
Розробка спеціалізованого фреймворку для серверного рендерингу React-додатків з оптимізацією продуктивності

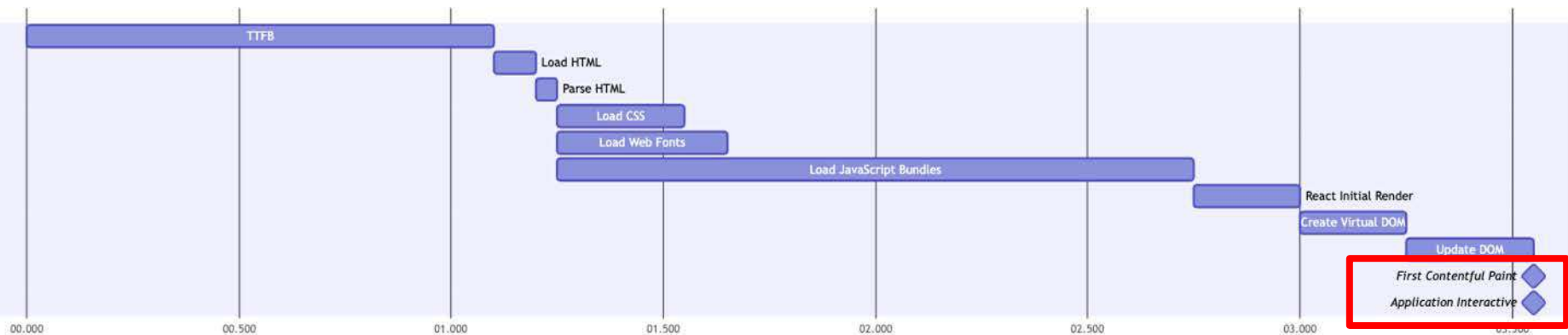
Підготував: Войлов Богдан Станіславович, МП ІПЗ-2

Науковий керівник: Афонін А.О.

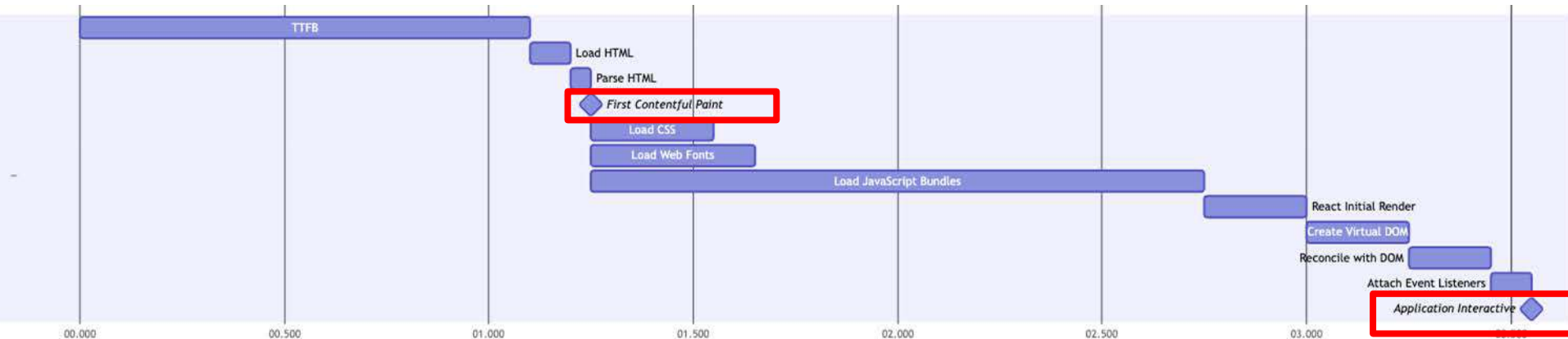
Мета роботи та завдання

Основною метою цієї роботи є створення прототипу фреймворку для розробки фронтенду веб додатків за допомогою бібліотеки React з оптимізацією завантаження сторінок.

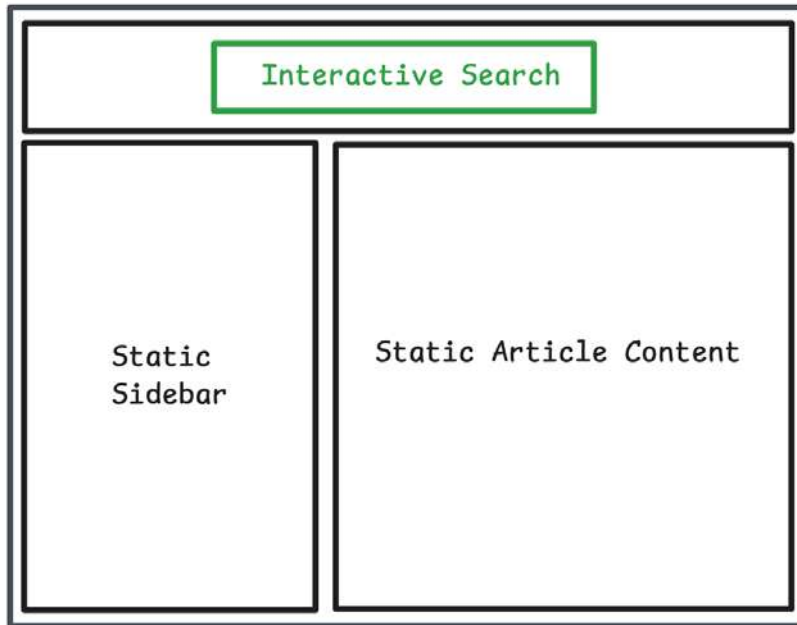
Завантаження сторінки класичного React веб додатку



Завантаження сторінки React веб додатку з SSR



Більшість веб додатків мають лише кілька динамічних елементів UI



Огляд існуючих фреймворків



Next.js



Remix



Astro

Функціональні вимоги до фреймворку

1. SSR
 2. Islands
 3. Component level asynchrony during SSR
 4. Cross island communication
 5. Client-side navigation
-

Архітектура фреймворку

1. Page router
 2. Island mount manager
 3. Island code bundler
-

Page Router

Сюди додам sequence діаграму взаємодії серверної та клієнтської частини роутера

Island mount manager

Додам flowchart/block діаграму що ілюструє як воно працює

Статичні частини в островах

```
<Typography variant="body2">Note</Typography>
<Dynamic>
  <CardActions>
    <Button size="small">Learn More</Button>
    <IconButton aria-label="delete" size="small" onClick={props.onDelete}>
      <DeleteIcon fontSize="inherit" />
    </IconButton>
  </CardActions>
</Dynamic>
```

Залежності/ Стек

1. NodeJS
2. Bun FS Router
3. Vite (Rollup)
4. Typescript
5. React

(додам іконки)

Висновки

Чи варто додати лістингів коду?

Можливо зменшити вступну частину якщо і так зрозуміло все?
