Дипломна робота на тему:

Розробка фреймворку для побудови веб-додатків на основі технології Web Components

Виконав: студент Михед Євгеній

Наук. керівник: Мамеко М.С.

Мета роботи

Дослідити технологію Web Components та фреймворки на базі даної технології. На основі досліджень розробити фреймворк для побудови веб-додатків.

Цілі роботи

- Аналіз технології та інструментів для створення фреймворку
- Дослідження існуючих аналогів та визначення функціоналу на базі результатів дослідження
- Реалізація фреймворку для побудови веб-додатків на основі технології Web Components

Преваги та недоліки технології Web Components

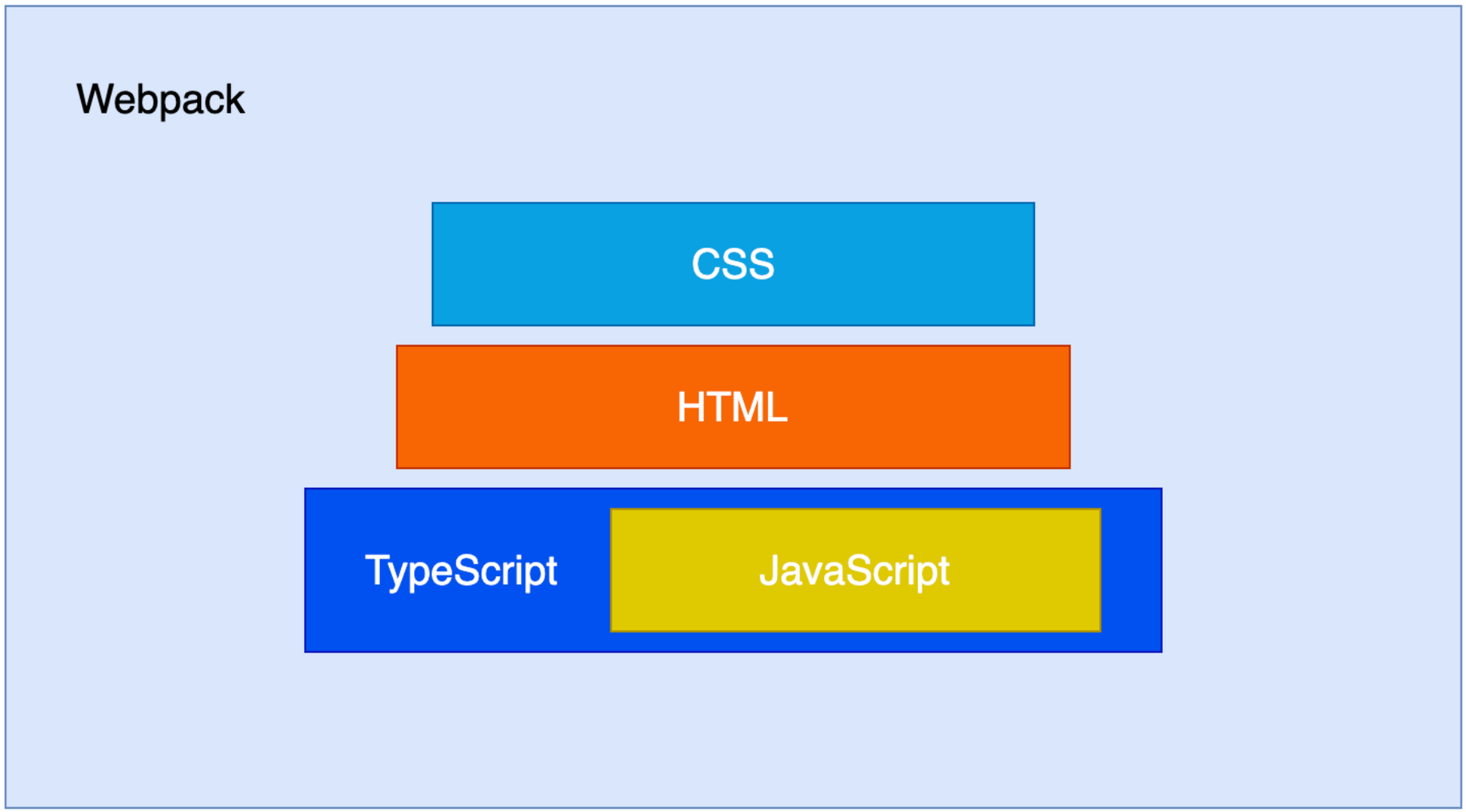
Переваги

- Компонентний підхід
- Можливість використовувати компоненти повторно
- Автономність технології
- Інкапсуляція компонента від загального документу
- Можливість використання технології з іншими фреймворками

Недоліки

- При великій кількості компонентів на сторінці знижується швидкість оновлення DOM
- Технологія не підходить для розробки веб-сторінок, які потребують SEO просування

Інструменти для побудови фреймворку



Порівняння TypeScript та JavaScript

Параметри	TypeScript	JavaScript
Типізація	Статична	Динамічна
Прив’язка даних	TypeScript використовує такі поняття, як типи і інтерфейси, для опису даних	В JavaScript немає такої функції
Використання звичного ООП	Так	Тільки з ES6
Компіляція коду	Потрібна	Не потрібна
Наявність декораторів	Так	Ні

Дослідження існуючих аналогів

Параметри	Polymer	Smart HTML Element	Slim.js	LitElements
Створення компоненту	За допомогою класу	За допомогою класу	За допомогою класу	За допомогою класу
Використання TypeScript	Так	Ні	Так	Так
Використання декораторів	Так	Ні	Так	Так
Стілізація компоненту	Shadow DOM правила стилізації	Shadow DOM правила стилізації	Shadow DOM правила стилізації	Shadow DOM правила стилізації
Наявність роутингу	Так	Ні	Так	Так

Перелік основних вимог для побудови фреймворку

- Для створення фреймворку використовувати JavaScript(TypeScript), HTML, CSS
- Для створення компонентів використовувати класи
- Використовувати Shadow DOM
- Реалізувати функціонал роутингу
- Використовувати декоратори для опису компонентів

Створення та використання компонента

Створення

```
import { Component } from '../framework/index'

@Component({
    name: "my-component",
    styles: `
        .title {
            color: red;
            margin: auto;
            width: fit-content;
        }
    `,
    template: `
    <h1 class="title">
        My component!
    </h1>`
})
export class MyComponent {};
```

Використання

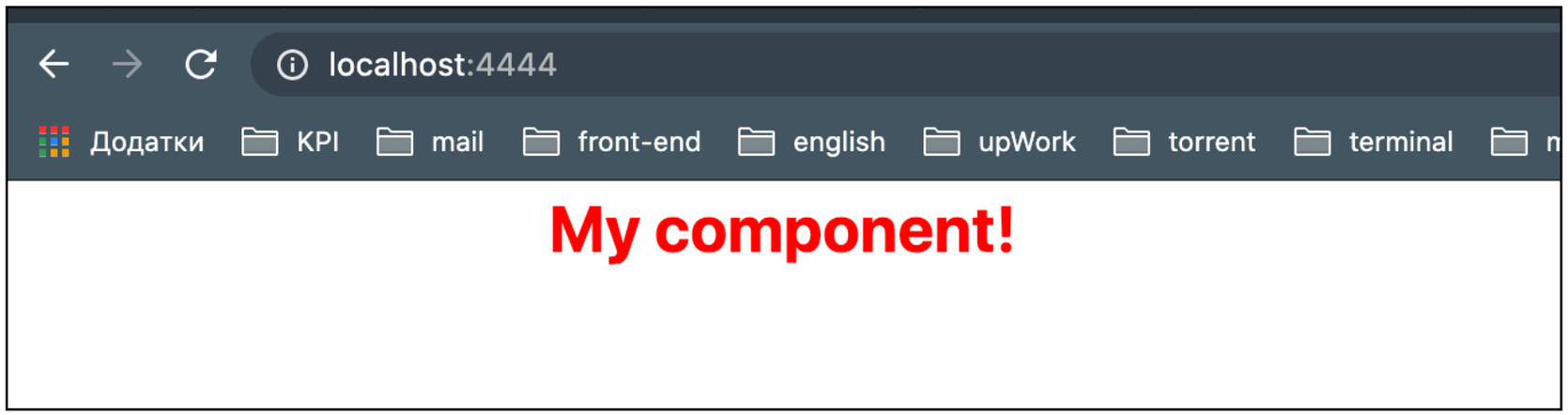
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
</head>
<body>
    <my-component />
</body>
</html>
```

Або

```
@Component({
    name: "my-home",
    styles: `
        .container {
            color: brown
        }
    `,
    template: `
    <div class="container">
        Home Page
        <my-component />
    </div>`
})
class Home {};
```

Рендер компонента

Рендер компонента на сторінці:



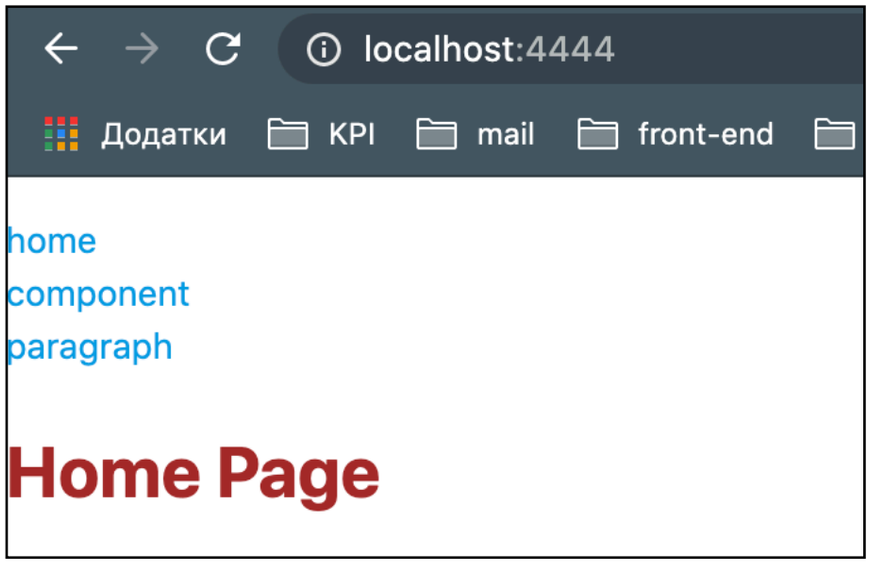
Вигляд розмітки в інструментах браузеру DevTools:

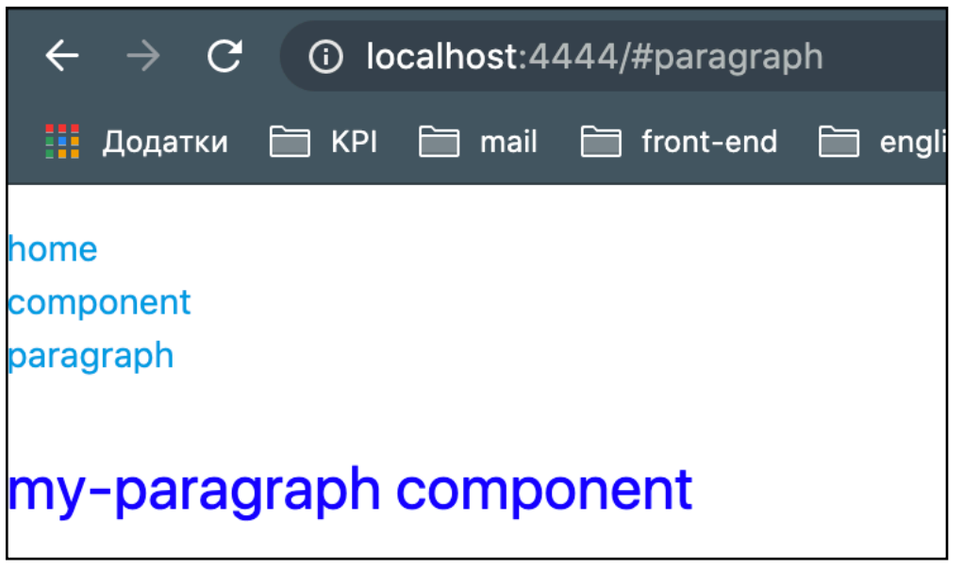
```
<!DOCTYPE html>
...<html lang="en"> == $0
  ▶ <head>…</head>
  ▼ <body class="vsc-initialized" cz-shortcut-listen="true">
    ▼ <my-component>
      ▼ #shadow-root (closed)
        ▼ <style>
                    .title {
                        color: red;
                        margin: auto;
                        width: fit-content;
                    }

          </style>
          <h1 class="title">
                  My component!
              </h1>
        <script type="text/javascript" src="bundle.js"></script>
      </my-component>
    </body>
  </html>
```

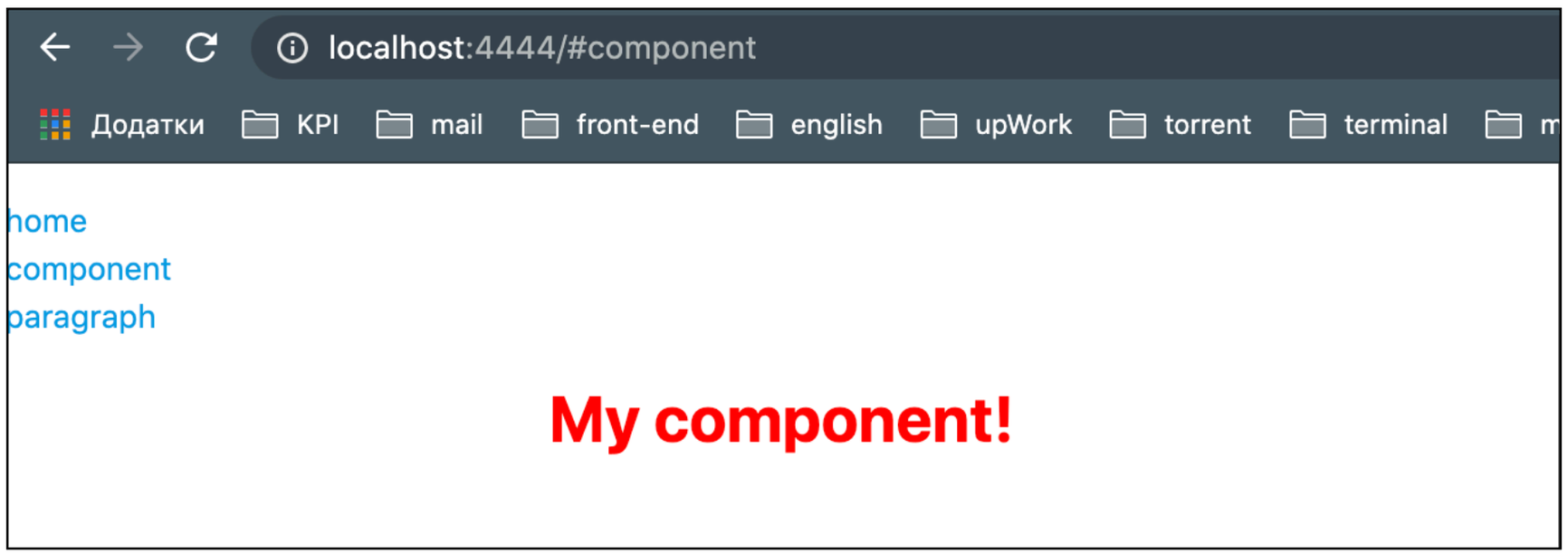
Використання роутера

```
<body>
    <ul>
        <li><a href="#">home</a></li>
        <li><a href="#comp">comp</a></li>
        <li><a href="#paragraph">paragraph</a></li>
    </ul>
    <wc-router>
        <wc-route path='' component='my-home'></wc-route>
        <wc-route path='comp' component='my-component'></wc-route>
        <wc-route path='paragraph' component='my-paragraph'></wc-route>
    </wc-router>
</body>
```

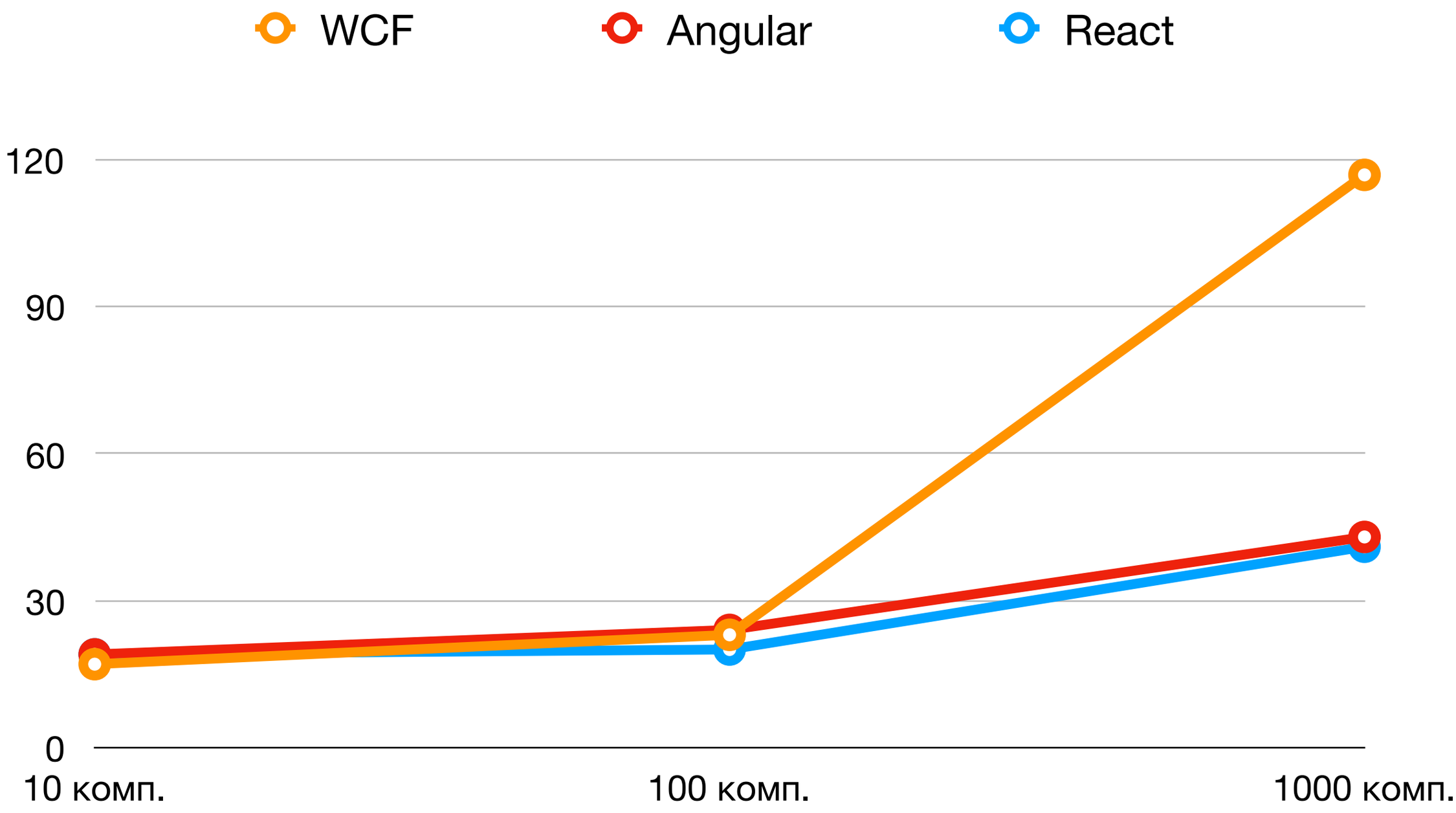
Рендер компонента по роуту







Порівняння швидкості рендеру власного фреймворку з Angular та React



Що можна додати в майбутньому?

- Дати можливість додавати фреймворк до проекту за допомогою пакетних менеджерів
- Створити набір готових компонентів
- Можливість змінювати стилі готових компонентів
- Додати функціонал тестування компонентів
- Оптимізувати рендер компонентів

Висновки

- Проаналізовано технологію Web Components та фреймворки побудовані на даній технології
- Визначено та реалізавоно функціонал фреймворку
- Наведено приклад використання фреймворку
- Порівняно продуктивність роботи власного фреймворку з Angular та React

Дякую за увагу!