

УДК 004.455:004.932:316.472.4

DOI: 10.31673/2412-9070.2024.051248

І. Р. АНДРІЙЧУК¹, студент;

ORCID: 0009-0001-1108-636X

О. В. НЕГОДЕНКО¹, канд. техн. наук, доцент;

ORCID: 0000-0001-6645-1566

С. М. ШЕВЧЕНКО², канд. педагогіч. наук, доцент,

ORCID: 0000-0002-9736-8623

¹Державний університет інформаційно-комунікаційних технологій, Київ

²Київський столичний університет імені Бориса Грінченка

МІНІМІЗАЦІЯ НЕГАТИВНИХ ВПЛИВІВ ВІД «ІНФОРМАЦІЙНОЇ БУЛЬБАШКИ» ПРИ РОЗРОБЦІ КЛІЄНТ-СЕРВЕРНОГО ЗАСТОСУНКУ СОЦІАЛЬНОЇ МЕРЕЖІ ДЛЯ РОЗПОВСЮДЖЕННЯ ФОТОГРАФІЙ З ВИКОРИСТАННЯМ ФРЕЙМВОРКІВ SPRING TA NEXTJS

Дана стаття присвячена проблемі розробки програмного забезпечення клієнт-серверного застосунку соціальної мережі, спрямованого на полегшення розповсюдження фотографій без використання алгоритмів оптимізації видачі для мінімізації ефекту «інформаційної бульбашки». Актуальність даного дослідження підтверджується стрімким зростанням кількості користувачів соціальних мереж по всьому світу та зростанням запиту на соціальні мережі з усвідомленим вибором контенту на тлі загострення проблеми «інформаційних бульбашок», створених алгоритмами фільтрації в існуючих додатках. На базі цього було вирішено підтримати неупередженість інформації та не використовувати персоналізовані та групові алгоритми фільтрації підбору контенту, запровадивши нефільтрований рандомний підбір контенту.

У дослідженні було використано сучасні технології для розробки програмного забезпечення, а саме: створено повнофункціональне, ізольоване та відтворюване середовище розробки на базі Visual Studio Code та контейнерів Docker, використано базу даних PostgreSQL як основну та Amazon Simple Storage Service (Amazon S3) для збереження фотографій, підібрано конфігурацію TypeScript + Next.js для стабільного, продуктивного та легко-читабельного фронтенду та обрано один з найпопулярніших та найнадійніших фреймворк мови програмування Java - Spring для написання бекенду.

Даний додаток доступний з браузера та створює зручний та простий механізм для обміну фотографіями у вигляді постів, підтримує профілі користувачів та логіку додавання друзів. Він також підтримує алгоритми підбору контенту по друзям та абсолютно рандомного контенту. Додаток захищений від CSRF атак за допомогою JWT.

Ключові слова: соціальна мережа, розповсюдження фотографій, інформаційна бульбашка, алгоритм фільтрації, технологія, розробка.

Вступ

Постановка проблеми. Сучасний світ характеризується інтенсивним прогресом у галузі інформаційних технологій, який супроводжується швидким розвитком соціальних мереж. Протягом останнього десятиліття соціальні мережі стали необхідною частиною повсякденного життя, відіграючи важливу роль у забезпеченні комунікації та формуванні віртуальних спільнот, даючи змогу відстежувати останні локальні та глобальні події, залишатися в потоці останніх подій та трендів.

Соціальні мережі є однією з найпопулярніших цифрових видів діяльності, чис поширення постійно зростає. Станом на квітень 2024 року глобальний рівень використання соціальних мереж становив 62.2%, що означає, що майже шість із кожних десяти людей у всьому світі ви-

© І. Р. Андрійчук, О. В. Негоденко, С. М. Шевченко, 2024

користують соціальні мережі [1]. Прогнозується, що ці показники зростатимуть, оскільки використання соціальних мереж, як і мобільних пристроїв загалом, набуває все більшої популярності на раніше неохоплених ринках: у 2023 році понад 4,89 мільярда осіб в усьому світі активно використовували соціальні мережі. Згідно з прогнозами кількість користувачів соціальних мереж досягне шести мільярдів до 2027 року [2].

У контексті розповсюдження соціальних мереж поширення фотографій є одним з найпопулярніших та найзручніших способів вираження та спілкування, що створює необхідність у постійному удосконаленні засобів для їх ефективного та зручного обміну. Однак, наявні соціальні мережі не завжди можуть задовольнити вимоги та потреби своїх користувачів. Недостатній функціонал або проблеми з безпекою можуть ускладнити використання наявних платформ та відвернути потенційних користувачів. Так, у багатьох сучасних соціальних мереж дуже гостро стає проблема "інформаційної бульбашки", коли користувач отримує обмежений обсяг інформації, адже система показує лише контент, відповідний до його попередніх взаємодій. Це може призводити до обмеженого сприйняття різноманітності поглядів і джерел інформації, а також до зростання поляризації і посилення впливу дезінформації, утворюючи так звану "інформаційну ехо-камеру".

Аналіз останніх досліджень і публікацій. Основою будь якої соціальної мережі є контент та спосіб його подання. Це те, від чого на пряму залежить зацікавленість користувачів у використанні конкретної платформи та її конкурентоспроможність. І хоча завдання створення контенту будь-якої соціальної мережі лежить на його користувачах, за підбір, стратегії та правила його показу несе відповідальність сама платформа.

У статті [3] було проведено дослідження п'яти алгоритмічних стратегій фільтрування та проведено експерименти над кожним із них. Кожна модель динаміки думок зосереджувалася на тому, що особа взаємодіє з фактичною думкою свого сусіда. Було виявлено, що це призводить до загального підвищення рівня полярності, особливо в мережах з екстремістами. Крім того, на ефективність стратегій фільтрації впливає наявність групового механізму. Було доведено особливості афективної поляризації — почуття до своєї групи та поза групою — у модель динаміки думок.

У наступній статті [4] було розглянуто проблему "інформаційної бульбашки" через призму демократії. На думку авторів, бульбашки фільтрів є проблемою не через перешкоду у доступі користувачам до бажаної інформації, а через зниження якості публічного обговорення. Демократія передбачає, що різноманітні погляди мають бути відкриті для користувачів, щоб вони могли виявити розбіжності, істини, перспективи та, зрештою, приймати найкращі рішення. Поляризовані користувачі або користувачі, які отримують низькоякісну, але приємну та актуальну інформацію, матимуть негативні наслідки та викривлене сприйняття. Для підвищення якості інформації, широкий спектр думок і поглядів потрібно зробити більш видимими. У даному дослідженні також було підібрано різні інструменти для боротьби з «бульбашками» фільтрів. Однак, оскільки дизайнери дотримуються різних цінностей і припускають різні моделі демократії прямо чи неявно, інструменти, які вони розробляють, відображатимуть ці цінності та моделі демократії. Технологія не є нейтральною і цінності та упередження, яких дотримуються дизайнери, проявляться у кінцевому продукті.

У статті [5] зображено вплив бульбашки фільтра на здоров'я, що є складною та проблемною сферою, адже від цього напряму залежить життя людей. Варіантом вирішення даної проблеми було запропоновано забезпечення можливості нефільтрованого пошуку. Це дозволить громадськості отримувати неупереджену інформацію на основі релевантності та якості вмісту, що є одним із шляхів вперед.

Проаналізувавши останні дослідження і публікації, було виявлено тенденцію та чіткий запит користувачів на соціальні мережі з усвідомленим вибором контенту. На базі цього було вирішено підтримати неупередженість до підбору інформації та вирішено не використовувати персоналізовані та групові алгоритми фільтрації підбору контенту та запровадити нефільтрований рандомний підбір контенту. Вище викладене підтверджує актуальність даного дослідження.

Мета статті. Представити розробку клієнт-серверного застосунку соціальної мережі, спрямованого на полегшення розповсюдження фотографій без використання алгоритмів оптимізації видачі для мінімізації ефекту “інформаційної бульбашки”. Створити рішення, яке не лише надає зручний механізм для обміну фотографіями, але й враховує сучасні вимоги до безпеки, швидкодії та забезпечення позитивного користувацького досвіду.

Основна частина

Для реалізації серверної частини застосунку було вирішено використати мову програмування Java. Java була створена понад 25 років тому, і вона досі залишається однією з найпопулярніших мов програмування. За даними PYPL у березні 2024 року Java була другою за популярністю мовою у світі. Цей рейтинг визначається шляхом аналізу запитів на мовні посібники в Google [6]. Java також четверта по індексу TIOBE, який базується на кількості кваліфікованих інженерів у всьому світі [7]. Однією з переваг при виборі мови стала наявність детальної документації, великої кількості ресурсів та її часте використання для написання сучасних бекенд рішень.

Для спрощення процесу написання бекенду було обрано один з найбільш надійних і універсальних серверних фреймворків Spring. Він високо цінується для розробки додатків Java корпоративного рівня через його надійність, модульність, впровадження залежностей, розширюваність і легку інтеграцію з іншими фреймворками. Spring, як легка і модульна платформа, заохочує модульність програми та робить конфігурацію гнучкою за допомогою XML, анотацій Java або конфігурації на основі Java. Одна з найвидатніших переваг Spring є його висока масштабованість, яка досягається завдяки побічним проектам, які він постачає, що робить його одним з найкращих фреймворком для бекенд-розробки [8].

Для збереження даних профілю користувача було використано базу даних PostgreSQL та Amazon Simple Storage Service (Amazon S3) для збереження фотографій.

PostgreSQL — це потужна об'єктно-реляційна база даних з відкритим вихідним кодом, активна розробка якої триває понад 35 років, завдяки чому вона заслужила міцну репутацію надійності, функціональності та продуктивності [9]. Вона забезпечує широкі можливості для зберігання та обробки даних, включаючи підтримку складних структур даних, транзакцій, індексів та різноманітні інструменти для адміністрування та оптимізації баз даних. Вона широко використовується у багатьох галузях, включаючи веб-розробку, аналітику даних, бізнес-застосунках, завдяки своїй надійності, масштабованості та гнучкості.

Для зберігання статичних файлів, таких як зображення, AWS S3 є одним із найкращих варіантів. S3 є одним із найдешевших хмарних сховищ, адже ціна S3 базується на використанні, що робить його економічно ефективним рішенням для розміщення зображень. Завдяки моделі ціноутворення S3 з оплатою за використання потрібно сплачувати лише за обсяг пам'яті та пропускну здатність, яка використовується, без жодних попередніх комісій чи довгострокових зобов'язань [10]. Використання вже наявних та налаштованих серверів Amazon для збереження великих файлів - фотографій, допоможе скоротити час виходу на ринок і легко масштабувати додаток у майбутньому.

Для написання клієнт-частини було обрано мову програмування TypeScript. TypeScript було створено для подолання обмежень у мові динамічної типізації JavaScript. Хоча код JavaScript може бути більш гнучким, відсутність статичної типізації може призвести до великої кількості помилок. TypeScript допомагає пом'якшити такі проблеми додаванням статичних типів. Статична перевірка типів TypeScript на етапі розробки виявляє потенційні помилки на ранніх стадіях, мінімізуючи конфлікти типів і типові проблеми. Проактивне виявлення помилок сприяє стабільнішим і надійнішим додаткам, що сприяє покращенню взаємодії з користувачем. Система типів TypeScript допомагає визначати та застосовувати типи в міру розширення кодової бази, що сприяє довгостроковій підтримці та масштабованості. Це покращує супроводжуваність та читабельність коду, підвищуючи тим самим продуктивність розробників і полегшуючи адаптацію нових розробників у майбутньому [11].

Також для реалізації фронтенду було використано бібліотеку з відкритим вихідним кодом React та його фреймворк Next.js. React використовується для створення користувацьких інтерфейсів, а Next.js для розширення їх функціональності та покращення продуктивності.

React є найпопулярнішою бібліотекою JavaScript за даними JetBrains за 2023 рік [12] і зайняв друге місце за популярністю у рейтингу веб-фреймворки серед розробників у всьому світі за 2023 рік [13]. React — це провідна технологія розробки надійних і динамічних інтерфейсів завдяки своїй віртуальній DOM і багаторазовим компонентам, які полегшують обробку даних у реальному часі. Обробка великого обсягу даних користувачів і одночасне оновлення програми є однією з головних технічних проблем кожного веб-додатку. React JS містить вбудовану функціональність віртуального DOM, що робить рендеринг швидшим і покращує загальну взаємодію з користувачем. Крім того, це робить програму легкою, що сприяє підвищенню продуктивності. Також він дає змогу розробникам створювати передові програми без урахування переходів між станами, що скорочує терміни розробки, а деревоподібне представлення реального DOM у пам'яті Virtual DOM значно прискорює процес між функціями виклику та відображення.

Next.js — потужний фреймворк React, який використовується для розробки багатофункціональних масштабованих веб-додатків. Він забезпечує потужне поєднання можливостей візуалізації на стороні сервера (SSR) і статичного створення сайтів (SSG), що робить його чудовим вибором для створення динамічних високопродуктивних веб-сайтів. Однією з головних переваг Next.js є його здатність покращувати продуктивність веб-сайту. Завдяки рендерингу на стороні сервера та генерації статичних сайтів Next.js мінімізує час, необхідний для завантаження веб-сторінки та може обробляти великі обсяги трафіку без шкоди для продуктивності. Також механізм автоматичного розділення коду впроваджує оптимізацію за рахунок завантаження лише необхідного JavaScript коду для кожної сторінки. Завдяки вбудованій підтримці компонентів React та маршрутизації використання Next.js спрощує процес створення та управління інтерфейсів сайту.

Для зовнішнього вигляду користувацького інтерфейсу програми було обрано Tailwind CSS. Tailwind CSS — це фреймворк CSS для створення інтерфейсів, що базується на наборі готових класів CSS. Замість написання власного CSS, розробники можуть використовувати вже готові класи для відображення різних стилів, таких як розміри, кольори, відступи. Це дозволяє прискорити розробку і зменшити кількість коду, не втрачаючи при цьому гнучкості.

Середовищем для розробки було обрано Visual Studio Code (VS Code). Visual Studio Code — це легкий, але потужний текстовий редактор, розроблений компанією Microsoft. Він надає розширені можливості для редагування коду та роботи з проектами, включаючи підтримку різних мов програмування, автоматичне доповнення коду, інтеграцію з системами керування версіями, вбудовану підтримку відладки та широкий вибір розширень для розширення функціональності за потреби користувача [14]. Він безкоштовний та ідеально підходить для розробки як на Spring так і на Next.js, створюючи можливість зручно працювати в одному редакторі відразу з двома технологіями. Більше того, VSCode має надзвичайно потужне та зручне розширення для роботи з Docker, що дозволяє використовувати контейнер Docker як повнофункціональне середовище розробки.

Docker — це платформа контейнеризації, яка дозволяє розробникам упаковувати програми та їхні залежності в самодостатні ізольовані блоки, які називаються контейнерами. Контейнери ізольовані один від одного та базової хост-системи, що робить їх простими та портативними для розгортання. Docker забезпечує простий і послідовний спосіб створення, розповсюдження та запуску програм, що допомагає оптимізувати процес розробки та скоротити час і зусилля, необхідні для налаштування та керування складною архітектурою програми та забезпечує стабільну роботу програмного забезпечення. Docker було використано при написанні даної роботи для ізольованості та незмінності середовища розробки для впевненості що клієнт, сервер та база даних є абсолютно автономними, а налаштування однієї програми ніяким чином

не впливають на інші. Також швидкість розгортання, зручність та відтворюваність середовища є не менш важливими, адже це на пряму впливає на стабільність релізних інстансів серверу.

На рис. 1 зображено діаграму варіантів використання (діаграму прецедентів) для кращого розуміння загального функціоналу.

Звідси видно, що при реєстрації користувача буде запит для підтвердження електронної пошти. У загальному підтримується логування та розлогування користувача, підтримуються його відношення з іншими користувачами, пошук по користувачам та контенту, перегляд контенту, створення та редагування постів та редагування профілю. Особливу увагу було приділено авторизації користувача, так як соціальні мережі доволі часто стають жертвами хакерів та махінацій у цілому. На рис. 2 зображено один із найпростіших способів реалізації авторизації через збереження індивідуального номеру сесії.

Проте, така модель абсолютно беззахисна перед міжсайтовою підробкою запитів (Cross-Site Request Forgery, CSRF). CSRF - це типова атака на веб-додатки, також відома як атака одним клацанням миші, яка використовується для виконання несанкціонованих дій під ім'я аутентифікованого користувача. Тому, було реалізовано авторизацію через JWT (JSON Web Tokens) для захисту від подібного роду атак. Деталі реалізації зображено на рис. 3.

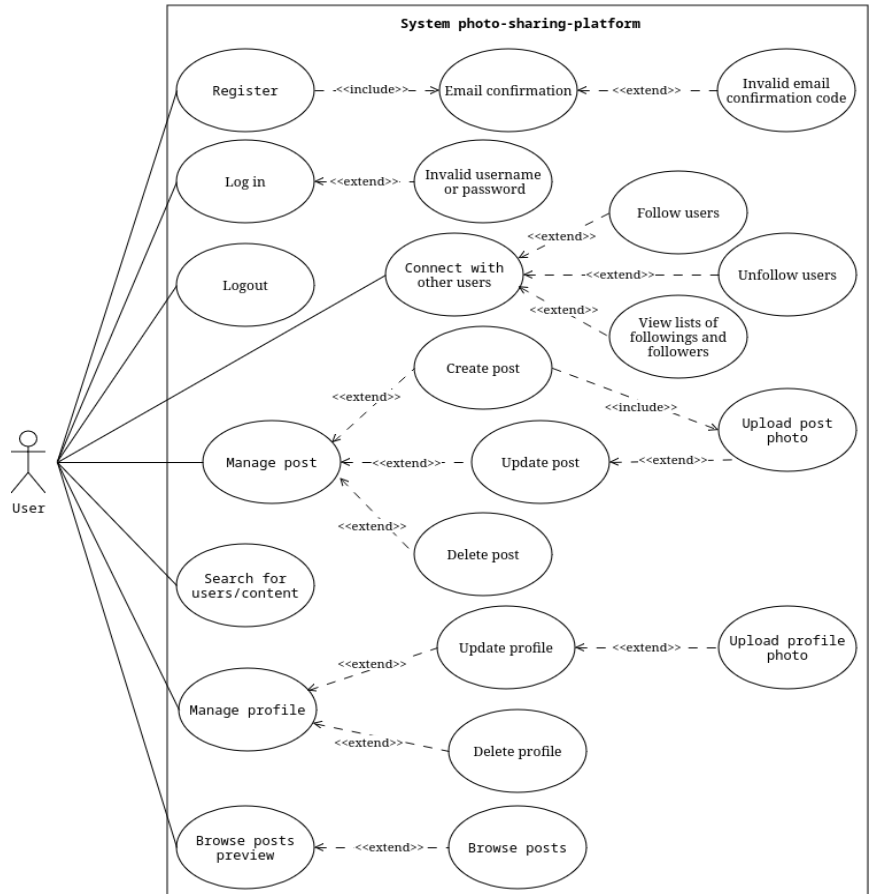


Рис. 1. Діаграма варіантів використання

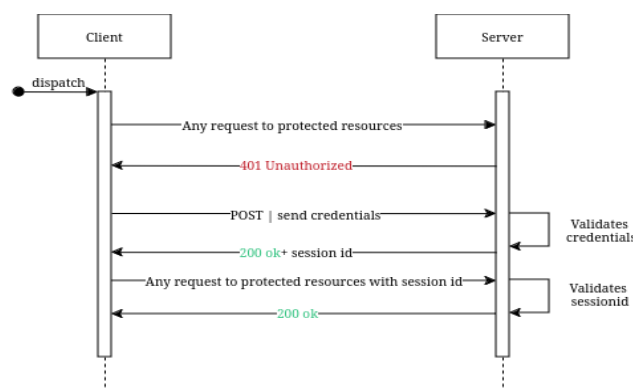


Рис. 2. UML діаграма послідовності простої авторизації

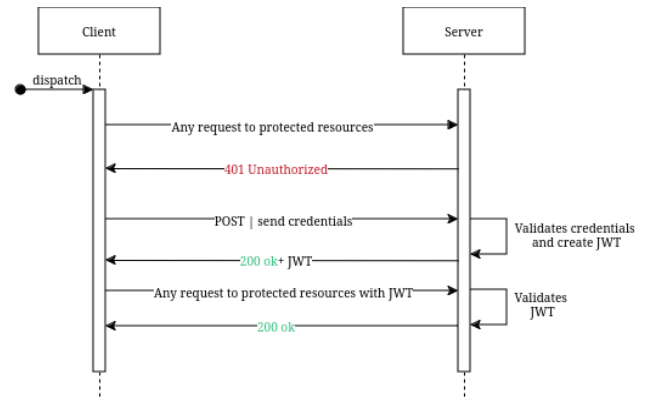


Рис. 3. UML діаграма послідовності авторизації з використанням JWT

Висновки

Результатом виконання даного дослідження є застосунок, який спрямований на спрощення розповсюдження та обміну фотографій без використання алгоритмів оптимізації видачі контенту з метою мінімізації ефекту “інформаційної бульбашки”. Цей застосунок враховує сучас-

сні вимоги до швидкодії та користувацького досвіду і створює легкий інтерфейс для обміну фотографіями, підтримує профілі користувачів та логіку додавання друзів. Він також підтримує алгоритми підбору контенту по друзям та абсолютно випадкового контенту та захищений від CSRF атак за допомогою JWT.

У перспективі дана система буде розроблена як мобільний застосунок також. Планується розробка більш сучасного та продуманого дизайну застосунку та розширення його функціоналу додаванням можливості коментування та додатковими варіантами фільтрації контенту за бажанням користувача; винести логіку, пов'язану з авторизацією, в окрему програму – сервер авторизації та підкорегувати логіку для відповідності стандарту OAuth 2.0; у фронтенд частині додати реєстр сесій у локальному сховищі для покращення захищеності користувача та додати інтерфейс для обрізання фотографій.

Список літератури

1. *Global social network penetration rate as of April 2024, by region* - *statista.com* – [Електронний ресурс] – Режим доступу: <https://www.statista.com/statistics/269615/social-network-penetration-by-region/>
2. *Number of social media users worldwide from 2017 to 2027* - *statista.com* – [Електронний ресурс] – Режим доступу: <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>
3. *Algorithmic Filtering, Out-Group Stereotype, and Polarization on Social Media* - *yeoh-lab.wustl.edu* – [Електронний ресурс] – Режим доступу: <https://yeoh-lab.wustl.edu/assets/pdf/aamas-SpringsteenYC24.pdf>
4. *Breaking the filter bubble: democracy and design* - *springer.com* – [Електронний ресурс] – Режим доступу: <https://link.springer.com/article/10.1007/s10676-015-9380-y>
5. *PYPL PopularitY of Programming Language* - *pypl.github.io* – [Електронний ресурс] – Режим доступу: <https://pypl.github.io/PYPL.html>
6. *TIOBE Index for May 2024* - *www.tiobe.com* – [Електронний ресурс] – Режим доступу: <https://www.tiobe.com/tiobe-index/>
7. *Spring Framework* - *spring.io* – [Електронний ресурс] – Режим доступу: <https://spring.io/projects/spring-framework>
8. *PostgreSQL* - *www.postgresql.org* – [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/>
9. *Amazon S3* - *aws.amazon.com* – [Електронний ресурс] – Режим доступу: <https://aws.amazon.com/s3/>
10. *TypeScript for the New Programmer*- *www.typescriptlang.org* – [Електронний ресурс] – Режим доступу: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>
11. *JavaScript* - *jetbrains.com* – [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/lp/devecosystem-2023/javascript/>
12. *Most used web frameworks among developers worldwide, as of 2023* - *www.statista.com* – [Електронний ресурс] – Режим доступу: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>
13. *Why did we build Visual Studio Code?*- *code.visualstudio.com* – [Електронний ресурс] – Режим доступу: <https://code.visualstudio.com/docs/editor/whyvscode>

I. Andriichuk, O. Nehodenko, S. Shevchenko

MINIMIZING THE NEGATIVE EFFECTS OF THE INFORMATION BUBBLE IN THE DEVELOPMENT OF A SOCIAL NETWORK CLIENT-SERVER APPLICATION FOR PHOTO DISTRIBUTION USING THE SPRING AND NEXTJS FRAMEWORK

This article is devoted to the problem of software development of a client-server application of a social network aimed at facilitating the distribution of photos without the use of optimization algorithms to minimize the information bubble effect. The relevance of this study is confirmed by the rapid

growth of the number of users of social networks around the world and the growing demand for social networks with a conscious choice of content against the background of the aggravation of the problem of information bubbles created by filtering algorithms in existing applications. Based on this, it was decided to support the impartiality of information and not to use personalized and group filtering algorithms for content selection, introducing unfiltered random selection of content.

The research used modern technologies for software development, namely: a fully functional, isolated and reproducible development environment based on Visual Studio Code and Docker containers was created, a PostgreSQL database was used as the main database and Amazon Simple Storage Service (Amazon S3) was used to store photos, the configuration of TypeScript + Next.js was selected for a stable, productive and easy-to-read frontend, and one of the most popular and reliable frameworks of the Java programming language - Spring was chosen for writing the backend.

This application is available from the browser and creates a convenient and simple mechanism for sharing photos in the form of posts, supports user profiles and the logic of adding friends. It also supports algorithms for selecting content from friends and completely random content. The application is protected against CSRF attacks using JWT.

Keywords: social network, photo sharing, information bubble, filtering algorithm, technology, development.
