

22. The Apache Iceberg Open Table Format. URL: <https://www.dremio.com/resources/guides/apache-iceberg/> (дата звернення: 28.11.2023).
23. Parquet. URL: <https://www.databricks.com/glossary/what-is-parquet> (дата звернення: 02.09.2023).
24. ETL Process in Data Warehouse. URL: <https://www.geeksforgeeks.org/etl-process-in-data-warehouse/> (дата звернення: 10.10.2023).

Стаття надійшла 12.04.2024

Стаття прийнята 01.05.2024

УДК 004.9:658.818

doi: 10.31498/2225-6733.48.2024.310670

© Кіріакіді С.К.¹, Кривенко О.В.²

ДОСЛІДЖЕННЯ МЕТОДОЛОГІЙ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ МЕДИЧНОЇ КЛІНІКИ З ВИКОРИСТАННЯМ REACT

У статті детально досліджуються методології розробки вебдодатків для медичних клінік з використанням популярної бібліотеки React. Основною метою цього дослідження є аналіз сучасних підходів до створення вебдодатків, які відповідають специфічним вимогам медичних установ. Особлива увага приділяється функціональним та нефункціональним характеристикам, що є критично важливими для таких систем, зокрема безпеці даних, користувацькому досвіду, інтеграції з існуючими системами та відповідності нормативним стандартам. Перш за все, проведено всебічний аналіз вимог медичних клінік до вебдодатків. Безпека даних є одним з найважливіших аспектів, оскільки медичні установи працюють з конфіденційною інформацією. У дослідженні розглянуто методи забезпечення конфіденційності та цілісності медичних даних, зокрема шляхом дотримання стандартів HIPAA та GDPR. React дозволяє легко інтегрувати сучасні засоби безпеки, такі як JWT (JSON Web Tokens) для аутентифікації та авторизації, а також використовувати захищені HTTP-з'єднання для передачі даних. Крім того, особлива увага приділена користувацькому досвіду (UX/UI), який має забезпечувати зручність та ефективність використання вебдодатку як медичним персоналом, так і пацієнтами. React дозволяє створювати динамічні та інтерактивні інтерфейси, що можуть адаптуватися до різних пристроїв та екранів, забезпечуючи таким чином високу якість користувацького досвіду. Це особливо важливо для медичних додатків, де точність і швидкість взаємодії можуть впливати на якість надання медичних послуг. Наведено детальний опис найкращих практик та підходів до розробки вебдодатку з використанням React. Розглянуто архітектуру компонентів, яка забезпечує модульність та повторне використання коду, що є ключовим фактором у створенні масштабованих та легко підтримуваних додатків. Досліджено різні методи керування станом у додатку, такі як використання Context API та Redux, що дозволяє ефективно керувати даними та забезпечувати стабільну роботу додатку. Описано рекомендації щодо налаштування маршрутизації за допомогою бібліотеки React Router, що дозволяє створювати динамічні та багатосторінкові додатки, забезпечуючи при цьому плавну навігацію та хорошиший користувацький досвід. Крім того, розглянуто способи забезпечення безпеки додатку, включаючи захист від XSS-атак, CSRF та інших типів вразливостей, що є критично важливим у медичних додатках.

¹ бакалавр, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, kiriakidi.softya60@gmail.com

² канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: 0009-0006-2860-6575, krivenkoov@gmail.com

У статті надано конкретні рекомендації щодо реалізації вебдодатку для медичної клініки. Вони включають практичні приклади коду, що демонструють реалізацію ключових функціональностей, таких як реєстрація та автентифікація користувачів, обробка медичних записів та взаємодія з базами даних. Описано процес налаштування інфраструктури для розробки, тестування та розгортання додатку, включаючи використання CI/CD інструментів для автоматизації цих процесів, що сприяє підвищенню ефективності розробки та якості кінцевого продукту. Стаття надає цінну інформацію для розробників, які прагнуть створити ефективні та безпечні вебдодатки для медичних клінік, використовуючи сучасні технології. Висновки та рекомендації, представлені у дослідженні, допоможуть забезпечити високу якість та відповідність розроблених рішень сучасним стандартам та вимогам галузі, що сприятиме підвищенню ефективності роботи медичних установ та якості медичних послуг, що надаються пацієнтам.

Ключові слова: розробка вебдодатків, медична клініка, React, архітектура компонентів, керування станом, маршрутизація, безпека даних.

S.K. Kiriakidi, O.V. Kryvenko. Study of web application development methodology for a medical clinic using React. The article explores in detail the methodologies for developing web applications for medical clinics using the popular React library. The main goal of this study is the analysis of modern approaches to the creation of web applications that meet the specific requirements of medical institutions. Particular attention is paid to the functional and non-functional characteristics that are critical to such systems, including data security, user experience, integration with existing systems, and compliance with regulatory standards. First of all, a comprehensive analysis of the requirements of medical clinics for web applications was carried out. Data security is one of the most important aspects as healthcare institutions work with sensitive information. The study examines methods for ensuring the confidentiality and integrity of medical data, including compliance with HIPAA and GDPR standards. React makes it easy to integrate modern security tools such as JWT (JSON Web Tokens) for authentication and authorization, and use secure HTTP connections for data transfer. In addition, special attention is paid to the user experience (UX/UI), which should ensure the convenience and efficiency of using the web application by both medical personnel and patients. React allows you to create dynamic and interactive interfaces that can adapt to different devices and screens, thus providing a high-quality user experience. This is especially important for medical applications, where the accuracy and speed of interaction can affect the quality of healthcare delivery. A detailed description of best practices and approaches to web application development using React is provided. A component architecture is considered, which provides modularity and code reuse, which is a key factor in creating scalable and easily maintainable applications. Different methods of state management in the application, such as the use of Context API and Redux, have been explored, which allows for efficient data management and stable operation of the application. Guidelines for configuring routing using the React Router library are described, allowing you to create dynamic and multi-page applications while providing smooth navigation and a good user experience. In addition, ways to ensure application security are considered, including protection against XSS attacks, CSRF and other types of vulnerabilities, which is critical in medical applications. The article provides specific recommendations for implementing a web application for a medical clinic. They include practical code examples that demonstrate the implementation of key functionalities such as user registration and authentication, medical record processing, and database interaction. The process of setting up the infrastructure for development, testing and deployment of the application is described, including the use of CI/CD tools to automate these processes, which contributes to increasing the efficiency of development and the quality of the final product. The article provides valuable information for developers who seek to create effective and secure web applications for medical clinics using modern technologies. The conclusions and recommendations presented in the study will help ensure high quality and compliance of the developed solutions with modern industry standards and requirements,

which will contribute to increasing the efficiency of the work of medical institutions and the quality of medical services provided to patients.

Key words: *web application development, medical clinic, React, component architecture, state management, routing, data security.*

Постановка проблеми. Сучасна медична клініка потребує ефективного та зручного веб-додатка для оптимізації своїх процесів і поліпшення взаємодії з пацієнтами. Вебдодаток, що дозволяє пацієнтам записуватися на прийом, переглядати результати аналізів та рекомендації лікаря, стає необхідністю для забезпечення високоякісного медичного обслуговування. Крім того, лікарі також повинні мати доступ до додатку для управління прийомами та перегляду медичних даних пацієнтів.

Основна проблема полягає в забезпеченні безпеки та конфіденційності медичних даних. Вебдодаток повинен відповідати найвищим стандартам захисту інформації, щоб запобігти несанкціонованому доступу до чутливої інформації. Важливим є також створення інтуїтивно зрозумілого інтерфейсу, щоб пацієнти різного віку могли легко користуватися додатком, а лікарі могли швидко та ефективно виконувати свої завдання.

Інтеграція з існуючими медичними системами клініки є ще однією важливою складовою. Вебдодаток повинен бути сумісним з наявними електронними медичними записами та іншими інструментами, які вже використовуються в клініці, для забезпечення безперебійної роботи.

Масштабованість та продуктивність вебдодатку також відіграють ключову роль. Зростання кількості користувачів і даних не повинно впливати на швидкість та ефективність роботи системи.

Використання React як основної технології для розробки вебдодатку може допомогти вирішити проблеми завдяки його гнучкості, високій продуктивності та широкому набору інструментів, які забезпечують створення надійних та масштабованих рішень.

Аналіз останніх досліджень і публікацій. У сфері розробки вебдодатків для медичних клінік використовуються різноманітні технології. Останні дослідження та публікації свідчать про популярність таких бібліотек і фреймворків, як React, Angular, Vue.js, та Svelte. Кожна з технологій має свої переваги та недоліки, що робить вибір оптимального рішення важливим етапом у процесі розробки (Табл. 1).

Таблиця 1

Аналіз сучасних технологій

Технологія	Переваги	Недоліки
React	висока продуктивність; гнучкість та масштабованість; велика спільнота підтримки; велика кількість готових компонентів	вимагає більше налаштувань; розробка може бути складною для новачків
Angular	повний фреймворк; вбудовані рішення для багатьох задач; висока продуктивність	високий поріг входження; складність коду
Vue.js	легкість у вивченні; гнучкість; проста інтеграція	менша спільнота порівняно з React; обмежена кількість великих проєктів
Svelte	висока продуктивність; проста структура; компіляція в чистий JavaScript	відносно нова технологія; менша спільнота та підтримка

React є однією з найпопулярніших бібліотек для розробки вебдодатків завдяки своїм унікальним властивостям, що роблять її особливо підходящою для створення складних та масштабованих додатків, таких як медичні системи. Однією з ключових переваг React є її компонентна архітектура, яка дозволяє розробникам створювати повторно використовувані та ізольовані компоненти. Це особливо важливо у контексті медичних додатків, де модульність та надійність є критично важливими. Компоненти React можна легко тестувати та оновлювати, що сприяє

зниженню ймовірності помилок та підвищенню загальної якості коду.

React використовує віртуальний DOM, який забезпечує швидке оновлення інтерфейсу без необхідності повного перезавантаження сторінки. Це особливо важливо для вебдодатків, які мають багато інтерактивних елементів і динамічних даних.

Також React дозволяє створювати компоненти, які можна повторно використовувати в різних частинах додатку, що спрощує розробку, тестування та підтримку коду, особливо в масштабних проєктах.

Ще React має велику спільноту розробників і багатий набір готових рішень і бібліотек, що забезпечує швидке вирішення проблем і доступ до широкого спектра додаткових інструментів і ресурсів.

React легко інтегрується з іншими сучасними технологіями та інструментами, такими як Redux для керування станом, React Router для маршрутизації та багато інших. Це робить його гнучким інструментом для вирішення різноманітних завдань.

Вибір React для розробки вебдодатку для медичної клініки дозволяє скористатися перевагами цієї технології, забезпечуючи високу продуктивність, гнучкість та масштабованість, що є ключовими вимогами для успішної реалізації проєкту [1].

Метою статті є дослідження та аналіз методологій розробки вебдодатку для медичної клініки з використанням React. У роботі розглянуто, як використання цієї бібліотеки може оптимізувати процес створення вебдодатку, забезпечити його високу продуктивність, гнучкість та відповідність вимогам сучасної медицини.

Основні завдання роботи включають:

- визначення основних функціональних та нефункціональних вимог, таких як безпека даних, користувацький досвід, інтеграція з існуючими системами та відповідність нормативним стандартам;
- порівняння різних фреймворків та бібліотек, таких як React, Angular, Vue.js та Svelte, з метою визначення найбільш підходящої технології для розробки медичних вебдодатків;
- вивчення найкращих практик та підходів до розробки з використанням React, включаючи архітектуру компонентів, керування станом, маршрутизацію та безпеку;
- надання конкретних рекомендацій щодо реалізації вебдодатку для медичної клініки з використанням React.

Робота мала на меті не лише теоретичний аналіз, але й практичне застосування отриманих результатів, що дозволить створити ефективний, безпечний і зручний вебдодаток для медичних клінік, який відповідатиме всім сучасним вимогам та стандартам.

Виклад основного матеріалу. Розробка вебдодатка для медичної клініки потребує ретельного аналізу як функціональних, так і нефункціональних вимог, щоб забезпечити ефективність, безпеку та зручність використання.

Функціональні вимоги:

- запис на прийом. Вебдодаток повинен дозволяти пацієнтам легко записуватися на прийом до лікаря, обирати зручний час і дату, а також отримувати підтвердження запису. Лікарі повинні мати можливість переглядати та керувати своїм розкладом прийомів;
- перегляд результатів аналізів. Пацієнти повинні мати доступ до своїх медичних результатів, включаючи аналізи та рекомендації лікарів. Це забезпечує прозорість та доступність медичної інформації.

Нефункціональні вимоги:

- безпека даних. Захист чутливих медичних даних є ключовим аспектом. Вебдодаток повинен відповідати стандартам безпеки, таким як шифрування даних, багатофакторна аутентифікація та регулярні аудити безпеки. Це важливо для запобігання несанкціонованому доступу до медичних записів і захисту конфіденційності пацієнтів;
- інтерфейс повинен бути інтуїтивно зрозумілим і зручним для використання. Це включає просту навігацію, зрозумілий дизайн і адаптивність до різних пристроїв (мобільних телефонів, планшетів, комп'ютерів). Важливо забезпечити легкий доступ до всіх функцій, особливо для користувачів з обмеженими технічними навичками.

Аналіз цих вимог є критично важливим етапом у розробці вебдодатку для медичної клініки. Він дозволяє визначити ключові аспекти, які потрібно врахувати для створення

ефективного, безпечного та зручного продукту, що задовольняє потреби як пацієнтів, так і медичного персоналу [2].

React є однією з найпопулярніших бібліотек для створення динамічних і масштабованих вебдодатків. Для розробки вебдодатку для медичної клініки з використанням React важливо врахувати найкращі практики та підходи, які забезпечать ефективність, безпеку та зручність використання.

React побудований на концепції компонентів, які є незалежними, повторно використовуваними частинами інтерфейсу користувача.

Основними принципами архітектури компонентів є:

1) Поділ інтерфейсу на дрібні, ізольовані компоненти для полегшення управління кодом, тестування його та повторного використання в різних частинах додатку. Наприклад, компонент форми запису на прийом можна використовувати як у головному розділі, так і в особистому кабінеті пацієнта.

2) Поділ компонентів на інші компоненти для створення складних інтерфейсів з простих будівельних блоків, що сприяє зручному масштабуванню додатку та зменшує дублікацію коду.

Вагомим аспектом розробки складних вебдодатків є керування станом. Завдяки використанню React існують кілька підходів до керування станом:

1) Локальний стан компонентів, який використовується для керування станом тільки окремого компонента. Локальний стан компонентів зручно застосовувати для простих компонентів з обмеженою логікою.

2) Контекст API, який дозволяє передавати стан через дерево компонентів без необхідності передавати пропси на кожному рівні. Контекст API застосовується для стану, який потребує доступу у багатьох компонентах, наприклад, автентифікаційні дані користувача.

3) Redux (бібліотека для керування глобальним станом додатку), яка надає централізоване сховище для всіх даних. Redux забезпечує передбачуваність стану та полегшує дебагінг, особливо у великих додатках.

Бібліотека React Router у React використовується для створення багатосторінкових вебдодатків. Вона дозволяє створювати:

1) Маршрути. Визначати шляхи для різних сторінок додатку та зв'язувати їх з відповідними компонентами. Призначена для забезпечення інтуїтивної навігації користувачів.

2) Динамічні маршрути. Призначена для підтримки параметризованих маршрутів при відображенні динамічного контенту, наприклад, сторінок пацієнтів з унікальними ID.

3) Захищені маршрути. Призначена обмежувати доступ до певних сторінок тільки для авторизованих користувачів, що важливо для забезпечення безпеки медичних даних [3].

Дотримання цих методологій та практик забезпечує створення надійного, безпечного та масштабованого вебдодатку для медичної клініки, що відповідає всім сучасним вимогам та стандартам. Використання React у поєднанні з цими підходами дозволяє ефективно вирішувати завдання, що стоять перед розробниками, та забезпечити високоякісний продукт для користувачів.

При створенні веб-сайту було надано значного значення формуванню DOM структури та правильній розмітці HTML. Усі розділи сайту побудовані на React компонентах, що дозволяє ефективно управляти структурою та станом елементів (рис. 1).

В результаті реалізації вебдодатку для медичної клініки з використанням React отримали наступні рекомендації, які потрібно враховувати перед розробкою вебдодатка:

1. Дизайн з точки зору користувача: необхідно аналізувати потреби та зручність для користувачів, зосереджуючись на їхньому комфорті та ефективності використання.

2. Конфіденційність та безпека даних: необхідно враховувати найвищі стандарти безпеки для зберігання медичних даних, використовувати механізми шифрування та визначати точки доступу для забезпечення конфіденційності пацієнтів.

3. Масштабованість та розширюваність: передбачити можливість масштабування та розширення вебдодатку з урахуванням зростання клініки та зміни вимог користувачів.

4. Документація та коментування коду: забезпечувати належну документацію та коментування коду для полегшення розуміння та співпраці між членами команди розробників.

5. Тестування: ретельно тестувати вебдодаток для виявлення та усунення помилок і недоліків до випуску в продакшн.

6. Підтримка різних пристроїв і браузерів: необхідно врахувати коректну роботу

вебдодатку на різних пристроях та веб-переглядачах, забезпечуючи однаковий досвід користувача для всіх користувачів.

7. Відкритість для зворотного зв'язку: надати можливість користувачам залишати відгуки та пропозиції щодо поліпшення вебдодатку для його постійного вдосконалювання.

8. Резервне копіювання та відновлення даних: забезпечити систему резервного копіювання та відновлення даних для запобігання втрати важливої інформації.

9. Дотримання стандартів та правил законодавства: вебдодаток повинен відповідати всім вимогам і стандартам законодавства про медичну інформацію та захист даних.

10. Постійне оновлення та підтримка: забезпечити регулярне оновлення та підтримку вебдодатку, враховуючи нові технології та вимоги користувачів [4].

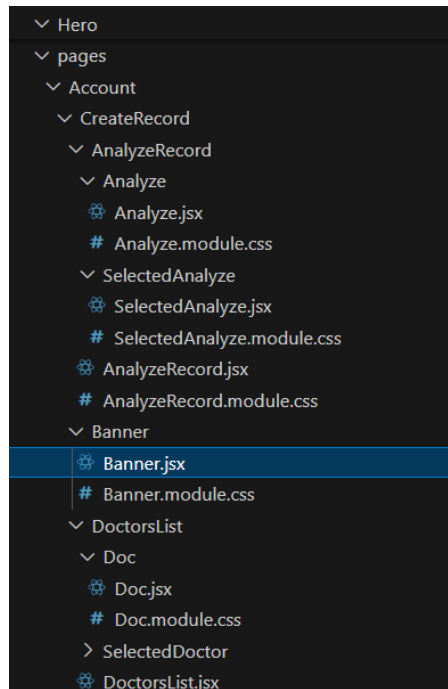


Рис. 1 – Архітектура компонентів

При реалізації вебдодатку для медичної клініки з використанням React з урахуванням наведених рекомендацій буде забезпечена його ефективність, безпека та зручність використання для користувачів.

Висновки

React, з його декларативним підходом до створення інтерфейсів користувача, пропонує ефективні засоби для реалізації складних медичних додатків. React сприяє покращенню швидкості та продуктивності розробки, дозволяючи розробникам швидко створювати динамічні та інтерактивні веб-інтерфейси.

Під час дослідження підкреслена увага архітектурі додатку на основі React, зокрема використанню компонентного підходу, керуванню станом та маршрутизації. Дані концепції дозволяють створювати гнучкі та масштабовані вебдодатки, що відповідають потребам медичних клінік.

Загалом, React є потужним інструментом для розробки вебдодатків для медичних клінік, і дана робота допоможе розробникам зрозуміти, як ефективно використовувати його можливості для досягнення своїх цілей у цій галузі.

Перелік використаних джерел:

1. Ключков Д. В. Загальні компоненти при кросплатформовій розробці для web- та мобільних додатків з використанням React та React-native. *Молодий вчений*. 2018. № 36(222). С. 1-5.

2. Вопросы по React. Версия 1. URL: <https://my-js.org/docs/other/react-questions/> (дата звернення: 19.05.2024).
3. Griffiths D., Griffiths D. React Cookbook: Recipes for Mastering the React Framework. 2021. 510 p.
4. Stefanov S. React: Up & Running: Building Web Applications. 2nd ed. 2021. 230 p.

References:

1. D.V. Klochkov, «Zahalni komponenty pry krosplatformovii rozrobtsi dlia web- ta mobilnykh dodatkov z vykorystanniam React ta React-native» [«Common components for cross-platform development for web and mobile applications using React and React-native»], *Molodyi vchenyi – Young scientist*, № 36 (222), pp. 1-5, 2018. (Ukr.)
2. Voprosi po React. Versyia 1 (React Questions. Version 1). [Online]. Available: <https://my-js.org/docs/other/react-questions/>. Accessed on: May 19, 2024.
3. D. Griffiths, and D. Griffiths, *React Cookbook: Recipes for Mastering the React Framework*. 2021.
4. S. Stoyan, *React: Up & Running: Building Web Applications*, 2nd Ed. 2021.

Стаття надійшла 13.04.2024

Стаття прийнята 10.05.2024