**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

ЗВІТ

До контрольної роботи

**З дисципліни:** *“* *Програмування в Інтернет”*

**Лектор:**

проф. каф. ПЗ

Мельник Р. А.

**Виконав:**

ст. гр. ПЗ-23 Михалевич П.-І. В.

**Прийняв:**

доц. каф. ПЗ

Грицай О.Д.

« » 2022 р.

∑= .

Львів – 2022

**Тема роботи:** Створення власної системи для пошуку зображень.

**Мета роботи:** Закріпити знання отримані під час вивчення дисципліни “Програмування в Інтернет”.

**ЗАВДАННЯ**

1. Створити сайт для пошуку зображень.

2. Реалізувати авторизацію та валідацію даних.

3. Створити окремі сторінки для адміністратора, звичайного користувача та

анонімного користувача.

4. Реалізувати можливість публікації зображень.

5. Реалізувати пошук зображень із бази даних на стороні користувача.

**Результати виконання**

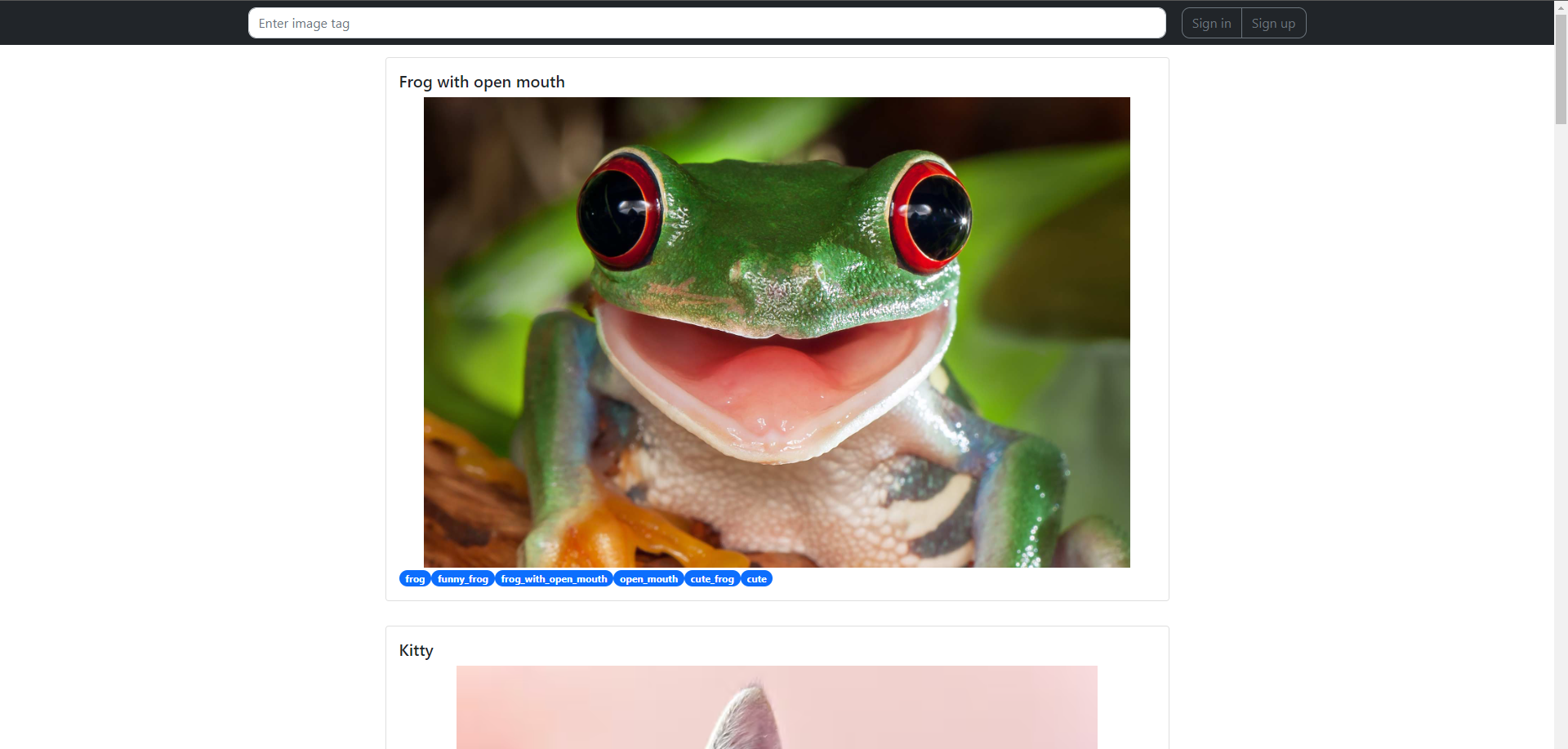
****

Рис.1. Головна сторінка для анонімного юзера

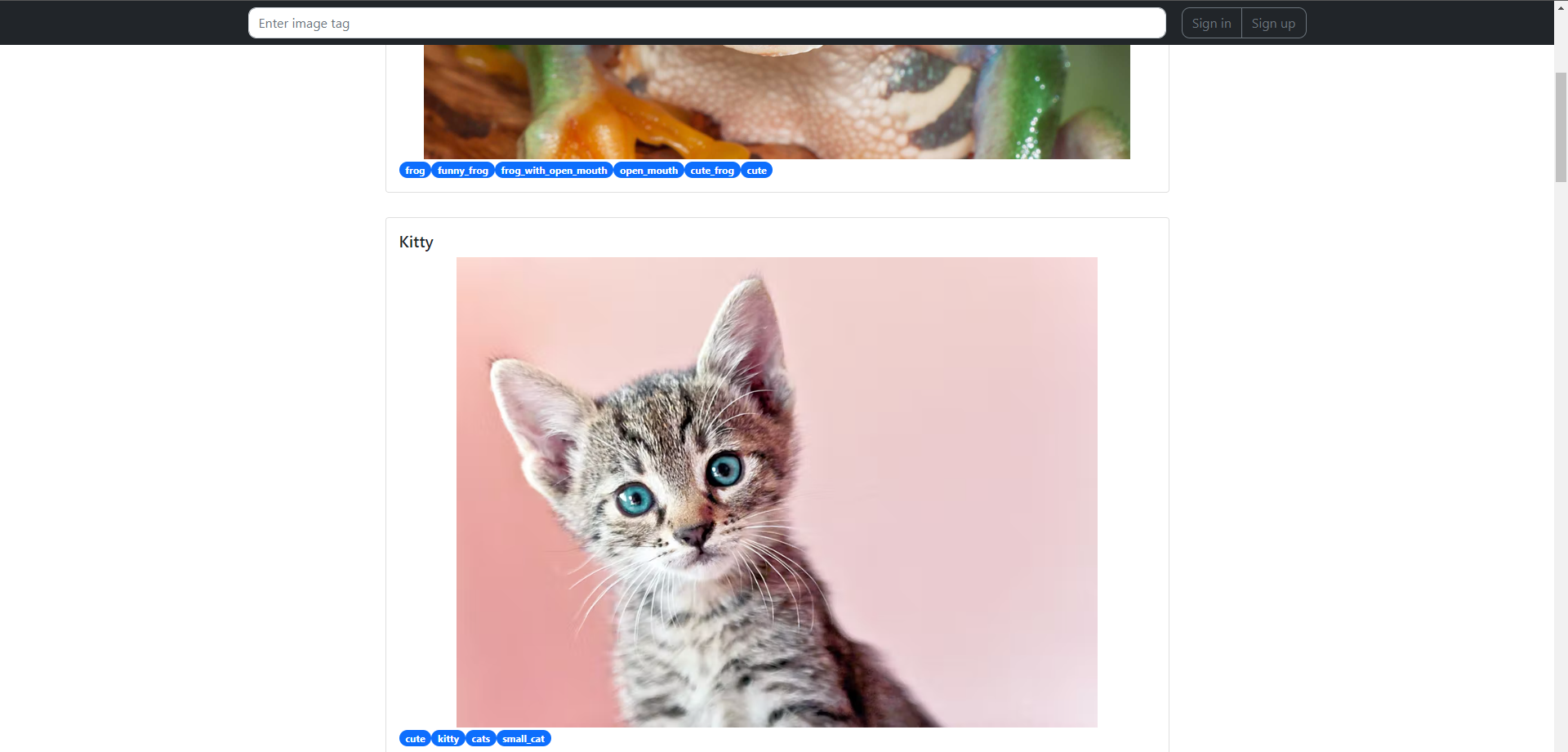


Рис.2. Головна сторінка для анонімного юзера

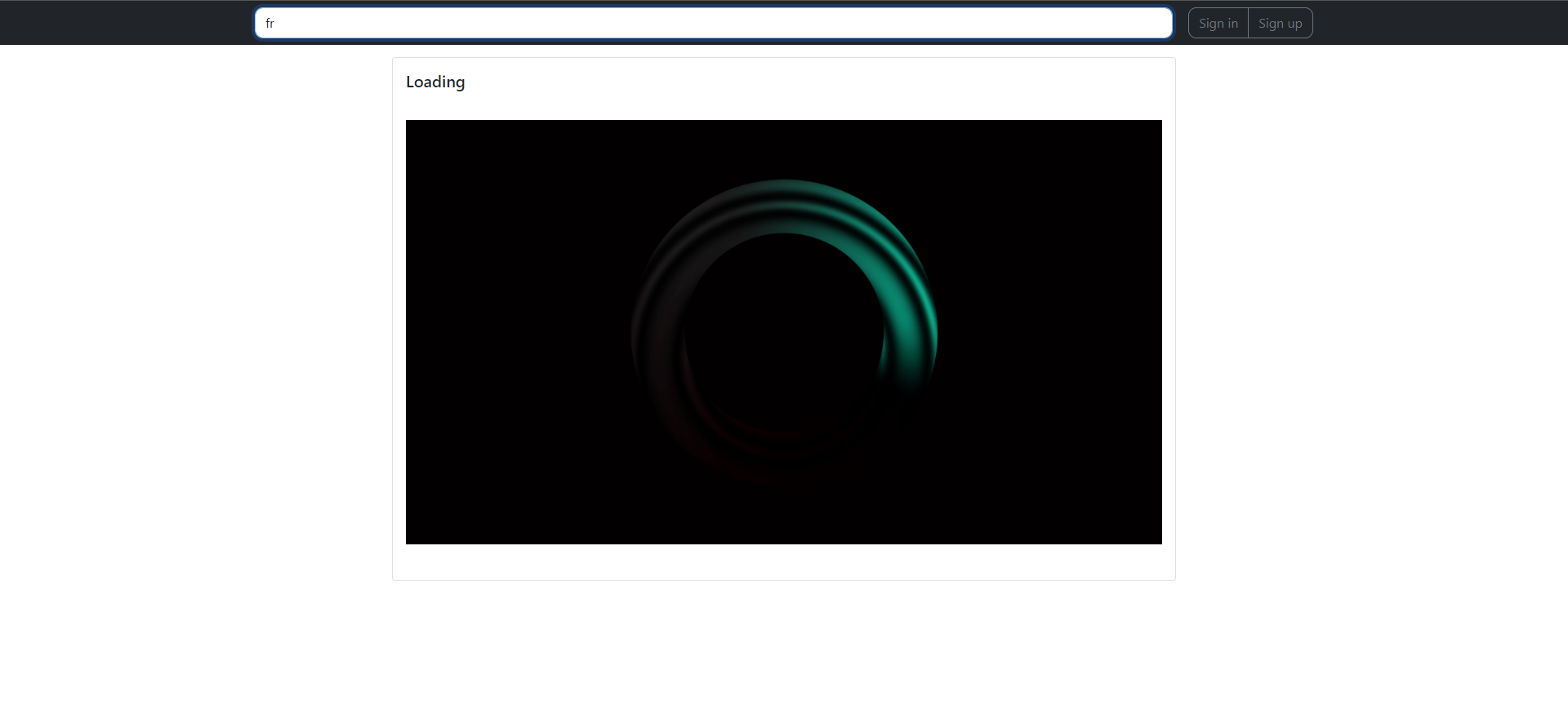


Рис.3. Очікування на відповідь від бази даних

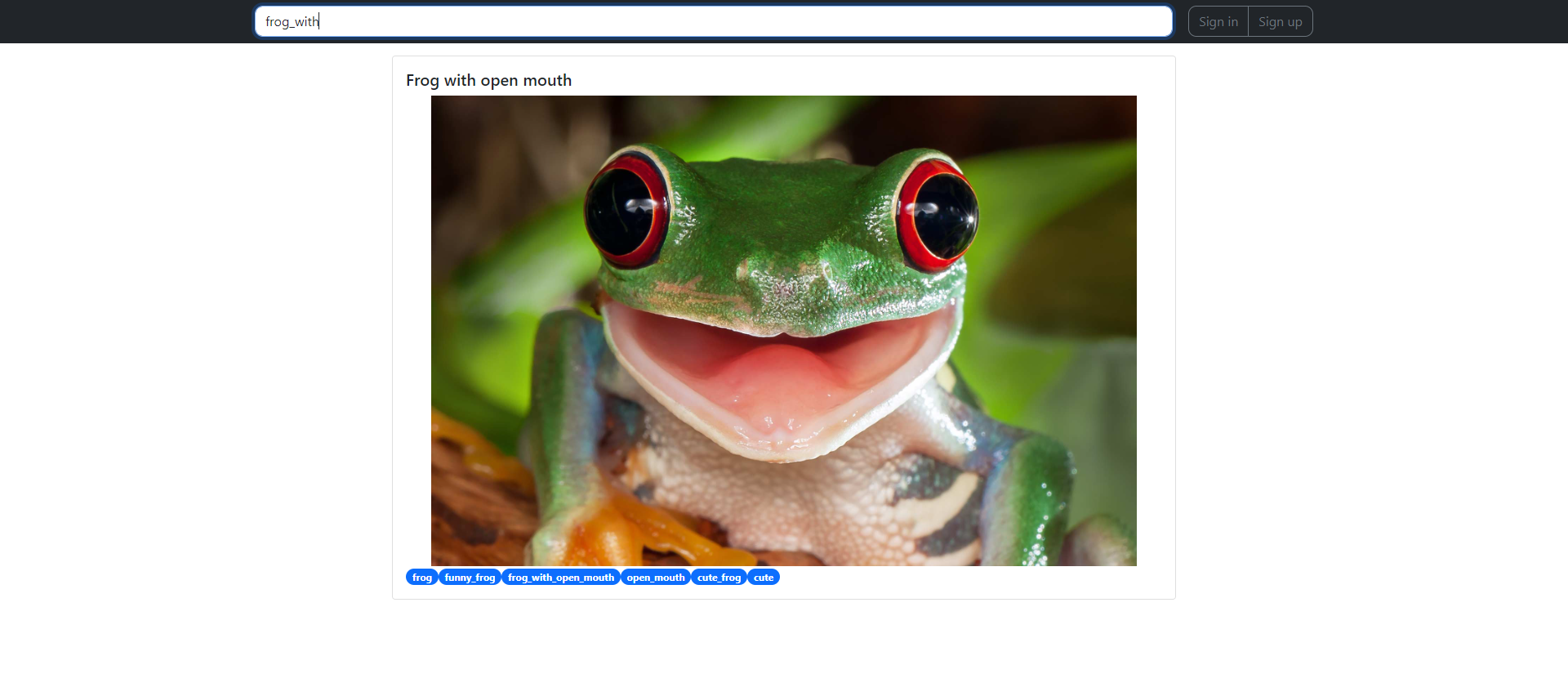


Рис. 4. Пошук по тегу

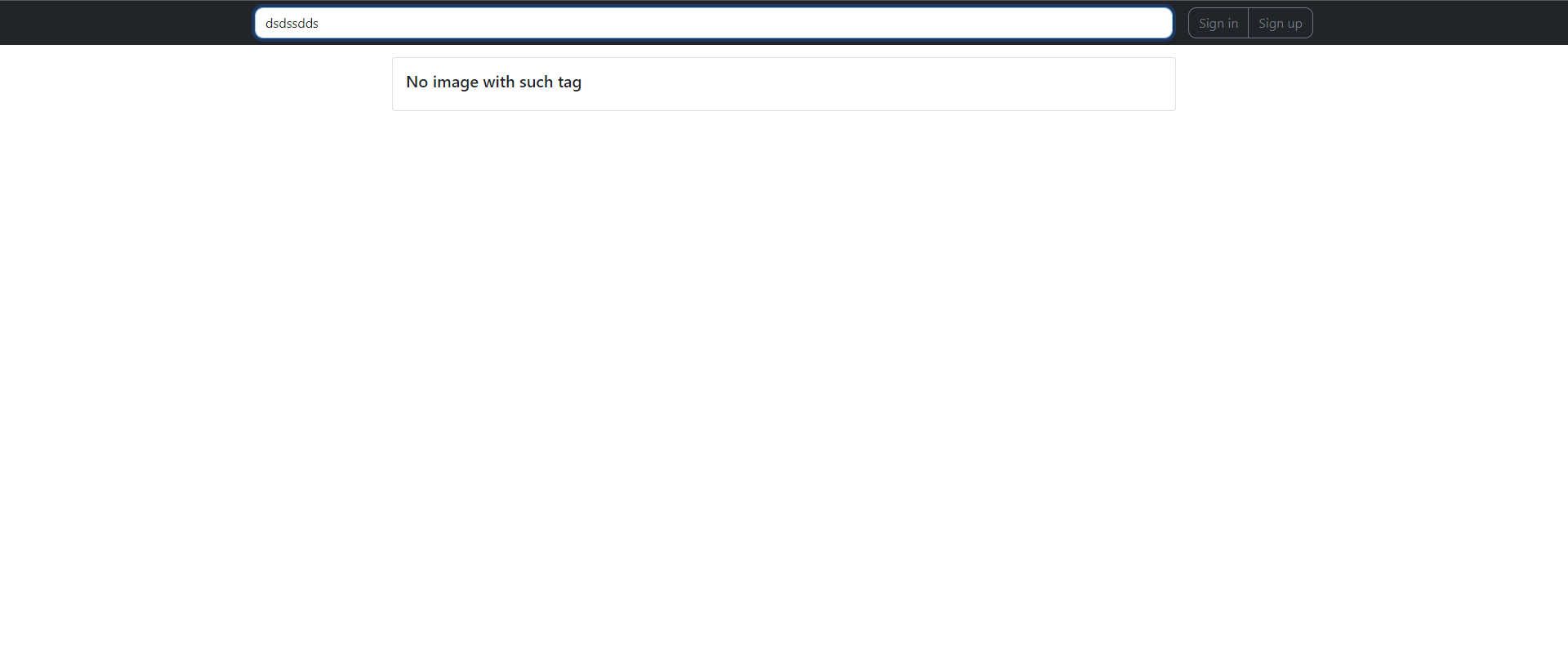


Рис. 5. Відсутність картинок з поданим тегом

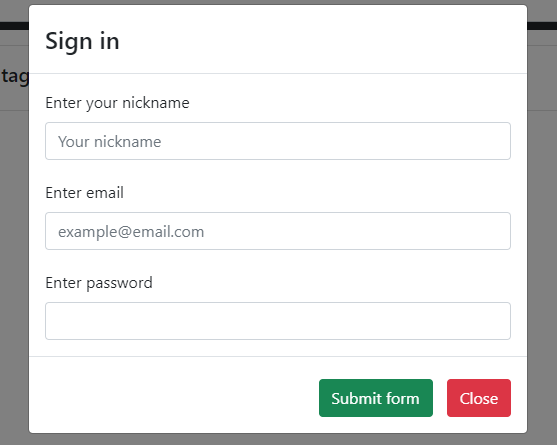


Рис. 6. Форма для входу

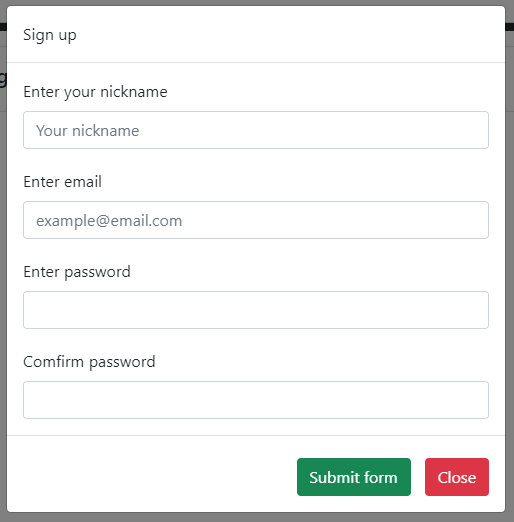


Рис. 7. Форма для реєстрування

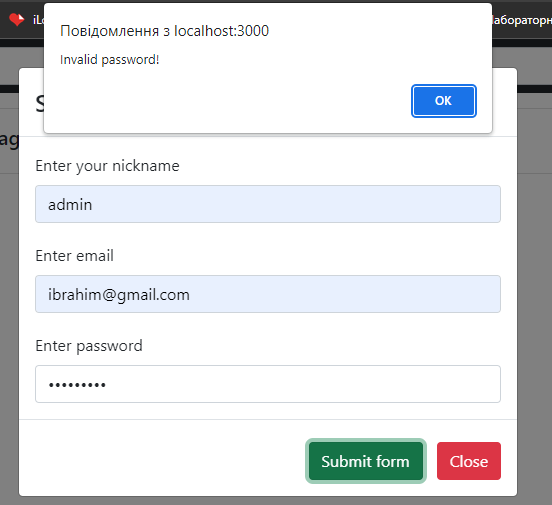


Рис. 8. Валідація форми

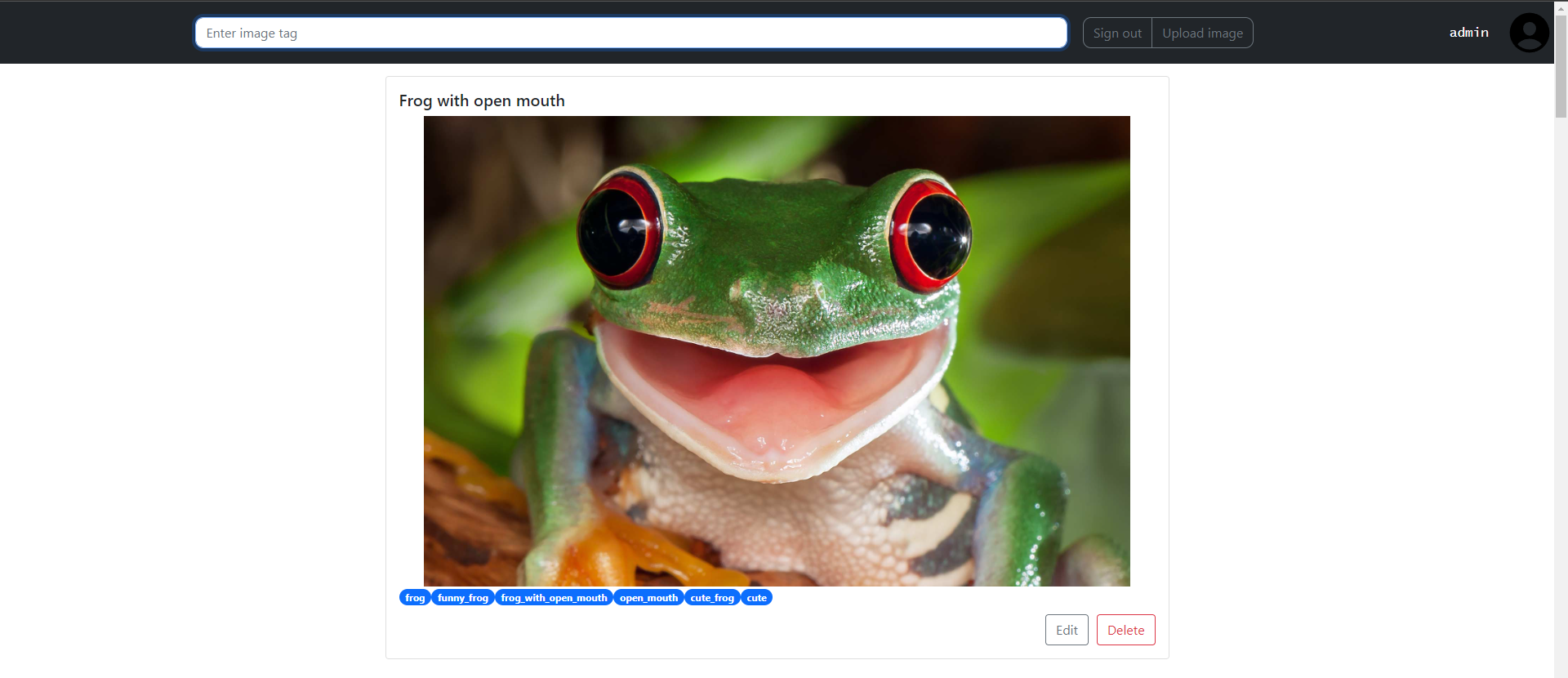


Рис. 9. Головна сторінка для зареєстрованого юзера. Добавляється можливість видалення або редагування картинки (для адміна всіх картинок, для юзера тільки своєї)

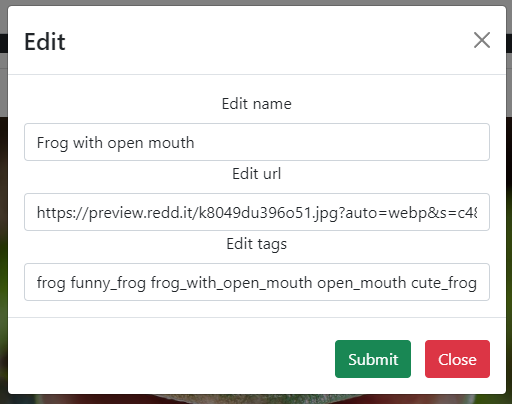


Рис. 10. Форма для редагування картинки

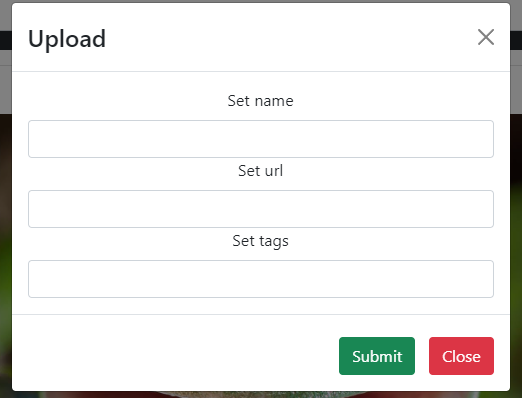


Рис. 11. Форма для завантаження картинки

Код програми:

ImageBox.jsx:

import { Modal, Form, Button, FormGroup, Card, Badge } from "react-bootstrap";

import {React, useState} from 'react'

import LoadingImage from "../pages/Loading";

function ImageBox(props) {

    const [modalEditShow, setModalEditShow] = useState(false);

    var tags = props.tags.split(' ');

    function updateImage(){

        var name\_upd = document.getElementById("name\_upd").value;

        var url\_upd = document.getElementById("url\_upd").value;

        var tags\_upd = document.getElementById("tags\_upd").value;

        var server = 'http://kr/update\_image.php';

        var formdata = new FormData();

        formdata.append('old\_name', props.name);

        formdata.append('name', name\_upd);

        formdata.append('url', url\_upd);

        formdata.append('tags', tags\_upd);

        formdata.append('user\_id', props.user\_id);

        props.setImages(Array(<LoadingImage />));

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/x-www-form-urlencoded'},

            body : formdata

        })

        .then(response => { return response.json(); })

        .then(result\_data => {

            if(result\_data.sever\_error){

                alert(result\_data.sever\_error\_text);

                props.showImages();

                return false;

            }

            setModalEditShow(false);

            props.showImages();

        });

    }

    function delete\_image(){

        var server = 'http://kr/delete\_image.php';

        var formdata = new FormData();

        formdata.append('name', props.name);

        formdata.append('user\_id', props.user\_id);

        props.setImages(Array(<LoadingImage />));

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/x-www-form-urlencoded'},

            body : formdata

        })

        .then(response => { return response.json(); })

        .then(result\_data => {

            if(result\_data.sever\_error){

                alert(result\_data.sever\_error\_text);

                props.showImages();

                return false;

            }

            props.showImages();

        });

    }

    return(

        <div style={{width:'100%', display: 'flex', justifyContent: 'center'}}>

            <Card style={{ width: '60rem', margin: '15px' }}>

                <Card.Body>

                    <Card.Title>{props.name}</Card.Title>

                    {props.src !== '' && <Card.Img style={{width: '100%',  height: '36rem', objectFit: 'contain'}} src={props.src} alt={props.name} />}

                    {tags.map(tag => <Badge pill bg="primary">{tag}</Badge>)}

                    <div style={{display: 'flex', justifyContent: 'right'}}>

                        {props.CanEdit && <Button variant="outline-secondary" onClick={() => setModalEditShow(true)} style={{marginTop: 10}}>

                            Edit

                        </Button>}

                        {props.CanEdit && <Button variant="outline-danger" onClick={() => delete\_image()} style={{marginTop: 10, marginLeft: 10}}>

                            Delete

                        </Button>}

                    </div>

                </Card.Body>

            </Card>

            <Modal

            show = {modalEditShow}

            onHide = {() => setModalEditShow(false)}

            className="text-center"

            >

                <Modal.Header closeButton>

                    <Modal.Title>

                        Edit

                    </Modal.Title>

                </Modal.Header>

                <Modal.Body>

                    <Form>

                        <FormGroup>

                            <Form.Label>Edit name</Form.Label>

                            <Form.Control

                            id = "name\_upd"

                            name = "name\_upd"

                            type = "text"

                            defaultValue = {props.name}

                            />

                        </FormGroup>

                        <FormGroup>

                            <Form.Label>Edit url</Form.Label>

                            <Form.Control

                            id = "url\_upd"

                            name = "url\_upd"

                            type = "text"

                            defaultValue = {props.src}

                            />

                        </FormGroup>

                        <FormGroup>

                            <Form.Label>Edit tags</Form.Label>

                            <Form.Control

                            id = "tags\_upd"

                            name = "tags\_upd"

                            type = "text"

                            defaultValue = {props.tags}

                            />

                        </FormGroup>

                    </Form>

                </Modal.Body>

                <Modal.Footer>

                    <Button variant="success" onClick={() => updateImage()} style={{marginTop: 10}}>

                        Submit

                    </Button>

                    <Button variant="danger" onClick={() => setModalEditShow(false)} style={{marginTop: 10, marginLeft: 10}}>

                        Close

                    </Button>

                </Modal.Footer>

            </Modal>

        </div>

    )

}

export default ImageBox;

UploadImageForm.jsx:

import { Modal, Form, Button, FormGroup } from "react-bootstrap";

import LoadingImage from "../pages/Loading";

function UploadImageForm(props) {

    function sendImage(){

        var name\_insert = document.getElementById("name\_insert").value;

        var url\_insert = document.getElementById("url\_insert").value;

        var tags\_insert = document.getElementById("tags\_insert").value;

        var server = 'http://kr/insert\_image.php';

        var formdata = new FormData();

        formdata.append('name', name\_insert);

        formdata.append('url', url\_insert);

        formdata.append('tags', tags\_insert);

        formdata.append('user\_id', props.user\_id);

        props.setImages(Array(<LoadingImage />));

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/x-www-form-urlencoded'},

            body : formdata

        })

        .then(response => { return response.json(); })

        .then(result\_data => {

            if(result\_data.sever\_error){

                alert(result\_data.sever\_error\_text);

                return false;

            }

            props.onHide();

            props.showImages();

        });

    }

    return(

        <>

            <Modal

            {...props}

            className="text-center"

            >

                <Modal.Header closeButton>

                    <Modal.Title>

                        Upload

                    </Modal.Title>

                </Modal.Header>

                <Modal.Body>

                    <Form>

                        <FormGroup>

                            <Form.Label>Set name</Form.Label>

                            <Form.Control

                            id = "name\_insert"

                            name = "name\_insert"

                            type = "text"

                            />

                        </FormGroup>

                        <FormGroup>

                            <Form.Label>Set url</Form.Label>

                            <Form.Control

                            id = "url\_insert"

                            name = "url\_insert"

                            type = "text"

                            />

                        </FormGroup>

                        <FormGroup>

                            <Form.Label>Set tags</Form.Label>

                            <Form.Control

                            id = "tags\_insert"

                            name = "tags\_insert"

                            type = "text"

                            />

                        </FormGroup>

                    </Form>

                </Modal.Body>

                <Modal.Footer>

                    <Button variant="success" onClick={() => sendImage()} style={{marginTop: 10}}>

                        Submit

                    </Button>

                    <Button variant="danger" onClick={props.onHide} style={{marginTop: 10, marginLeft: 10}}>

                        Close

                    </Button>

                </Modal.Footer>

            </Modal>

        </>

    )

}

export default UploadImageForm;

LoadingImage.jsx:

import ImageBox from "../Components/ImageBox";

function LoadingImage(){

    return(

        <>

            <ImageBox tags='' src="https://mir-s3-cdn-cf.behance.net/project\_modules/max\_1200/b6e0b072897469.5bf6e79950d23.gif" alt="Loading" name="Loading" />

        </>

    )

}

export default LoadingImage;

LoginForm.jsx:

import { Form, FormGroup, Button, Modal } from "react-bootstrap";

import "bootstrap/dist/css/bootstrap.min.css";

function LoginForm(props) {

    function validate(){

        var nickname\_log = document.getElementById("nickname\_log").value;

        var email = document.getElementById("email\_log").value;

        var pwd = document.getElementById("password\_log").value;

        var server = 'http://kr/login.php';

        var formdata = new FormData();

        formdata.append('nickname', nickname\_log);

        formdata.append('email', email);

        formdata.append('password', pwd);

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/x-www-form-urlencoded'},

            body : formdata

        })

        .then(response => { return response.json(); })

        .then(data => {

            if(!data.validated){

                alert(data.validation\_error\_text);

                return false;

            }

            if(data.sever\_error){

                alert(data.sever\_error\_text);

                return false;

            }

            props.setUserId(data.id);

            props.setUserName(data.nickname);

            props.onHide();

            props.setImages();

        });

    }

    return (

        <>

            <Modal

            {...props}

            >

                <Modal.Header>

                    <Modal.Title>

                        Sign in

                    </Modal.Title>

                </Modal.Header>

                <Modal.Body>

                    <Form>

                        <FormGroup>

                            <Form.Label>Enter your nickname</Form.Label>

                            <Form.Control

                            id="nickname\_log"

                            name="nickname\_log"

                            type="text"

                            placeholder="Your nickname"

                            />

                        </FormGroup>

                        <FormGroup style={{marginTop: 20}}>

                            <Form.Label>Enter email</Form.Label>

                            <Form.Control

                            id="email\_log"

                            name="email\_log"

                            type="email"

                            placeholder="example@email.com"

                            />

                        </FormGroup>

                        <FormGroup style={{marginTop: 20}}>

                            <Form.Label>Enter password</Form.Label>

                            <Form.Control

                            id="password\_log"

                            name="password\_log"

                            type="password"

                            pattern="^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\S+$).{6,}$"

                            />

                        </FormGroup>

                    </Form>

                </Modal.Body>

                <Modal.Footer>

                    <Button variant="success" onClick={() => validate()} style={{marginTop: 10}}>

                        Submit form

                    </Button>

                    <Button variant="danger" onClick={props.onHide} style={{marginTop: 10, marginLeft: 10}}>

                        Close

                    </Button>

                </Modal.Footer>

            </Modal>

        </>

    );

  }

  export default LoginForm;

MainPage.jsx:

import { Image, Navbar, InputGroup, FormControl, Button, Container } from "react-bootstrap";

import LoginForm from "./Login";

import RegisterForm from "./Registration";

import React, { useState, useEffect } from 'react';

import ImageBox from "../Components/ImageBox";

import LoadingImage from "./Loading";

import NoImage from "./NoImages";

import UploadImageForm from "../Components/UploadImageForm";

function MainPage() {

    const [modalLoginShow, setModalLoginShow] = useState(false);

    const [modalRegisterShow, setModalRegisterShow] = useState(false);

    const [modalUploadShow, setModalUploadShow] = useState(false);

    const [user\_id, setUserId] = useState(0);

    const [user\_name, setUserName] = useState("");

    const [images, setImages] = useState(Array(<NoImage />));

    function showImages(tags, user\_id){

        var server = 'http://kr/get\_images.php';

        var formdata = new FormData();

        formdata.append('tags', tags);

        setImages(Array(<LoadingImage />));

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/x-www-form-urlencoded'},

            body : formdata

        })

        .then(response => { return response.json(); })

        .then(result\_data => {

            if(result\_data.sever\_error){

                alert(result\_data.sever\_error\_text);

                return false;

            }

            if(result\_data.data.length > 0){

                var new\_images = Array();

                result\_data.data.map(image => {

                    new\_images.push(

                    <ImageBox

                        user\_id={user\_id}

                        name={image.name}

                        src={image.url}

                        CanEdit={((image.user\_id === user\_id) || (user\_id === '1'))}

                        tags={image.tags}

                        showImages = {()=>showImages(document.getElementById('input\_tag').value, user\_id)}

                        setImages = {setImages}

                    />)})

                setImages(new\_images);

            }else{

                setImages(Array(<NoImage />));

            }

        });

    }

    function setId(id){

        setUserId(id);

        showImages(document.getElementById('input\_tag').value, id);

    }

    useEffect(() => {

        showImages(document.getElementById('input\_tag').value, user\_id);

    }, []);

    return (

        <>

            <Navbar sticky='top' onChange={(e)=>showImages(e.target.value, user\_id)}

            bg = "dark">

                <Container>

                    <InputGroup>

                        <FormControl style={{marginRight:'20px', borderRadius: '10px'}}

                        placeholder="Enter image tag"

                        id = 'input\_tag'

                        />

                        {user\_name === '' &&

                        <Button variant="outline-secondary" onClick={() => setModalLoginShow(true)} style={{borderTopLeftRadius: '10px', borderBottomLeftRadius: '10px'}}>

                            Sign in

                        </Button>}

                        {user\_name === '' &&

                        <Button variant="outline-secondary" onClick={() => setModalRegisterShow(true)} style={{borderTopRightRadius: '10px', borderBottomRightRadius: '10px'}}>

                            Sign up

                        </Button>}

                        {user\_name !== '' &&

                        <Button

                        variant="outline-secondary"

                        onClick={() => {setId(0); setUserName('');}}

                        style={{borderTopLeftRadius: '10px', borderBottomLeftRadius: '10px'}}>Sign out</Button>}

                        {user\_name !== '' &&

                        <Button

                        variant="outline-secondary"

                        onClick={() => {setModalUploadShow(true);}}

                        style={{borderTopRightRadius: '10px', borderBottomRightRadius: '10px'}}>Upload image</Button>}

                    </InputGroup>

                </Container>

                {user\_name !== '' && <span style={{color: 'white', fontFamily: 'Lucida Console', marginRight: 20}}>{user\_name}</span>}

                {user\_name !== '' && <Image

                        style={{

                        width: 60,

                        height: 60,

                        }}

                        src="https://www.svgrepo.com/show/327465/person-circle.svg"

                        roundedCircle

                    />}

            </Navbar>

            <LoginForm

                show={modalLoginShow}

                onHide={() => {setModalLoginShow(false)}}

                setUserId = {setId}

                setUserName = {setUserName}

            />

            <RegisterForm

                show={modalRegisterShow}

                onHide={() => {setModalRegisterShow(false)}}

                setUserId = {setId}

                setUserName = {setUserName}

            />

            <UploadImageForm

                show={modalUploadShow}

                onHide={()=>{setModalUploadShow(false)}}

                user\_id = {user\_id}

                setImages = {setImages}

                showImages = {()=>showImages(document.getElementById('input\_tag').value, user\_id)}

            />

            {images}

        </>

    )

}

export default MainPage;

NoImage.jsx:

import ImageBox from "../Components/ImageBox";

function NoImage(){

    return(

        <>

            <ImageBox tags='' src='' alt="Loading" name="No image with such tag" />

        </>

    )

}

export default NoImage;

RegisterForm.jsx:

import { Form, FormGroup, Button, Modal } from "react-bootstrap";

import "bootstrap/dist/css/bootstrap.min.css";

function RegisterForm(props) {

    function validate(){

        var fpsw = document.getElementById("password").value;

        var fpsw\_repeat = document.getElementById("confirmpass").value;

        var nickname = document.getElementById("nickname").value;

        var email = document.getElementById("email").value;

        if(fpsw !== fpsw\_repeat)

        {

            alert("Passwords are not the same");

            return false;

        }

        var server = 'http://kr/registration.php';

        var formdata = new FormData();

        formdata.append('nickname', nickname);

        formdata.append('email', email);

        formdata.append('password', fpsw);

        fetch(server, {

            method : 'POST',

            header : {'Content-Type' : 'application/json'},

            body : formdata

        })

        .then(response => {return response.json(); } )

        .then(data => {

            if(!data.validated){

                alert(data.validation\_error\_text);

                return false;

            }

            if(data.sever\_error){

                alert(data.sever\_error\_text);

                return false;

            }

            props.setUserId(data.id);

            props.setUserName(data.nickname);

            props.onHide();

            props.setImages();

        });

    }

    return (

        <>

            <Modal

            {...props}

            >

                <Modal.Header>

                    Sign up

                </Modal.Header>

                <Modal.Body>

                    <Form>

                        <FormGroup>

                            <Form.Label>Enter your nickname</Form.Label>

                            <Form.Control

                                id="nickname"

                                name="nickname"

                                type="text"

                                placeholder="Your nickname"

                            />

                        </FormGroup>

                        <FormGroup style={{marginTop: 20}}>

                            <Form.Label>Enter email</Form.Label>

                            <Form.Control

                                id="email"

                                name="email"

                                type="email"

                                placeholder="example@email.com"

                            />

                        </FormGroup>

                        <FormGroup style={{marginTop: 20}}>

                            <Form.Label>Enter password</Form.Label>

                            <Form.Control

                                id="password"

                                name="password"

                                type="password"

                                pattern="^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\S+$).{6,}$"

                            />

                        </FormGroup>

                        <FormGroup style={{marginTop: 20}}>

                            <Form.Label>Comfirm password</Form.Label>

                            <Form.Control

                                id="confirmpass"

                                name="confirmpass"

                                type="password"

                                pattern="^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\S+$).{6,}$"

                            />

                        </FormGroup>

                    </Form>

                </Modal.Body>

                <Modal.Footer>

                    <Button type="submit" variant="success" onClick={() => validate()} style={{marginTop: 10}}>

                        Submit form

                    </Button>

                    <Button variant="danger" onClick={props.onHide} style={{marginTop: 10, marginLeft: 10}}>

                        Close

                    </Button>

                </Modal.Footer>

            </Modal>

        </>

    );

  }

  export default RegisterForm;

App.js:

import "./App.css";

import "bootstrap/dist/css/bootstrap.min.css";

import MainPage from "./pages/MainPage";

function App() {

  return (

    <>

      <MainPage />

    </>

  );

}

export default App;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import { BrowserRouter } from 'react-router-dom'

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

    <BrowserRouter>

        <App />

    </BrowserRouter>

);

index.html:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <title>Photos</title>

  </head>

  <body>

    <div id="root"></div>

  </body>

</html>

delete\_image.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "");

// Check connection

if (!$conn) {

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "Connection failed: " . mysqli\_connect\_error();

}

$name = $\_POST['name'];

$user\_id = $\_POST['user\_id'];

$db\_answer = mysqli\_query($conn, "DELETE FROM posts WHERE name = '$name' AND (user\_id = '$user\_id' OR '$user\_id' = '1')");

if (!$db\_answer) {

    if($response['sever\_error'] != true){

        $response['sever\_error'] = true;

        $response['server\_error\_text'] = "DB error! Can't delete! ";

      }

}

mysqli\_close($conn);

echo json\_encode($response);

?>

get\_images.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "",

                  'data' => array());

$tags = $\_POST['tags'];

$db\_answer = mysqli\_query($conn, "SELECT \* FROM posts WHERE (tags like '%$tags%')");

if ($db\_answer->num\_rows > 0) {

    while($row = $db\_answer->fetch\_assoc()) {

        $response['data'][] = array('name' => $row['name'],

                                    'url' => $row['url'],

                                    'tags' => $row['tags'],

                                    'user\_id' => $row['user\_id']);

    }

}

mysqli\_close($conn);

echo json\_encode($response);

?>

insert\_image.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "");

// Check connection

if (!$conn) {

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "Connection failed: " . mysqli\_connect\_error();

}

$name = $\_POST['name'];

$url = $\_POST['url'];

$tags = $\_POST['tags'];

$user\_id = $\_POST['user\_id'];

$db\_answer = mysqli\_query($conn, "SELECT \* FROM posts WHERE (name = '$name')");

if ($db\_answer) {

  $user = mysqli\_fetch\_assoc($db\_answer);

  if($user != null){

    if($response['sever\_error'] != true){

      $response['sever\_error'] = true;

      $response['server\_error\_text'] = "Such nickname already exist! ";

    }

  }else{

    $result = mysqli\_query($conn, "INSERT INTO posts (name, url, tags, user\_id) VALUES ('$name', '$url', '$tags', '$user\_id')");

    if (!$result) {

        if($response['sever\_error'] != true){

            $response['sever\_error'] = true;

            $response['server\_error\_text'] = "DB error! Can't insert! ";

        }

    }

  }

} else {

  if($response['sever\_error'] != true){

    $response['sever\_error'] = true;

    $response['server\_error\_text'] = "DB error! Can't get data! ";

  }

}

mysqli\_close($conn);

echo json\_encode($response);

?>

login.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "",

                  'validated'=>true,

                  'validation\_error\_text' => '',

                  'nickname' => '',

                  'id' => 0);

// Check connection

if (!$conn) {

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "Connection failed: " . mysqli\_connect\_error();

}

$nickname = $\_POST['nickname'];

$email = $\_POST['email'];

$pwd = $\_POST['password'];

//Validation

if(empty($nickname) or empty($email) or empty($pwd)){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Not all data was provided!";

}

if(!filter\_var($email, FILTER\_VALIDATE\_EMAIL)  and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Email is not correct! ";

}

if(!preg\_match("/^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])[0-9a-zA-Z]{8,}$/",$pwd)  and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Password must have at least 1 upper and lower case letter, 1 number and consist of 8 characters! ";

}

if(strlen($nickname) < 3 and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Nickname must be at least 3 characters! ";

}

$db\_answer = mysqli\_query($conn, "SELECT \* FROM users WHERE (user\_login = '$nickname')");

if (db\_answer) {

  $user = mysqli\_fetch\_assoc($db\_answer);

  if($user != null){

    if($pwd != $user['user\_password']){

      if($response['validated'] != false){

        $response['validated'] = false;

        $response['validation\_error\_text'] = "Invalid password! ";

      }

    }else{

      $response['nickname'] = $user['user\_login'];

      $response['id'] = $user['id'];

    }

  }else if($response['validated'] != false){

    $response['validated'] = false;

    $response['validation\_error\_text'] = "No such nickname registered! ";

  }

} else if($response['sever\_error'] != true){

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "DB error! ";

}

mysqli\_close($conn);

echo json\_encode($response);

?>

registration.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "",

                  'validated'=> true,

                  'validation\_error\_text' => '',

                  'nickname' => '',

                  'id' => 0);

// Check connection

if (!$conn) {

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "Connection failed: " . mysqli\_connect\_error();

}

$nickname = $\_POST['nickname'];

$email = $\_POST['email'];

$pwd = $\_POST['password'];

//Validation

if(empty($nickname) or empty($email) or empty($pwd)){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Not all data was provided!";

}

if(!filter\_var($email, FILTER\_VALIDATE\_EMAIL) and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Email is not correct! ";

}

if(!preg\_match("/^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])[0-9a-zA-Z]{8,}$/",$pwd) and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Password must have at least 1 upper and lower case letter, 1 number and consist of 8 characters! ";

}

if(strlen($nickname) < 3 and $response['validated'] != false){

  $response['validated'] = false;

  $response['validation\_error\_text'] = "Nickname must be at least 3 characters! ";

}

$db\_answer = mysqli\_query($conn, "SELECT \* FROM users WHERE (user\_login = '$nickname')");

if ($db\_answer) {

  $user = mysqli\_fetch\_assoc($db\_answer);

  if($user != null){

    if($response['validated'] != false){

      $response['validated'] = false;

      $response['validation\_error\_text'] = "Such nickname already exist! ";

    }

  }else{

    $result = mysqli\_query($conn, "INSERT INTO users (user\_login, user\_email, user\_password ) VALUES ('$nickname', '$email', '$pwd')");

    if ($result) {

      $last\_id = mysqli\_insert\_id($conn);

      $response['nickname'] = $nickname;

      $response['id'] = $last\_id;

    } else {

      if($response['sever\_error'] != true){

        $response['sever\_error'] = true;

        $response['server\_error\_text'] = "DB error! Can't insert! ";

      }

    }

  }

} else {

  if($response['sever\_error'] != true){

    $response['sever\_error'] = true;

    $response['server\_error\_text'] = "DB error! Can't get data! ";

  }

}

mysqli\_close($conn);

echo json\_encode($response);

?>

update\_image.php:

<?php

header('Access-Control-Allow-Origin: \*');

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "photos";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

$response = array('sever\_error' => false,

                  'server\_error\_text' => "");

// Check connection

if (!$conn) {

  $response['sever\_error'] = true;

  $response['server\_error\_text'] = "Connection failed: " . mysqli\_connect\_error();

}

$old\_name = $\_POST['old\_name'];

$name = $\_POST['name'];

$url = $\_POST['url'];

$tags = $\_POST['tags'];

$user\_id = $\_POST['user\_id'];

if($old\_name == $name){

    $db\_answer = mysqli\_query($conn, "UPDATE posts SET url = '$url', tags = '$tags' WHERE name = '$old\_name'");

    if (!$db\_answer) {

        if($response['sever\_error'] != true){

            $response['sever\_error'] = true;

            $response['server\_error\_text'] = "DB error! Can't delete! ";

        }

    }

}else{

    $db\_answer = mysqli\_query($conn, "SELECT \* FROM posts WHERE (name = '$name')");

    if ($db\_answer) {

      $user = mysqli\_fetch\_assoc($db\_answer);

      if($user != null){

        if($response['sever\_error'] != true){

          $response['sever\_error'] = true;

          $response['server\_error\_text'] = "Such name already exist! ";

        }

      }else{

        $db\_answer = mysqli\_query($conn, "UPDATE posts SET name = '$name', url = '$url', tags = '$tags' WHERE name = '$old\_name'");

        if (!$db\_answer) {

            if($response['sever\_error'] != true){

                $response['sever\_error'] = true;

                $response['server\_error\_text'] = "DB error! Can't update! ";

              }

        }

      }

    } else {

      if($response['sever\_error'] != true){

        $response['sever\_error'] = true;

        $response['server\_error\_text'] = "DB error! Can't get data! ";

      }

    }

}

mysqli\_close($conn);

echo json\_encode($response);

?>

**Висновок**

Під час виконання даної контрольної роботи було закріплення знання, отримані на курсі “Програмування в інтернеті”. Було розроблено просту вебсторінку, яка дає можливість пошуку зображень за їхніми тегами. Frontend було розроблено за допомогою React JS, а серверну частину на PHP. Для бази даних використано MySQL. Даний сайт надає можливість створення нових облікових записів та вхід у існуючі. Реалізовано валідацію даних засобами js для перевірки правильності вхідних даних, також валідація проходить і на сервері, для того, щоб перевірити, чи такий користувач існує та перевірити, чи дане ім’я не є зайнятим. Також сторінка надає різний функціонал в залежності від прав доступу користувача (адмін, звичайний користувач та не зареєстрований). Не зареєстровані користувачі можуть тільки переглядати картинки. Зареєстровані користувачі мають змогу публікації власних зображень. Також вони можуть видаляти та редагувати картинки, які вони опублікували. Адміністратор має всі згадані раніше можливості, проте він може редагувати та видаляти всі картинки.