

РОЗРОБКА ВЕБ-ЗАСТОСУНКІВ ДЛЯ ОНЛАЙН КУПІВЛІ ВІДЕО-ІГОР

Хомсі Как С.М., Колесник Л.В.

Харківський національний університет радіоелектроніки, Харків

Зі зростанням популярності відео-ігор у всьому світі, онлайн-продажі цих ігор стають все більш актуальними. У минулому це були фізичні копії дисків або картриджі, які могли бути вкрадені в ігрових магазинах або видавництвах та продані на локальному ринку.

Наразі цю проблему все частіше вирішує цифрова дистрибуція, яка надає можливість купувати ліцензійну копію гри та не хвилюватися, що вона може бути не оригінальною. За статистикою, користувачі, що захоплюються відеоіграми, витрачають мільярди доларів на купівлю відео-ігор та пов'язаних з ними додатків. Щоб забезпечити успішну роботу онлайн-магазину і задовольнити потреби клієнтів, необхідно розробити простий, максимально функціональний та швидкий веб-застосунок [1].

Однією з ключових проблем, з якою стикаються такі користувачі, є безальтернативність таких онлайн-магазинів. Відсутність альтернативи та висока концентрація на ринку може призвести до зловживань та несприятливих умов для споживачів. Тому розробка конкурентоздатного онлайн-магазину важлива для забезпечення доступом до найкращих пропозицій та ліцензійних копій ігор.

Першим етапом розробки онлайн-магазину є аналіз клієнтської частини, тобто визначення потреб цільової аудиторії та створення інтерфейсу, що задовольняє їх очікування [2]. Клієнти хочуть мати інтерфейс з усім, що їм потрібно та доступ до необхідних функцій, наприклад, пошук, покупка та завантаження відео-ігор. Необхідно розуміти, що інтерфейс повинен бути швидким та мінімалістичним, щоб це спростило взаємодію користувача з сайтом. Завдяки зручному інтерфейсу, користувачі відео-ігор зможуть легко здійснювати покупки та спілкуватися на платформі [3].

Але інтерфейс не є єдиною важливою складовою онлайн-магазину. Захист даних

користувачів, видавництв та виробників є однією з ключових особливостей такого застосунку. Втрата або витік таких даних може призвести до серйозних проблем, включаючи втрату даних, репутації та фінансових втрат. Тому необхідно належно налаштувати системи захисту та запобігання втратам даних під час використання онлайн-магазину.

Ще одним важливим критерієм є забезпечення надійності і захищеності транзакцій. Розробка системи оплати, яка гарантує, що гроші користувачів будуть витрачені на ліцензійні копії ігор, а не потраплять в руки шахраїв, є необхідною [4]. Захист від шахрайства та зловживань повинен бути вбудованим в інфраструктуру онлайн-магазину, щоб забезпечити високий рівень довіри користувачів та забезпечити безпеку їхніх фінансових операцій.

Останнім ключовим аспектом має бути система підтримки у вирішенні питань та проблем користувачів. Розробка ефективної системи підтримки, яка надає відповіді на запитання та вирішує можливі проблеми може сприяти вдячністю та довірою платформі. Вона повинна мати різноманітні способи спілкування, а саме: електронна пошта, чат-бот та автоматичний обробник подій, які полегшують вирішення проблем.

Також, серед додаткових компонентів можна виділити розробку маркетингової стратегії та її просування для привертання нових клієнтів та збільшення продажів. Однією з можливою стратегій може бути планування рекламних кампаній, або використання соціальних медіа, що приверне увагу користувачів. Не менш важливим є розгляд можливостей монетизації, включаючи різні моделі продажу, додаткові послуги та партнерські програми, або комісії при використанні додатку, які допоможуть збільшити прибуток та підтримувати розвиток платформи.

Для реалізації такого типу проєкта може бути використані Spring Boot Framework, MySQL та Next.js.

Spring Boot Framework відмінно підходить для розробки серверної частини, оскільки надає зручний та ефективний фреймворк для створення Java-додатків. Його високий рівень абстракції дозволяє швидко, просто розгортати проєкти та ефективно взаємодіяти з базою даних.

На рис. 1 – 2 можна побачити, як відбувається взаємодія між реляційною базою даних та серверною частиною.

```
@Query("select g from Game g " +  
      "where upper(g.title) like upper(concat('%', :title, '%'))")  
Page<Game> findGameByTitleContainingIgnoreCase(@Param("title")  
                                              String title, Pageable pageable);
```

Рисунок 1 – Використання запиту для отримання даних із бази даних

```
@GetMapping("/{game-id}")  
@Operation(summary = "Get game by id")  
@ResponseStatus(HttpStatus.OK)  
public Game showGameById(  
    @PathVariable("game-id")  
    @Min(1) @Max(Long.MAX_VALUE) Long gameId) {  
    return gameService.getGameById(gameId);  
}
```

Рисунок 2 – Опрацювання отриманих даних із бази даних

Як можемо побачити, на рис. 1 створюється запит за допомогою якого код отримує доступ до бази даних, після чого пишеться інший код, який використовує це для написання бізнес-логіки. Кінцеву точку опрацювання цих даних можна побачити на рис. 2.

Spring Boot Framework також відомий своєю розширюваністю, що дозволяє розширювати функціонал застосунку відповідно до змінюючись потреб проєкту. У майбутньому такий проєкт може бути перероблений під модульний проєкт, що має надати змогу ще більше розширити проєкт та зробити компоненти незалежними один від одного [4].

При виборі бази даних найпопулярнішою є MySQL, бо це потужна та надійна система управління базами даних, яка забезпечить ефективне зберігання та обробку даних. Її широкі можливості та добре документовані характеристики роблять її відмінним вибором для потреб проєкту такого характеру. На рис. 3 розглянемо приклад роботи з базою даних MySQL.

Після створення таблиці, MySQL надає нам змогу використовувати тригери та процедури, які можуть бути корисними при заповненні бази даних. Для прикладу

візьмемо псевдокод, який наведено на рис. 4. Коли процедура буде успішно додана, її можна викликати для опрацювання потрібних даних.

```
CREATE TABLE IF NOT EXISTS `GridDB`.`games`
(
  `game_id` INT NOT NULL AUTO_INCREMENT,
  `title` VARCHAR(255) NOT NULL,
  `description` TEXT NULL,
  `release_date` DATE NULL,
  `system_requirements` TEXT NULL,
  `price` DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
  `discount` DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
  `permit_age` ENUM ("0", "3", "7", "12", "16", "18", "!") NOT NULL DEFAULT '0',
  `cover_image_url` TEXT NULL,
  `developer_id` INT NOT NULL,
  `publisher_id` INT NOT NULL,
  PRIMARY KEY (`game_id`),
  INDEX `fk_games_developers1_idx` (`developer_id` ASC) VISIBLE,
  INDEX `fk_games_publishers1_idx` (`publisher_id` ASC) VISIBLE,
  CONSTRAINT `fk_games_developers1`
    FOREIGN KEY (`developer_id`)
      REFERENCES `GridDB`.`developers` (`developer_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_games_publishers1`
    FOREIGN KEY (`publisher_id`)
      REFERENCES `GridDB`.`publishers` (`publisher_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
)
```

Рисунок 3 – Створення таблиця для опрацювання запитів у базі даних

```
CREATE PROCEDURE insert_game(
  IN game_title VARCHAR(255),
  IN game_description TEXT,
  IN game_release_date DATE,
  IN game_system_requirements TEXT,
  IN game_price DECIMAL(10, 2),
  IN game_permitted_age TEXT,
  IN game_cover_image_url TEXT,
  IN game_banner_image_url TEXT,
  IN game_trailer_url TEXT,
  IN game_screenshot_url TEXT,
  IN game_trailer_screenshot_url TEXT,
  IN p_developer_name VARCHAR(255),
  IN p_publisher_name VARCHAR(255),
  IN p_tags TEXT,
  IN p_genres TEXT,
  IN p_platforms TEXT
)
```

Рисунок 4 – Псевдокод по створенню процедури у MySQL

Next.js використовується для розробки клієнтської частини і гармонійно поєднує в собі високу продуктивність та простоту використання [2]. Використання Next.js дозволяє побудувати швидкий та реактивний інтерфейс, забезпечуючи при цьому простоту розробки та підтримки.

Підсумовуючи усе сказане вище, кінцеву схему взаємодії усіх цих технологій можемо побачити на рис. 5.

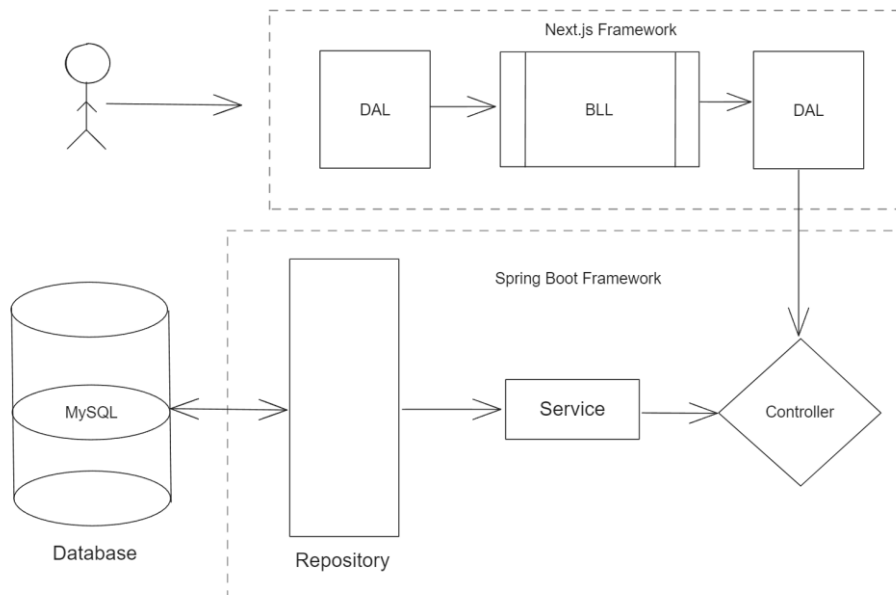


Рисунок 5 – Архітектура проєкту з використанням MySQL, Spring Boot та Next.js.

Отже, розробка компонентів онлайн-магазину для відео-ігор має важливу роль для зростання популярності цього ринку. Розуміючи це, інтерфейс, захист даних, система оплати та підтримка користувачів мають ключові пріоритети, необхідними для забезпечення задоволення потреб клієнтів та успішності платформи.

Розробка цих компонентів є складним завданням, але їх успішне впровадження може призвести до значних переваг як для користувачів, що захоплюються відеоіграми, так і для розробників ігор і видавців.

Література:

1. Основні етапи створення сайту, 2020. [Он-лайн]. Доступно: <https://impulse-design.com.ua/ua/etapy-razrabotki-sajta.html>
2. Як розробляють ігри, 2023. [Он-лайн]. Доступно: <https://lemon.school/blog/yak-rozroblyayut-igry>
3. Mastering Website Navigation: The Ultimate Guide, 2022. [Он-лайн]. Доступно: <https://ahrefs.com/blog/website-navigation/>
4. U. Cosmina, R. Harrop, C. Schaefer, and C. Ho, Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools. New York, USA: Apress, 2017.