

РОЗРОБКА ВЕБ-ОРІЄНТОВАНОГО ДОДАТКУ В СФЕРІ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

НТУ «Дніпровська політехніка»

Файнштейн Данііл Володимирович

Науковий керівник: к.т.н., доц. Спірінцев В'ячеслав Васильович

Сучасний період розвитку національної економіки характеризується посиленням конкуренції між суб'єктами комерційної діяльності, що обумовлює необхідність оперативного реагування на зміни ринку товарів і послуг та швидкого і ефективного здійснення комерційних операцій. Перспективним напрямком ведення сучасного інтерактивного бізнесу в умовах жорсткої конкуренції на фоні воєнного стану стала електронна комерція, яка охоплює всі види торгівлі через Internet.

Під час обговорення нових проектів часто виникає питання про вибір підходу, щодо розробки, зокрема:

- використання вже готових систем (систем управління контентом, CMS);
- застосування кастомної розробки.

Розглянемо основні особливості кожного з підходів, їх переваги та недоліки.

До переваг використання CMS [1] відносять: зручність; економію фінансових вкладень та часу на розробку; наявність добре проробленої документації; мультишаблонність; легко розширювану функціональність за допомогою плагінів; відокремлення контенту від візуального представлення; комплексність; незалежність; керованість; добру технічну підтримку у вигляді спільноти користувачів і т.ін.

Однак, не зважаючи на численні переваги, CMS притаманні наступні недоліки: шаблонність, обмеженість функціоналу; простота управління контентом є лише відносною, оскільки якщо виникне бажання реалізувати на сайті якісь нестандартні рішення у функціоналі або дизайні, це буде проблематично; надмірна функціональність CMS; невисока стійкість до навантаження; низький рівень безпеки, в наслідок відкритості коду і безкоштовності систем; необхідність в розширенні доступного функціоналу; складнощі вивчення.

Кастомні рішення базуються на розробці з використанням фреймворків (від англ. framework – “каркас”) або бібліотек. Фреймворк – надбудова над мовою, набір бібліотек, що полегшує розробку та об'єднання різних модулів програмного проекту. Такий варіант чудово підійде компаніям, які впроваджують нестандартні торгові пропозиції та реалізують індивідуальні системи ведення бізнесу.

До переваг кастомної розробки слід віднести:

- Гнучкість. Можливість реалізувати всі задуми та доопрацювання без обмежень; можливість додавання унікального функціоналу під час розробки та виключення функціоналу, що не використовується; можливість нарощувати

функціонал та масштабувати будь-який проект із мінімальними ризиками; Висока швидкість завдяки оптимізованій функціональності; ексклюзивний дизайн і т.ін.

– Продуктивність. Сайт на CMS може іноді поступатись в продуктивності добре написаному сайту на фреймворку або з використанням бібліотеки.

Враховуючи зазначені переваги в роботі було прийнято рішення щодо застосування кастомної розробки. Для реалізації веб-орієнтованого додатку було використано наступні технології: бібліотека React, фреймворк Spring, база даних MySQL. На рис.1 наведено архітектуру додатку.

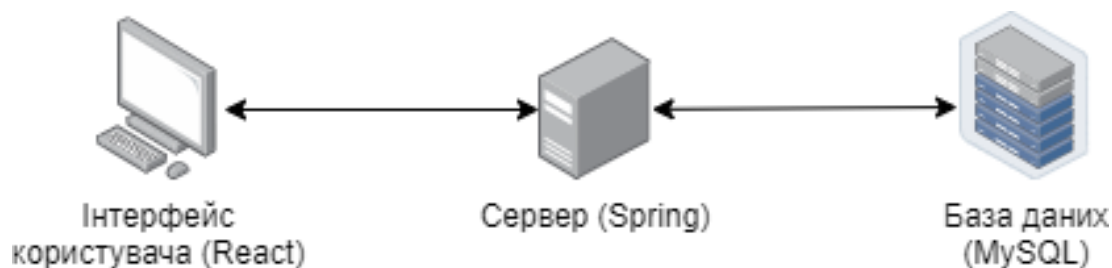


Рис. 1 Архітектура додатку

Для розробки інтерфейсу користувача було використано бібліотеку React [2], що надає швидкість і продуктивність проекту та дозволяє створювати інтерфейси, які реагують на зміни стану додатку. Це означає, що веб-сторінки оновлюються автоматично без перезавантаження (що забезпечує більш швидкий та зручний інтерфейс для користувачів) та надає компонентну структуру (веб-додаток складається з кількох компонентів і кожен компонент має власну логіку та елементи керування). Ці компоненти відповідають за виведення невеликого багаторазового фрагмента HTML-коду, який можна повторно використовувати.

Використовуючи Axios API (це HTTP-клієнт, заснований на Promise для Node.js та браузера), з інтерфейсу користувача відправляється запит на сервер Spring, де виконується його подальша обробка. Приведемо приклад таких запитів:

GET-запити:

- отримання списку товарів з використанням пагінації;
- отримання товару за його ідентифікатором та кольором;
- сортування товару за його ознаками.

POST-запити:

- реєстрація нового користувача;
- вхід до системи вже існуючого користувача;
- створення замовлення.

Метод обробки запиту задіє Spring Boot JPA – це специфікація Java для керування реляційними даними в програмах Java. Вона дозволяє отримати доступ до даних і зберігати їх між об'єктом/класом Java і реляційною базою даних [3].

На рис.2 наведено приклад JPA репозиторію.

```
1 usage
public interface EyeglassRepository extends JpaRepository<EyeglassView, Integer> {
    1 usage
    @Query("SELECT e FROM EyeglassView e WHERE e.id = :id AND e.color != :color")
    List<EyeglassView> findAltColorsById(@Param("id") int id, @Param("color") String color);

    1 usage
    EyeglassView findByIdAndColor(int id, String color);

    List<EyeglassView> findAll();

    @Query("SELECT e FROM EyeglassView e WHERE e.position = 1")
    List<EyeglassView> findAllMain();
}
```

Рис. 2 Приклад JPA репозиторію

Отриманий об'єкт з бази даних MySQL чи повідомлення повертається у відповіді до запиту на інтерфейс користувача. React використовує JSX (JavaScript XML) – це розширення React, яке дозволяє писати код JavaScript, який виглядає як HTML, що дає можливість з легкістю використовувати отриманий об'єкт разом з HTML розміткою [4].

Дизайн же сайту створюється з використанням CSS та допоміжних інструментів таких як Bootstrap.

Отже, використання запропонованих в роботі технологій React, Spring та MySQL дозволяє створювати потужні веб-додатки, які задовільняють потреби користувачів. Оскільки інтерфейс користувача працює окремо від серверної частини, то відкривається можливість до більш легшого розширення додатку на інші платформи у разі необхідності.

Перелік посилань

1. Призначення та особливості CMS [Електронний ресурс]. URL: <https://armedsoft.com/ua/blog/pryznachennya-ta-osoblyvosti-cms> (дата звернення: 14.04.2023).

2. The benefits of ReactJS and reasons to choose it for your project [Електронний ресурс]. URL: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html> (дата звернення: 14.04.2023).

3. Introduction to Spring Data JPA [Електронний ресурс]. URL: <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa> (дата звернення: 19.04.2023).

4. Introducing JSX [Електронний ресурс]. URL: <https://legacy.reactjs.org/docs/introducing-jsx.html> (дата звернення: 19.04.2023).