УДК 004.42

В. В. Федорова
Я. В. Іванчук

БІБЛІОТЕКА REACT ЯК ЕФЕКТИВНИЙ ІНСТРУМЕНТ РОЗРОБКИ WEB-ДОДАТКІВ

Вінницький національний технічний університет

Анотація

Проведено аналіз фреймворку React мови програмування JavaScript. Визначено основні переваги та недоліки вибору саме бібліотеки React. Представлено головні особливості використання фреймворку.

Ключові слова: фреймворк, мова програмування, розробка, React, JavaScript

Abstract

An analysis of the React framework of the JavaScript programming language was carried out. The main advantages and disadvantages of choosing the React library are defined. The main features of using the framework are presented.

Keywords: framework, programming language, development, React, JavaScript

Вступ

З розвитком технологій все більше почало з'являтися нових засобів для розробки додатків та веб-сайтів. На сьогоднішній день існує велика кількість JavaScript фреймворків і бібліотек (React, Angular, Svetle, Vue.js, та інші), які допомагають розробникам створювати різні WEB-додатки [1]. Більшість з даних інструментів засновані на різних евристиках, як наслідок цього, розробники постійно покращують свої рішення [2].

Сьогодні React – найпопулярніша бібліотека JavaScript у світі і популярність інструменту невпинно зростає [3]. Технологія дозволяє користувачеві взаємодіяти з інтерфейсом без перезавантаження сторінки – ставити лайки, заповнювати форми, дивитися геолокацію, оплачувати покупки онлайн, додавати товари в кошик, писати повідомлення тощо. Миттєвий відгук інтерфейсу на дії користувача сприяє значному поліпшенню поведінкових факторів користувача [2].

Метою дослідження є визначення основних переваг і недоліків використання бібліотеки React, що дозволить ефективно обирати бібліотеки прикладного програмування JavaScript для розробки WEB-додатків.

Аналіз дослідження

React JS – один із сучасних та ефективних фреймворків на основі мови JavaScript [3]. Основна сфера застосування – розробка користувацьких інтерфейсів. React використовують для створення: сайтів, односторінкових сервісів, мобільних додатків.

На відміну від звичайних веб-сайтів, додаток на React JS – це можливість створити інтерактивний продукт, в якому інтерфейс буде максимально швидко відповідати на будь-які дії відвідувача платформи [1]. Також він дозволяє реалізувати більш складні, інтерактивні елементи інтерфейсу, якщо порівнювати з реалізацією на сайті.

На стороні клієнта можна виконувати: більш інтерактивну анімацію; складні розрахунки; створювати offline-first додатки, що дозволяють функціонувати при втраті інтернет-з'єднання; максимально чуйні форми і таблиці, які не потребують перезавантаження сторінки. Односторінковий додаток React (SPA) не є чимось особливим. Усі WEB-ресурси, зроблені за допомогою фреймворків, односторінкові за своєю структурою. SPA дозволяють відразу повністю завантажити всю бізнес-логіку продукту, і працювати подібно настільним додаткам, без постійних запитів сторінок окремо. Подібні сайти запитують тільки той контент, що вимагає поновлення.

Кожна WEB-сторінка є «деревом» із відгалуженнями з HTML та CSS-об'єктів [2] – воно називається Document Object Model (DOM). Технологія дозволяє накладати на Document Object Model лише окремі компоненти, з якими працює користувач, а решта елементів залишатимуться без змін. Будь-який з цих компонентів можна оновити, не перезавантажуючи всю сторінку. Один із способів застосування Reactjs –

односторінкові програми, створені на базі технології SPA (Single Page Application). Однакові елементи залишаються на місці, а при виконанні дій користувача підтягуються лише окремі React JS компоненти. Наприклад, якщо на кожній сторінці однакова шапка – не потрібно витрачати ресурси на її малювання. Це значно підвищує продуктивність програми та робить інтерфейс більш чутливим.

Даний фреймворк наділений наступними властивостями [2, 3]:

1) Однонаправленість передачі даних – передача властивостей від основних компонентів до дочірніх, при чому компоненти отримують властивості у незмінному вигляді, що не дає можливості їх змінити.

2) Віртуальний DOM (VDOM) – це концепція програмування, в якій ідеальне чи «віртуальне» представлення інтерфейсу користувача зберігається в пам'яті і синхронізується зі «справжнім» DOM за допомогою бібліотеки, такої як ReactDOM. Цей процес називається узгодженням.

Власне такий підхід і робить API React декларативним: ви вказуєте React, у якому стані повинен знаходитись інтерфейс користувача, а React, у свою чергу, вже домагається того, щоб DOM відповідав цьому стану. Це абстрагує маніпуляції з атрибутами, обробку подій та оновлення DOM вручну, які, в іншому випадку, довелося б використовувати при розробці додатків. У світі React термін «віртуальний DOM» зазвичай асоціюється з React елементами, оскільки вони є об'єктами, що представляють інтерфейс користувача. Проте, React також використовує внутрішні об'єкти, так звані «волокна» (fibers) для зберігання додаткової інформації про дерево компонентів. Вони також можуть вважатися частиною реалізації «віртуального DOM» в React.

3) Розширення JSX – це розширення синтаксису мови JavaScript, він використовується для пояснення React зовнішнього вигляду інтерфейсу користувача (UI).

Цей кумедний синтаксис тегів не є ні рядком, ні HTML. Він має назву JSX, і це розширення синтаксису для JavaScript. JSX може нагадувати мову шаблонів, але з усіма перевагами JavaScript. JSX створює "React-елементи". React використовує той факт, що логіка виводу пов'язана з іншою логікою інтерфейсу користувача: як обробляються події, як змінюється стан з часом і як дані готуються для рендерингу. Замість того, щоб штучно відокремлювати технології, розмістивши розмітку і логіку в окремих файлах, React розділяє відповідальність між вільно зв'язаними одиницями, що містять обидві технології і називаються "компонентами".

4) React Hooks допомагає використовувати можливості React. При цьому відпадає необхідність у написанні класів. Хуки – це функції JavaScript, але вони накладають два додаткових правила. Хуки слід викликати тільки на найвищому рівні. Хуки слід викликати тільки з функційних React-компонентів.

5) Методи життєвого цикла – це методи, які допомагають запустити код на будь-якій стадії життєвого циклу компонента:

– componentWillMount – використовується перед рендерингом, в основному для налаштування компоненту;

– render - процес рендерингу;

– componentDidMount - вказує, на те, що компонент з'єднаний із DOM деревом;

– componentWillReceiveProps - вказує, на те, що надходять нові властивості, які входять в компонент;

– shouldComponentUpdate - повертає true або false і слугує для оптимізації. Вирішує, чи потрібно робити рендеринг;

– componentWillUpdate – повідомляє, що документ було поновлено;

– componentDidUpdate - повідомляє, що документ було поновлено;

– componentWillUnmount – використовується для видалення слухачів та очистки компоненту. Викликається перед видаленням компоненту.

Основним недоліком фреймворку є те, що даний фреймворк охоплює тільки рівень представлення WEB-системи, тобто призначений тільки для створення клієнтської частини додатку, інтерфейсу користувач.

Висновки

Проведено аналіз фреймворку мови програмування JavaScript. Дослідженf архітектура «клієнт - сервер», яка дала змогу обґрунтувати використання фреймворку React,який забезпечує однонаправленість передачі даних, віртуальний DOM, розширення JSX, react Hooks та методи життєвого циклу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Клієнт-серверна архітектура та ролі серверів. (2017). [Електронний ресурс]. Режим доступу: https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та - ролі-серверів-9893d8048229. Дата звернення: Березень 2023.

2. Розробка додатків на React JS. (2022). [Електронний ресурс]. Режим доступу: https://artjoker.ua/uslugi/razrabotka-prilozheniy-na-react-js/. Дата звернення: Березень 2023

3. ReactJS. (2023). [Електронний ресурс]. Режим доступу: https://uk.reactjs.org/. Дата звернення: Березень 2023.

Федорова Вероніка Володимирівна – студентка групи 2КН-22м, факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: stud.fedorova.veronika@vntu.edu.ua.

Іванчук Ярослав Володимирович – д-р техн. наук, професор, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця.

Fedorova Veronika V. - Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: stud.fedorova.veronika@vntu.edu.ua.

Ivanchuk Yaroslav V.- Dr. Sc. (Eng.), Professor of the Computer Science Department, Vinnytsia National Technical University, Vinnytsia.