Метод уніфікованих авторських елементів інтерфейсу веб-додатків

Виконав: студент гр. 1КІ-16м Філіпчук В.С.

Науковий керівник: к.т.н., доц. Савицька Л.А.

Предмет та мета роботи

- **Предметом** дослідження є методи і засоби створення авторських веб-компонентів та побудови програмного забезпечення, що їх реалізує.
- **Метою роботи** є розробка методу уніфікованих авторських веб-елементів з можливістю зменшення складності за рахунок ізоляції пов'язаних груп коду для виконання загальної функціональності.

Актуальність роботи

- Код, який можна повторно використовувати в різних проектах, є основою сучасних веб-технологій. Використання сучасних засобів розробки веб-інтерфейсів користувача (front-end frameworks) і модулів для спрощення файлів каскадних таблиць стилів (Sass-бібліотек) істотно прискорює робочий процес розробки веб-додатків, та дозволяє швидше знайти рішення у порівнянні з традиційним підходом. Однак багаторазові мобільні елементи все ще знаходяться в стадії розробки.

Основні проблеми при розробці інтерфейсу веб-додатку

- Пересічення CSS-стилів;
- Пересічення областей видимості JavaScript-коду;
- Складна для розуміння та заповнена зайвими елементами HTML-розмітка;
- Збільшення складності розробки при зростанні розміру проекту.

Огляд існуючих рішень

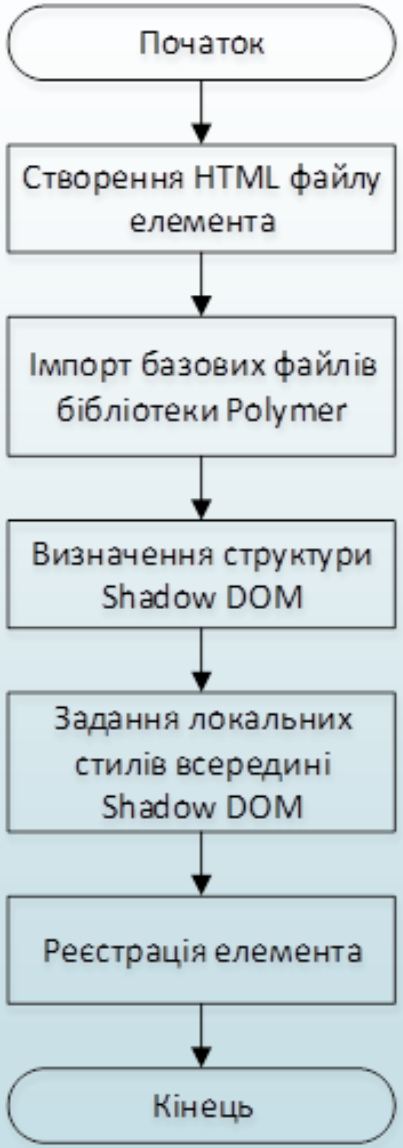
React.js

Angular

Vue.js

- **Custom Elements** – технологія, що дозволяє визначати нові HTML елементи.
- **Shadow DOM** – технологія для інкапсуляції DOM та стилів.
- **HTML Templates** – дозволяє документам містити інертні (не включені в сторінку) частини DOM.
- **HTML Imports** – декларативні методи імпортування HTML документів в інші документи.

Алгоритм методу



Шаблон авторського елементу

```
<link rel="import" href="bower_components/polymer/polymer-element.html">

<dom-module id="my-web-component">
  <template>
    <style>
      //Задання стилів елементу
    </style>
    //Задання структури елементу
    <p>I'm in the Shadow Root!</p>
  </template>

  <script>
    //Задання javascript-коду елементу та його реєстрація
    class MyWebComponent extends Polymer.Element {
      static get is() { return 'my-web-component' }
    customElements.define(MyWebComponent.is, MyWebComponent);
  </script>
</dom-module>
```

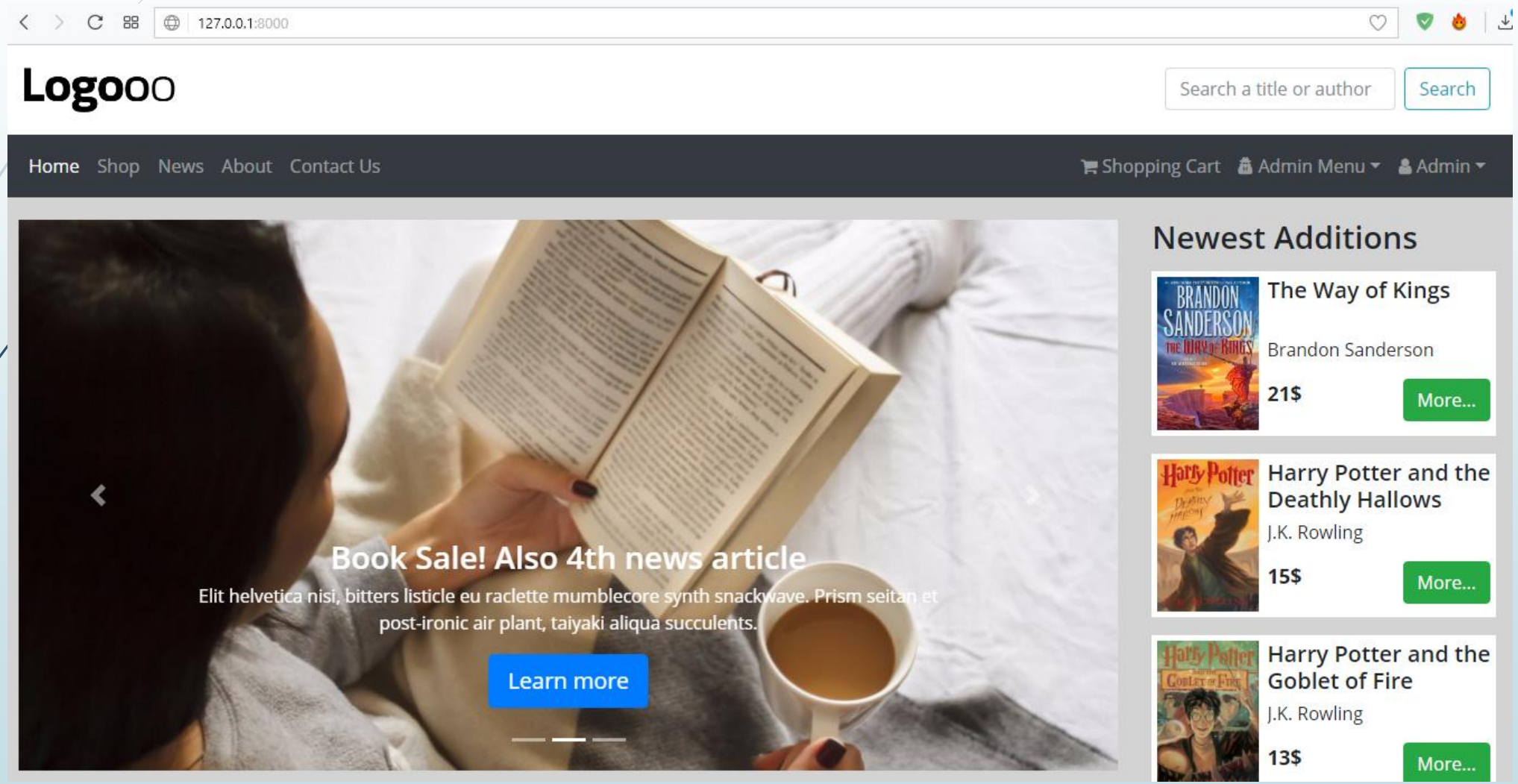
Вигляд коду у браузері

```
<!DOCTYPE html>
<html lang="en">
▶<head>…</head>
▼<body>
  ▼<my-web-component> == $0
    ▼#shadow-root (open)
      <p>I'm in the Shadow Root!</p>
    </my-web-component>
  </body>
</html>
```

Порівняння розробленого методу з існуючими засобами розробки інтерфейсу веб-додатків

Технологія	Компонентний підхід	Уніфікованість	Приховування внутрішньої розмітки компоненту	Інкапсуляція стилів
React.js	Так	Ні	Ні	Частково
Angular	Так	Ні	Ні	Частково
Vue.js	Так	Ні	Ні	Частково
Метод	Так	Так	Так	Так

Для демонстрації методу було розроблено веб-додаток типу інтернет-магазин, інтерфейс якого було розроблено з використанням запропонованого методу.



Публікації

- Савицька Л. А. Метод розробки інкапсульованих авторських компонентів веб-інтерфейсу / Л. А. Савицька, Т. І. Трояновська, В. С. Філіпчук // Наукові підсумки 2017 року, XV Міжнародна науково-практична інтернет-конференція. – м. Вінниця, 15 грудня 2017 року. – Ч.6, С. 69-72с.
- Трояновська Т. І. Метод уніфікованих авторських елементів інтерфейсу веб-додатків / Т. І. Трояновська, В. С. Філіпчук // «Інформаційні технології та взаємодії» : IV Міжнародна науково-практична конференція, 8-10 листопада 2017 р., м. Київ. – С. 135-136.

Висновок

Запропонований метод за рахунок використання компонентного підходу дозволяє створювати елементи користувацького інтерфейсу веб-додатків з можливістю вирішення проблем інкапсуляції стилів оформлення та внутрішнього функціоналу елементів за допомогою приховування внутрішньої структури одних елементів від інших, а також покращення структури та зрозумілості веб-сторінок. Крім того, такий підхід дозволяє спростити складність коду при розробці великих веб-додатків за рахунок можливості повторного використання розроблених елементів.

Дякую за увагу!