

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I  
KOMUNIKACIONE TEHNOLOGIJE  
INTERNET TEHNOLOGIJE

Projektni zadatak

Tema: Razvoj blog sajta web aplikacije

Predmet: Praktikum iz PHP-a

Student: Jana Matović 139/2019

Beograd, 2022.

# Sadržaj

|  |    |
|--|----|
| 1. Opis aplikacije.....                | 4  |
| 1.1. Opis funkcionalnosti .....        | 4  |
| 2. Struktura aplikacije .....          | 5  |
| 3. Struktura baze .....                | 7  |
| 4. Stranice .....                      | 8  |
| 4.1. home.php .....                    | 8  |
| 4.2. show.php.....                     | 9  |
| 4.3. add.php .....                     | 12 |
| 4.4. edit.php .....                    | 12 |
| 4.5. profile.php.....                  | 13 |
| 4.6. admin.php .....                   | 13 |
| 4.7. about.php .....                   | 14 |
| 4.8. login.php .....                   | 14 |
| 4.9. register.php .....                | 14 |
| 5. Kod.....                            | 16 |
| 5.1. assets.css .....                  | 16 |
| 5.2. main.js.....                      | 17 |
| 5.3. admin.js.....                     | 21 |
| 5.4. dogs.js .....                     | 22 |
| 5.5. favorites.js .....                | 36 |
| 5.6. users.js .....                    | 37 |
| 5.7. .env .....                        | 38 |
| 5.8. config.php.....                   | 38 |
| 5.9. connection.php.....               | 39 |
| 5.10. models/comments/delete.php.....  | 40 |
| 5.11. models/comments/insert.php.....  | 41 |
| 5.12. models/comments/update.php ..... | 43 |
| 5.13. models/dogs/delete.php.....      | 44 |

|   |    |
|---|----|
| 5.14. models/dogs/filter.php .....      | 45 |
| 5.15. models/dogs/get.php.....          | 47 |
| 5.16. models/dogs/getAll.php.....       | 47 |
| 5.17. models/dogs/insert.php.....       | 48 |
| 5.18. models/dogs/show.php .....        | 50 |
| 5.19. models/dogs/update.php .....      | 52 |
| 5.20. models/favorites/delete.php ..... | 54 |
| 5.21. models/favorites/insert.php ..... | 55 |
| 5.22. models/users/admin.php.....       | 56 |
| 5.23. models/users/insert.php.....      | 59 |
| 5.24. models/users/login.php .....      | 62 |
| 5.25. models/users/profile.php .....    | 64 |
| 5.26. views/fixed/footer.php .....      | 65 |
| 5.27. views/fixed/head.php .....        | 66 |
| 5.28. views/dogs/add.php .....          | 68 |
| 5.29. views/dogs/edit.php .....         | 69 |
| 5.30. views/dogs/show.php.....          | 70 |
| 5.31. views/about.php.....              | 70 |
| 5.32. views/admin.php .....             | 72 |
| 5.33. views/home.php.....               | 73 |
| 5.34. views/login.php .....             | 74 |
| 5.35. views/logout.php.....             | 75 |
| 5.36. views/profile.php.....            | 75 |
| 5.37. views/register.php .....          | 77 |
| 5.38. views/index.php .....             | 78 |

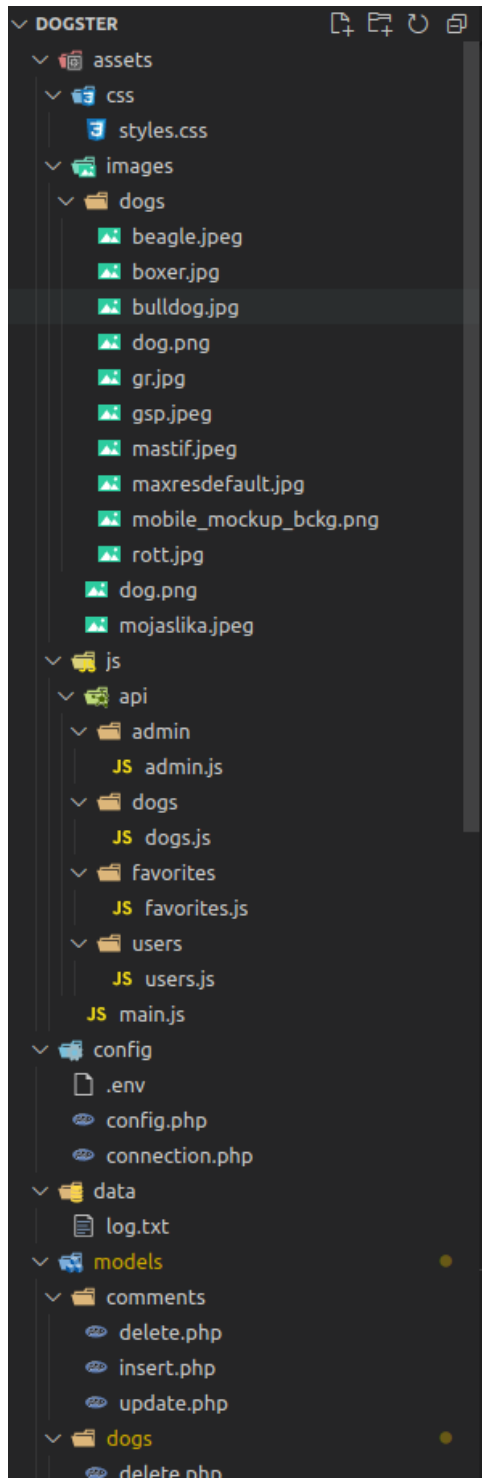
## 1. Opis aplikacije

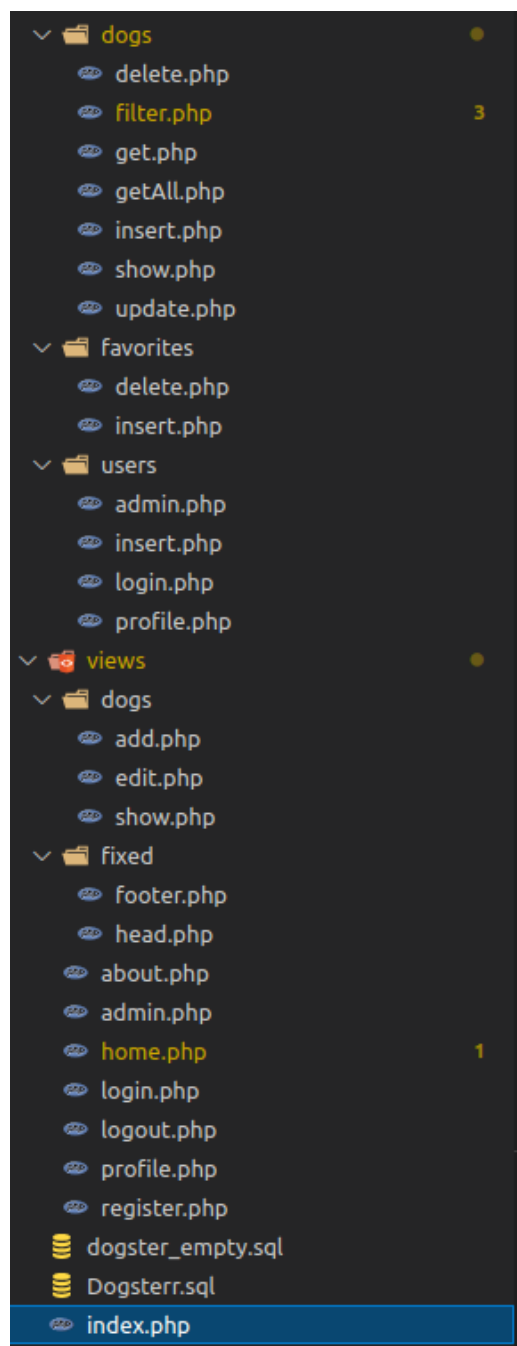
Aplikacija je pisana u programskom jeziku PHP. Na front delu je korišćen HTML, CSS, Bootstrap, Javascript(jQuery), za bazu je korišćena MySQL baza podataka. Unutar aplikacije korisnici se mogu registrovati i prijaviti na sajt korišćenjem svoje email adrese ili korisničkog imena i izabrane šifre. Aplikacija predstavlja mini blog gde korisnici mogu komentarisati i lajkovati pse (post-ove) koje je administrator okačio. Pored običnih korisnika postoje administratori koji imaju pristup svemu i mogu sve da menjaju, takođe imaju posebnu stranicu kojoj je ograničen pristup i na kojoj se ispisuje statistika pristupa stranicama aplikacije.

### 1.1. Opis funkcionalnosti

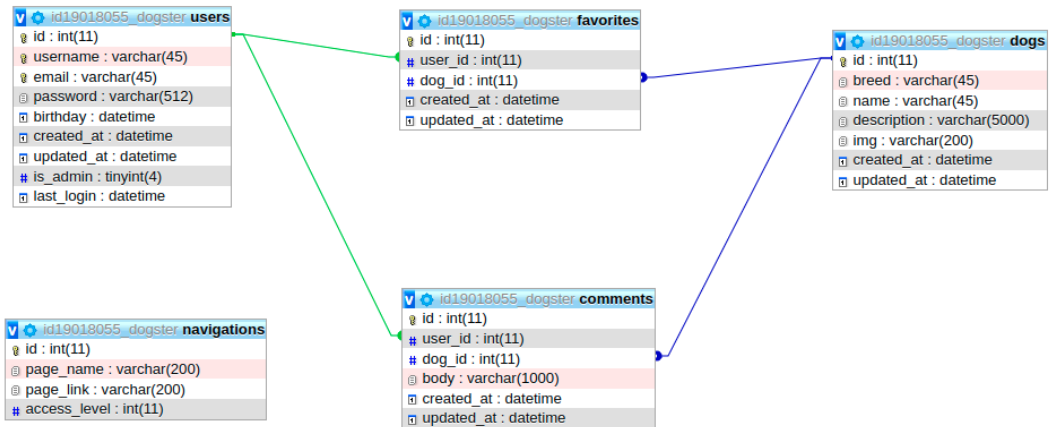
- Dodavanje, izmena, prikaz, brisanje pasa (postova)
- Dodavanje, izmena, prikaz, brisanje komentara
- Registracija i logovanje korisnika
- Filtriranje i pretraga pasa
- Profil stranica
- Admin statistika sajta

## 2. Struktura aplikacije






### 3. Struktura baze



## 4. Stranice

### 4.1. home.php


Stranica za prikaz svih pasa (postova). Odrađena je paginacija i filtriranje (sortiranje i pretraga).

Dogster  Home About me

Login Register

Sort by  
Date


Search  
Search



**Dzeki**  
GOLDEN RETRIEVER

The Golden Retriever is a large-sized breed of dog bred as gun dogs to retrieve shot waterfowl such as ducks and upland game birds during hunting and shooting parties, and were named 'retriever' because ... Wikipedia Life span: 10 – 12 years Weight: ...


[Read more](#)



**Rea**  
GERMAN SHEPHERD

The German Shepherd is a breed of medium to large-sized working dog that originated in Germany. The breed's officially recognized name is German Shepherd Dog in the English language. The breed was once known as the Alsatian in Britain and Ireland. Wi...


[Read more](#)





**Beki**  
ROTTWEILER

The Rottweiler is a breed of domestic dog, regarded as medium-to-large or large. The dogs were known in German as Rottweiler Metzgerhund, meaning Rottweil butchers' dogs, because their main use was to ... Wikipedia Life span: 8 – 10 years Origin: Ger...

[Read more](#)


[Read more](#)

[Read more](#)



to ... Wikipedia Life span: 8 – 10 years Origin: Ger...


[Read more](#)



**Megi**  
BULLDOG

A Bulldog is a medium-sized breed of dog commonly referred to as the English Bulldog or British Bulldog. It is a muscular, hefty dog with a wrinkled face and a distinctive pushed-in nose. Wikipedia Life span: 8 – 10 years Temperament: Docile, Willful....


[Read more](#)



**Dugi**  
BEAGLE

The beagle is a breed of small hound that is similar in appearance to the much larger foxhound. The beagle is a scent hound, developed primarily for hunting hare. Life span: 12 – 15 years Temperament: Amiable, Gentle, Determined, Excitable, Intellige....

[Read more](#)



**Borac**  
BOXER

The Boxer is a medium-sized, short-haired breed of dog, developed in Germany. The coat is smooth and tight-fitting; colors are fawn or brindle, with or without white markings, and white. Life span: 10 – 12 years Temperament: Devoted, Friendly, Fearl....

[Read more](#)

1 2


Copyright © 2022  
Created by Jana Matović

localhost/dogster/index.php?page=dogs&id=21

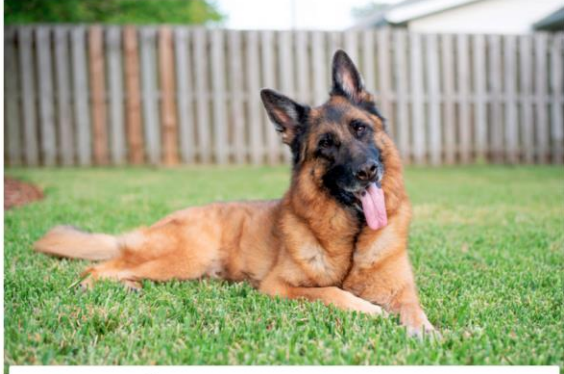
### 4.2. show.php



Stranica za detaljan prikaz podataka o izabranom psu. Ispis komentara i moguće dodavanje novih, dodavanje psa u omiljene, editovanje i brisanje psa, komentara, lajkova.

DogsterHomeAbout meAdd New Dog

jana



AdminCreated at: 2022-05-28 23:24:25h5 comments2 favorites

EditDelete

**Rea**  
GERMAN SHEPHERD

AdminCreated at: 2022-05-28 23:24:25h5 comments2 favorites



EditDelete

**Rea**  
GERMAN SHEPHERD



The German Shepherd is a breed of medium to large-sized working dog that originated in Germany. The breed's officially recognized name is German Shepherd Dog in the English language. The breed was once known as the Alsatian in Britain and Ireland. Wikipedia Life span: 9 – 13 years Temperament: Obedient, Curious, Loyal, Alert, Confident, intelligent, Watchful, Courageous Weight: Male: 30-40 kg, Female: 22-32 kg Colors: Black, Black & Tan, Red & Black, Black & Silver, Sable, Grey Height: Male: 60-65 cm, Female: 55-60 cm Did you know: German Shepherd is the second-most popular dog breed by registrations (129,186) worldwide.

Remove from favorites



CommentsAdd New Comment

janaCreated at: 2022-05-30 15:37:10



Nice

jana2Created at: 2022-05-29 21:28:37

Komentar 2

janaCreated at: 2022-05-29 15:43:58

Komentar

janaCreated at: 2022-05-29 15:39:35

Komentar 3

9

Comments

+ Add New Comment

Comment \*

Enter comment

Submit

👤 jana



Created at: 2022-05-30 15:37:10

Nice

Nice

Edit

👤 jana2



Created at: 2022-05-29 21:28:37

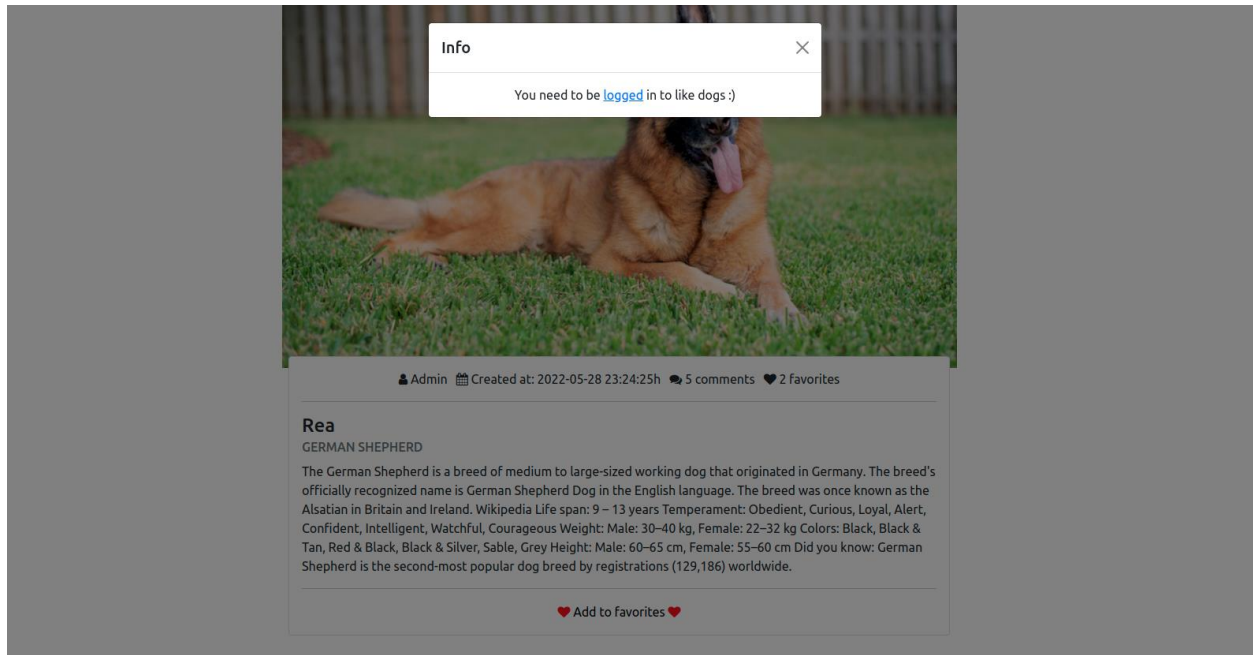
Komentar 2

👤 jana



Created at: 2022-05-29 15:43:58

Komentar



Comments

+ Add New Comment

Please [login](#) to leave a comment.

jana

Nice


Created at: 2022-05-30 15:37:10

jana2

Created at: 2022-05-29 21:28:37

## 4.3. add.php

Stranica za dodavanje novog psa.

Dogster  Home About me Add New Dog

jana ▾

### Submit new dog

Fields marked with \* are required.

Breed \*

Name \*


Description \*

Upload Image \*  
 No file chosen

Copyright © 2022  
Created by Jana Matović

## 4.4. edit.php

Stranica za ažuriranje podataka o postojećem psu.

Dogster  Home About me Add New Dog

jana ▾

### Edit "Rea"

Fields marked with \* are required.

Breed \*

Name \*

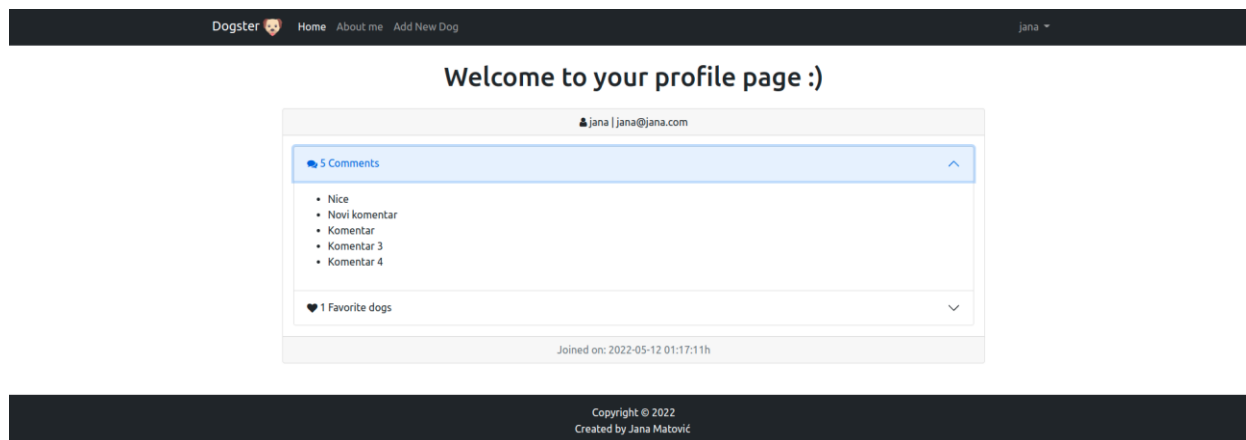
Description \*

Upload Image  
 No file chosen

Copyright © 2022  
Created by Jana Matović

## 4.5. profile.php

Stranica na kojoj se nalaze podaci ulogovanog korisnika, lista njegovih komentara i omiljenih pasa.



## 4.6. admin.php

Stranica na kojoj se nalazi statistika pristupa stranicama, broj ulogovanih korisnika u poslednja 24h.

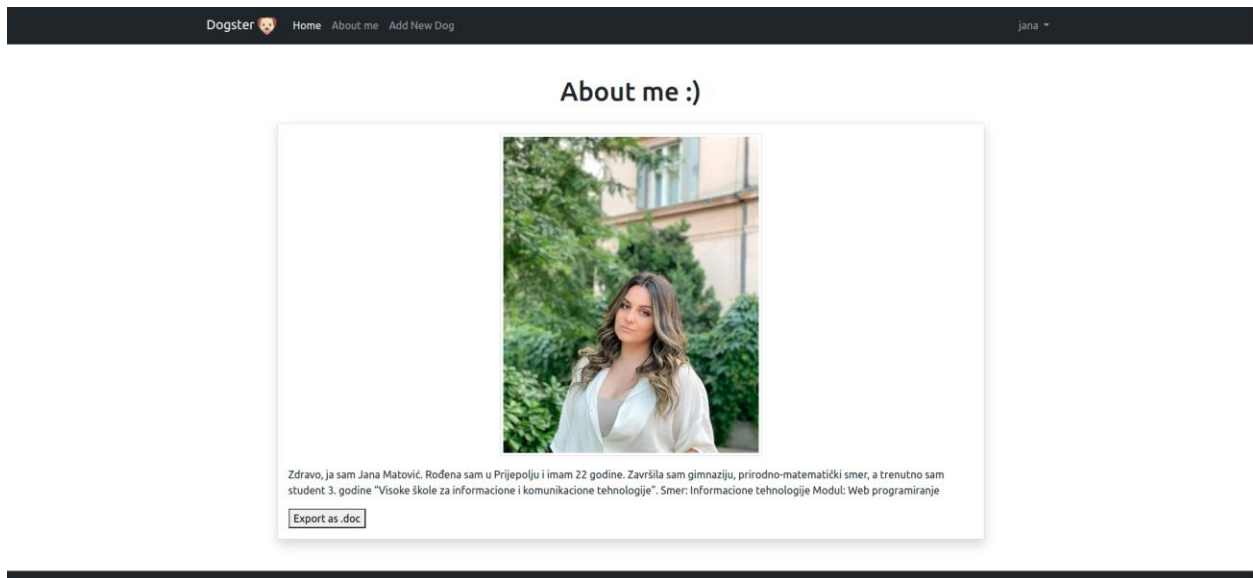
The screenshot shows the admin page with a table titled 'Pages statistics'. The table has three columns: NAME, TOTAL VISITS, and LAST 24h VISITS. The data is as follows:

| NAME        | TOTAL VISITS | LAST 24h VISITS |
|-------------|--------------|-----------------|
| index       | 460 (64.25%) | 403 (67.85%)    |
| admin       | 72 (10.06%)  | 44 (7.41%)      |
| about       | 6 (0.84%)    | 3 (0.51%)       |
| dogs-edit   | 24 (3.35%)   | 22 (3.7%)       |
| dogs        | 99 (13.83%)  | 89 (14.98%)     |
| add-new-dog | 5 (0.7%)     | 3 (0.51%)       |
| login       | 14 (1.96%)   | 6 (1.01%)       |
| register    | 3 (0.42%)    | 1 (0.17%)       |
| profile     | 23 (3.21%)   | 20 (3.37%)      |
| logout      | 10 (1.4%)    | 3 (0.51%)       |

Below the table, it states 'Number of users in last 24h: 1'. The footer contains copyright information: 'Copyright © 2022 Created by Jana Matović'.

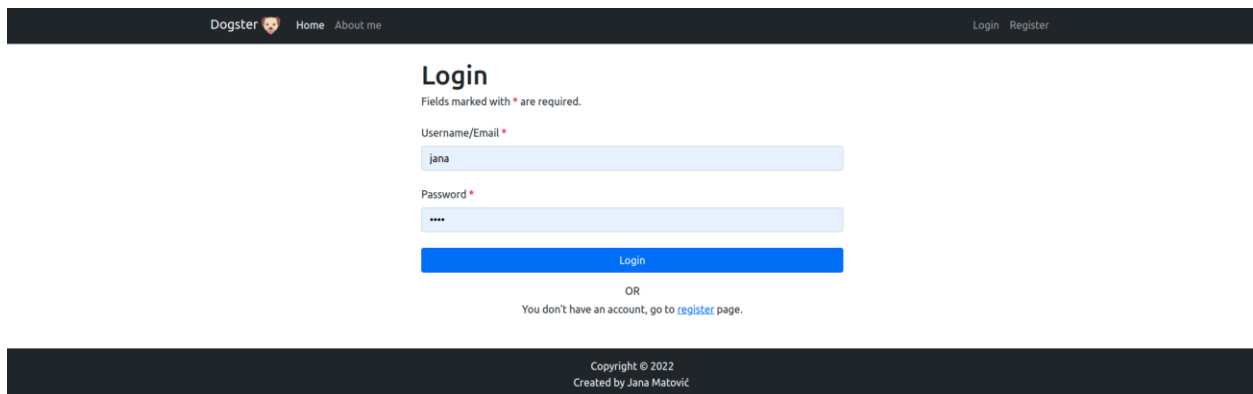
## 4.7. about.php

Stranica o meni.



## 4.8. login.php

Stranica za logovanje korisnika na sajt.



## 4.9. register.php

Stranica za registrovanje korisnika na sajt.

## Register

Fields marked with \* are required.

Username \*

Email \*

Password \*

Confirm password \*

Birthday

Register

OR

You already have account, go to [login](#) page.

## 5. Kod

### 5.1. assets.css

```
footer {  
  color: white;  
  padding-top: 15px;  
  padding-bottom: 15px;  
  margin-top: 10px;  
}
```

```
footer p {  
  margin: 0;  
  line-height: 25px;  
}
```

```
.dog-logo {  
  width: 30px;  
}
```

```
.error-msg {  
  color: red;  
  font-weight: bold;  
}
```

```
.dog-card a.dog-name {  
  text-decoration: none;  
  color: black;  
}
```

```
.dog-card-show {  
  width: 98%;  
  margin: 0 auto;  
  margin-top: -15px;  
}
```

```
.card-please-login {  
  width: 90%;  
  margin: 0 auto;  
  margin-top: 25px;  
  border-left: 5px solid gray;  
  margin-bottom: 10px;
```



```
}
```

```
.add-to-favorites:hover {  
  cursor: pointer;  
}
```

```
.add-to-favorites:hover i {  
  color: red;  
}
```

```
.mojaslika {  
  transition: transform .3s;  
}
```

```
.mojaslika:hover {  
  transform: scale(1.2);  
  cursor: pointer;  
}
```

```
.image-container {  
  margin: 0 auto;  
  margin-top: 15px;  
  overflow: hidden;  
}
```

## 5.2. main.js

```
let registerForm = document.getElementById('register-form');  
let loginForm = document.getElementById('login-form');
```

```
if(registerForm){  
  registerForm.addEventListener('submit', (e) => {  
    e.preventDefault();  
    let username = (document.querySelector(`input[name="username"]`)).value;  
    let email = (document.querySelector(`input[name="email"]`)).value;  
    let password = (document.querySelector(`input[name="password"]`)).value;  
    let confirmPassword = (document.querySelector(`input[name="password2"]`)).value;  
    let birthday = (document.querySelector(`input[name="birthday"]`)).value;  
    $.ajax({  
      url: './models/users/insert.php',  
      type: 'POST',  
      data: {  
        username,  
        email,
```



```

console.log(response)
console.log(response.responseText)
let errorsDiv = document.getElementById('login-errors');
let responseTextParsed = JSON.parse(response.responseText);
let errors = responseTextParsed['errors'];
let errorKeys = Object.keys(responseTextParsed["errors"]);
errorsDiv.innerHTML = "";
errorKeys.forEach(err => {
errorsDiv.innerHTML += errors[err];
});
}
});
}
}
}
}

```

```

let searchInput = document.querySelector('.search-input');

```

```

if(searchInput){
searchInput.addEventListener('keyup', (e) => {

```

```

$.ajax({
url: './models/dogs/filter.php',
type: 'POST',
data: {
searchValue: searchInput.value,
sortValue: sortInput.value,
pagination: 1,
submit: true,
},
dataType: "json",
success: (response) => {
createPagination(response.totalPages);
console.log(response.dogs)
let dogs = response.dogs;
let dogsDiv = document.getElementById('dogs');
dogsDiv.innerHTML = "";

dogs.forEach(dog => {
dogsDiv.innerHTML += `<div class="col-md-4 mb-3">
<div class="card dog-card">
<a href="index.php?page=dogs&id=${dog.id}"></a>
<div class="card-body">
<a class="dog-name" href="index.php?page=dogs&id=${dog.id}">

```

```

<h5 class="card-title">${ dog.name }</h5>
</a>
<h6 class="card-subtitle mb-2 text-muted">${ dog.breed }</h6>
<p class="card-text">${ dog.description.substring(0, 250) }...</p>
<a href="index.php?page=dogs&id=${ dog.id }" class="btn btn-sm btn-secondary">Read more</a>
</div>
</div>
</div>`;
});
};
error: (response) => {
console.log(response)
console.log(response.responseText)
}
});
}}
}

```

```

let sortInput = document.querySelector('.sort-input');

```

```

if(sortInput){
sortInput.addEventListener('change', (e) => {
$.ajax({
url: './models/dogs/filter.php',
type: 'POST',
data: {
sortValue: sortInput.value,
searchValue: searchInput.value,
pagination: 1,
submit: true,
},
dataType: "json",
success: (response) => {
createPagination(response.totalPages);
console.log(response.dogs)
let dogs = response.dogs;
let dogsDiv = document.getElementById('dogs');
dogsDiv.innerHTML = "";

dogs.forEach(dog => {
dogsDiv.innerHTML += `<div class="col-md-4 mb-3">
<div class="card dog-card">
<a href="index.php?page=dogs&id=${ dog.id }"></a>
<div class="card-body">

```

```

<a class="dog-name" href="index.php?page=dogs&id=${ dog.id}">
<h5 class="card-title">${ dog.name}</h5>
</a>
<h6 class="card-subtitle mb-2 text-muted">${ dog.breed}</h6>
<p class="card-text">${ dog.description.substring(0, 250)}....</p>
<a href="index.php?page=dogs&id=${ dog.id}" class="btn btn-sm btn-secondary">Read more</a>
</div>
</div>
</div>`;
});
},
error: (response) => {
console.log(response)
console.log(response.responseText)
}
});
})
}

```

### 5.3. admin.js

```
let pagesStat = document.querySelector('.pages-stat');
```

```
let activeUsers = document.querySelector('.active-users');
```

```

$.ajax({
url: './models/users/admin.php',
type: 'GET',
success: (response) => {
let pages = Object.keys(response.pages);

pages.forEach(page => {
pagesStat.innerHTML += `<tr>
<td>${ page}</td>
<td>${ response.pages[page].total } (${Math.round(response.pages[page].total / response.total * 100 * 100) / 100}%)</td>
<td>${ response.pages[page].last24 } (${Math.round(response.pages[page].last24 / response.total24 * 100 * 100) / 100}%)</td>
</tr>`;
});
}

```

```

activeUsers.innerHTML = `Number of users in last 24h: ${response.users.length}`
},
error: (response) => {
console.log(response)
}

```

```

console.log(response.responseText)
let errorsDiv = document.getElementById('add-new-dog-errors');
let responseTextParsed = JSON.parse(response.responseText);
let errors = responseTextParsed['errors'];
let errorKeys = Object.keys(responseTextParsed["errors"]);

```

```

errorsDiv.innerHTML = "";

```

```

errorKeys.forEach(err => {
errorsDiv.innerHTML += errors[err];
});
}
});

```

## 5.4. dogs.js

```

// EDIT DOGS

```

```

=====

```

```

let editDiv = document.querySelector('.edit-dog');

```

```

if (editDiv) {
const urlParams = new URLSearchParams(window.location.search);
const id = urlParams.get('id');

```

```

$.ajax({
url: './models/dogs/get.php?id=' + id,
type: 'GET',
success: (response) => {
editDiv.innerHTML = `<h1 class="mb-1">Edit "${response.dog.name}" &#128021;</h1>
<p class="mb-4">Fields marked with <strong class="text-danger">*</strong> are required.</p>
<div id="edit-dog-errors" class="mb-3"></div>
<form id="edit-dog-form">
<div class="mb-4">
<label class="form-label">Breed <strong class="text-danger">*</strong></label>
<input name="breed" type="text" class="form-control" placeholder="Enter breed"
value="${response.dog.breed}">
</div>
<div class="mb-4">
<label class="form-label">Name <strong class="text-danger">*</strong></label>
<input name="name" type="text" class="form-control" placeholder="Enter name"
value="${response.dog.name}">
</div>
<div class="mb-4">
<label class="form-label">Description <strong class="text-danger">*</strong></label>

```

```

<textarea class="form-control" name="description" placeholder="Enter description" id="floatingTextarea2"
style="height: 100px">${response.dog.description}</textarea>
</div>
<input type="hidden" name="dogID" value="${response.dog.id}">
<div class="mb-4">
<label class="form-label">Upload Image</label> <br>
<input type="file" name="file">
</div>
<button type="submit" class="btn btn-secondary w-100">Submit</button>
</form>;

```

```

let editDogForm = document.getElementById('edit-dog-form');

```

```

editDogForm.addEventListener('submit', (e) => {
e.preventDefault();
console.log("A");

```

```

let formData = new FormData();
formData.append('dogID', (document.querySelector(`input[name="dogID"]`)).value);
formData.append('breed', (document.querySelector(`input[name="breed"]`)).value);
formData.append('name', (document.querySelector(`input[name="name"]`)).value);
formData.append('breedDescription', (document.querySelector(`textarea[name="description"]`)).value);
formData.append('breedImage', (document.querySelector(`input[name="file"]`)).files[0]);
formData.append('submit', true);

```

```

$.ajax({
url: './models/dogs/update.php',
type: 'POST',
data: formData,
processData: false,
contentType: false,
dataType: "json",
success: (response) => {
alert("Dog Updated Successfully");
window.location.href = './index.php';
},
error: (response) => {
console.log(response)
console.log(response.responseText)
let errorsDiv = document.getElementById('edit-dog-errors');
let responseTextParsed = JSON.parse(response.responseText);
let errors = responseTextParsed['errors'];
let errorKeys = Object.keys(responseTextParsed["errors"]);

```

```
errorsDiv.innerHTML = "";
```

```
errorKeys.forEach(err => {  
  errorsDiv.innerHTML += errors[err];  
});  
}  
});  
})
```

```
},  
error: (response) => {  
  console.log(response)  
  console.log(response.responseText)  
  let errorsDiv = document.getElementById('add-new-dog-errors');  
  let responseTextParsed = JSON.parse(response.responseText);  
  let errors = responseTextParsed['errors'];  
  let errorKeys = Object.keys(responseTextParsed["errors"]);
```

```
errorsDiv.innerHTML = "";
```

```
errorKeys.forEach(err => {  
  errorsDiv.innerHTML += errors[err];  
});  
}  
});  
}
```

```
// ADD DOGS
```

```
let submitNewDogForm = document.getElementById('submit-new-dog-form');
```

```
if (submitNewDogForm) {  
  submitNewDogForm.addEventListener('submit', (e) => {  
    e.preventDefault();
```

```
    let formData = new FormData();  
    formData.append('breed', (document.querySelector(`input[name="breed"]`)).value);  
    formData.append('name', (document.querySelector(`input[name="name"]`)).value);  
    formData.append('breedDescription', (document.querySelector(`textarea[name="description"]`)).value);  
    formData.append('breedImage', (document.querySelector(`input[name="file"]`)).files[0]);  
    formData.append('submit', true);
```

```
$.ajax({
```



```

url: './models/dogs/insert.php',
type: 'POST',
data: formData,
processData: false,
contentType: false,
dataType: "json",
success: (response) => {
  alert("Dog Added Successfully");
  window.location.href = './index.php';
},
error: (response) => {
  console.log(response)
  console.log(response.responseText)
  let errorsDiv = document.getElementById('add-new-dog-errors');
  let responseTextParsed = JSON.parse(response.responseText);
  let errors = responseTextParsed['errors'];
  let errorKeys = Object.keys(responseTextParsed["errors"]);

  errorsDiv.innerHTML = "";

  errorKeys.forEach(err => {
    errorsDiv.innerHTML += errors[err];
  });
}
});
})
}

```

```
// DELETE DOGS
```

```

function generateDeleteDogEvent(){
  let deleteDogBtn = document.querySelector('.delete-dog');

  if (deleteDogBtn) {
    deleteDogBtn.addEventListener('click', (e) => {
      let dogID = deleteDogBtn.getAttribute('data-dog-id');
      $.ajax({
        url: './models/dogs/delete.php',
        type: 'POST',
        data: {
          dogID,
        },
        submit: true,
      },

```

```

dataType: "json",
success: (response) => {
  alert("Dog Deleted Successfully");
  window.location.href = './index.php';
},
error: (response) => {
  console.log(response)
  console.log(response.responseText)
}
});
});
}
}

```

```
// SHOW ALL DOGS HOME
```

```
let dogsDivG = document.getElementById('dogs');
```

```

if (dogsDivG) {
$.ajax({
  url: './models/dogs/getAll.php?pagination=1',
  type: 'GET',
  success: (response) => {
    createPagination(response.totalPages);

    response.dogs.forEach(dog => {
      dogsDivG.innerHTML += `
<div class="col-md-4 mb-3">
<div class="card dog-card">
<a href="index.php?page=dogs&id=${dog.id}"></a>
<div class="card-body">
<a class="dog-name" href="index.php?page=dogs&id=${dog.id}">
<h5 class="card-title">${dog.name}</h5>
</a>
<h6 class="card-subtitle mb-2 text-muted">${dog.breed}</h6>
<p class="card-text">${dog.description.substring(0, 250)}....</p>
<a href="index.php?page=dogs&id=${dog.id}" class="btn btn-sm btn-secondary">Read more</a>
</div>
</div>
</div>`
    });
  }
});
}

```

```

    },
    error: (response) => {
      console.log(response)
      console.log(response.responseText)
    }
  });
}

```

```

function createPagination(totalPages) {
  let pagination = document.querySelector('.pagination');
  let html = ``;

```

```

  for(let i = 0; i < totalPages; i++){
    html += `<li class="page-item"><a data-page="${i+1}" class="page-link" href="#">${i+1}</a></li>`;
  }

```

```

  pagination.innerHTML = html;

```

```

  let lis = document.querySelectorAll('.page-link');

```

```

  lis.forEach(li => {
    li.addEventListener('click', (e) => {
      $.ajax({
        url: './models/dogs/filter.php',
        type: 'POST',
        data: {
          sortValue: sortInput.value,
          searchValue: searchInput.value,
          pagination: li.getAttribute('data-page'),
          submit: true,
        },
        dataType: "json",
        success: (response) => {
          console.log(response.dogs)
          let dogs = response.dogs;
          let dogsDiv = document.getElementById('dogs');
          dogsDiv.innerHTML = "";
          dogs.forEach(dog => {
            dogsDiv.innerHTML += `<div class="col-md-4 mb-3">
              <div class="card dog-card">
                <a href="index.php?page=dogs&id=${dog.id}"></a>
                <div class="card-body">
                  <a class="dog-name" href="index.php?page=dogs&id=${dog.id}">
                    <h5 class="card-title">${dog.name}</h5>

```

```

</a>
<h6 class="card-subtitle mb-2 text-muted">${ dog.breed }</h6>
<p class="card-text">${ dog.description.substring(0, 250) }....</p>
<a href="index.php?page=dogs&id=${ dog.id }" class="btn btn-sm btn-secondary">Read more</a>
</div>
</div>
</div>;
});
},
error: (response) => {
console.log(response)
console.log(response.responseText)
}
});
})
})
}
}

```

```
// SHOW DOGS
```

```
let data = document.querySelector('.data');
```

```

if (data) {
const urlParams = new URLSearchParams(window.location.search);
const id = urlParams.get('id');

```

```

$.ajax({
url: './models/dogs/show.php?id=' + id,
type: 'GET',
success: (response) => {
generateHTML(response, data);
generateDeleteDogEvent();
generateDeleteEventListeners();
generateEditEventListeners();
generateFavoritesEvent();
}
}

```

```
let submitNewCommentForm = document.getElementById('submit-new-comment-form');
```

```

submitNewCommentForm.addEventListener('submit', (e) => {
e.preventDefault();

```

```

let comment = (document.querySelector(`textarea[name="comment"]`)).value;
let dog_id = (document.querySelector(`input[name="dog_id"]`)).value;

```

```

$.ajax({
  url: './models/comments/insert.php',
  type: 'POST',
  data: {
    comment,
    dog_id,
    submit: true,
  },
  dataType: 'json',
  success: (response) => {
    console.log(response)
    alert("Comment Added Successfully");
    let allCommentsDIV = document.querySelector('.all-comments');

    let newComment = document.createElement('div');
    newComment.classList.add('comment');
    newComment.classList.add('mb-2');
    newComment.setAttribute('data-div-comment-id', response.comment.id);

    newComment.innerHTML = `<div class="d-flex justify-content-between mb-2">
<div>
<i class="fa fa-user" aria-hidden="true"></i> ${response.comment.username} &nbsp;
<button class="btn btn-sm btn-outline-warning" type="button" data-bs-toggle="collapse" data-bs-
target="#collapseExample_${response.comment.id}" aria-expanded="false" aria-
controls="collapseExample_${response.comment.id}">
<i class="fa fa-pencil-square-o" aria-hidden="true"></i>
</button>
<button data-comment-id="${response.comment.id}" class="btn btn-sm btn-outline-danger delete-
comment"><i class="fa fa-trash" aria-hidden="true"></i></button>
</div>
<div>Created at: ${response.comment.created_at} </div>
</div>
<p data-comment-id="${response.comment.id}" class="card-text">${response.comment.body}</p>
<div class="collapse" id="collapseExample_${response.comment.id}">
<textarea data-comment-id="${response.comment.id}" class="form-control" placeholder="Leave a comment
here" style="height: 100px">${response.comment.body}</textarea>
<button data-comment-id="${response.comment.id}" class="btn btn-warning mt-3 edit-
comment">Edit</button>
</div>
<hr>`;
    allCommentsDIV.prepend(newComment);

    let spanNumberOfComments = document.querySelector('.number-of-comments');
    let numberOfComments = spanNumberOfComments.innerHTML;

```

```
spanNumberOfComments.innerHTML = ++numberOfComments;
```

```
generateDeleteEventListeners();  
generateEditEventListeners();  
(document.querySelector(`textarea[name="comment"]`)).value = "";  
},  
error: (response) => {  
  console.log(response)  
  console.log(response.responseText)  
  let errorsDiv = document.getElementById('comment-errors');  
  let responseTextParsed = JSON.parse(response.responseText);  
  let errors = responseTextParsed['errors'];  
  let errorKeys = Object.keys(responseTextParsed["errors"]);
```

```
errorsDiv.innerHTML = "";
```

```
errorKeys.forEach(err => {  
  errorsDiv.innerHTML += errors[err];  
});  
}  
});  
});  
});  
},
```

```
error: (response) => {  
  console.log(response)  
  console.log(response.responseText)  
  let errorsDiv = document.getElementById('add-new-dog-errors');  
  let responseTextParsed = JSON.parse(response.responseText);  
  let errors = responseTextParsed['errors'];  
  let errorKeys = Object.keys(responseTextParsed["errors"]);
```

```
errorsDiv.innerHTML = "";
```

```
errorKeys.forEach(err => {  
  errorsDiv.innerHTML += errors[err];  
});  
}  
});  
});  
}
```

```
function generateHTML(response, data) {  
  let html = `<div class="card dog-card-show">  
<div class="card-body">  
<div class="text-center">
```

```
<i class="fa fa-user" aria-hidden="true"></i>
Admin &nbsp;
<i class="fa fa-calendar" aria-hidden="true"></i>
Created at: ${response.dog.created_at}h &nbsp;
<i class="fa fa-comments" aria-hidden="true"></i>
<span class="number-of-comments"> ${response.comments.length}</span> comments &nbsp;
<i class="fa fa-heart" aria-hidden="true"></i>
<span class="number-of-favorites"> ${response.favorites.length}</span> favorites
</div>
${response.admin ? `<div class="text-center mt-2">
<a class="btn btn-sm btn-warning" href="index.php?page=dogs-edit&id=${response.dog.id}">Edit</a>
<button data-dog-id="${response.dog.id}" class="btn btn-sm btn-danger delete-dog">Delete</button>
</div>` : ``}
<hr>
<h4 class="card-title">${response.dog.name}</h4>
<h6 class="card-subtitle mb-2 text-muted">${response.dog.breed}</h6>
<p class="card-text">${response.dog.description}</p>
<hr>
<div data-favorite="${response.favorite}" data-dog-id="${response.dog.id}" class="text-center add-to-favorites">
${response.favorite ?
`<i class="fa fa-heart" aria-hidden="true"></i> Remove from favorites <i class="fa fa-heart" aria-
hidden="true"></i>` : `${response.user ? <i class="fa fa-heart" aria-hidden="true"></i> Add to favorites <i
class="fa fa-heart" aria-hidden="true"></i>
` : `<!-- Button trigger modal -->
<button style="border: none; background-color: transparent;" type="button" data-bs-toggle="modal" data-bs-
target="#exampleModal">
<i class="fa fa-heart" aria-hidden="true"></i> Add to favorites <i class="fa fa-heart" aria-hidden="true"></i>
</button>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-
hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Info</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
You need to be <a href="index.php?page=login">logged</a> in to like dogs :)
</div>
</div>
</div>
</div>
```

```

</div>` }` }
</div>
</div>
</div>

```

```

<div id="comments" class="mt-5">
<div class="card">
<div class="card-header d-flex justify-content-between">
<div class="mt-1">
<i class="fa fa-comment" aria-hidden="true"></i> Comments
</div>
<div>
<button class="btn btn-sm btn-secondary" type="button" data-bs-toggle="collapse" data-bs-
target="#collapseExample" aria-expanded="false" aria-controls="collapseExample">
<i class="fa fa-plus" aria-hidden="true"></i>
Add New Comment
</button>

```

```

</div>
</div>
<div class="collapse" id="collapseExample">
${response.user ? ` <div class="card card-body">
<div id="comment-errors" class="mb-3"></div>
<form id="submit-new-comment-form">
<input type="hidden" name="dog_id" value="${response.dog.id}">
<div class="mb-4">
<label class="form-label">Comment <strong class="text-danger">*</strong></label>
<textarea class="form-control" name="comment" placeholder="Enter comment" id="floatingTextarea2"
style="height: 100px"></textarea>
</div>
<button type="submit" class="btn btn-secondary w-100">Submit</button>
</form>
</div>` : ` <div class="card card-body card-please-login">
<p class="mb-0">Please <a href="index.php?page=login">login</a> to leave a comment.</p>
</div>` }
</div>
<div class="card-body all-comments">`

```

```

response.comments.forEach(comment => {
html += `<div data-div-comment-id="${comment.id}" class="comment mb-2">
<div class="d-flex justify-content-between mb-2">
<div>
<i class="fa fa-user" aria-hidden="true"></i> ${comment.username} &nbsp;

```



```

    ${ (response.user && response.user.id == comment.UID) || response.admin ? `<button class="btn btn-sm btn-outline-warning" type="button" data-bs-toggle="collapse" data-bs-target="#collapseExample_${comment.id}"
    aria-expanded="false" aria-controls="collapseExample_${comment.id}">
    <i class="fa fa-pencil-square-o" aria-hidden="true"></i>
    </button>
    <button data-comment-id="${comment.id}" class="btn btn-sm btn-outline-danger delete-comment"><i
    class="fa fa-trash" aria-hidden="true"></i></button>` : `` }
  </div>
</div>
<div>Created at: ${comment.created_at}</div>
</div>
<p data-comment-id="${comment.id}" class="card-text">${comment.body}</p>
<div class="collapse" id="collapseExample_${comment.id}">
  <textarea data-comment-id="${comment.id}" class="form-control" placeholder="Leave a comment here"
  style="height: 100px">${comment.body}</textarea>
  <button data-comment-id="${comment.id}" class="btn btn-warning mt-3 edit-comment">Edit</button>
</div>
<hr>
</div>`;
  })
}

```

```

html += `
</div>
</div>
</div>`

```

```

data.innerHTML = html;
}

```

```

function generateDeleteEventListeners() {
  let deleteCommentBtns = document.querySelectorAll('.delete-comment');

```

```

  deleteCommentBtns.forEach(deleteCommentBtn => {
    deleteCommentBtn.removeEventListener('click', handleCommentDeleteClick);
    deleteCommentBtn.addEventListener('click', handleCommentDeleteClick);
  });
}

```

```

function handleCommentDeleteClick() {
  let commentID = this.getAttribute('data-comment-id');

```

```

$.ajax({
  url: '/models/comments/delete.php',
  type: 'POST',

```

```

data: {
  commentID,
  submit: true,
},
dataType: "json",
success: (response) => {
  alert("Comment Deleted Successfully");
  let element = document.querySelector(`[data-div-comment-id="${commentID}"]`);
  element.remove();
  let spanNumberOfComments = document.querySelector('.number-of-comments');
  let numberOfComments = spanNumberOfComments.innerHTML;
  spanNumberOfComments.innerHTML = --numberOfComments;
},
error: (response) => {
  console.log(response)
  console.log(response.responseText)
}
});
}

```

```

function generateEditEventListeners() {
  let editCommentBtns = document.querySelectorAll('.edit-comment');

  editCommentBtns.forEach(editCommentBtn => {
    editCommentBtn.removeEventListener('click', handleCommentEditClick);
    editCommentBtn.addEventListener('click', handleCommentEditClick);
  });
}

```

```

function handleCommentEditClick() {
  let commentID = this.getAttribute('data-comment-id');
  let comment = (document.querySelector(`textarea[data-comment-id="${commentID}"]`)).value;

```

```

$.ajax({
  url: './models/comments/update.php',
  type: 'POST',
  data: {
    commentID,
    comment,
    submit: true,
  },
  dataType: "json",
  success: (response) => {
    console.log(response)
    alert("Comment Edited Successfully");

```

```

(document.querySelector(`p[data-comment-id="${commentID}"]`)).innerHTML = comment;
},
error: (response) => {
console.log(response)
console.log(response.responseText)
}
});
}

```

## 5.5. favorites.js

```

function generateFavoritesEvent() {
let addToFavorites = document.querySelector('.add-to-favorites');

if (addToFavorites) {
addToFavorites.addEventListener('click', (e) => {
let favorite = addToFavorites.getAttribute('data-favorite');
console.log(favorite);

$.ajax({
url: favorite == "false" ? './models/favorites/insert.php' : './models/favorites/delete.php',
type: 'POST',
data: {
dogID: addToFavorites.getAttribute('data-dog-id'),
submit: true,
},
dataType: "json",
success: (response) => {
console.log(response)
if (favorite == "false") {
alert("Dog Added To Favorites Successfully");
addToFavorites.innerHTML = `<i class="fa fa-heart" aria-hidden="true"></i> Remove from favorites <i
class="fa fa-heart" aria-hidden="true"></i>`;
addToFavorites.setAttribute('data-favorite', 'true');
let spanNumberOfFavorites = document.querySelector('.number-of-favorites');
let numberOfFavorites = spanNumberOfFavorites.innerHTML;
spanNumberOfFavorites.innerHTML = ++numberOfFavorites;

```



```

favoritesNumber.innerHTML = response.favorites.length;

response.favorites.forEach(favorite => {
  favoritesList.innerHTML += `<li><a class="text-decoration-none"
href="index.php?page=dogs&id=${ favorite.dog_id}">${ favorite.name} (${ favorite.breed})</a></li>
`;
});

userJoined.innerHTML = `Joined on: ${response.user.created_at}h`;
},
error: (response) => {
  console.log(response)
  console.log(response.responseText)
  let errorsDiv = document.getElementById('add-new-dog-errors');
  let responseTextParsed = JSON.parse(response.responseText);
  let errors = responseTextParsed['errors'];
  let errorKeys = Object.keys(responseTextParsed["errors"]);

  errorsDiv.innerHTML = "";

  errorKeys.forEach(err => {
    errorsDiv.innerHTML += errors[err];
  });
}
});

```

## 5.7. .env

```

DBNAME=id19018055_dogster
SERVER=localhost
USERNAME=id19018055_root
PASSWORD=q$LARRs?z!>%6lr}

```

## 5.8. config.php

```

<?php

error_reporting(E_ALL);
ini_set('display_errors', 1);

// Osnovna podesavanja
define("BASE_URL", "https://jana-dogster.000webhostapp.com/");
define("ABSOLUTE_PATH", $_SERVER["DOCUMENT_ROOT"]);

```

```

// Ostala podesavanja
define("ENV_FAJL", ABSOLUTE_PATH . "/config/env");
define("LOG_FAJL", ABSOLUTE_PATH . "/data/log.txt");
define("ADRESAR", ABSOLUTE_PATH . "/data/adresar.txt");
define("SEPARTOR", "&");

// Podesavanja za bazu
define("SERVER", env("SERVER"));
define("DATABASE", env("DBNAME"));
define("USERNAME", env("USERNAME"));
define("PASSWORD", env("PASSWORD"));

function env($naziv)
{
    // $podaci = explode("\n", file_get_contents(BASE_URL . "/config/env"));
    $open = fopen(ENV_FAJL, "r");
    $podaci = file(ENV_FAJL);
    $vrednost = "";

    foreach ($podaci as $key => $value) {
        $konfig = explode("=", $value);
        if ($konfig[0] == $naziv) {
            $vrednost = trim($konfig[1]); // trim() zbog \n
        }
    }

    return $vrednost;
}

```

## 5.9. connection.php

```

<?php
ob_start();
session_start();
require_once "config.php";

zabeleziPristupStranici($_GET['page'] ?? "index");

try {
    $conn = new PDO("mysql:host=" . SERVER . ";dbname=" . DATABASE . ";charset=utf8", USERNAME,
    PASSWORD);
    $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}

```

```

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

```

```

function executeQuery($query)
{
    global $conn;
    return $conn->query($query)->fetchAll();
}

```

```

function zabeleziPristupStranici($page = "index")
{
    $open = fopen(LOG_FAJL, "a");
    $date = date('Y-m-d H:i:s');
    if ($open) {
        fwrite($open,
            "page=$page\ttime=$date\t{$_SERVER['PHP_SELF']}\t{$_SERVER['REMOTE_ADDR']}\n");
        fclose($open);
    }
}

```

## 5.10. models/comments/delete.php

```
<?php
```

```

header('Content-Type: application/json');
require_once '../config/connection.php';

```

```

if (isset($_POST['submit'])) {
    try {
        $commentID = $_POST['commentID'];
    }
}

```

```

$data = array(
    "commentID" => $commentID,
);

```

```

if (deleteComment($data, $conn)) {
    http_response_code(200);
    echo json_encode([
        "message" => "Comment deleted.",
    ]);
}

```

```
exit;
```

```
}
```

```
http_response_code(400);  
echo json_encode(["errors" => $errors]);  
} catch (PDOException $ex) {  
echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);  
http_response_code(500);  
}  
} else {  
http_response_code(400); // 400 - Bad request  
}  
}
```

```
function deleteComment($data, $conn)  
{  
try {  
$query = $conn->prepare("DELETE FROM comments WHERE id = :id LIMIT 1");  
$result = $query->execute(array(':id' => $data['commentID']));  
  
return $result;  
} catch (PDOException $e) {  
throw $e;  
}  
}
```

## 5.11. models/comments/insert.php

```
<?php
```

```
header('Content-Type: application/json');  
require_once '../config/connection.php';
```

```
if (isset($_POST['submit'])) {  
try {  
$comment = $_POST['comment'];  
$dog_id = $_POST['dog_id'];
```

```
$errors = array();
```

```
if (empty($comment)) {  
$errors['comment'] = '<div class="mb-0 error-msg">Please enter comment.</div>';  
}
```

```
if (count($errors) == 0) {
```



```

$data = array(
    "comment" => $comment,
    'user_id' => $_SESSION['user']['id'],
    'dog_id' => $dog_id
);

```

```

if (addComment($data, $conn)) {
    http_response_code(201);
    echo json_encode([
        "message" => "Comment added.",
        'comment' => selectLastComment($_SESSION['user']['id'], $conn)
    ]);
}

```

```

exit;
}
}

```

```

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {
    http_response_code(400); // 400 - Bad request
}

```

```

function addComment($data, $conn)
{
    try {
        $query = $conn->prepare("INSERT INTO comments (user_id, dog_id, body) VALUES (?, ?, ?)");
    }
}

```

```

return $query->execute([
    $data['user_id'],
    $data['dog_id'],
    $data['comment'],
]);
} catch (PDOException $e) {
    throw $e;
}
}

```

```

function selectLastComment($userID, $conn)
{
    try {

```

```

$query = $conn->prepare("SELECT c.*, u.username, u.id AS UID
FROM comments c
JOIN users u
ON u.id = c.user_id
WHERE u.id = :id
ORDER BY c.id DESC
LIMIT 1");
$query->execute(array(
'id' => $userID,
));

return $query->fetch(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
throw $e;
}
}

```

## 5.12. models/comments/update.php

```
<?php
```

```

header('Content-Type: application/json');
require_once '../config/connection.php';

if (isset($_POST['submit'])) {
try {
    $comment = $_POST['comment'];
    $commentID = $_POST['commentID'];

    $errors = array();

    if (empty($comment)) {
        $errors['comment'] = '<div class="mb-0 error-msg">Please enter comment.</div>';
    }

    if (count($errors) == 0) {
        $data = array(
            "comment" => $comment,
            "commentID" => $commentID,
        );

        if (updateComment($data, $conn)) {
            http_response_code(201);

```

```

echo json_encode([
    "message" => "Comment updated.",
]);

exit;
}
}

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {
    http_response_code(400); // 400 - Bad request
}

function updateComment($data, $conn)
{
    try {
        $query = $conn->prepare("UPDATE comments SET body = ? WHERE id = ?");
        $result = $query->execute([
            $data['comment'],
            $data['commentID'],
        ]);
    } catch (PDOException $e) {
        throw $e;
    }
}

return $result;
} catch (PDOException $e) {
    throw $e;
}
}

```

### 5.13. models/dogs/delete.php

```

<?php

header('Content-Type: application/json');
require_once '../config/connection.php';

if (isset($_POST['submit'])) {
    try {
        $dogID = $_POST['dogID'];
    }
}

```

```

$data = array(
    "dogID" => $dogID,
);

if (deleteDog($data, $conn)) {
    http_response_code(200);
    echo json_encode([
        "message" => "Dog deleted.",
    ]);
}

exit;
}

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {
    http_response_code(400); // 400 - Bad request
}

function deleteDog($data, $conn)
{
    try {
        $query = $conn->prepare("DELETE FROM dogs WHERE id = :id LIMIT 1");
        $result = $query->execute(array(':id' => $data['dogID']));

        return $result;
    } catch (PDOException $e) {
        throw $e;
    }
}

```

## 5.14. models/dogs/filter.php

```

<?php

header('Content-Type: application/json');
require_once '../config/connection.php';

```

```

if (isset($_POST['submit'])) {
    try {
        $sortValue = $_POST['sortValue'];
        $searchValue = $_POST['searchValue'];

        $perPage = 6;
        $page = $_POST['pagination'];
        $offset = $page * $perPage - $perPage;

        switch ($sortValue) {
            case 1:
                $query = $conn->prepare("SELECT * FROM dogs WHERE name LIKE ? OR breed LIKE ? LIMIT $perPage
                OFFSET $offset");
                $query->execute(array(
                    "%$searchValue%",
                    "%$searchValue%"
                ));

                $dogs = $query->fetchAll(PDO::FETCH_ASSOC);
                break;
            case 2:
                $query = $conn->prepare("SELECT d.*, COUNT(c.dog_id) AS BROJ_KOM FROM dogs d LEFT JOIN
                comments c ON c.dog_id = d.id WHERE name LIKE ? OR breed LIKE ? GROUP BY d.id, c.dog_id ORDER
                BY BROJ_KOM DESC LIMIT $perPage OFFSET $offset");
                $query->execute(array(
                    "%$searchValue%",
                    "%$searchValue%"
                ));

                $dogs = $query->fetchAll(PDO::FETCH_ASSOC);
                break;
            case 3:
                $query = $conn->prepare("SELECT d.*, COUNT(f.dog_id) AS BROJ_FAV FROM dogs d LEFT JOIN
                favorites f ON f.dog_id = d.id WHERE name LIKE ? OR breed LIKE ? GROUP BY d.id, f.dog_id ORDER
                BY BROJ_FAV DESC LIMIT $perPage OFFSET $offset");
                $query->execute(array(
                    "%$searchValue%",
                    "%$searchValue%"
                ));

                $dogs = $query->fetchAll(PDO::FETCH_ASSOC);
                break;
        }

        if ($searchValue === "") {

```

```

$query = $conn->prepare("SELECT * FROM `dogs`");
$query->execute();
$totalPages = ceil(sizeof($query->fetchAll(PDO::FETCH_ASSOC)) / $perPage);
} else {
$totalPages = ceil(sizeof($dogs) / $perPage);
}

http_response_code(201);
echo json_encode(['dogs' => $dogs, 'totalPages' => $totalPages]);

exit;
} catch (PDOException $ex) {
echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
http_response_code(500);
}
} else {
http_response_code(400); // 400 - Bad request
}
}

```

## 5.15. models/dogs/get.php

```
<?php
```

```

header('Content-Type: application/json');
require_once '../config/connection.php';

```

```

try {
$id = $_GET['id'];

```

```

$query = $conn->prepare("SELECT * FROM dogs WHERE id = ?");
$query->execute(array(
$id
));

```

```
$dog = $query->fetch(PDO::FETCH_ASSOC);
```

```

http_response_code(201);
echo json_encode(["dog" => $dog]);

```

```

exit;
} catch (PDOException $ex) {
echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
http_response_code(500);

```

```
}
```

## 5.16. models/dogs/getAll.php

```
<?php
```

```
header('Content-Type: application/json');
require_once '../config/connection.php';

try {
    $perPage = 6;
    $page = $_GET['pagination'];
    $offset = $page * $perPage - $perPage;

    $query = $conn->prepare("SELECT * FROM `dogs`");
    $query->execute();
    $totalPages = ceil(sizeof($query->fetchAll(PDO::FETCH_ASSOC)) / $perPage);

    $query = $conn->prepare("SELECT * FROM `dogs` LIMIT $perPage OFFSET $offset");
    $query->execute();
    $dogs = $query->fetchAll(PDO::FETCH_ASSOC);

    http_response_code(201);
    echo json_encode(["dogs" => $dogs, 'totalPages' => $totalPages]);

    exit;
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
```

## 5.17. models/dogs/insert.php

```
<?php
```

```
header('Content-Type: application/json');
require_once '../config/connection.php';

if (isset($_POST['submit'])) {
    try {
        $breed = $_POST['breed'];
        $name = $_POST['name'];
```

```

$breedDescription = $_POST['breedDescription'];

$errors = array();

if (empty($breed)) {
    $errors['breed'] = '<div class="mb-0 error-msg">Please enter breed.</div>';
}

if (empty($name)) {
    $errors['name'] = '<div class="mb-0 error-msg">Please enter name.</div>';
}

if (empty($breedDescription)) {
    $errors['breedDescription'] = '<div class="mb-0 error-msg">Please enter breed name.</div>';
}

if (empty($_FILES['breedImage'])) {
    $errors['breedImage'] = '<div class="mb-0 error-msg">Please upload image.</div>';
}

if (!empty($_FILES['breedImage']['name'])) { // Da li je poslao neke fajlove
    $file = $_FILES['breedImage'];

    $file_name = $file['name'];
    $file_tmp = $file['tmp_name'];
    $file_size = $file['size'];
    $file_error = $file['error'];

    $file_ext = explode('.', $file_name);
    $file_ext = strtolower(end($file_ext));

    $file_name_new = $file_name;

    if (!file_exists($_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs')) {
        mkdir($_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs');
    }

    $file_destination = $_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs/' . $file_name_new;
    move_uploaded_file($file_tmp, $file_destination);
    $file_destination = 'assets/images/dogs/' . $file_name_new;
}

if (count($errors) == 0) {
    $data = array(
        "breed" => $breed,

```



```

"name" => $name,
"breedDescription" => $breedDescription,
"breedImage" => $file_destination,
);

if (addDog($data, $conn)) {
    http_response_code(201);
    echo json_encode(["message" => "Dog added."]);

    exit;
}
}

```

```

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {
    http_response_code(400); // 400 - Bad request
}

```

```

function addDog($data, $conn)
{
    try {
        $query = $conn->prepare("INSERT INTO dogs (breed, name, description, img) VALUES (?, ?, ?, ?)");

        return $query->execute([
            $data['breed'],
            $data['name'],
            $data['breedDescription'],
            $data['breedImage'],
        ]);
    } catch (PDOException $e) {
        throw $e;
    }
}

```

## 5.18. models/dogs/show.php

```
<?php
```

```
header('Content-Type: application/json');
require_once '../config/connection.php';
```

```
try {
    $id = $_GET['id'];
```

```
    $query = $conn->prepare("SELECT * FROM dogs WHERE id = :id");
    $query->execute(array(
        ':id' => $id,
    ));
```

```
    $dog = $query->fetch(PDO::FETCH_ASSOC);
```

```
    $query = $conn->prepare("SELECT c.*, u.username, u.id AS UID FROM comments c JOIN users u ON u.id
    = c.user_id WHERE dog_id = :id ORDER BY c.id DESC");
    $query->execute(array(
        ':id' => $id,
    ));
```

```
    $comments = $query->fetchAll(PDO::FETCH_ASSOC);
```

```
    $query = $conn->prepare("SELECT * FROM favorites WHERE dog_id = :dog_id");
    $query->execute(array(
        ':dog_id' => $id,
    ));
```

```
    $favorites = $query->fetchAll(PDO::FETCH_ASSOC);
```

```
    $query = $conn->prepare("SELECT * FROM favorites WHERE user_id = :user_id AND dog_id = :dog_id");
    $query->execute(array(
        ':user_id' => $_SESSION['user']['id'] ?? -1,
        ':dog_id' => $id,
    ));
```

```
    $favorite = $query->fetch(PDO::FETCH_ASSOC);
    $favorite = $favorite ? true : false;
```

```
    http_response_code(200);
    echo json_encode([
        'dog' => $dog,
        'comments' => $comments,
        'favorites' => $favorites,
        'favorite' => $favorite,
        'admin' => isset($_SESSION['user']) && $_SESSION['user']['is_admin'],
    ]);
```

```

'user' => isset($_SESSION['user'])
]);

exit;
} catch (PDOException $ex) {
echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
http_response_code(500);
}

```

## 5.19. models/dogs/update.php

<?php

```

header('Content-Type: application/json');
require_once '../config/connection.php';

```

```

if (isset($_POST['submit'])) {
try {
$breed = $_POST['breed'];
$dogID = $_POST['dogID'];
$name = $_POST['name'];
$breedDescription = $_POST['breedDescription'];
$noIMG = false;

```

```

$errors = array();

```

```

if (empty($breed)) {
$errors['breed'] = '<div class="mb-0 error-msg">Please enter breed.</div>';
}

```

```

if (empty($name)) {
$errors['name'] = '<div class="mb-0 error-msg">Please enter name.</div>';
}

```

```

if (empty($breedDescription)) {
$errors['breedDescription'] = '<div class="mb-0 error-msg">Please enter breed name.</div>';
}

```

```

if (empty($_FILES['breedImage'])) {
$noIMG = true;
}

```

```

if (!empty($_FILES['breedImage']['name'])) { // Da li je poslao neke fajlove

```

```

$file = $_FILES['breedImage'];

$file_name = $file['name'];
$file_tmp = $file['tmp_name'];
$file_size = $file['size'];
$file_error = $file['error'];

$file_ext = explode('.', $file_name);
$file_ext = strtolower(end($file_ext));

$file_name_new = $file_name;

if (!file_exists($_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs')) {
    mkdir($_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs');
}

$file_destination = $_SERVER['DOCUMENT_ROOT'] . '/assets/images/dogs/' . $file_name_new;
move_uploaded_file($file_tmp, $file_destination);
$file_destination = 'assets/images/dogs/' . $file_name_new;
}

if (count($errors) == 0) {
    $data = array(
        "breed" => $breed,
        "name" => $name,
        "breedDescription" => $breedDescription,
        "breedImage" => $file_destination ?? false,
        "dogID" => $dogID,
    );

    if (updateDog($data, $conn)) {
        http_response_code(200);
        echo json_encode(["message" => "Dog updated."]);

        exit;
    }

    http_response_code(400);
    echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {

```

```
http_response_code(400); // 400 - Bad request
}
```

```
function updateDog($data, $conn)
```

```
{
    try {
        if ($data['breedImage'] !== false) {
            $query = $conn->prepare("UPDATE dogs SET name = ?, breed = ?, description = ?, img = ? WHERE id = ?");
            $result = $query->execute([
                $data['name'],
                $data['breed'],
                $data['breedDescription'],
                $data['breedImage'],
                $data['dogID'],
            ]);
        }
    }
}
```

```
return $result;
```

```
}
```

```
$query = $conn->prepare("UPDATE dogs SET name = ?, breed = ?, description = ? WHERE id = ?");
$result = $query->execute([
    $data['name'],
    $data['breed'],
    $data['breedDescription'],
    $data['dogID'],
]);
```

```
return $result;
```

```
} catch (PDOException $e) {
```

```
    throw $e;
```

```
}
```

```
}
```

## 5.20. models/favorites/delete.php

```
<?php
```

```
header('Content-Type: application/json');
require_once '../config/connection.php';
```

```
if (isset($_POST['submit'])) {
```

```
    try {
```

```
        $dogID = $_POST['dogID'];
```

```

$data = array(
    "dog_id" => $dogID,
    "user_id" => $_SESSION['user']['id'],
);

if (deleteFromFavorites($data, $conn)) {
    http_response_code(200);
    echo json_encode([
        "message" => "Favorite deleted.",
    ]);
}

exit;
}

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
    echo json_encode(["errors" => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
    http_response_code(500);
}
} else {
    http_response_code(400); // 400 - Bad request
}

function deleteFromFavorites($data, $conn)
{
    try {
        $query = $conn->prepare("DELETE FROM favorites WHERE user_id = :user_id AND dog_id = :dog_id");
        $result = $query->execute(array(
            ':user_id' => $data['user_id'],
            ':dog_id' => $data['dog_id']
        ));
    } catch (PDOException $e) {
        throw $e;
    }
}

```

## 5.21. models/favorites/insert.php

```
<?php
```

```

header('Content-Type: application/json');
require_once '../config/connection.php';

if (isset($_POST['submit'])) {
    try {
        $dogID = $_POST['dogID'];

        $data = array(
            "dog_id" => $dogID,
            "user_id" => $_SESSION['user']['id'],
        );

        if (addDogToFavorites($data, $conn)) {
            http_response_code(201);
            echo json_encode(["message" => "Dog added to favorites."]);

            exit;
        }

        http_response_code(400);
        echo json_encode(["errors" => $errors]);
    } catch (PDOException $ex) {
        echo json_encode(["errors" => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
        http_response_code(500);
    }
} else {
    http_response_code(400); // 400 - Bad request
}

function addDogToFavorites($data, $conn)
{
    try {
        $query = $conn->prepare("INSERT INTO favorites (user_id, dog_id) VALUES (?, ?)");

        return $query->execute([
            $data['user_id'],
            $data['dog_id'],
        ]);
    } catch (PDOException $e) {
        throw $e;
    }
}

```

## 5.22. models/users/admin.php

**<?php**

```
header('Content-Type: application/json');  
require_once '../config/connection.php';
```

```
  
$pages = [  
    'index' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'admin' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'about' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'dogs-edit' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'dogs' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'add-new-dog' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'login' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'register' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
    'profile' => [  
        'total' => 0,  
        'last24' => 0  
    ],  
]
```



```

'admin' => [
'total' => 0,
'last24' => 0
],
'logout' => [
'total' => 0,
'last24' => 0
],
'home' => [
'total' => 0,
'last24' => 0
],
];

```

```

$file = fopen(BASE_URL . "data/log.txt", "r");

```

```

if ($file) {
while (($line = fgets($file)) !== false) {
$lineExploded = explode("\t", $line);
$page = explode('=', $lineExploded[0])[1];
$pages[$page]['total']++;
$date = explode('=', $lineExploded[1])[1];
$secondsDiff = strtotime(date("Y-m-d H:i:s")) - strtotime($date);
if ($secondsDiff < 86400) {
$pages[$page]['last24']++;
}
}
}

```

```

fclose($file);
}

```

```

$total = 0;
$total24 = 0;

```

```

foreach ($pages as $page) {
$total += $page['total'];
$total24 += $page['last24'];
}

```

```

$query = $conn->prepare("SELECT * FROM users WHERE last_login >= now() - INTERVAL 1 DAY");
$query->execute();
$users = $query->fetchAll(PDO::FETCH_ASSOC);

```

```

echo json_encode([

```

```
'pages' => $pages,
'total' => $total,
'total24' => $total24,
'users' => $users
]);
```

## 5.23. models/users/insert.php

```
<?php
```

```
header('Content-Type: application/json');
require_once '../config/connection.php';
```

```
if (isset($_POST['register'])) {
    try {
        $email = $_POST['email'];
        $password = $_POST['password'];
        $confirmPassword = $_POST['confirmPassword'];
        $username = $_POST['username'];
        $birthday = $_POST['birthday'];
```

```
$errors = array();
```

```
if (empty($email)) {
    $errors['email'] = '<div class="mb-0 error-msg">Please enter email.</div>';
} else {
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors['email'] = '<div class="mb-0 error-msg">Wrong email format.</div>';
    }
}
```

```
if (empty($birthday)) {
    $birthday = null;
}
```

```
if (empty($password)) {
    $errors['password'] = '<div class="mb-0 error-msg">Please enter password šifru.</div>';
} else {
    if ($password != $confirmPassword) {
        $errors['password_confirm'] = '<div class="mb-0 error-msg">Passwords do not match.</div>';
    } else {
        if (empty($confirmPassword)) {
            $errors['confirmPassword'] = '<div class="mb-0 error-msg">Please confirm password.</div>';
        }
    }
}
```

```
}  
}  
}
```

```
if (empty($username)) {  
$errors['username'] = '<div class="mb-0 error-msg">Please enter username.</div>';  
} else {  
if (!ctype_alnum($username)) {  
$errors['username'] = '<div class="mb-0 error-msg">Only letters and numbers allowed.</div>';  
}  
}
```

```
if (count($errors) == 0) {  
$data = array(  
"email" => $email,  
"password" => $password,  
"username" => $username,  
"birthday" => $birthday,  
);
```

```
if (takenEmail($email, $conn)) {  
$errors['taken_email'] = '<div class="alert alert-danger alert-dismissible show" role="alert">  
<strong>Error!</strong> Email is taken.  
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>  
</div>';  
} elseif (takenUsername($username, $conn)) {  
$errors['taken_username'] = '<div class="alert alert-danger alert-dismissible show" role="alert">  
<strong>Error!</strong> Username is taken.  
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>  
</div>';  
} elseif (registerUser($data, $conn)) {  
http_response_code(201);  
echo json_encode(["message" => "User registered."]);
```

```
exit;  
}  
}
```

```
http_response_code(400);  
echo json_encode(["errors" => $errors]);  
} catch (PDOException $ex) {  
echo json_encode(["errors" => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);  
http_response_code(500);  
}  
} else {
```

```
http_response_code(400); // 400 - Bad request  
}
```

```
function takenEmail($email, $conn)  
{  
    try {  
        $query = $conn->prepare("SELECT * FROM users WHERE email = :email");  
        $query->execute(array(  
            ':email' => $email,  
        ));  
    }
```

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

```
    if ($user) {  
        return true;  
    }
```

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

```
function takenUsername($username, $conn)  
{  
    try {  
        $query = $conn->prepare("SELECT * FROM users WHERE username = :username");  
        $query->execute(array(  
            ':username' => $username,  
        ));  
    }
```

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

```
    if ($user) {  
        return true;  
    }
```

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

```
function registerUser($data, $conn)  
{
```

```

try {
    $data['password'] = password_hash($data['password'], PASSWORD_BCRYPT); // Hashovanje sifre sa md5
    $query = $conn->prepare("INSERT INTO users (email, password, username, birthday) VALUES (?, ?, ?, ?)");
    $result = $query->execute([
        $data['email'],
        $data['password'],
        $data['username'],
        $data['birthday'] ?? null,
    ]);

    $data['id'] = $conn->lastInsertId();
    $data['is_admin'] = false;

    // $data = $this->fixData($data);

    if ($result) { // Ukoliko je query uspesan pravimo sesiju i vracamo true
        $_SESSION['user'] = $data;

        return true;
    }

    return false;
} catch (PDOException $e) {
    throw $e;
}
}

```

## 5.24. models/users/login.php

```

<?php

header('Content-Type: application/json');
require_once '../config/connection.php';

if (isset($_POST['login'])) {
    try {
        $usernameOrEmail = $_POST['usernameOrEmail'];
        $password = $_POST['password'];

        $errors = array();

        if (empty($usernameOrEmail)) {
            $errors['usernameOrEmail'] = '<div class="mb-0 error-msg">Please enter username/email.</div>';

```

```

}

if (empty($password)) {
$errors['password'] = '<div class="mb-0 error-msg">Please enter password šifru.</div>';
}

if (count($errors) == 0) {
$data = array(
'usernameOrEmail' => $usernameOrEmail,
'password' => $password,
);

if (loginUser($data, $conn)) {
http_response_code(200);
echo json_encode(["message" => "User logged in."]);
}

exit;
} else {
$errors['wrong_combination'] = '<div class="mb-0 error-msg">Wrong combination.</div>';
}
}

http_response_code(400);
echo json_encode(["errors" => $errors]);
} catch (PDOException $ex) {
echo json_encode(['errors' => ['db_error' => 'DB Error: ' . $ex->getMessage()]]);
http_response_code(500);
}
} else {
http_response_code(400); // 400 - Bad request
}

function loginUser($data, $conn)
{
try {
$query = $conn->prepare("SELECT * FROM users WHERE email = :email OR username = :username");
$query->execute(array(
'email' => $data['usernameOrEmail'],
'username' => $data['usernameOrEmail'],
));

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

if ($user && password_verify($data['password'], $user['password'])) {
$_SESSION['user'] = $user;

```

```

$query = $conn->prepare("UPDATE users SET last_login = ? WHERE id = ?");
$query->execute([
    date("Y-m-d H:i:s"),
    $user['id']
]);

return true;
}

return false;
} catch (PDOException $e) {
    throw $e;
}
}

```

## 5.25. models/users/profile.php

```
<?php
```

```

header('Content-Type: application/json');
require_once '../config/connection.php';

```

```
$user = $_SESSION['user'];
```

```

$query = $conn->prepare("SELECT * FROM comments WHERE user_id = :id ORDER BY id DESC");
$query->execute(array(
    ':id' => $user['id'],
));

```

```
$comments = $query->fetchAll(PDO::FETCH_ASSOC);
```

```

$query = $conn->prepare("SELECT d.name as name, d.breed as breed, d.id as dog_id FROM favorites f JOIN
dogs d ON d.id = f.dog_id WHERE user_id = :user_id");
$query->execute(array(
    ':user_id' => $user['id'],
));

```

```
$favorites = $query->fetchAll(PDO::FETCH_ASSOC);
```

```

echo json_encode([
    'user' => $user,
    'comments' => $comments,

```

```
'favorites' => $favorites,  
]);
```

## 5.26. views/fixed/footer.php

```
</div>  
<footer class="footer bg-dark color-primary">  
<p class="text-center">  
Copyright &copy; <script>  
document.write(new Date().getFullYear())  
</script><br>  
Created by Jana Matović&nbsp;  
</p>  
</footer>  
  
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmW15/YESvpZ13" crossorigin="anonymous"></script>  
  
<!-- ZA VEZBU -->  
<script src="assets/js/main.js"></script>  
  
</body>  
  
</html>
```

## 5.27. views/fixed/head.php

```
<?php  
  
$query = $conn->prepare("SELECT * FROM `navigations`");  
$query->execute();  
$navigations = $query->fetchAll(PDO::FETCH_ASSOC);  
  
?>  
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
<meta charset="utf-8" />
```



```

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<title> Dogster </title>
<meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0, shrink-to-fit=no'
name='viewport' />
<link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700|Roboto+Slab:400,700|Material+Icons
" />
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-
awesome.min.css">
<!-- <link href="assets/css/material-dashboard.css?v=2.1.1" rel="stylesheet" /> -->
<!-- <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"> -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
<link rel="stylesheet" href="assets/css/styles.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.js" integrity="sha512-
n/4gHW3atM3QqRcbCn6ewmpxcLAHGaDjpEBu4xZd47N0W2oQ+6q7oc3PXstrJYXcbNU1OHdQ1T7pAP+
gi5Yu8g==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
</head>

```

```

<body class="">

```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container">
<a class="navbar-brand" href="index.php">Dogster </a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<?php foreach ($navigations as $navigation) : ?>
<?php if ($navigation['page_name'] == "Home" || $navigation['page_name'] == "About me") { ?>
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="<?php echo $navigation['page_link']; ?>"><?php echo
$navigation['page_name']; ?></a>
</li>
<?php } ?>
<?php if (isset($_SESSION['user']) && $_SESSION['user']['is_admin'] == 1 && $navigation['page_name'] ==
"Add New Dog") { ?>
<li class="nav-item">

```

```

<a class="nav-link active" aria-current="page" href="<?php echo $navigation['page_link']; ?>"><?php echo
$navigation['page_name']; ?></a>
</li>
<?php } ?>
<?php endforeach; ?>
</ul>
<ul class="navbar-nav ms-auto mb-2 mb-lg-0">
<?php if (!isset($_SESSION['user'])) { ?>
<?php foreach ($navigations as $navigation) : ?>
<?php if ($navigation['page_name'] == "Login" || $navigation['page_name'] == "Register") { ?>
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="<?php echo $navigation['page_link']; ?>"><?php echo
$navigation['page_name']; ?></a>
</li>
<?php } ?>
<?php endforeach; ?>
<?php } ?>
<?php if (isset($_SESSION['user'])) { ?>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
<?php echo $_SESSION['user']['username']; ?>
</a>
<ul class="dropdown-menu" aria-labelledby="navbarDropdown">
<?php foreach ($navigations as $navigation) : ?>
<?php if ($navigation['page_name'] == "Profile" || $navigation['page_name'] == "Logout") { ?>
<li class="nav-item">
<a class="dropdown-item" aria-current="page" href="<?php echo $navigation['page_link']; ?>"><?php echo
$navigation['page_name']; ?></a>
</li>
<?php } ?>
<?php if (isset($_SESSION['user']) && $_SESSION['user']['is_admin'] == 1 && $navigation['page_name'] ==
"Admin") { ?>
<li><a class="dropdown-item" href="<?php echo $navigation['page_link']; ?>">Admin panel</a></li>
<?php } ?>
<?php endforeach; ?>
</ul>
</li>
<?php } ?>
</ul>
</div>
</div>
</nav>

```

```

<div class="container">

```

## 5.28. views/dogs/add.php

```
<?php
```

```
if (!isset($_SESSION['user']) || (isset($_SESSION['user']) && $_SESSION['user']['is_admin'] == 0)) {  
    header("Location: index.php");
```

```
    exit;  
}
```

```
?>
```

```
<section class="mt-4 mb-5">  
    <div class="container">  
        <div class="row justify-content-center">  
            <div class="col-lg-6 col-md-8 col-sm-10 col-10 g-0">  
                <div>  
                    <h1 class="mb-1">Submit new dog &#128021;</h1>  
                    <p class="mb-4">Fields marked with <strong class="text-danger">*</strong> are required.</p>
```

```
<div id="add-new-dog-errors" class="mb-3"></div>
```

```
<form id="submit-new-dog-form">  
    <div class="mb-4">  
        <label class="form-label">Breed <strong class="text-danger">*</strong></label>  
        <input name="breed" type="text" class="form-control" placeholder="Enter breed">  
    </div>
```

```
<div class="mb-4">  
    <label class="form-label">Name <strong class="text-danger">*</strong></label>  
    <input name="name" type="text" class="form-control" placeholder="Enter name">  
</div>
```

```
<div class="mb-4">  
    <label class="form-label">Description <strong class="text-danger">*</strong></label>  
    <textarea class="form-control" name="description" placeholder="Enter description" id="floatingTextarea2"  
    style="height: 100px"></textarea>  
</div>
```

```
<div class="mb-4">  
    <label class="form-label">Upload Image <strong class="text-danger">*</strong></label> <br>  
    <input type="file" name="file">  
</div>
```

```

<button type="submit" class="btn btn-primary w-100">Submit</button>
</form>
</div>
</div>
</div>
</div>
</section>

```

```

<script src="assets/js/api/dogs/dogs.js"></script>

```

## 5.29. views/dogs/edit.php

```

<?php

```

```

if (!isset($_SESSION['user']) || (isset($_SESSION['user']) && $_SESSION['user']['is_admin'] == 0)) {
header("Location: index.php");

```

```

exit;
}

```

```

$id = $_GET['id'];

```

```

$query = $conn->prepare("SELECT * FROM dogs WHERE id = :id");
$query->execute(array(
    ':id' => $id,
));

```

```

$dog = $query->fetch(PDO::FETCH_ASSOC);

```

```

?>

```

```

<section class="mt-4 mb-5">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-6 col-md-8 col-sm-10 col-10 g-0">
<div class="edit-dog">

```

```

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

```

```
<script src="assets/js/api/dogs/dogs.js"></script>
```

### 5.30. views/dogs/show.php

```
<div class="row mt-5">
```

```
<div class="col-md-8 mb-3 offset-md-2 data">
```

```
</div>
```

```
</div>
```

```
<script src="assets/js/api/dogs/dogs.js"></script>
```

```
<script src="assets/js/api/favorites/favorites.js"></script>
```

### 5.31. views/about.php

```
<div id="export">
```

```
<div class="col-md-10 offset-md-1 mt-5 mb-5">
```

```
<h1 class="text-center mb-4">About me :)</h1>
```

```
<div class="card mb-3 shadow">
```

```
<div class="image-container">
```

```

```

```
</div>
```

```
<div class="card-body">
```

```
<p class="card-text mt-0">
```

Zdravo, ja sam Jana Matović. Rođena sam u Prijepolju i imam 22 godine. Završila sam gimnaziju, prirodno-matematički smer, a trenutno sam student 3. godine “Visoke škole za informacione i komunikacione tehnologije”. Smer: Informacione tehnologije Modul: Web programiranje

```
</p>
```

```
<button onclick="ExportToDoc('export');">Export as .doc</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
function ExportToDoc(element, filename = ") {
```

```
var header = "<html xmlns:o='urn:schemas-microsoft-com:office:office' xmlns:w='urn:schemas-microsoft-com:office:word' xmlns='http://www.w3.org/TR/REC-html40'><head><meta charset='utf-8'><title>Export HTML to Word Document with JavaScript</title></head><body>";
```

```
var footer = "</body></html>";
```

```

var html = header + document.getElementById(element).innerHTML + footer;

var blob = new Blob(["\u0000", html], {
  type: 'application/msword'
});

// Specify link url
var url = 'data:application/vnd.ms-word;charset=utf-8,' + encodeURIComponent(html);

// Specify file name
filename = filename ? filename + '.docx' : 'document.docx';

// Create download link element
var downloadLink = document.createElement("a");

document.body.appendChild(downloadLink);

if (navigator.msSaveOrOpenBlob) {
  navigator.msSaveOrOpenBlob(blob, filename);
} else {
  // Create a link to the file
  downloadLink.href = url;

  // Setting the file name
  downloadLink.download = filename;

  //triggering the function
  downloadLink.click();
}

document.body.removeChild(downloadLink);
}
</script>

```

## 5.32. views/admin.php

```

<?php

if (!isset($_SESSION['user']) || (isset($_SESSION['user']) && $_SESSION['user']['is_admin'] == 0)) {
  header("Location: index.php");

  exit;
}

```

```
?>
```

```
<div class="col-md-10 offset-md-1 mt-5">
<div class="card shadow">
<div class="card-body">
<h5 class="card-title">Pages statistics</h5>
<hr>
<p class="card-text">
<table class="table text-center table-hover">
<thead>
<tr>
<th scope="col">NAME</th>
<th scope="col">TOTAL VISITS</th>
<th scope="col">LAST 24h VISITS</th>
</tr>
</thead>
<tbody class="pages-stat">
```

```
</tbody>
</table>
<h5 class="card-title active-users"></h5>
<hr>
<p>
</div>
</div>
</div>
```

```
<script src="assets/js/api/admin/admin.js"></script>
```

### 5.33. views/home.php

```
<div class="mt-5">
<div class="mb-3">
<div class="d-flex w-50">
<div>
<label for="">Sort by</label>
<select class="form-select sort-input" aria-label="Default select example">
<option selected value="1">Date</option>
<option value="2">Most Comments</option>
<option value="3">Most Likes</option>
</select>
</div>
<div class="ms-2">
<label for="">Search</label>
```

```

<input class="form-control me-2 search-input" type="search" placeholder="Search" aria-label="Search">
</div>
</div>
</div>
<div id="dogs" class="row">

</div>

<nav aria-label="Page navigation example">
<ul class="pagination" style="justify-content:center;">

</ul>
</nav>
</div>

<script src="assets/js/api/dogs/dogs.js"></script>

```

## 5.34. views/login.php

```
<?php
```

```

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

```

```

exit;
}

```

```
?>
```

```

<!-- ----- REGISTRACIJA ----- -->
<section class="mt-4 mb-5">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-6 col-md-8 col-sm-10 col-10 g-0">
<div class="login-form-container">
<h1 class="mb-1">Login</h1>
<p class="mb-4">Fields marked with <strong class="text-danger">*</strong> are required.</p>

<div id="login-errors" class="mb-3"></div>

<form id="login-form">
<div class="mb-4">
<label class="form-label">Username/Email <strong class="text-danger">*</strong></label>

```



```
<input name="usernameOrEmail" type="text" class="form-control" placeholder="Enter Username or email"
value="<?php echo $username ?? ""; ?>">
</div>
```

```
<div class="mb-4">
<label class="form-label">Password <strong class="text-danger">*</strong></label>
<input name="password" type="password" class="form-control" placeholder="Enter password">
</div>
```

```
<button type="submit" class="btn btn-primary w-100">Login</button>
```

```
<div class="text-center">
<div class="mt-3 mb-1">OR</div>
</div>
```

```
<div class="text-center mt-0">
You don't have an account, go to <a href="<?php echo BASE_URL ?>?page=register">register</a> page.
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
<!-- ----- REGISTRACIJA ----- -->
```

### 5.35. views/logout.php

```
<?php
```

```
unset($_SESSION['user']);
header("Location: index.php");
```

```
exit;
```

### 5.36. views/profile.php

```
<?php
```

```
if (!isset($_SESSION['user'])) {
header("Location: index.php");
```

```
exit;
```

```
}
```

```
?>
```

```
<div class="col-md-10 offset-md-1 mt-4 mb-5">  
<h1 class="mb-4 text-center">Welcome to your profile page :)</h1>  
<div class="card text-center">  
<div class="card-header profile-header">
```

```
</div>  
<div class="card-body">  
<div class="accordion" id="accordionExample">
```

```
<div class="accordion-item">  
<h2 class="accordion-header">  
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-  
target="#collapseComments" aria-expanded="false" aria-controls="collapseComments">  
<i class="fa fa-comments" aria-hidden="true"></i> &nbsp; <span class="comments-  
number"></span>&nbsp;Comments  
</button>  
</h2>  
<div id="collapseComments" class="accordion-collapse collapse" aria-labelledby="headingTwo" data-bs-  
parent="#accordionExample">  
<div class="accordion-body text-start">  
<ul class="comments-list">
```

```
</ul>  
</div>  
</div>  
</div>
```

```
<div class="accordion-item">  
<h2 class="accordion-header">  
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-  
target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">  
<i class="fa fa-heart" aria-hidden="true"></i> &nbsp; <span class="favorites-  
number"></span>&nbsp;Favorite dogs  
</button>  
</h2>  
<div id="collapseTwo" class="accordion-collapse collapse" aria-labelledby="headingTwo" data-bs-  
parent="#accordionExample">  
<div class="accordion-body text-start">  
<ul class="favorites-list">
```

```
</ul>
```

```

</div>
</div>
</div>
</div>
</div>
<div class="card-footer text-muted user-joined">
</div>
</div>
</div>

```

```

<script src="assets/js/api/users/users.js"></script>

```

### 5.37. views/register.php

```

<?php

```

```

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

```

```

    exit;
}

```

```

?>
<!-- ----- REGISTRACIJA ----- -->
<section class="mt-4 mb-5">
<div class="container">
<div class="row justify-content-center">
<div class="col-lg-6 col-md-8 col-sm-10 col-10 g-0">
<div class="login-form-container">
<h1 class="mb-1">Register</h1>
<p class="mb-4">Fields marked with <strong class="text-danger">*</strong> are required.</p>

```

```

<div id="register-errors" class="mb-3"></div>

```

```

<form id="register-form">
<div class="mb-4">
<label class="form-label">Username <strong class="text-danger">*</strong></label>
<input name="username" type="text" class="form-control" placeholder="Enter Username" value="<?php
echo $username ?? ""; ?>">
</div>

```

```

<div class="mb-4">
<label class="form-label">Email <strong class="text-danger">*</strong></label>

```

```
<input name="email" type="email" class="form-control" placeholder="Enter email" value="<?php echo $email ?? ''; ?>">
</div>
```

```
<div class="mb-4">
<label class="form-label">Password <strong class="text-danger">*</strong></label>
<input name="password" type="password" class="form-control" placeholder="Confirm password">
</div>
```

```
<div class="mb-4">
<label class="form-label">Confirm password <strong class="text-danger">*</strong></label>
<input name="password2" type="password" class="form-control" placeholder="Potvrdite šifru">
</div>
```

```
<div class="mb-4">
<label class="form-label">Birthday</label>
<input name="birthday" type="date" class="form-control" placeholder="Enter birthday" value="<?php echo $birthday ?? ''; ?>">
</div>
```

```
<button name="register" type="submit" class="btn btn-primary w-100">Register</button>
```

```
<div class="text-center">
<div class="mt-3 mb-1">OR</div>
</div>
```

```
<div class="text-center mt-0">
You already have account, go to <a href="<?php echo BASE_URL ?>?page=login">login</a> page.
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
<!-- ----- REGISTRACIJA ----- -->
```

## 5.38. views/index.php

```
<?php
```

```
require_once "config/connection.php";
include "views/fixed/head.php";
```

```
if (!isset($_GET['page'])) {  
    include "views/home.php";  
} else {  
    switch ($_GET['page']) {  
        case 'admin':  
            include "views/admin.php";  
            break;  
        case 'about':  
            include "views/about.php";  
            break;  
        case 'dogs-edit':  
            include "views/dogs/edit.php";  
            break;  
        case 'dogs':  
            include "views/dogs/show.php";  
            break;  
        case 'add-new-dog':  
            include "views/dogs/add.php";  
            break;  
        case 'login':  
            include "views/login.php";  
            break;  
        case 'register':  
            include "views/register.php";  
            break;  
        case 'profile':  
            include "views/profile.php";  
            break;  
        case 'admin':  
            include "views/admin.php";  
            break;  
        case 'logout':  
            include "views/logout.php";  
            break;  
        default:  
            include "views/home.php";  
            break;  
    }  
}  
  
include "views/footer.php";
```

