

УДК 004.05

Шумченко С.Г.<sup>1</sup>, Киричек Г.Г.<sup>2</sup>

<sup>1</sup> студ. гр. КНТ-527, НУ «Запорізька політехніка»

<sup>2</sup> канд. техн. наук, доц., НУ «Запорізька політехніка»

## **АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ДИНАМІЧНИХ ВЕБДОДАТКІВ**

У сучасному світі веброзробок найбільшу роль грає швидкість, простота і можливості масштабування додатку. На даний час існує багато технологій, які використовуються для розробки веб-додатків та мають готові рішення: бібліотеки та фреймворки [1].

Метою роботи є проведення досліджень і визначення технології, яку найдоцільніше використовувати для розробки вебсистем різного розміру та призначення. Об'єктом дослідження є бібліотеки та фреймворки. Предметом – архітектура, методи, властивості та особливості їх застосування.

Для досягнення поставленої мети проведені порівняння за наступними характеристиками: бібліотека/фреймворк; вбудована архітектура; Single Page Application/Multi Page Application; переваги для інтерфейсів різного розміру; функціональність для роботи з DOM (Document Object Model), тобто швидкість розробки, представлення кінцевого результату, зворотня сумісність, поріг входження, тощо.

Бібліотека є комплексом підпрограм, які використовуються при розробці програмного забезпечення [2]. А фреймворк - програмна платформа, яка створює структуру вебдодатку та полегшує розробку різних компонентів системи [3]. Розглянемо дві популярні бібліотеки: React, jQuery та фреймворк Angular, визначимо їх переваги та недоліки, враховуючи те, що кожен з них створений для вирішення конкретних проблем.

jQuery є швидкою та багатофункціональною бібліотекою JavaScript. Вона спрощує роботу з документами HTML, обробку подій, анімацію та робить Ajax запити на сервер простішими за допомогою API. Завдяки поєднанню універсальності та розширюваності, jQuery змінив спосіб написання JavaScript. React є більш сучасною JavaScript бібліотекою, яка дозволяє створювати інтерактивні інтерфейси та односторінкові додатки (Single Page Applications). Дія у вкладці браузера не потребує перезавантаження сторінки, яка просто динамічно перезаписується, не завантажуючи нові сторінки. React одразу завантажує увесь скрипт, запитує у

сервера дані, які змінилися та вносить зміни в ті місця, які повинні оновитись. У випадку з jQuery завжди при взаємодії користувача відбувається перезавантаження сторінки [4].

Для маленьких додатків, які складаються з невеликої кількості компонентів, логіки та сторінок, краще використовувати jQuery, через його простоту та швидкість. JQuery дозволяє швидко створювати анімації та працювати з елементами DOM. Для landing page (спеціальний односторінковий сайт) jQuery також є найкращим вибором. React майже завжди може замінити jQuery, тому сучасні розробники використовують React. Виходячи з цього більш доцільним є порівняння React з іншою сучасною технологією Angular. Але React є бібліотекою, а Angular – фреймворком для створення динамічних SPA (Single Page Application) веб-додатків. Він використовує TypeScript та не має вбудованої типізації. Це на даний час є його перевагою над React [5]. Та щоб зрозуміти в яких випадках краще використовувати Angular, а в яких React, порівняємо їх за основними характеристиками.

Angular побудований на архітектурному патерні MVC (Model-View-Controller). Він відповідає: за дані та визначає структуру додатку; за зовнішній вигляд додатку та способи його використання; за зв'язок між моделлю та представленням, тобто як реагує сайт на дії користувача. А React має лише вбудовані функції для роботи з View. Тому в React потрібно створювати свою архітектуру додатку або використовувати Flux-архітектуру [6].

Flux-архітектура потрібна для керування станами на рівні усього додатку. Це часто потрібно для багаторівневих складних додатків. React-додаток який має деревоподібну структуру компонентів та їх вкладеність. При занадто великій вкладеності компонентів, важко керувати даними: потрібно створювати контексти для їх передачі по одній великій гілці. Flux-архітектура допомагає зберігати всі стани додатку, обробляти та отримувати до них доступ з різних частини додатку (рис. 1).

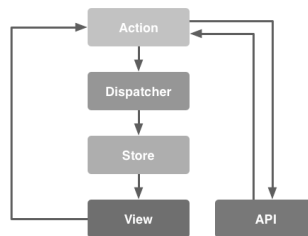


Рисунок 1 - Схема роботи Flux-архітектури.

У випадку з Angular, доступ до даних можна отримати за допомогою Angular сервісів, які є постачальниками даних, або за допомогою Angular Output декоратору, що дозволяє у батьківському компоненті відслідковувати зміни, які вносить дочірній. Тобто, завдяки впровадженій архітектурі, дочірній компонент маніпулює даними батьківських компонентів.

Що стосується відображення кінцевого результату: React створює копію DOM, обробляє її, а потім порівнює результат з початковою версією та відображає тільки ті частини сторінки, які відрізняються від результатів обробки. Це прискорює завантаження та відображення сторінки. React і Angular підходять по-різному до обробки DOM, що і впливає на відображення кінцевої сторінки. За рендерінг DOM сторінки в Angular відповідає браузер, а за створення директив, завантаження коду та сервісів – сервер [7]. Тобто, React відображає сторінки швидше, що є критичним для великих систем.

Оновлення архітектури для додавання нових модулів та бібліотек є важливою. Різниця між Angular та React у неможливості перейти з однієї версії Angular на іншу без наслідків. Модуль, який застосовувався у старій версії Angular можуть просто замінити на інший, що потребує використання нового коду. Якщо ви плануєте вдосконалити проект, додавши нові функції, React є найкращим вибором, тому що він має повну зворотну сумісність, яка дозволяє додавати різні версії бібліотек до програм та оновлювати старі [8]. Окрім того, обидві технології на початку розробки потребують високого рівня знань JavaScript, але увійти у React легше, бо він частіше використовує чистий JavaScript синтаксис. Angular використовує двосторонній підхід прив'язки, що автоматично змінює стан моделі при змінах в елементі інтерфейсу, а React використовує односторонній підхід, тому в Angular для досягнення потрібного результату необхідно менше дій.

При веброзробках використовується багато вебтехнологій, які часто замінюють одна одну. JQuery є застарілою бібліотекою, яка має перевагу у швидкості розробки маленьких, неодносторінкових додатків. React і Angular є чудовими варіантами для односторінкових додатків, мають невеликий поріг входження, багато переваг та небагато недоліків, тому вибір серед них потрібно робити, виходячи з потреб, функціональності та зручності використання.

## **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Wohlgethan, E. Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js /E. Wohlgethan //дис. – Hochschule für Angewandte Wissenschaften Hamburg, 2018. – 84 p.
2. jQuery, [Електронний ресурс]. – Режим доступу: <https://jquery.com/>.
3. Le, T. Implementation of React-Redux in web application /T. Le, N. Tran. – 2020. – 46 p.

4. React, [Электронный ресурс]. – Режим доступа: <https://reactjs.org/>.
5. Fedosejev, A. React. js essentials /A. Fedosejev. – Packt Publishing Ltd, 2015. – 179 p.
6. Freeman, A. Understanding React /A. Freeman //Pro React 16. – Apress, Berkeley, CA, 2019. – P. 31-36.
7. Saks, E. JavaScript Frameworks: Angular vs React vs Vue /E. Saks. – 2019. – 47 p.
8. Angular, [Электронный ресурс]. – Режим доступа: <https://angular.io/>.