PostgreSQL Database Integration with a Windows Forms Application

Jonathan Glasgow

Project Description

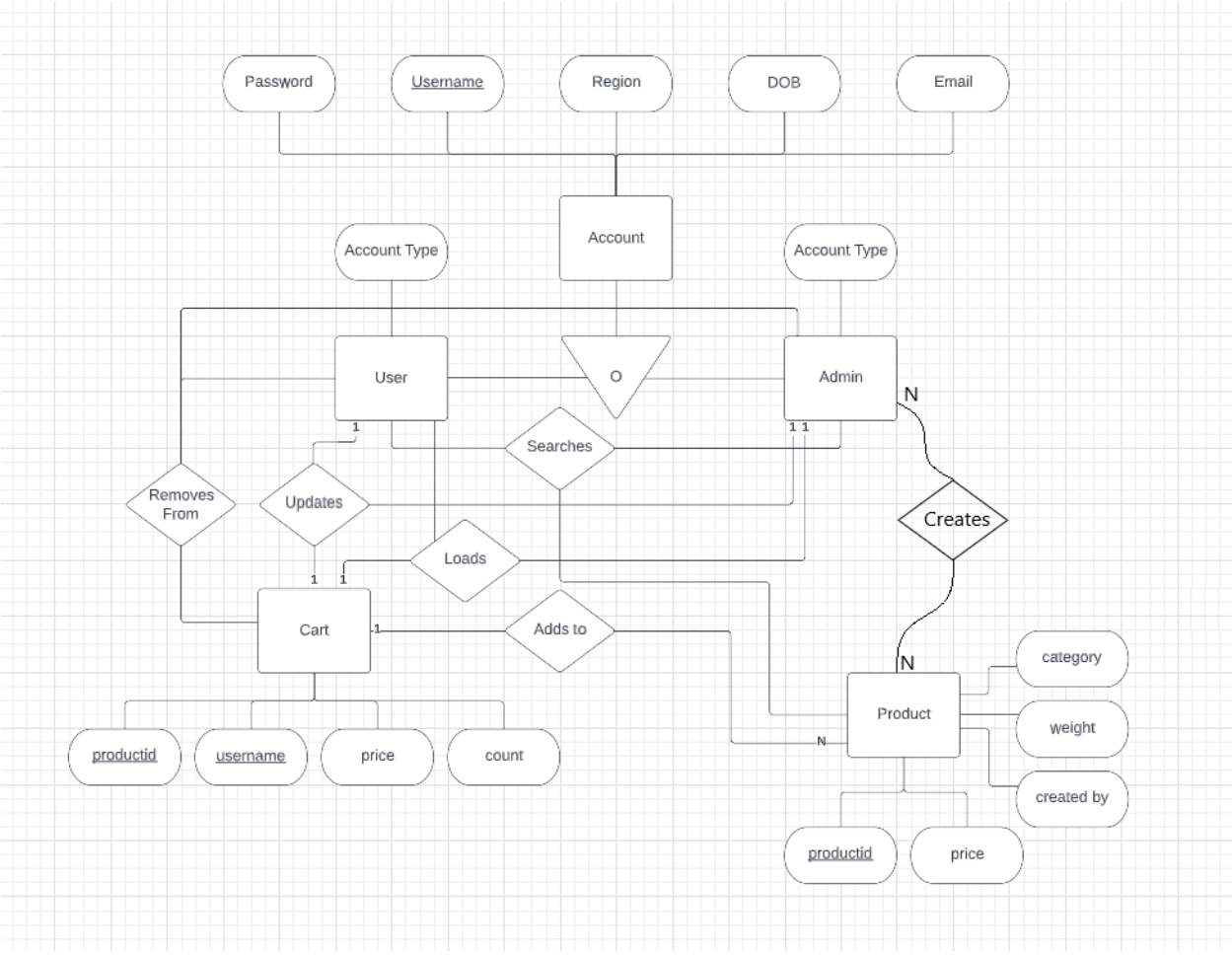
This project was conducted to gain experience designing a database from scratch and accessing it through software. I decided to create a Windows Form Application to simulate a checkout application. The database was created using PostgreSQL. The application was made in Microsoft Visual Studio. There were five different pages implemented in the program including the registration page, login page, home page, profile page, and profile edit page. These pages allow the user to register, login, view their profile, make changes to some of their profile information and search and view the home page. Each designer page had its own individual code that referenced the databases held in pgAdmin 4. Each page references the database in some way or another, either storing information that was inputted, or retrieving information that is already in the database. Each page also contains an exit button to leave the program at any time to avoid getting stuck and to allow for a “reload” in case there is some bug that is encountered during use of the program.

It is to be noted that **this project is meant to be a demo and may include bugs.** The connection string was created to be used on a local PostgreSQL database. If I had a need to properly publish this or create a product similar for a company, I would be sure to connect through more proper means to a database hosted through a cloud service. It is also important to note that password authentication and encryption methods were also omitted from this demo, and more proper procedures would be in place if this were to ever be properly fleshed out. Admin account features were also never properly fleshed out. The purpose of this project was to create an environment where I could properly test database interactions with software.

The current version of the program includes the following:

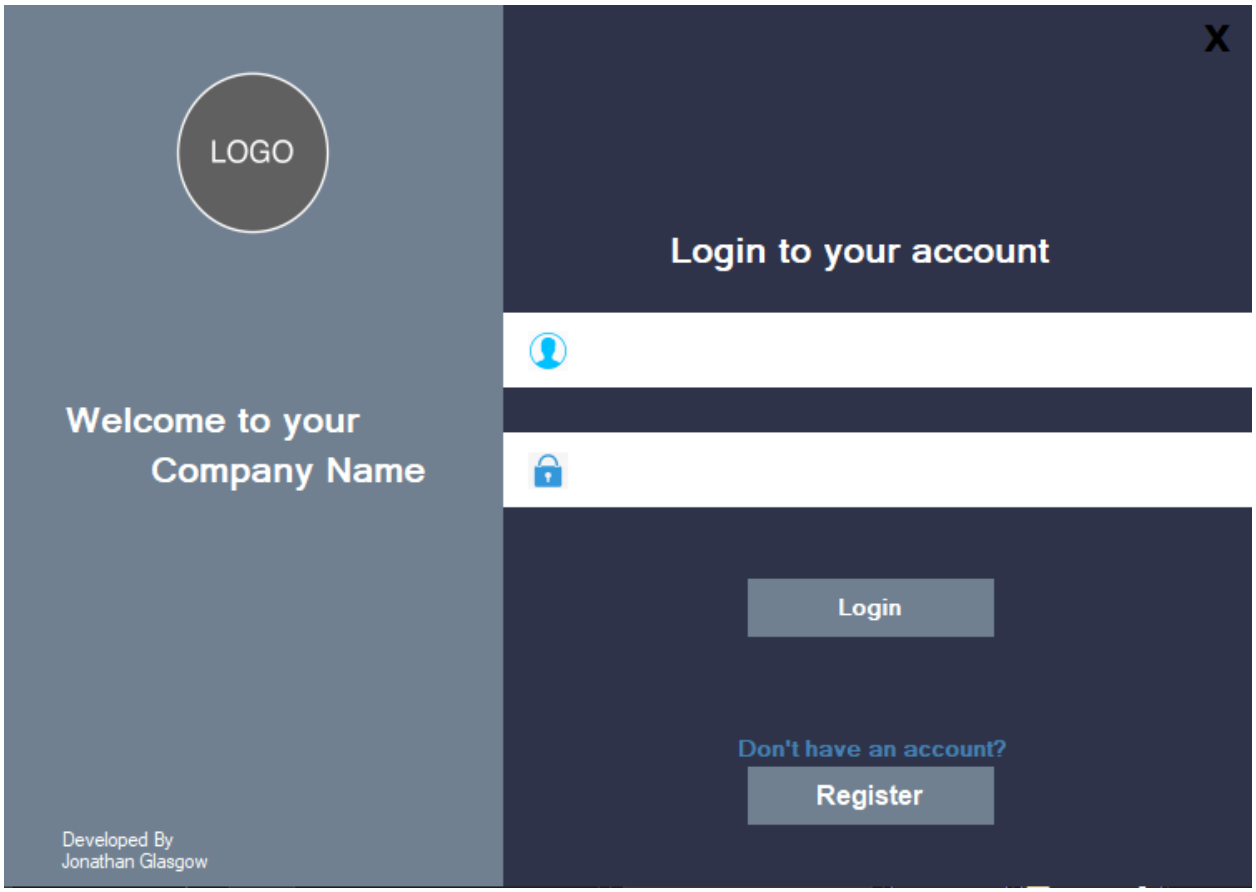
- Login menu
- Registration menu
- Product selection screen (including pre-determined buttons, a search function, add and remove buttons, update cart and load cart button)
- Profile menu
- Profile edit menu

ER Diagram

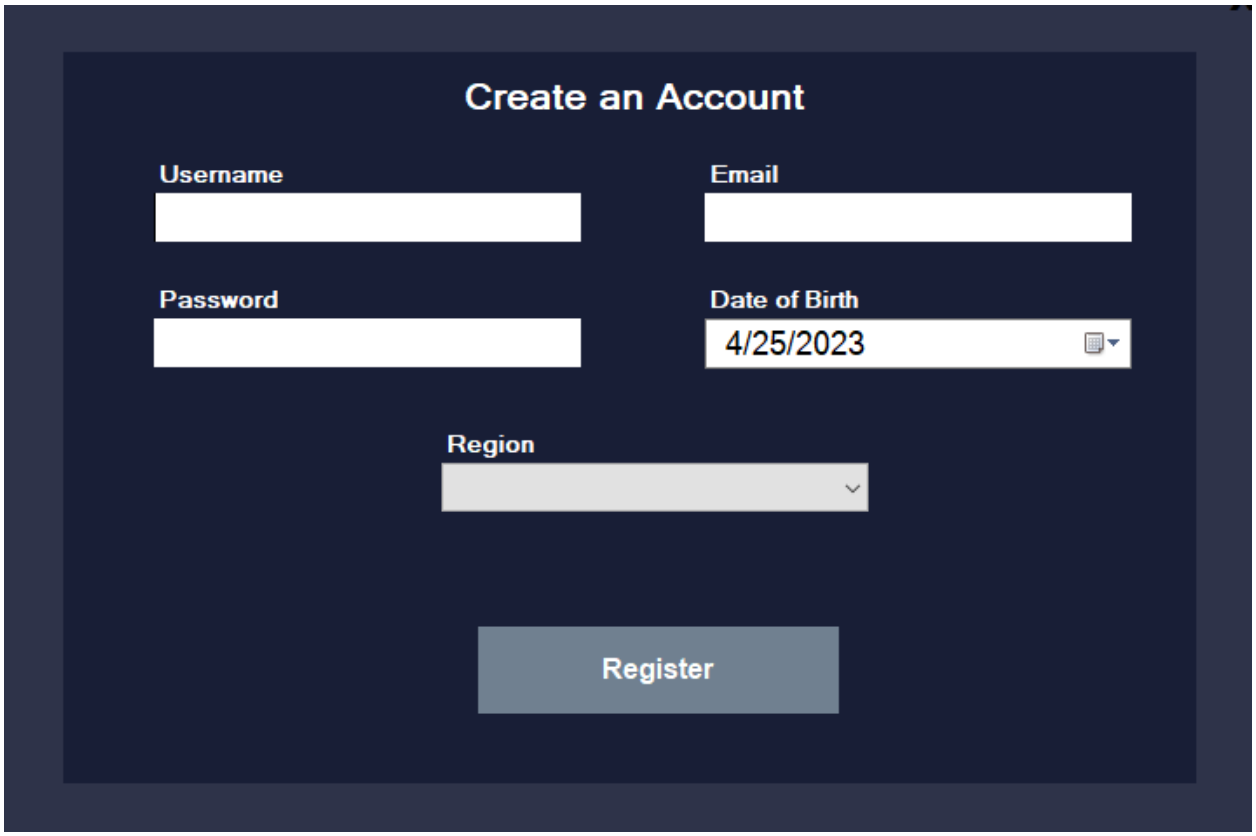


Login Page / Registration Page

This is the first page that is encountered when loading the program. There are 2 different buttons displayed “Login” and “Register”. Every page will have an exit button located in the top right that will allow them to exit the program at any time, it will first pop up with a warning message asking them to confirm. There are then two text boxes, one for the username input and the other for password input. Both are stored as strings in the program. Once a proper username and password have been entered, hitting the “Login” button will then search in the database for a matching username and password, if found, the program will proceed to the home screen with a successful notification.

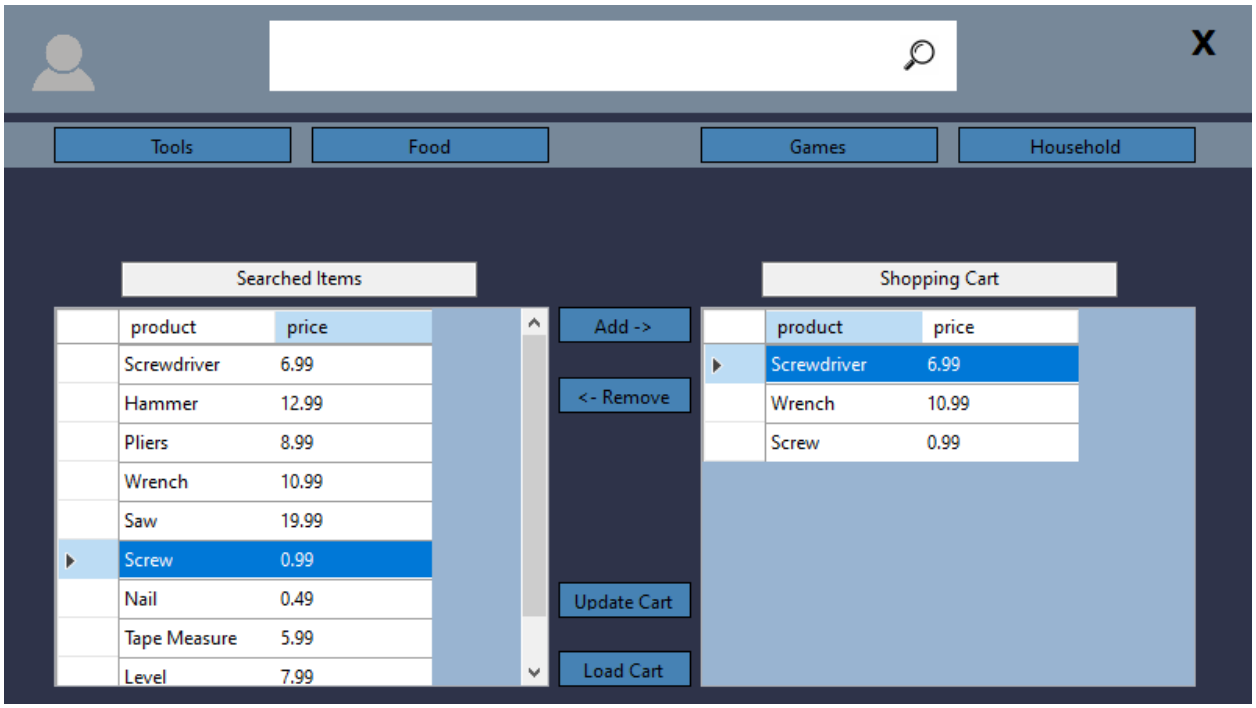


The first button to be used should be the “Register” button. A first-time user will have no login information. Selecting the “Register” button will allow the user to input a username, password, confirm the password, insert an email, region, and date of birth. Once completing the registration information, the user can then use the username and password to log into the program.



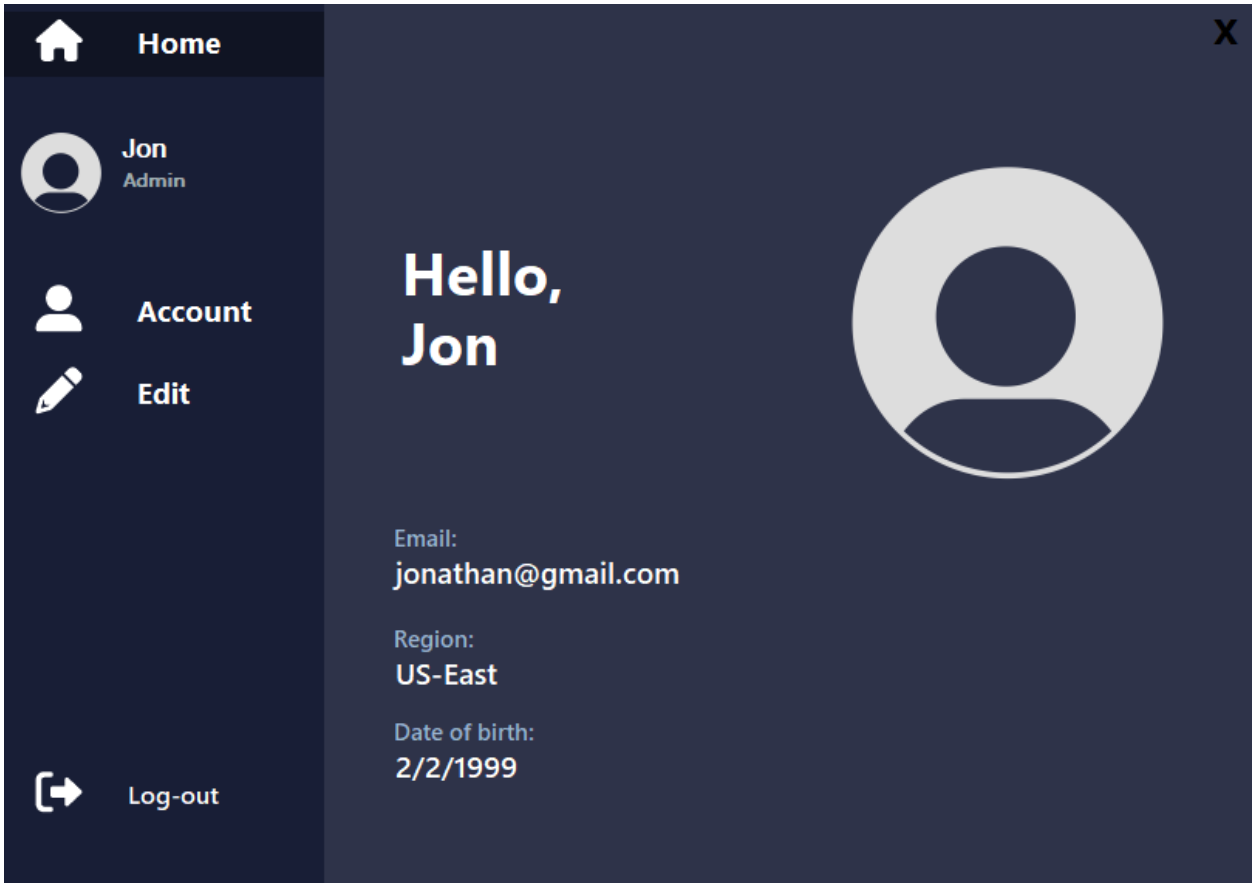
Home Page

This is the next page that will be encountered after the registration and login page. This page displays the main function of the program. The user can select some of the predetermined buttons to display items from that specific category. For example, when selecting the “Tools" button, the program will search in the database for any products with their category label as “Tools”. This is then filled into the table displaying only the product name and price. Other information such as weight, product category, and who created the product is hidden.



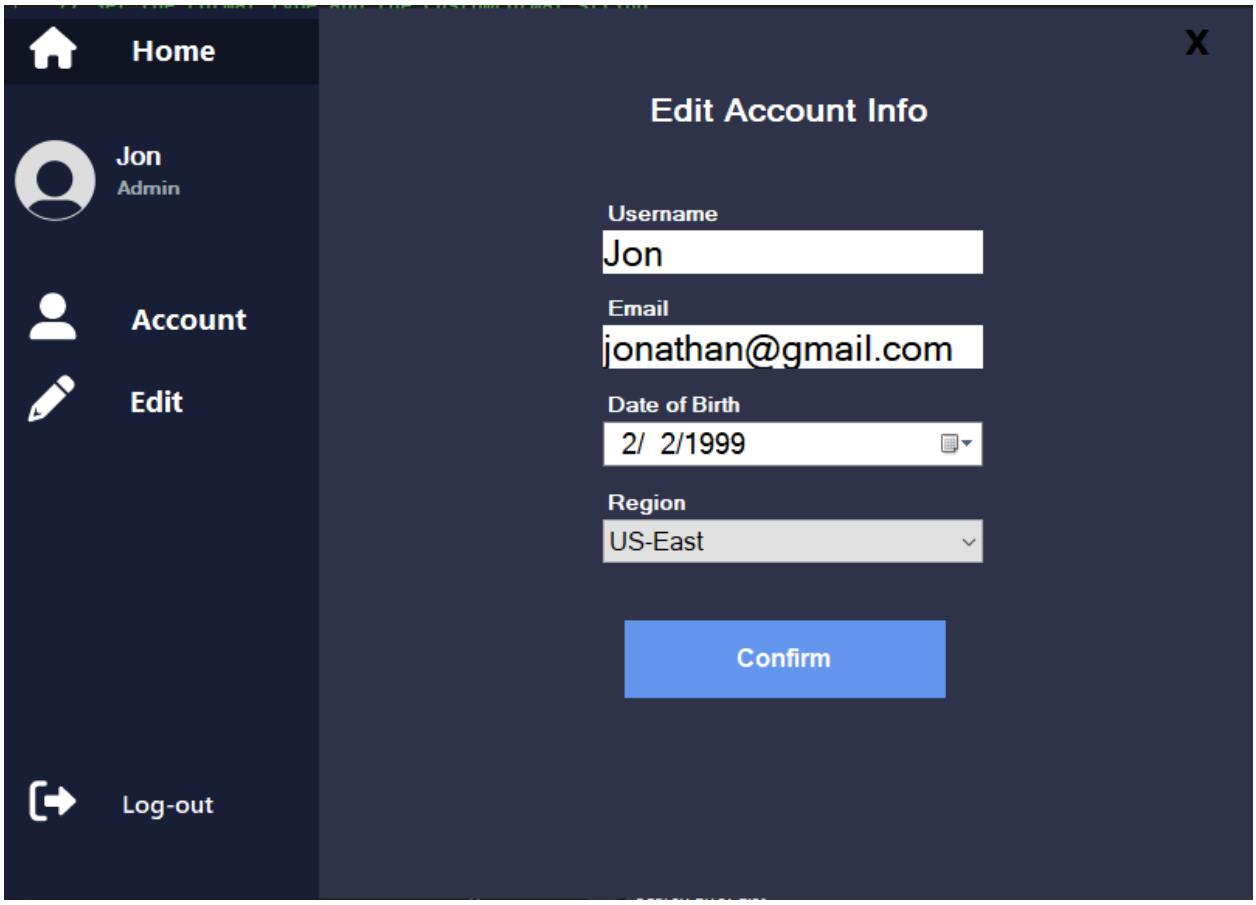
Once items are displayed in the “Searched Items” data grid, the user can then select those items and choose whether they would like to add them to the “Shopping Cart” data grid by selecting the “Add” button. The add button creates a clone of the row selected and duplicates it on the other data grid. The “Remove” button does simply what it states, it just removes the selected row in the shopping cart when pressed. Once the user has the desired items loaded into the “Shopping Cart!” list, he/she may then choose to select the “Update Cart” button. This will upload the current shopping list to the database which can then be loaded later by selecting the “Load Cart” button. The home page also features a search bar to manually search for products in the database, and buttons to update your cart or load your previous cart if you are just logging back in.

Profile Page / Profile Edit Page



The profile page is the next form that would be encountered by the user. It displays simple user information such as email, region, and date of birth. All of which are stored into the database upon registration. There are four different buttons presented on the profile page. The “Home” button will return the user back to the home screen where they can search, add, and remove to their shopping cart. The “Account” button simply takes the user back to the original profile page screen.

The “Edit” button allows the user to change their stored account information in the database including their username, email, region, and date of birth. And finally, the log out button returns the user to the login page and closes the profile page. To proceed forward again, the user will have to log in to the login page again using either the same or a different account.



SQL Product Data Table

	productid [PK] character varying (30)	category character varying (15)	price real	weight real	createdby character varying (30)
1	Apple	Food	2.99	0.25	Jon
2	Banana	Food	1.99	0.3	Jon
3	Beans	Food	1.99	0.75	Jon
4	Chess	Games	29.99	2	Jon
5	Clue	Games	22.99	1.25	Jon
6	Dish Soap	Household	5.99	1	Jon
7	Hammer	Tools	12.99	1	Jon
8	Jenga	Games	14.99	0.75	Jon
9	Laundry Detergent	Household	12.99	3	Jon
10	Level	Tools	7.99	0.75	Jon
11	Monopoly	Games	24.99	1	Jon
12	Nail	Tools	0.49	0.01	Jon
13	Orange	Food	3.49	0.4	Jon
14	Paint Brush	Tools	4.99	0.25	Jon
15	Paper Towels	Household	9.99	4	Jon
16	Pasta	Food	2.49	0.5	Jon
17	Pineapple	Food	5.99	2	Jon
18	Pliers	Tools	8.99	0.75	Jon

The table can include any category, item, price or weight. The query used to create this demo list will be included in the “test_products_query.sql” file.

SQL Queries in Solution

Table queries:

```
CREATE TABLE Account (Username VARCHAR(30) PRIMARY KEY, Password VARCHAR(30) NOT NULL, Email VARCHAR(50), DOB DATE, Region VARCHAR(30));

CREATE TABLE UserAcc (Username VARCHAR(30) PRIMARY KEY REFERENCES Account ON DELETE CASCADE, AccountType VARCHAR(10) NOT NULL);

CREATE TABLE AdminAcc (Username VARCHAR(30) PRIMARY KEY REFERENCES Account ON DELETE CASCADE, AccountType VARCHAR(10) NOT NULL);

CREATE TABLE Cart (ProductID VARCHAR(300) PRIMARY KEY, Username VARCHAR(30) PRIMARY KEY, Price REAL NOT NULL, Count REAL);

CREATE TABLE Product(ProductID VARCHAR(300) PRIMARY KEY, Category VARCHAR(15), Price REAL NOT NULL, Weight REAL, CreatedBy VARCHAR(30) REFERENCES AdminAcc NOT NULL);
```

Registration Page:

```
INSERT INTO Account (Username, Password, Email, DoB, Region) values('{regUsername}', '{regPassword}', '{regEmail}', '{regDOB}', '{regRegion}');
```

Login Page:

```
SELECT (Username, Password) FROM Account;
```

Home Page:

```
SELECT productid AS Product, price AS Price FROM product WHERE productid LIKE '%" + SearchText.Text + "%';

SELECT productid AS product, price AS Price FROM product WHERE product.category = 'Tools';

SELECT productid AS product, price AS Price FROM product WHERE product.category = 'Food';

SELECT productid AS product, price AS Price FROM product WHERE product.category = 'Games';

SELECT productid AS product, price AS Price FROM product WHERE product.category = 'Bathroom';

INSERT INTO cart (productid, username, price, count) values('{strOne}', '{currentUser}', '{strTwo}', {count});

UPDATE cart SET count = count + 1;

SELECT productid AS Product, price AS Price, count AS Quantity FROM cart WHERE username = '{0}';
```

Profile Form:

```
SELECT Username FROM Adminacc;

SELECT (Email, DOB, Region) FROM Account WHERE Username='{currentUser}';
```

Profile Edit Form:

```
UPDATE Account SET Username='{newUsername}', Email='{newEmail}', DoB='{newDOB}', Region='{newRegion}' WHERE Username='{currentUser}';
```