

**ITCS212**

**Web Programming**

By

Mr. Sittiphon Lerdtomolsakul 6488037

Mr. Kitiphum Mueangthongkham 6488123

Mr. Purinat Pattanakeaw 6488051

Mr. Thanaphat Ritluechai 6488193

Mr. Pattaradnai Kaeodumkoeng 6488217

Present to

Aj. Wudhichart Sawangphol ([wudhichart.saw@mahidol.edu](mailto:wudhichart.saw@mahidol.edu))

Aj. Jidapa Kraisangka ([jidapa.kra@mahidol.edu](mailto:jidapa.kra@mahidol.edu))

Aj. Pisit Praiwattana ([pisit.pra@mahidol.edu](mailto:pisit.pra@mahidol.edu))

Faculty of Information and Communication Technology

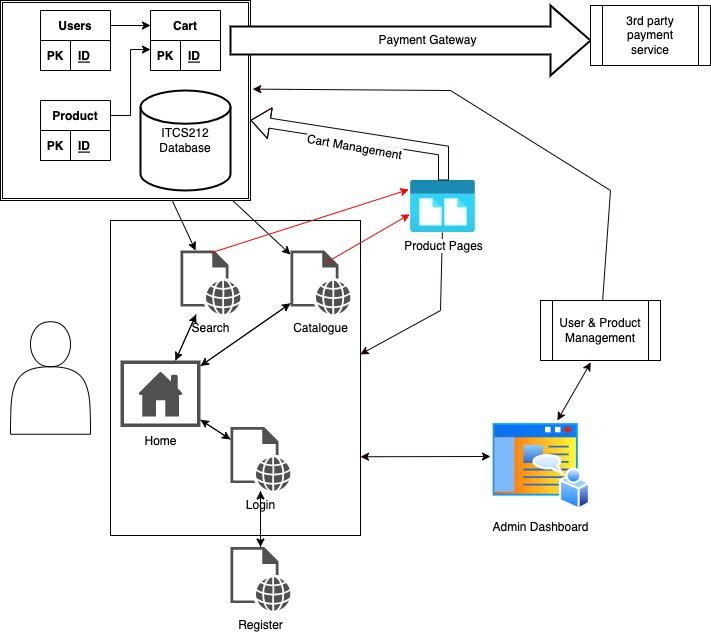
Mahidol University

2022

**1.Project Overview**

The web application is a game store website where users can buy game products. Users will need to login before entering the website. Once they login successfully, users will be able to browse game products, add them to their cart, and proceed to checkout. Admins will have a privilege, allowing them to manage the product and user information from the dashboard. The Product/Service Management page will allow admins to add new game products, modify existing game products, and delete specific game products. The User Account Management page will allow admins to add new user accounts, modify existing user accounts, and delete user accounts. The website will also have an authentication page for admins to login using their admin account. The search page allows users to search for game products based on criteria including product title, product genre, and its creator. Lastly, the website also has an "About Us" page to showcase information about the team behind the website. Overall, the goal of this project is to create a user-friendly and secure game store website that provides an enjoyable shopping experience for users while giving admins full control over the product and user information.

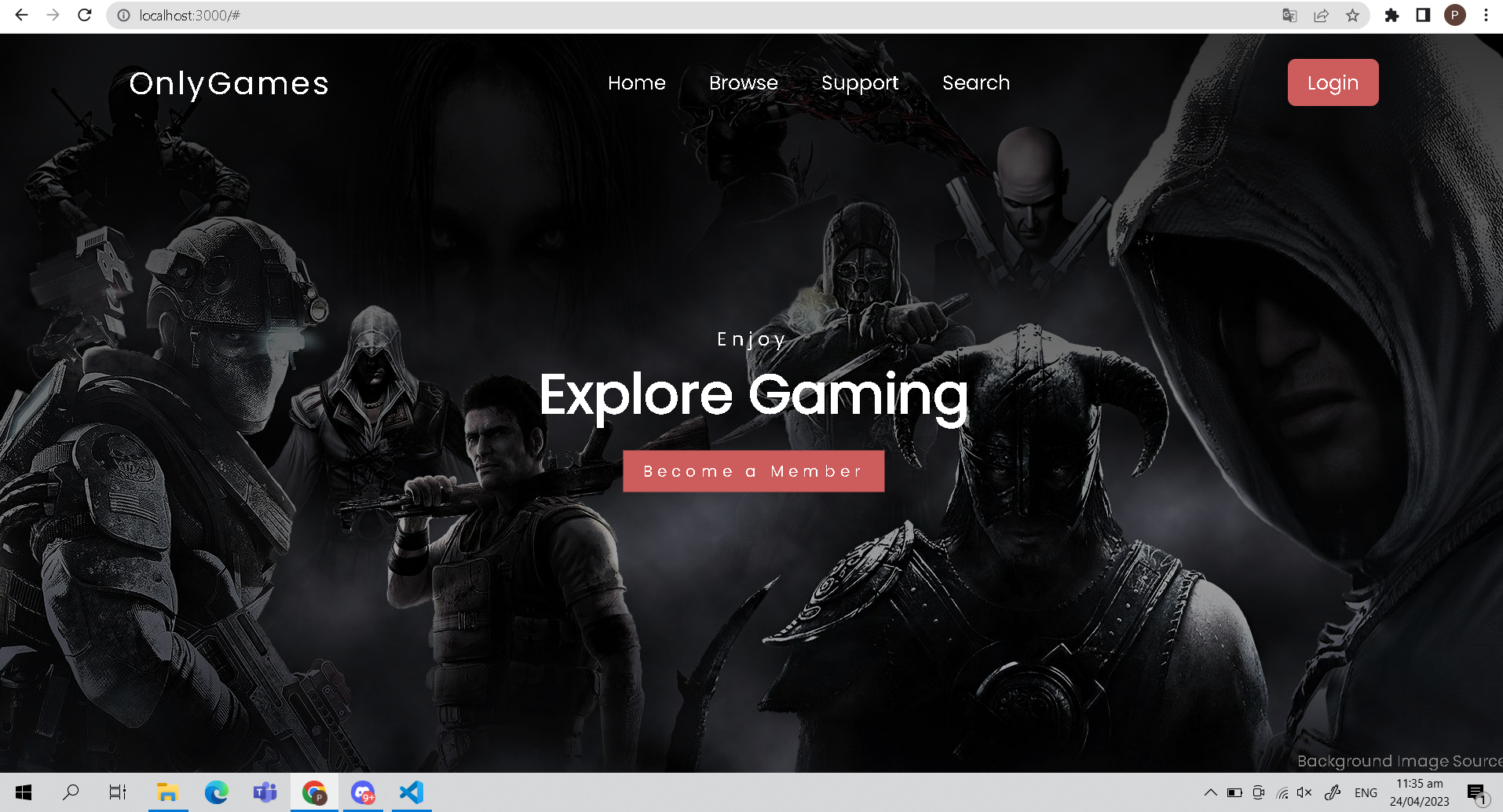
**2. Navigation Diagram of application**



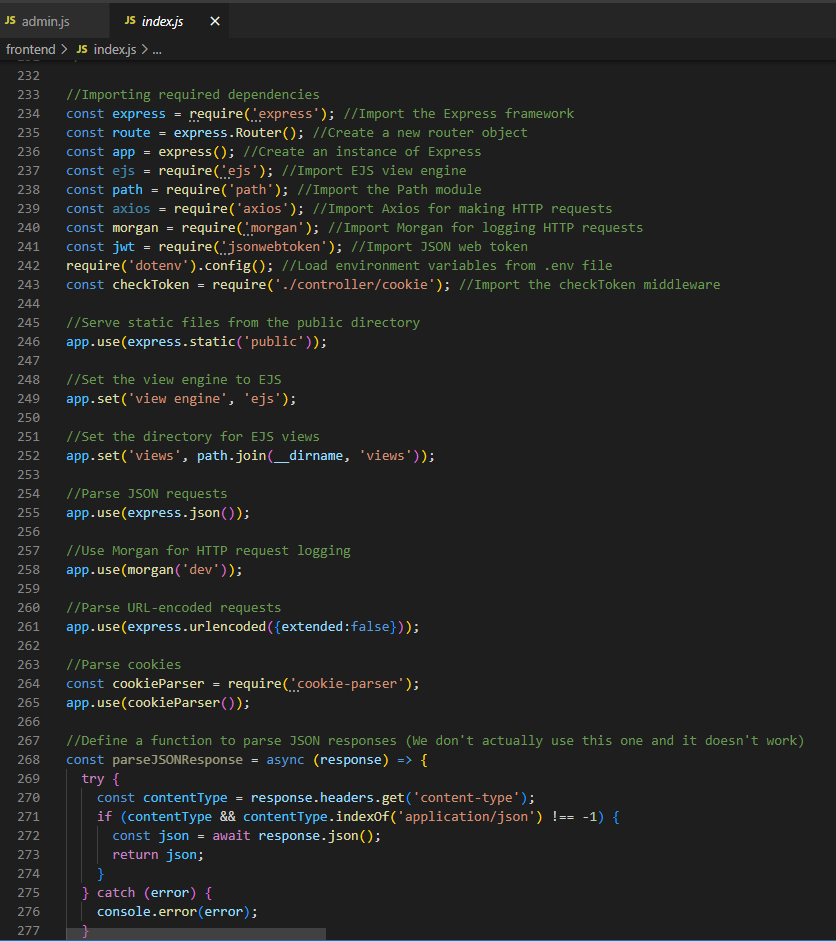
Web Routing

The program requires users to register and log in to access pages like cart and dashboard. All users can use the search page and catalog page to access detailed product pages, which are linked to a database containing the user's ID, Cart's ID, and Product's ID. The admin has the ability to manage both product and user information.

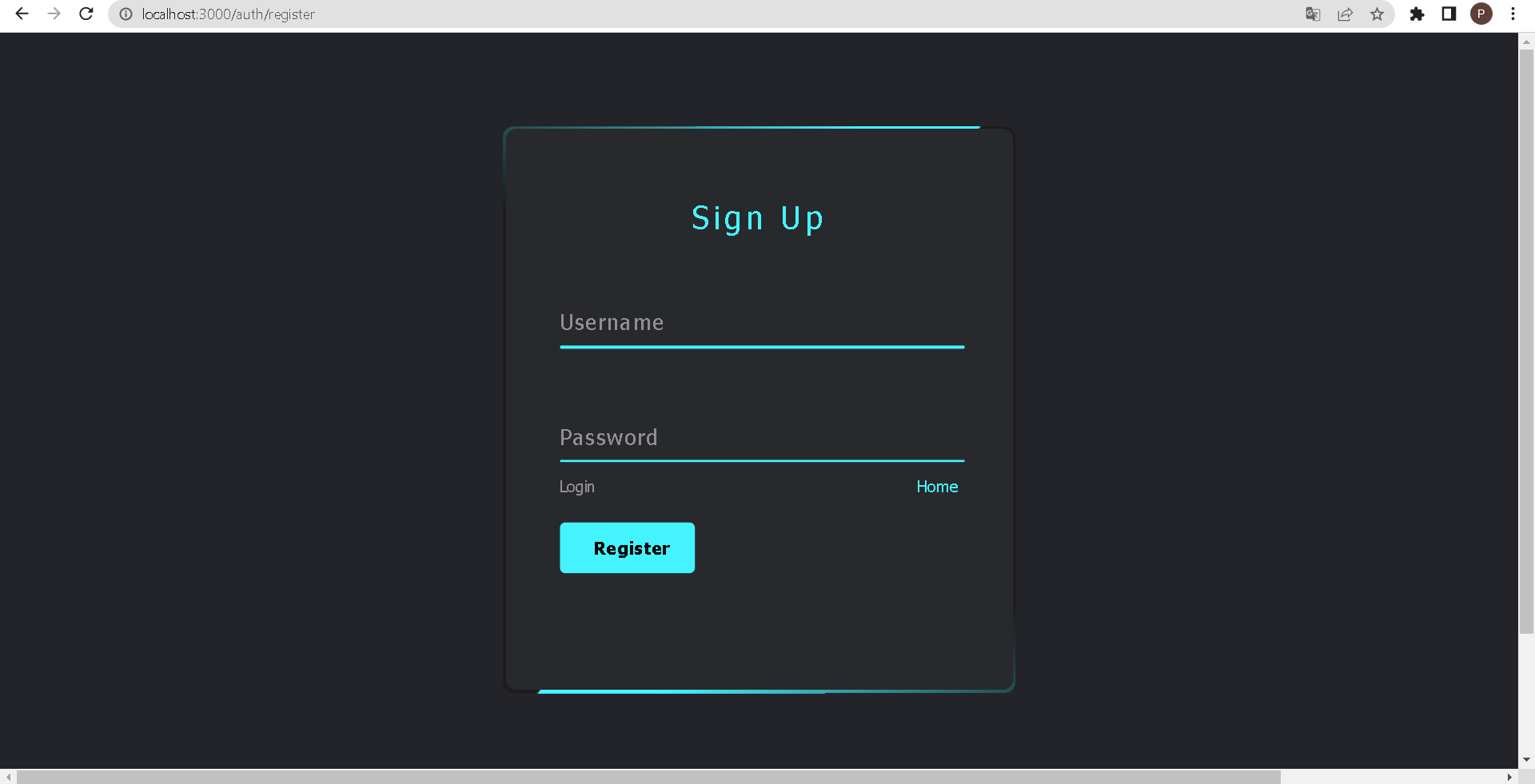
**3. Details of application and code**

****

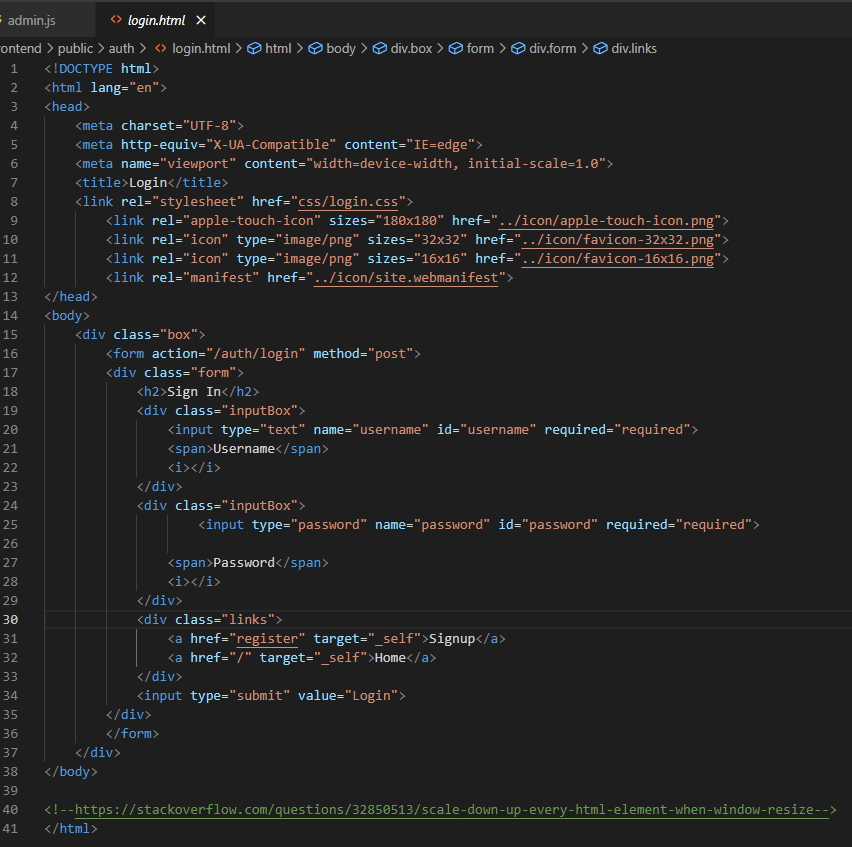
**Home page**

****

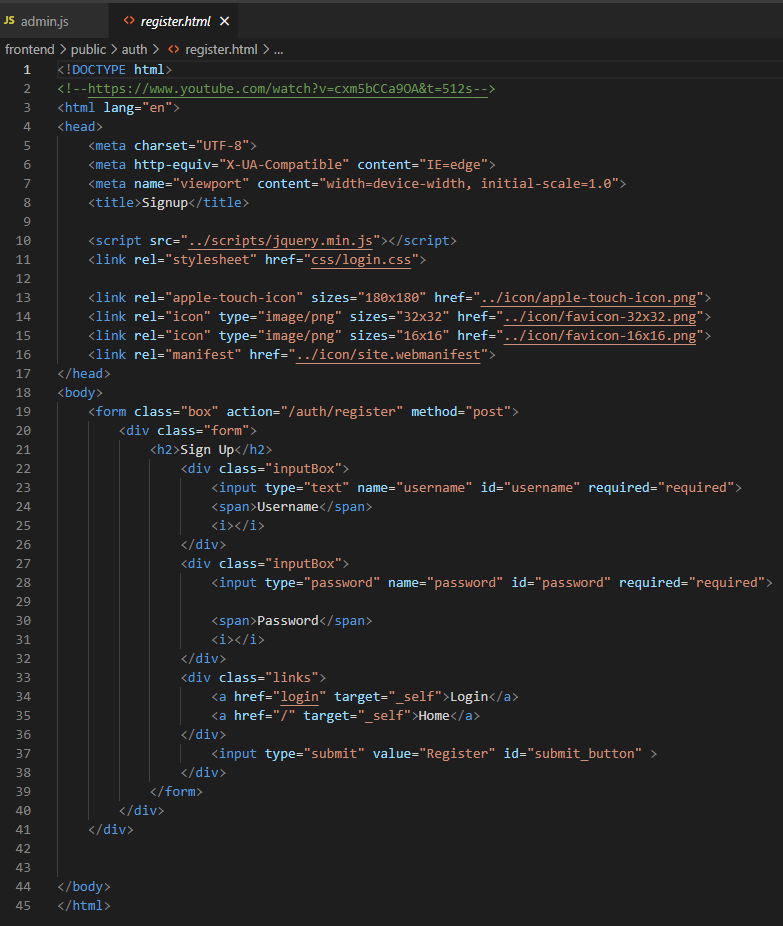
**frontend/Index.js**

****

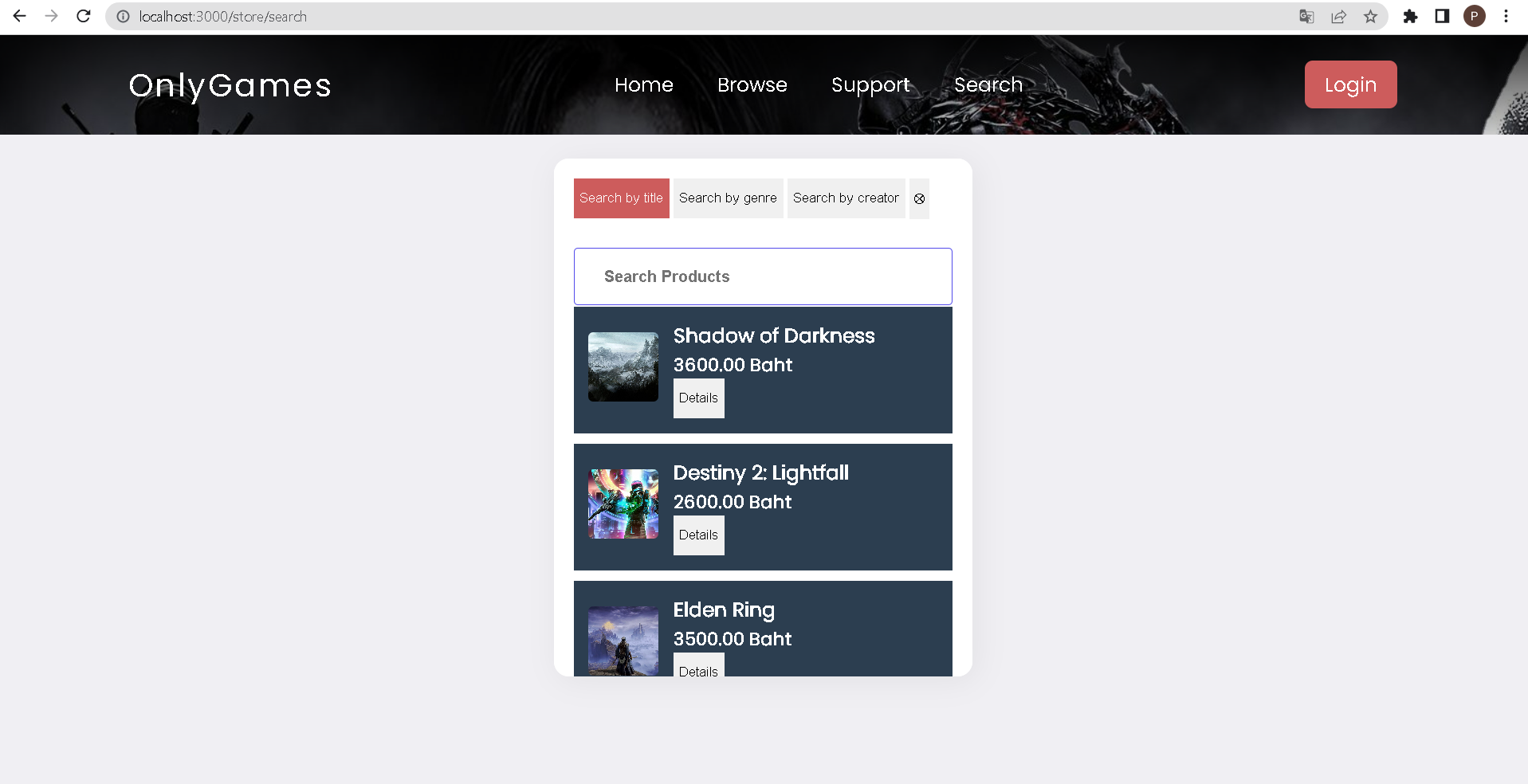
**Login page for Administrators**

****

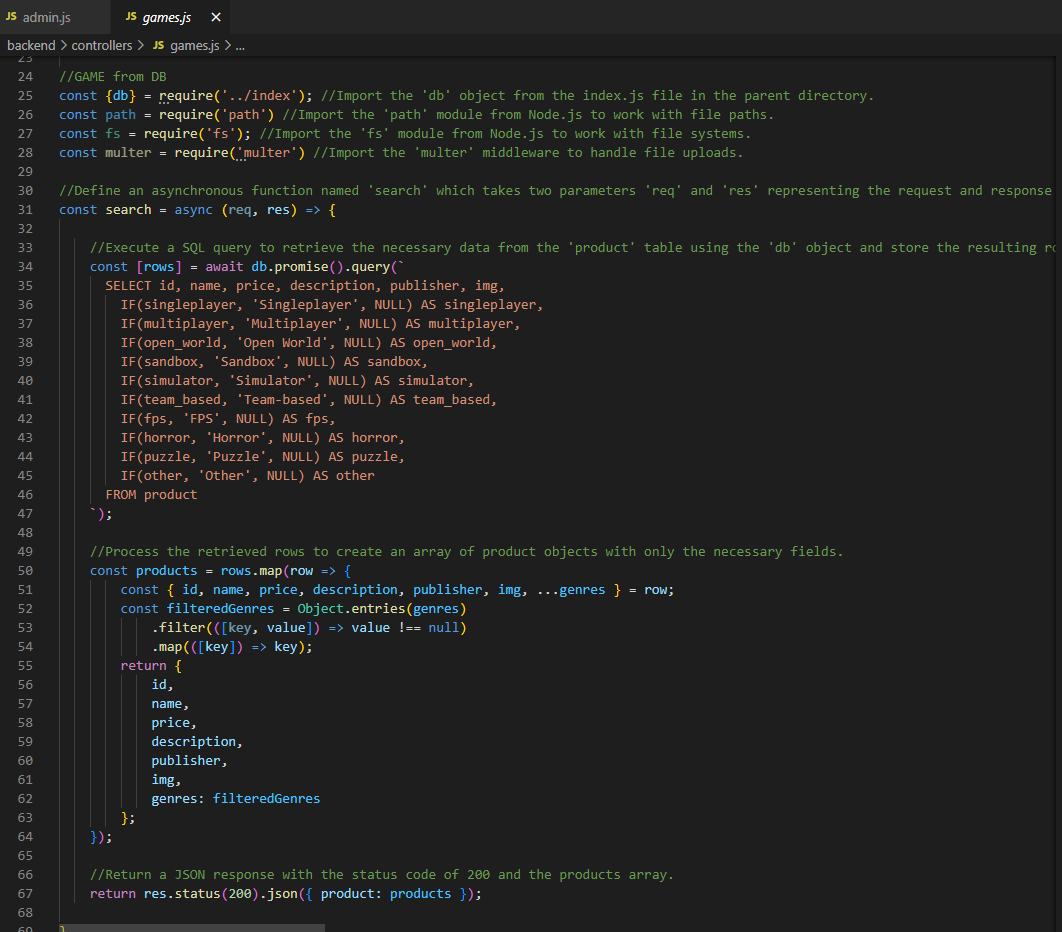
**Login.html**

****

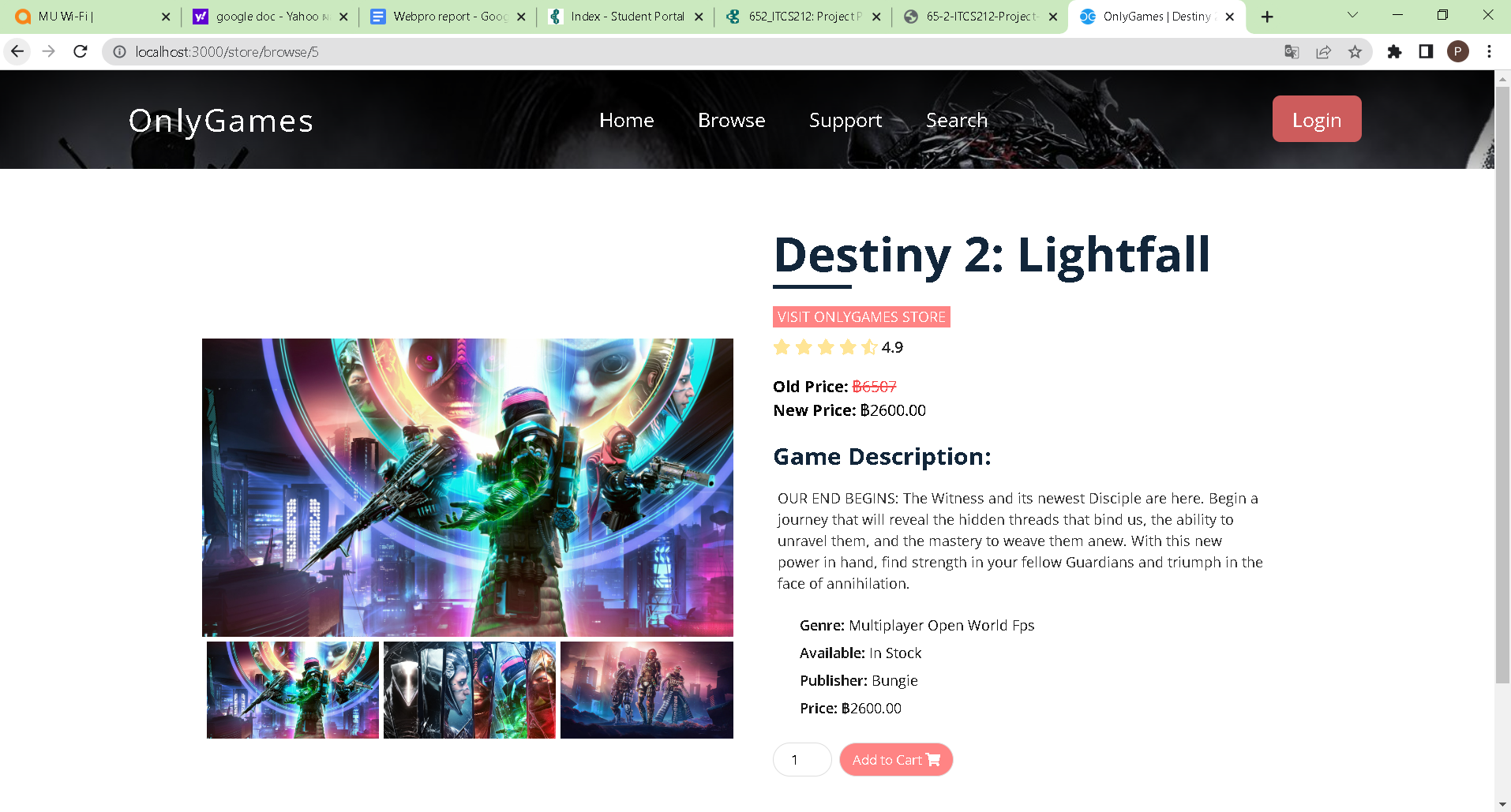
**register.html**

****

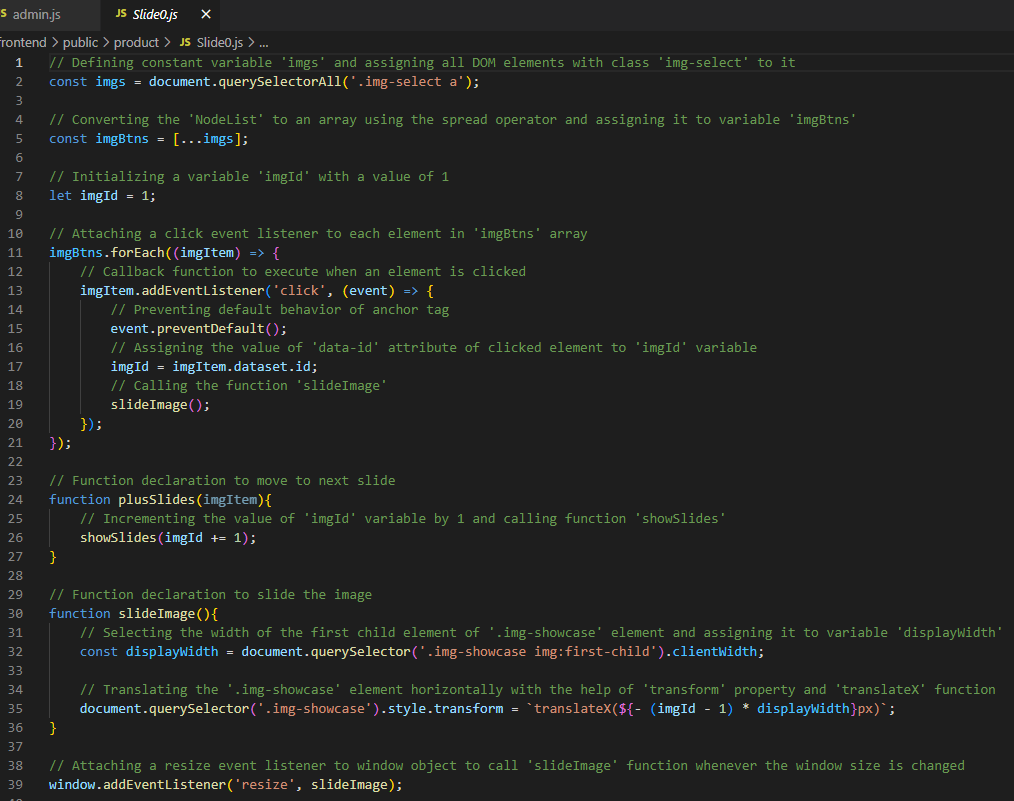
**Search page for searching product/services with criteria searchings**

****

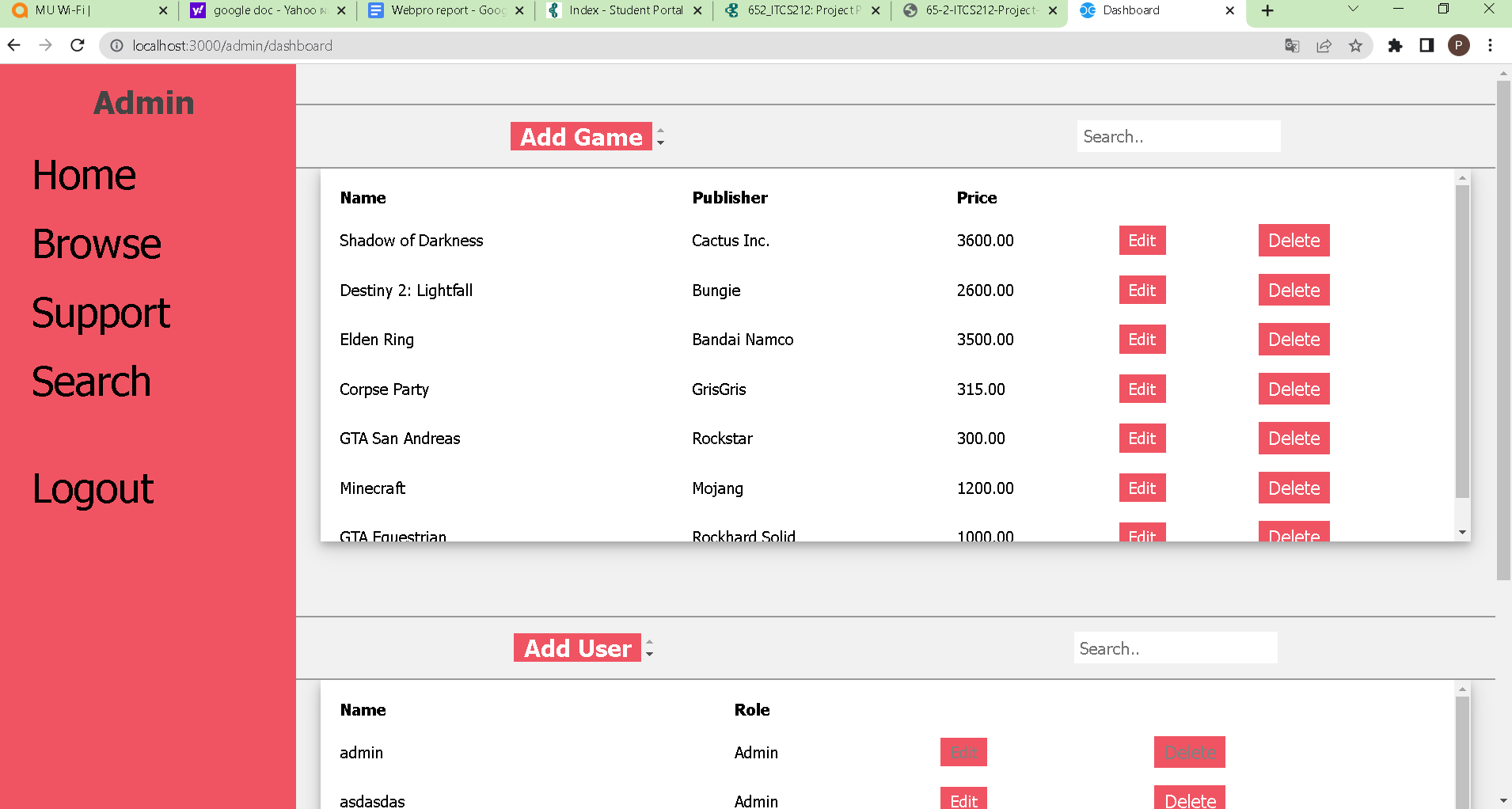
**Game.js ( Both Catalog and Product detail pages)**

****

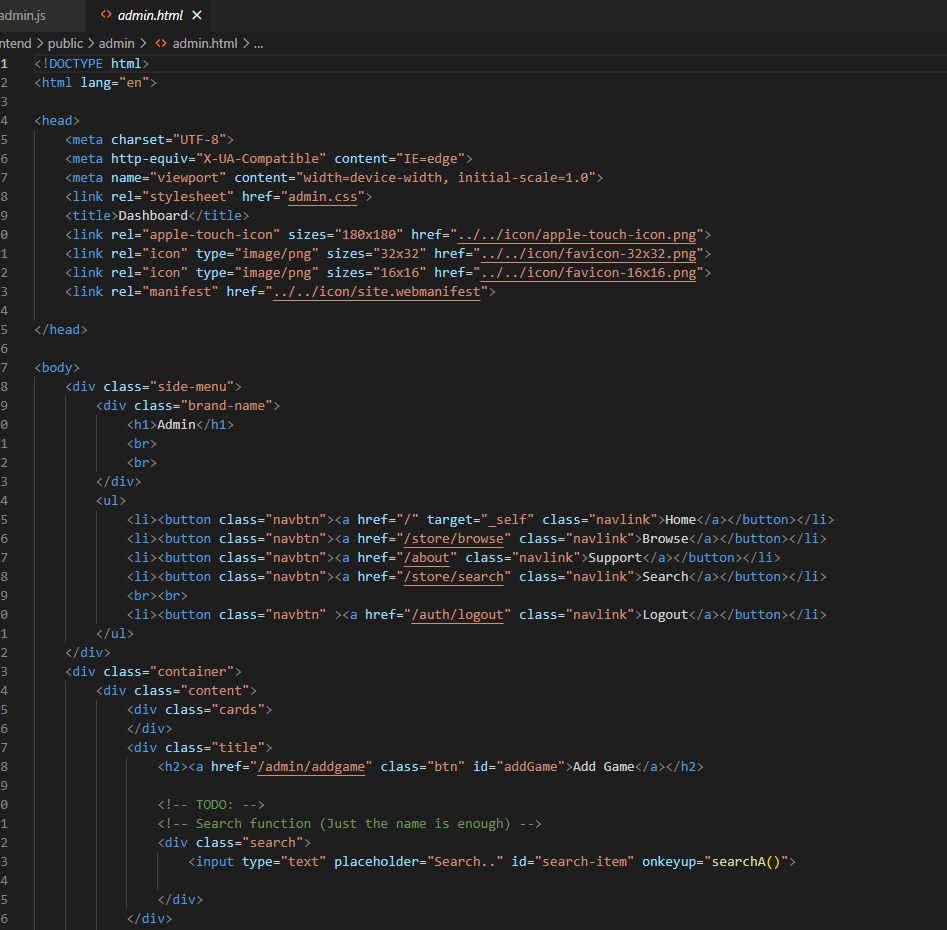
**Detail page for each product/service result**

****

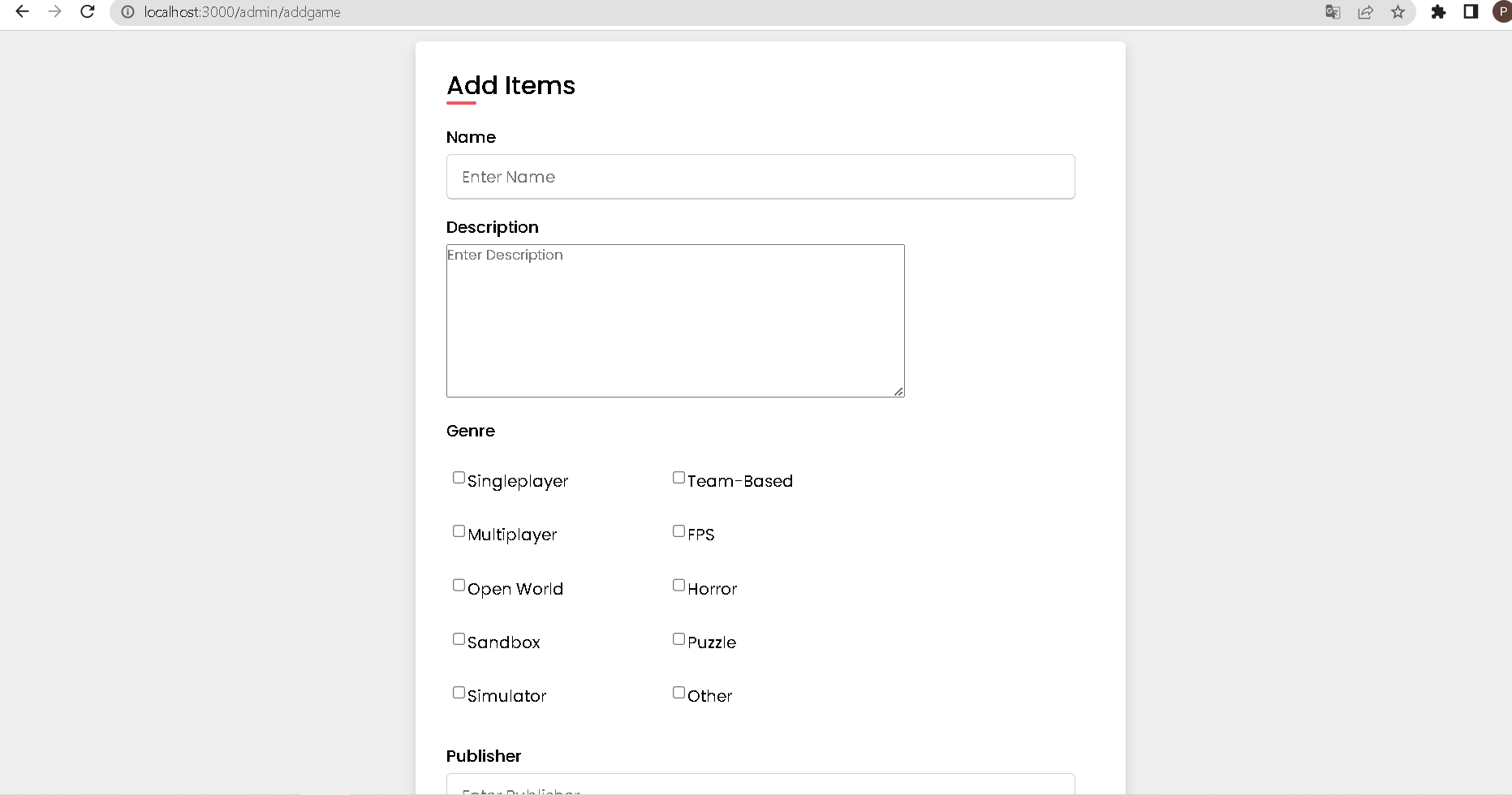
**Slider on Product page**

****

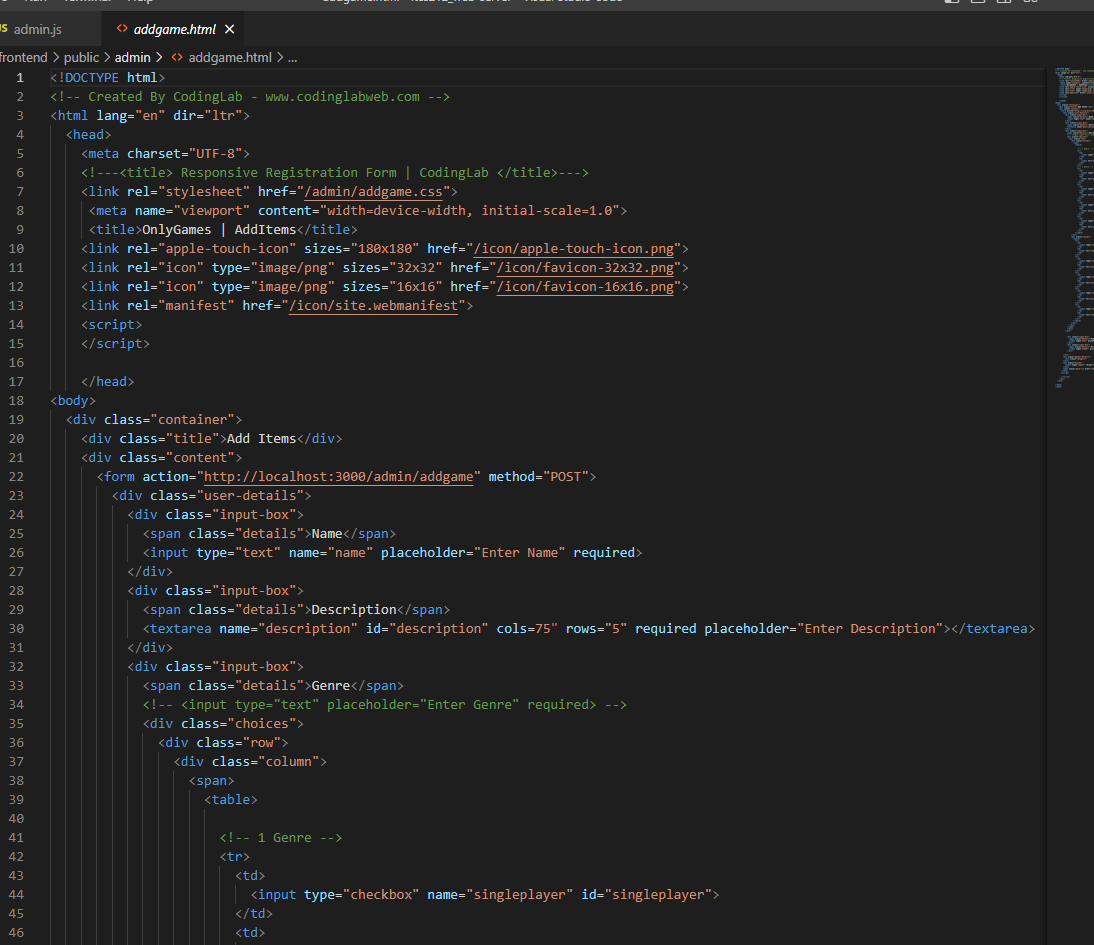
**Dashboard**

****

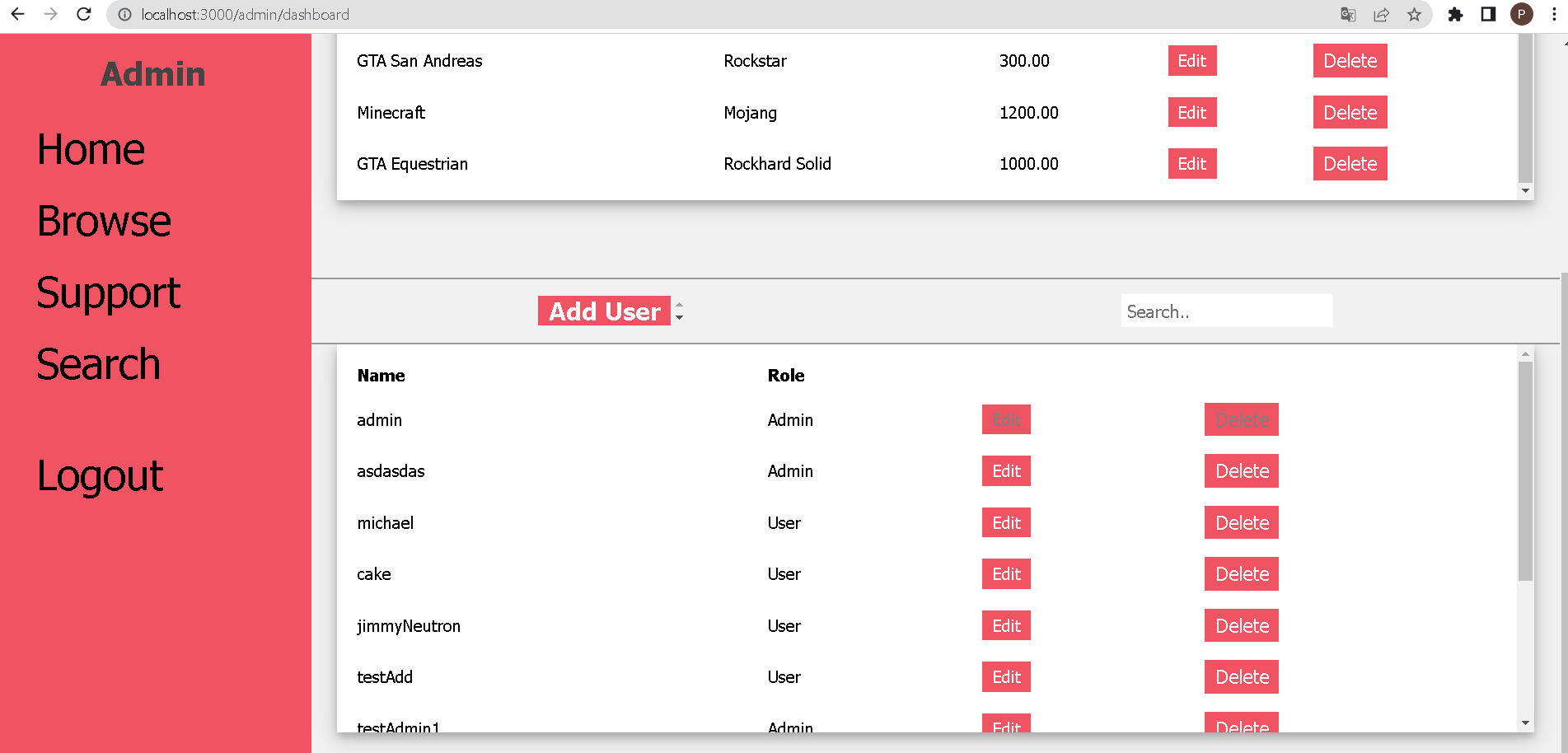
**admin.html**

****

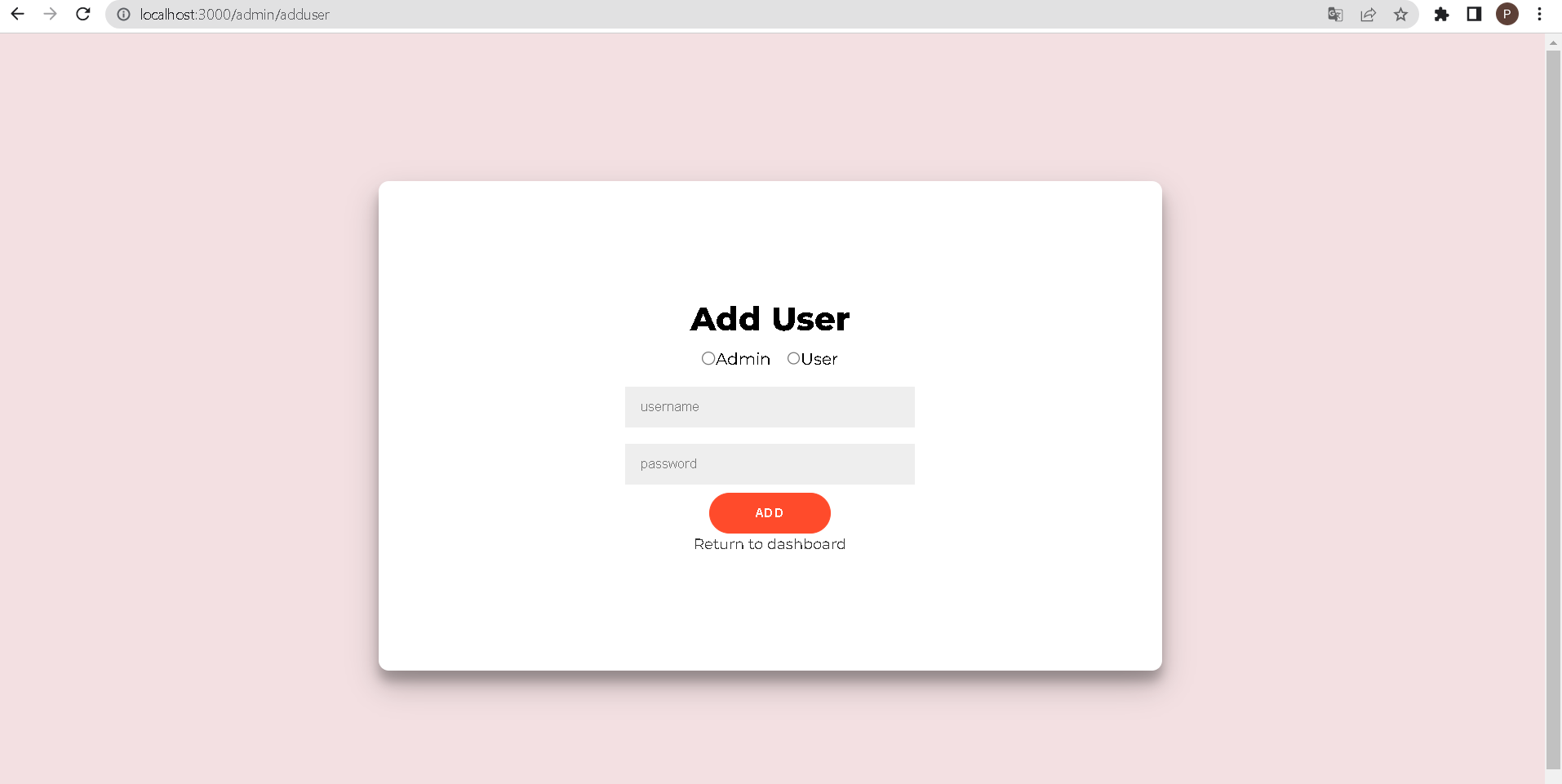
**Add item page**

****

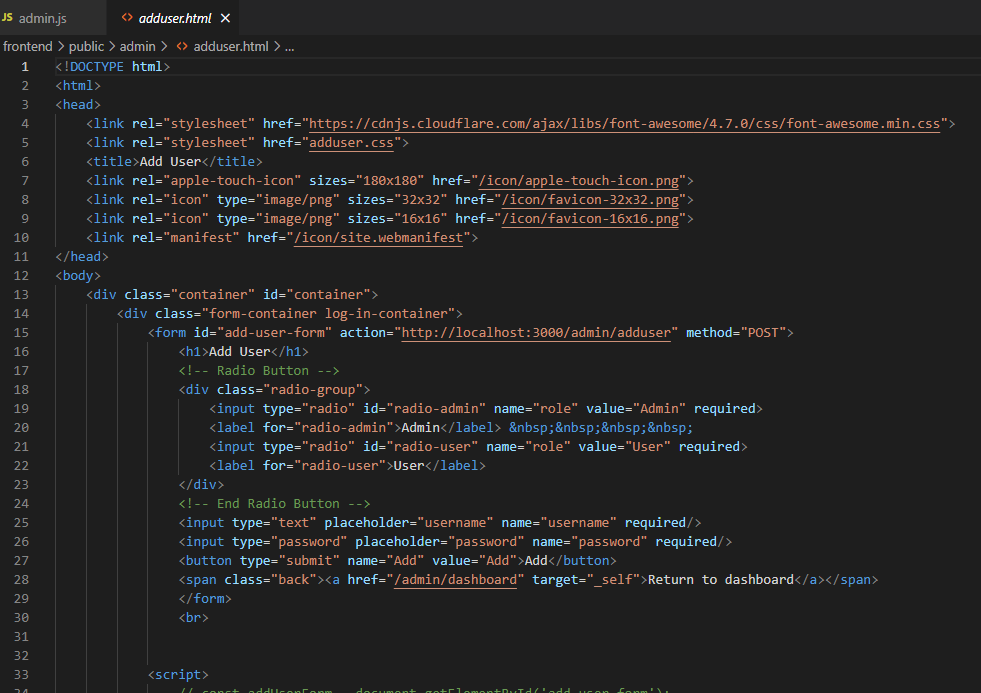
**Product/Service management Page for administrators**

****

**Add user from Admin dashboard**

****

**User Account Management” page for administrators**

****

**Adduser.html**

**4.Details of our web services and code**

BACKEND:

/store/browse/:id

routes.get('/browse/:id', getGame); //Return 1 specific game

FRONTEND:

index.js

// Handles GET requests to the '/store/browse/:id' endpoint

app.get('/store/browse/:id', async (req,res)=>{

// Send a GET request to the 'http://localhost:80/store/browse/:id' endpoint

// (where :id is a parameter in the URL path)

const response = await fetch(`http://localhost:80/store/browse/${req.params.id}`,{

headers:{

'Content-Type': `application/json`

}

});

// Extract the JSON data from the response

const data = await response.json();

// Render the 'product' view, passing in the 'data' as the 'product' variable

res.render('product', {product: data})

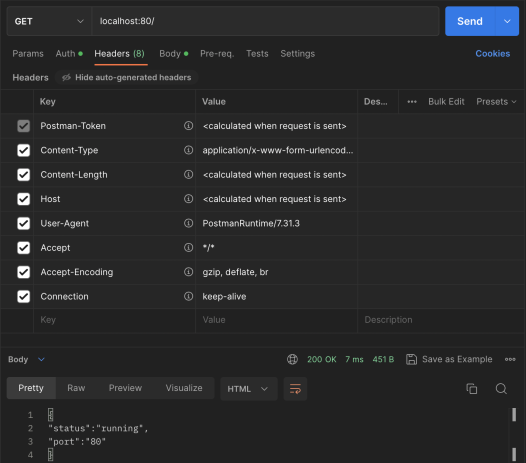
})

**5.The testing result of web services using Postman**

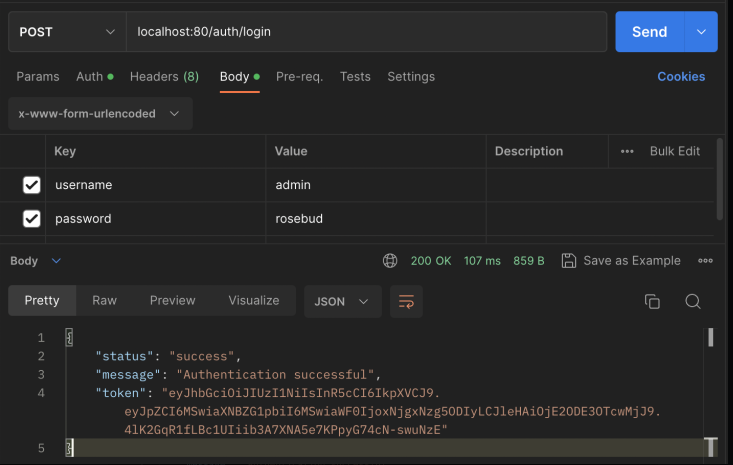
Note:

1.To know product's ID, it can be seen in the URL of a product page (/store/browse/:id)

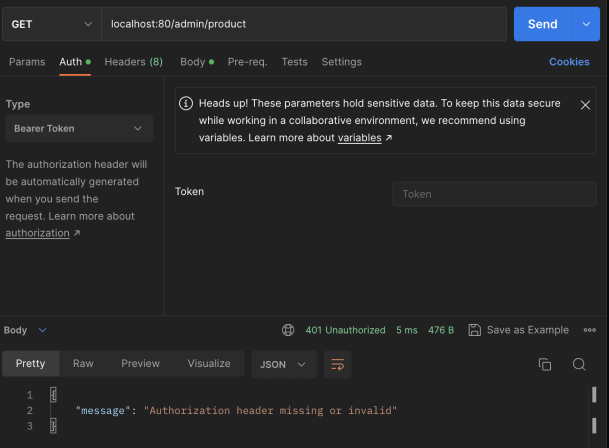
2.To know User's ID, it can be found by opening the editing page in the admin's dashboard (/admin/dashboard/edit/:id)



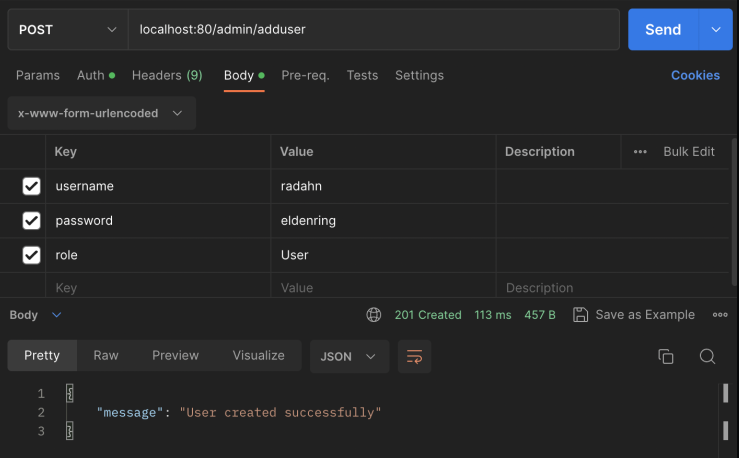
GET /



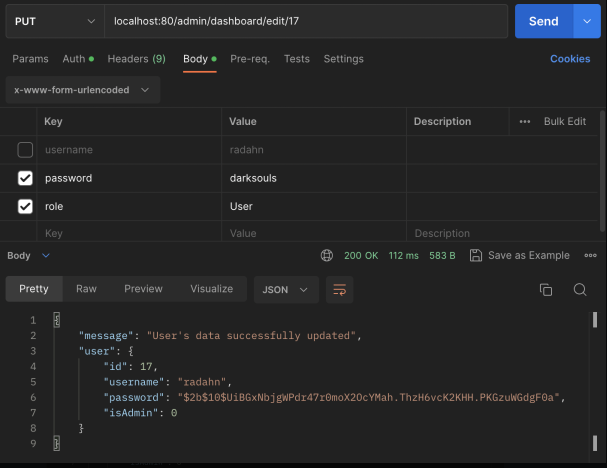
POST /login



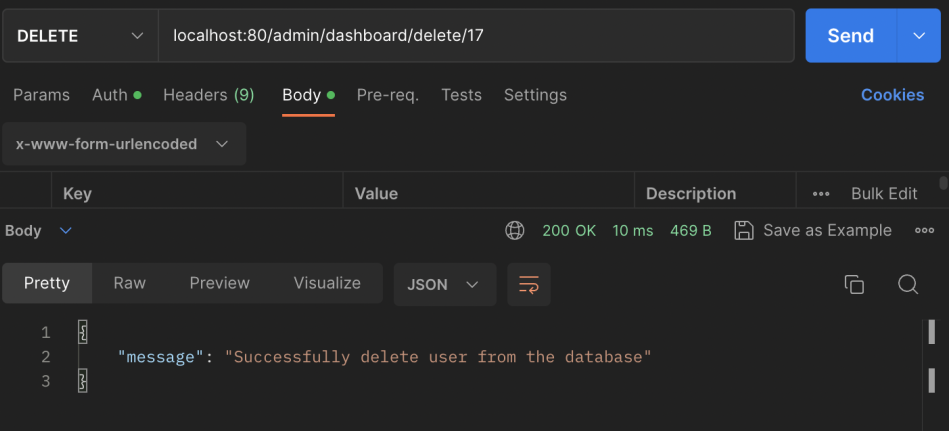
GET /product



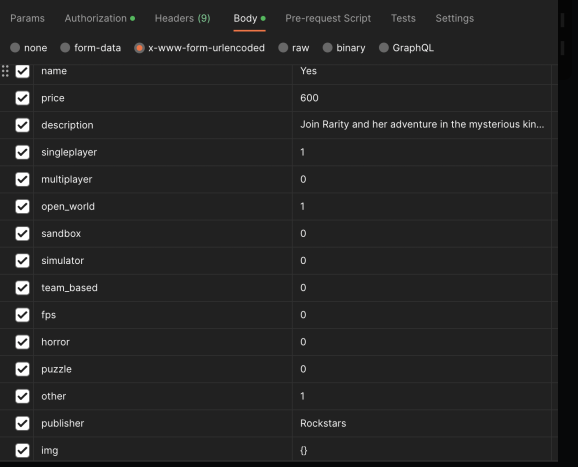
POST /adduser



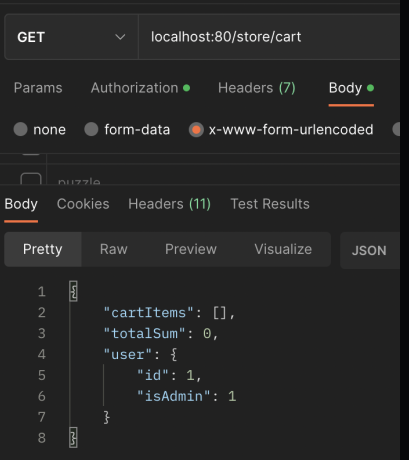
PUT /admin/dashboard/edit/:id



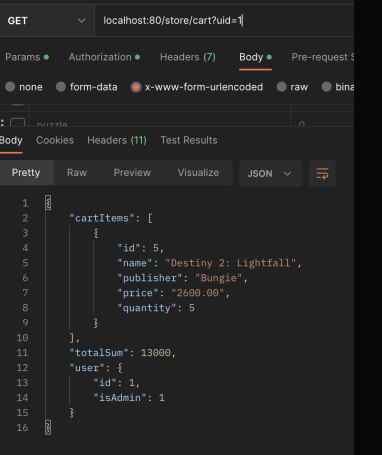
DELETE



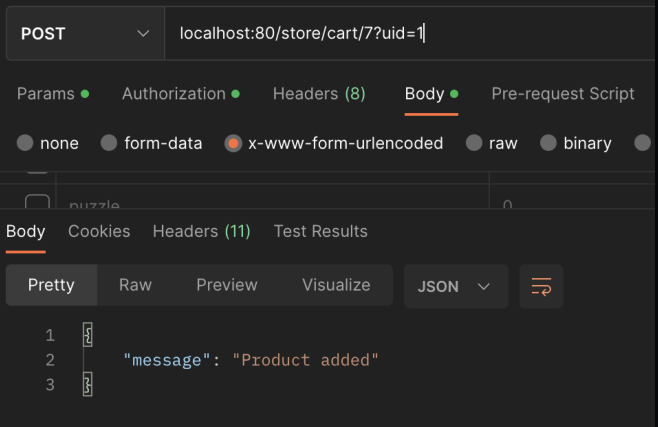
Body for game



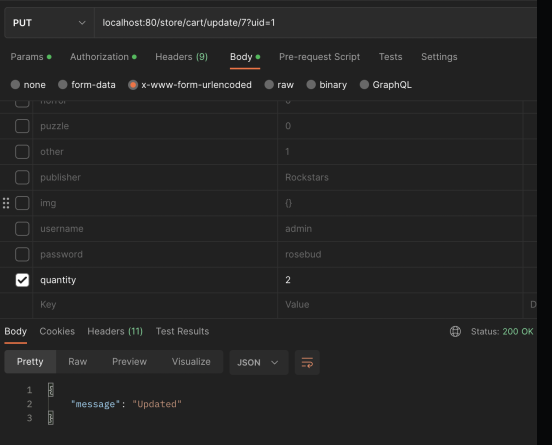
GET /cart



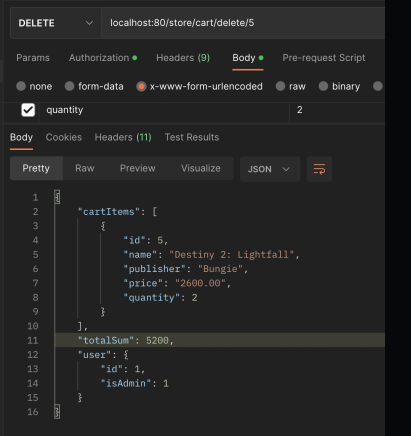
GET /cart?uid=:id



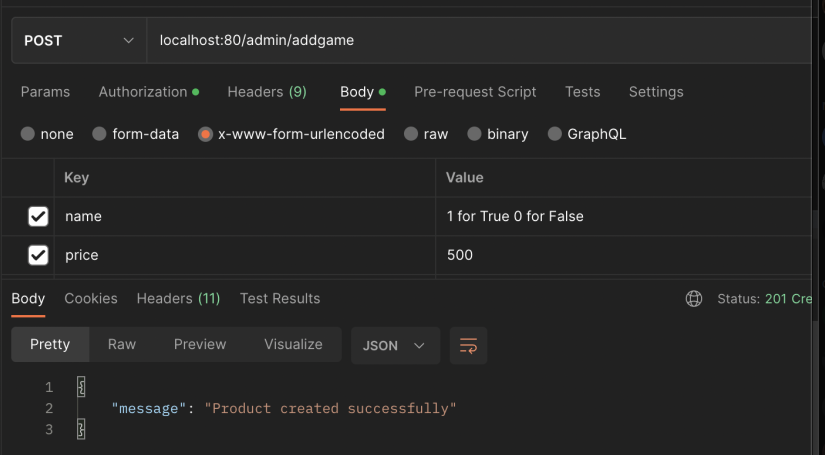
POST /cart



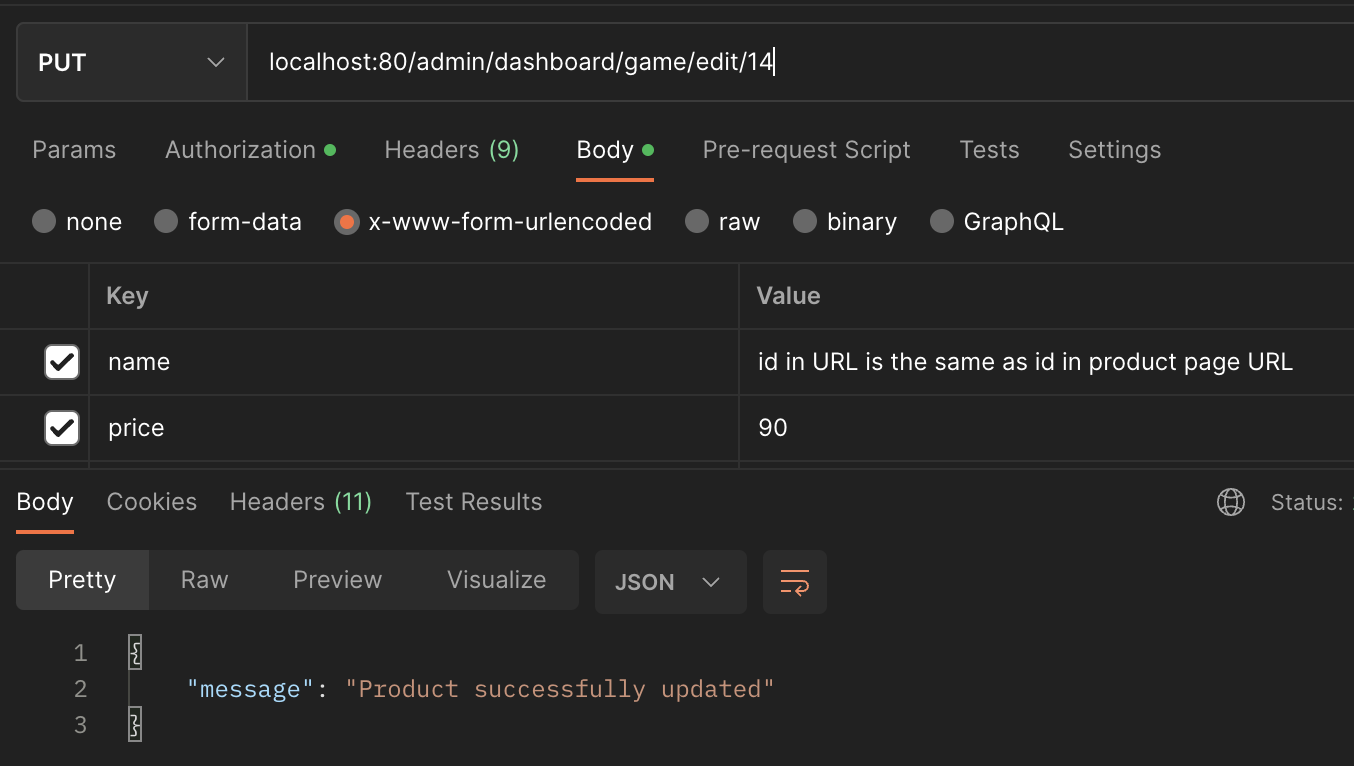
PUT /cart



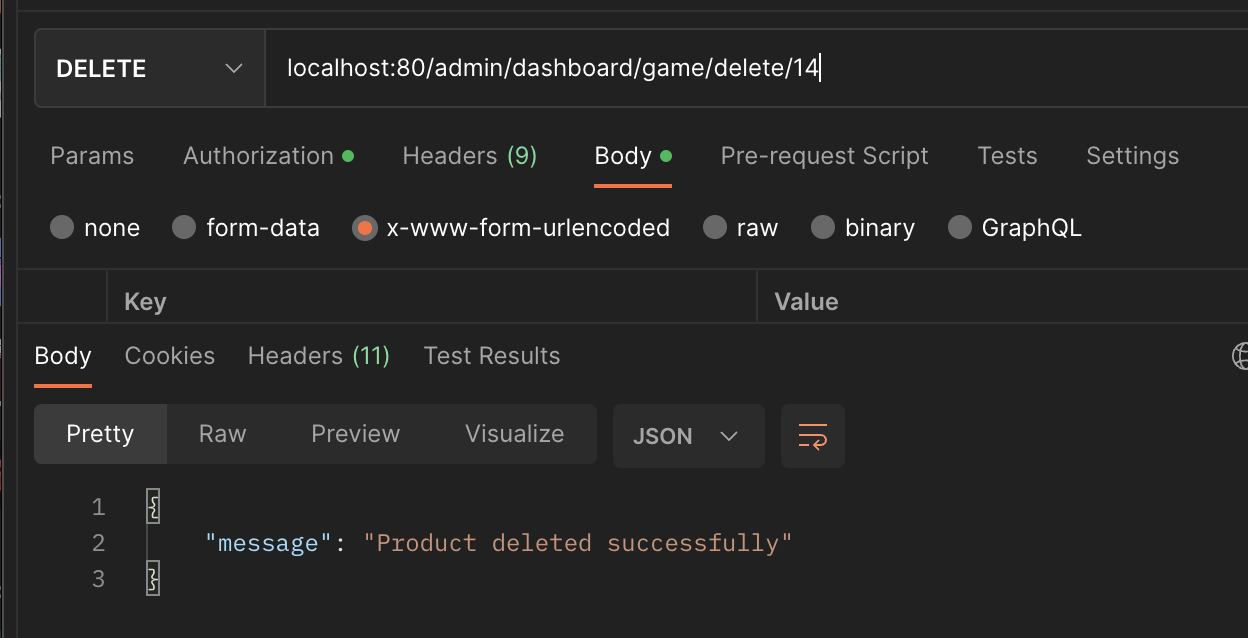
DELETE /cart



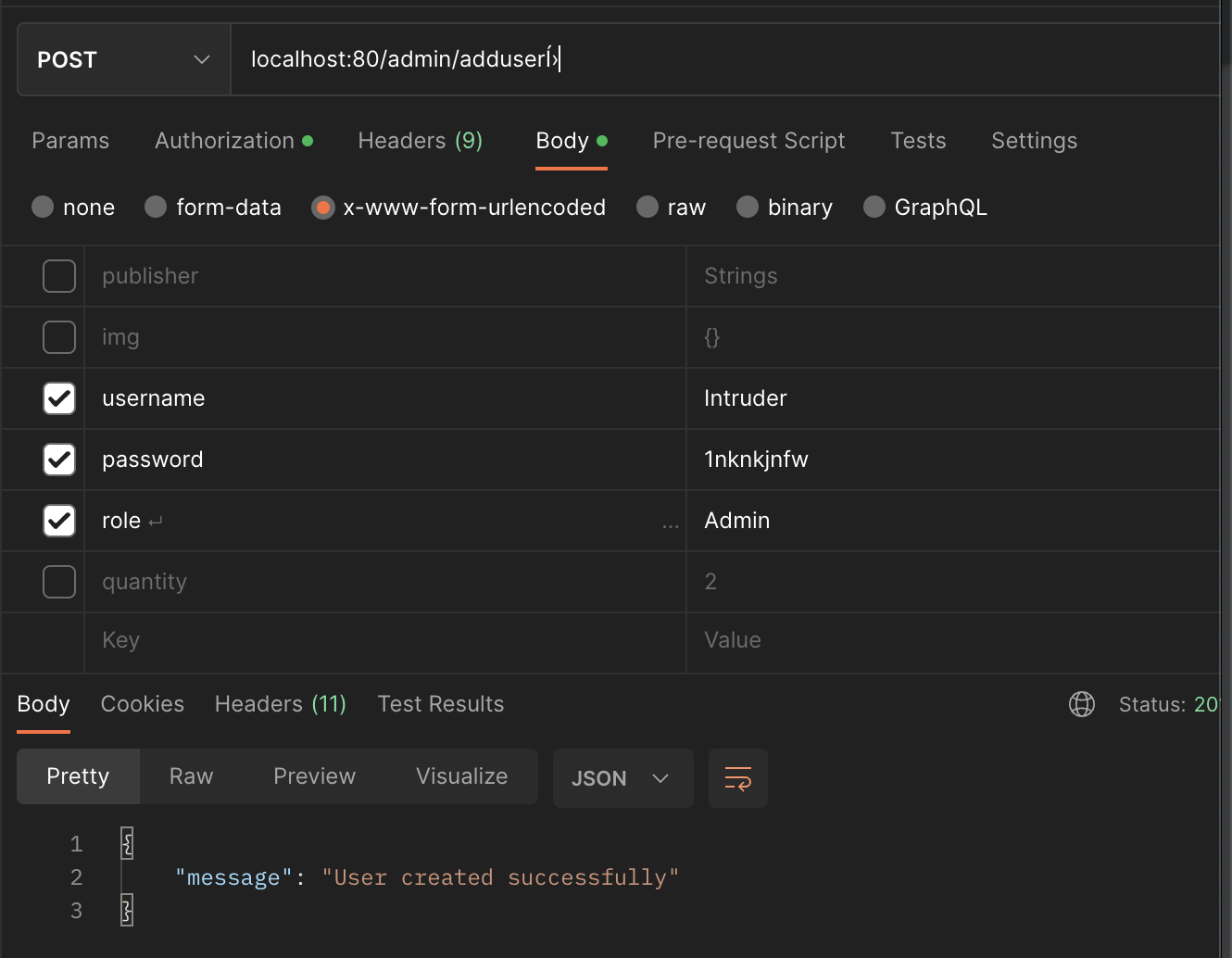
POST /addgame



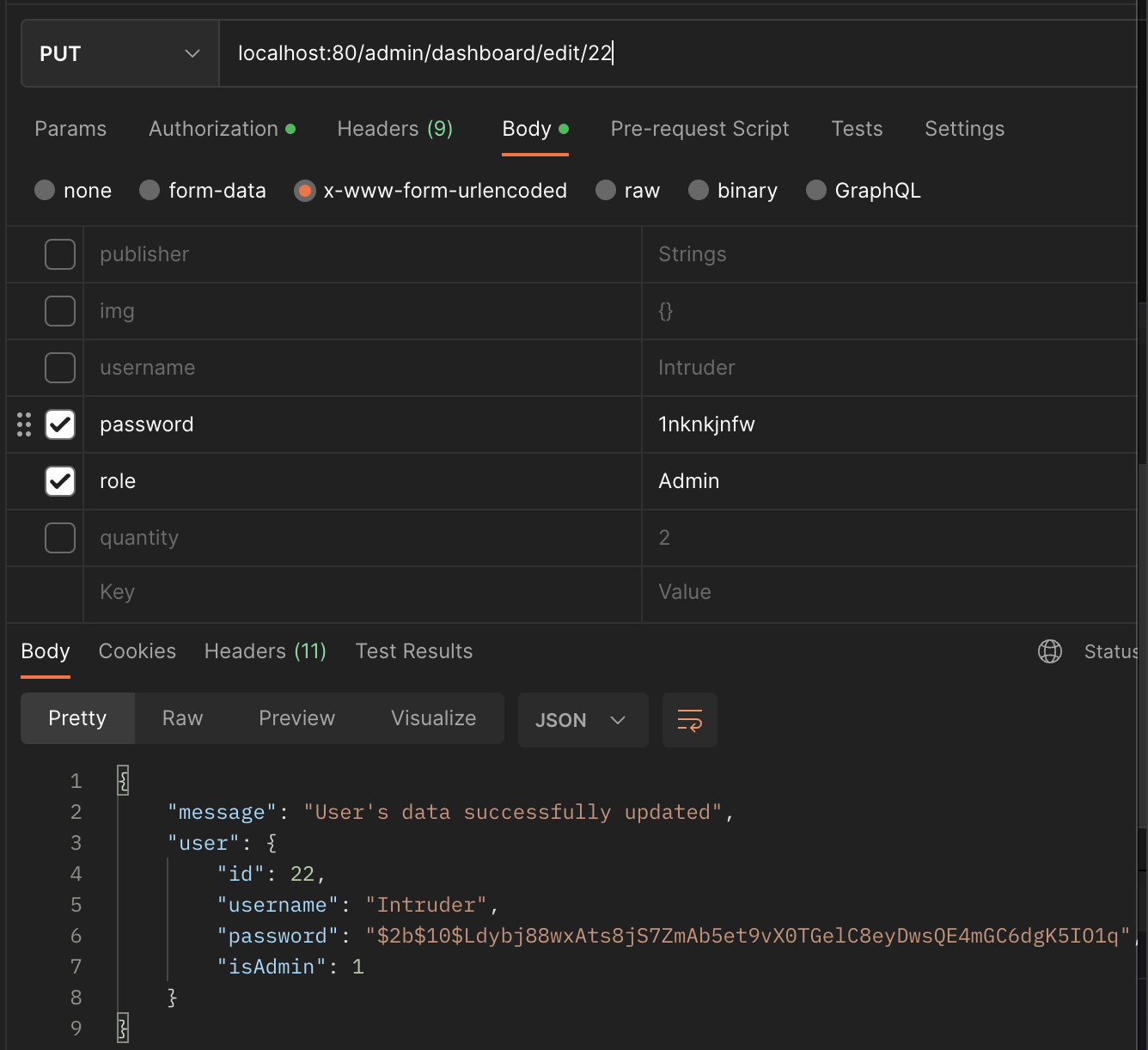
PUT /admin/dashboard/game/edit/:id



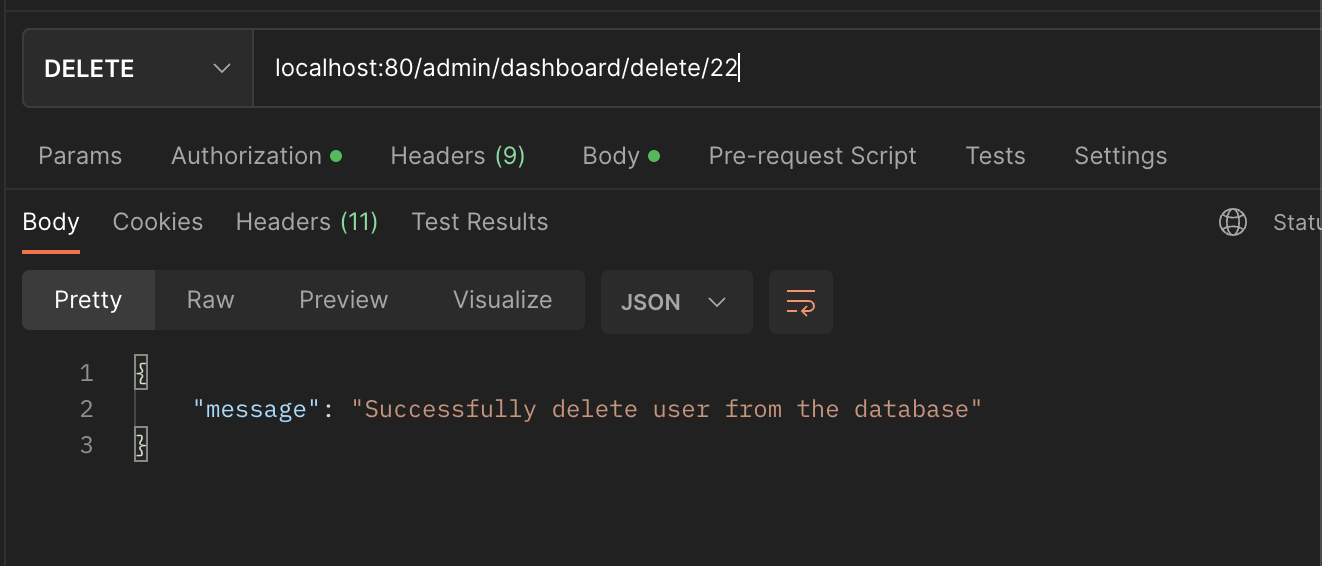
DELETE /admin/dashboard/game/delete/:id



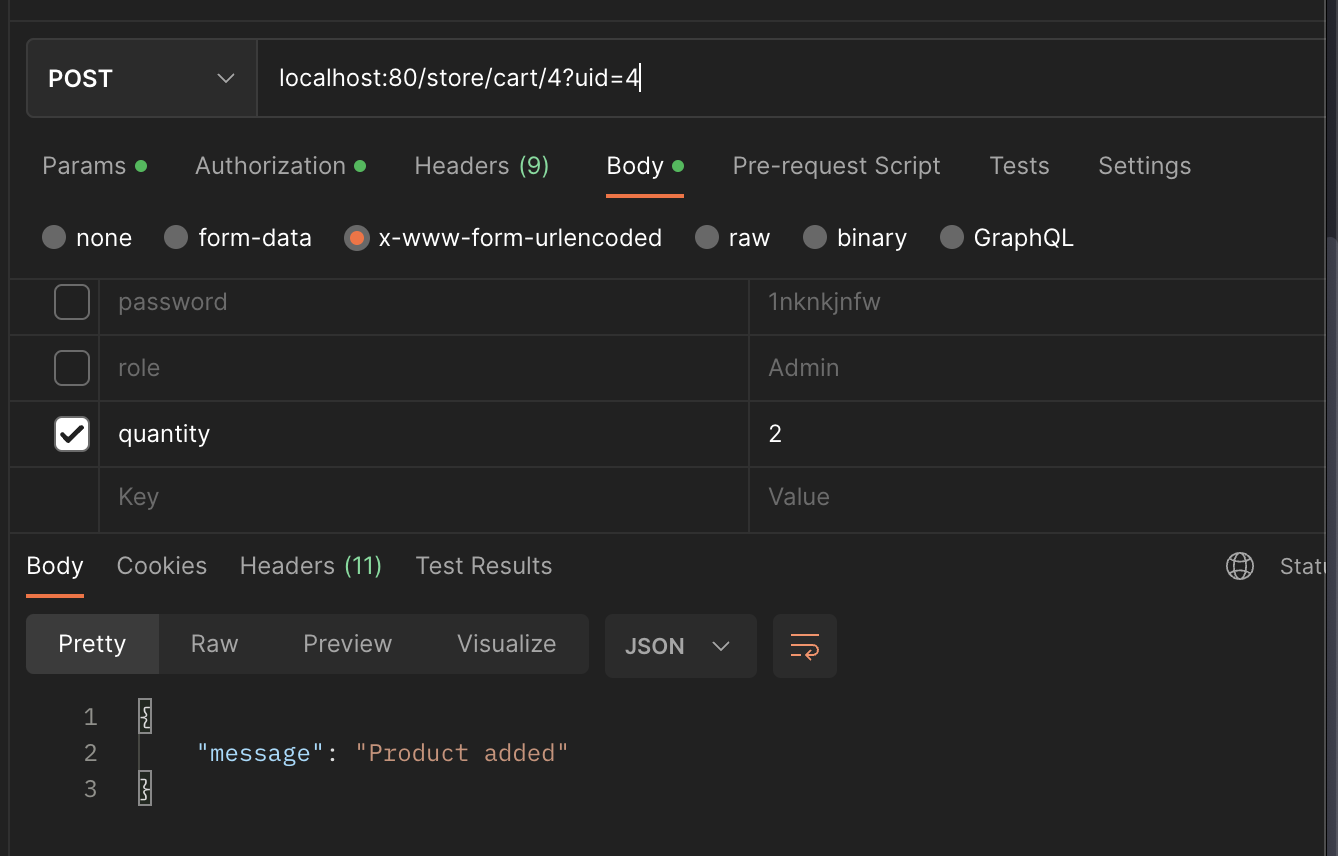
POST /admin/dashboard/adduser



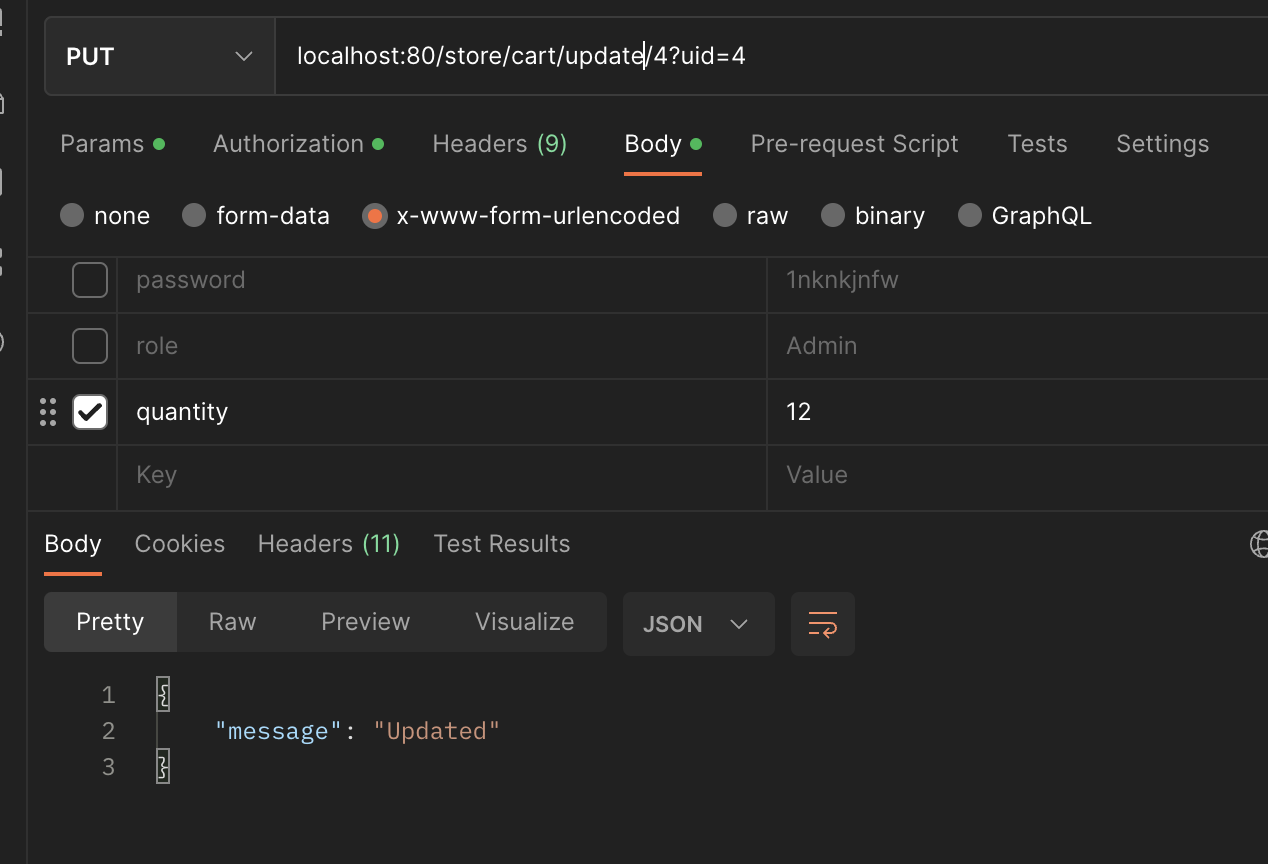
PUT /admin/dashboard/edit/:id



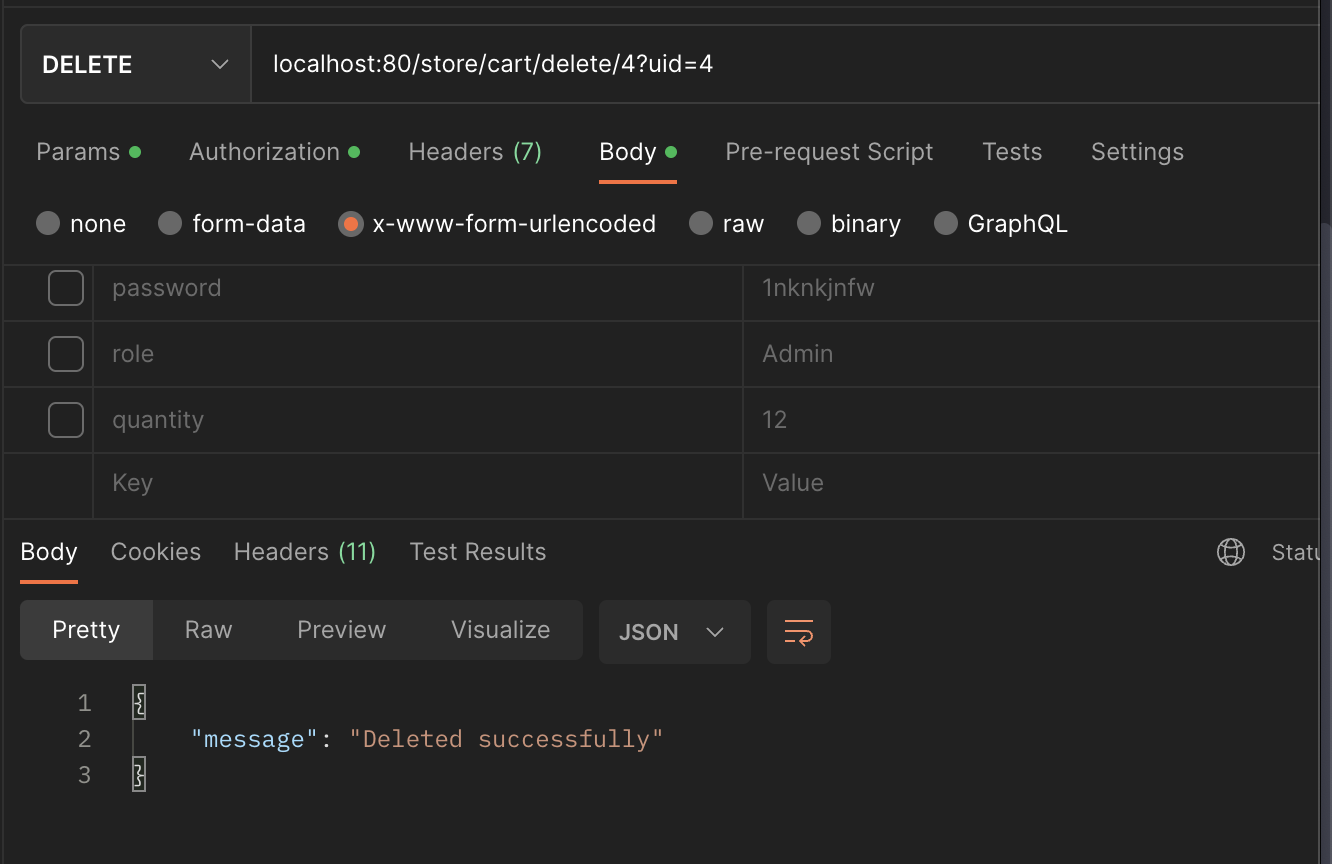
DELETE /admin/dashboard/delete/:id



POST /store/cart/:id



PUT /store/cart/update/:id



DELETE /store/cart/delete/:id

**Reference**:

1. Github Documentation: <https://github.com/Purinat33/itcs212_web-server/blob/main/Documentation.md>
2. Stripe:

<https://stripe.com/en-th>