

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ VUE.JS ТА REACT.JS У ПРОЦЕСІ НАВЧАННЯ СТУДЕНТІВ

У наш час, більшість людей пов'язана із Всесвітньою павутиною. Незалежно від віку або статі, Інтернет увійшов у повсякденне життя. Він допомагає задовольняти свої розважальні, соціальні, дослідницькі та споживчі потреби. Інтернет щодня розвивається і буде свої корені в основі Інтернету. Ми використовуємо його для всього, починаючи від покупок, закінчуючи банківською справою та читаючи новини.

Попит на таланти веб-розробки як завжди високий, а сфера можливостей розробників надзвичайно величезна. Якщо веб-сайт регулярно оновлюється та розробляється, то ми побачимо оновлення веб-сайту. Пошукові системи завжди підбирають свіжий вміст та оновлені веб-сайти. Компанія, що займається розробкою веб-сайтів, повинна зосередитись на переробці веб-сайту шляхом оновлення структури вмісту, інтерфейсного коду, обміну повідомленнями про пропозиції, швидкого реагування та структур посилань. Всі ці зміни прискорюють зростання пошукових систем, збільшуючи охоплення сайту. Хоча сприйняття нового веб-сайту здається м'яким, але зміна сприйняття допомагає зміцнити довіру. Це призводить до створення зусиль, продажу та конкурентоспроможності.

Технологія та тенденції розробки веб-сайтів, змінюються відповідно до потреб та інтересів клієнтів. Наприклад, усі компанії, які запустили свій сайт без адаптивного інтерфейсу, стикалися з втратами. Таким чином, веб-дизайн допомагає модернізувати веб-сайт бренду, задовольняючи потреби цільової аудиторії. Це збільшує потенційну аудиторію, що знижує рівень відмов та збільшує охоплення клієнтів. Щоразу, коли веб-сайт оновлюється, компанія має можливість вчитися з минулого. Вивчаючи веб-аналітику, ви можете дізнатися, який вміст допоможе вам збільшити охоплення.

JavaScript є найбільш часто використовуваною мовою в середовищі веб-розробників, 67,7% людей обирають його. Основною причиною його популярності є той факт, що JavaScript є універсальним і може використовуватися як для інтерфейсу, так і для внутрішньої розробки, а також для тестування веб-сайтів або веб-додатків.

Займаючись пошуком найкращих фреймворків для JS, можна натрапити на різні фреймворки, кожен зі своїми перевагами та використанням. Оскільки настільки багато можливостей для фреймворків JavaScript для інтерфейсу, внутрішньої розробки або навіть тестування, важко вибрати правильний фреймворк для ваших вимог. Може бути важко знайти ідеальну основу для ваших вимог.

Проблематика. Фреймворки розробляються для того, щоб звільнити від написання однотипного коду. Але, у міру того як кодова база деяких фреймворків сильно розростається, вони починають привносити свою частку складності в проєкт. Через це при плануванні розробки потрібно враховувати два фактори: складність нашого застосування; складність фреймворка, який ми використовуємо.

Два варіанти вибору інструменту:

1. Варіант недостатності. Коли інструменту недостатньо, щоб перекрити внутрішню складність. Функції, необхідні для реалізації прог-рами, відсутні в фреймворку, і нам доводиться вручну допрацьовувати і додавати необхідний інструментарій.

2. Варіант надмірності. Коли необхідний для додатка функціонал перекривається тільки невеликою частиною інструменту. Інша ж частина інструментарію висить мертвим вантажем і тільки створює нам проблеми: обмеження при розробці, уповільнення завантажити прог-раму і т.п.

Мета полягає в ознайомленні з сильними та слабкими сторонами Vue.js та React.js. У визначенні переваг від використання цих фреймворків у процесі навчання студентів. Vue та React - це JavaScript фреймворки, які користуються попитом в світі веб-розробників, React.js займає перше місце останні декілька років. Вони пропонують розробникам продуктивний підхід до створення різних веб-додатків, але кожен з них має власні особливості використання. Є безліч причин популярності Vue і React серед розробників, але перш ніж заглибитися в їх особливості, визначимося з тим, що робить їх такими популярними:

– Virtual DOM - замість того, щоб повторно відображати всю сторінку, як це відбувається у звичайному режимі оновлення DOM, React і Vue відображає лише ті об'єкти, які змінилися, економлячи час та ресурси, які інакше вимагають важких маніпуляцій з DOM;

– Розробка інтерфейсу на основі компонентів - значні бібліотеки компонентів Vue та React полегшують повторне використання коду, прискорюють процес розробки;

– Офіційна бібліотека компонентів для створення мобільних додатків - досить довгоочікувана у Vue, але зараз підтримується обома фреймворками.

Однією з найбільших відмінностей між Vue та React є спосіб побудови шару подання. За замовчуванням Vue використовує шаблони HTML, але є можливість писати в JSX. В React існує виключно JSX. Традиційне розділення проблем Vue на HTML, CSS та JS допомагає полегшити створення веб-додатків. Шаблони HTML також знайомі багатьом веб-дизайнерам і тим самим покращують співпрацю між розробниками та дизайнерами.

JSX React поєднує HTML і CSS разом у JavaScript. Цей синтаксис, схожий на XML, він дозволяє створювати автономні компоненти інтерфейсу користувача з включеними інструкціями щодо перегляду подання. З моменту появи Vue пройшло не дуже багато часу. Це робить даний фреймворк наймолодшою розробкою серед подібних систем. Однією з особливостей Vue є той факт, що в ньому простежується поєднання можливостей деяких інших успішних JS-фреймворків.

Бібліотека React відома як потужний засіб для створення динамічних і призначених для користувача інтерфейсів. 19 березня 2020 року побачила найсвіжіша версія React (16.13.1). Ця бібліотека використовується для розробки інтерфейсів в багатьох відомих компаніях.

Vue легкий у навчанні та приємний для запису. Завдяки звичному синтаксису шаблонів та використанню компонентів, інтеграція чи переміщення існуючих проєктів до Vue відбувається швидше та легше. Vue.js чудово підходить для стартапів, але може бути так само добре використаний у масштабних програмах. Виконання продуктивності, Vue.js нарівні з React, але кінцеві результати залежать від зусиль з оптимізації та розміру програми. Відмінна документація, все детально пояснює. За допомогою Vue CLI 3 та Vue UI налаштування проєкту та підготовка його до розгортання відбувається дуже швидко.

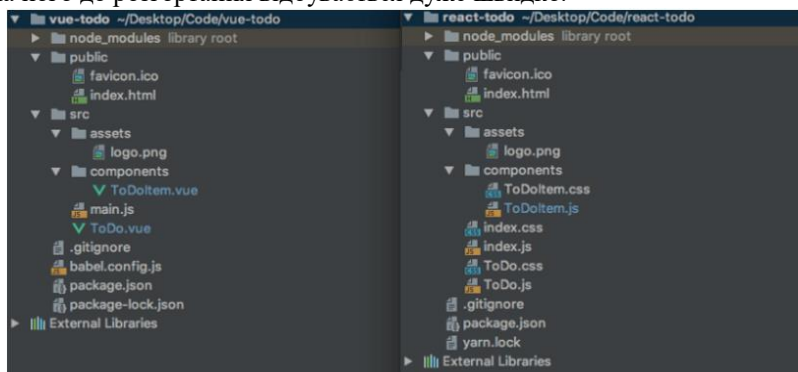


Рис. 1. Структура проєктів на Vue.js та React.js

Можна помітити, що структура цих двох проєктів практично ідентична. Серйозна відмінність полягає в тому, що у React-додатки є три CSS-файли, в той час як у Vue-додатки їх немає зовсім. Причина полягає в тому, що, при використанні create-react-app, компоненти React оснащуються супутніми CSS-файлами, а CLI Vue використовує інший підхід, коли стилі оголошуються всередині конкретного файлу компонента.

В результаті обидва підходи дозволяють досягти однієї і тієї ж мети, при цьому, при бажанні, ніщо не заважає організувати стилі інакше в Vue або в React. Все зводиться до особистих вподобань при створенні web-проєкт. Наприклад, тема структурування CSS постійно обговорюється в спільнотах розробників.

Скільки часу потрібно витратити на вивчення даних фреймворків? React надає інструкції, завдяки яким налаштувати даний продукт можна за годину. Однак він не є повним набором рішень, тому є необхідність використання сторонніх бібліотек. Vue інтуїтивно зрозумілий, тому його освоєння буде досить швидким. Однак, при роботі з ним допустимий брак коду, тому він вимагає постійного налагодження і додаткового тестування.

Висновок: вибір правильного інструменту залежить від поставленої задачі. Vue і React - це гарні інструменти для створення інтерактивних інтерфейсів користувачів. Щоб вибрати, який із них краще підходить для проєкту, слід врахувати ряд факторів, які включають конкретний випадок використання, потреби бізнесу, бюджет та часовий період.

Список використаних джерел

1. What is Vue.js? [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/v2/guide/>.
2. Руководство по Vue.js [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/web/vuejs/>.
3. Изучение React. Полное руководство по React [Електронний ресурс] – Режим доступу до ресурсу: <https://learn-reactjs.ru/home>.
4. Стефанов С. React.js. Быстрый старт / Стоян Стефанов. – Питер, 2016. – 304 с.
5. Хэнчеп Э. Vue.js в действии / Э. Хэнчеп, Б. Листуон. – Питер, 2018. – 304 с.