,
name) VALUES (:property_id, :image_name)";
                $stmtInsertAdditionalImage =
$conn->prepare($sqlInsertAdditionalImage);
                $stmtInsertAdditionalImage->bindParam(':property_id',
$propertyId);
```



```
                $stmtInsertAdditionalImage->bindParam(':image_name',
$imageName);
                $stmtInsertAdditionalImage->execute();
            }
        }

        // Update property details
        $sqlUpdateProperty = "UPDATE property SET name = :name, type_id =
:type_id, price = :price, description = :description, city_id = :city_id,
address = :address, zip_code = :zip, size = :size, full_description =
:full_description WHERE id = :property_id";
        $stmtUpdateProperty = $conn->prepare($sqlUpdateProperty);
        $stmtUpdateProperty->bindParam(':name', $name);
        $stmtUpdateProperty->bindParam(':type_id', $type_id);
        $stmtUpdateProperty->bindParam(':price', $price);
        $stmtUpdateProperty->bindParam(':description', $description);
        $stmtUpdateProperty->bindParam(':city_id', $city_id);
        $stmtUpdateProperty->bindParam(':address', $address);
        $stmtUpdateProperty->bindParam(':zip', $zip);
        $stmtUpdateProperty->bindParam(':size', $size);
        $stmtUpdateProperty->bindParam(':property_id', $propertyId);
        $stmtUpdateProperty->bindParam(':full_description', $fullDescription);
        $stmtUpdateProperty->execute();

        // Update property structure details
        $sqlUpdateStructure = "UPDATE structure SET bedrooms = :bedrooms,
bathrooms = :bathrooms, balconies = :balconies WHERE id = :property_id";
        $stmtUpdateStructure = $conn->prepare($sqlUpdateStructure);
        $stmtUpdateStructure->bindParam(':bedrooms', $bedrooms);
        $stmtUpdateStructure->bindParam(':bathrooms', $bathrooms);
        $stmtUpdateStructure->bindParam(':balconies', $balconies);
        $stmtUpdateStructure->bindParam(':property_id', $propertyId);
        $stmtUpdateStructure->execute();

        // Commit the transaction
        $conn->commit();

    } catch (PDOException $e) {
        // Rollback the transaction and handle database errors
        $conn->rollBack();
        echo "Error: " . $e->getMessage();
    } catch (Exception $e) {
        $conn->rollBack();
        echo "Error: " . $e->getMessage();
    }

    $conn = null;

    //Redirect to my properties
```

```
   header("Location: ../../my-properties.php");
}

// Function to generate a random string
function generateRandomString($length = 10) {
   $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
   $randomString = '';
   for ($i = 0; $i < $length; $i++) {
       $randomString .= $characters[rand(0, strlen($characters) - 1)];
   }
   return $randomString;
}
?>
```

services/property/get-cities.php

```
<?php
require '../../config/conn.php';

$cities = null;

try {
   // Check if the country parameter is set
   $conn = $createConnection();
   if(isset($_GET['country_id'])) {
       $countryId = $_GET['country_id'];

       // SQL query to select cities based on the country
       $query = "SELECT city.* FROM city RIGHT JOIN country ON city.country_id
= country.id WHERE country.id = :countryId AND city.id IS NOT NULL AND
city.name IS NOT NULL";

       // Prepare the SQL statement
       $statement = $conn->prepare($query);

       // Bind the country parameter to the query
       $statement->bindParam(':countryId', $countryId);

       // Execute the SQL statement
       $statement->execute();

       // Fetch all rows as an associative array
       $cities = $statement->fetchAll(PDO::FETCH_ASSOC);

   } else {
       // If country parameter is not set, return all cities
       $query = "SELECT * FROM city WHERE id IS NOT NULL AND name IS NOT NULL";
       $statement = $conn->query($query);
       $cities = $statement->fetchAll(PDO::FETCH_ASSOC);
   }

   // Check if cities array is empty

       // Encode cities into JSON and output
       echo json_encode($cities);


} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;
?>
```

services/get-countries.php

```
<?php
require '../../config/conn.php';
$countries = null;
try {
   $conn = $createConnection();
   // SQL query to select all property
   $query = "SELECT * FROM country";


   // Prepare the SQL statement
   $statement = $conn->prepare($query);

   // Execute the SQL statement
   $statement->execute();

   // Fetch all rows as an associative array
   $countries = $statement->fetchAll(PDO::FETCH_ASSOC);

   // Output the property in PHP

} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;

// Encode property into JSON
```

```
echo json_encode($countries);
?>
```

services/property/get-images.php

```
<?php
require '../../config/conn.php';
$properties = null;

try {
   $conn = $createConnection();
   if(isset($_GET['id']) && filter_var($_GET['id'], FILTER_VALIDATE_INT)) {
       $property_id = $_GET['property_id'];
       // Prepare and execute SQL query to select image names for the given
property_id
       $query = "SELECT name FROM image WHERE property_id = :property_id";
       $statement = $conn->prepare($query);
       $statement->bindParam(':property_id', $_GET['id'], PDO::PARAM_INT);
       $statement->execute();

       // Fetch image names
       $image_names = $statement->fetchAll(PDO::FETCH_COLUMN, 'name');

       // Return image names as JSON
       echo json_encode($image_names);
   } else {
       // Return an error message if property_id is missing or invalid
       echo json_encode(['error' => 'Invalid property_id']);
   }

} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;
?>
```

services/get-my-properties.php

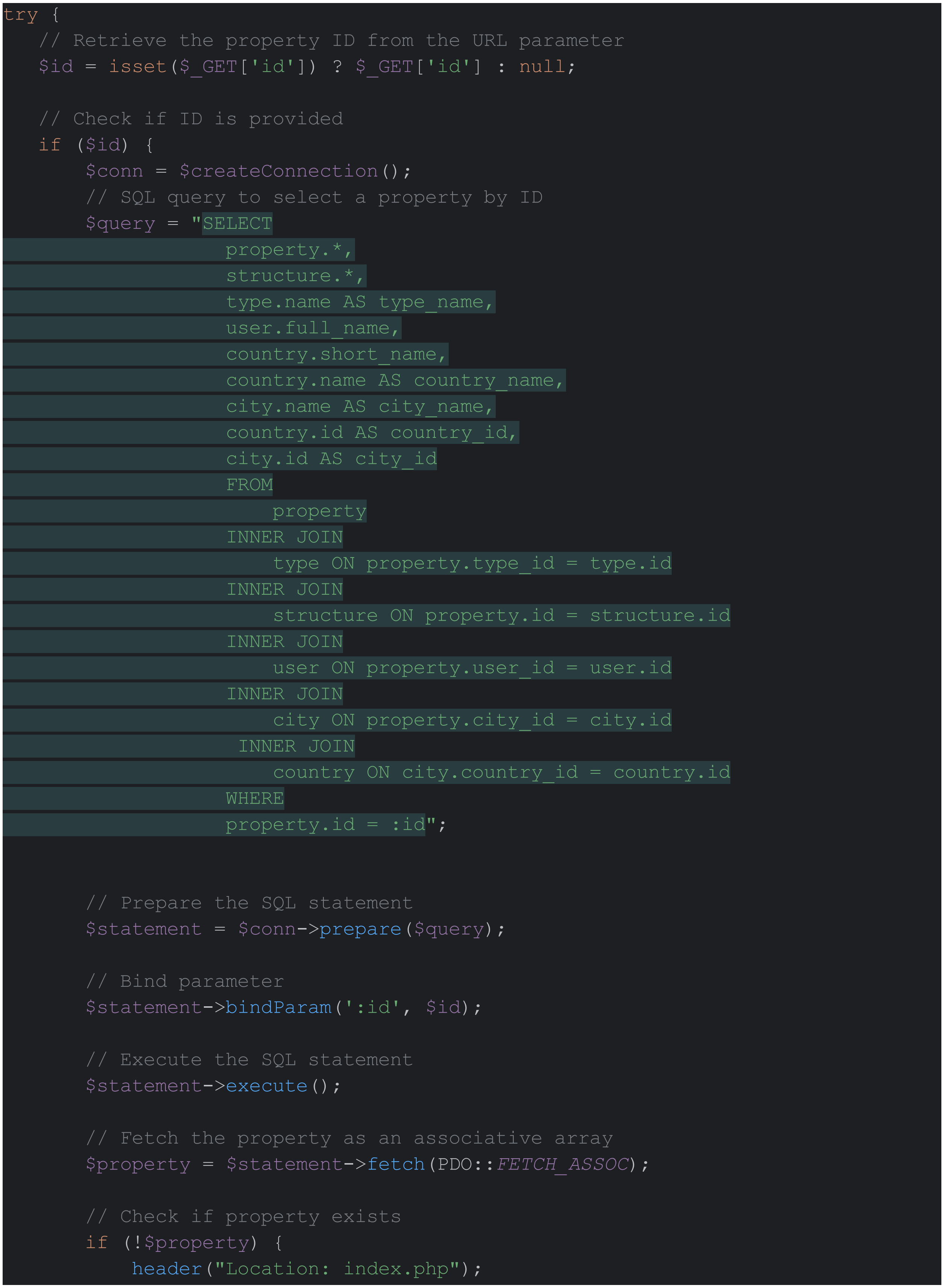
```
<?php
require '../../config/conn.php';
$properties = null;
try {
   $conn = $createConnection();
   // SQL query to select all user property

   $userId = $_GET['id'];
   $query = "SELECT property.*, type.id AS type_id, type.name AS type_name,
      country.name AS country_name, country.short_name AS country_short_name, city.name AS city_name
       FROM property
       INNER JOIN type ON property.type_id = type.id
       INNER JOIN city ON property.city_id = city.id
       INNER JOIN country ON city.country_id = country.id
       WHERE user_id = :id";

   // Prepare the SQL statement
   $statement = $conn->prepare($query);

   $statement->bindParam(':id', $userId);

   // Execute the SQL statement
   $statement->execute();

   // Fetch all rows as an associative array
   $properties = $statement->fetchAll(PDO::FETCH_ASSOC);

   // Output the property in PHP

} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;

// Encode property into JSON
echo json_encode($properties);
?>
```

services/get-properties.php

```
<?php
require '../../config/conn.php';
$properties = null;

try {
   $conn = $createConnection();

   // Retrieve the search term from the URL parameter
   $search = isset($_GET['search']) ? $_GET['search'] : '';

   // SQL query to select properties, with search functionality
   $query = "
       SELECT
           property.*,
           type.id AS type_id,
           type.name AS type_name,
           country.name AS country_name,
           country.short_name AS country_short_name,
           city.name AS city_name
       FROM
           property
       INNER JOIN
           type ON property.type_id = type.id
       INNER JOIN
           city ON property.city_id = city.id
       INNER JOIN
           country ON city.country_id = country.id
       WHERE
           :search = '' OR property.name LIKE :searchWild
   ";

   // Prepare the SQL statement
   $statement = $conn->prepare($query);

   // Bind parameters
   $searchWild = '%' . $search . '%';
   $statement->bindParam(':search', $search);
   $statement->bindParam(':searchWild', $searchWild);

   // Execute the SQL statement
   $statement->execute();

   // Fetch all rows as an associative array
   $properties = $statement->fetchAll(PDO::FETCH_ASSOC);

//      header("Location: ../../index.php?search=" . $search);
   // Output the properties in JSON format
   echo json_encode($properties);


} catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;
?>
```

services/property/get-property.php

```
<?php
require 'config/conn.php';
```

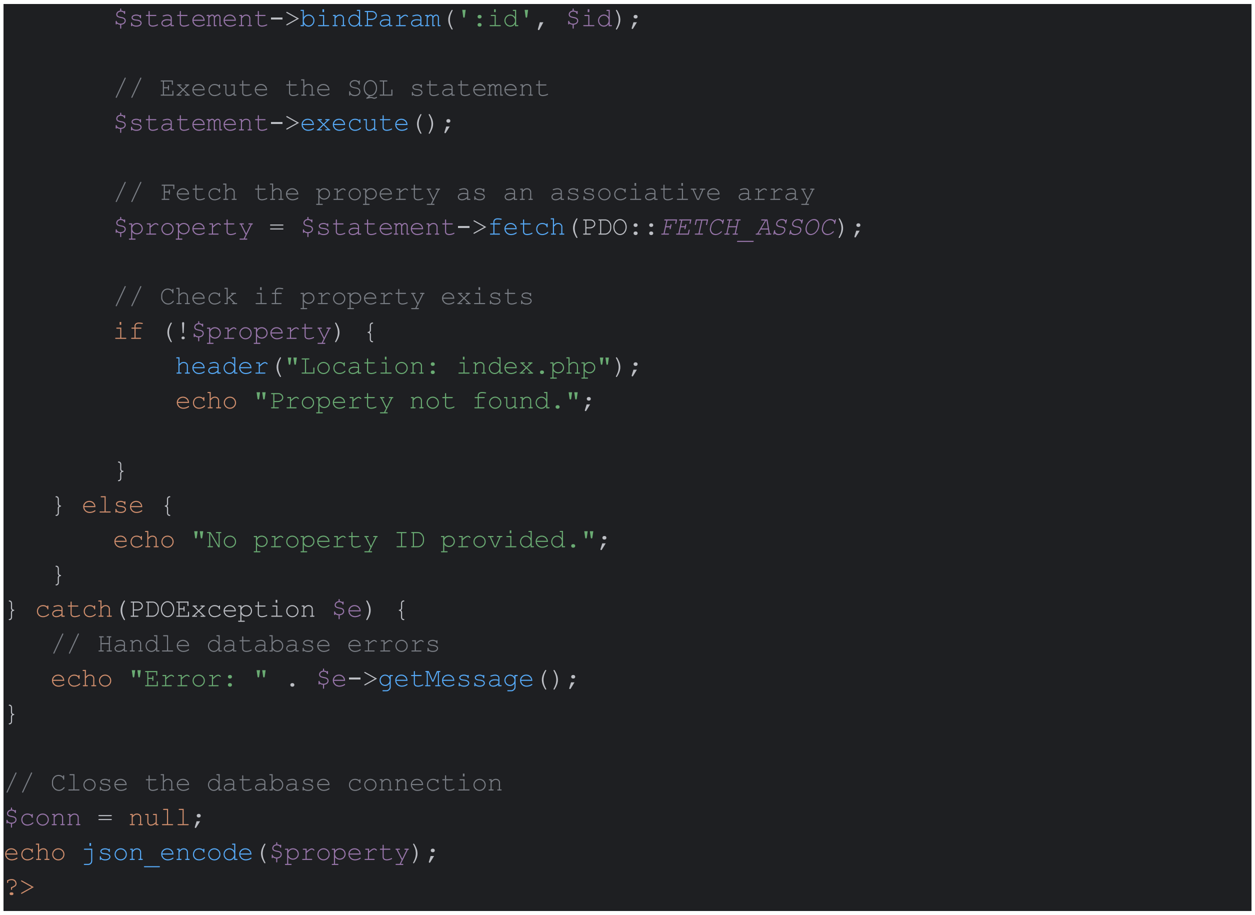


```php
try {
    // Retrieve the property ID from the URL parameter
    $id = isset($_GET['id']) ? $_GET['id'] : null;

    // Check if ID is provided
    if ($id) {
        $conn = $createConnection();
        // SQL query to select a property by ID
        $query = "SELECT
                    property.*,
                    structure.*,
                    type.name AS type_name,
                    user.full_name,
                    country.short_name,
                    country.name AS country_name,
                    city.name AS city_name,
                    country.id AS country_id,
                    city.id AS city_id
                    FROM
                        property
                    INNER JOIN
                        type ON property.type_id = type.id
                    INNER JOIN
                        structure ON property.id = structure.id
                    INNER JOIN
                        user ON property.user_id = user.id
                    INNER JOIN
                        city ON property.city_id = city.id
                     INNER JOIN
                        country ON city.country_id = country.id
                    WHERE
                    property.id = :id";


        // Prepare the SQL statement
        $statement = $conn->prepare($query);

        // Bind parameter
        $statement->bindParam(':id', $id);

        // Execute the SQL statement
        $statement->execute();

        // Fetch the property as an associative array
        $property = $statement->fetch(PDO::FETCH_ASSOC);

        // Check if property exists
        if (!$property) {
            header("Location: index.php");
```

```
           echo "Property not found.";

       }
   } else {
       echo "No property ID provided.";
   }
} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;
?>
```

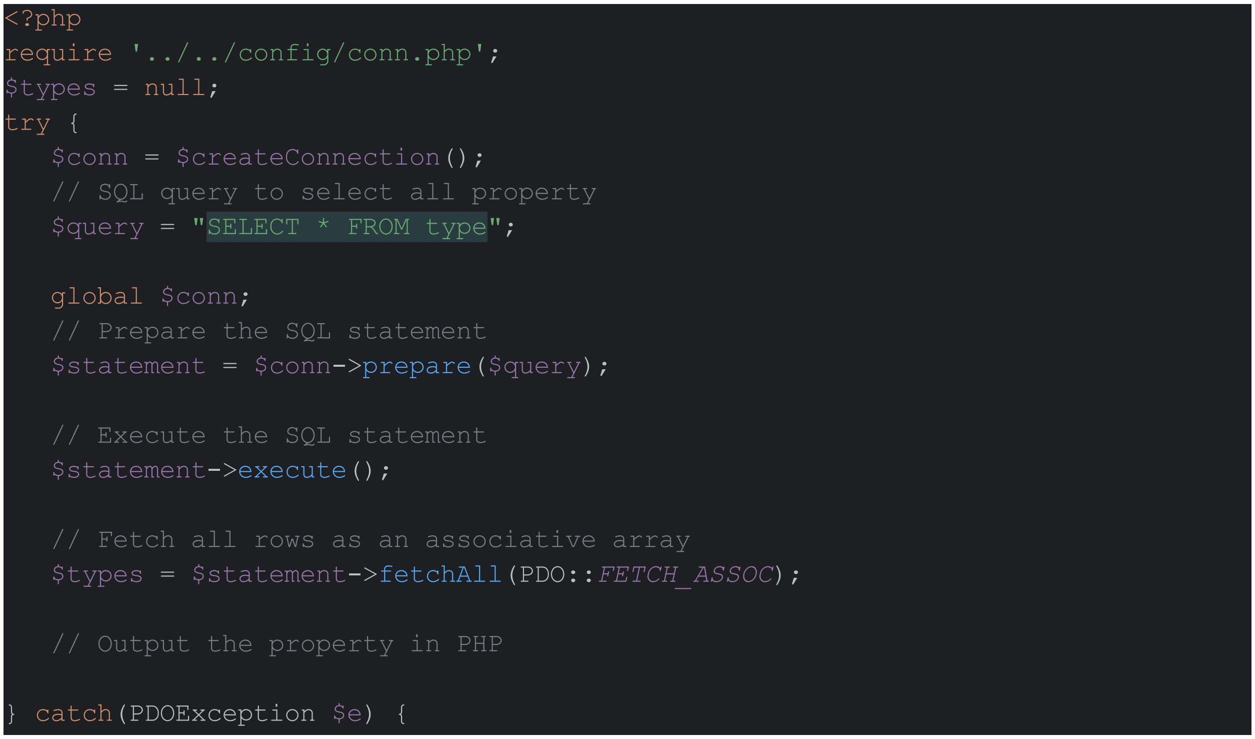
services/property/get-property-json.php

```
<?php
require 'config/conn.php';

try {
   // Retrieve the property ID from the URL parameter
   $id = isset($_GET['id']) ? $_GET['id'] : null;

   // Check if ID is provided
   if ($id) {
       $conn = $createConnection();
       // SQL query to select a property by ID
       $query = "SELECT
                   property.*,
                   structure.*,
                   type.name AS type_name,
                   user.full_name
                   FROM
                       property
                   INNER JOIN
                       type ON property.type_id = type.id
                   INNER JOIN
                       structure ON property.id = structure.id
                   INNER JOIN
                           user ON property.user_id = user.id
                   WHERE
                   property.id = :id";


       // Prepare the SQL statement
       $statement = $conn->prepare($query);

       // Bind parameter
```



```php
        $statement->bindParam(':id', $id);

        // Execute the SQL statement
        $statement->execute();

        // Fetch the property as an associative array
        $property = $statement->fetch(PDO::FETCH_ASSOC);

        // Check if property exists
        if (!$property) {
            header("Location: index.php");
            echo "Property not found.";

        }
    } else {
        echo "No property ID provided.";
    }
} catch(PDOException $e) {
    // Handle database errors
    echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;
echo json_encode($property);
?>
```

services/property/get-types.php



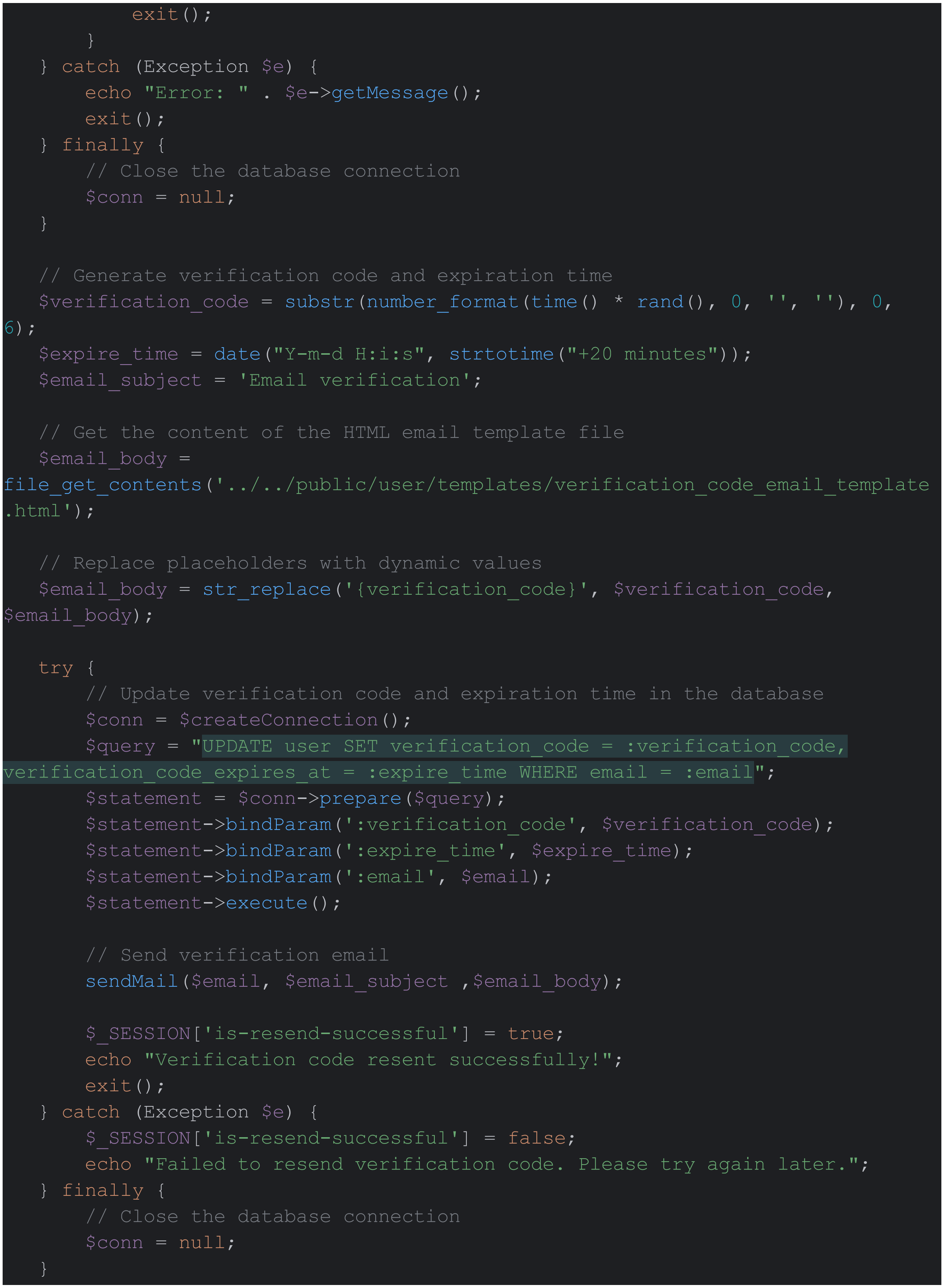
```php
<?php
require '../../config/conn.php';
$types = null;
try {
    $conn = $createConnection();
    // SQL query to select all property
    $query = "SELECT * FROM type";

    global $conn;
    // Prepare the SQL statement
    $statement = $conn->prepare($query);

    // Execute the SQL statement
    $statement->execute();

    // Fetch all rows as an associative array
    $types = $statement->fetchAll(PDO::FETCH_ASSOC);

    // Output the property in PHP

} catch(PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
}

// Close the database connection
$conn = null;

// Encode property into JSON
echo json_encode($types);
?>
```

Change-settings.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';


// Check if the form is submitted
if(is_post()) {
   // Include your database connection file
   $conn = $createConnection();

   // Retrieve user ID from session
   $userId = $_SESSION["user"]['id'];

   // Retrieve form data
   $language = $_POST["language"];
   $currency = $_POST["currency"];
   $cardsPerRow = $_POST["cards_per_row"];
   $theme = $_POST["theme"];

   // Prepare SQL statement
   $sql = "UPDATE user_settings SET language = :language, currency = :currency, cards_per_row = :cards_per_row, theme = :theme WHERE user_id = :user_id";

   // Prepare the statement
   $stmt = $conn->prepare($sql);

   // Bind parameters
   $stmt->bindParam(":language", $language);
   $stmt->bindParam(":currency", $currency);
   $stmt->bindParam(":cards_per_row", $cardsPerRow);
   $stmt->bindParam(":theme", $theme);
   $stmt->bindParam(":user_id", $userId);

   // Execute the statement
   if ($stmt->execute()) {
       // Settings updated successfully
       // You can redirect the user to a success page or display a success
message
       // For example:

       require 'get-settings.php';
       setFlashMessage("settings-changed-success", 'Settings successfully
changed.');
       header("Location: ../../account.php");
       // exit;
   } else {
       // Error occurred while updating settings
       // You can redirect the user to an error page or display an error
message
       // For example:
       setFlashMessage("error", 'Something went wrong.');
//        header("Location: ../../account.php");
       // exit;
   }


   // Close statement
   $stmt = null;

   // Close database connection (optional if PDO is in persistent mode)
   // $pdo = null;
}
?>
```

Email-sesend-code.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-get.php';
require '../../util/send-mail.php';

if(is_get()) {
    $email = $_SESSION['registration-email'];

    // Check if the user is already verified
    try {
        $conn = $createConnection();
        $query = "SELECT email_verified_at FROM user WHERE email = :email";
        $statement = $conn->prepare($query);
        $statement->bindParam(':email', $email);
        $statement->execute();
        $result = $statement->fetch();

        if ($result['email_verified_at'] !== null) {
            echo "User is already verified.";
```



```
            exit();
        }
    } catch (Exception $e) {
        echo "Error: " . $e->getMessage();
        exit();
    } finally {
        // Close the database connection
        $conn = null;
    }

    // Generate verification code and expiration time
    $verification_code = substr(number_format(time() * rand(), 0, '', ''), 0,
6);
    $expire_time = date("Y-m-d H:i:s", strtotime("+20 minutes"));
    $email_subject = 'Email verification';

    // Get the content of the HTML email template file
    $email_body =
file_get_contents('../../public/user/templates/verification_code_email_template
.html');

    // Replace placeholders with dynamic values
    $email_body = str_replace('{verification_code}', $verification_code,
$email_body);

    try {
        // Update verification code and expiration time in the database
        $conn = $createConnection();
        $query = "UPDATE user SET verification_code = :verification_code,
verification_code_expires_at = :expire_time WHERE email = :email";
        $statement = $conn->prepare($query);
        $statement->bindParam(':verification_code', $verification_code);
        $statement->bindParam(':expire_time', $expire_time);
        $statement->bindParam(':email', $email);
        $statement->execute();

        // Send verification email
        sendMail($email, $email_subject ,$email_body);

        $_SESSION['is-resend-successful'] = true;
        echo "Verification code resent successfully!";
        exit();
    } catch (Exception $e) {
        $_SESSION['is-resend-successful'] = false;
        echo "Failed to resend verification code. Please try again later.";
    } finally {
        // Close the database connection
        $conn = null;
    }
```

```
}
?>
```

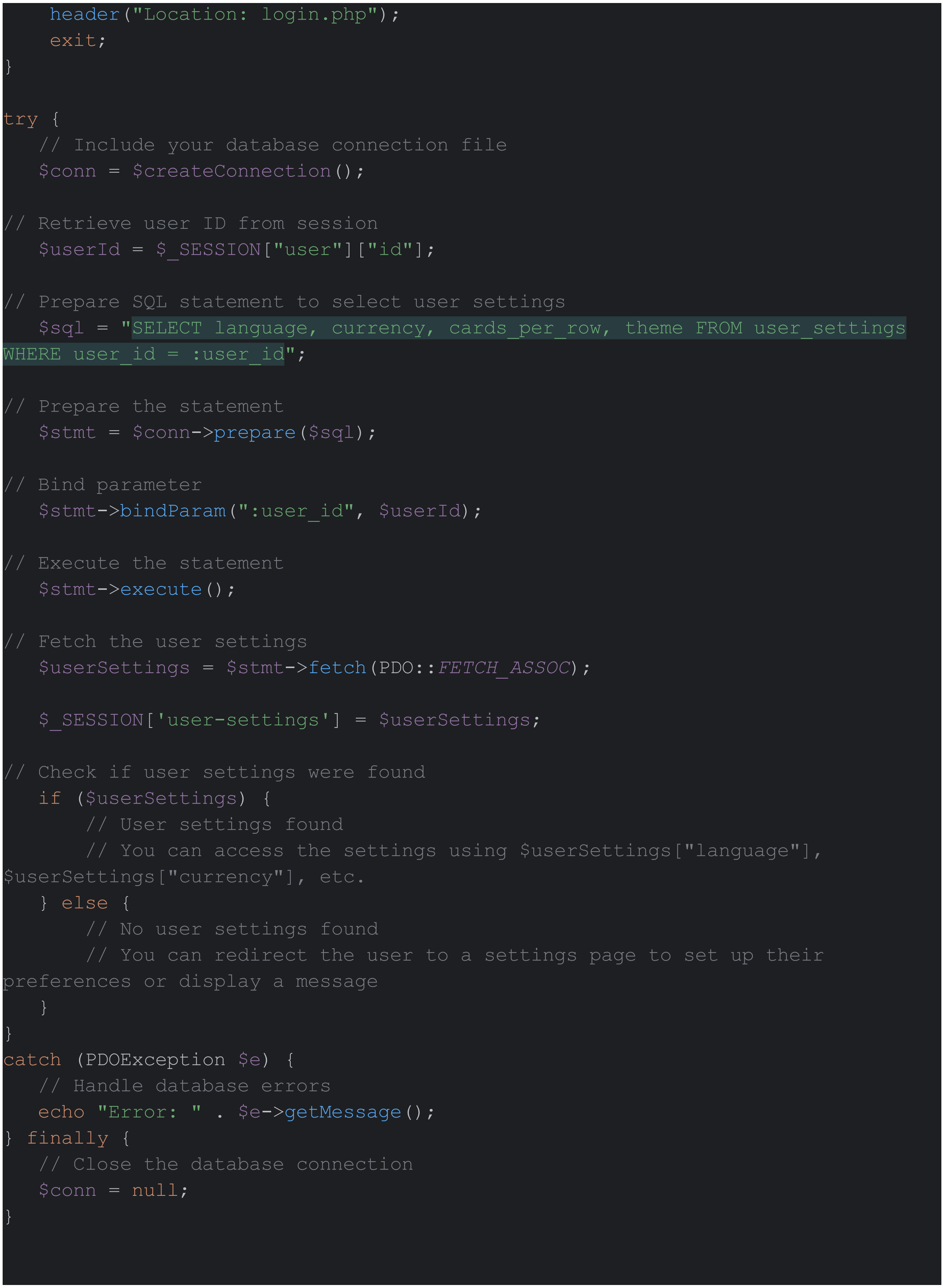
Email-verify.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-post.php';

if(is_post()) {
   $email = $_SESSION["registration-email"];
   $postData = file_get_contents("php://input");

   // Decode the JSON data
   $requestData = json_decode($postData, true); // true to get an associative array

   $verification_code = null;
   // Check if 'verificationCode' exists in the decoded data
   if (isset($requestData['verificationCode'])) {
       $verification_code = $requestData['verificationCode'];
       // Now you have $verificationCode, you can use it in your PHP script
   } else {
       // Handle case where 'verificationCode' is not found
       http_response_code(400); // Bad Request
       exit("Verification code not found in request.");
   }

   $now = date("Y-m-d H:i:s");

   // Check if the user is already verified
   try {
       $conn = $createConnection();
       $query = "SELECT email_verified_at FROM user WHERE email = :email";
       $statement = $conn->prepare($query);
       $statement->bindParam(':email', $email);
       $statement->execute();
       $result = $statement->fetch();

       if ($result['email_verified_at'] !== null) {
           echo "User is already verified.";
           exit();
       }
   } catch (Exception $e) {
       echo "Error: " . $e->getMessage();
       exit();
   } finally {
       // Close the database connection
       $conn = null;
```

```
    }

    try {

        $conn = $createConnection();
        $query = "UPDATE user SET email_verified_at = :now WHERE email = :email AND verification_code = :verification_code AND verification_code_expires_at > :now";

        $stmt = $conn->prepare($query);
        // Bind parameters
        $stmt->bindParam(':email', $email);
        $stmt->bindParam(':now', $now);
        $stmt->bindParam(':verification_code', $verification_code);

        // Execute the SQL statement
        $stmt->execute();

        $rowsAffected = $stmt->rowCount();
        if ($rowsAffected > 0) {
            // Verification successful
            // Redirect or perform further actions
            header("Location: ../../email-verification-success.php");
            exit(); // Stop script execution after redirection
        } else {
            // No rows updated, verification failed
            echo 'Verification code is invalid.';

        }
    } catch (PDOException $e) {
        // Handle database errors
        echo "Error: " . $e->getMessage();
    } finally {
        // Close the database connection
        $conn = null;
    }
}
?>
```

Get-settings.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-get.php';


if (!isset($_SESSION["user"])) {
    // Redirect the user to the login page or display an error message
    // For example:
```



```
    header("Location: login.php");
    exit;
}

try {
   // Include your database connection file
   $conn = $createConnection();

// Retrieve user ID from session
   $userId = $_SESSION["user"]["id"];

// Prepare SQL statement to select user settings
   $sql = "SELECT language, currency, cards_per_row, theme FROM user_settings
WHERE user_id = :user_id";

// Prepare the statement
   $stmt = $conn->prepare($sql);

// Bind parameter
   $stmt->bindParam(":user_id", $userId);

// Execute the statement
   $stmt->execute();

// Fetch the user settings
   $userSettings = $stmt->fetch(PDO::FETCH_ASSOC);

   $_SESSION['user-settings'] = $userSettings;

// Check if user settings were found
   if ($userSettings) {
       // User settings found
       // You can access the settings using $userSettings["language"],
$userSettings["currency"], etc.
   } else {
       // No user settings found
       // You can redirect the user to a settings page to set up their
preferences or display a message
   }
}
catch (PDOException $e) {
   // Handle database errors
   echo "Error: " . $e->getMessage();
} finally {
   // Close the database connection
   $conn = null;
}
```

Login.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/flash.php';

if(is_post()) {
   if(empty($_SESSION['user'])) {
       try {
           $conn = $createConnection();
           $query = "SELECT id, email, full_name, role_id, email_verified_at,
language, theme, cards_per_row, currency, questionnaire_submitted_at,
created_at, password FROM user WHERE email = :email";

           $stmt = $conn->prepare($query);
           // Bind parameters
           $stmt->bindParam(':email', $_POST['email']);

           // Execute the SQL statement
           $stmt->execute();

           // Fetch the user data
           $user = $stmt->fetch(PDO::FETCH_ASSOC);

           if ($user) {
               // Verify the hashed password
               if (password_verify($_POST['password'], $user['password'])) {
                   if($user['email_verified_at'] == null) {
                       die("Please verify your account <a
href='../../email-verification.php'>Verify account</a>");
                   }

                   // Password is correct
                   $_SESSION['user'] = $user;
                   setFlashMessage("fullname", $user['full_name']);
                   $_SESSION['role_id'] = $user['role_id']; // Add role_id to
session
                   // Redirect to index.php or any other page after login

                   require 'get-settings.php';
                   header("Location: ../../index.php");
                   exit();
               } else {
                   // Incorrect password
                   $_SESSION['error'] = "Please try again. Username or password
is incorrect.";
```

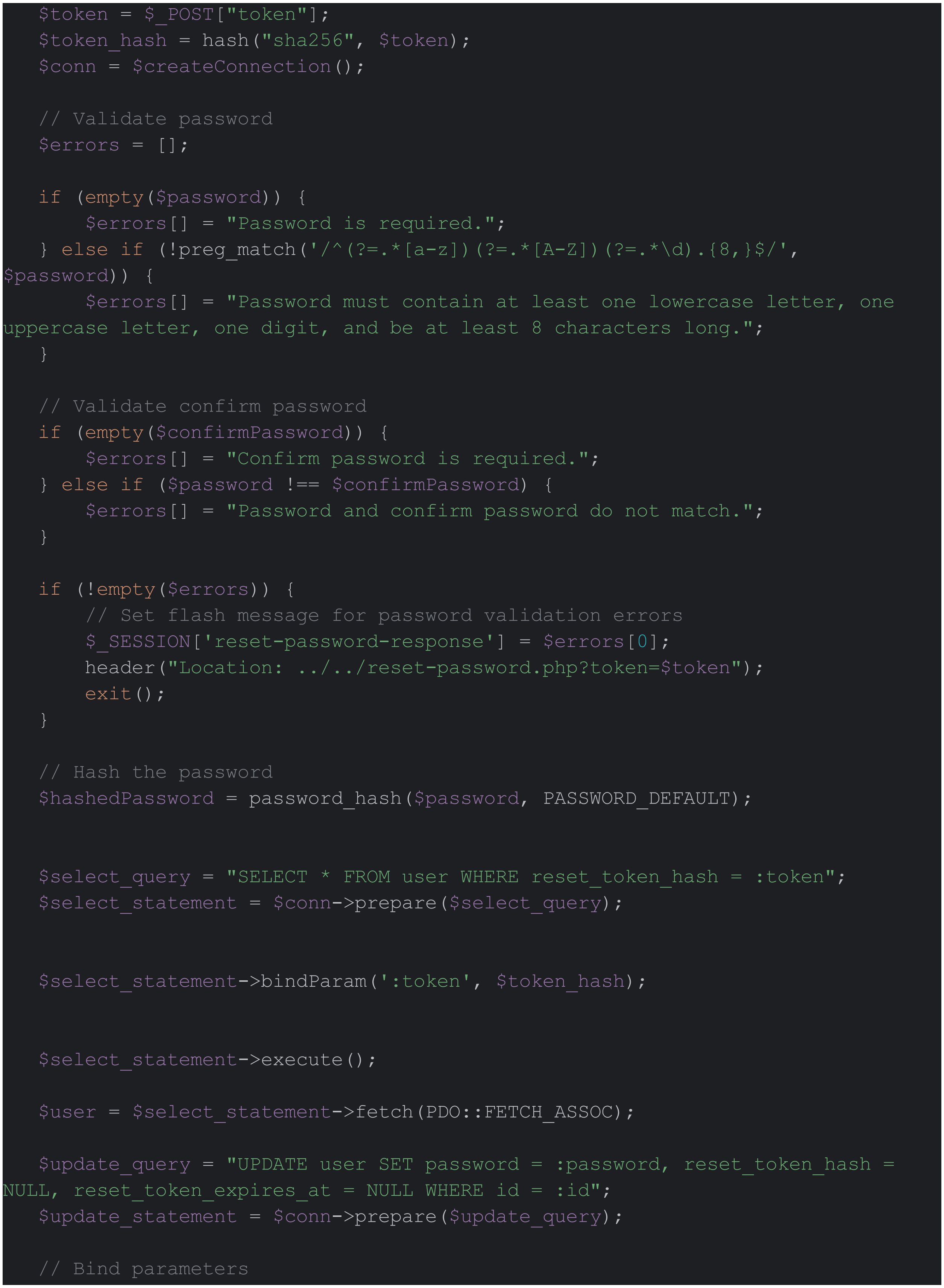
```
                    header("Location: ../../login.php");
                    exit();
                }
            } else {
                // User not found
                $_SESSION['error'] = "Please try again. Username or password is incorrect.";
                header("Location: ../../login.php");
                exit();
            }
        } catch (PDOException $e) {
            // Handle database errors
            echo "Error: " . $e->getMessage();
        } finally {
            // Close the database connection
            $conn = null;
        }
    } else {
        $_SESSION['error'] = "You are already logged in. Please log out first.";
        header("Location: ../../login.php");
        exit();
    }
}
?>
```

Logout.php

```
<?php
session_start();

// Unset all session variables
$_SESSION = array();

// Destroy the session
session_destroy();

// Redirect to the homepage or any other page after logout
header("Location: ../../login.php");
exit;
?>
```

Process-reset-password.php

```
<?php
session_start();
require '../../util/is-get.php';
require '../../config/conn.php';

try {
    $password = $_POST['new_password'];
    $confirmPassword = $_POST['confirm_password'];
```



```
    $token = $_POST["token"];
    $token_hash = hash("sha256", $token);
    $conn = $createConnection();

    // Validate password
    $errors = [];

    if (empty($password)) {
        $errors[] = "Password is required.";
    } else if (!preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d).{8,}$/',
$password)) {
        $errors[] = "Password must contain at least one lowercase letter, one
uppercase letter, one digit, and be at least 8 characters long.";
    }

    // Validate confirm password
    if (empty($confirmPassword)) {
        $errors[] = "Confirm password is required.";
    } else if ($password !== $confirmPassword) {
        $errors[] = "Password and confirm password do not match.";
    }

    if (!empty($errors)) {
        // Set flash message for password validation errors
        $_SESSION['reset-password-response'] = $errors[0];
        header("Location: ../../reset-password.php?token=$token");
        exit();
    }

    // Hash the password
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);


    $select_query = "SELECT * FROM user WHERE reset_token_hash = :token";
    $select_statement = $conn->prepare($select_query);


    $select_statement->bindParam(':token', $token_hash);


    $select_statement->execute();

    $user = $select_statement->fetch(PDO::FETCH_ASSOC);

    $update_query = "UPDATE user SET password = :password, reset_token_hash =
NULL, reset_token_expires_at = NULL WHERE id = :id";
    $update_statement = $conn->prepare($update_query);

    // Bind parameters
```

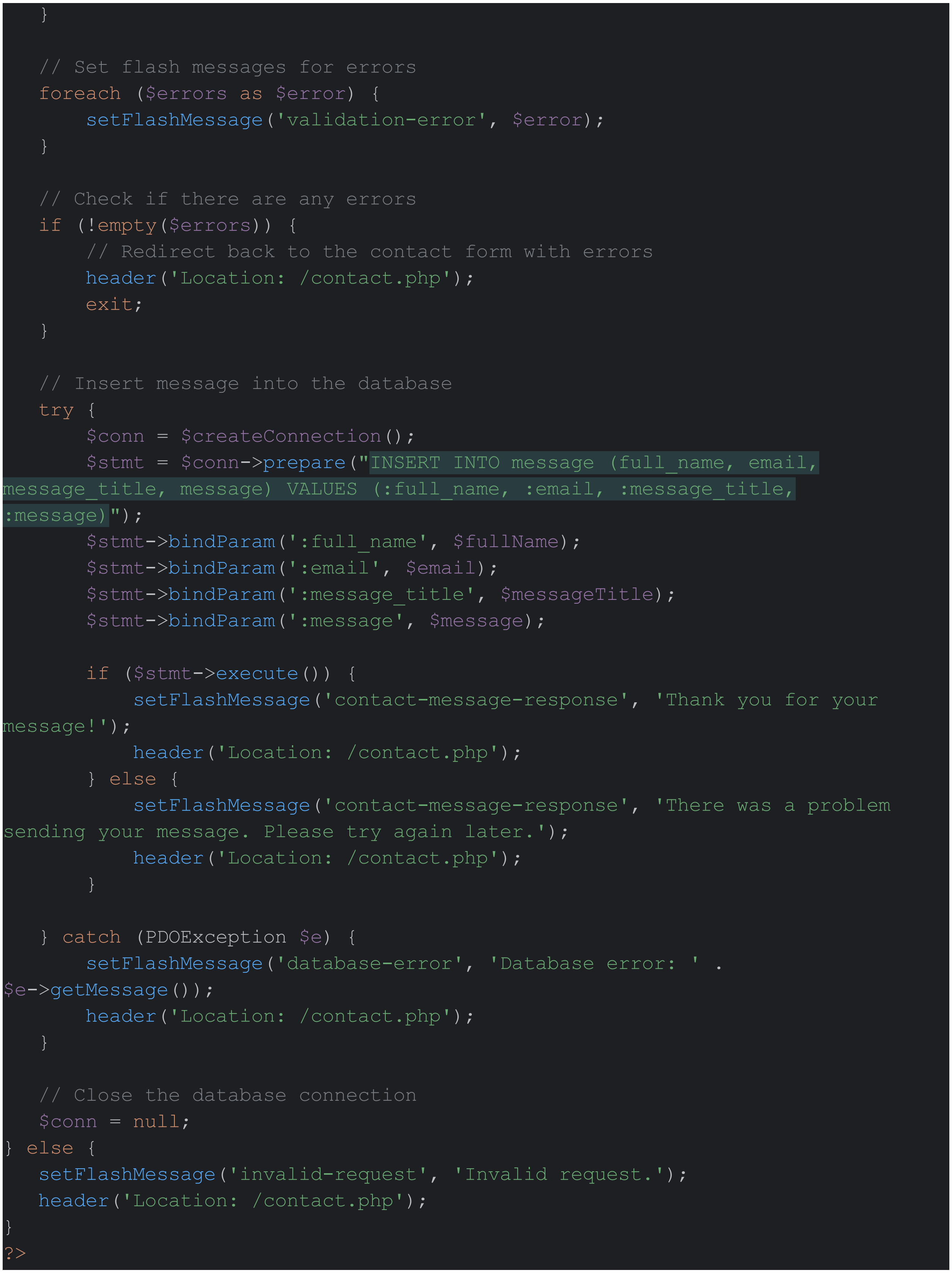
```php
    $update_statement->bindParam(':password', $hashedPassword);
    $update_statement->bindParam(':id', $user['id']);

    $update_statement->execute();

    header("Location: ../../reset-password-success.php");

} catch (Exception $e) {
    // Handle exceptions
    echo "Error: " . $e->getMessage();
} finally {
    // Close the database connection
    $conn = null;
}
?>
```

Register.php

```php
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/send-mail.php';
require '../../util/flash.php';

if(is_post()) {

    function isValidName($name) {
        return preg_match('/^[A-Z][a-zA-Z\s]{1,}[a-zA-Z]$/', $name);
    }
    // Validation of inputs
    $errors = [];

    // Validate full name
    if(empty($_POST["fullname"])) {
        $errors[] = "Full name is required.";
    } else if(!isValidName($_POST["fullname"])) {
        $errors[] = "Full name is not valid. It must be at least 2 characters
long and start with a capital letter. Only letters allowed.";
    } else {
        $fullname = $_POST["fullname"];
    }

    // Validate email
    if(empty($_POST["email"])) {
        $errors[] = "Email is required.";
    } else if(!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
        $errors[] = "Invalid email format.";
    } else {
        $email = $_POST["email"];
    }

    // Validate confirm password
    if(empty($_POST["confirm_password"])) {
        $errors[] = "Confirm password is required.";
    } elseif ($_POST["password"] !== $_POST["confirm_password"]) {
        $errors[] = "Passwords do not match.";
    }

    // Validate password
    if(empty($_POST["password"])) {
        $errors[] = "Password is required.";
    } elseif (!preg_match('/^(?=.*\d)(?=.*[A-Z])[A-Za-z\d]{8,}$/',
$_POST["password"])) {
        $errors[] = "Password must contain at least one number and one capital
letter, and be at least 8 characters long.";
    } else {
        $password = password_hash($_POST["password"], PASSWORD_DEFAULT);
    }

    // Validate terms checkbox
    if(empty($_POST["agree_terms"])) {
        $errors[] = "You must agree to the terms and conditions.";
    }

    if(empty($errors)) {
        // Sending email
        $mail = null;
        $email_verified_time = null;

        // Calculate expiration time 20 minutes in the future
        $expire_time = date("Y-m-d H:i:s", strtotime("+20 minutes"));
        $verification_code = substr(number_format(time() * rand(), 0, '', ''),
0, 6);
        $email_subject = 'Email verification';

        // Get the content of the HTML email template file
        $email_body =
file_get_contents('../../public/user/templates/verification_code_email_template
.html');

        // Replace placeholders with dynamic values
        $email_body = str_replace('{verification_code}', $verification_code,
$email_body);

        try {
            //REGISTRATION
            $conn = $createConnection();
            // SQL query to insert user in database
            $query = "INSERT INTO user (full_name, email, password,
verification_code, email_verified_at, verification_code_expires_at)
                     VALUES (:fullname, :email, :password, :verification_code,
:email_verified_at, :expire_time)";
            $statement = $conn->prepare($query);

            // Bind parameters
            $statement->bindParam(':fullname', $fullname);
            $statement->bindParam(':email', $email);
            $statement->bindParam(':password', $password);
            $statement->bindParam(':verification_code', $verification_code);
            $statement->bindParam(':email_verified_at', $email_verified_time);
            $statement->bindParam(':expire_time', $expire_time);

            // Execute the SQL statement
            $statement->execute();

            // Get the ID of the inserted user
            $user_id = $conn->lastInsertId();

            // Insert the user ID into user_settings table
            $insertQuery = "INSERT INTO user_settings (user_id) VALUES
(:user_id)";
            $insertStmt = $conn->prepare($insertQuery);
            $insertStmt->bindParam(':user_id', $user_id);
            $insertStmt->execute();

            $_SESSION['registration-email'] = $email;

            sendMail($email, $email_subject, $email_body);

            header("Location: ../../email-verification.php");
            exit();
        } catch (Exception $e) {
            // Handle exceptions
            echo "Error: " . $e->getMessage();
        } finally {
            // Close the database connection
            $conn = null;
        }
    } else {
        // Display errors
        setFlashMessage("register-response", $errors[0]);
        header("Location: ../../register.php");
        exit();
    }
}
?>
```

Send-contact-message.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/flash.php';
require '../../util/is-post.php';

function isValidName($name) {
   return preg_match('/^[A-Z][a-zA-Z\s]{1,}[a-zA-Z]$/', $name);
}

if (is_post()) {
   $errors = [];

   // Sanitize and validate input
   $fullName = isset($_POST['full_name']) ? trim($_POST['full_name']) : '';
   $email = isset($_POST['email']) ? trim($_POST['email']) : '';
   $messageTitle = isset($_POST['message_title']) ?
trim($_POST['message_title']) : '';
   $message = isset($_POST['message']) ? trim($_POST['message']) : '';

   if (empty($fullName)) {
       $errors[] = 'Full name is required.';
       setFlashMessage('fullname-validation-msg', 'Full name is required.');
   } elseif (!isValidName($fullName)) {
       $errors[] = 'Full name is not valid. It must be at least 2 characters
long and start with a capital letter. Only letters allowed.';
       setFlashMessage('fullname-validation-msg', 'Full name is not valid. It
must be at least 2 characters long and start with a capital letter. Only
letters allowed.');
   }

   if (empty($email)) {
       $errors[] = 'Email is required.';
       setFlashMessage('email-validation-msg', 'Email is required.');
   } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
       $errors[] = 'Email is not valid.';
       setFlashMessage('email-validation-msg', 'Email is not valid.');
   }

   if (empty($messageTitle)) {
       $errors[] = 'Message title is required.';
       setFlashMessage('message-title-validation-msg', 'Message title is
required.');
   }

   if (empty($message)) {
       $errors[] = 'Message is required.';
       setFlashMessage('message-validation-msg', 'Message is required.');
```



```
    }

    // Set flash messages for errors
    foreach ($errors as $error) {
        setFlashMessage('validation-error', $error);
    }

    // Check if there are any errors
    if (!empty($errors)) {
        // Redirect back to the contact form with errors
        header('Location: /contact.php');
        exit;
    }

    // Insert message into the database
    try {
        $conn = $createConnection();
        $stmt = $conn->prepare("INSERT INTO message (full_name, email,
message_title, message) VALUES (:full_name, :email, :message_title,
:message)");
        $stmt->bindParam(':full_name', $fullName);
        $stmt->bindParam(':email', $email);
        $stmt->bindParam(':message_title', $messageTitle);
        $stmt->bindParam(':message', $message);

        if ($stmt->execute()) {
            setFlashMessage('contact-message-response', 'Thank you for your
message!');
            header('Location: /contact.php');
        } else {
            setFlashMessage('contact-message-response', 'There was a problem
sending your message. Please try again later.');
            header('Location: /contact.php');
        }

    } catch (PDOException $e) {
        setFlashMessage('database-error', 'Database error: ' .
$e->getMessage());
        header('Location: /contact.php');
    }

    // Close the database connection
    $conn = null;
} else {
    setFlashMessage('invalid-request', 'Invalid request.');
    header('Location: /contact.php');
}
?>
```

Send-reset-password.php

```
<?php
session_start();
require '../../config/conn.php';
require '../../util/is-post.php';
require '../../util/send-mail.php';

   if(is_post()) {
       $email = $_POST["email"];

       $token = bin2hex(random_bytes(16));
       $token_hash = hash("sha256", $token);
       $expire_time = date("Y-m-d H:i:s", strtotime("+20 minutes"));
       $email_subject = 'Reset Password';


       // Get the content of the HTML email template file
       $email_body =
file_get_contents('../../public/user/templates/reset_password_email_template.ht
ml');

       // Replace dynamic content if needed
       $email_body = str_replace('{token}', $token, $email_body);

       try {
           $conn = $createConnection();

           $query = "UPDATE user SET reset_token_hash = :token,
reset_token_expires_at = :expiry WHERE email = :email";
           $statement = $conn->prepare($query);

           // Bind parameters
           $statement->bindParam(':token', $token_hash);
           $statement->bindParam(':expiry', $expire_time);
           $statement->bindParam(':email', $email);


           // Execute the SQL statement
           $statement->execute();

           $rowsAffected = $statement->rowCount();
           if ($rowsAffected > 0) {
              sendMail($email, $email_subject, $email_body);

               $_SESSION['is-password-reset-email-valid'] = true;
               header("Location: ../../reset-password-start.php");
               exit();
           } else {
               // No rows updated, verification failed
               $_SESSION['is-password-reset-email-valid'] = false;
               $_SESSION['reset-password-error'] = 'Email provided is not
found.';
               header("Location: ../../reset-password-start.php");
           }
       } catch (Exception $e) {
           // Handle exceptions
           echo "Error: " . $e->getMessage();
       } finally {
           // Close the database connection
           $conn = null;
       }



   }

?>
```

Account.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";


require 'public/user/account.phtml';
```

Add-property.php

```
<?php

session_start();
require "util/flash.php";
require "util/is-logged-in.php";
require "util/is-admin.php";
if(empty($_SESSION['user'])) {
   header("Location: index.php");
}

require 'public/property/add-property.phtml';
```

Admin-panel.php

```
<?php

session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";
require 'util/redirect-if-not-admin.php';



$source = $_GET['source'];

if($source == 'show_users') {
   require "services/admin/get-users.php";
}

if($source == 'show_cities') {
   require "services/admin/get-cities.php";
}

if($source == 'show_countries') {
   require "services/admin/get-countries.php";
}

if($source == 'show_messages') {
   require "services/admin/get-messages.php";
}

if(strpos($source, 'edit_country') !== false) {
   require "services/admin/get-country.php";
}

if(strpos($source, 'edit_city') !== false) {
   require "services/admin/get-city.php";
}

if(strpos($source, 'edit_message') !== false) {
   require "services/admin/get-message.php";
}

require "public/admin/admin-panel.phtml";
```

Contact.php

```
<?php

require "util/flash.php";
require "util/is-logged-in.php";
require "util/is-admin.php";
session_start();
require 'public/user/contact.phtml';
```

Delete-property.php

```
<?php
require "util/flash.php";
require "util/is-logged-in.php";
session_start();

if(empty($_SESSION['user'])) {
   header("Location: index.php");
}

require 'services/property/delete-property.phtml';
```

Edit-property.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";


if(empty($_SESSION['user'])) {
   header("Location: index.php");
}

require 'services/property/get-property.php' . $_GET['property_id'];

require 'public/property/edit-property.phtml';
```

Email-verification.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";

require 'util/unset-session-prop.php';
require 'public/user/email-verification.phtml';
```

Email-verification-success.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";
require 'public/user/email-verification-success.phtml';
```

Login.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";

require 'util/unset-session-prop.php';
require 'public/user/login.phtml';
unsetSessionProp('error');

?>
```

My-properties.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";


if(empty($_SESSION['user'])) {
   header("Location: index.php");
}

require 'public/property/my-properties.phtml';
```

Property.php

```
<?php
session_start();
require "util/is-logged-in.php";
require "util/is-admin.php";



$property = null;

require 'services/property/get-property.php';

require 'public/property/property.phtml';
```

Register.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-logged-in.php";
require "util/is-admin.php";


require 'public/user/register.phtml';
?>
```

Reset-password.php

```
<?php

session_start();
require "util/flash.php";
require "util/is-logged-in.php";
require 'util/is-get.php';
require "util/is-admin.php";
require 'util/unset-session-prop.php';
require 'config/conn.php';


if(is_get()) {
   $token = $_GET["token"];
   $token_hash = hash("sha256", $token);

   try {
       $conn = $createConnection();

       $query = "SELECT * FROM user WHERE reset_token_hash = :token";
       $statement = $conn->prepare($query);

       // Bind parameters
       $statement->bindParam(':token', $token_hash);


       // Execute the SQL statement
       $statement->execute();

       $user = $statement->fetch(PDO::FETCH_ASSOC);

       if (!$user) {
           die("Token not found");
       }

       if(strtotime($user["reset_token_expires_at"] <= time())) {
           die("Token has expired");
       }

       $_SESSION['token'] = $token;

       require 'public/user/reset-password.phtml';
   } catch (Exception $e) {
       // Handle exceptions
       echo "Error: " . $e->getMessage();
   } finally {
       // Close the database connection
       $conn = null;
   }
   //unsetSessionProp('token');
}
?>
```

Reset-passsword-start.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-logged-in.php";
require 'util/unset-session-prop.php';
require 'public/user/reset-password-start.phtml';

unsetSessionProp("is-password-reset-email-valid");
?>
```

Reset-password-success.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-logged-in.php";
require 'util/unset-session-prop.php';
require "util/is-admin.php";


require 'public/user/reset-password-success.phtml';
unsetSessionProp("token");
```

Statistics.php

```
<?php
session_start();
require "util/flash.php";
require "util/is-admin.php";
require "util/is-logged-in.php";
require 'util/redirect-if-not-admin.php';


if (!isset($_SESSION["user"]) || $_SESSION["role_id"] != 2) {
    // Redirect the user to the login page or display an error message
    // For example:
    header("Location: login.php");
    exit;
}
```

```
require 'services/admin/get-statistics.php';
require 'public/admin/statistics.phtml';
```

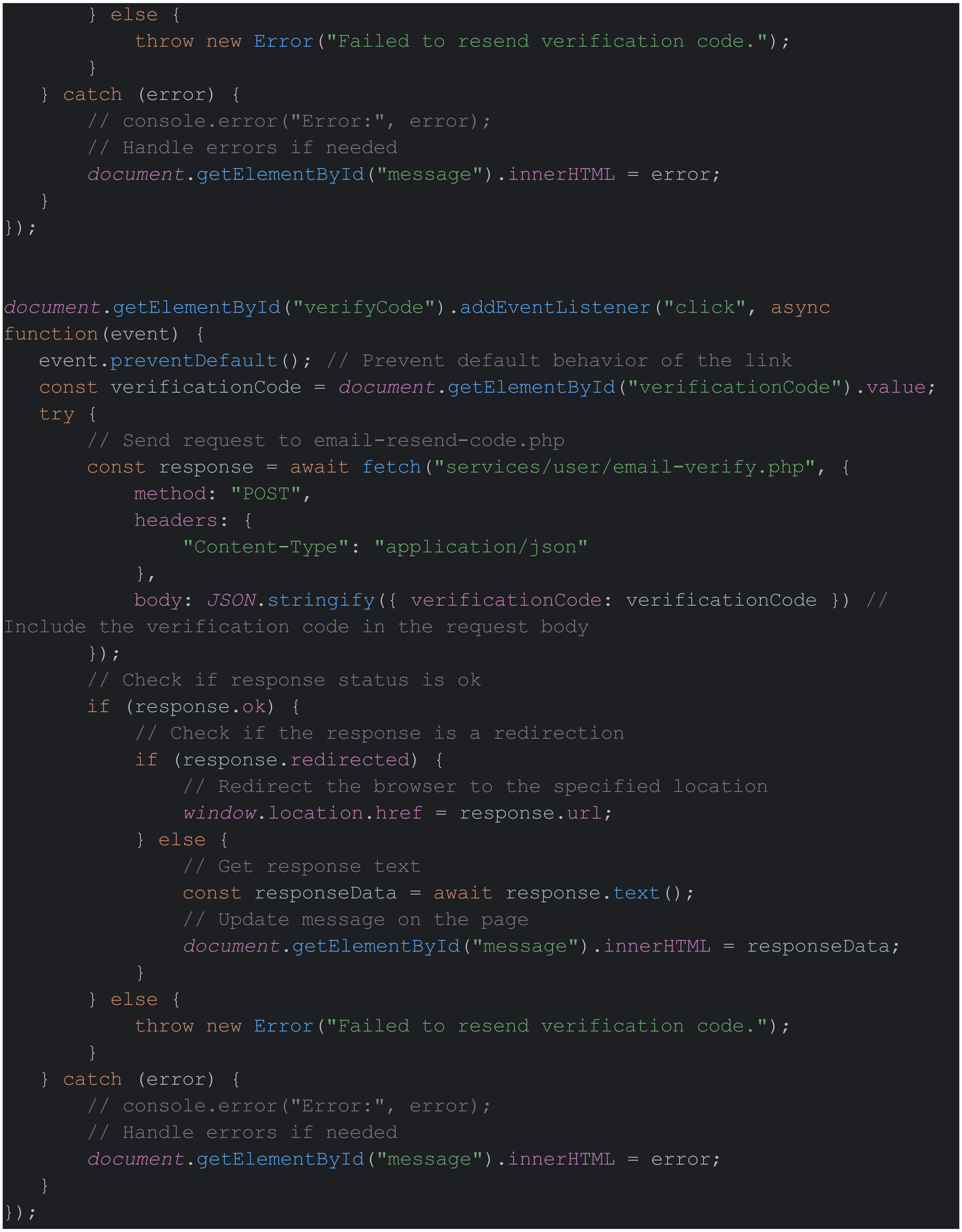
account.js

```


// User info
function initializeAccount(userFullName) {
   // Use userFullName in your script

   // Example usage
   const userImgs = document.querySelectorAll(".userImg");
   userImgs.forEach(img => {
       console.log(123)
       img.src = getAvatarImage(userFullName);
   })

}

function getAvatarImage(fullName) {
   // Split the full name into first name and last name
   console.log(fullName)
   const [firstName, lastName] = fullName.split(" ");

   // Construct the URL to fetch the avatar image
   const apiUrl =
`https://ui-avatars.com/api/?background=random&color=fff&name=${firstName}+${la
stName}`;

   // Return the URL
   return apiUrl;
}

// Call the function when the DOM content is loaded
document.addEventListener("DOMContentLoaded", () => {
   const fullNameElement = document.getElementById("userName");
   const userFullName = fullNameElement.textContent.trim();
   initializeAccount(userFullName);
});





// User option navigation
function showSection(sectionName) {
   const sections = document.querySelectorAll('.user-section');
```

```
    sections.forEach(section => {
        if (section.id === `${sectionName}Section`) {
            section.classList.remove('hidden');
        } else {
            section.classList.add('hidden');
        }
    });

    const buttons = document.querySelectorAll('.account-options-navigation button');
    buttons.forEach(button => {
        if (button.id === `${sectionName}Btn`) {
            button.classList.add('border', 'border-indigo-600', 'border-2' , 'bg-indigo-100');
        } else {
            button.classList.remove('border', 'border-indigo-600', 'border-2', 'bg-indigo-100');
        }
    });
}


document.querySelectorAll('.account-options-navigation button').forEach(btn => {
    btn.addEventListener('click', function() {
        // Get the section name from the button's id
        const sectionName = this.id.replace('Btn', ''); // Remove 'Btn' from the button's id
        showSection(sectionName); // Call the showSection function with the section name
    });
});
```

email-verification.js

```
// Add event listener to the "Resend Code" link
document.getElementById("resendCode").addEventListener("click", async function(event) {
    event.preventDefault(); // Prevent default behavior of the link
    try {
        // Send request to email-resend-code.php
        const response = await fetch("services/user/email-resend-code.php", {
            method: "GET"
        });
        // Check if response status is ok
        if (response.ok) {
            // Get response text
            const responseData = await response.text();
            // Update message on the page
            document.getElementById("message").innerHTML = responseData;
```



```
        } else {
            throw new Error("Failed to resend verification code.");
        }
    } catch (error) {
        // console.error("Error:", error);
        // Handle errors if needed
        document.getElementById("message").innerHTML = error;
    }
});


document.getElementById("verifyCode").addEventListener("click", async
function(event) {
    event.preventDefault(); // Prevent default behavior of the link
    const verificationCode = document.getElementById("verificationCode").value;
    try {
        // Send request to email-resend-code.php
        const response = await fetch("services/user/email-verify.php", {
            method: "POST",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({ verificationCode: verificationCode }) //
Include the verification code in the request body
        });
        // Check if response status is ok
        if (response.ok) {
            // Check if the response is a redirection
            if (response.redirected) {
                // Redirect the browser to the specified location
                window.location.href = response.url;
            } else {
                // Get response text
                const responseData = await response.text();
                // Update message on the page
                document.getElementById("message").innerHTML = responseData;
            }
        } else {
            throw new Error("Failed to resend verification code.");
        }
    } catch (error) {
        // console.error("Error:", error);
        // Handle errors if needed
        document.getElementById("message").innerHTML = error;
    }
});
```