



EUROPEAN CONFERENCE

Conference Proceedings



**I International Science Conference
«New ways of creating scientific ideas
for implementation»**

September 18 – 20, 2023

Varna, Bulgaria

NEW WAYS OF CREATING SCIENTIFIC IDEAS FOR IMPLEMENTATION

Abstracts of I International Scientific and Practical Conference

Varna, Bulgaria
(September 18-20, 2023)

55.	Герасимчук О. РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ ЦУКРОВОГО ПЕЧИВА НА ОСНОВІ ГРЕЧАНОГО ТА КУКУРУДЗЯНОГО БОРОШНА	261
56.	Пікуль І. АНАЛІЗ СУЧАСНИХ АРХІТЕКТУРНИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ ВЕБЗАСТОСУНКІВ	264
57.	Стебаєв І. ДОСЛІДЖЕННЯ ВЕЛИКОМОВНОЇ МОДЕЛІ ДЛЯ ПЕРЕКЛАДУ УКРАЇНСЬКОЇ МОВИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	269
58.	Стебаєв Д. ДОСЛІДЖЕННЯ "АЛМАЗНОЇ МОДЕЛІ" ЩОДО ВРАХУВАННЯ ВИЗНАЧЕННЯ ЗВ'ЯЗКУ МІЖ МОТИВАЦІЄЮ ПРИ ЗДІЙСНЕННІ ХАКЕРОМ КІБЕРАТАКИ	273
59.	Тарасенко Д. ВИРІШЕННЯ ЗАДАЧІ ЗНАХОЖДЕННЯ СТАБІЛЬНИХ ВІДПОВІДНОСТЕЙ ЗА ДОПОМОГОЮ АЛГОРИТМУ ГЕЙЛА- ШЕПЛІ	277
60.	Шахматенко Д. ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ БЛОКЧЕЙНУ	281

АНАЛІЗ СУЧАСНИХ АРХІТЕКТУРНИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ ВЕБЗАСТОСУНКІВ

Пікуль Інна,
магістрант кафедри інформатики
Харківський національний університет радіоелектроніки,

Вибір вебархітектури є критичним етапом перед початком розробки вебзастосунку і може мати значний вплив на успіх проєкту. При виборі вебархітектури важливо ретельно аналізувати всі аспекти проєкту та враховувати його майбутні потреби. Важливо також провести вивчення і порівняння різних архітектур та підходів для забезпечення оптимального рішення для конкретного вебзастосунку [1-5].

Сучасні вебзастосунки використовують концепцію трирівневої архітектури, яка розділяє розробку вебзастосунку на такі рівні [1]:

- презентаційний рівень (presentation tier);
- рівень бізнес-логіки (application tier) [2];
- рівень даних (data tier) [3].

Зосередимося на вивченні типів архітектури презентаційного рівня.

Multi-Page Application для підтримки SEO [4].

Multi-Page Application (MPA) – це тип вебзастосунків, які складаються з декількох окремих сторінок, які взаємодіють між собою.

Переваги MPA полягають у наступних факторах.

Простота в розробці та обслуговуванні: MPA є простішими в розробці та обслуговуванні порівняно з SPA. Це пов'язано з тим, що кожна сторінка MPA є автономною і незалежною від інших сторінок.

Оптимізація для пошукових систем: MPA краще оптимізовані для пошукових систем у порівнянні з SPA. Кожна сторінка MPA має свій власний URL-адрес, який може бути проіндексований пошуковими системами.

Можливості для персоналізації: MPA надають більше можливостей для персоналізації користувачам. Кожна сторінка MPA може бути налаштована індивідуально для кожного користувача.

До недоліків MPA віднесемо наступне.

Повільне завантаження: MPA можуть бути повільнішими в порівнянні з SPA, оскільки кожному сторінку потрібно завантажувати окремо.

Більший обсяг даних: MPA вимагають завантаження більшого обсягу даних, оскільки кожна сторінка має свій власний HTML, CSS та JavaScript [4].

SPA для інтерактивної взаємодії [5].

Однією з ключових архітектурних моделей для створення клієнтської частини вебзастосунків є підхід SPA (Single Page Application). SPA дозволяють завантажувати одну стартову сторінку і динамічно оновлювати її вміст без перезавантаження. Для реалізації SPA часто використовують фреймворки, такі

як React, Angular або Vue.js, які надають структуру і інструменти для створення інтерактивних інтерфейсів.

Переваги полягають у наступних факторах.

Інтерактивність: SPA дозволяє створити інтерактивну вебпрограму, яка взаємодіє з користувачем без перезавантаження сторінки.

Масштабованість: Ця архітектура підходить для легкого масштабування продукту. Додавання нового функціоналу може бути досить простим завданням.

Спільний API: Якщо потрібен мобільний додаток, SPA дозволяє використовувати той самий API, що і вебверсія, що спрощує розробку і підтримку обох платформ.

Швидка візуалізація: SPA забезпечує швидку візуалізацію після повного завантаження програми у браузері, що створює високо реактивне програмне забезпечення для кінцевих користувачів.

До недоліків віднесемо наступне.

Погіршення SEO: SPA може погіршити SEO, оскільки реалізація функціоналу для оптимізації пошукових систем може вимагати додаткових зусиль.

Обмежена зв'язуваність: SPA може обмежити зв'язуваність між окремими сторінками, оскільки вони динамічно завантажуються без перезавантаження браузера.

Тривалий час завантаження: Інколи SPA може мати тривалий час завантаження при першому відкритті програми через необхідність завантажити весь код.

Погана маршрутизація: Неправильно налаштована маршрутизація може призвести до проблем у навігації користувачів.

Обмежена підтримка застарілих браузерів: Деякі застарілі браузери можуть не підтримувати повноцінно SPA.

Вища вартість розробки: SPA може бути дорожчим типом вебархітектури та, зазвичай, більше підходить для створення інтерфейсу для користувачів B2C (бізнес до споживача) [5].

PWA для покращення мобільної взаємодії [6].

Progressive Web Apps (PWA) поєднують в собі переваги вебсайтів і мобільних застосунків. Вони можуть працювати офлайн, надсилати сповіщення і забезпечувати високий рівень інтерактивності на мобільних пристроях. Використання PWA дозволяє покращити мобільний досвід користувачів.

Переваги полягають у наступних факторах.

Підтримка багатьох платформ: PWA підтримується на операційних системах Windows, Android та iOS (за умови, що офлайн-режим вимкнено для iOS).

Віддалені оновлення: Розробники можуть надсилати оновлення до вебпрограми віддалено, що дозволяє покращувати функціональність без необхідності переустановлення застосунку на пристрої користувачів.

Безпека: PWA використовує протокол HTTPS, що робить його безпечним для використання та передачі даних.

Легка інсталяція: Користувачі можуть встановити PWA без необхідності відвідувати магазини застосунків, такі як Play Market або App Store.

Основні недоліки такі.

Обмеження браузера та ОС: Для повного використання PWA необхідно вибирати браузери та операційні системи, які повністю підтримують цей тип архітектури [6].

Micro-frontends для створення складних вебзастосунків [7].

Micro-frontends – це архітектурний стиль, який дозволяє розробникам створювати вебзастосунки з використанням декількох незалежних фронтенд-технологій. Це може бути корисним для застосунків, які складаються з різних модулів та можуть розроблятися та підтримуватися різними командами. До таких можна віднести застосунки, що здійснюють аналіз та оброблення зорової інформації [8-20].

Переваги полягають у наступних факторах.

Зосередженість на модулях: micro-frontends дозволяють розробникам та командам зосередитися на окремому модулі або функції, спрощуючи розробку, тестування та підтримку великих застосунків.

Незалежне масштабування: Можливість масштабування кожного мікро-фронтенду незалежно від інших може бути корисною для застосунків, які потребують підтримки великого обсягу трафіку або мають вимоги до масштабування.

Легкість впровадження нового функціоналу: micro-frontends дозволяють додавати нові модулі або функції без необхідності змінювати весь додаток, що полегшує впровадження нового функціоналу та змін.

Основні недоліки такі.

Відмінності середовища: Розробка мікроінтерфейсу окремо від інших команд і його запуск на порожній сторінці під час розробки спрощує процес, але створює ризики. Відмінності між середовищами розробки та виробництва, особливо в застарілих кодових базах, можуть призвести до несправної або непослідовної поведінки.

Розмір корисного навантаження: Незалежно створені пакети JavaScript можуть призвести до дублювання загальних залежностей, збільшуючи дані, що надсилаються через мережу. Це може вплинути на продуктивність сторінки та залучення користувачів, особливо в регіонах з повільною інтернет-інфраструктурою.

Операційна та управлінська складність [7]: micro-frontends ускладнюють роботу, вимагаючи більше сховищ, інструментів, конвеєрів, серверів і доменів.

Залежно від конкретних потреб та вимог проєкту, може бути обрано відповідну архітектуру та підхід для створення вебзастосунку, який задовольняє користувачів та досягає поставлених цілей.

У кожному випадку важливо ретельно аналізувати і обирати ті технології, які найкраще підходять для конкретного проєкту.

Список літератури:

1. Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." *Open Information Systems* 10, 1 (January 1995).
2. Martin Fowler's Service Layer.
3. Microsoft Application Architecture Guide.
4. Ben Nadel (O'Reilly Media, 2019) "Multi-Page Applications: A Complete Guide to Building and Deploying Modern Web Applications".
5. John Papa "Single-Page Applications: Building Dynamic Web Apps with HTML, CSS, and JavaScript".
6. Ben Nadel "Progressive Web Apps: A Complete Guide to Building and Deploying Modern Web Applications".
7. martinowler.com. "Micro Frontends".
8. Гороховатський В.О., Пупченко Д.В., Солодченко К.Г. (2018) Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. *Системи управління, навігації та зв'язку*, С. 93-98.
9. Гороховатский В.А. Структурный анализ и интеллектуальная обработка данных в компьютерном зрении: монография, Комп. СМИТ, 2014. 316 с.
10. Gorokhovatskyi, V., Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. *Advanced Information Systems*, 5(4), pp. 10-16.
11. Гороховатский, В.А., Путятин, Е.П. (2008) Структурное распознавание изображений на основе моделей голосования признаков характерных точек. *Реєстрація, зберігання і обробка даних*, Т. 10, № 4. С.75-85.
12. Гороховатский, В.А., Путятин, Е.П., Столяров, В.С. (2017) Исследование результативности структурных методов классификации изображений с применением кластерной модели данных. *Радиоэлектроника, информатика, управление*, №3 (42). С. 78-85.
13. Gorokhovatskyi V., Gadetska S., Ponomarenko R. (2020) Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. Proc. of the XV Int. Scientific Conference "Intellectual Systems of Decision Making and Problems of Computational Intelligence" (ISDMCI'2019), Ukraine, May 21–25, 2019, pp. 501-512.
14. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
15. Gorokhovatskyi, V., Peredrii, O., Tvoroshenko, I., Markov, T. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень. *Advanced Information Systems*, 7(1), С. 5-13.
16. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In *CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022)*, 3137, pp. 68-77.

17. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., Vlasenko, N. (2023) Explanation of CNN Image Classifiers with Hiding Parts. In: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quenot (eds), *Explainable Deep Learning Artificial Intelligence*, pp. 125-146, Academic Press, 346 p.

18. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжнародний науковий симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці міжн. наук. симпозіуму (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

19. Творошенко, І. С., & Табашник, В. А. (2018). Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, (1), 122-128.

20. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., Vlasenko, N. (2023) Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.