

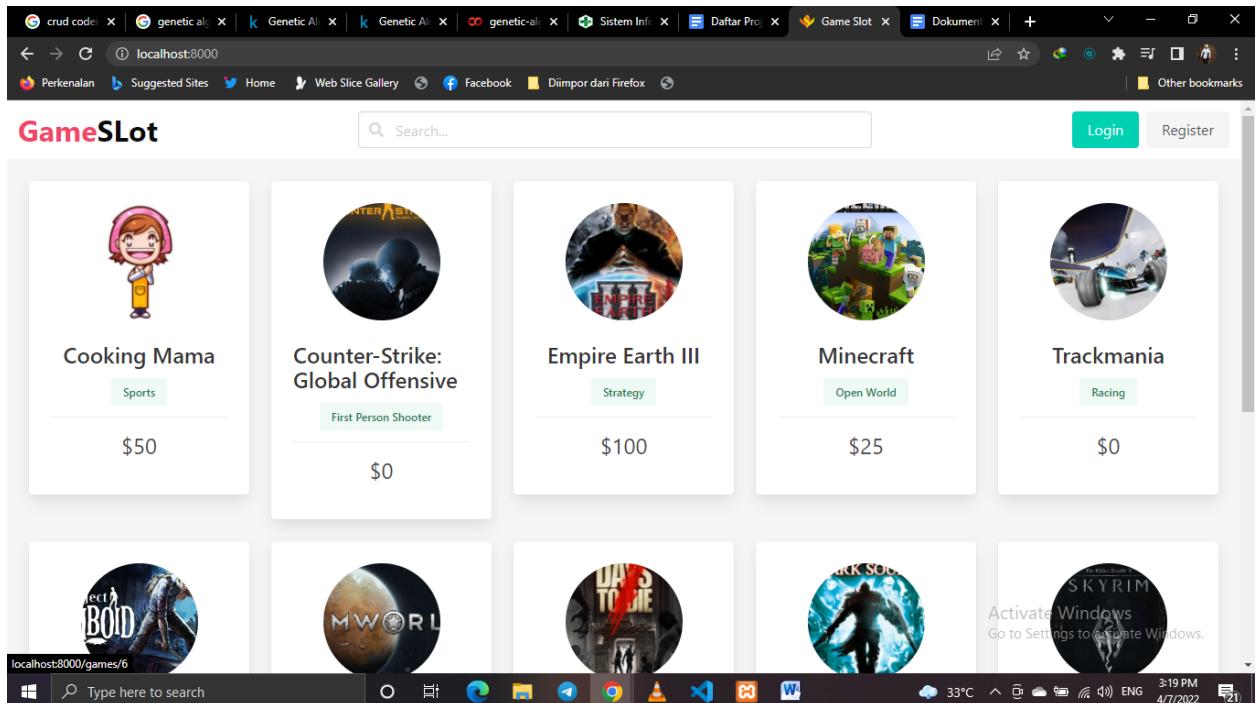
Nama : Zhafirah Rizqy Nur Shabrina Prasetija

## Dokumentasi Project GameSlot

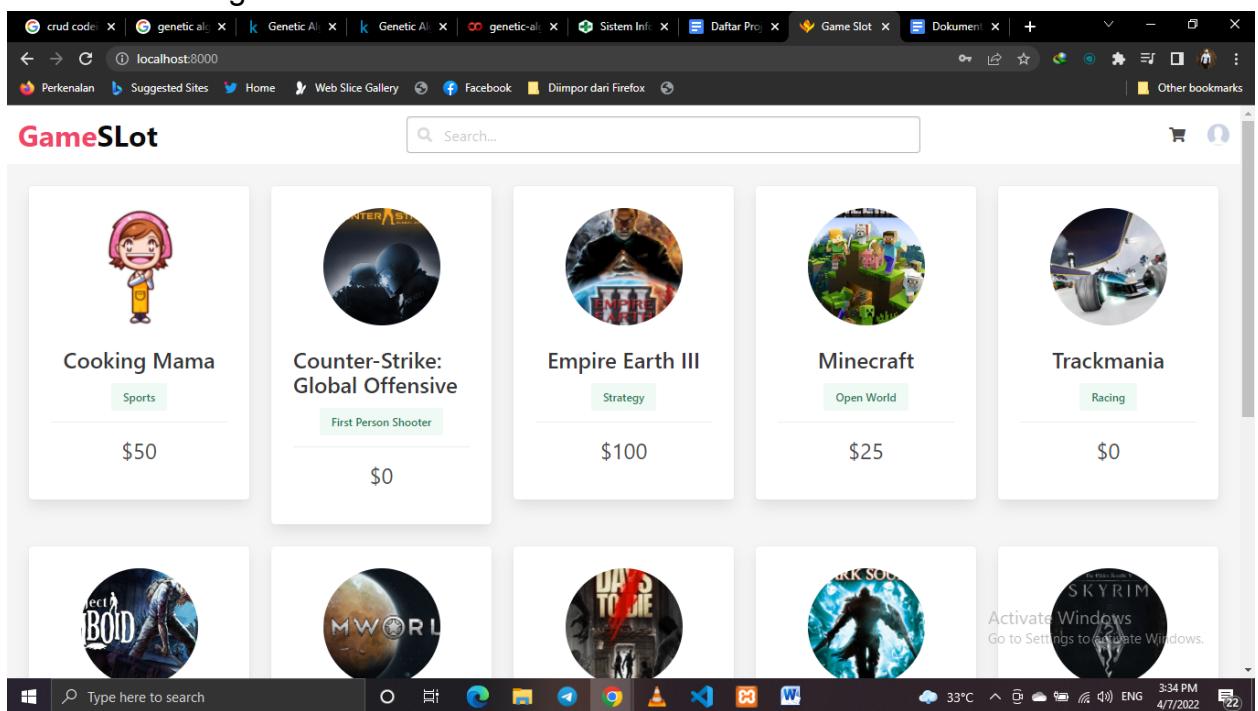
GameSlot merupakan website penjualan game online. Website ini menggunakan konsep MVC dengan framework laravel. Website ini dibangun bersama rekan saya untuk project pribadi.

### 1. Halaman Utama

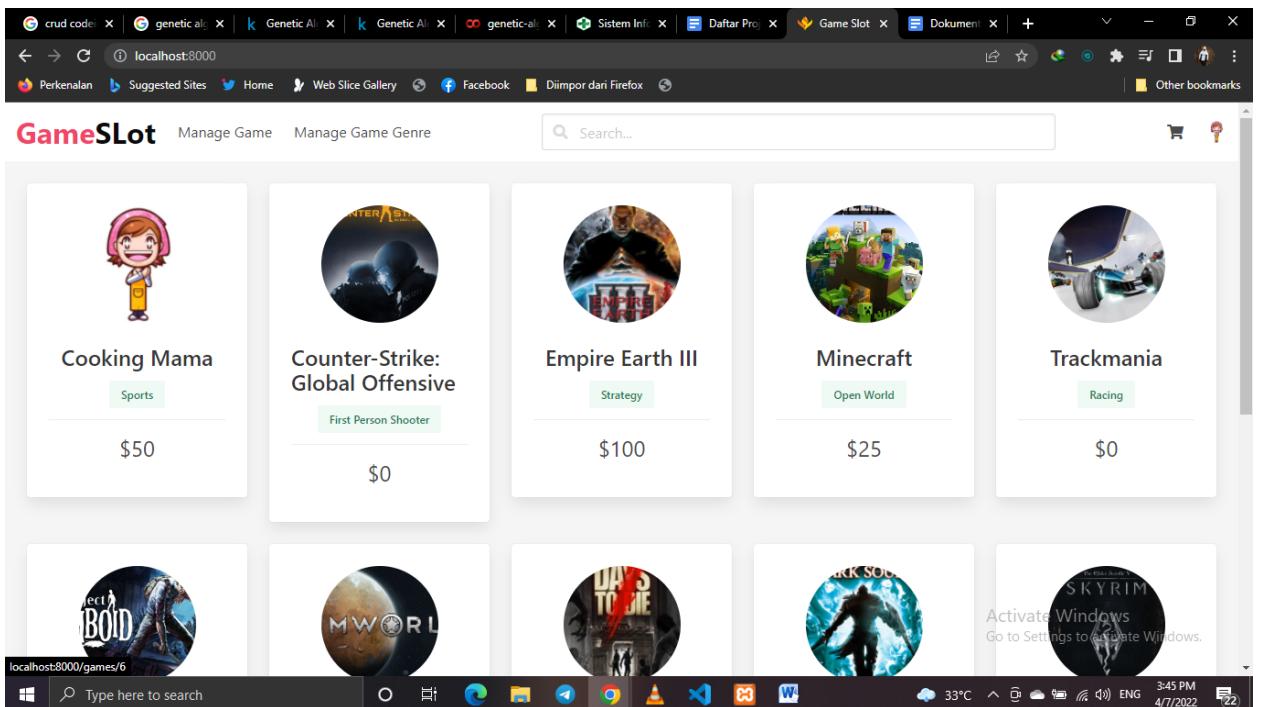
- Guest (User tanpa akun)



- User dengan Account



- Admin



- Berikut ini adalah controller dari tampilan Halaman Utama

```
class HomeController extends Controller{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct(){
        // $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index(){
        $userAge = -1;
        if(auth()->user()){
            $userAge = (int) date('Y') - (int) date('Y', strtotime(auth()->user()->dob));
        }
        // Get user age from DOB, if not there, set to 20
        if ($userAge < 0) {
            $userAge = 20;
        }
        // Get all games below user age and paginate
        $games = Game::where('pegi_rating', '<', $userAge)->paginate(10);
        $item_count = $games->count();
    }
}
```

```

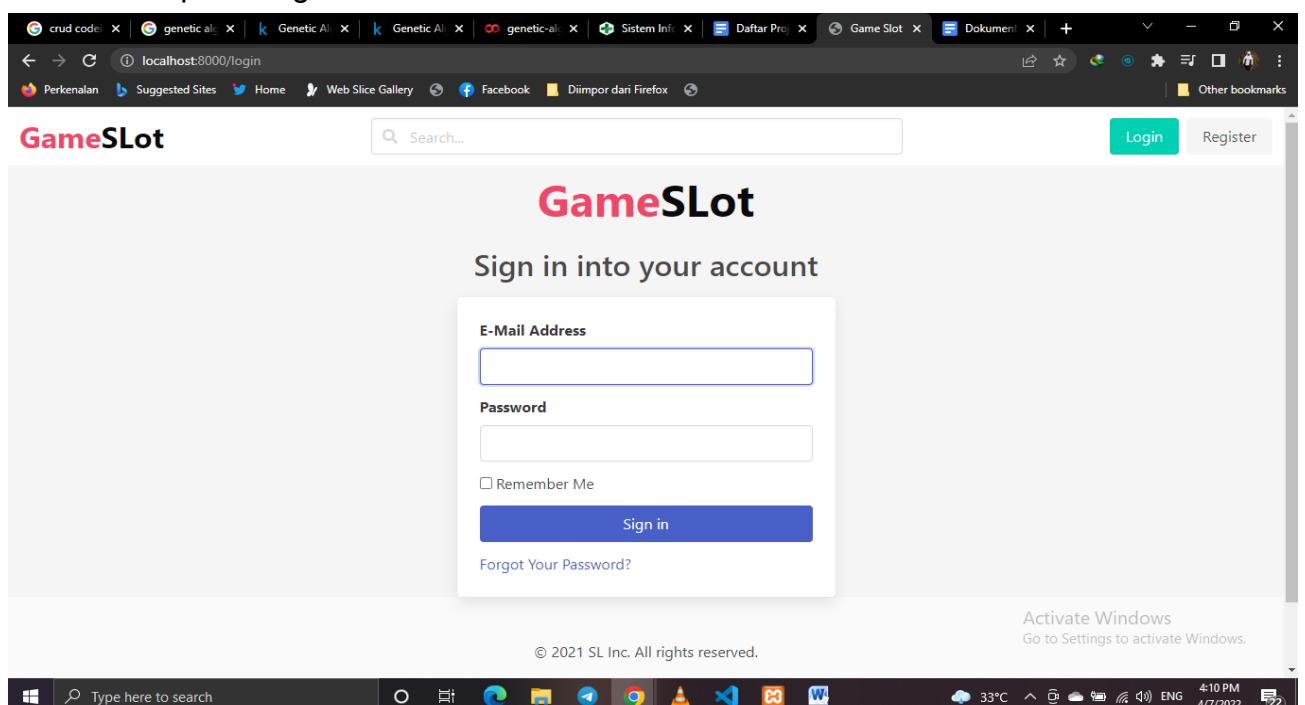
class HomeController extends Controller
{
    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        $userAge = -1;
        if(auth()->user()){
            $userAge = (int) date('Y') - (int) date('Y', strtotime(auth()->user()->dob));
        }
        // Get user age from DOB, if not there, set to 20
        if ($userAge < 0) {
            $userAge = 20;
        }
        // Get all games below user age and paginate
        $games = Game::where('pegi_rating', '<', $userAge)->paginate(10);
        $item_count = $games->count();
        $per_grid = 5;
        return view('home', compact('item_count', 'per_grid', 'games'));
    }

    /**
     * Search for games by keyword.
     *
     * @param Request $request
     * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
     */
    public function search(Request $request)
    {
        $keyword = $request->search;
        $games = Game::where('title', 'like', '%' . $keyword . '%')->paginate(10);
        $item_count = $games->count();
        $per_grid = 5;
        return view('home', compact('item_count', 'per_grid', 'games'));
    }
}

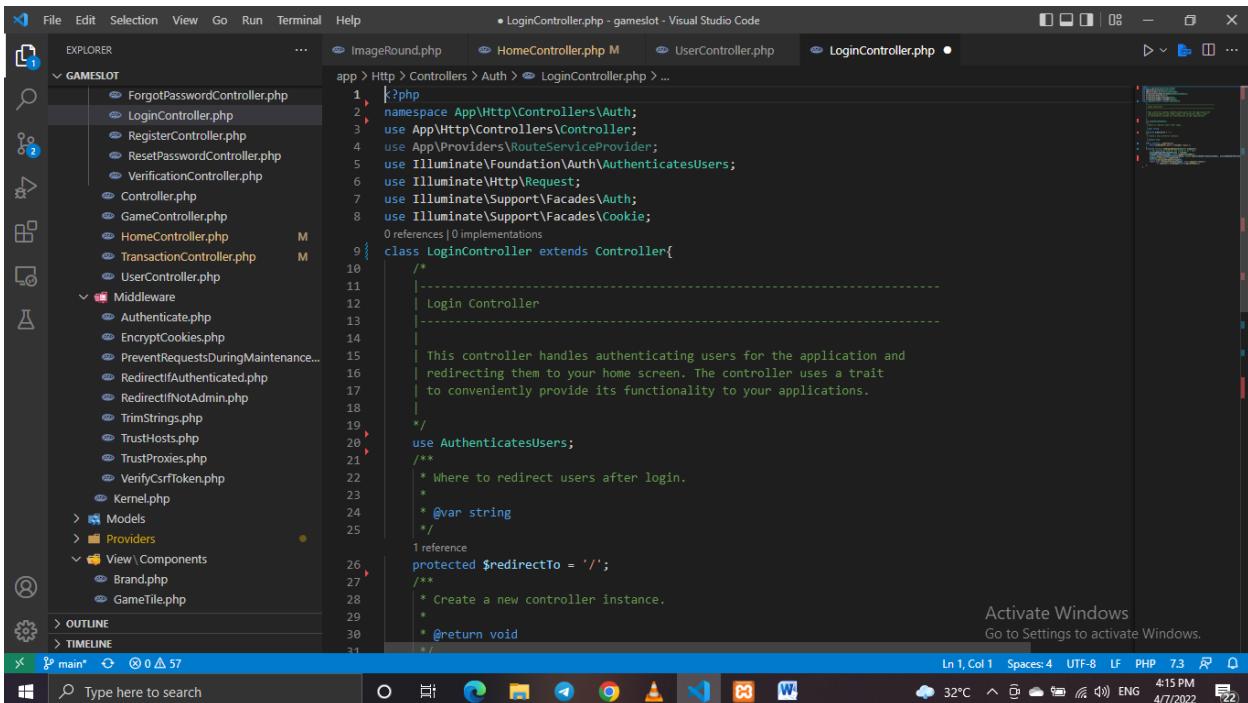
```

## 2. Login

### - Tampilan Login

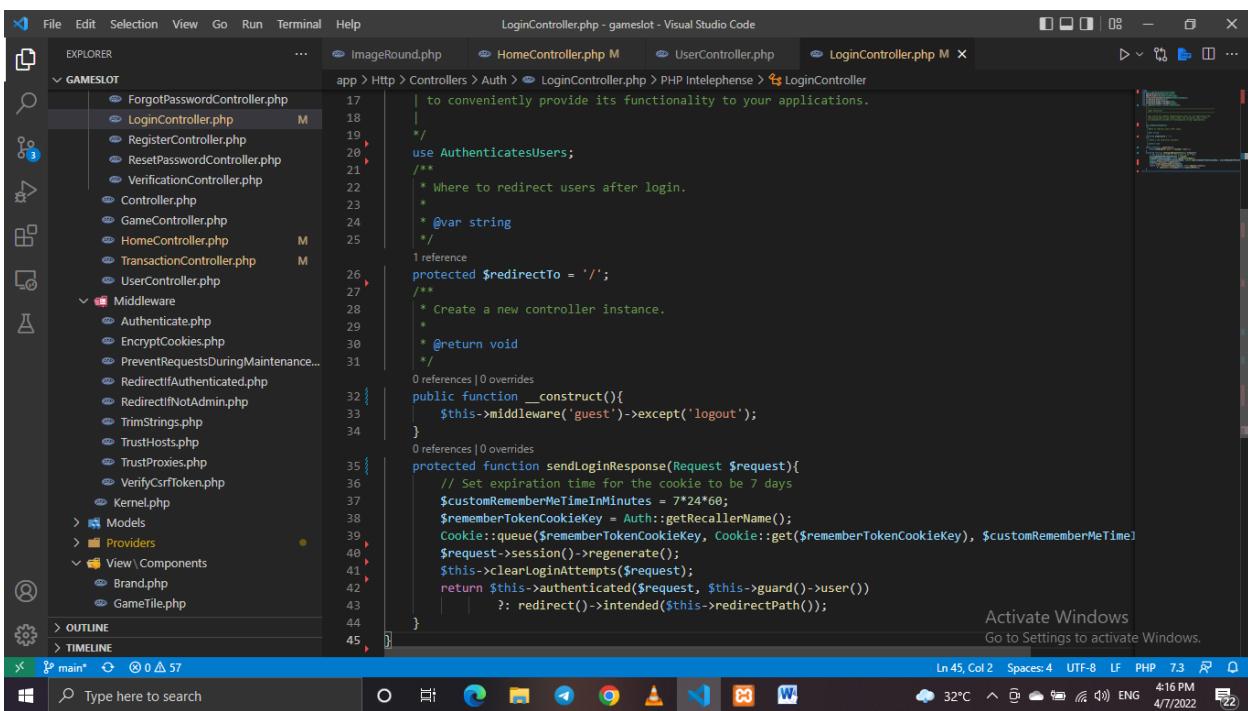


- Berikut ini merupakan controller dari tampilan Login



The screenshot shows the Visual Studio Code interface with the following details:

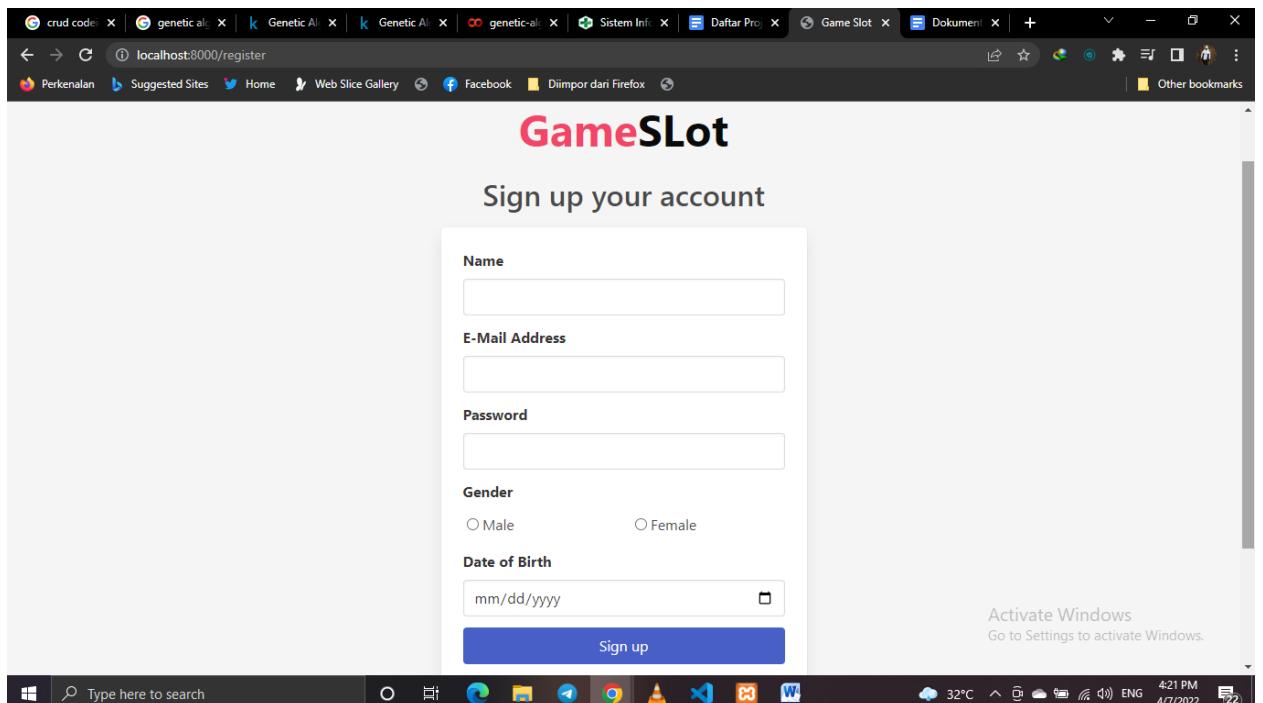
- File Explorer:** Shows the project structure under the "GAMESLOT" folder, including files like ForgotPasswordController.php, LoginController.php, RegisterController.php, etc.
- Code Editor:** Displays the content of LoginController.php. The code is a PHP class named LoginController that extends Controller. It includes comments explaining its purpose and usage of the Illuminate\Foundation\Auth\AuthenticatesUsers trait. It also defines protected properties \$redirectTo and \$guard, and a public \_\_construct() method that calls middleware('guest') except('logout').
- Status Bar:** Shows the file path as app > Http > Controllers > Auth > LoginController.php, the line count as Ln 1, Col 1, and other system information like temperature (32°C), battery level, and network status.



This screenshot shows the same Visual Studio Code session with a slightly different view of the LoginController.php code. The code editor now highlights the protected function sendLoginResponse(Request \$request) and its implementation, which sets cookie expiration and performs authentication logic.

### 3. Register

#### - Tampilan Register



#### - Controller dari tampilan Register

The screenshot shows the Visual Studio Code interface with the file 'RegisterController.php' open. The code is a PHP class for handling user registration. It includes methods for constructing the controller, validating input data, and creating a new user instance. The code uses annotations like '@param' and '@return' to describe the parameters and return types of its methods. The left sidebar shows the project structure with files like HomeController.php, GameController.php, and UserController.php. The status bar at the bottom provides information about the file: 'RegisterController.php - gameslot - Visual Studio Code', 'app > Http > Controllers > Auth > RegisterController.php', 'Line 61, Col 43', 'Spaces: 4', 'UTF-8', 'LF', 'PHP 7.3', and the date '4/7/2022'. There is also a watermark: 'Activate Windows Go to Settings to activate Windows.'

```
RegisterController.php - gameslot - Visual Studio Code
File Edit Selection View Go Run Terminal Help
HomeController.php M RegisterController.php M
app > Http > Controllers > Auth > RegisterController.php - PHP Intelphense > RegisterController > create
0 references | 0 overrides
public function __construct(){
    $this->middleware('guest');
}
/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
1 reference | 0 overrides
protected function validator(array $data){
    // Validate the user's input
    // Name should be alpha with spaces
    // Email should be valid and unique
    // Password should be at least 8 characters
    // Date of birth must be at max 13 years ago
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8'],
        'dob' => ['required', 'date', 'before:13 years ago'],
        'gender' => ['required', 'in:male,female']
    ]);
}
/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
72 references | 0 overrides
protected function create(array $data){}
```

```

RegisterController.php - gameslot - Visual Studio Code
File Edit Selection View Go Run Terminal Help
... HomeController.php M RegisterController.php M
app > Http > Controllers > Auth > RegisterController.php > PHP Intelphense > RegisterController > create
1 reference | 0 overrides
protected function validator(array $data){
    // Validate the user's input
    // Name should be alpha with spaces
    // Email should be valid and unique
    // Password should be at least 8 characters
    // Date of birth must be at max 13 years ago
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8'],
        'dob' => ['required', 'date', 'before:13 years ago'],
        'gender' => ['required', 'in:male,female']
    ]);
}
/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
72 references | 0 overrides
protected function create(array $data): User
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
        'dob' => $data['dob'],
        'gender' => $data['gender'],
    ]);
}

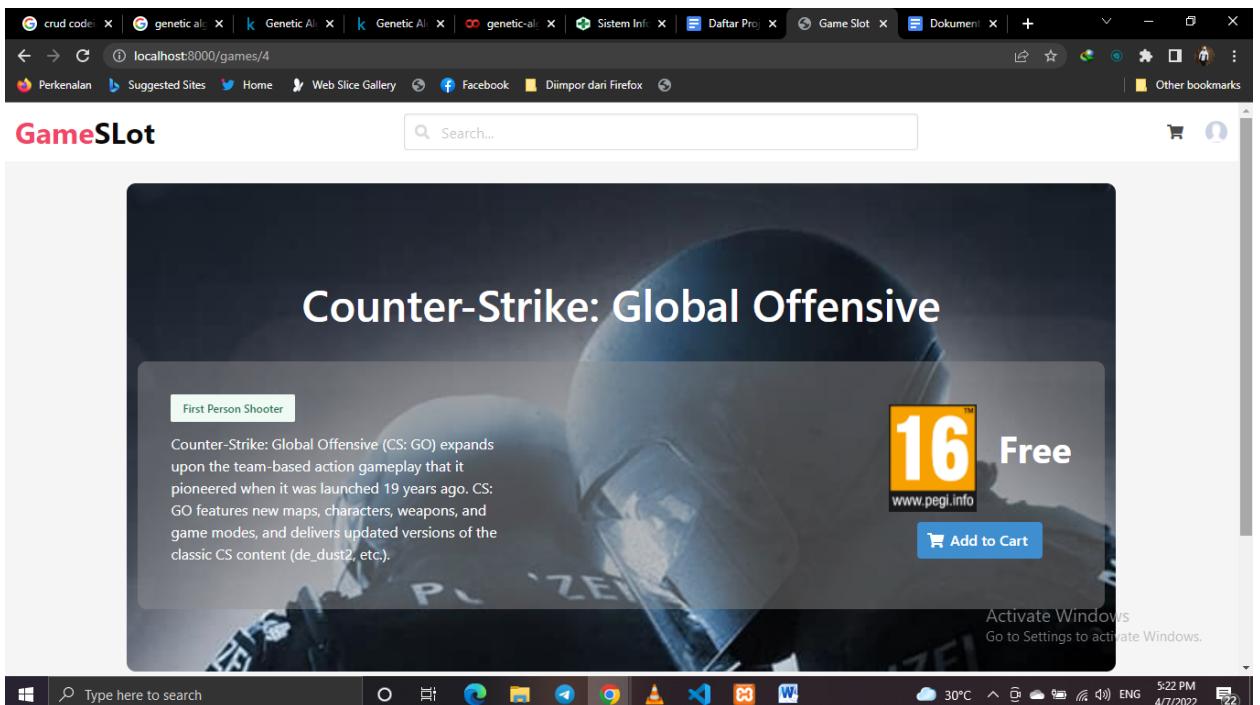
```

Activate Windows  
Go to Settings to activate Windows.

Ln 61 Col 43 Spaces: 4 UTF-8 LF PHP 7.3 4:22 PM 4/7/2022

#### 4. User

- Halaman Detail Game : Halaman ini bisa terbuka jika user sudah login. Jika ada user yang belum login atau ada guest yang megakses halaman ini, maka akan diarahkan ke halaman login



- Tetapi, jika mengklik tombol Add to Cart akan keluar notifikasi berwarna hijau seperti ini

The screenshot shows a browser window with the title 'Game Slot'. A green header bar displays the message 'Game added to cart successfully'. Below this, there is a grid of five game cards. Each card includes a small icon, the game title, a category tag, and a price. The games listed are Cooking Mama (Sports, \$50), Counter-Strike: Global Offensive (First Person Shooter, \$0), Empire Earth III (Strategy, \$100), Minecraft (Open World, \$25), and Trackmania (Racing, \$0). The Windows taskbar at the bottom shows various pinned icons and the date/time as 4/7/2022 5:28 PM.

- Cart : Menu cart atau keranjang berfungsi untuk menampung game yang akan dibeli. Lalu klik checkout untuk melakukan pembayaran

The screenshot shows a browser window with the title 'Game Slot'. A green header bar displays the message 'Cart updated successfully'. Below this, there is a table representing the contents of the cart. The table has columns for 'Game Title', 'Game Price', and 'Quantity'. It contains one item: 'Counter-Strike: Global Offensive' with a price of 'Free' and a quantity of '1'. To the right of the table is a blue 'Checkout' button. The Windows taskbar at the bottom shows various pinned icons and the date/time as 4/7/2022 5:30 PM.

- Setelah di klik checkout, muncullah bukti transaksi

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "localhost:8000/transaction/12". The page content is as follows:

**GameSLot**

Transaction successful!

Transaction ID: 12      Transaction Date: 2022-04-07 10:34:08

Customer Name: F

Game Title	Game Price	Quantity	Subtotal
Counter-Strike: Global Offensive	\$0	1	\$0

Total: \$0

© 2021 SL Inc. All rights reserved.

Activate Windows  
Go to Settings to activate Windows.

Windows taskbar at the bottom with various icons and system status.

## 5. Admin

- Halaman Manage Game : Halaman ini berfungsi untuk menampilkan game yang sudah terdaftar di database

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "localhost:8000/manage-game". The page content is as follows:

**GameSLot** Manage Game    Manage Game Genre

+ Add game

Game Title	PEGI Rating	Game Genre	Game Price
Cooking Mama	7+ Kids	Sports	50
Counter-Strike: Global Offensive	16+ Mature	First Person Shooter	Free
Empire Earth III	12+ Teen	Strategy	100
Minecraft	0+ Everyone	Open World	25
Trackmania	0+ Everyone	Racing	Free
Project Zomboid	18+ Adult	Zombies	7
Rimworld	12+ Teen	Simulation	5

Activate Windows  
Go to Settings to activate Windows.

Windows taskbar at the bottom with various icons and system status.

- Halaman Add Game : Halaman ini berfungsi untuk menambahkan game pada database

Add Game

Game Title

Photo

Choose File No file chosen

Game Description

Game Price

0

Game Genre

Select a genre or write a new one

PEGI Rating

Everyone

Activate Windows  
Go to Settings & Update Windows.

- Halaman Edit Game : Berfungsi untuk menambahkan atau mengganti informasi pada game

Edit Game

Game Title

Cooking Mama

Photo

Choose File No file chosen

Game Description

Cooking Mama is a Japanese video game series and media franchise owned by Cooking Mama Limited. The series is a cookery simulation-styled minigame compilation of many video games and adventures for Nintendo gaming platforms

Game Price

50

Game Genre

Sports

PEGI Rating

7+ Kids

Activate Windows  
Go to Settings & Update Windows.

- Halaman Manage Game Genre : Halaman ini berfungsi untuk menampilkan daftar genre game yang tercantum di database

The screenshot shows a web browser window with the URL [localhost:8000/manage-game-genre](http://localhost:8000/manage-game-genre). The page title is "Game Slot". The main content area is titled "Game Genre" and lists the following genres:

Genre	Count	Action
Something	0	<a href="#">Edit</a>
Adventure	0	<a href="#">Edit</a>
Casual	0	<a href="#">Edit</a>
Indie	0	<a href="#">Edit</a>
MMO	0	<a href="#">Edit</a>
Racing	1	<a href="#">Edit</a>
RPG	3	<a href="#">Edit</a>
Simulation	1	<a href="#">Edit</a>
Sports	1	<a href="#">Edit</a>

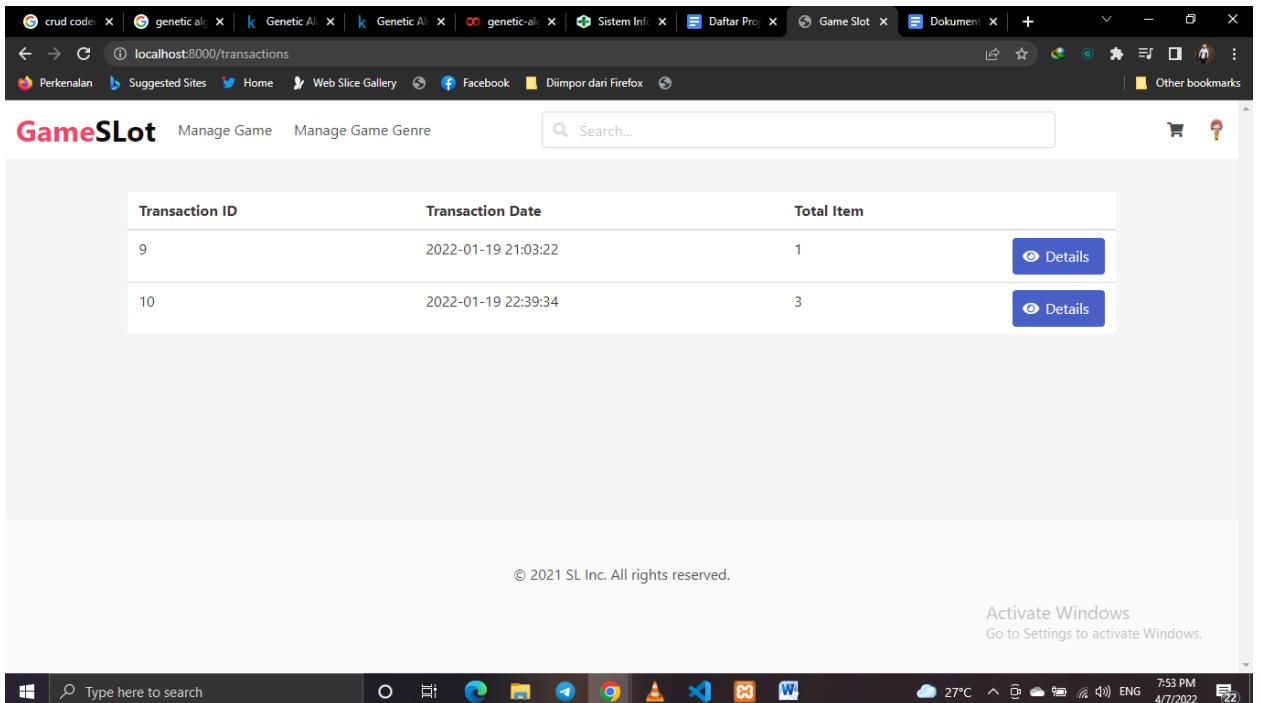
At the bottom right of the page, there is a message: "Activate Windows Go to Settings to activate Windows." The browser's status bar shows the date and time as 4/7/2022 6:12 PM.

- Halaman Update Game Genre : Berfungsi untuk mengedit genre game.

The screenshot shows a web browser window with the URL [localhost:8000/manage-game-genre/](http://localhost:8000/manage-game-genre/). The page title is "Game Slot". The main content area is titled "Update Genre" and contains a single input field labeled "Name" with the value "Adventure". Below the input field is a blue "Update" button.

At the bottom of the page, there is a copyright notice: "© 2021 SL Inc. All rights reserved." and a message: "Activate Windows Go to Settings to activate Windows." The browser's status bar shows the date and time as 4/7/2022 7:51 PM.

- Halaman Transaction : Halaman ini berfungsi untuk menampilkan daftar transaksi yang dilakukan

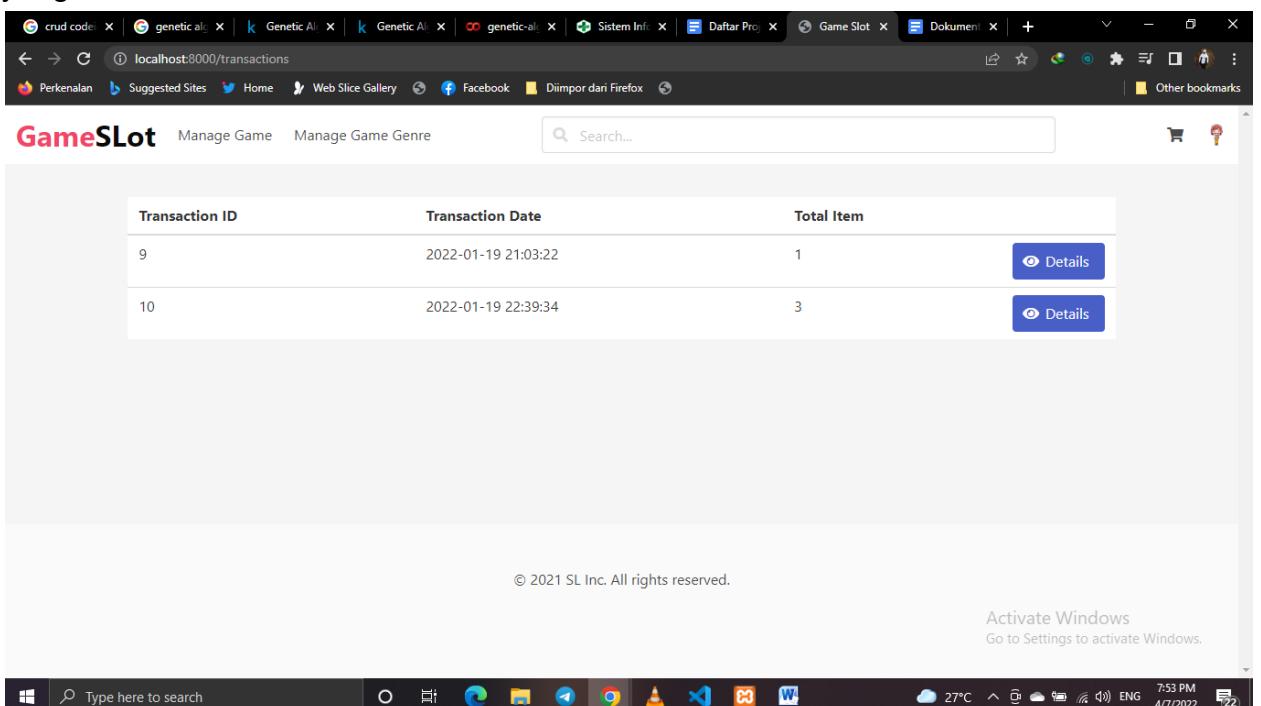


The screenshot shows a web browser window with the URL [localhost:8000/transactions](http://localhost:8000/transactions). The page title is "GameSlot". The main content is a table with three columns: "Transaction ID", "Transaction Date", and "Total Item". There are two rows of data:

Transaction ID	Transaction Date	Total Item
9	2022-01-19 21:03:22	1
10	2022-01-19 22:39:34	3

Each row has a blue "Details" button on the right. At the bottom of the page, there is a copyright notice: "© 2021 SL Inc. All rights reserved." and an "Activate Windows" watermark.

- Halaman Transaction History : Berfungsi untuk menampilkan detail transaksi yang dilakukan oleh user



The screenshot shows a web browser window with the URL [localhost:8000/transactions](http://localhost:8000/transactions). The page title is "GameSlot". The main content is a table with three columns: "Transaction ID", "Transaction Date", and "Total Item". There are two rows of data:

Transaction ID	Transaction Date	Total Item
9	2022-01-19 21:03:22	1
10	2022-01-19 22:39:34	3

Each row has a blue "Details" button on the right. At the bottom of the page, there is a copyright notice: "© 2021 SL Inc. All rights reserved." and an "Activate Windows" watermark.

- Source Code GameController

The screenshot shows the Visual Studio Code interface with the file `GameController.php` open. The code implements a `GameController` class that extends `Controller`. It contains methods for creating and storing new games, validating requests, and handling file uploads. The code uses namespaces like `App\Http\Controllers` and `Illuminate\Http\Request`.

```

<?php
namespace App\Http\Controllers;
use App\Models\Game;
use App\Models\GameGenre;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
class GameController extends Controller{
    /**
     * Show the form for creating a new game.
     * @return \Illuminate\Http\Response
     */
    public function create(){
        $gameGenres = GameGenre::all();
        return view('games.create', compact('gameGenres'));
    }
    /**
     * Store a newly created game in storage.
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request){
        // Validate the request
        $request->validate([
            'title' => 'required|string',
            'image' => 'required|image',
            'description' => 'required|string',
            'price' => 'required|numeric',
            'genre' => 'required|string',
            'pegi_rating' => 'required|in:0,3,7,12,16,18',
        ]);
        // Check the genre
        $gameGenre = GameGenre::where('name', $request->genre)->first();
        if (!$gameGenre) {
            $gameGenre = GameGenre::create([
                'name' => $request->genre,
            ]);
        }
        // Handle the image
        $image = $request->file('image');
        $imageName = time() . '.' . $image->getClientOriginalExtension();
        $image->move(public_path('images/game'), $imageName);
        // Create the game
        $game = new Game();
        $game->title = $request->title;
        $game->description = $request->description;
        $game->price = $request->price;
        $game->pegi_rating = $request->pegi_rating;
        $game->genre_id = $gameGenre->id;
        $game->image = $imageName;
        if ($game->save()) {
            return redirect()->route('manage.game')->with('success', 'Game created successfully');
        }
        return redirect()->route('manage.game.create')->with('error', 'Something went wrong');
    }
    /**
     * Display the specified game edit form
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id){
        $game = Game::find($id);
    }
}

```

The screenshot continues the `GameController.php` code. It includes a method for editing an existing game, which retrieves the game by its ID. The code uses the `findOrFail` method to handle cases where the game does not exist.

```

public function edit($id){
    $game = Game::findOrFail($id);
}

```

File Edit Selection View Go Run Terminal Help

GameController.php - gameslot - Visual Studio Code

app > Http > Controllers > GameController.php > PHP Tools > App\Http\Controllers\GameController > update

```
public function edit($id){
    $game = Game::find($id);
    $gameGenres = GameGenre::all();
    return view('games.edit', compact('game', 'gameGenres'));
}

/**
 * Update the specified game in storage.
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */

23 references | 0 overrides

public function update(Request $request, $id){
    // Validate the request
    $request->validate([
        'title' => 'required|string',
        'description' => 'required|string',
        'price' => 'required|numeric',
        'genre' => 'required|string',
        'pegi_rating' => 'required|in:0,3,7,12,16,18',
    ]);
    // Check the genre
    $gameGenre = GameGenre::where('name', $request->genre)->first();
    if (!$gameGenre) {
        $gameGenre = GameGenre::create([
            'name' => $request->genre,
        ]);
    }
    $game = Game::find($id);
    // Update the game
    $game->title = $request->title;
    $game->description = $request->description;
    $game->price = $request->price;
    $game->pegi_rating = $request->pegi_rating;
    $game->genre_id = $gameGenre->id;
    // If the user has uploaded a new image
    if ($request->hasFile('image')) {
        $request->validate([
            'image' => 'image|mimes:jpeg,png,jpg,svg',
        ]);
        // Delete the old image
        $image_path = public_path('images/game/' . $game->image);
        if (file_exists($image_path)) {
            @unlink($image_path);
        }
        // Handle the image
        $image = $request->file('image');
        $imageName = time() . '.' . $image->getClientOriginalExtension();
        $image->move(public_path('images/game'), $imageName);
        $game->image = $imageName;
    }
    if ($game->save()) {
        return redirect()->route('manage.game')->with('success', 'Game updated successfully');
    }
    return redirect()->route('manage.game.edit', $id)->with('error', 'Something went wrong');
}

/**
 * Show the specified game.
 * @param int $id
 * @return \Illuminate\Http\Response
 */

23 references | 0 overrides
```

Activate Windows  
Go to Settings to activate Windows.

Ln 71, Col 5 Spaces: 4 UTF-8 LF PHP 7.3 7:59 PM 4/7/2022

File Edit Selection View Go Run Terminal Help

GameController.php - gameslot - Visual Studio Code

app > Http > Controllers > GameController.php > PHP Tools > App\Http\Controllers\GameController > update

```
$game->title = $request->title;
$game->description = $request->description;
$game->price = $request->price;
$game->pegi_rating = $request->pegi_rating;
$game->genre_id = $gameGenre->id;
// If the user has uploaded a new image
if ($request->hasFile('image')) {
    $request->validate([
        'image' => 'image|mimes:jpeg,png,jpg,svg',
    ]);
    // Delete the old image
    $image_path = public_path('images/game/' . $game->image);
    if (file_exists($image_path)) {
        @unlink($image_path);
    }
    // Handle the image
    $image = $request->file('image');
    $imageName = time() . '.' . $image->getClientOriginalExtension();
    $image->move(public_path('images/game'), $imageName);
    $game->image = $imageName;
}
if ($game->save()) {
    return redirect()->route('manage.game')->with('success', 'Game updated successfully');
}
return redirect()->route('manage.game.edit', $id)->with('error', 'Something went wrong');

/**
 * Show the specified game.
 * @param int $id
 * @return \Illuminate\Http\Response
 */

2 references | 0 overrides
```

Activate Windows  
Go to Settings to activate Windows.

Ln 71, Col 5 Spaces: 4 UTF-8 LF PHP 7.3 8:00 PM 4/7/2022

File Edit Selection View Go Run Terminal Help GameController.php - gameslot - Visual Studio Code

EXPLORER app > Http > Controllers > GameController.php M GameController.php M TransactionController.php M

```
public function updateGameGenre(Request $request, $id){  
    // Validate the request  
    $request->validate([  
        'name' => 'required|string|unique:game_genres,name,' . $id,  
    ]);  
    // Get game genre from database  
    $gameGenre = GameGenre::find($id);  
    // Update the game genre  
    $gameGenre->name = $request->name;  
    if ($gameGenre->save()) {  
        return redirect()->route('manage.game.genre')->with('success', 'Game genre updated successfully');  
    }  
    return redirect()->route('game-genres.edit', ['id' => $id])->with('error', 'Game genre could not be updated');  
}  
/**  
 * Delete game  
 * @param int $id  
 * @return \Illuminate\Http\Response  
 */  
public function destroy($id){  
    // Get game from database  
    $game = Game::find($id);  
    // Delete the game  
    if ($game->delete()) {  
        return redirect()->route('manage.game')->with('success', 'Game deleted successfully');  
    }  
    return redirect()->route('manage.game')->with('error', 'Game could not be deleted');  
}  
/**  
 * Add game to cart  
 * @param int $id  
 * @return \Illuminate\Http\Response  
 */  
public function cart($id){  
    // If user is not logged in, redirect to login page  
    if (!Auth::check()) {  
        return redirect()->route('login');  
    }  
    // Get game from database  
    $game = Game::find($id);  
    // Add the game to cart in session  
    $cart = session()->get('cart');  
    if (!$cart) {  
        $cart = [  
            $id => 1  
        ];  
        session()->put('cart', $cart);  
        return redirect()->route('home')->with('success', 'Game added to cart successfully');  
    }  
    if (isset($cart[$id])) {  
        $cart[$id]++;  
        session()->put('cart', $cart);  
        return redirect()->route('home')->with('success', 'Game added to cart successfully');  
    }  
    $cart[$id] = 1;  
    session()->put('cart', $cart);  
    return redirect()->route('home')->with('success', 'Game added to cart successfully');  
}
```

Activate Windows  
Go to Settings to activate Windows.

Ln 130 Col 40 Spaces: 4 UTF-8 LF PHP 7.3 8:55 PM 4/7/2022

File Edit Selection View Go Run Terminal Help GameController.php - gameslot - Visual Studio Code

EXPLORER app > Http > Controllers > GameController.php M GameController.php M TransactionController.php M

```
public function cart($id){  
    // If user is not logged in, redirect to login page  
    if (!Auth::check()) {  
        return redirect()->route('login');  
    }  
    // Get game from database  
    $game = Game::find($id);  
    // Add the game to cart in session  
    $cart = session()->get('cart');  
    if (!$cart) {  
        $cart = [  
            $id => 1  
        ];  
        session()->put('cart', $cart);  
        return redirect()->route('home')->with('success', 'Game added to cart successfully');  
    }  
    if (isset($cart[$id])) {  
        $cart[$id]++;  
        session()->put('cart', $cart);  
        return redirect()->route('home')->with('success', 'Game added to cart successfully');  
    }  
    $cart[$id] = 1;  
    session()->put('cart', $cart);  
    return redirect()->route('home')->with('success', 'Game added to cart successfully');  
}
```

Activate Windows  
Go to Settings to activate Windows.

Ln 130 Col 40 Spaces: 4 UTF-8 LF PHP 7.3 8:56 PM 4/7/2022

- Source Code TransactionController

```

<?php
namespace App\Http\Controllers;
use App\Models\Game;
use App\Models\Transaction;
use App\Models\TransactionItem;
use Illuminate\Http\Request;
class TransactionController extends Controller{
    public function index(Request $request){
        $transactions = $request->user()->transactions()->get();
        return view('transaction.index', compact('transactions'));
    }
    /**
     * Show transaction details
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id){
        $transaction = Transaction::with('user')->with('items')->find($id);
        // Check if user is owner of transaction
        if ($transaction->user_id != $request->user()->id) {
            return redirect()->route('home')->with('error', 'You are not allowed to view this transaction');
        }
        return view('transaction.show', compact('transaction'));
    }
    /**
     * Show cart items from session
     */
    public function cart(){
        $cart = session()->get('cart');
    }
}

```

```

* Show cart items from session
*/
public function cart(){
    $cart = session()->get('cart');
    // If cart is empty
    if (empty($cart)) {
        $cart = [];
    }
    // Iterate through $cart keys and get the corresponding game
    $game = [];
    foreach ($cart as $gameId => $quantity) {
        $game[$gameId] = Game::find($gameId);
    }
    return view('transaction.cart', compact('game', 'cart'));
}
/**
 * Edit cart items from session
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit_cart(Request $request, $id){
    // Validate quantity
    $request->validate([
        'quantity' => 'required|numeric|min:1',
    ]);
    $cart = session()->get('cart');
    $cart[$id] = $request->quantity;
    session()->put('cart', $cart);
    return redirect()->route('user.cart')->with('success', 'Cart updated successfully');
}

```

```
public function delete_cart($id){
    $cart = session()->get('cart');
    unset($cart[$id]);
    session()->put('cart', $cart);
    return redirect()->route('user.cart')->with('success', 'Cart item removed successfully');
}

// Checkout form handling
public function store(Request $request){
    // Get the cart items from session
    $cart = session()->get('cart');
    // Create a new transaction
    $transaction = new Transaction();
    $transaction->user_id = auth()->user()->id;
    if ($transaction->save()) {
        try {
            // Create a new transaction item for each cart item
            foreach ($cart as $id => $quantity) {
                $game = Game::find($id);
                $transaction_item = new TransactionItem();
                $transaction_item->transaction_id = $transaction->id;
                $transaction_item->title = $game->title;
                $transaction_item->price = $game->price;
                $transaction_item->quantity = $quantity;
                $transaction_item->save();
            }
            // Delete the cart items
            session()->forget('cart');
            // Redirect to home with success message
            return redirect()->route('user.transaction.detail', ['id' => $transaction->id])->with('success', 'Transaction successful! Your items have been added to the transaction.');
        } catch (\Exception $e) {
            // Rollback the transaction
            // $transaction->delete();
            // Redirect back to the cart with an error message
            return redirect()->route('user.cart')
                ->with('error', 'Transaction failed! Please try again.');
        }
    }
    return redirect()->route('cart');
}
```

```
$cart = session()->get('cart');
// Create a new transaction
$transaction = new Transaction();
$transaction->user_id = auth()->user()->id;
if ($transaction->save()) {
    try {
        // Create a new transaction item for each cart item
        foreach ($cart as $id => $quantity) {
            $game = Game::find($id);
            $transaction_item = new TransactionItem();
            $transaction_item->transaction_id = $transaction->id;
            $transaction_item->title = $game->title;
            $transaction_item->price = $game->price;
            $transaction_item->quantity = $quantity;
            $transaction_item->save();
        }
        // Delete the cart items
        session()->forget('cart');
        // Redirect to home with success message
        return redirect()->route('user.transaction.detail', ['id' => $transaction->id])->with('success', 'Transaction successful! Your items have been added to the transaction.');
    } catch (\Exception $e) {
        // Rollback the transaction
        // $transaction->delete();
        // Redirect back to the cart with an error message
        return redirect()->route('user.cart')
            ->with('error', 'Transaction failed! Please try again.');
    }
}
return redirect()->route('cart');
```