ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра інженерії програмного забезпечення

ЛАБОРАТОРНА РОБОТА №8

з дисципліни «Веб-технології»

.(назва дисципліни)

на тему: «Створення Frontend. Profile»

ЛРІПЗс.2401075.08.07.ЛР

(шифр, назва)

Рівень вищої освіти                     Перший (бакалаврський)

Галузь знань                      12 «Інформаційні технології»

Спеціальність     121 «Інженерія програмного забезпечення»

Освітня програма   «Інженерія програмного забезпечення»

Студента. І  курсу, група . ІПЗс-24. \_\_\_\_\_\_\_\_\_\_\_ М.О. Лапко.

            (шифр)                         (підпис)     (ініціали, прізвище)

Викладач канд. техн. наук, доцент  .   \_\_\_\_\_\_\_\_\_\_\_ О.М. Яшина

              (посада, вчене звання, науковий ступінь)         (підпис)   (ініціали, прізвище)

Кількість балів \_\_\_\_\_\_\_

Оцінка за шкалою:

національною\_\_\_\_\_\_\_\_\_/ЄКТС\_\_\_\_

Хмельницький, 2024

ЗВІТ

**Варіант 7**

**Мета:** Розробити та впровадити Frontend для функціоналу сторінки профіля користувача, забезпечивши інтеграцію з backend через REST API.

**ХІД РОБОТИ**

**Завдання №1.** Підготовка проекту

Налаштування структури проекту.

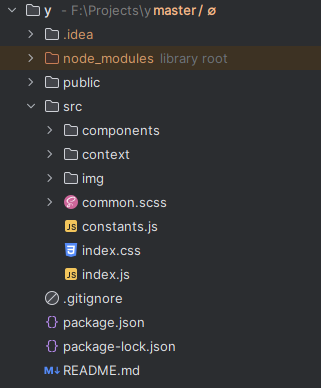


Рисунок 1 — Структура проекту

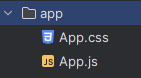


Рисунок 2 — Структура папки app

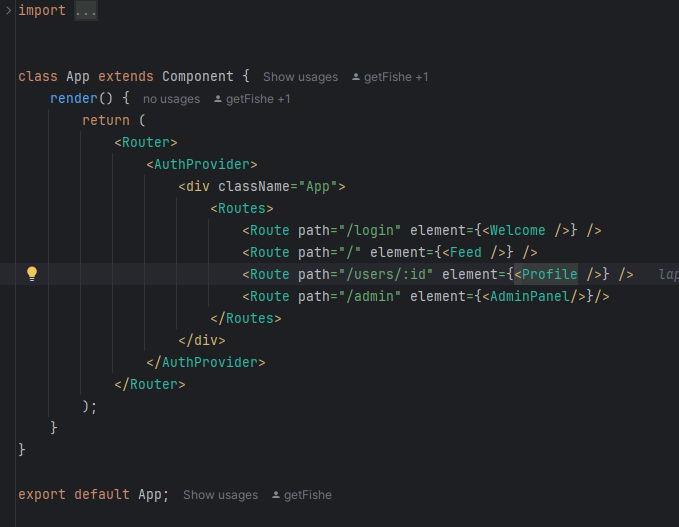


Рисунок 3 — Код компоненту App

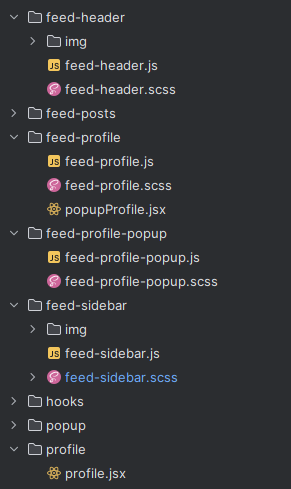


Рисунок 4 — Структура feed для profile

Код profile.jsx:

|  |
| --- |
| import { Component } from "react"; import {BrowserRouter as Router, Routes, Route, Link, useParams} from "react-router-dom";  import FeedHeader from "../feed-header/feed-header"; import FeedSidebar from "../feed-sidebar/feed-sidebar"; import FeedProfile from '../feed-profile/feed-profile'  export default function Profile(){  const { id } = useParams();   return (  <div className="App">  <div className="site-container">  <FeedHeader />  <FeedProfile userId={id}/>  <FeedSidebar />  </div>  </div>  ); } |

Код feed-header.js:

|  |
| --- |
| import { Link } from "react-router-dom";  import "./feed-header.scss"; import navHome from "./img/nav-home.svg"; import navProfile from "./img/user.svg"; import logo from "../../img/logo.svg"; import {useAuth} from "../../context/authContext";   const FeedHeader = () => {  const { user } = useAuth({id: 1})   return (  <header className="header">  <div className="container">  <div className="header\_\_container">  <nav className="nav">  <ul>  <li><Link to="/"><img src={logo} alt=""/></Link></li>  <li><Link to="/"><img src={navHome} alt=""/><span>Home</span></Link></li>  <li><Link to={`/users/${user.id}`}><img src={navProfile} alt=""/><span>Profile</span></Link></li>  </ul>  </nav>  </div>  </div>  </header>  ); };  export default FeedHeader; |

Код feed-header.scss:

|  |
| --- |
| @import "../mixins"; @import "../fonts"; @import "../vars";  .header {  width: rem(275);  flex-shrink: 0;  .container {  height: 100%;  }  &\_\_container {  padding: rem(12);  height: 100vh;  display: flex;  flex-direction: column;  position: sticky;  top: 0;  }   .nav {  li {  img {  width: rem(24);  height: rem(24);  }  svg {  width: rem(30);  height: rem(30);  }  margin-top: rem(8);  &:hover {  cursor: pointer;  a {  background-color: rgba(231,233,234,0.1);  }  }  a {  transition-duration: 0.2s;  border-radius: 99999px;  padding: rem(13);  display: flex;  align-items: center;  width: fit-content;  font-size: remF(19);  font-weight: bold;  >:not(:last-child) {  margin-right: rem(20);   }  span {  margin-right: rem(16);  color: var(--white);  }  }  }  }  button {  margin-top: rem(32);  }  .account {  margin-top: auto;  } } |

Код feed-profile.jsx:

|  |
| --- |
| import "./feed-profile.scss"; import React, {Component, useEffect, useState} from "react"; import postAva from "../feed-posts/img/tweet-logo1.png"; import postImg from "../feed-posts/img/tweet-img1.png"; import { handleInputFocus, handleInputBlur } from "../popup/popup"; import Popup from "../popup/popup"; import {useAuth} from "../../context/authContext"; import {*constants*} from "../../constants"; import postSettingsImg from "../feed-posts/img/dots.svg"; import Comment from "../feed-posts/post-comments"; import Post from "../feed-posts/Post";  export default function FeedProfile({userId}) {  const [isPopupVisible, setIsPopupVisible] = useState(false);  const [user, setUser] = useState(null);  const [posts, setPosts] = useState([]);  const [postStart, setPostStart] = useState(0);  const [postEnd, setPostEnd] = useState(50);  const {user: currentUser, sendRequest: sendRequest, updateUser: updateUser, getUser: getUser} = useAuth()   useEffect(() => {  async function fetchUser() {  setUser(await getUser(userId))  }  fetchUser()  }, []);    function changeAvatar(e) {  const file = e.target.files[0];  if (!file){  alert("File required");  return;  }   *console*.log(file)   if (!file.type.includes("image/")) {  alert("Image required");  return;  }   *console*.log(file)   const formData = new FormData();  formData.append("avatar", file);   for (let [key, value] of formData.entries()) {  *console*.log(key, value);  }   sendRequest({  url: `${*constants*.userApiV1}/users/${user.id}`,  method: 'PATCH',  data: formData  }).then((response) => response.text())  .then((result) => {  *console*.log(result);  updateUser();  }).catch((error) => *console*.error(error));  }   useEffect(() => {  if (!user) {  return;  }   fetch(`${*constants*.postApiV1}/posts/users/${user.id}?skip=${postStart}&limit=${postEnd}`)  .then((response) => {  if (!response.ok) {  throw new Error(`HTTP error! Status: ${response.status}`);  }  return response.json();  })  .then((posts) => {  setPosts(posts)  *console*.log(posts);  })  .catch((error) => {  *console*.error("error:", error);  });  }, [user])   return (  <section className="feed-profile">  <div className="container">  <div className="feed-profile\_\_head">  <div className="feed-profile\_\_title">  <div className="feed-profile\_\_title-avatar" onClick={() => {if (user && currentUser && currentUser.id === user.id) *document*.getElementById("fileInput").click()}}>  {user && currentUser && currentUser.id === user.id && <input id={'fileInput'} type='file' onChange={changeAvatar} style={{display: "none"}}/>}  <img src={user && user.avatar} alt="avatar"/>  </div>  {  user && currentUser && currentUser.id === user.id &&  <div className="feed-profile\_\_title-btn">  <button onClick={() => {  setIsPopupVisible(true)  }}><span>Change profile</span></button>  </div>  }  </div>  <div className="feed-profile\_\_info">  <div className="feed-profile\_\_info-name"><span>{user && user.nickname}</span></div>  <div className="feed-profile\_\_info-username"><span>@{user && user.username}</span></div>  </div>  </div>  <div className="feed-profile\_\_tabs">  <div data-tab="1" className='feed-profile\_\_tabs-tab active'>  <span>Posts</span>  </div>  </div>  <div className="feed-profile\_\_content">  {posts.map((post) => (  <Post key={post.id} postData={post}/>  ))}  </div>  </div>  {isPopupVisible && <Popup isOpen={isPopupVisible} onClose={() => {  setIsPopupVisible(false)  }}><ChangeProfilePopup user={user}/></Popup>}  </section>  ); }  function ChangeProfilePopup({user}) {  const [nickname, setNickname] = useState(user.nickname);  const [username, setUsername] = useState(user.username);  const [pass, setPass] = useState('');  const [confirmPass, setConfirmPass] = useState('');  const [error, setError] = useState('');   const {updateUser, sendRequest, changePass} = useAuth();   const [isBtnBlocked, setBtnBlocking] = useState(false);   const maxLength = 52;   async function submitForm(event) {  event.preventDefault();  setError('');  setBtnBlocking(true);   const formData = new FormData();   if (user.nickname !== nickname && nickname.length > 0) {  formData.append('nickname', nickname);  }   if (user.username !== username && *constants*.loginRegex.test(username)) {  formData.append('username', username);  }   const response = await sendRequest({  url: `${*constants*.userApiV1}/users/${user.id}`,  method: 'PATCH',  body: formData  })   if (!response.ok) {  const json = await response.json();  setError(`Error updating profile: ${json.message}`);  }   if (pass.length > 0) {  if (pass === confirmPass && *constants*.passwordRegex.test(pass)) {  await changePass(pass);  setPass('')  setConfirmPass('');  } else {  setError('Passwords do not match or are invalid');  }  }   updateUser()  setBtnBlocking(false);    }   function usernameOnChange(event) {  setUsername(event.target.value);   const popupInput = event.currentTarget.closest(".popup-input");   if (username.length > maxLength) {  popupInput.classList.add("error");  } else {  popupInput.classList.remove("error");  }   }    function nicknameOnChange(event) {  setNickname(event.target.value);   const popupInput = event.currentTarget.closest(".popup-input");   if (nickname.length > maxLength) {  popupInput.classList.add("error");  } else {  popupInput.classList.remove("error");  }   }   function passwordOnChange(event) {  setPass(event.target.value);   const popupInput = event.currentTarget.closest(".popup-input");   if (username.length > maxLength) {  popupInput.classList.add("error");  } else {  popupInput.classList.remove("error");  }  }   return (  <form onSubmit={submitForm} className="popup-tab">  <div className="popup-title"><h3>Change your account</h3></div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Nickname</span>  <span className='popup-input\_\_text-symbols'>{nickname.length}/{maxLength}</span>  </div>  <input onBlur={handleInputBlur} onChange={nicknameOnChange} type="text" value={nickname}/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Username</span>  <span className='popup-input\_\_text-symbols'>{username.length}/{maxLength}</span>  </div>  <input onBlur={handleInputBlur} type="text" onChange={usernameOnChange} value={username}/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>New password</span>  </div>  <input onBlur={handleInputBlur} type="password" onChange={passwordOnChange} value={pass}/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Confirm password</span>  </div>  <input onBlur={handleInputBlur} type="password" onChange={(event) => {setConfirmPass(event.target.value)}} value={confirmPass}/>  </div>  <button disabled={isBtnBlocked} type='submit' className="popup-confirm">  <span>Confirm</span>  </button>  {error && <div className="popup-error">{error}</div>}  </form>  ); } |

Код feed-profile.scss:

|  |
| --- |
| @import "../mixins"; @import "../fonts"; @import "../vars";   .feed-profile {  width: rem(600);  min-height: 120vh;  flex-shrink: 0;  border-left: 1px solid var(--border-color);  border-right: 1px solid var(--border-color);  &\_\_head {   }  &\_\_title {  display: flex;  align-items: center;  justify-content: space-between;  margin-top: rem(60);  &-avatar {  width: rem(135);  height: rem(135);  border-radius: 50%;  border:1px solid black;  overflow: hidden;  cursor: pointer;  }  &-btn {  padding: rem(5) rem(16);  border:1px solid white;  border-radius: 99999px;  transition: background-color var(--hover-transition);  margin-top: rem(50);  &:hover {  background-color: #272727;  }  }  }  &\_\_info {  text-align: left;  margin-top: rem(25);  &-name {  font-size: remF(20);  line-height: remF(23);  font-weight: 700;  color: #E7E9EA;  margin-bottom: rem(10);  }  &-username,&-registered {  color: #71767B;  font-size: remF(15);  line-height: remF(22);  }  }  &\_\_tabs {  display: flex;  align-items: center;  margin: rem(25) 0 0 rem(-12);  &:not(:last-child) {  margin-right: rem(40);  }  &-tab {  cursor: pointer;  padding: rem(16);  transition: background-color var(--hover-transition);  color: #71767B;  font-weight: 600;  position: relative;  &:hover {  background-color: #272727;  }  &.active {  color: white;  &::before {  content: '';  position: absolute;  bottom: 0;  left: 50%;  transform: translateX(-50%);  width: calc(100% - rem(32));  height: 5px;  background-color: var(--btn-color);  border-radius: 99999px;  }  }  }  }  &\_\_content {  margin: 0 rem(-12);  }  &\_\_page {  &-tab {   }  } } |

Код popupProfile.jsx:

|  |
| --- |
| import {handleInputFocus,handleInputBlur} from "../popup/popup"; import { *constants* } from '../../constants' import { useState } from 'react'   export async function popupAuth() {  *event*.preventDefault();   const [nickname, setNickname] = useState('');  const [username, setUsername] = useState('');  const [pass, setPass] = useState('');  const [confirmPass, setConfirmPass] = useState('');    const [error, setError] = useState('');  const [isBtnBlocked,setBtnBlocking] = useState(false);   setBtnBlocking(true)   setError('');   async function dataPickUp(event) {  try {  if (nickname !== nickname) {  setError('Confirm Password Error');  return;  }  if (!username.username) {  setError('Username Error');  return;  }  if (!*constants*.passwordRegex.test(pass)) {  setError('Password Error');  return;  }  if (pass !== confirmPass) {  setError('Confirm Password Error');  return;  }   let changeUser = await fetch(`${*constants*.userApiV1}/login`, {  method: 'POST',  headers: {  'Content-Type': 'application/json;charset=utf-8',  },  body: *JSON*.stringify(user),  });   let json = await response.json();  if (!response.ok) {  setError(`Error request: ${json.message}`);  return;  }   if (!activate.ok) {  setError(`Error request: ${responseActivationToken.message}`);  return;  }   setAccessToken(responseActivationToken.token);  navigate("/feed");  } catch (error) {  setError(`Unexpected error: ${error.message}`);  } finally {  setBtnBlocking(false)  }   }        return (  <form onSubmit={dataPickUp} className="popup-tab">  <div className="popup-title"><h3>Create your account</h3></div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Name</span>  <span className='popup-input\_\_text-symbols'>0/52</span>  </div>  <input onBlur={handleInputBlur} type="text"/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Email</span>  </div>  <input onBlur={handleInputBlur} type="text"/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Password</span>  <span className='popup-input\_\_text-symbols'>0/30</span>  </div>  <input onBlur={handleInputBlur} type="password"/>  </div>  <div onClick={handleInputFocus} className="popup-input">  <div className="popup-input\_\_text">  <span className='popup-input\_\_text-name'>Confirm password</span>  </div>  <input onBlur={handleInputBlur} type="password"/>  </div>  <button disabled={isBtnBlocked} type='submit' className="popup-next">  <span onClick={*window*.close}>Confirm</span>  </button>  {error && <div className="popup-error">{error}</div>}  </form>  ) } |

Код feed-profile-popup.jsx:

|  |
| --- |
| import { Component } from "react"; import "./feed-profile-popup.scss";  class Popup extends Component {  constructor(props) {  super(props);  this.state = {   };  }    handleInputFocus = (event) => {  const popupInput = event.currentTarget;  const input = popupInput.querySelector("input");   if (!popupInput.classList.contains("focus")) {  input.focus();  popupInput.classList.add("focus");  }  };   handleInputBlur = (event) => {  const popupInput = event.currentTarget.closest(".popup-input");  if (!event.currentTarget.classList.contains("popup-input")) {  popupInput.classList.remove("focus");  }  }; }   export default Popup; |

Код feed-sidebar.js:

|  |
| --- |
| import "./feed-sidebar.scss" import {Component, useEffect} from "react";  import dots from './img/dots.svg' import settings from './img/settings.svg' import search from './img/search.svg' import logo from "../feed-header/img/logo.svg"; import {useAuth} from "../../context/authContext"; import {useNavigate} from "react-router-dom";     function FeedSidebar() {  const {user} = useAuth();   useEffect(() => {  *console*.log("Current user:", user)  }, [user]);   return (  <div className="feed-sidebar">  <div className="container">  {user &&  <a className="account" href={`/users/${user.id}`}>  <div className="account-avatar"><img src={user.avatar} alt={logo}/></div>  <div className="account\_\_text">  <span className="account\_\_text-name">{user.nickname}</span>  <span className="account\_\_text-id">@{user.username}</span>  </div>  </a>  }  </div>  </div>  ) }  export default FeedSidebar; |

Код feed-sidebar.scss:

|  |
| --- |
| @import "../mixins";  .feed-sidebar {  width: 100%;   .container {  height: 100%;  }   .account {  width: rem(200);  position: sticky;  top: rem(20);  }   &>:not(:last-child) {  margin-bottom: rem(12);  }  &\_\_search {  cursor: text;  display: flex;  align-items: center;  padding: rem(12) 0;  background-color: var(--dark-blue);  border-radius:42px;  img {  margin-left: rem(12);  }  input {  &::placeholder {  font-size: remF(15);  line-height:remF(18);  color: var(--light-gray);  }  }  &>:not(:last-child) {  margin-right: rem(24);  }  }  &\_\_relevant {  background-color:var(--dark-blue);  border-radius: rem(15);  text-align: left;  overflow: hidden;  }  &\_\_title {  padding: rem(16) rem(16) 0;  display: flex;  align-items: center;  justify-content: space-between;  margin-bottom: rem(12);  img {  width: rem(20);  height: rem(20);  }  }  &\_\_topics {  transition: background-color 300ms ease-in-out;  &:hover {  background-color: #1c1e22;  }  &-elem {  cursor: pointer;  padding:rem(12) rem(16);  span {  font-size: remF(13);  line-height: remF(16);  font-weight: 700;  color: var(--light-gray);  }  >:not(:last-child) {  margin-bottom: rem(4);  }  &\_title {  display: flex;  justify-content: space-between;  align-items: center;  span {  font-size: remF(15);  line-height: remF(16);  color: white;  }  }  }  }  &\_\_btn {  padding: rem(16);  cursor: pointer;  transition: background-color 300ms ease-in-out;  &:hover {  background-color: #1c1e22;  }  span{  font-size: remF(15);  line-height: remF(18);  font-weight: 700;  color: var(--btn-color);  }  } } |

При натисканні на nickname користувача, сторінка змінюєтся на його профіль.

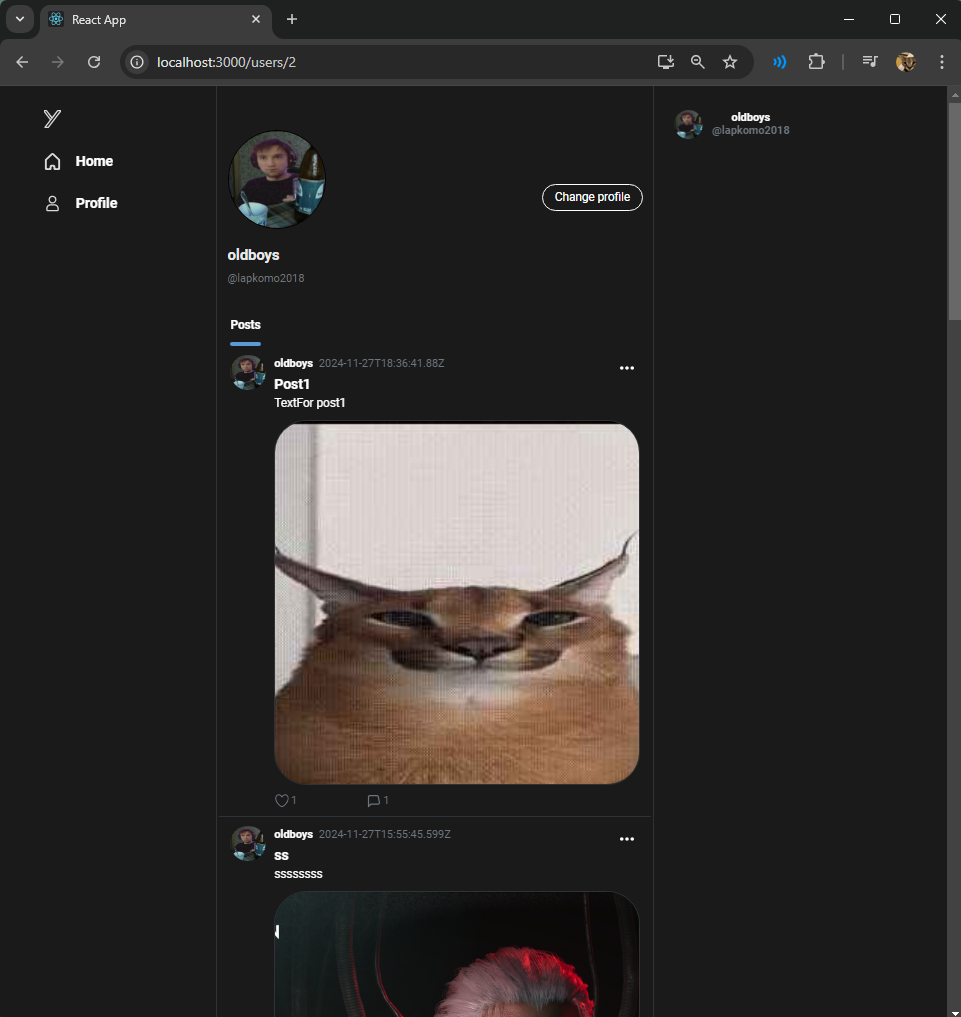


Рисунок 5 — Рисунок сторінки користувача oldboys

Через те, що це сторінка власника акаунта, йому доступна кнопка Change Profile для зміни даних профілю.

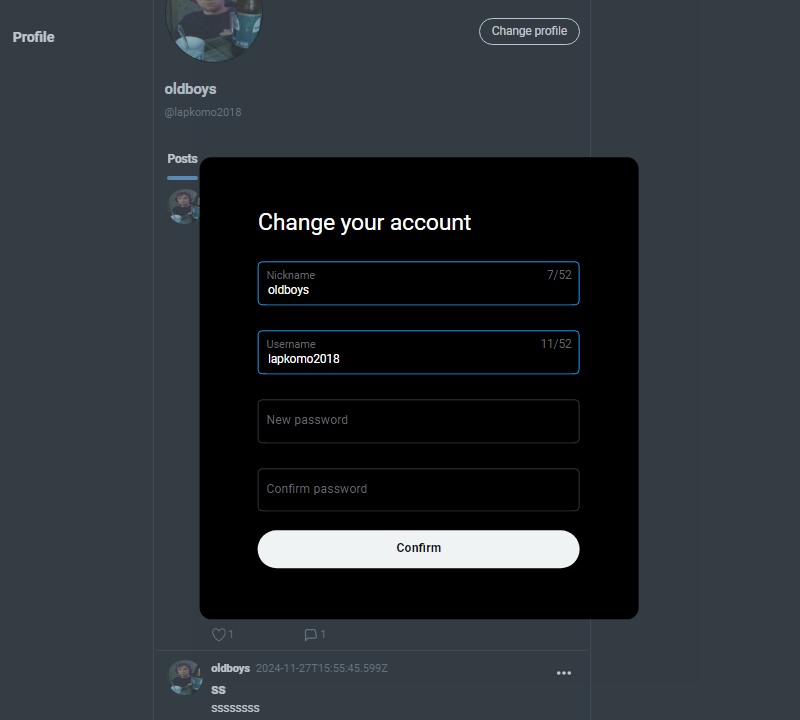


Рисунок 6 — Рисунок діалогового вікна Change profile

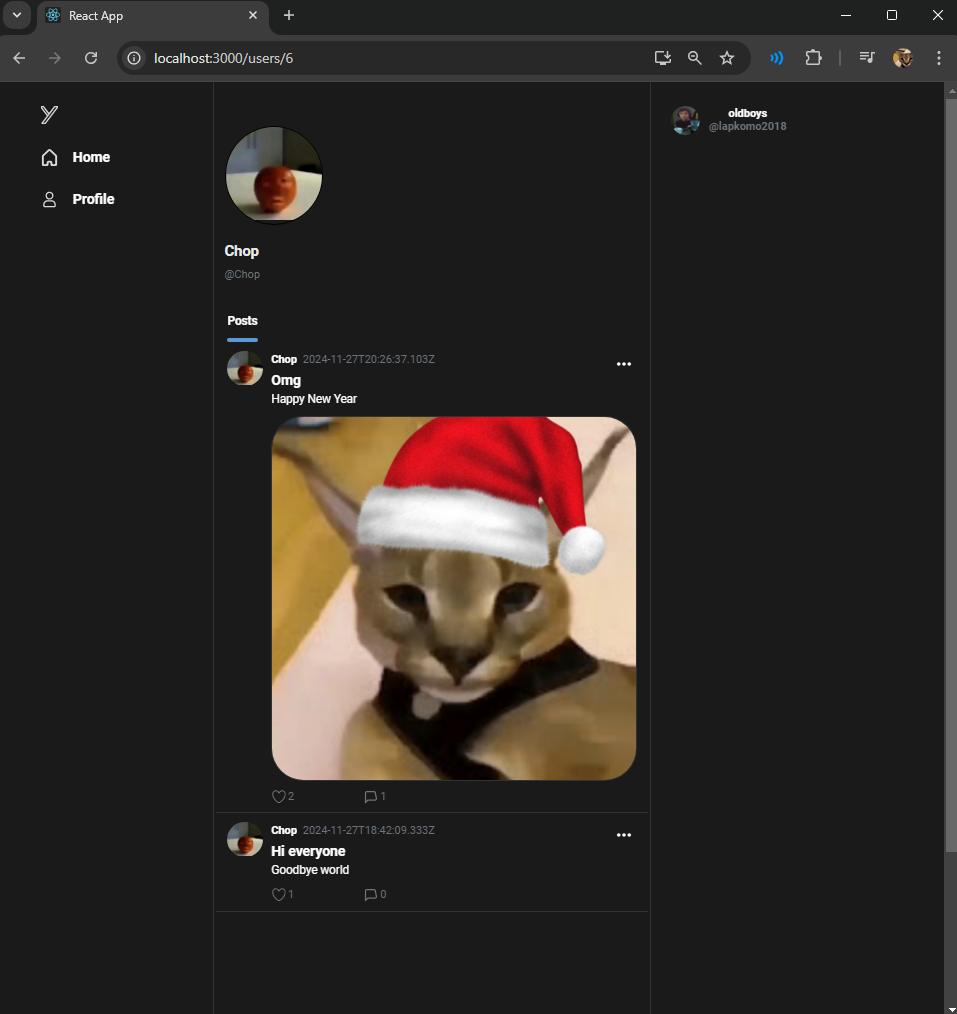


Рисунок 7 — Рисунок сторінки користувача Chop