УДК 004.4
UDC 004.4

DOI:10.33744/0365-8171-2022-111-226-233

ТИПОВІ ОСОБЛИВОСТІ АРХІТЕКТУРИ ОДНОСТРІНКОВИХ КРОСПЛАТФОРМЕНИХ ВЕБ ДОДАТКІВ

TYPICAL FEATURES OF THE ARCHITECTURE OF ONE-PAGE CROSSPLATFORM WEB APPLICATIONS

Безверхий Олександр Ігорович, *доктор фізико-математичних наук, професор, Національний транспортний університет, професор кафедри інформаційних систем і технолоій, o_bezver@ukr.net, тел. +380676849131, Scopus ID:6603638908, 14067133500*

https://orcid.org/0000-0002-0834-6335

Куценко Олександр Іванович, *Національний транспортний університет, аспірант кафедри інформаційних систем і технолоій, alexkutsenko95@gmail.com*

https://orcid.org/0000-0003-0047-4874

Шкабура Олександр Юрійович, *Національний транспортний університет, аспірант кафедри інформаційних систем і технолоій, ashkabura@gmail.com*

https://orcid.org/0000-0002-3335-066X

Анотація. Робота присвячена дослідженню розробки односторінкових веб-додатків, елементів архітектури, популярних шаблонів проектування, інструментів, мов програмування та фреймворків. Під час роботи були розглянуті існуючі односторінкові вебдодатки, детально досліджено одну з бібліотек Reakt, яка наразі є дуже ефективною та популярною. Для управління станом данних та станом інтерфейса в джаваскріпт додатках застосовується інструмент Redux, він підходить для односторінкових додатків. Проаналізовані недоліки та переваги концепції односторінкових веб-додатків у порівнянні з традиційним веб-сайтом та нативним додатком. Розроблені рекомендації що до створення додатків.

Ключові слова: односторінковий веб-додаток, нативний додаток, ДжаваСкріпт, Реакт, шаблони проектування.

Вступ Сьогодні веб-технології розвиваються дуже стрімко. Open-sourse відкриває для програмування безмежні можливості. Кроссплатформеність та доступність веб сайтів роблять їх дуже привабливими для людей. Розвиток веб технологій сьогодні дозволяє будувати не лише веб сайти у звичайному сенсі цього слова, коли сайт складається зі сторінок, а й однострінкові веб-сервіси які можна називати повноцінними додатками. Такі веб-сервіси вже становлять значну конкуренцію нативним додаткам, що змушує розробників створювати онлайн версії популярних додатків.

Односторінковий веб-додаток (SPA) – це клієнтська (front-end) концепція веб-сайту, що робить його схожим на звичайний додаток, зберігаючи при цьому всі переваги повноцінного веб-сайту. Ідея SPA не нова, вона почала з'являтися як тільки з'явилися JavaScript та AJAX й веб-сайти почали набувати динаміки будучі вже не лише статичними сторінками. Мабуть кожному розробнику сайтів рано чи пізно приходили в голову ідеї про сайт-додаток, який завантажується лише один раз, а все інше відбувається за рахунок скриптів та асинхронних запитів. Проте об'єм роботи, який треба було зробити не був вартий цього, адже для створення веб-сайту за концепцією SPA треба прикласти значних зусиль. Полегшити розробку такого виду сайтів допоможе використання безліч технологій, фреймворків й інструментів, що були для цього розроблені в останній час. Використовуючи ці технології можна будувати чудові веб-додатки витрачаючи на це час порівняний з часом, необхідним на розробку традиційного нативного додатку.

Метою даної роботи є збір теоретичних відомостей про концепцію односторінкового веб-додатку та типові особливості їх архітектури, огляд існуючих популярних та відомих SPA додатків, технологій, що були використані при їх розробці, огляд інструментів та фреймворків, що використовуються при їх розробці, набуття практичних навичок у створенні односторінкового вебдодатку з використанням одного з фреймворків.

1. Огляд концепції односторінкового веб-додатку

Розглянемо звичайний, традиційний сайт, наприклад Wikipedia[8]. Відкриваючи статтю користувач чекає поки браузер зв'яжиться з сервером й останній обробить його запит. Потім браузер підвантажує різні ресурси необхідні для відображення сторінки, паралельно відтворюючі ті, що вже завантажились. Під час цього процесу користувач бачить як елементи сторінки скачуть й мерехтять, перераховуючи місце необхідне під зображення та блоки, що підвантажились пізніше. В залежності від пропускної здатності мережі та завантаженості сервера цей процес може тривати доволі довго, дратуючи користувача. Коли користувач захоче переглянути іншу статтю, відредагувати цю, або зробити ще якісь дії на цьому сайті, його чекає перезавантаження сторінки за новим URL. Частина ресурсів буде кешована браузером й завантаження нової сторінки має пройти швидше, але всі недоліки даного процесу прибрати все одно не вдасться.

Так працює більшість сайтів в інтернеті. Але з моменту створення Web пройшло багато років й багато змін, з'явилось багато технологій, JavaScript, AJAX та інші, сайти почали набувати інтерактивності й бути чимось більшим ніж просто сторінками. Зараз розробники мають достатньо технологій, щоб змінити поведінку традиційного сайту й зробити його схожим на нативний додаток операційної системи. Для цього всі зміни на сторінці веб-сайту роблять за допомогою скриптів, а запити до сервера використовують AJAX. Відкриваючи такий сайт користувачу доводиться лише раз завантажити його сторінку, а зміни на сторінці, які не потребують нових даних, виконуються швидше, оскілки не чекають обробки запитів на сервері. Така концепція веб-сайту отримала назву односторінковий додаток (Single Page Application). Все більше й більше нових сайтів створюють саме такими.

Отже, незважаючи на назву, суть концепції односторінкового додатку полягає не в обмеженні сторінок веб-сайту, а в наданні сайту відчуття нативного додатка, шляхом позбавлення перезавантажень й перенесенні відтворюючої логіки з сервера до клієнта, тим самим зменшуючи залежність роботи сайту від сервера, інтернет з'єднання, а найголовніше швидкості роботі на сайті.

2. Порівняння з традиційним веб-сайтом та нативним додатком

Концепція односторінкового додатку виглядає дуже привабливо, проте вона теж має певні недоліки та переваги.

Переваги односторінкових додатків:

1. Насичений функціоналом інтерфейс
2. Швидка реакція інтерфейсу, завдяки відсутній необхідності звертатися до серверу при кожній дії.
3. Значне зменшення навантаження на сервер.
4. Значне спрощення логіки та складності серверу.

5. Схожість додатків з нативними додатками операційної системи.

6. Підвищення навантаження на клієнт завдяки великої кількості JavaScript , зараз навіть у дешевих гаджетах 8 ядерні процессори.

7. Адаптивне відображення на мобілних єкранах.

Недоліків немає.

Нативні додатки це те, на що намагається бути схожим односторінковий веб-додаток, тобто те до чого він прагне. Справедливо буде поставити питання «Чому не створити справжній нативний додаток?». Справа в тому, що нативні додатки також мають свої недоліки. Односторінковий веб-додаток намагається перейняти найкраще від обох, веб-сайту й нативного додатку та адоптивно відображатися за вашого мобільного девайся (телефон планшет і тд…).

Однак, в певних випадках, використання звичайного нативного додатку все ще може бути вигіднішим, або й навіть необхідним. Порівняємо їх недоліки й переваги.

Переваги односторінкових веб-додатків:

1. Багатоплатформність.
2. Відсутність необхідності встановлення.
3. Не займає місце на жорсткому диску.
4. Можливість віддалених розрахунків.

Переваги нативних додатків:

1. Вільність у вибору мов програмування та інтерфейсів.
2. Продуктивність нативного коду.
3. Використання нативних можливостей платформи.
4. Доступ до будь якого обладнання комп’ютера.
5. Незалежність від інтернет з’єднання.

Таблиця 1 – Порівняння традиційного веб-сайту, односторінкового веб-додатку та нативного додатку.
Table 1 – Comparison of a traditional website, a one-page web application and a native application.

	Веб-сайт	Односторінковий веб-додаток	Нативний додаток
Чутливість інтерфейсу	- маленька	+ велика	+ велика
Встановлення	+ не потрібно	+ не потрібно	- потрібно
Місце на ПЗУ	+ не потрібно	+ не потрібно	- потрібно
Інтернет з’єднання	--дуже потрібно	- потрібно	+ не потрібно
Наявність серверу	--дуже потрібно	- потрібно	+ не потрібно
Навантаження на клієнт	+ середнє	- велике	+ середнє
Доступ до обладнання клієнта та можливостей платформи	- не має	- не має	+ є
Багатоплатформність	+ чудова	+ чудова	- погана
Гнучкість у вибору мови й інструментів розробки	- погана	- погана	+ гарна

3. Бібліотека для відображення React

React – відкрита JavaScript-бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків[1,2]. Розробляється Facebook, Instagram і спільнотою індивідуальних розробників.

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб піклуватися про частини без користувацького інтерфейсу побудови веб-застосунків[3].

В даний час React використовують Khan Academy, Netflix, Yahoo, Airbnb, Sony, Atlassian та інші..

React підтримує віртуальну реальність у браузері. Віртуа́льна реа́льність (англ. Virtual reality) – різновид реальності в формі тотожності матеріального й ідеального, що створюється та існує завдяки іншій реальності. В вужчому розумінні — ілюзія дійсності, створювана за допомогою комп'ютерних систем, які забезпечують зорові, звукові та інші відчуття. На React також можна писати кроссплатформені додатки які однаково відкриваються на Android та IOS

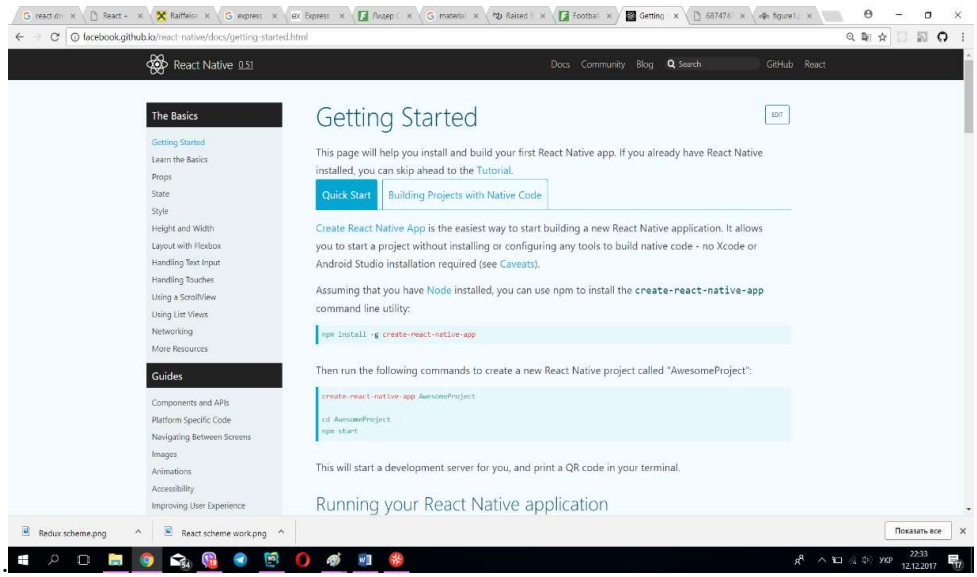

Рисунок 1 – Сайт з офіційної документації створений компанією Facebook
Figure 1 – The site from the official documentation was created by Facebook

Основні особливості:

1. Одностороння передача данних: Властивості передаються в рендерер компоненту, як властивості html тега. Компонент не може напряму змінювати властивості, що йому передані, але може їх змінювати через callback функції. Такий механізм називають «властивості донизу, події нагору».

2. Підтримує віртуальний DOM, а не покладається виключно на DOM браузера. Це дозволяє бібліотеці визначити, які частини DOM змінилися, порівняно (diff) зі збереженою версією віртуального DOM, і таким чином визначити, як найефективніше оновити DOM браузера. Таким чином програміст працює зі сторінкою, вважаючи що вона оновлюється вся, але бібліотека самостійно вирішує які компоненти сторінки треба оновити

3. Компоненти React зазвичай написані на JSX. Код написаний на JSX компілюється у виклики методів бібліотеки React. Розробники можуть так само писати на чистому JavaScript або ES6+. JSX нагадує іншу мову, яку створили у компанії Фейсбук для розширення PHP, XHP. Представлення з декларативною обробкою подій

4. Може використовуватись не лише для рендерингу HTML в браузері. Наприклад, Facebook має динамічні графіки які рендеряться в теги <canvas>, Netflix та PayPal використовують ізоморфне завантаження для рендерингу ідентичного HTML на сервері та клієнті.

Для організації клієнтської маршрутизації в React існує бібліотека react-router , такою є redux-router для зберігання маршрутизації в Redux. Вона досить легка для написання і не потребує особливого пояснення на мій погляд.

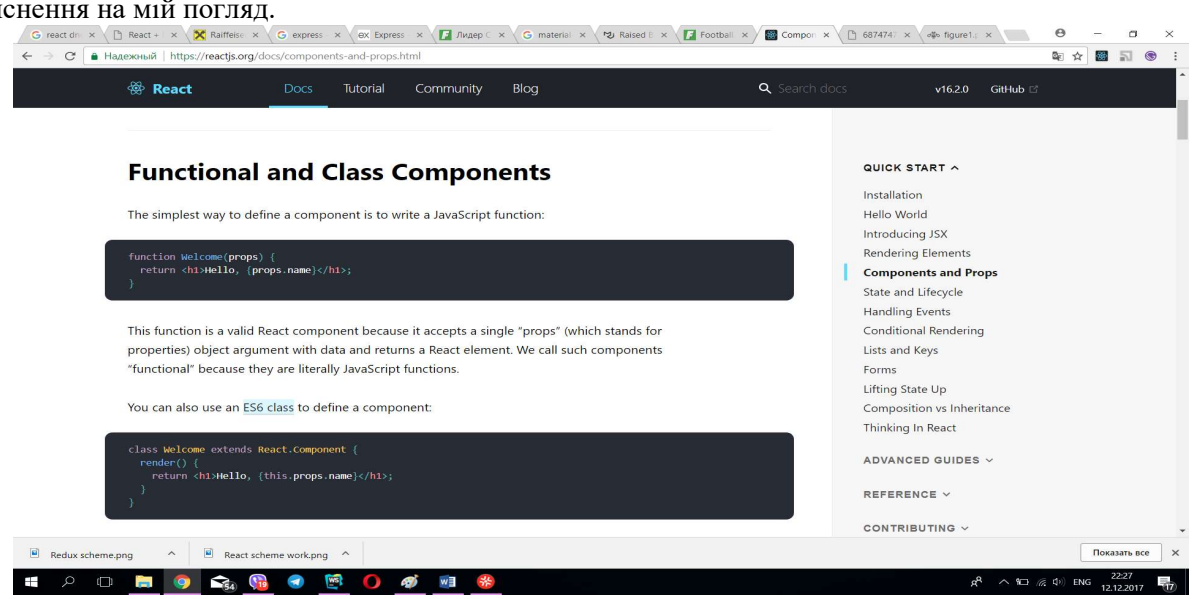

Рисунок 2 – Компоненти React
Figure 2 – React components

Веб-додаток React складається з компонентів. Є 2 види компонентів: функціональні

```
function Hello(name = Eugene) {
  return <h1>Привет, `${name}`</h1>;
}
```

та класові.

```
class Hello extends React.Component {
  constructor(props) {
  super(props);
  this.state = {
    color: props.initialColor
  };
}
render() {
    return <h1>Привет, Eugene</h1>;  }}
```

В класових компонентах є стейт (стан) , методи , констркутор , в функціональних тільки параметри. Стейт можна змінювати та передавати через проперті в дочірній компонент. У React немаю автобінду , тому кожен метод треба прибінджувати вручну.

Компоненти можуть використовувати інші компоненти як дочірні, складаючи зручну ієрархію елементів додатку. Зв'язування так само працюють й між батьківськими та дочірніми компонентами.

У React також є проп тайпи , там можна відстежувати типи обьектів які передаються до компонента (Метод , масив , moment , date , func etc….)

Для кращого відображення компонента у React створили звичані методи для маніпулювання ним.

Component Will Mount , Component Did Mount , Component Did Catch , componentWillReceiveProps , componentWillUnmount, shouldComponentUpdate. Ці методи

створенні для того щоб повністю контролювати процесс перемальовки компонента та досягнення максимальної продуктивності .

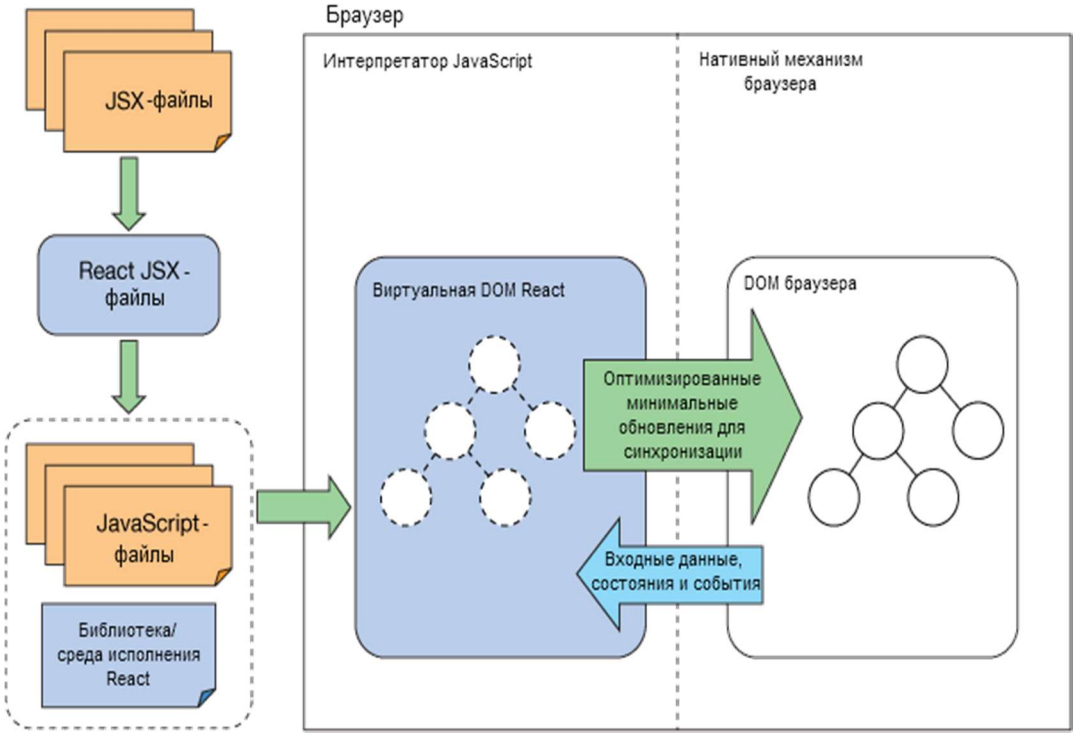

Рисунок 3 – Прицип роботи бібліотеки React
Figure 3 – The principle of the React library

Для зберігання стану використовуется бібліотека Redux (наприклад для локалізації , маршрутизації та інш).

Redux – це інструмент для управління станом данних та станом інтерфейса в джаваскріпт додатках[4,5]. Він підходить для односторінкових додатків. Його використовують для того щоб не передавати React стани через проперті через то що це робить код поганим та веде до спагетті коду. Місце зберігання відбувається в обьекті (store) та міняются через спеціальні методи (actions).Після зміну стану Редакс повністью перемальовуеться тому що він є незмінним обьектом. Дані зберігаються у форматі JSON. Зміни робляться за допомогою Диспатчера.

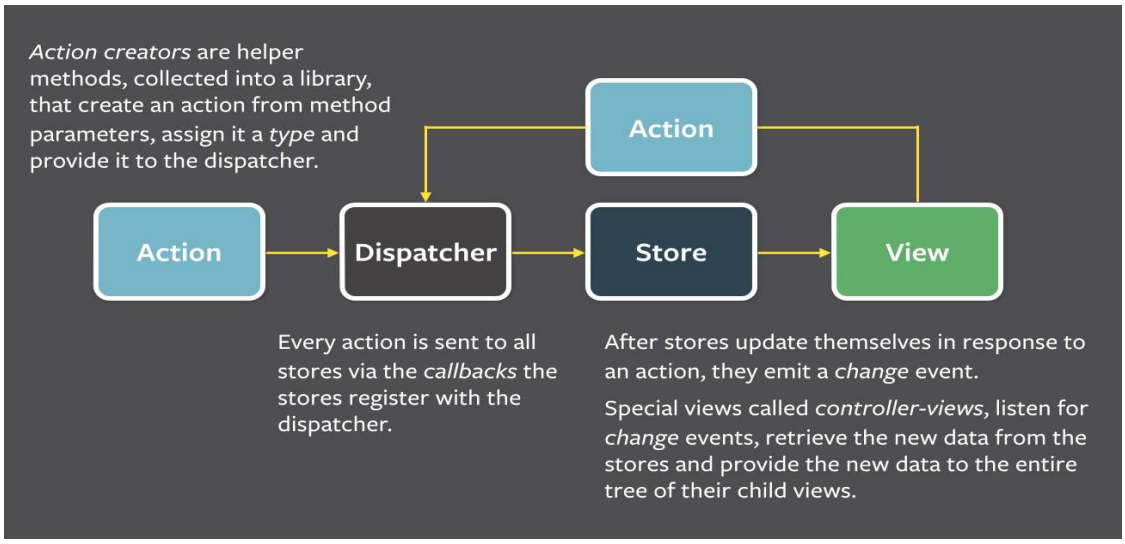

Рисунок 4 – Принцип роботи бібліотеки Redux
Figure 4 – The principle of the Redux library

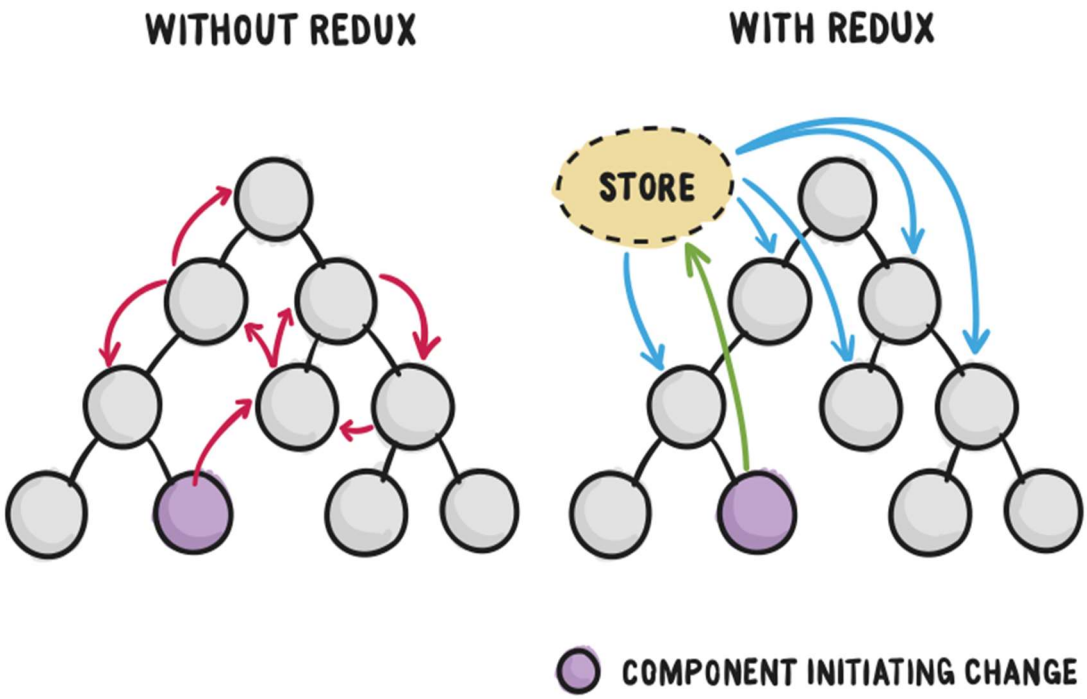

Рисунок 5 – Принцип роботи бібліотеки Redux
Figure 5 – The principle of the Redux library

Висновки

В роботі, було детально досліджено популярну сьогодні концепцію веб-сайту “односторінковий додаток” та ключові елементи що її складають, наведені приклади сучасних односторінкових додатків. Проаналізувавши недоліки та переваги концепції у порівнянні з традиційним веб-сайтом та нативним додатком було встановлено, що на сьогодні нативні додатки мають перевагу у продуктивності, мережевою незалежності та більших можливостях. Односторінкові веб-додатки в свою чергу мають переваги в багатоплатформності й відсутній необхідності встановлення.

Однак у майбутньому межі між ними будуть все більш розмиватися. Також було встановлено що концепція проста й була придумана вже давно, проте набула популярності тільки нещодавно, оскільки з’явились інструменти, фреймворки та технології що дозволяють значно зменшити великий об’єм що необхідний для реалізації даної концепції та значно спростити процес розробки.

Досліджені типові особливості архітектури односторінкового веб-додатку відображують тенденцію до адаптації вже відомих шаблонів проектування та рішень що використовуються при розробці нативних додатків. Це ще раз підкреслює мету концепції односторінкових вебдодатків, що полягає у надаванні веб-сайтам властивостей нативних додатків.

Встановлено, що архітектура односторінкового додатку має певні необхідні елементи, проте велика кількість підходів не дозволяють створити єдиний правильний варіант. Вибір технології, архітектурних рішень та підходів дуже залежить від задач які буде вирішувати додаток та його складності.

Підсумовуючи можна сказати що концепція односторінкових додатків має значні переваги і гарну перспективу розвитку у майбутньому. Це підтверджується значним використанням її при створенні нових веб-сайтів та великою кількістю існуючих. Кількість інструментів, фреймворків та технологій основним використанням яких є розробка односторінкових веб-додатків швидко росте. Це суттєво спрощує створення нових додатків, проте створення великого додатку сьогодні все ще залишається складною задачею.

Перелік посилань

1. https://reactjs.org/ Офіційний сайт React
2.https://codeguida.com/post/3084
3.https://habr.com/ru/company/kts/blog/598571/
4. https://redux.js.org/docs/introduction/ Офіційний сайт Redux
5.https://hawkingbros.com/article/redux-i-vse-vse-vse
6.https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types
7. http://expressjs.com/ru/ Офіційний сайт сервера Експреса
8. https://uk.wikipedia.org/ Офіційний сайт Вікіпедії

TYPICAL FEATURES OF THE ARCHITECTURE OF ONE-PAGE CROSSPLATFORM WEB APPLICATIONS

Bezverkhy Oleksandr I., Doctor of Physical and Mathematical Sciences, Professor, National Transport University, Professor of the Department of Information Systems and Technologies, o_bezver@ukr.net, tel. +380676849131
ORCID: 0000-0002-0834-6335, Scopus ID: 6603638908, 14067133500
Kutsenko Alexander I., National Transport University, graduate student of the Department of Information Systems and Technologies, alexkutsenko95@gmail.com,
ORCID: 0000-0003-0047-4874
Shkabura Oleksandr Yu., National Transport University, graduate student of the Department of Information Systems and Technologies, ashkabura@gmail.com

Abstract. The work is devoted to the study of the development of one-page web applications, architectural elements, popular design templates, tools, programming languages and frameworks. During the work, the existing one-page web applications were reviewed, and one of the Reakt libraries, which is currently very effective and popular, was studied in detail. To control the state of the data and the state of the interface in javascript applications, the Redux tool is used, it is suitable for single-page applications. The disadvantages and advantages of the concept of one-page web applications in comparison with the traditional website and the native application are analyzed. Developed recommendations for creating applications.

Key words: one-page web application, native application, web application, JavaScript, React, design patterns.

Referens

1. https://reactjs.org/ Official site of React
2.https://codeguida.com/post/3084
3.https://habr.com/ru/company/kts/blog/598571/
4. https://redux.js.org/docs/introduction/ Official site of Redux
5.https://hawkingbros.com/article/redux-i-vse-vse-vse
6.https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types
7. http://expressjs.com/ru/ Official site of the Express server
8. https://uk.wikipedia.org/ Official site of Wikipedia