

## ДОДАТОК А

Перелік джерел посилання за науковими напрямами керівника та науковців  
кафедри програмної інженерії

3. Перов О.С., Афанасьєва І.В. ФРЕЙМВОРК ДЛЯ РЕНДЕРІНГУ 3D СЦЕН НА ПЛАТФОРМАХ, ЩО ПІДТРИМУЮТЬ METAL API//Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Тези доповідей дванадцятої міжнародної науково-технічної конференції 27-28 квітня 2022 року. -Том2:секція 5. Баку-Харків-Жиліна-2022. -С. 117

7. Dudar Z., Shubin I., Skovorodnikova V., Litvin S. Research of Ways to Increase the Efficiency of Functioning Between Firewalls in the Protection of Information Web-Portals in Telecommunications Networks Lecture Notes in Networks and Systems, 2021, 212 LNNS, стр. 272–292 DOI 10.1007/978-3-030-76343-5\_14

8. Хомицкий И.А., Новіков Ю. С. Проблема обновления контента при использовании cdn// 20-Й Юбилейный Международный молодежный форум «Радиоэлектроника и молодежь в XXI ВЕКЕ».

12. Turevska O., Shubin, I. Improving the automated testing of Web-based services by reflecting the social habits of target audiences//2015 Information Technologies in Innovation Business Conference, ITIB 2015 - Proceedings, 2015, с. 93-96, 7355062

## ДОДАТОК Б

## Звіт результатів перевірки на плагіат



Ім'я користувача:  
Кардаш Євген Вікторович каф.ПІ

ID перевірки:  
1016380379

Дата перевірки:  
21.06.2024 12:01:50 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
21.06.2024 12:06:35 EEST

ID користувача:  
100013622

Назва документа: 2024\_М\_ПІ\_ІПЗм-22-1\_Васильєв\_Д\_О\_скорочений

Кількість сторінок: 44 Кількість слів: 8445 Кількість символів: 65528 Розмір файлу: 531.27 KB ID файлу: 1016189478

**3.51%**  
**Схожість**

Найбільша схожість: 2.63% з джерелом з Бібліотеки (ID файлу: 1016127780)

0.58% Джерела з Інтернету

91

Сторінка 46

3.13% Джерела з Бібліотеки

53

Сторінка 46

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

ДОДАТОК В  
Апробація результатів роботи



*Certificate*  
OF ATTENDANCE  
№ SH21C-0569  
Duration of the conference:  
15 hours / 0.5 ECTS credits

This certificate is awarded to:

**Denys VASYLIEV**

in recognition of participation in  
International scientific conference  
«Scientific Horizons of the XXI Century: Multidisciplinary Researches»  
which took place at Uzhhorod National University on May 16-17, 2024,  
in synchronous mixed mode (offline and online)

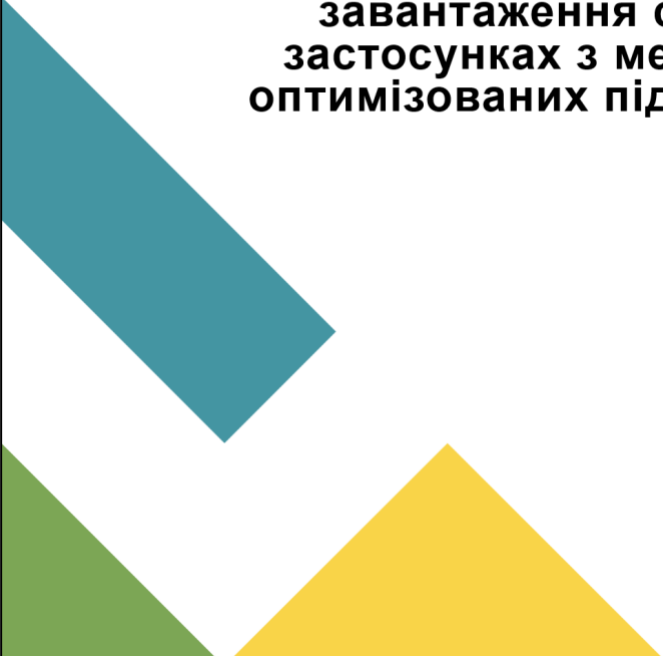
  
Volodymyr Smolanka  
Rector of Uzhhorod National University,  
Doctor of Sciences (Medicine),  
Professor, Head of Neurology,  
Neurosurgery and Psychiatry Department

  
Taras Griadil  
Head of Council Of Young Scientists of  
Uzhhorod National University



MATERIALS OF  
THE CONFERENCE

ДОДАТОК Г  
Слайди презентації




## Дослідження моделей та технологій завантаження сторінок у web-застосунках з метою визначення оптимізованих підходів рендерингу

Виконав:  
ст. гр. ІПЗм-22-1 Васильєв Д.О.

Керівник:  
Доц. каф. ПІ Афанасьєва І.В.

Слайд Д.1 – Слайд 1



## Актуальність дослідження

- React є найбільш популярною бібліотекою для створення веб-застосунків в світі на мові програмування JavaScript.
- Next.js в свою чергу є найбільш популярною бібліотекою для самого React, доповнюючи його функціональність та можливості. Тож, зв'язка цих технологій широко використовується при створенні проєктів різної складності.
- Дослідження швидкості рендерингу та вибору його концепції може допомогти обрати найбільш відповідний та корисний варіант для задач бізнесу.

2

Слайд Д.2 – Слайд 2

## Аналіз предметної галузі

- Обираючи вид рендерингу, ми визначаємо, як саме сторінки будуть відображатися для користувачів та як вони будуть генеруватися на сервері чи клієнті. Це має великий вплив на продуктивність, швидкодію, індексацію пошуковими системами, безпеку та інші аспекти веб-додатку.
- Вибір підходу до рендерингу (CSR, SSR, SSG тощо) визначається різними факторами, такими як вимоги бізнесу, потреби користувачів, характеристики проекту та його масштаби, а також стратегії SEO.

3

Слайд Д.3 – Слайд 3

## Постановка задачі

- Провести аналіз різних методів рендерингу.
- Визначити, який метод є кращим в залежності від задач.
- Проаналізувати сучасні технології для створення веб-застосунків, такі як React та Next.js.
- Провести експериментальне дослідження та визначити який метод рендерингу є швидшим за інші.

4

Слайд Д.4 – Слайд 4

## Роль React та Next.js в SSR

- Автоматичний SSR: Next.js автоматично рендерить сторінки на сервері, якщо це необхідно. Він визначає, які сторінки повинні бути відрендерені на сервері, а які можна рендерити на клієнті.
- Статична генерація: Окрім SSR, Next.js підтримує статичну генерацію сторінок (SSG), що дозволяє попередньо створювати сторінки під час збірки застосунку.
- React надає базову інфраструктуру для створення компонентів, які можуть бути відрендерені на сервері та гідровані на клієнті. Next.js, у свою чергу, доповнює React, пропонуючи потужні інструменти для автоматизації та оптимізації процесу серверного рендерингу, спрощуючи розробку високопродуктивних та SEO-оптимізованих веб-застосунків.

5

Слайд Д.5 – Слайд 5

## Різниця між видами рендерингу

- Рендеринг на стороні клієнта (CSR) відбувається у браузері, після отримання сторінки, що дозволяє побудувати динамічні інтерфейси, але може призвести до більшого часу завантаження сторінки.
- Рендеринг на стороні сервера (SSR) генерує HTML на сервері перед відправленням його клієнту, що поліпшує швидкість завантаження сторінок та забезпечує кращу індексацію пошуковими системами.
- Рендеринг на стороні сервера (SSG) генерує статичні файли під час збірки, що дозволяє швидко завантажувати сторінки та знижує навантаження на сервер.

6

Слайд Д.6 – Слайд 6

## Метод рендерингу в залежності від проєкту

- Прості проєкти, такі як односторінкові інтернет-магазини або особисті блоги, часто використовують **CSR (Client-Side Rendering)**. Цей підхід дозволяє створювати динамічний та інтерактивний вміст без перезавантаження сторінки, що дуже зручно для користувачів, особливо при взаємодії з продуктами або контентом.
- Більш складні проєкти, такі як фінансові платформи або біржі акцій, найчастіше використовують **SSR (Server-Side Rendering)**. Цей метод забезпечує швидке завантаження сторінок та забезпечує кращу безпеку, оскільки HTML генерується на сервері перед відправкою клієнту. Це особливо важливо для фінансових додатків, де точність та безпека даних є пріоритетними.
- Для статичних сайтів або лендінгових сторінок, які не потребують динамічної зміни вмісту, ідеально підходить **SSG (Static Site Generation)**. Цей метод дозволяє швидко завантажувати сторінки та знижує навантаження на сервер, що особливо важливо для сайтів з великим обсягом відвідувачів.

7

Слайд Д.7 – Слайд 7

## Розробка веб-сайту з використанням сучасних технологій

React - це бібліотека JavaScript, створена компанією Facebook. Вона призначена для створення інтерфейсів користувача, які відображаються на веб-сторінках.

Основна ідея React полягає в тому, щоб розбити інтерфейс користувача на невеликі компоненти, які можна перевикористовувати та легко управляти.

React використовує концепцію віртуального DOM (Document Object Model), що дозволяє ефективно оновлювати тільки ті частини сторінки, які змінилися, замість повного перерендерингу всього дерева DOM.

Next.js - це фреймворк для розробки веб-додатків на основі React.

Він надає розширений набір функцій та інструментів, що полегшують створення як статичних, так і динамічних веб-додатків.

Next.js додає до React додаткові можливості, такі як попереднє рендеринг (SSR та SSG), маршрутизація на основі файлів, оптимізоване завантаження сторінок та інші.

8

Слайд Д.8 – Слайд 8

# Приклади програмної реалізації компонентів з різними видами рендерингу

```
export default function CSR() {
  const [serverTime, setServerTime] = useState("");

  useEffect(() => {
    const fetchServerTime = async () => {
      const response = await fetch("/api/time");
      const data = await response.json();
      setServerTime(data.serverTime);
    };

    fetchServerTime();
  }, []);

  return (
    <div>
      <Head>
        <title>Create Next App</title>
        <meta name="description" content="Generated by create next app" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="icon" href="/favicon.ico" />
      </Head>
      <main className={` ${styles.main} ${inter.className}`} >
        <div className={styles.description}>
          <p>
            This is CSR page
          </p>
        </div>
      </main>
    </div>
  );
}
```

```
export async function getServerSideProps() {
  // fetch server time
  const serverTime = new Date().toTimeString();

  return {
    props: {
      serverTime,
    },
  };
}

export default function Home({ serverTime }) {
  return (
    <div>
      <Head>
        <title>Create Next App</title>
        <meta name="description" content="Generated by create next app" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="icon" href="/favicon.ico" />
      </Head>
      <main className={` ${styles.main} ${inter.className}`} >
        <div className={styles.description}>
          <p>
            This is SSR page
          </p>
        </div>
      </main>
    </div>
  );
}
```

9

Слайд Д.9 – Слайд 9

# Результат швидкості рендерингу клієнтського компонента (CSR)

Name	Status	Type	Initiator	Size	Time
next.svg	304	svg+xml	ssg:0	242 B	6 ms
react-refresh.js	200	script	ssg:0	25.2 kB	11 ms
webpack.js	200	script	ssg:0	9.5 kB	13 ms
main.js	200	script	ssg:0	1.1 MB	292 ms
_app.js	200	script	ssg:0	53.1 kB	26 ms
ssg.js	200	script	ssg:0	76.8 kB	38 ms
_buildManifest.js	200	script	ssg:0	698 B	14 ms
_ssgManifest.js	200	script	ssg:0	411 B	14 ms
_devMiddlewareManifest.json	200	fetch	main.js:809	213 B	2 ms
webpack-hmr	101	websocket	main.js:809	0 B	Pending
favicon.ico	304	x-icon	Other	243 B	4 ms

14 requests | 1.3 MB transferred | 5.7 MB resources | Finish: 736 ms | DOMContentLoaded: 662 ms | Load: 686 ms

10

Слайд Д.10 – Слайд 10



## Результат швидкості рендерингу серверного компоненту (SSR)

Name	Status	Type	Initiator	Size	Time
vercel.svg	304	svg+xml	index:0	242 B	5 ms
next.svg	304	svg+xml	index:0	242 B	5 ms
react-refresh.js	200	script	index:0	25.2 kB	11 ms
webpack.js	200	script	index:0	9.5 kB	9 ms
main.js	200	script	index:0	1.1 MB	201 ms
_app.js	200	script	index:0	53.1 kB	22 ms
index.js	200	script	index:0	76.4 kB	28 ms
_buildManifest.js	200	script	index:0	698 B	7 ms
_ssgManifest.js	200	script	index:0	411 B	7 ms
_devMiddlewareManifest.json	200	fetch	main.js:809	213 B	2 ms
webpack-hmr	101	websocket	main.js:809	0 B	Pending

13 requests | 1.3 MB transferred | 5.6 MB resources | Finish: 499 ms | DOMContentLoaded: 498 ms | Load: 518 ms

Console | What's new X

11

Слайд Д.11 – Слайд 11

## Опис сформованих рекомендацій

- Метод рендерингу необхідно обирати в залежності від потреб та можливостей бізнесу.
- Сучасні технології React та Next.js суттєво спрощують розробку веб-застосунків та додають значну кількість функціонала.
- Бібліотека Next.js гарно підходить як для серверного, так і для клієнтського рендерингу.
- Метод серверного рендерингу швидше, за клієнтський.

12

Слайд Д.12 – Слайд 12

## Висновки

- Проведено аналіз серверного та клієнтських видів рендерингу.
- Проведено дослідження в ході якого був зроблений висновок, що серверний вид рендерингу є швидшим за клієнтський.
- На основі проведених експериментів були сформовані рекомендації про вибір технологій та рендерингу.
- Подано тези доповіді на двадцять восьмий міжнародний форум «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ».

13

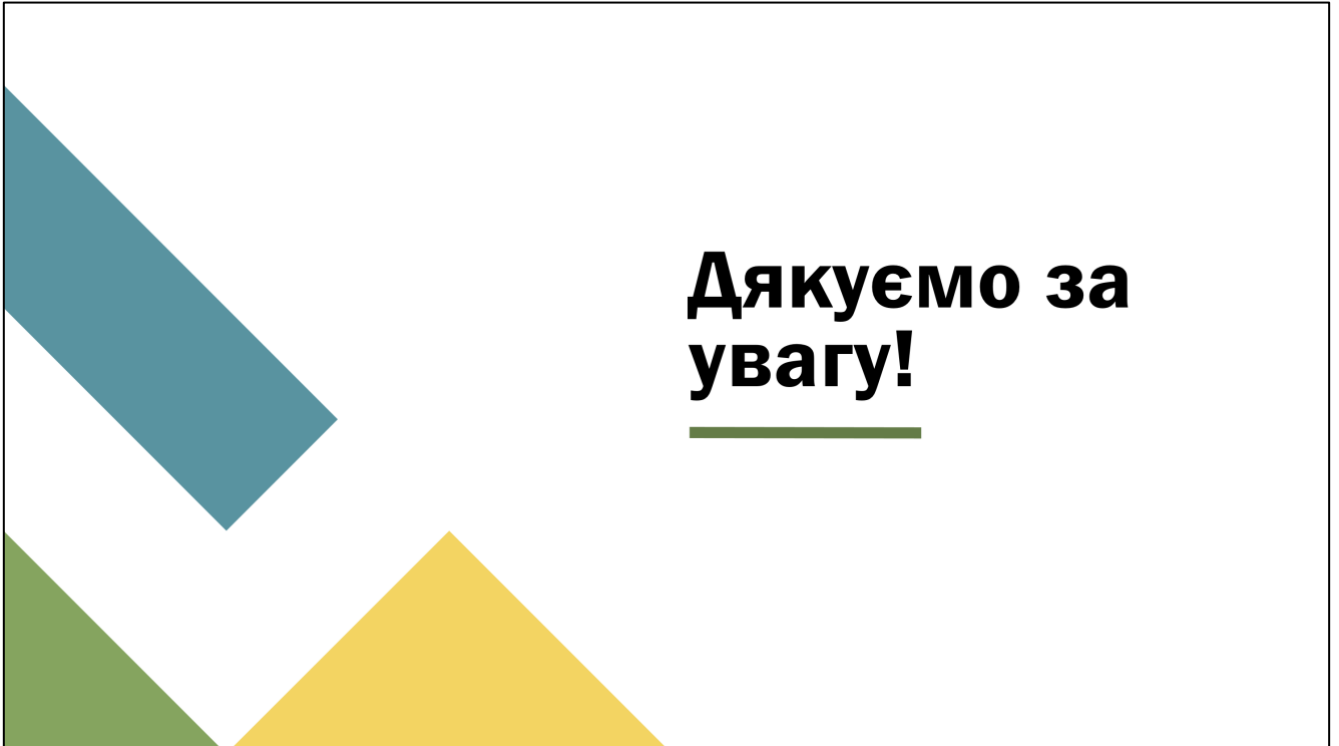
Слайд Д.13 – Слайд 13

## Підсумки

На основі отриманих даних було визначено, що інтеграція SSG та SSR може бути оптимальним рішенням для додатків, що вимагають гнучкості в обслуговуванні великої кількості користувачів із різними потребами. Такий гібридний підхід дозволяє максимально використовувати переваги обох методів, оптимізуючи загальну продуктивність та задовольняючи специфічні вимоги проектів.

14

Слайд Д.14 – Слайд 14



Слайд Д.15 – Слайд 15

# ДОДАТОК Д

## Експертний висновок результатів перевірки на відповідність оформлення вимогам ДСТУ

1

Експертний висновок результатів перевірки кваліфікаційної роботи

студент  
(посада)

програмної інженерії  
(кафедра)

ППЗМ-22-1  
(група)

Васильєв Д.О.

( прізвище, ім'я, по батькові )

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	<b>7.1 Загальні положення</b>	
	<b>7.3 Нумерація сторінок звіту</b>	
	<b>7.5 Рисунки</b>	
	<b>7.6 Таблиці</b>	
	<b>7.7 Переліки</b>	
	<b>7.8 Примітки</b>	
	<b>7.9 Виноски</b>	
	<b>7.10 Формули та рівняння</b>	
	<b>7.11 Посилання</b>	
	<b>7.14 Скорочення та умовні позначки</b>	
	<b>7.15 Додатки</b>	
<p>Методичні вказівки до виконання кваліфікаційної роботи магістра... <b>ЗАТВЕРДЖЕНО</b> кафедрою ПІ протокол № 5 від 13.11.2023р. 3.2 Оформлення пояснювальної записки згідно з ДСТУ 3008:2015 Звіти у сфері науки і техніки. Структура та правила оформлення. <b>Шаблон</b> затверджений засіданням кафедри №3 від 16.10.2023.</p>	<p>Рисунок повинен розміщуватися одразу після його згадування у тексті, або на наступній сторінці. Під рисунком повинен бути підпис із словом Рисунок, порядковим номером цього рисунку, через тире з великої літери – назва рисунку та <b>в круглих дужках вказується джерело з якого взятий цей рисунок, або то, що його виконано самостійно.</b></p>	30, далі за текстом.
<p>Методичні вказівки до виконання кваліфікаційної роботи магістра... <b>ЗАТВЕРДЖЕНО</b> кафедрою ПІ протокол № 5 від 13.11.2023р. 3.2 Оформлення пояснювальної записки згідно з ДСТУ 3008:2015 Звіти у сфері науки і техніки. Структура та правила оформлення. <b>Шаблон</b> затверджений засіданням кафедри №3 від 16.10.2023.</p>	<p>Кількість сторінок (рисуноків, таблиць, джерел) заявлених в рефераті повинна співпадати з кількістю сторінок (рисуноків, таблиць, джерел) в записці.</p>	4

Експерт

(підпис)

Вадим НЕЧВОЛОД

(прізвище, ініціали)

23.06.2024