ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра інженерії програмного забезпечення

ЛАБОРАТОРНА РОБОТА №7

з дисципліни «Веб-технології»

.(назва дисципліни)

на тему: «Створення Frontend. Feed»

ЛРІПЗс.2401075.07.07.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 головної сторінки з постами, забезпечивши інтеграцію з backend через REST API.

**ХІД РОБОТИ**

**Завдання №1.** Підготовка проекту

Налаштування структури проекту.

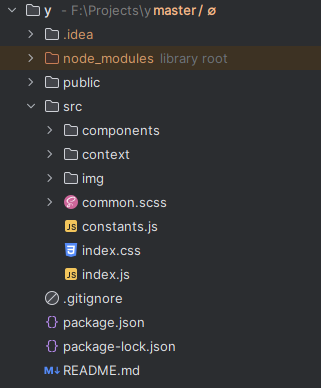


Рисунок 1 — Структура проекту

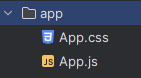


Рисунок 2 — Структура папки app

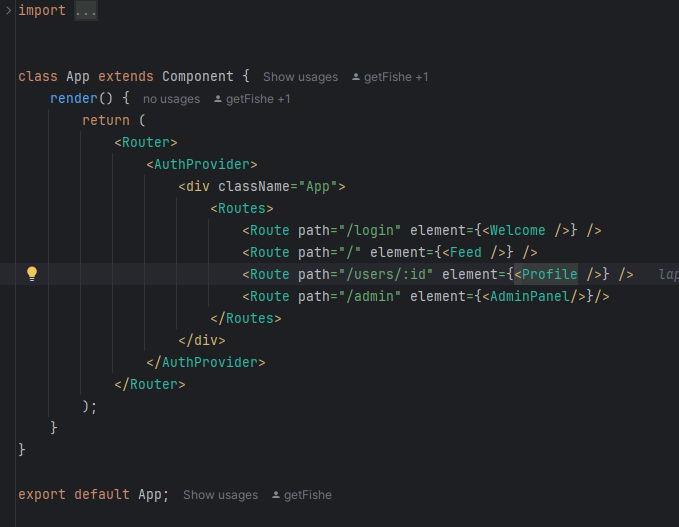


Рисунок 3 — Код компоненту App

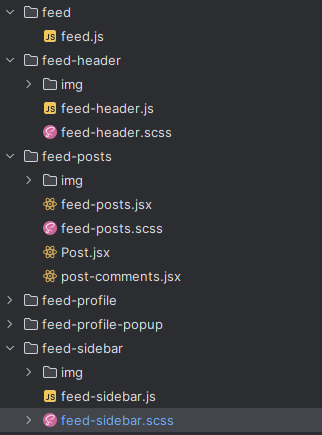


Рисунок 4 — Структура feed

Код feed.js:

|  |
| --- |
| import { Component } from "react"; import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";  import FeedHeader from "../feed-header/feed-header"; import FeedPosts from "../feed-posts/feed-posts"; import FeedSidebar from "../feed-sidebar/feed-sidebar";  class Feed extends Component {  render() {  return (  <div className="App">  <div className="site-container">  <FeedHeader />  <FeedPosts />  <FeedSidebar />  </div>  </div>  );  } }  export default Feed; |

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-posts.jsx:

|  |
| --- |
| import React, { Component, useState } from 'react' import {useEffect} from 'react' import "./feed-posts.scss";  import Post from "./Post"; import search from "../feed-sidebar/img/search.svg"; import { *constants* } from '../../constants' import { useAuth } from '../../context/authContext' import uploadImg from './img/svgrepo-com.svg'   export default function FeedPosts() {  const [posts, setPosts] = useState([]);  const [postStart, setPostStart] = useState(0);  const [postEnd, setPostEnd] = useState(50);  const [newPost, setNewPost] = useState({ title: "", content: "", tags: [], images: [], videos: [] });  const {sendRequest} = useAuth()   useEffect(() => {  fetch(`${*constants*.postApiV1}/posts?skip=${postStart}&limit=${postEnd}`)  .then((response) => {  if (!response.ok) {  throw new Error(`HTTP error! Status: ${response.status}`);  }  return response.json();  })  .then((posts) => {  setPosts(posts)  *console*.log(posts);  })  .catch((error) => {  *console*.error("error:", error);  });  }, [])   function handleAddPost (){  if (!newPost.title || !newPost.content) return;   const formData = new FormData();  formData.append('title', newPost.title)  formData.append('content', newPost.content)  newPost.tags.forEach((tag) => {  formData.append("tags", tag)  })  newPost.images.forEach((image) => formData.append(`images`, image));  newPost.videos.forEach((video) => formData.append(`videos`, video));   console.log(newPost)  sendRequest({  url: `${constants.postApiV1}/posts`,  method: 'POST',  body: formData,  }).then(async (response) => {  if (!response.ok) {  throw new Error(`HTTP error! Status: ${response.status}`);  }  return response.json();  }).then(post => {  setPosts((prevPosts) => [post, ...prevPosts]);  setNewPost({ title: "", content: "", tags: [], videos: [], images: [] });  }).catch((error) => {  console.error("error:", error);  })  }   function handleInputChange (e){  const { name, value } = e.target;  setNewPost((prev) => ({ ...prev, [name]: value }));  }   const handleFilesChange = (e) => {  const selectedFiles = Array.from(e.target.files);   const images = selectedFiles.filter((file) => file.type.startsWith("image/"));  const videos = selectedFiles.filter((file) => file.type.startsWith("video/"));   setNewPost((prevPost) => ({  ...prevPost,  images: images,  videos: videos,  }));  };   return (  <section className="feed">  <div className="feed\_\_container">  <div className="feed\_\_title">  <div className="feed\_\_title-search">  <img src={search} alt=""/>  <input type="text" placeholder="Search Y"/>  </div>  </div>  <div>  <div className="feed\_\_create">  <div className="feed\_\_create-inner">  <div className="feed\_\_create-inputs">  <input  type="text"  name="title"  value={newPost.title}  placeholder="Title"  onChange={handleInputChange}  />  <input  type="text"  name="tags"  placeholder="Type Tags"  onChange={(e) => {  setNewPost((prevPost) => ({  ...prevPost,  tags: e.target.value.split(',').map(tag => tag.trim()).filter((tag) => tag),  }))  }}  />  <textarea  name="content"  value={newPost.content}  placeholder="Write your post..."  onChange={handleInputChange}  ></textarea>  </div>  </div>  <div className="feed\_\_create-buttons">  <label className="feed\_\_create-input">  <img src={uploadImg} alt=""/>  <input name="" type="file" onChange={handleFilesChange}/>  </label>  <button className='btn' onClick={handleAddPost}>Add Post</button>  </div>  </div>   <div className="posts-list">  {posts.map((post) => (  <Post key={post.id} postData={post}/>  ))}  </div>  </div>  </div>  </section>  ); } |

Код feed-posts.scss:

|  |
| --- |
| @import "../mixins";  .feed{  flex-shrink: 0;  width: rem(600);  border-left: 1px solid var(--border-color);  border-right: 1px solid var(--border-color);  &\_\_container {  >\* {  border-bottom: 1px solid var(--border-color);  }  }  &\_\_title {  padding: rem(15) rem(16);  display: flex;  justify-content: space-between;  &-search {  width: 100%;  margin-left: auto;  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 {  width: 100%;  padding-right: rem(20);  &::placeholder {  font-size: remF(15);  line-height:remF(18);  color: var(--light-gray);  }  }  &>:not(:last-child) {  margin-right: rem(24);  }  }  }  &\_\_create {  border-bottom: 1px solid var(--border-color);  &-avatar {  img {  width: rem(48);  height: rem(48);  border-radius: 50%;  }  }  &-inner {  display: flex;  width: 100%;  padding: rem(20) rem(12) 0 ;  >:not(:last-child) {  margin-right: rem(20);  }  }  &-inputs {  padding-bottom: rem(10);  display: flex;  width: 100%;  flex-direction: column;  >\* {  padding: 0 rem(15);  &::placeholder {  font-size: remF(18);  }  }  &>:not(:last-child) {  margin-bottom: rem(20);  }  input {  height: rem(40);  border-bottom: 1px solid var(--border-color);  }  textarea {  padding: rem(15);  resize: none;  width: 100%;  height: rem(130);  &:focus {  outline:none;  }  }  }  &-buttons {  position: relative;  display: flex;  align-items: center;  justify-content: space-between;  padding: rem(20);  &::before {  border-top: 1px solid var(--border-color);  content: '';  position: absolute;  top: 0;  width: calc(100% - rem(24));  left: 50%;  transform: translateX(-50%);  }  }  &-input {  position: relative;  width: 40px;  height: 40px;  border-radius: 50%;  padding: 10px;  transition: background-color 200ms ease-in-out;  &:hover {  background-color: #292929;  }  input {  position: absolute;  left: -999999px;  }  }  }  .create-tweet {  &:hover {  background-color: #232323;  transition: background-color 0.2s;  }  display: flex;  padding: rem(12) rem(16);  >:not(:last-child) {  margin-right: rem(12);  }  &\_\_btn {  background-color: var(--btn-color);  transition: background-color var(--hover-transition);  padding: rem(9) rem(16);  border-radius: 36px;  margin-left: auto;  &:hover {  background-color: var(--btn-color-hover);  }  }  &\_\_avatar {  flex-shrink: 0;  width: rem(48);  height: rem(48);  border-radius: 50%;  overflow: hidden;  }  &\_\_content {  width: 100%;  textarea {  resize: none;  width: 100%;  min-height: rem(50);  &::placeholder {  font-size: remF(20);  line-height: remF(24);  color: var(--gray);  }  &:active, &:hover, &:focus {  outline: 0;  outline-offset: 0;  }  }  &-btns {  display: flex;  align-items: center;  }  }  } } |

Код Post.jsx:

|  |
| --- |
| import React, {useEffect, useRef, useState} from 'react' import postAva from './img/tweet-logo1.png'; import Comment from './post-comments'; import postSettingsImg from './img/dots.svg' import { useAuth } from '../../context/authContext' import {*constants*} from "../../constants";  export default function Post({ postData }) {  const [newComment, setNewComment] = useState("");  const [showComments, setShowComments] = useState(false);  const [postOwner, setPostOwner] = useState(null);  const [post, setPost] = useState(postData);  const {user, sendRequest, getUser} = useAuth();  const [showSettings, setShowSettings] = useState(false);  const [value, setValue] = useState('');  const textareaRef = useRef(null)  ;  useEffect(() => {  async function fetchPostOwner() {  setPostOwner(await getUser(post.user\_id))  }  fetchPostOwner();  }, []);    function updatePostInfo() {  sendRequest({  url: `${*constants*.postApiV1}/posts/${post.id}`,  method: 'GET',  }).then(async (response) => {  if (!response.ok) {  *console*.log('Error liking post: ', response);  return  }  return await response.json();  }).then((post) => {  setPost(post)  })  }   const handleCommentChange = (e) => {  setNewComment(e.target.value);  setValue(e.target.value);  };   function likePost() {  sendRequest({  url: `${*constants*.postApiV1}/posts/${post.id}/${post.is\_liked ? 'unlike' : 'like'}`,  method: 'GET',  }).then((response) => {  if (!response.ok) {  *console*.log('Error liking post: ', response);  return  }  updatePostInfo()  })  }    function commentPost() {  if (newComment.trim() === "") {  return;  }  sendRequest({  url: `${*constants*.postApiV1}/posts/${post.id}/comments`,  method: 'POST',  headers: {  "Content-Type": "application/json",  },  body: *JSON*.stringify({ content: newComment }),  }).then((response) => {  if (!response.ok) {  *console*.log('Error commenting post: ', response);  return  }  updatePostInfo()  })  setNewComment("");  }    useEffect(() => {  if (textareaRef.current) {  textareaRef.current.style.height = 'auto';  textareaRef.current.style.height = `${textareaRef.current.scrollHeight}px`;  }  }, [value]);   return (  <div className="tweet">  <div className="tweet\_\_inner">  <a href="/" className="tweet\_\_avatar"><img src={postOwner && postOwner.avatar} alt=""/></a>  <div className="tweet\_\_information">  <div className={`tweet\_\_settings ${showSettings ? 'show' : ''}`}>  <div onClick={() => {  setShowSettings(!showSettings)  }} className={"tweet\_\_settings-btn "}><img src={postSettingsImg} alt=""/></div>  <div className="tweet\_\_settings-list">  <ul>  {(post.user\_id !== user.id && !user.admin) && <li><span>Report</span></li>}   {(user.admin || post.user\_id === user.id) &&  <>  <li><span>Edit</span></li>  <li><span>Delete</span></li>  </>  }  </ul>  </div>  </div>  <div className="tweet\_\_title">  {/\*<a href="/" className="tweet\_\_title-name">{postData.name}</a>\*/}  <a href={`users/${postOwner ? postOwner.id : 1}`} className={`tweet\_\_title-name`}>{postOwner ? postOwner.nickname : 'postownernick'}</a>  <div className="tweet\_\_title-tag">{post.tag}</div>  <div className="tweet\_\_title-time">{post.posted\_at}</div>  </div>  <div className="tweet\_\_content">  <span>{post.title}</span>  <p>{post.content}</p>  {  post && post.image\_urls && post.image\_urls.length > 0 &&  <div className="tweet\_\_content-img">  <img src={post.image\_urls[0]} alt='post img'/>  </div>  }  </div>  <div className="tweet\_\_controls">  <div  className={`tweet\_\_btn tweet\_\_btn--like ${post && post.is\_liked && 'active'}`}  onClick={likePost}  >  <div className="tweet\_\_btn-icon">  <svg  width="18"  height="17"  viewBox="0 0 18 17"  fill="none"  xmlns="http://www.w3.org/2000/svg"  >  <path  d="M9 16.0317H8.98833C6.83583 15.9917 0.625 10.38 0.625 5.065C0.625 2.51167 2.72917 0.27 5.1275 0.27C7.03583 0.27 8.31917 1.58667 8.99917 2.545C9.6775 1.58833 10.9608 0.27 12.87 0.27C15.27 0.27 17.3733 2.51167 17.3733 5.06583C17.3733 10.3792 11.1617 15.9908 9.00917 16.03H9V16.0317ZM5.12833 1.52083C3.395 1.52083 1.87583 3.1775 1.87583 5.06667C1.87583 9.85 7.7375 14.73 9.00083 14.7817C10.2658 14.73 16.1258 9.85083 16.1258 5.06667C16.1258 3.1775 14.6067 1.52083 12.8733 1.52083C10.7667 1.52083 9.59 3.9675 9.58 3.99167C9.38833 4.46 8.61667 4.46 8.42417 3.99167C8.4125 3.96667 7.23667 1.52083 5.12917 1.52083H5.12833Z"  fill="#6E767D"  />  </svg>  </div>  <span>{post.likes\_amount}</span>  </div>   <div  className="tweet\_\_btn tweet\_\_btn--comments"  onClick={() => {  setShowComments(!showComments)  }}  >  <div className="tweet\_\_btn-icon">  <svg  xmlns="http://www.w3.org/2000/svg"  viewBox="0 0 24 24"  fill="none"  stroke="#6E767D"  strokeWidth="2"  strokeLinecap="round"  strokeLinejoin="round"  width="18"  height="18"  >  <path d="M21 15a2 2 0 0 1-2 2H7l-4 4V5a2 2 0 0 1 2-2h12a2 2 0 0 1 2 2z"></path>  </svg>  </div>  <span>{post.comments\_amount}</span>  </div>  </div>  </div>  </div>  {showComments && (  <div className="tweet\_\_comments">  <div className="tweet\_\_comments-controls">  <textarea  value={newComment}  onChange={handleCommentChange}  placeholder="Write a comment..."  />  <button className="btn tweet\_\_comments-controls\_btn" onClick={commentPost}>  Add  </button>  </div>  <div className="tweet\_\_comments-list">  {post.comments.map((comment) => (  <Comment key={comment.id} commentData={comment}/>  ))}  </div>  </div>  )}  </div>  ); } |

Код post-comments.jsx:

|  |
| --- |
| import React, {useEffect, useState} from "react"; import {useAuth} from "../../context/authContext";  export default function Comment({commentData}) {  const [comment, setComment] = useState(commentData);  const [commentOwner, setCommentOwner] = useState(null);  const {getUser} = useAuth();   useEffect(() => {  async function fetchOwner() {  setCommentOwner(await getUser(comment.user\_id))  }  fetchOwner();  }, []);   return (  <div className="tweet\_\_comment">  <div className="tweet\_\_comment-avatar">  <img src={commentOwner && commentOwner.avatar} alt=""/>  </div>  <div className="tweet\_\_comment-inner">  <div className="tweet\_\_comment-title">  <div className="tweet\_\_comment-title\_user"><a href={`users/${commentOwner ? commentOwner.id : 1}`}>{commentOwner && commentOwner.nickname}</a></div>  <div className="tweet\_\_comment-title\_date"><span>{comment && comment.created\_at}</span></div>  </div>  <div className="tweet\_\_comment-text">  {comment.content}  </div>  </div>  </div>  ); } |

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);  }  } } |

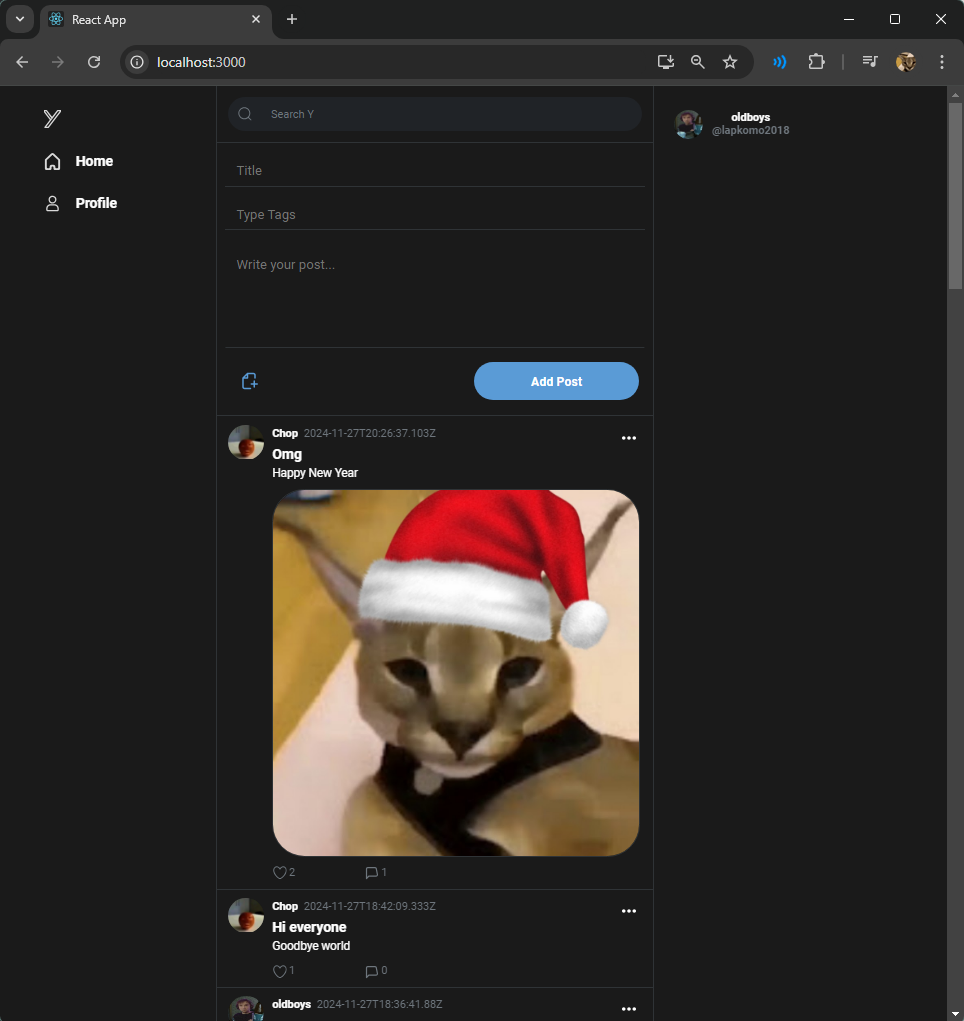


Рисунок 5 — Рисунок головної сторінки

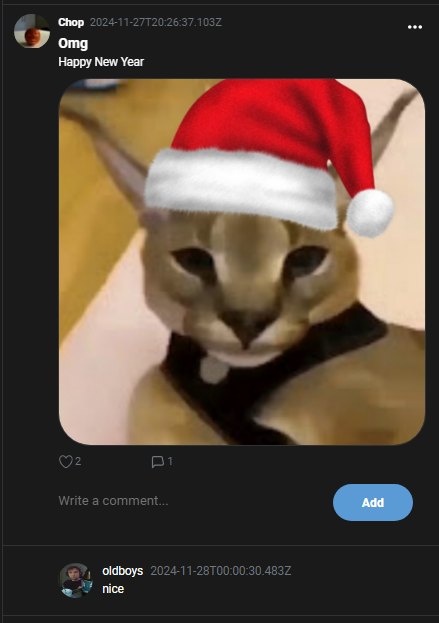


Рисунок 6 — Рисунок поста з можливістю лайкати та коментувати пост