

УДК 004.4'236

РОЗРОБКА ІНТЕРАКТИВНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА ДЛЯ ВЕБ-ПРОГРАМУВАННЯ НА ОСНОВІ JAVASCRIPT-ФРЕЙМВОРКУ VUE.JS

Пузікова Тетяна

Центральноукраїнський державний педагогічний університет імені

Володимира Винниченка, м. Кропивницький, Україна

У статті описано досвід розробки навчального інтерактивного веб-додатку з використанням фреймворку Vue.js. Особливістю представленого веб-додатку є реалізація такої спеціальної задачі, як можливість інтерпретації і виконання програмного коду, введеного користувачем (студентом), безпосередньо у навчальному середовищі. Іншою зручною особливістю є відображення структури навчального проекту, тобто CSS-стилів, HTML-розмітки і Javascript-коду в одному редакторі. Також реалізовані такі стандартні для навчального веб-додатку задачі, як відображення довідки та навчальних завдань.

Ключові слова: Javascript-фреймворки, Vue.js, інтерактивне навчальне середовище, веб-програмування, Node.js, Codemirror, онлайн-редактор коду.

DEVELOPMENT OF INTERACTIVE EDUCATIONAL ENVIRONMENT FOR WEB-PROGRAMMING BASED ON JAVASCRIPT-FRAMEWORK VUE.JS

Puzikova Tetiana

Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,

Kropyvnytsky, Ukraine

The article describes the experience of developing an interactive Web-based educational application using the Vue.js framework. The peculiarity of the submitted web application is the realization of such a special task as the ability to interpret and execute the code introduced by the user (student) directly in the learning environment. Another convenient feature is to display the structure of the learning project, that is, CSS-styles, HTML markup and Javascript-code in one editor. Also implemented are standard tasks for the educational Web application, such as the display of help and training tasks.

Keywords: Javascript-frameworks, Vue.js, Interactive Learning Environment, Web Programming, Node.js, Codemirror, Online Code Editor.

Сфера інформаційних технологій вимагає від ІТ-спеціалістів та студентів постійного підвищення власного професійного рівня. Час та зусилля, витрачені

на навчання, залежать безпосередньо від якості відповідного інтерактивного середовища, у разі, якщо навчання відбувається онлайн.

Більшість інтерактивних навчальних середовищ для веб-програмування не відображають структуру проекту, що не є зручним для новачків. До того ж, новий та популярний на даний час фреймворк Vue.js не має довідника українською мовою. Таким чином, розробка інтерактивного навчального середовища для веб-програмування на основі Javascript-фреймворку Vue.js, який би містив такі чітко визначені блоки, як теоретичні відомості, завдання, приклад використання коду та відображення структури проекту з можливістю використання даного коду безпосередньо у середовищі, є актуальною задачею.

Для реалізації інтерактивного навчального середовища з використанням Vue.js було використано таке програмне забезпечення:

- Node.js – програмна платформа, що здійснює трансляцію JavaScript в машинний код та перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js надає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API, підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду [2].
- Visual Studio Code – редактор вихідного коду, що підтримує ряд мов програмування, підсвічування синтаксису, IntelliSense, рефакторинг, налагодження, навігацію по коду, підтримку Git та інші можливості [3].
- CodeMirror – універсальний текстовий редактор, реалізований мовою JavaScript для браузера. Він спеціалізується на редагуванні коду, і поставляється з низкою мовних режимів і додатків, які реалізують більш розширені функціональні можливості редагування [4].
- Vue CLI – повноцінна система для швидкої розробки на Vue.js, що надає:
 - можливість інтерактивного створення проекту через @vue/cli;
 - швидке прототипування через @vue/cli+ @vue/cli-service-global без зміни;

- повноцінний графічний користувальницький інтерфейс для створення і управління проектами Vue.js [1].

Створене середовище розробки веб-інтерфейсів складається з таких частин:

- Теоретичні відомості. Приклад зовнішнього вигляду блоку наведено на рисунку 1.

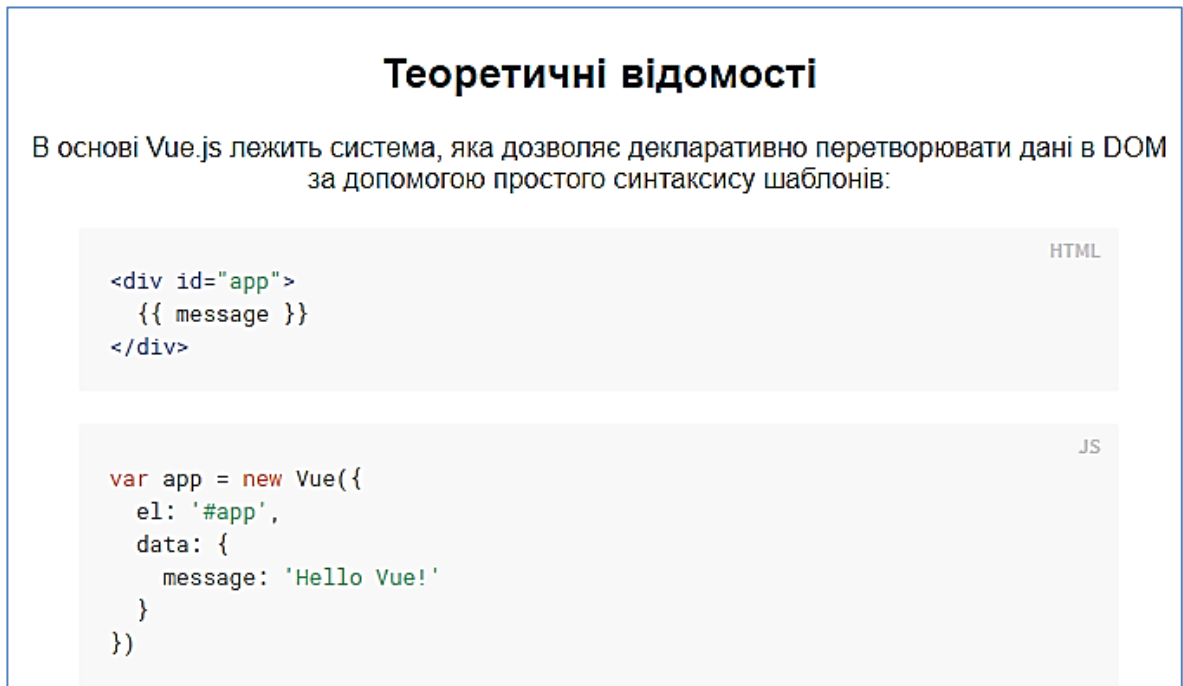


Рис. 1. Приклад зовнішнього вигляду блоку «Теоретичні відомості»

- Завдання для виконання. Приклад зовнішнього вигляду блоку наведено на рисунку 2.
- Інтерактивний онлайн редактор коду з базовими тегамі, що показують структуру проекту.
- Для досягнення інтерактивності середовища була створена функція `render` для обробки коду, який буде писати користувач у вікні редактора; виклик даної функції здійснюється при кожній зміні коду у вікні редактора.

Завдання №1

Вивести на екран повідомлення "Привіт, Vue!" за допомогою Vue.js. Результат програми повинен виглядати наступним чином:

Hello Vue!

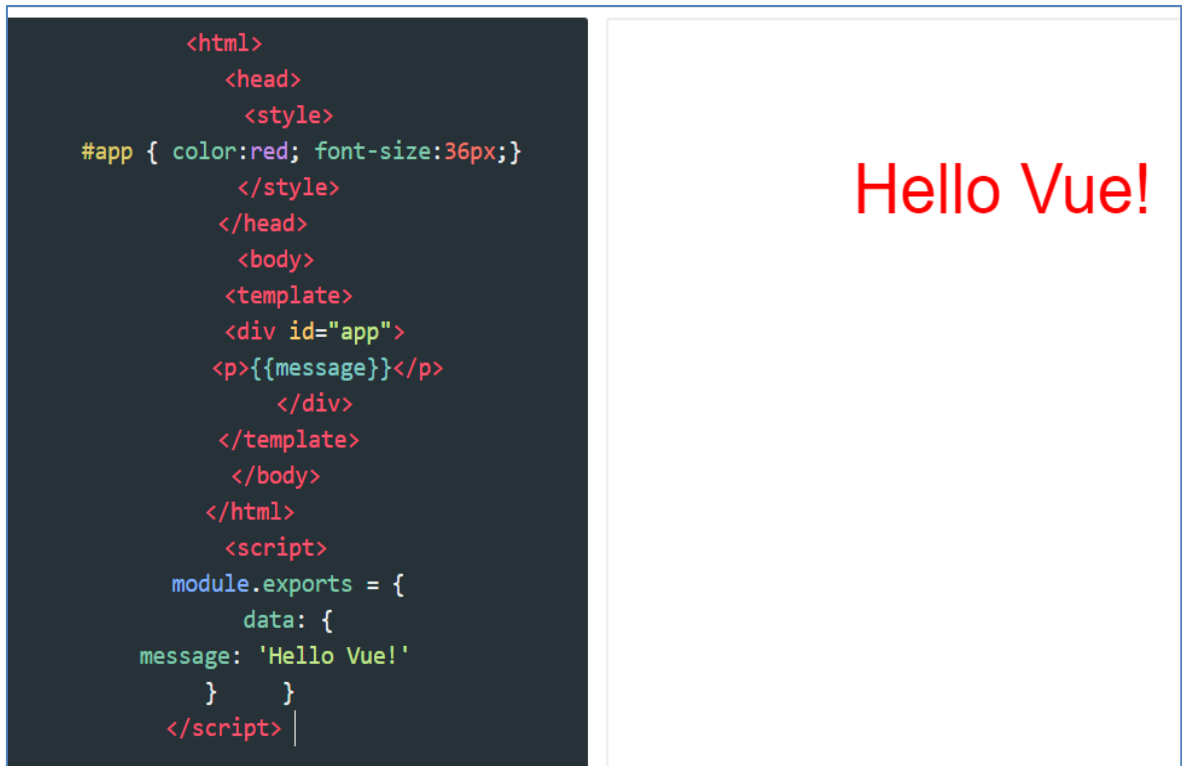
Рис. 2. Приклад зовнішнього вигляду блоку «Завдання для виконання»

Тіло функції:

```
render (h) { // назва функції
  let win
  if (this.error) { // обробка помилок коду користувача
    win = h('div', {
      class: 'vuep-error'
    }, [this.error])
  } else {
    win = h(Preview, { //виконання коду за відсутності помилки
      class: 'vuep-preview', // парсинг написаного коду
      props: {
        value: this.preview,
        styles: this.styles,
        keepData: this.keepData,
        iframe: this.iframe
      },
      on: {
        error: this.handleError
      }
    })
  }
  return h('div', { class: 'vuep' }, [
    h(Editor, {
      class: 'vuep-editor',
      props: {
        value: this.content,
        options: this.options
      },
      on: { // виклик функції при кожній зміні поля редактору коду
        change: [this.executeCode, val => this.$emit('input', val)]
      }
    })
  ], win
```

```
  1)  
}
```

Як наслідок, код, написаний у редакторі при виконанні завдання, автоматично виконується в режимі реального часу. Приклад відображення структури проекту та виконання коду безпосередньо у навчальному середовищі наведено на рисунку 3.

The image shows a split-screen view of a code editor. The left side has a dark background with red and green text representing code. The right side has a white background with the rendered output. The code on the left is a simple Vue.js application that sets a style for a red font and a data property for the message 'Hello Vue!'. The rendered output on the right is the text 'Hello Vue!' in a large red font.

```
<html>  
  <head>  
    <style>  
#app { color:red; font-size:36px;}  
    </style>  
  </head>  
  <body>  
    <template>  
      <div id="app">  
        <p>{{message}}</p>  
      </div>  
    </template>  
  </body>  
</html>  
<script>  
  module.exports = {  
    data: {  
      message: 'Hello Vue!'  
    }  
  }  
</script> |
```

Hello Vue!

Рис. 3. Приклад виконання коду у навчальному середовищі

Таким чином, було продемонстровано, що використання JavaScript-фреймворку Vue.js є зручним для розробки власного навчального інтерактивного середовища, в якому реалізуються як стандартні задачі (наприклад, створення сторінок із довідкою та завданнями) так і спеціальні задачі (наприклад, відображення структури проекту з можливістю виконання певного коду безпосередньо у середовищі). Створений додаток у подальшому може бути використаний як основа для демонстрації процесу інтерпретації синтаксису і виконання програмного коду фреймворку Vue.js під час вивчення курсу «Програмування крос-платформних та розподілених додатків».

Список літератури

1. Alex Kyriakidis, Kostas Maniatis. *The Majesty of Vue.js*. Packt Publishing Ltd, 2016. 230 p.
2. Thiago Pontes, Maxwell Da Silva. Building a Cross Platform 360-degree Video Experience at The New York Times. *New York Times*. 2016. 11 November.
3. Tony Patton. jsFiddle: An online playground for your JavaScript, HTML, CSS. *TechRepublic*. 13 April 2012.
4. Tony Patton. Implementing a Syntax-Highlighting JavaScript Editor—in JavaScript. *TechRepublic*. 24 April 2017.