

УДК 004.9**DOI: 10.31866/2617-796X.4.2.2021.247490****Овчарук Ірина,***кандидат технічних наук,**доцент кафедри інформаційних технологій і дизайну,**Державний університет інфраструктури та технологій,**Київ, Україна**ovch05@ukr.net**<https://orcid.org/0000-0003-4255-5816>***Горбенко Олександр,***магістрант,**кафедра інформаційних технологій і дизайну,**Державний університет інфраструктури та технологій,**Київ, Україна**jioji888@gmail.com**<https://orcid.org/0000-0001-8010-7294>*

SINGLE PAGE APPLICATION ДЛЯ ЗАСТОСУНКУ ВІДСТЕЖЕННЯ ПОГОДНИХ УМОВ

Метою статті є огляд концепцій для створення вебзастосунків, аналіз переваг і недоліків сучасних технологій для прогнозування погодних умов. У статті представлено сучасні клієнт-серверні технології. Докладно розглянуто концепцію Single Page Application та інші технології, що дають змогу взаємодіяти між клієнтом і сервером у режимі онлайн. Представлена розробка застосунку для відстеження погодних умов на основі архітектури застосунку SPA.

Методами дослідження є опрацювання теоретичної інформації та проведення аналізу архітектури застосунку, побудованого за принципом SPA, проведення аналізу переваг і недоліків підходів взаємодії клієнтської частини із сервером, використання бібліотек Angular, Vue, React, використання API для створення застосунків.

Новизною проведеного дослідження є аналіз сучасних концепцій і технологій, що дають змогу розробляти вебзастосунки, за допомогою яких можна збудувати сучасний вебзастосунок концепції Single Page Application; наведено докладний аналіз бібліотек Angular, Vue, React і взаємодію користувача з такими системами, аналіз технології Service Workers як способу хешування.

Висновки. У статті проведено аналіз сучасних технологій створення вебзастосунків, наведено їх переваги та недоліки, розглянуто концепцію Single Page Application та описано розробку застосунку для відстеження погодних умов, наведено аналоги для порівняльної характеристики, розглянуто проблеми, які виникали в процесі взаємодії клієнтської частини із сервером до застосування принципів SPA.

Ключові слова: односторінковий вебзастосунок SPA; фреймворк React; фреймворк Angular; фреймворк Vue.js; технологія AJAX.

Вступ. Прогнозами погоди цікавляться майже всі. Важливими прогнозами є штормові попередження, оскільки, завчасно знаючи про наближення природного катаклізму, можемо зберегти життя людей і їхнє майно. Прогнози температури й опадів важливі для сільського господарства, а отже, для трейдерів на фондових ринках. Температурні прогнози потрібні працівникам теплових мереж для оцінки потреби в тепловій енергії. Щодня люди користуються прогнозом погоди, щоб вирішити, що одягнути. Прогнози дощів, снігу та сильних вітрів потрібні для планування робіт і відпочинку на свіжому повітрі. Досить важливими є прогнози сейсмічної активності та небезпечних явищ, якщо довідатися про них заздалегідь, то можна врятувати життя багатьом людям.

З розвитком інформаційних технологій більшість людей користується вебзастосунками для того, щоб знати прогноз погоди. Отже, аналіз новітніх концепцій і технологій, за допомогою яких можна збудувати сучасний вебзастосунок, і створення застосунку для відстеження погодних умов є актуальним питанням.

Результати дослідження. Концепція SPA (Single Page Application) дає змогу створити безшовний застосунок, який завантажується лише раз. Вона досить ефективна та зрозуміла користувачу, адже навігація в такій системі між різними сторінками досить зручна, замість білих екранів завантаження сторінки користувач має змогу спостерігати розповсюджені анімації підзавантаження, тому це досить спрощує роботу UI/UX-дизайнерів.

– Раніше, до створення концепції SPA, використовували так званий Server-side rendering, який генерував кожен вебсторінку окремо, залежно від зміненої інформації. Цей підхід був досить громіздкий, тому запропонували ідею для використання асинхронної передачі даних (Мак-Фердис, 2002; Херман, 2015). Цю ідею вперше описано в статті Джесі Джеймса Гарретта «Новий підхід до вебзастосунків». Наразі такий підхід називається AJAX (Asynchronous JavaScript And XML). За його допомогою стало можливо отримувати дані без залежності від генерації сторінки сервера, тобто змінювати лише незначну частину вже наявної інформації на сторінці. На базі цієї ідеї збудовано концепцію SPA, яка дає змогу уникнути так званих «white screen», що спантеличують користувача, адже неприємно щоразу спостерігати білий екран під час будь-якої дії. Тож AJAX -Service Worker – окремий від основного змісту скрипт, який дає змогу хешувати файли;

– Web App Manifest – JSON файл, який дає змогу браузеру використати додаткові можливості: встановити іконку сайту до головного екрану смартфона, як буде називатися застосунок, іконка сайту, розмір вікна в браузері для телефонів;

– HTTPS – безпечний протокол передачі даних;

– App shell – це каркас застосунку, який має функціонувати навіть без даних, що змінюються динамічно;

– Push Notification – ще одна можливість Service Worker, яка дає змогу відправляти нотифікації як на мобільний пристрій, так і на комп'ютер.

PWA найчастіше створюють адаптивним, він має підлаштовуватися під розмір екрану та мати добрий вигляд як на мобільних пристроях, так і на комп'ютері.

Ще одним кроком для розвитку PWA є використання технологій, які портують вебзастосунок у мобільний. Для цього використовують такі програмні засоби, як

Cordova. Ця бібліотека дає змогу звичайному застосунку запускатися в нативному середовищі, яке вона створює. Середовище має доступ до різних апаратних та програмних засобів мобільного пристрою: камери, ліхтарика, сканера відбитка пальців, технології NFC, push-нотифікації.

Push Notification ще одна можливість Service Worker, адже Service Worker може працювати автономно, навіть коли сайт не перебуває в активній вкладці. Тож це надає змогу користувачеві одержувати нотифікації під час отримання певної інформації із сайту. Це корисно, тому що можна отримувати інформацію досить швидко та її можна легко замітити, особливо це зручно в месенджерах та інших схожих застосунках (рис. 1).



Рис. 1. Push Notification на мобільному пристрої

Ще однією особливістю Push Notification є те, що вони будуть працювати лише за згоди користувача. Отже, користувач сам може вирішувати, які нотифікації йому потрібні, а які – ні.

Розглянемо відомі аналоги застосунку для відстеження погодних умов. Наразі є безліч сервісів, які дають змогу отримати інформацію про погодні умови (рис. 2).

Основними недоліками таких сервісів є досить незручний інтерфейс, який не має змоги адаптуватися до мобільних пристроїв. Наприклад, змінимо розмір екрану (рис. 3).



Рис. 2. Сервіс надання інформації про погодні умови



Рис. 3. Розміром вікна зменшено

Цей сайт не є адаптивним, тому користувач не має змоги зручно використовувати його на будь-якому пристрої. Ще одним недоліком таких сайтів є те, що в полі зору користувача перебуває інформація щодо погодних умов лише за певним місцем розташування.

У цій статті описано розробку застосунку для відстеження погодних умов, головне завдання розробки якого – можливість використовувати на будь-яких пристроях. Застосунок також має містити інформацію про погодні умови в певному місці. За основу взято бібліотеку React, за допомогою якої створено інтерфейс застосунку. Це вписується до моделі PWA. Для вебзастосунку, що в процесі розробки, також будуть використовуватися різні допоміжні інструменти:

1. Redux – спрощує підхід для контролювання станів компонентів та надає змогу створювати один суцільний стан для контролю всього застосунку.

2. Redux-Saga – інструмент, який полегшує створення індикаторів станів. Це спрощує звернення до сервера, та маємо змогу завжди контролювати запити.

3. React-Router-Dom – інструмент для навігування по застосунку.

Для отримання певної інформації буде використано сервіс OpenWeather, який завжди має актуальну інформацію щодо погодних умов. Він має відкритий API.

Під час підготовки також буде розроблено сервер за допомогою мови Node.js, яка базується на синтаксисі JavaScript, але має кардинальні відмінності. Отже, Node.js має доступ до файлової системи директорій. Під час створення сервера буде використано підхід REST, який забезпечить бібліотека Express.js. Для збереження інформації користувача за основу буде взято базу даних MongoDB та бібліотеку Mongoose, яка спрощує створення моделей і надає змогу для створення валідаторів. Застосунок має відповідати критеріям: якості, швидкодії, надійності, адаптивності, дружелюбності.

Однією з відмінностей застосунку є те, що клієнт майже напряму має зв'язок з API від OpenWeather (рис. 4). У такому разі застосунку не потрібен посередник у вигляді сервера Node.js, але він слугує для збереження вподобань користувача, а саме списку місць, в яких потрібно відстежувати погодні умови; отже, кожен користувач має власний набір інформації, яка буде збережена в базі даних.

Застосунок матиме такий вигляд (рис. 5). Головним функціоналом буде доступ до інформації про погодні умови.

Кожен шаблон матиме погодну інформацію за певною місцевістю (рис. 6). Один шаблон поділено на три секції, кожна з яких має інформацію за різним часовим проміжком. Шаблон також має значок для додавання його до вибраного та значок для видалення шаблону.

Залежно від погодних умов рисунок заднього фону застосунку буде змінюватися (рис. 7).

Друга вкладка шаблону (рис. 8) демонструє прогноз погоди погодинно, також зображено індикатор помірних опадів. Ще вона має індикатор, який вказує на час, та індикатори, що відображають різні опади. Отже, є 4 кольори індикаторів станів:

1. Білий – без опадів.
2. Зелений – помірні опади.
3. Помаранчевий – сильні опади (град, шторм і т. ін.).
4. Червоний – небезпечні ситуації (виверження вулканів, попіл, землетруси).

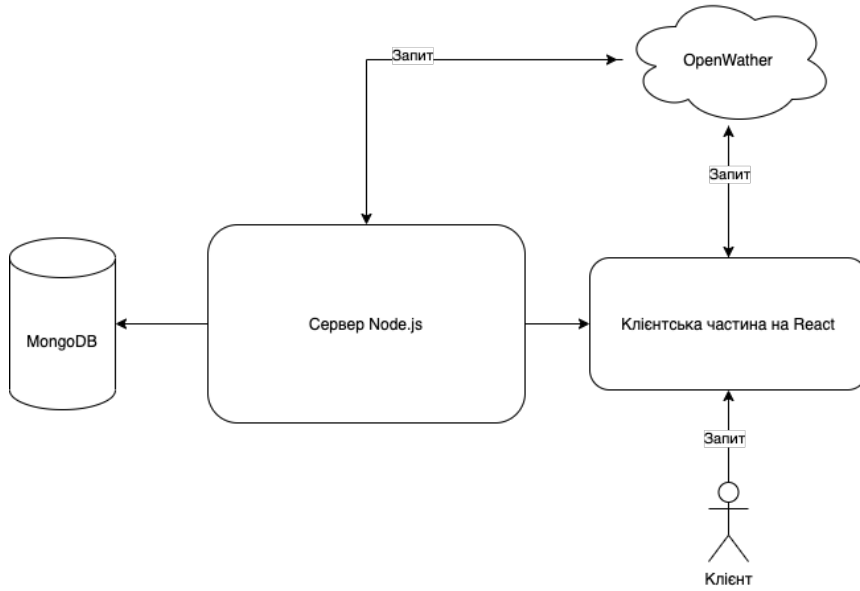


Рис. 4. Схема роботи застосунку

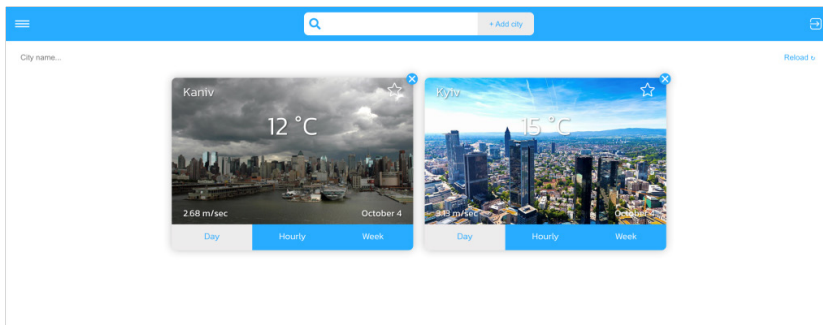


Рис. 5. Інтерфейс застосунку

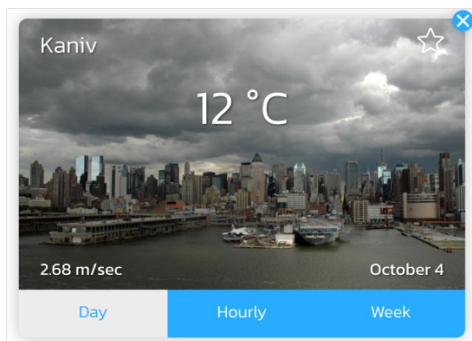


Рис. 6. Шаблон погодних умов за місцевістю

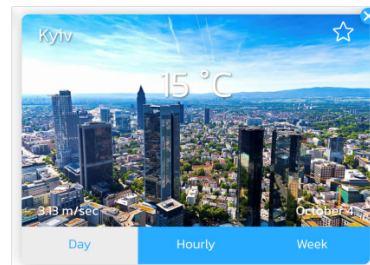


Рис. 7. Рисунок на шаблоні за ясної погоди

Третя вкладка демонструє погодні умови на наступний тиждень (рис. 9).

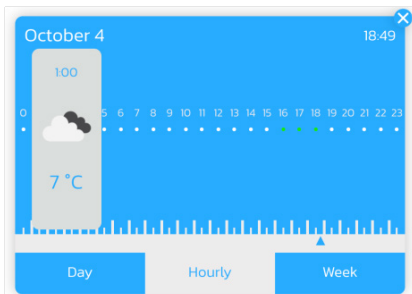


Рис. 8. Погодинна вкладка шаблону

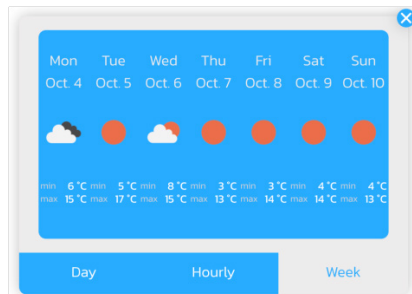


Рис. 9. Вкладка прогнозу погоди на тиждень

Для додавання нових шаблонів було створено пошуковий рядок, який є багатомовним (рис. 10).

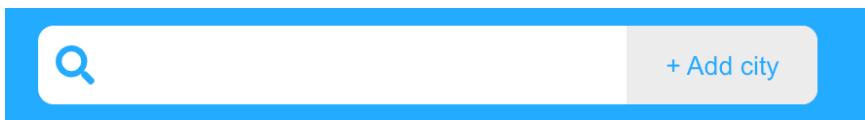


Рис. 10. Пошуковий рядок

Для кожного шаблону розроблено окрему сторінку, яка дає більш детальну інформацію про погодні умови за різним проміжком часу (рис. 11).

У застосунку створено можливість для авторизації користувача (рис. 12).

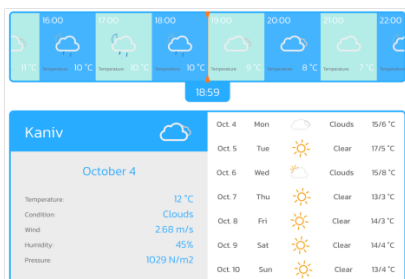


Рис. 11. Детальна сторінка шаблону



Рис. 12. Форма авторизації користувача

Висновки. У цій статті розглянуто концепцію SPA, описано випадки її використання та основні технології, які застосовують у процесі її створення. Описано переваги й недоліки наведених концепцій та похідні концепції, які розширюють поняття Single Page Application. Наразі часто використовують допоміжну технологію Service Worker, яка дає змогу хешувати дані вебзастосунку та створювати користувацькі нотифікації. Описано застосунок, що розроблюється, який призначений для відстеження погодних умов із упровадженням розглянутих технологій.

СПИСОК ПОСИЛАНЬ

- Гарретт, Д., 2015. *Веб-дизайн. Элементы опыта взаимодействия*. Перевод С. Иноземцева. Санкт-Петербург: Символ-плюс.
- Гудман, Д., 2015. *JavaScript и DHTML. Сборник рецептов. Для профессионалов*. Москва: Питер.
- Дронов, В.А., 2015. *JavaScript и AJAX в Web-дизайне*. Москва: БХВ-Петербург.
- Мак-Фердис, П., 2002. *Использование JavaScript*. Киев: Вильямс.
- Современный учебник JavaScript*. [online] Available at: <<https://learn.javascript.ru/>> [Дата обращения 05 сентября 2021].
- Херман, Д., 2015. *Сила JavaScript. 68 способов эффективного использования JS*. Москва: Питер.
- Хэррон, Д., 2014. *Node.js Разработка серверных веб-приложений на JavaScript*. Перевод с английского А.А. Слинкина. Москва: ДМК Пресс.
- Mongoose*. [online] Available at: <<https://mongoosejs.com/docs/guide.html>> [Accessed 05 September 2021].
- React documentation*. [online] Available at: <<https://reactjs.org/docs/getting-started.html>> [Accessed 05 September 2021].

REFERENCES

- Garrett, D., 2015. *Veb-dizain. Elementy opyta vzaimodeistviia* [Web Design. Elements of interaction experience]. Translated by S. Inozemtsev. St. Petersburg: Simvol-plus.
- Gudman, D., 2015. *JavaScript i DHTML. Sbornik retceptov. Dlia professionalov* [JavaScript and DHTML. Collection of recipes. For professionals]. Moscow: Piter.
- Dronov, V.A., 2015. *JavaScript i AJAX v Web-dizaine* [JavaScript and AJAX in Web Design]. Moscow: BXV-Peterburg.
- Mak-Ferdis, P., 2002. *Ispolzovanie JavaScript* [Using JavaScript]. Kyiv: Viliams.
- Sovremennyi uchebnik JavaScript* [A modern JavaScript tutorial]. [online] Available at: <<https://learn.javascript.ru/>> [Accessed 05 September 2021].
- Kherman, D., 2015. *Sila JavaScript. 68 sposobov effektivnogo ispolzovaniia JS* [The Power of JavaScript. 68 Ways to Use JS Effectively]. Moscow: Piter.
- Kherron, D., 2014. *Node.js Razrabotka servernykh veb-prilozhenii na JavaScript* [Node.js Developing Server-Side JavaScript Web Applications]. Translated from English by A.A. Slinkin. Moscow: DMK Press.
- Mongoose*. [online] Available at: <<https://mongoosejs.com/docs/guide.html>> [Accessed 05 September 2021].
- React documentation*. [online] Available at: <<https://reactjs.org/docs/getting-started.html>> [Accessed 05 September 2021].

УДК 004.9**Ovcharuk Iryna,***PhD in Technical Sciences,**Associate Professor at the Department of Information Technologies and Design,**State University of Infrastructure and Technology,**Kyiv, Ukraine**ovch05@ukr.net**<https://orcid.org/0000-0003-4255-5816>***Horbenko Oleksandr,***Master's Student at the Department of Information Technologies and Design,**State University of Infrastructure and Technology,**Kyiv, Ukraine**jioji888@gmail.com**<https://orcid.org/0000-0001-8010-7294>***SINGLE PAGE APPLICATION FOR WEATHER TRACKING APP**

The purpose of the article is to review the concepts for creating web applications, analyze the advantages and disadvantages of modern technologies for weather forecasting. The article presents modern client-server technologies. The concept of a single-page application and other technologies that allow interaction between the client and the server online are discussed in detail. The development of an application for tracking weather conditions based on the architecture of the SPA is presented.

The research methodology includes processing theoretical information and analyzing the architecture of the application, built on the principle of SPA, analyzing the advantages and disadvantages of approaches to client-server interaction, using Angular, Vue, React libraries, using APIs to create applications.

The novelty of the research is the analysis of the modern concepts and technologies that allow you to develop web applications that can be used to build a modern web application of the Single Page Application concept, a detailed analysis of Angular, Vue, React libraries and user interaction with such systems, analysis of Service Workers technology as a way of hashing.

Conclusions. The article analyzes modern technologies for creating web applications, presents their advantages and disadvantages, discusses the concept of Single Page Application and describes the development of an application for tracking weather conditions, provides analogues for comparative characteristics, discusses problems that arise when the client part interacts with the server. application of SPA principles.

Keywords: one-page web application SPA; React framework; Angular framework; Vue.js framework; AJAX technology.

УДК 004.9**Овчарук Ирина,***кандидат технических наук,**доцент кафедры информационных технологий и дизайна,**Государственный университет инфраструктуры и технологий,**Киев, Украина**ovch05@ukr.net**<https://orcid.org/0000-0003-4255-5816>***Горбенко Александр,***магистрант,**кафедра информационных технологий и дизайна,**Государственный университет инфраструктуры и технологий,**Киев Украина**jioji888@gmail.com**<https://orcid.org/0000-0001-8010-7294>***SINGLE PAGE APPLICATION ДЛЯ ПРИЛОЖЕНИЯ
ОТСЛЕЖИВАНИЯ ПОГОДНЫХ УСЛОВИЙ**

Целью статьи является обзор концепций для создания веб-приложений, анализ преимуществ и недостатков современных технологий для прогнозирования погодных условий. В статье представлены современные клиент-серверные технологии. Подробно рассмотрены концепция Single Page Application и другие технологии, позволяющие взаимодействовать между клиентом и сервером в режиме онлайн. Представлена разработка приложения для отслеживания погодных условий на основе архитектуры приложения SPA.

Методами исследования является обработка теоретической информации и проведение анализа архитектуры приложения, построенного по принципу SPA, проведение анализа преимуществ и недостатков подходов взаимодействия клиентской части с сервером, использование библиотек Angular, Vue, React, использование API для создания приложений.

Новизной проведенного исследования является анализ современных концепций и технологий, позволяющих разрабатывать веб-приложения, с помощью которых можно построить современное веб-приложение концепции Single Page Application, приведен подробный анализ библиотек Angular, Vue, React и взаимодействие пользователя с такими системами, анализ технологии Service Workers как способа хеширования.

Выводы. В статье проведен анализ современных технологий создания веб-приложений, приведены их преимущества и недостатки, рассмотрена концепция Single Page Application и описана разработка приложения для отслеживания погодных условий, приведены аналоги для сравнительной характеристики, рассмотрены проблемы, возникающие при взаимодействии клиентской части с сервером до применения принципов SPA.

Ключевые слова: одностраничное веб-приложение SPA; фреймворк React; фреймворк Angular; фреймворк Vue.js; технология AJAX.

18.10.2021