**FINAL PROJECT REPORT**

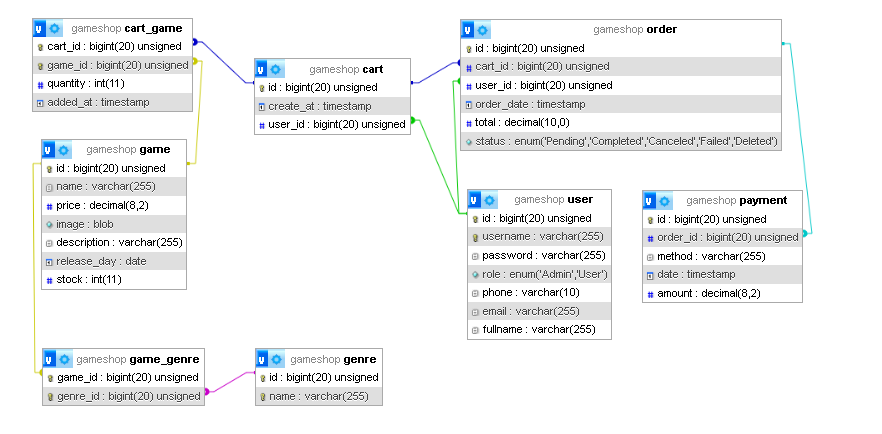
SEMESTER 1, ACADEMIC YEAR: 2024-2025

CT313H: WEB TECHNOLOGIES AND SERVICES

* **Project/Application name: Game Shop**
* **GitHub links (for both frontend and backend):** [**24-25Sem1-Courses/ct313h02-project-Tanchuoi: ct313hm02-ct313h02-project-ct313h-project created by GitHub Classroom**](https://github.com/24-25Sem1-Courses/ct313h02-project-Tanchuoi)
* **Student ID 1: B2111950**
* **Student Name 1: Đinh Hồ Thanh Tân**
* **Student ID 2: B2111923**
* **Student Name 2: Lê Tào Quốc Hải**
* **Class/Group Number: CT313HM02**

1. **Introduction**

* Project/application description:A few words introducing an overview of the project/web application.
* Database:



* A task assignment sheet for each member if working in groups.

|  |  |
| --- | --- |
| Member | Work |
| Đinh Hồ Thanh Tân | Front-end |
| Lê Tào Quốc Hải | Back-end |

1. **Details of implemented features**
2. **Feature: Login**

* **Description:** The login feature allows users to authenticate themselves by providing their username and password. Upon successful authentication, the user receives a JWT token, which is used for subsequent authenticated requests to the server.
* **Screenshots:** some screenshots of this feature / application page.
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
      * jsonwebtoken: Used for generating JWT tokens.
      * bcrypt: Used for hashing and comparing passwords.
    - Frontend:
  + **Server-Side APIs:**
    - Login
      * Endpoint: POST /api/v1/login/
      * Description: This endpoint authenticates a user by their username and password.
      * Request Body:
      * Response:
  + **Tables:** user
  + Which client-side states are needed to implement this feature?

1. **Feature: User Registration**

* **Description:** The User Registration feature allows new users to create an account on the platform. This feature collects user information such as username, email, and password, and stores it securely in the database. It is used to authenticate users and provide them access to the platform's services.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
      * Bcrypt: For hashing passwords.
    - Frontend:
  + **Server-Side APIs:**
    - **Register**
      * Endpoint: POST /api/v1/register
      * Description: This endpoint is used to register a new user.
      * Request Body:
      * Response:
  + **Tables:** user
  + Which client-side states are needed to implement this feature?

1. **Feature: Update User**

* **Description:** The Update User feature allows existing users to update their account information, such as email, password, phone and full name. This feature ensures that users can keep their account details up-to-date.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
      * Bcrypt: For hashing passwords.
    - Frontend:
  + **Server-Side APIs:**
    - **UpdateUser:**
      * Endpoint: PUT /api/v1/user/:id
      * Description: This endpoint is used to update an existing user's information.
      * Request Body:
      * Response:
  + **Tables:** user
  + Which client-side states are needed to implement this feature?

1. **Feature: Add Game To Cart**

* **Description:** The "Add Game To Cart" feature allows users to add a selected game to their shopping cart, if user add the first game, the system will create a cart for user. This feature is essential for users to compile a list of games they wish to purchase before proceeding to checkout.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **AddGameToCart**
      * Endpoint: POST /api/v1/cart/games
      * Description: This endpoint is used to add a game to the user's cart.
      * Request Body:
      * Response:
    - **AddCart**
      * Endpoint: POST /api/v1/cart
      * Description: This endpoint is used to add a cart of the user's.
      * Request Body:
      * Response:
  + **Tables:** cart\_game, cart
  + Which client-side states are needed to implement this feature?

1. **Feature: Search Game**

* **Description:** The "Search Game" feature allows users to search for games in the database using keywords. This feature helps users quickly find games they are interested in by entering search terms related to the game's title, genre, or other attributes.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **GetGameBySearch**
      * Endpoint: GET /api/v1/games/search/:search
      * Description: This endpoint is used to search for games based on a search term.
      * Request Body:
      * Response:
  + **Tables:** game
  + Which client-side states are needed to implement this feature?

1. **Feature: Get Game By Category**

* **Description:** The "Get Game By Category" feature allows users to retrieve a list of games that belong to a specific category. This feature helps users explore games based on their preferred genres or categories.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **GetGameByCategory**
      * Endpoint: GET /api/v1/games/category/:id
      * Description: This endpoint is used to retrieve games based on a specific category.
      * Request Body:
      * Response:
  + **Tables:** game, game\_genre
  + Which client-side states are needed to implement this feature?

1. **Feature: Delete User(Admin)**

* **Description:** The "Delete User (Admin)" feature allows administrators to delete a user account from the system. This feature is essential for maintaining the integrity of the user base and removing accounts that violate terms of service or are otherwise deemed unnecessary.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **DeleteUser**
      * Endpoint: DELETE /api/v1/user/:id
      * Description: This endpoint is used by administrators to delete a user account.
      * Request Body:
      * Response:
  + **Tables:** user, order, cart, cart\_game, payment
  + Which client-side states are needed to implement this feature?

1. **Feature: Update Game(Admin)**

* **Description:** The "Update Game (Admin)" feature allows administrators to update the details of an existing game in the system. This feature is essential for maintaining accurate and up-to-date information about the games available on the platform.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **UpdateGame**
      * Endpoint: PUT /api/v1/games/:id
      * Description: This endpoint is used by administrators to update the details of an existing game.
      * Request Body:
      * Response:
  + **Tables: game**
  + Which client-side states are needed to implement this feature?

1. **Feature: Update Game In Cart**

* **Description:** The "Update Game In Cart" feature allows users to update the quantity or details of a game that they have already added to their shopping cart. This feature is essential for users to manage their cart contents before proceeding to checkout.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **UpdateGameInCart**
      * Endpoint: PUT /api/v1/cart/:id
      * Description: This endpoint is used to update the details of a game in the user's cart.
      * Request Body:
      * Response:
  + **Tables: cart\_game**
  + Which client-side states are needed to implement this feature?

1. **Feature: Get Game**

* **Description:** The "Get Game" feature allows users to retrieve detailed information in a list of game, or by id. This feature is essential for users to view comprehensive details about a game, including its title, genre, release date, and price.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **GetGame**
      * Endpoint: GET /api/v1/games/
      * Description: This endpoint is used to retrieve detailed information of a list of game.
      * Request Body:
      * Response:
    - **GetGameById**
      * Endpoint: GET /api/v1/games/:id
      * Description: This endpoint is used to retrieve detailed information about a specific game by its ID.
      * Request Body:
      * Response:
  + **Tables:** game
  + Which client-side states are needed to implement this feature?

1. **Feature: Create Game**

* **Description:** The "Create Game" feature allows administrators to add new games to the system. This feature is essential for keeping the game catalog up-to-date with the latest releases and ensuring that users have access to a wide variety of games. **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **CreateGame**
      * Endpoint: POST /api/v1/games/
      * Description:  This endpoint is used by administrators to create a new game.
      * Request Body:
      * Response:
    - **AddGameGenres**
      * Endpoint: POST /api/v1/games/ category/
      * Description:  This endpoint is used by administrators to create genres for a game.
      * Request Body:
      * Response:
  + **Tables:** game, game\_genre
  + Which client-side states are needed to implement this feature?

1. **Feature: Delete Game**

* **Description:** The "Delete Game" feature allows administrators to remove a game from the system. This feature is essential for maintaining the integrity of the game catalog and removing outdated or unwanted games.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **DeleteGame**
      * Endpoint: DELETE /api/v1/games/:id
      * Description: This endpoint is used by administrators to delete a game.
      * Request Body:
      * Response:
  + **Tables:** game, game\_genre, cart, cart\_game, order, payment
  + Which client-side states are needed to implement this feature?

1. **Feature: Add Category**

* **Description:** The "Add Category" feature allows administrators to add new game categories to the system. This feature is essential for organizing games into different genres or categories, making it easier for users to find games based on their interests.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **AddGenre**
      * Endpoint: POST /api/v1/category/
      * Description: This endpoint is used by administrators to create a new game category.
      * Request Body:
      * Response:
  + **Tables:** genre
  + Which client-side states are needed to implement this feature?

1. **Feature: Delete Category**

* **Description:** The "Delete Category" feature allows administrators to remove a game category from the system. This feature is essential for maintaining the integrity of the category list and removing outdated or unnecessary categories.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **DeleteCategory**
      * Endpoint: DELETE /api/v1/category/:id
      * Description: This endpoint is used by administrators to delete a game category.
      * Request Body:
      * Response:
  + **Tables:** genre, game\_genre
  + Which client-side states are needed to implement this feature?

1. **Feature: Remove Game From Cart**

* **Description:** The "Remove Game From Cart" feature allows users to remove a specific game from their shopping cart. This feature is essential for users to manage their cart contents before proceeding to checkout.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **RemoveGameFromCart**
      * Endpoint: DELETE /api/v1/cart/games/
      * Description: This endpoint is used to remove a specific game from the user's cart.
      * Request Body:
      * Response:
  + **Tables:** cart\_game
  + Which client-side states are needed to implement this feature?

1. **Feature: Get Cart**

* **Description:** The "Get Cart" feature allows users to retrieve the contents of their shopping cart. This feature is essential for users to view the games they have added to their cart before proceeding to checkout.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **GetCart**
      * Endpoint: GET /api/v1/cart/
      * Description: This endpoint is used to retrieve the contents of the user's cart.
      * Request Body:
      * Response:
  + **Tables: cart,** cart\_game
  + Which client-side states are needed to implement this feature?

1. **Feature: Get Game In Cart**

* **Description:** The "Get Game in Cart" feature allows users to retrieve detailed information about a specific game in their shopping cart. This feature is essential for users to view the details of a game they have added to their cart, such as the title, quantity, and price.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **GetGamesInCart**
      * Endpoint: GET /api/v1/cart/games/:id
      * Description: This endpoint is used to retrieve detailed information about a specific game in the user's cart.
      * Request Body:
      * Response:
  + **Tables:** game,cart\_game
  + Which client-side states are needed to implement this feature?

1. **Feature: Create Order**

* **Description:** The "Create Order" feature allows users to place an order for the games in their shopping cart. This feature is essential for completing the purchase process and generating an order record in the system.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **CreateOrder**
      * Endpoint: POST /api/v1/order/
      * Description: This endpoint is used to create a new order for the games in the user's cart.
      * Request Body:
      * Response:
  + **Tables:** order
  + Which client-side states are needed to implement this feature?

1. **Feature: Delete Order**

* **Description:** The "Delete Order" feature allows administrators to remove an order from the system. This feature is essential for maintaining the integrity of the order records and removing outdated or erroneous orders.
* **Screenshots:**
* **Implementation details:**
  + **Libraries Used:**
    - Backend:
    - Frontend:
  + **Server-Side APIs:**
    - **DeleteOrder**
      * Endpoint: DELETE /api/v1/order/:id
      * Description: This endpoint is used by administrators to delete an order.
      * Request Body:
      * Response:
  + **Tables:** order, payment
  + Which client-side states are needed to implement this feature?