

Лабораторна робота 8

Тема: приклади використання Vue.JS

Мета: набути практичних навичок у використанні фреймворка Vue.JS

Теоретичні відомості:

Vue.js – JavaScript бібліотека з двосторонньою прив'язкою даних за принципом модель-вистава.

Особливість Vue.js полягає в тому, щоб надати найпростіший можливий API для створення в реальному часі двосторонньої прив'язки даних між поданням (HTML-шаблоном) та моделлю (JavaScript-об'єкт).

Найпростіший спосіб встановити Vue.js – підключити її тегом `<script>` в кінець HTML-документа. Вся бібліотека знаходиться в одному js-файлі, який можна завантажити з офіційного сайту або імпортувати безпосередньо через CDN:

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.0.5/dist/vue.min.js"></script>
```

Якщо ви хочете використовувати Node, Vue доступний як [модуль npm](#). Для модуля доступний власний інтерфейс у терміналі для гнучкого налаштування проекту.

Бібліотека Vue.js може працювати з анімацією, користувальницькими компонентами та має багато інших функцій. Тому до ознайомлення рекомендується чудова [офіційна документація](#) бібліотеки.

Хід роботи

Завдання 1. Меню навігації.

За прикладом на <https://codepen.io/> (посилання на код нижче) створіть свою навігаційну панель для сайту за темою, обраною в лабораторній роботі №6.

Врахуйте, що є кілька основних компонентів, які повинні мати всі програми на Vue.js:

1. Модель (Model) - іншими словами, дані програми. У рамках бібліотеки це js-об'єкт, що містить змінні та їх вихідні значення.
2. Подання (View) – HTML-код, що відображає дані для користувача і містить eventListener'и для взаємодії з моделлю.
3. ViewModel – у термінології MVC це був контролер. Фактично, в рамках Vue.js ViewModel – сполучна ланка між моделлю та поданням.

[Демо на CodePen](#)

Примітка: Як ви можете бачити, робота з даною бібліотекою надає:

- Простий js-об'єкт для опцій
- `{{Подвійні дужки}}` для шаблонів
- Вбудовані атрибути виду `v-something` для додавання функціональності безпосередньо в HTML.

Завдання 2. Вбудований редактор.

Додайте до Вашого сайту вбудований редактор за прикладом нижче.

В завданні 1 наша модель мала лише кілька наперед визначених значень. Для надання користувачам можливість змінювати будь-які дані, нам потрібно зробити двосторонню прив'язку даних та об'єднати поле введення з властивістю моделі. Коли текст вводиться, він автоматично зберігається в моделі `text_content`, яка потім викликає оновлення для подання.

[Демо на CodePen](#)

Ще одна річ, яку слід зазначити у наведеному коді, – це атрибут `v-if`. Він показує чи приховує цілий елемент залежно стану змінної. Прочитати про нього більше можна [тут](#).

Завдання 3. Форма замовлення.

Додайте до Вашого сайту форму замовлення за прикладом нижче.

У прикладі показано список послуг та сумарна вартість вибраних. Оскільки наші послуги зберігаються в масиві, ми можемо використовувати директиву `v-for` для перебору всіх записів та їх відображення. Якщо до масиву додано новий елемент або змінено будь-який із старих, Vue автоматично оновить дані та покаже нові.

[Демо на CodePen](#)

Щоб відобразити ціни у правильному форматі, нам необхідно визначити простий фільтр валюти. Фільтри дають змогу змінювати або фільтрувати дані моделі. Щоб визначити фільтр, що настроюється, використовуємо наступний синтаксис:

```
Vue.filter('currency', function (value) {  
  &nbsp; &nbsp; &nbsp; return '$' + value.toFixed(2);  
});
```

Фільтр досить простий – він додає знак долара та правильний цифровий дріб.

Завдання 4. Миттєвий пошук.

Створіть додаток (за прикладом нижче), в якому будуть представлені деякі статті-приклади. Додаток також матиме текстове поле пошуку, що дозволяє фільтрувати, які статті будуть відображатися. Всі статті будуть зберігатися в масиві, і ті з них, що відповідають пошуковому запиту, будуть потрапляти в масив, викладений матеріалами.

[Демо на CodePen](#)

Поле введення прив'язане до моделі `searchString`. Коли текст вводиться, модель миттєво оновлюється, і масив викладених речовин генерується знову. Таким чином, ми можемо створити пошук у реальному часі, не дбаючи про рендеринг чи налаштування прослуховувачів подій.

Завдання 5. Сітка, що перемикається.

За прикладом нижче перевірте на Vue загальний сценарій, коли сторінка має різні режими компонування. Як і в попередньому додатку, будемо показувати список статей, що зберігаються у масиві.

Натиснувши одну з кнопок у верхній панелі, перемикатиметься режим показу між сіткою, що містить великі зображення, та списком розкладок з меншими зображеннями та текстом.

[Демо на CodePen](#)

Завдання 6. Закомітити завдання до репозиторію <https://git.ztu.edu.ua/>.

Надати доступ maintainer до створеного репозиторію викладачам: kik_pvv@ztu.edu.ua
Болотіна Вікторія Василівна @graf_ztu Граф Марина Сергіївна @kkn_fdv Фуріхата Денис
Вікторович @kkn_oai Оринчак Андрій Іванович.