Project Step 7 Portfolio Assignment

Group: Runtime Errors
Group Members:Sean Murphy, Abraham Serrato Meza
CS:340
Project Step 7 Final Version

Executive Summary

Based on the feedback we received during the first round of review by the TAs we made the following critical edits. We changed the vendorsID to FK rather than a PK, changed some data types to be floats rather than ints, condescended our schema into a single diagram, and removed orderID from the customers entity. In the second round of peer review Jamie Mott brought to our attention several changes that needed to be made to our current logic scheme. She suggested we had inconsistent use of capitalization and that some relationships we had didn't make sense. We added a new entity called Orders, generated a more consistent capitalization scheme(for entities) and deleted the product shipper relationship as well as the vendor customer relationship due to the fact that these relationships didn't make much sense.

In the third round of review some of the biggest changes came from suggestions Zhaowei Si made. Si points out that we failed to meet the following requirements utilization of a SELECT query, Search filter functionality, delete functionality, update functionality. In order to adjust our project in accordance with this feedback we added a search bar, an orders page/admin page in order to meet the requirements for update and delete. We also added a mailingList entity in order to meet the requirement of having a null field. During the 4th round of reviews we received great feedback from Shylton Matta and David Anderson. They both suggested that we didn't have our malingList in the schema, no name attribute for our product entity, and the registers page did not insert into the customers page. We made the following edits that were suggested as well as we changed INSERT queries in DML so that it doesn't add ID, since it's already auto incrementing.

Based on the feedback we received during parts 5 and 6 we implemented; functional insert,update and delete buttons to the admin and orders pages.In order to allow admin usage of editing key information to our ecommerce website. Shylton Matta brought these issues to our attention.He commented asking why these buttons did not work in tandem with our database. We also had a bunch of JS files in our folder that we ended up getting rid of because they didn't serve much of a purpose.

Project and Database Outlines

Overview

RunTime is a company that operates as a niche computer parts/ computer accessories online webstore. RunTime will provide sales of electronics to a niche market of customers that are interested in high end products. The company does roughly around 5,000 sales per year with annual profits of roughly 4 Million US Dollars. RunTime will need a backend database that can support these specific needs: customer details, suppliers, shipping companies used and the types of products sold. This information will be critical in order for the company to provide an excellent customer experience.

We decided to focus mainly on the Admin side implementation of what the webpage would look like.

Purpose of Entity1: A container for customer data that will fulfill a customer's demands.

Entity1: customers:

- customerID: int(11), auto_increment, unique,not NULL, PK
- email: varchar(255), not NULL
- paymentTypes: varchar(255), not NULL
- address: varchar(255), not NULL
- phoneNumber: varchar(255), not NULL
- password: varchar(255), not NULL
- mailingList: boolean default NULL

Relationships:

customer←→ orders: One to Many: A customer may have many orders but an individual order can only be tied to a single customer.

Purpose of Entity2: Hold data of products we have available for sale to our customers.

Entity2: products:

- productID: int(11), auto_increment, unique, not NULL, PK
- price: float, not NULL
- inStock: boolean, default NULL
- productType: varchar(255), not NULL
- vendorID:int(11), not NULL, FK
- prodName: varchar(255)

Relationships:

products ←→ orders: Many to Many: Any number of products can be related to any number of orders.

products←→ vendors: One to Many: There are several different types of a product but each type only has one type of vendor.

Purpose of Entity3: Contains info of our current shipping providers.

Entity3: shippers :

- shipperID: int(11), auto_increment, unique, not NULL, PK
- orderID: int(11), not NULL, FK
- shipCost: float, not NULL

Relationships:

shippers ← → customers: 1 to Many: Each provider can have multiple customers that they will provide service to and each Customer only has one type per order.

Purpose of Entity4: Contains info of our current product suppliers.

Entity4: vendors

- vendorID: int(11), auto_increment,not NULL, FK
- productType: VarChar(255), not NULL
- isCurrentVendor: Bool, not NULL default TRUE

Relationships:

vendors ←→products: One to Many: There are several different types of a product but each type only has one type of vendor.

Purpose of Entity5: Will hold information regarding customers orders.

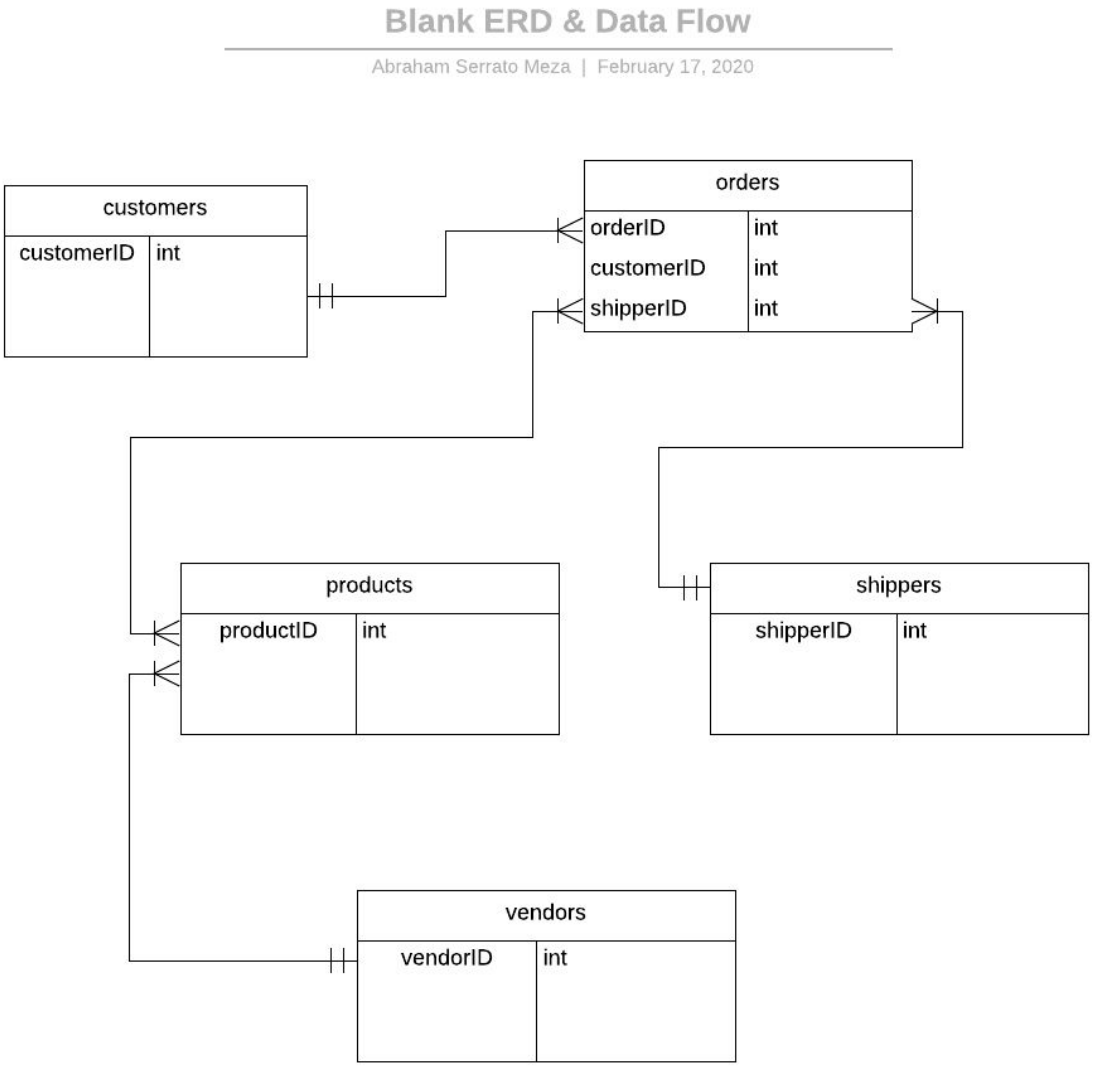
Entity5: orders:

-orderID: int(11), auto_increment, unique, not NULL, PK
-customerID: int(11), FK
-shipperID: int(11), FK
-productID: int(11), FK
-quantity: int

Relationships:

orders←→ products: Many to Many: Many orders can be tied to many different products.

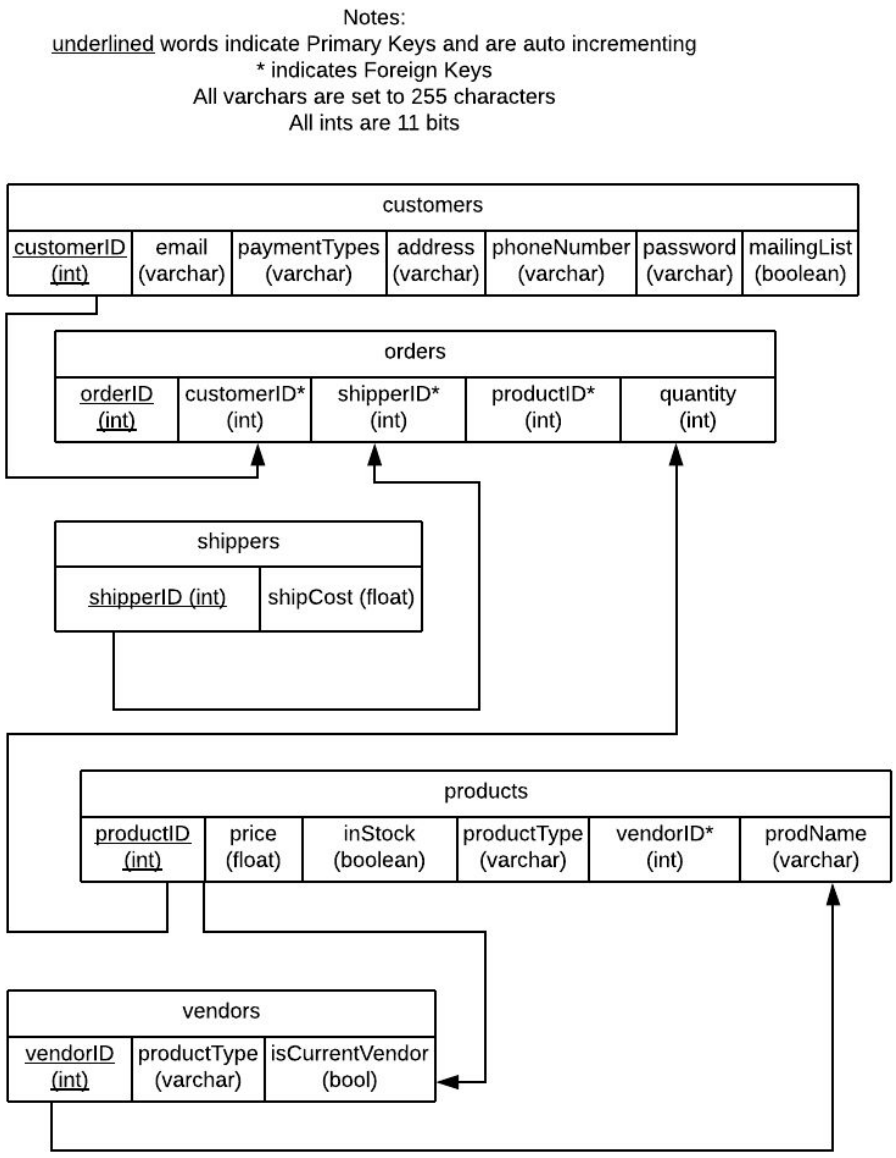
ER Diagram



Schema

List of basic info:

customers(customerID, email, paymentTypes, address, phoneNumber, password, mailingList)
products (productID, price, inStock, productType, vendorID, prodName)
shippers(shipperID, shipCost)
vendors(vendorID, productID, productType, isCurrentVendor)
orders(orderID, customerID, shipperID, productID, quantity)



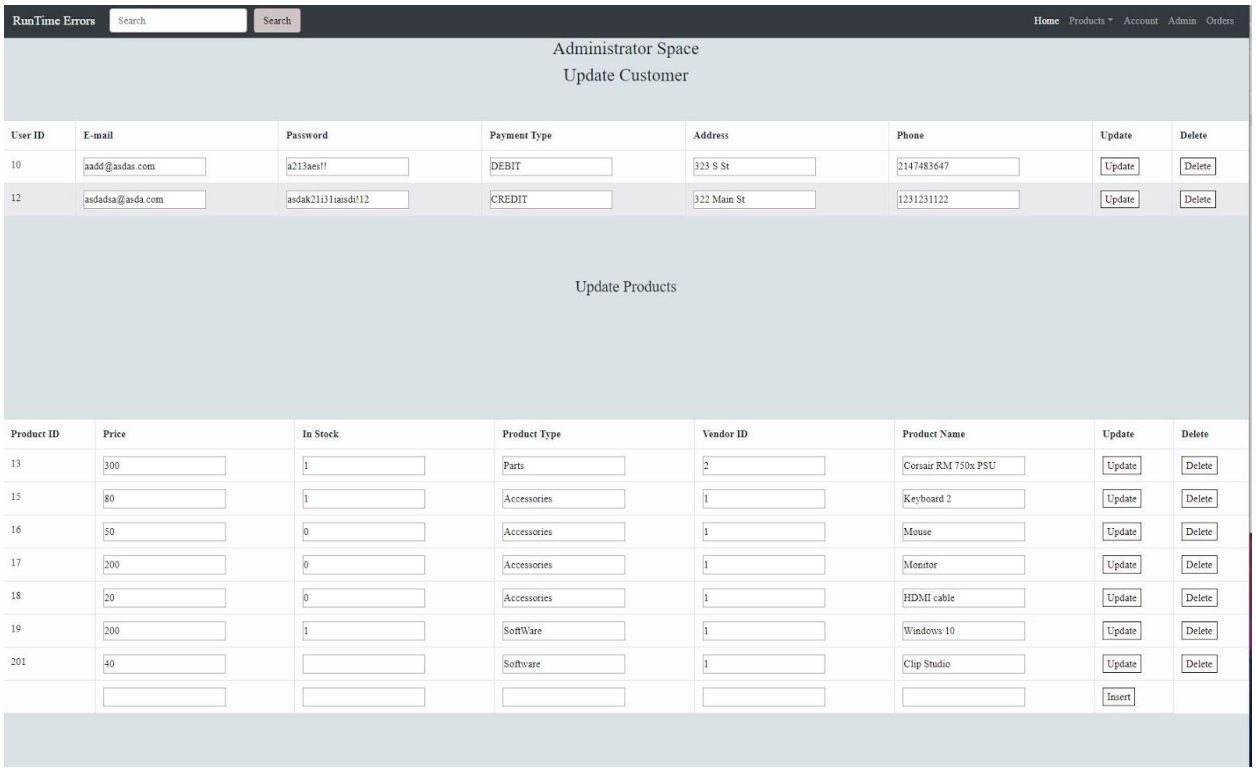
UI Screenshots with Informative Titles

(With the search bar you can essentially browse any page but we did not implement this)

Home page Just (gives user general information about the website).(browse/display)

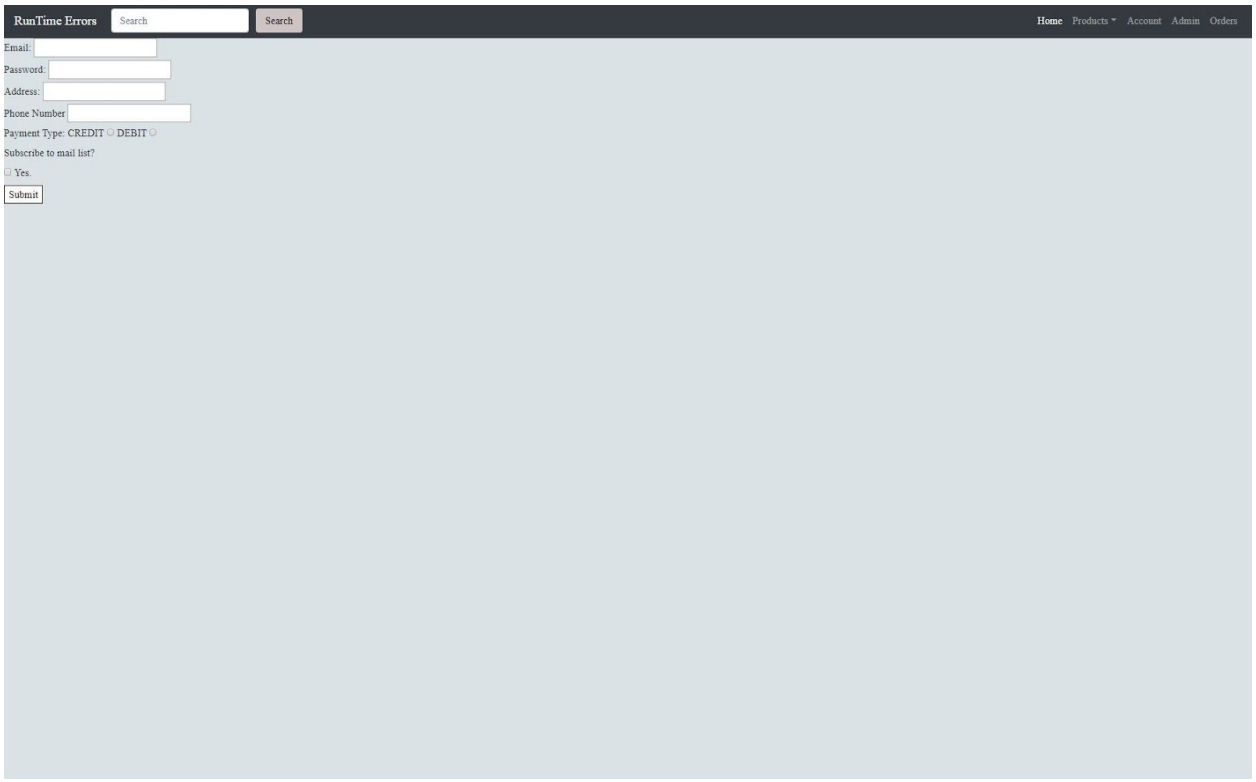
In the top menu you will find links to the other pages in this website.
Descriptions are as follows:

Admin page (Allows for Insertion,deletion and update of database Users and Products that are currently within our system).(create,insert,add,delete)



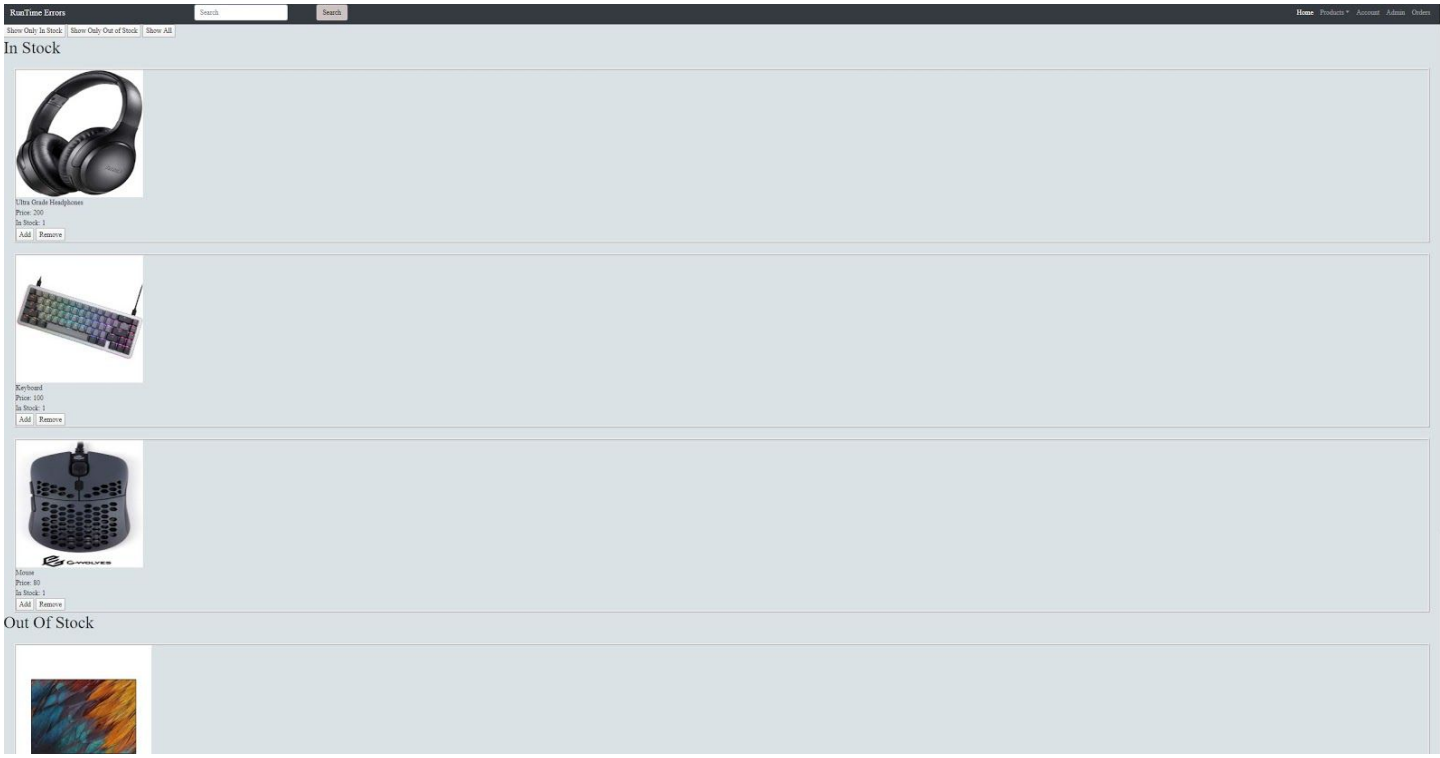
Account page(Is non functional at the moment would theoretically allow users to login that already have an account with us).(“create”)

Registration Page(Allows users to create an account with us. The created account is then inserted into the customers table on both the front and back end.)(“create”)

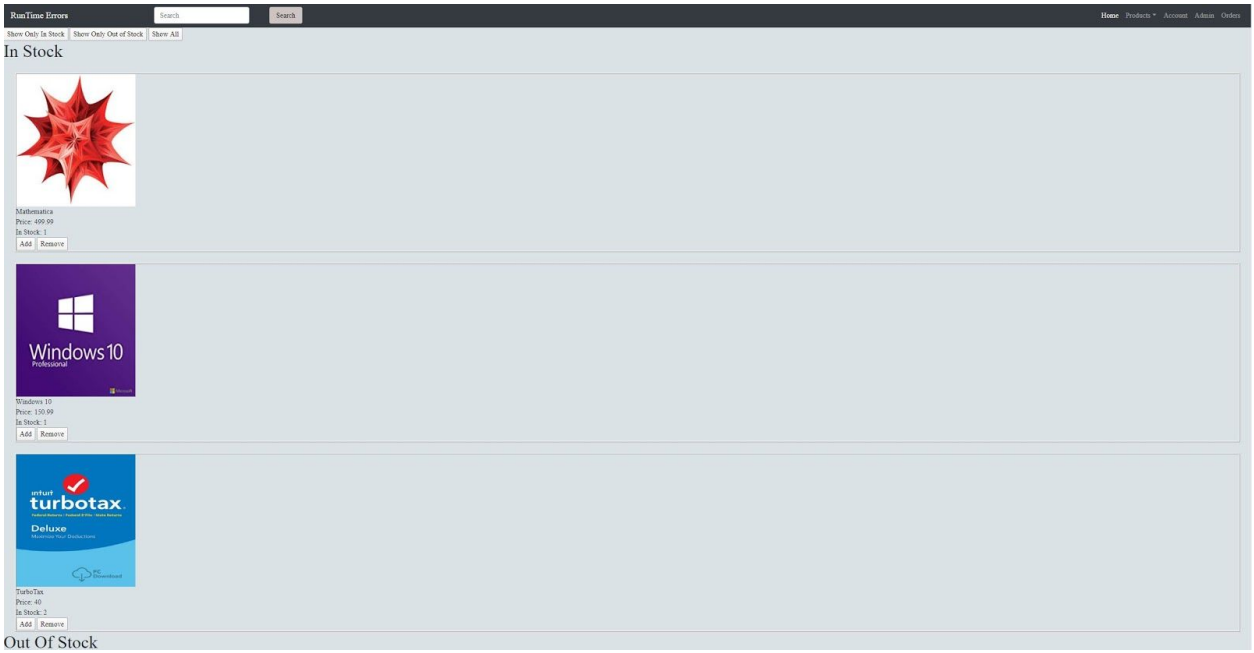


Product Pages(Software,Computer Parts, Accessories pages)
(images are updated based on ID (please beware of this)).

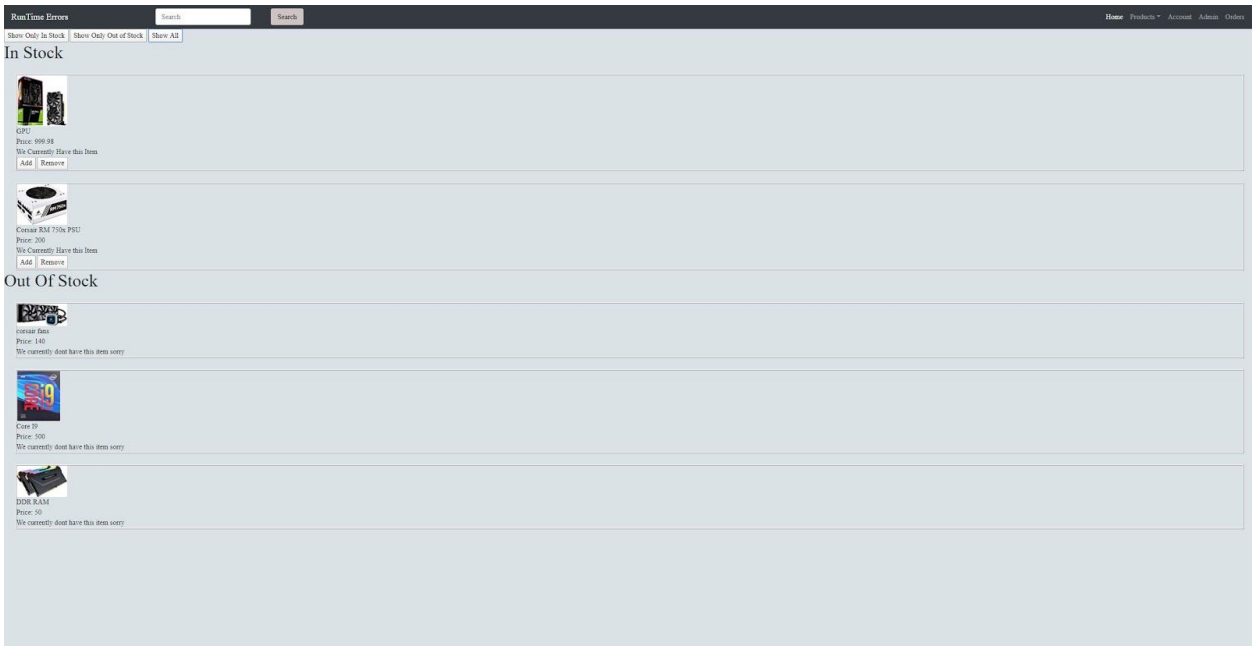
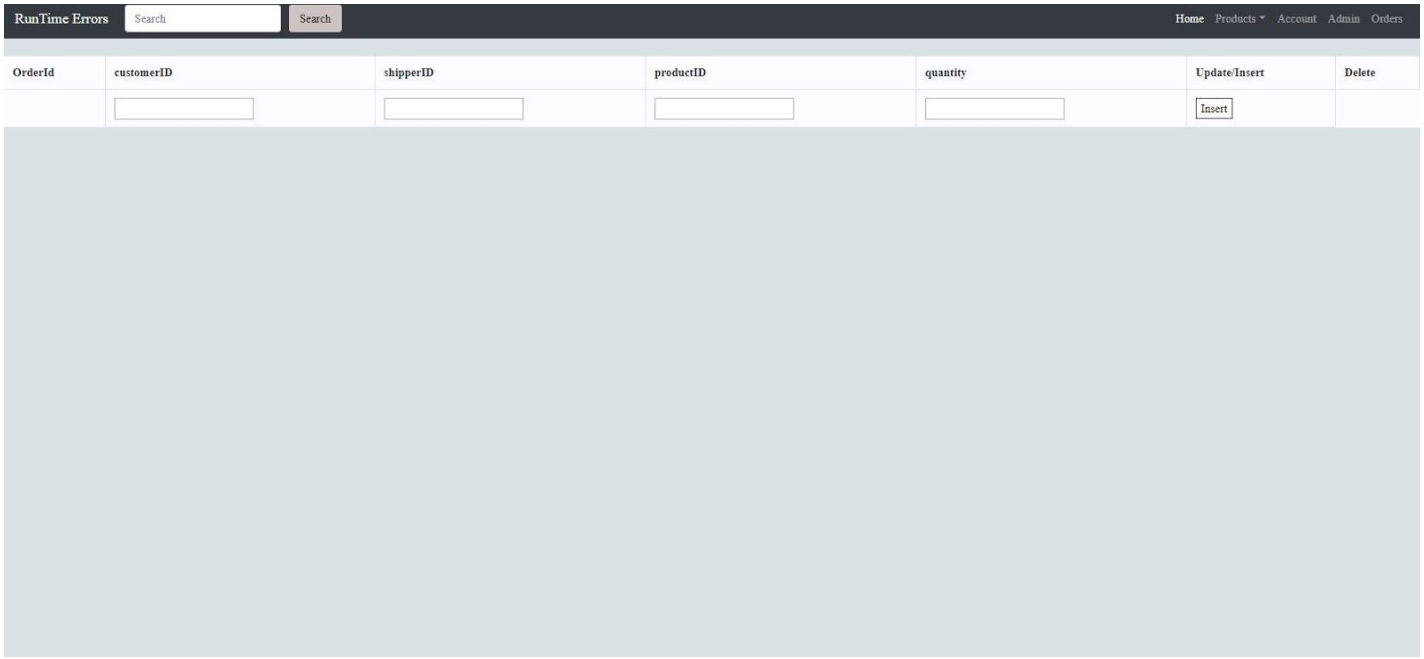
Accessories Page(if we allowed for a user to sign in we would have essentially had a shopping cart that users could have used to add items to there cart from these lists). That being said item images are updated placed as in stock or out of stock when edited in the Admin table or on the backend).(display/filter)(with buttons)



Software Page(if we allowed for a user to sign in we would have essentially had a shopping cart that users could have used to add items to their cart from these lists). That being said item images are updated placed as in stock or out of stock when edited in the Admin table or on the backend).(display/filter)



Computer Parts Page(if we allowed for a user to sign in we would have essentially had a shopping cart that users could have used to add items to their cart from these lists). That being said item images are updated placed as in stock or out of stock when edited in the Admin table or on the backend).(display/filter)



OrdersPage(displays information based on an order that we have in our system allows for insertion, deletion and update of the order into orders table on the backend of our database).(Create,insert,update,delete)