**Group-Project-Final-Storefront**

**Description**:

For this assignment, we used PostgreSQL for our database. We created three tables. The first table is the products table which stores all our product information. The second table is the orders table which uses the products table to dynamically create the carts page. The third table is the users table which stores the client’s username and encrypted password to our database. When the client registers, their username and encrypted password get stored. There’s an error handling function that checks if the unique username is already in the database or not. Once the client has their login credentials, they can sign in. The sign-in page checks for the correct username and password stored in the database. Once a client is logged in, they can access the cart and payment page which is restricted to only authenticated clients. Everything listed here works. To get everything started, you need to install the PostgreSQL database, install all the dependency packages with npm install, change the .env information for the database that’s on your machine, create the 3 tables (users, orders, and products) which is shown in the screenshots, and node app.js to start the application which defaults to port 3333.

**Approach / What I Did**:

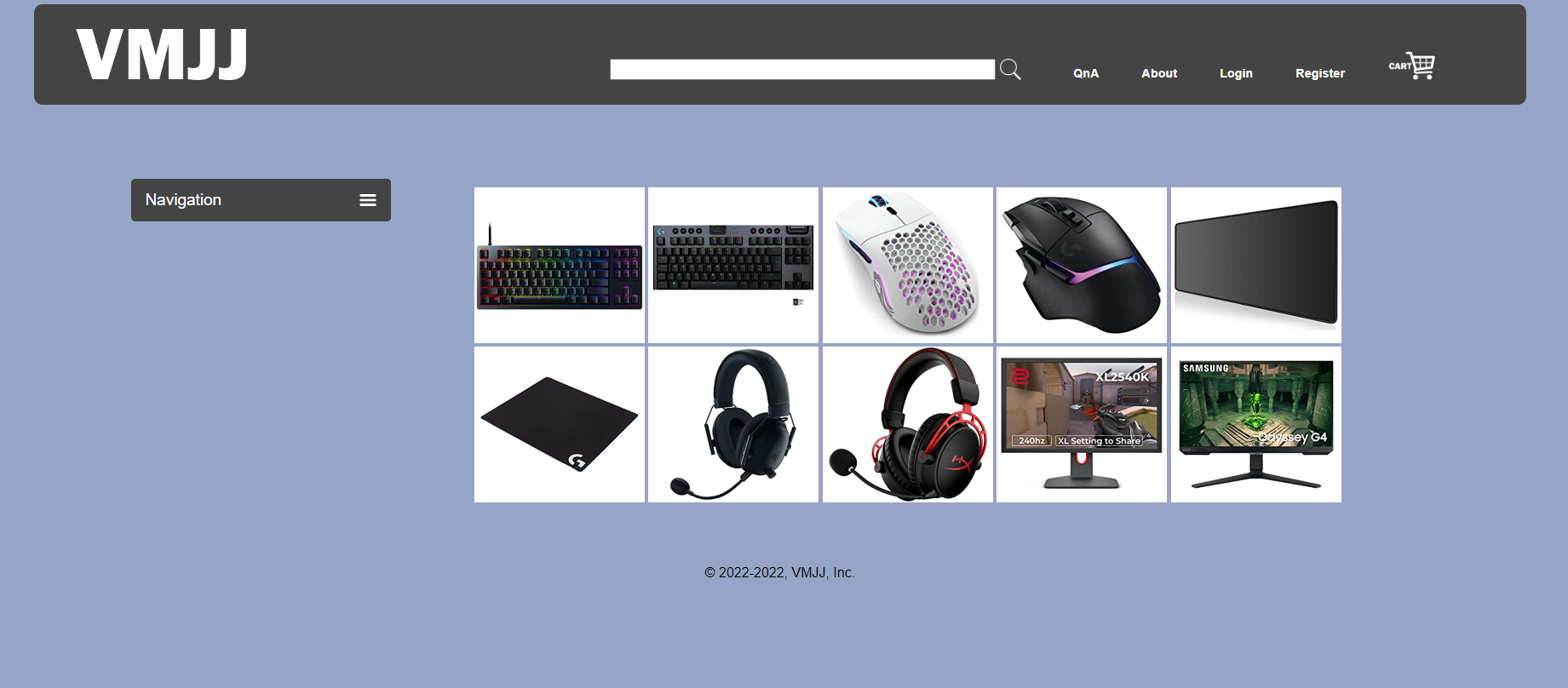
We separated each page so that only certain pages can be accessed without client authentication and certain pages can be accessed only with client authentication. We used the npm package passport for authentication. For storing client information, we used bcrypt to hash the client’s password before storing it in the database. Also, we used npm package handlebars to retrieve server-side data for our HTML web pages.

**Issues and Resolutions:**

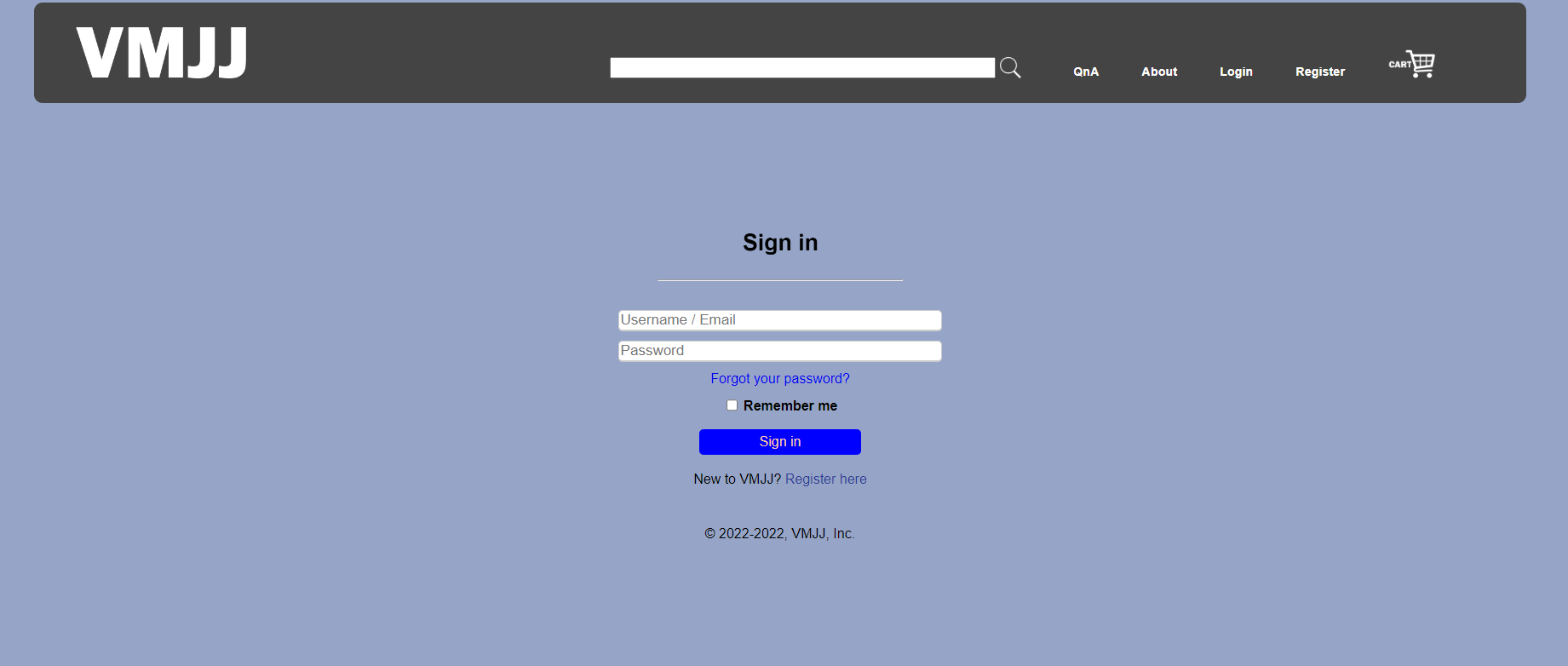
* The first issue was trying to connect to the database, it was resolved by trial and error.
* The second issue was figuring out how to retrieve information from the database, we found handlebars to be helpful.
* The third issue was figuring out how to create the cart page. We used database queries to copy products from the products table to the orders table.
* The fourth issue was fixing all the bugs on the website. That took trial and error.
* The fifth issue was trying to figure out how to make things work the way we wanted them to. That took some research and finding the correct syntax to use.

**Screenshots:**

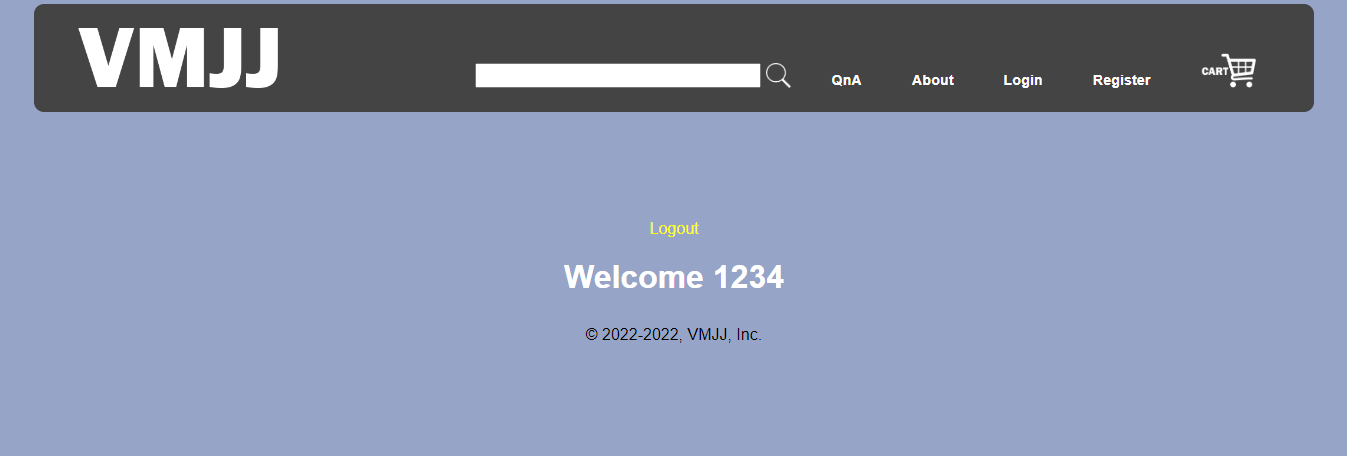
1. Home page



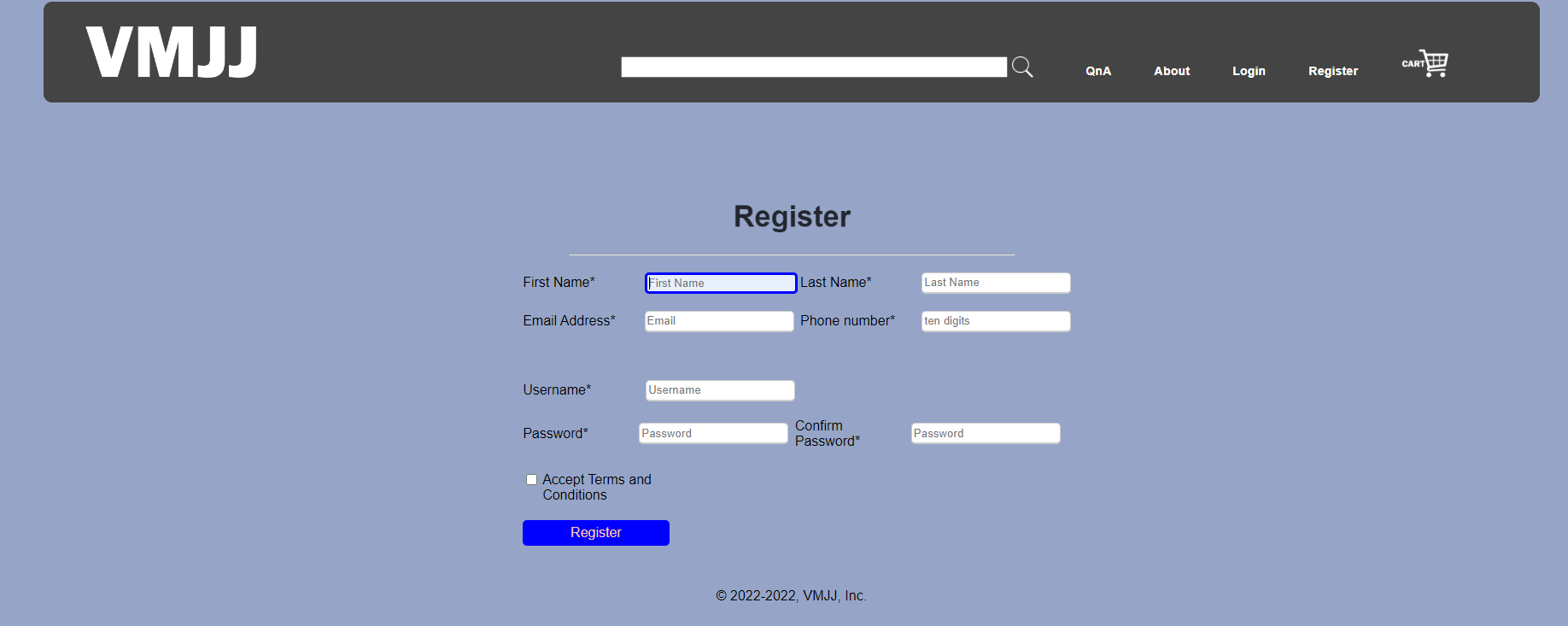
1. Sign in page



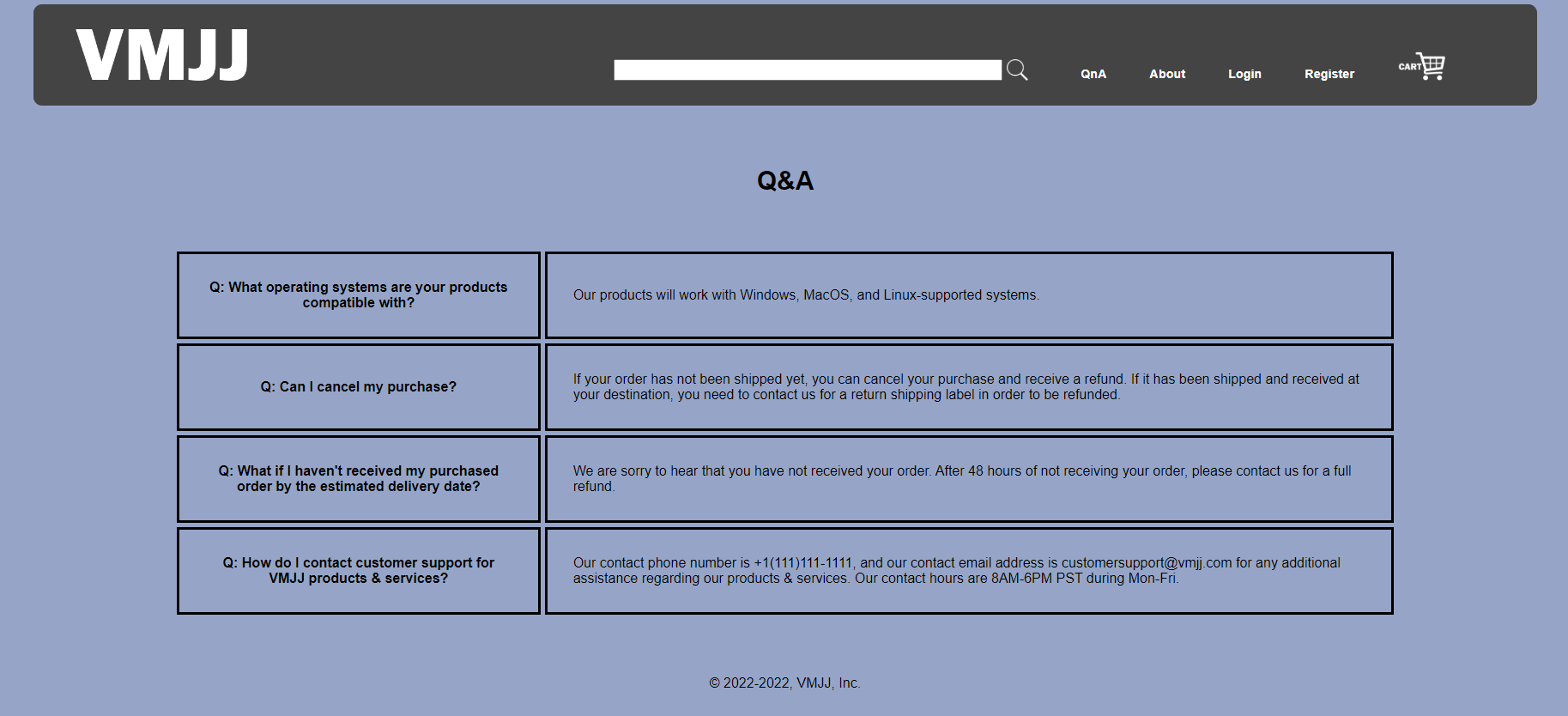
1. Successful sign in



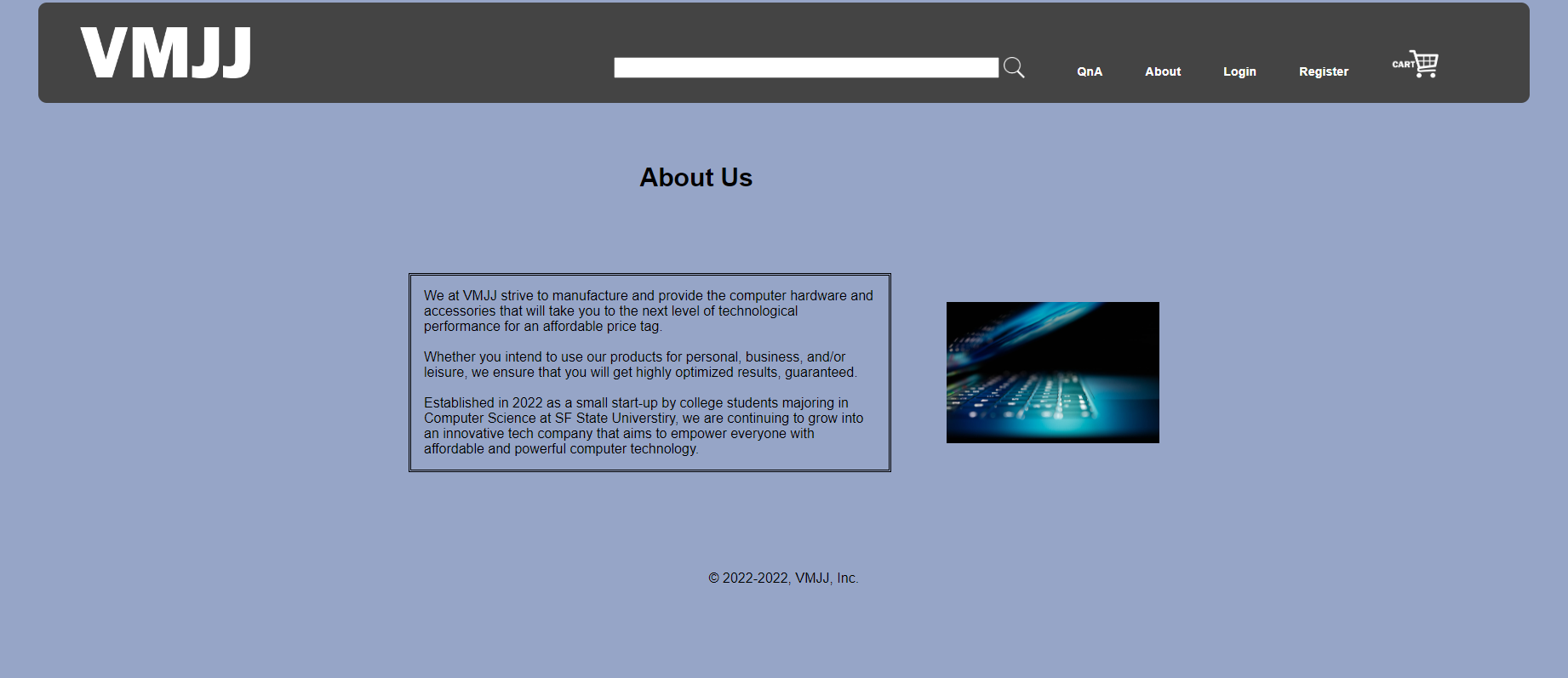
1. Registration page



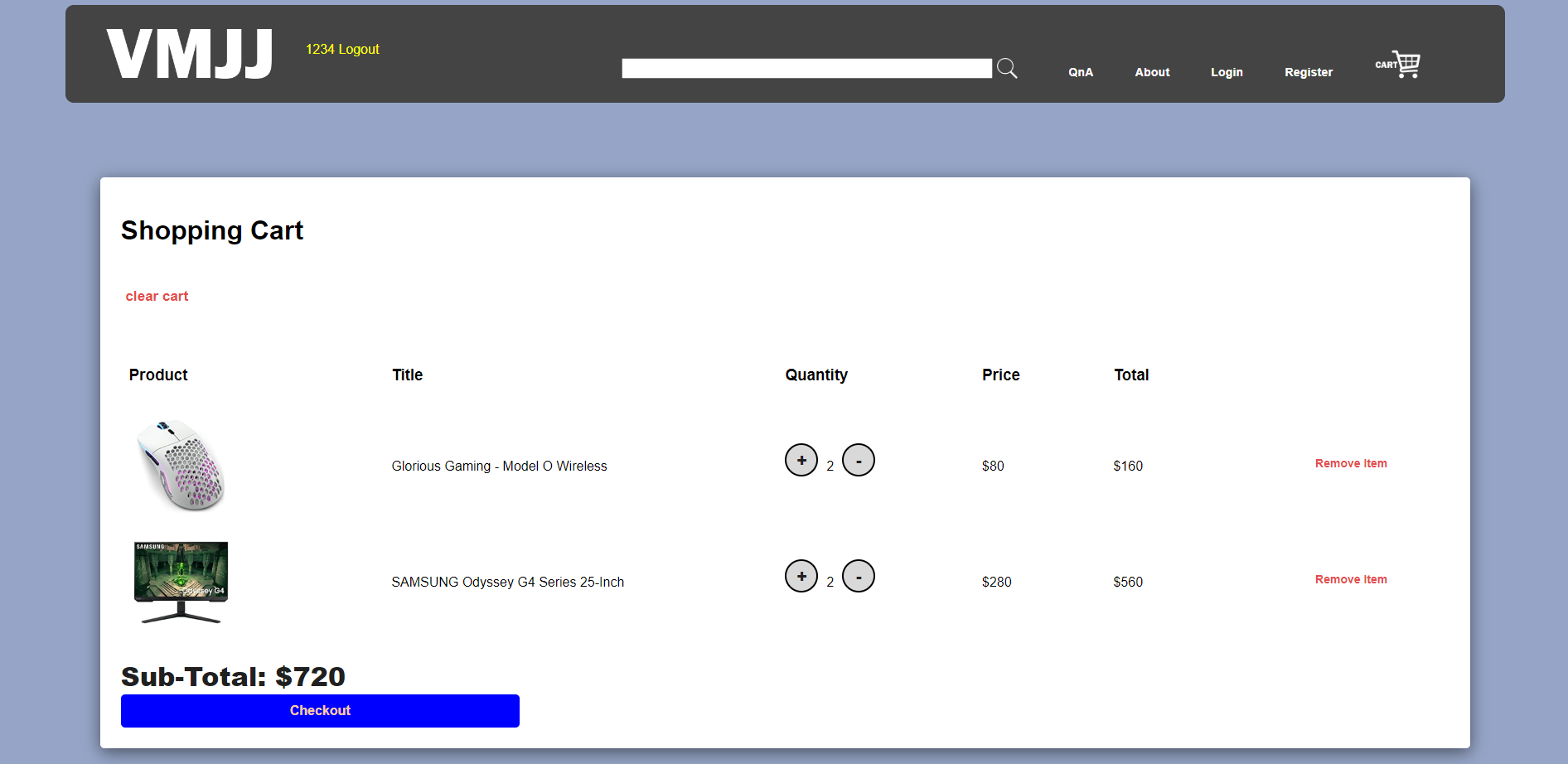
1. QnA page



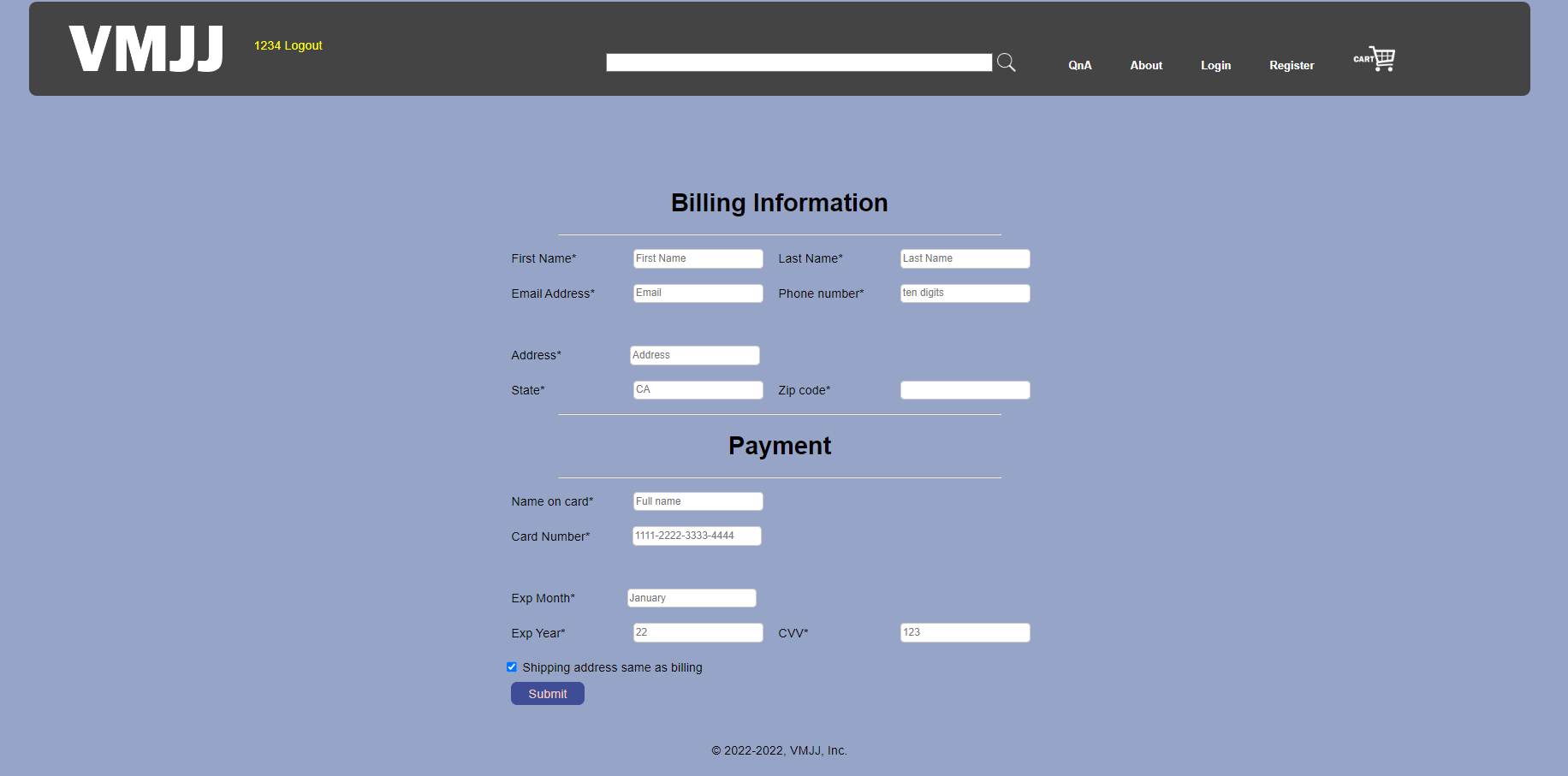
1. About page



1. Cart Page

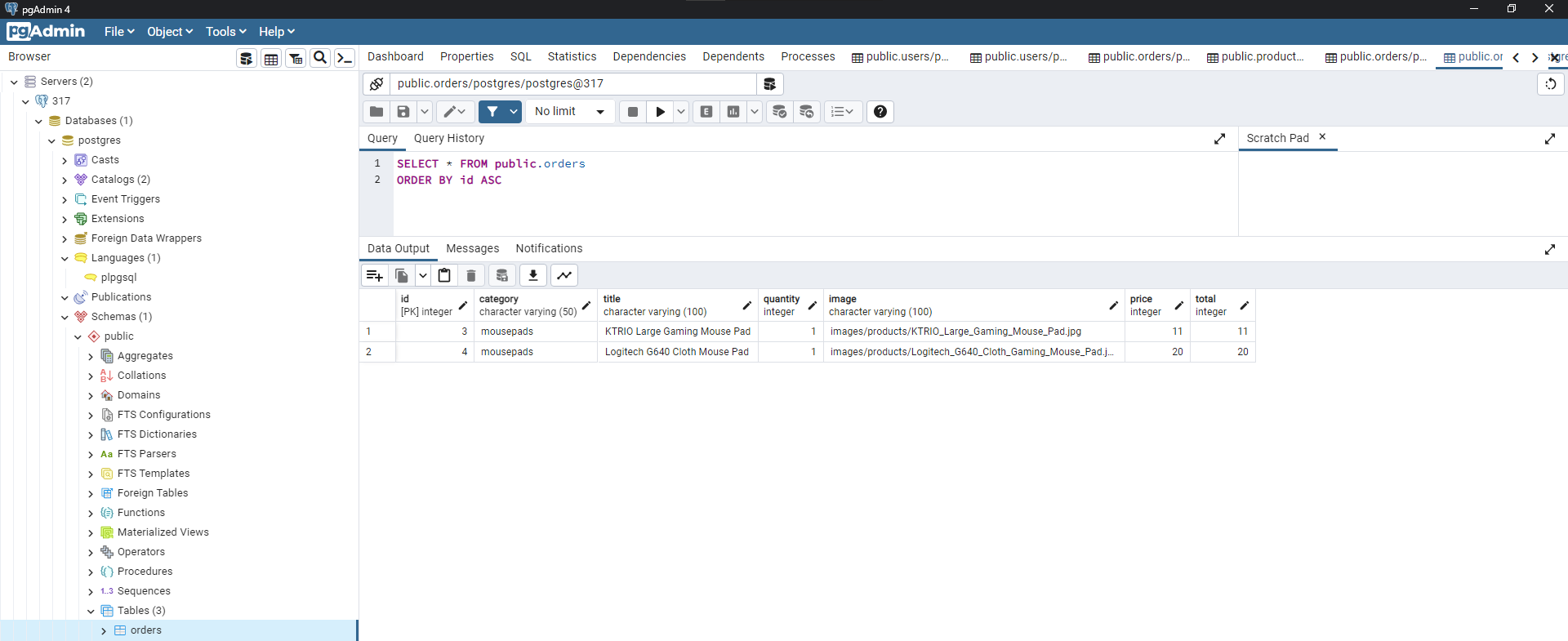


1. Payment Page

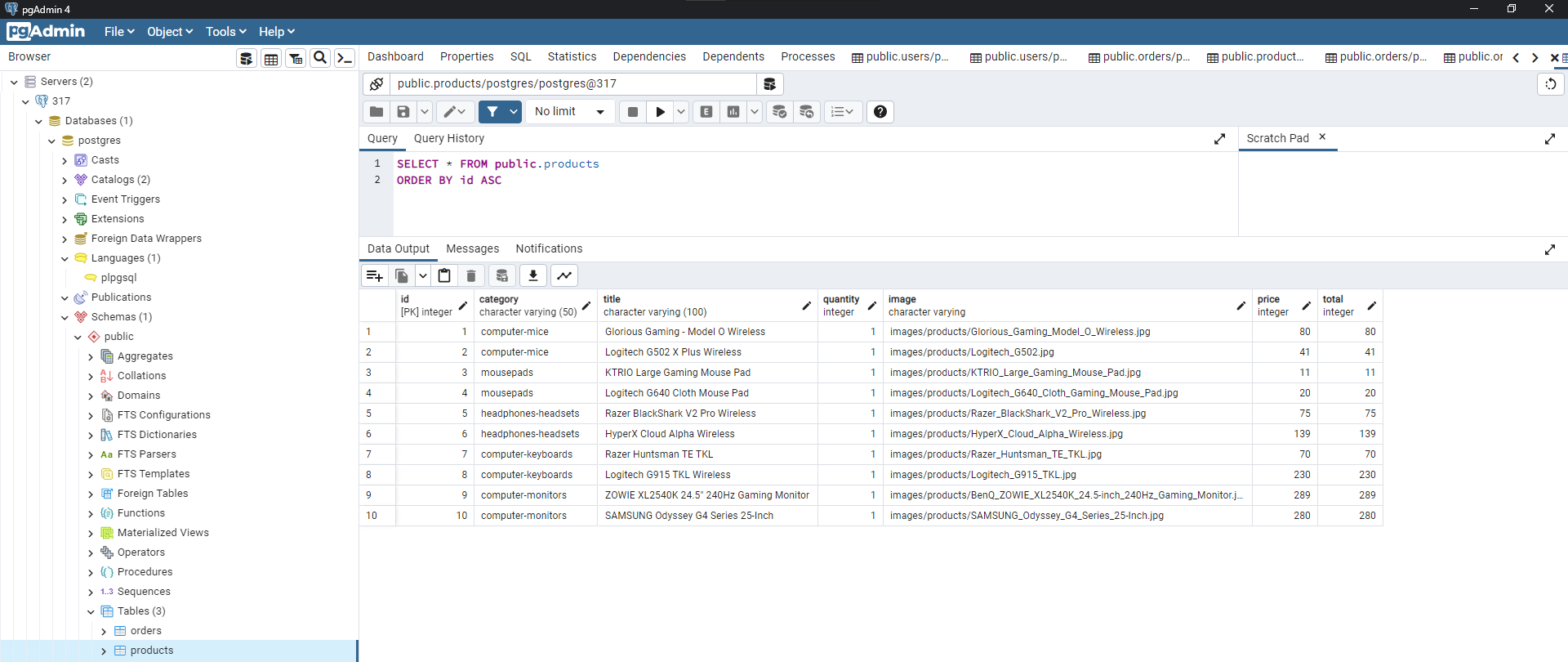


1. Database tables

* Orders:



* Products:



* Users:

