

|  |
| --- |
| Guilded Rose Boutique  Project |
|  |
| June 8  Authored by: Judith M Carbonell Alfonso |



# Project Assessment

The Gilded Rose Online Store

The Gilded Rose is small boutique inn on the outskirts of a prominent city that buys and sells items of the finest quality. The manager of the gift shop, Allison, is looking to create an online store that allows people from afar to purchase her wares.

Create a simple proof-of-concept e-commerce site

* Lists the items for sale
* Allows an authenticated user to purchase them.
  + Purchase only 1 item at a time by clicking a “Purchase” button.
  + Depletes the stock of that item by 1.
  + Not need to implement a shopping cart or use any payment providers.

## Deliverable

* A system that displays a listing of items and allows authenticated users to purchase them one at a time.

## Technical Requirements

* Use either ASP.NET Core Razor Pages / MVC or ASP.NET MVC 5
* Web app that can run/tested locally on Windows, Mac, or Linux
* Use of a database is optional

Application architecture decisions

## Project implemented using ASP.NET Core 3.1 Web Application with Razor Pages

* Clean architecture
* Open-source with a big community contributing and Flexible deployment
* Multiplatform: runs on macOS, Linux, Windows, and Docker
* Performance: It's faster than other popular web frameworks
* Modern & innovative: allow runtime components, APIs, compilers and languages evolve quickly, provide a stable and supported platform to keep apps running.

## MVC:

* Application logic is divided into three components on the basis of functionality:

Models, Views and Controllers.

* CRUD-based Regular Application

## Monolithic applications

* Initially the requirements are simple and the application expect to handle a limited amount of traffic
* The idea is starting with a monolithic architecture and then later scale out into a microservice architecture if it is necessary.

## SQL database:

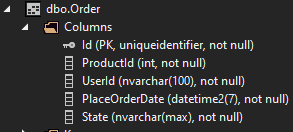
* Need to store a lot of relationships
* Transactions & Data Consistency

## Libraries:

* Bootstrap (responsiveness and aesthetics or user interface “look and feel”)
* Entity Framework (object-database mapper that supports LINQ queries, change tracking, updates, and schema migrations)
* SendGrid (world's largest cloud-based email delivery platform)

# Requirements and Decisions

Created a Table to manage the products of the store with the following fields:

## Considerations:

* Is not possible to buy a product if it is out of stock (the quantity is equal 0)
* -The user has to be authenticated to purchase an item.
* If a user is navigating the site anonymously and click the button of purchase, will we automatically redirected to the login page, after login the purchase will be effective.
* If the product is out of stock, there will be a sign with that message and no button for purchase
* When an authenticated user clicks the purchase link or put the URL for purchase on the browser, the system will first check if there is availability of the product requested, if there is no items, the system will alert that. If the product is available the system will place the order, notify the user by email about the purchase and deplete the amount of product in the Quantity field of the Product table.

With the use of the following keywords we will guarantee that the controller and the actions are accessed by the correct user:

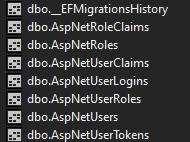
* [Authorize] – The user has to be authenticated to access this resource. ***Purchase***
* [Authorize (Roles = "Administrator")] – The user has to belong to the Administrator group to access the ***Update, Create or Delete*** actions
* [AllowAnonymous] – Anyone could see this options, Index, Details, Blog, About

Publish the web app as Portable Runtime-dependent applications for cross-platform.

# Shortcuts

ASP.NET Core Identity:

* Facilitates the user management, authentication and authorization.



SendGrid:

* Facilitates the process of sending emails.

**Entity Framework Core Scaffold-DbContext:**

* Helps to create models, controllers and views for CRUD using an existing Database structure

# Authentication mechanism

## ASP.NET Core Identity:

* Is an API that supports user interface (UI) login functionality.
* Manages users, passwords, profile data, roles, claims, tokens, email confirmation, and more.
* Users can create an account with the login information stored in Identity
* They can use an external login provider: Facebook and Google.

Using OAuth and token-based security can help build a better permission checking across the user base.

It facilitates the registration process for the users, helping to maximize the user acquisition.

Test Performed

# Authentication:

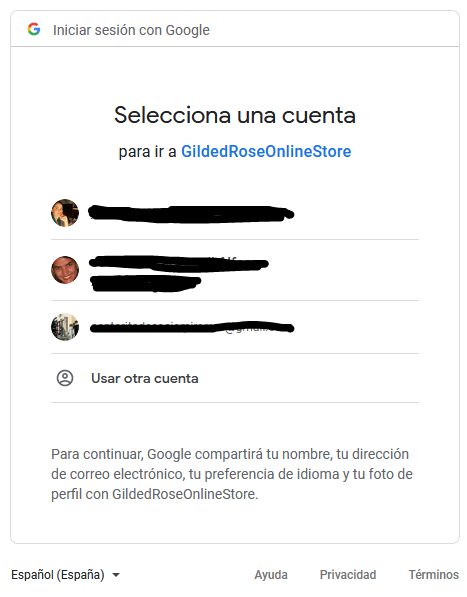
The user have the following options for create an account in the website:

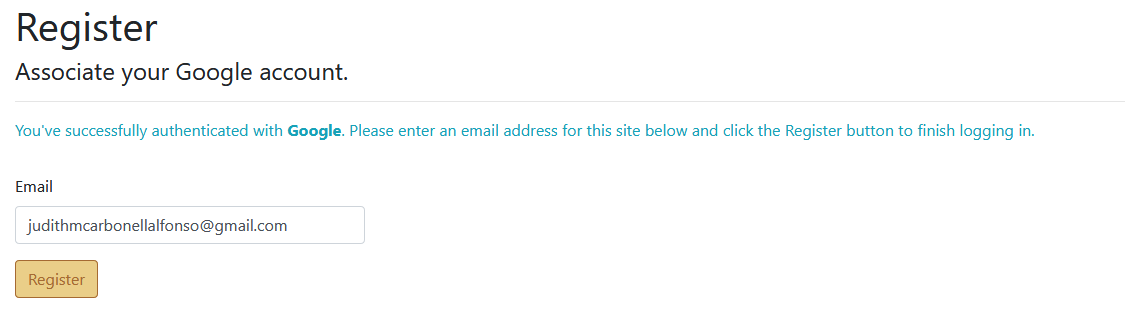
1. Register creating a new account, providing an email and password
2. Using OAuth authentication through Facebook or Google



**Google registration:**

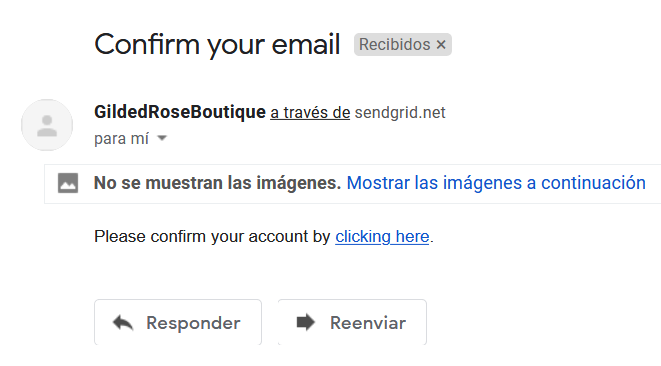
* Click on Google Button and shows next screen to choose which Google account the user wants to link to Guilded Rose Boutique



* After choosing the account, the user has to click on Register to confirm that wants to link his account to the website. 
* After clicking on the Register button, the next screen ask to check email an confirm registration:



* The email has a very basic structure, and could be improve in the future to better fits to the corporative identity of the client.



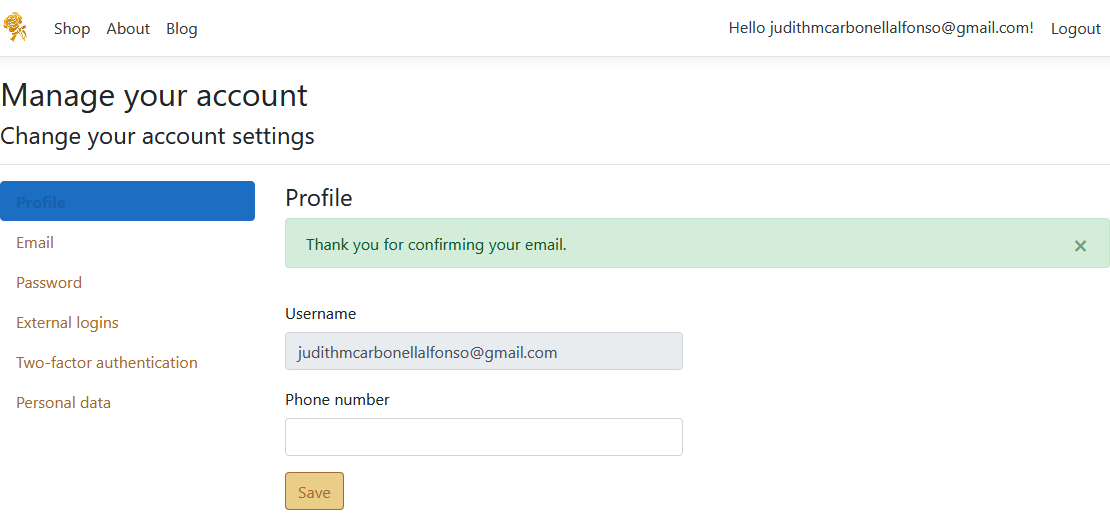
* Token based link has an structure similar than the following:

<https://localhost:44370/Identity/Account/ConfirmEmail?userId=5cd6a963-9100-47f4-b286-341a1881d6c3&code=Q2ZESjhHRHBCWEFiK3JwUGtUSldnRjgrT05WZEFTbUZTbk42L3pxSks4b09kSGlUNzBJR1UrMkl6NDZESTJicUZ2Z1RXeGZET2ROQkZHVEJyb2pyNmFwSVUwRkdQSDdEQW03SThNTXNCa2p4TlBlV1ZoQkU3aG5tQ25nNFp3VlVNc09PL1FnK0RuLzY4MVhsWkNqc3c4WTYzSzF6aUd6VkxqaWpHUFc1bUYyek9CUmM1TmZpVGdsQWVHV1pqcWcwNmwxS2NlTVN4K3cxYkJLZGtXQ1ZJcUpiNHMyRTZhR1dyZXR0ZXlIZkJVQk5hNnVGc2huYkpsZHY0TUxPNE9OeVRWVFZ2UT09>

* After confirming the link, is possible to Login to the website using the same method of registration:



* Is possible to Manage your using account if you click on the username besides the Hello and Logout links



Is this view, it is possible to set or change password, configure other elements of the account or even delete it.

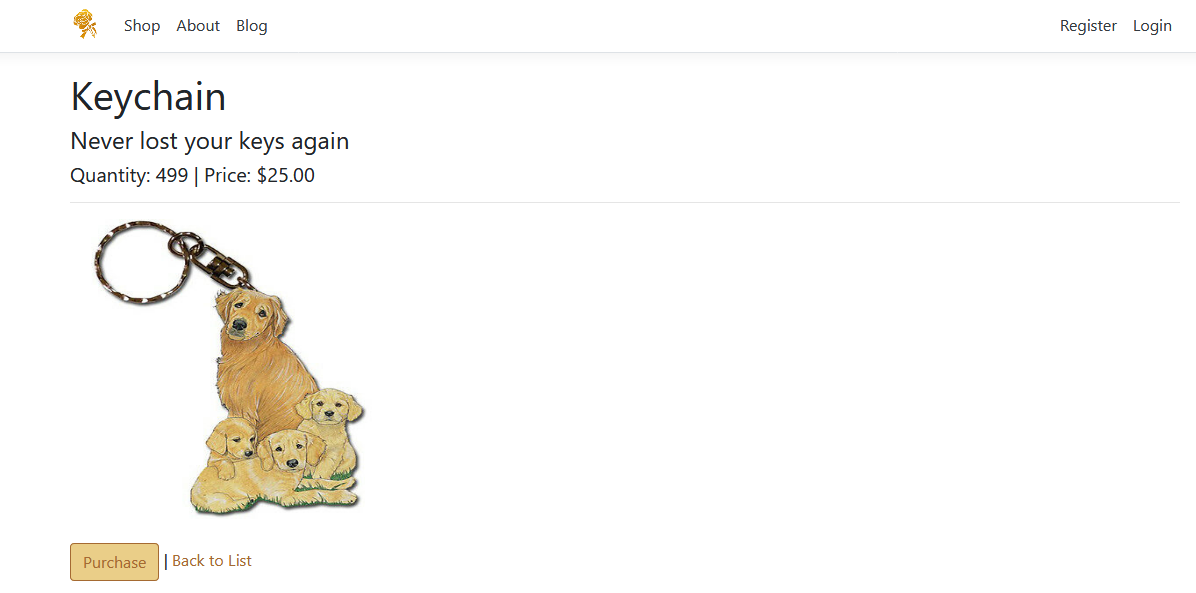
# Display a listing of Products

Created 5 products to populate the list and see the system in action:



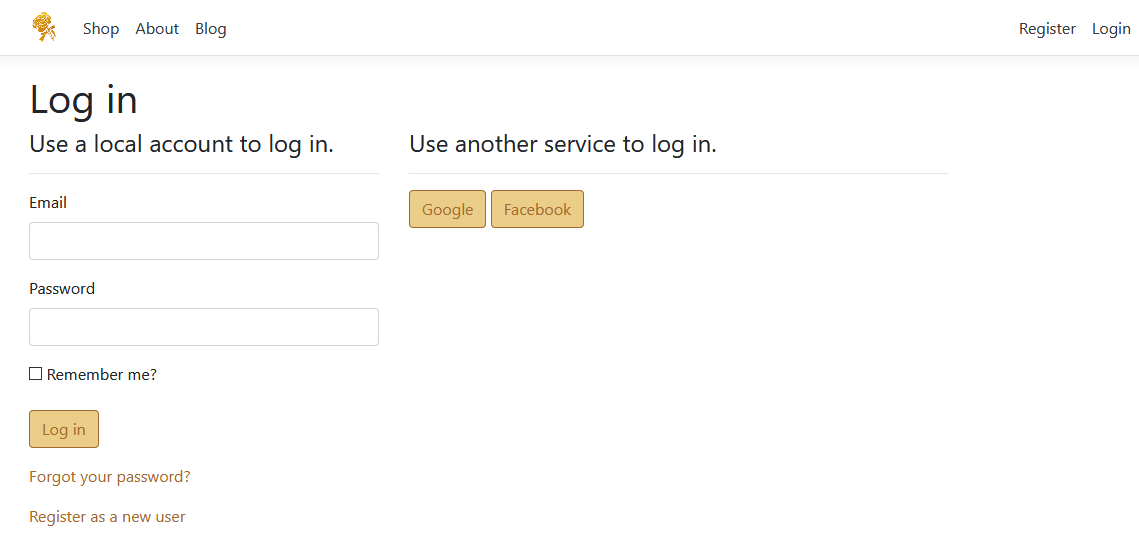
# Detail of Product:

It is possible to click on every box to see details of the product. In this space in the future and if the requirements ask it, you can see different variants of the product and all its properties.

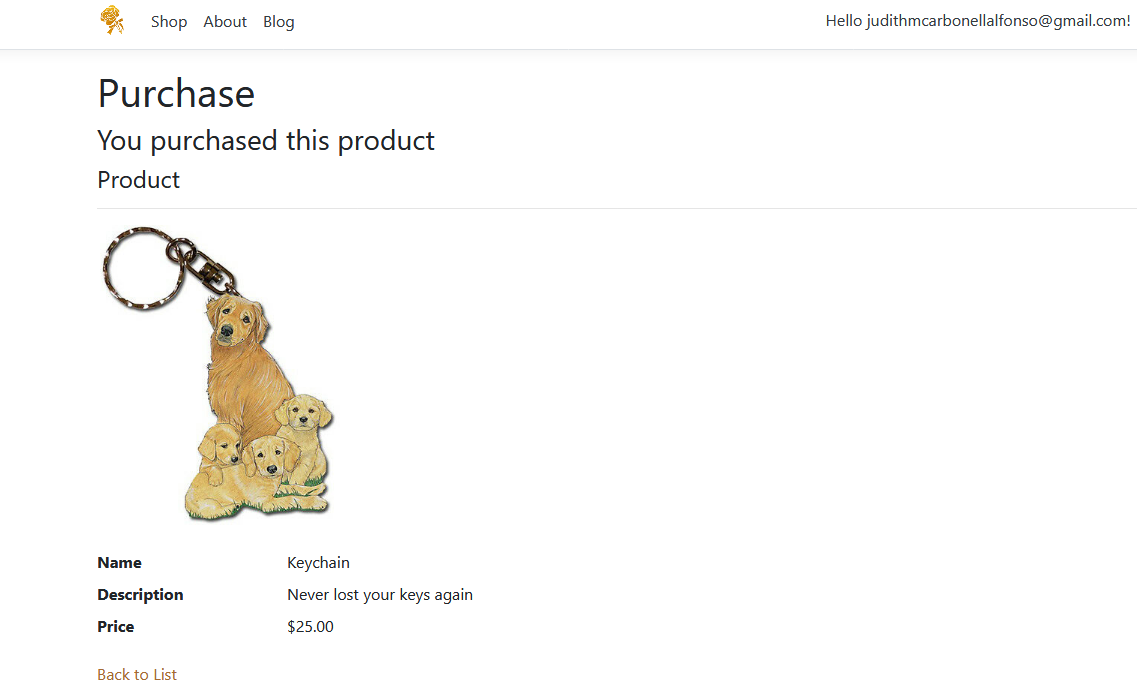


# Purchase:

It is possible to click the purchase button from the Details window or from the Home page. If the user is not authenticated it will redirect to the next window:



After login, it is possible to Purchase the button:

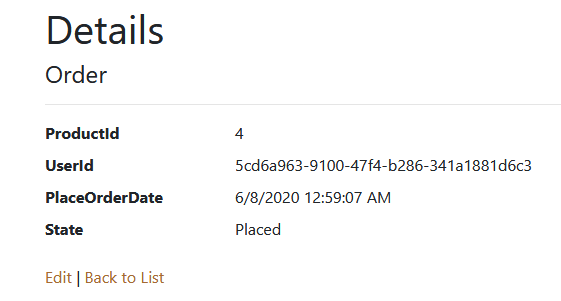


If you enter the detail of the product will see that the quantity depleted by one, There are 498 Keychains now:

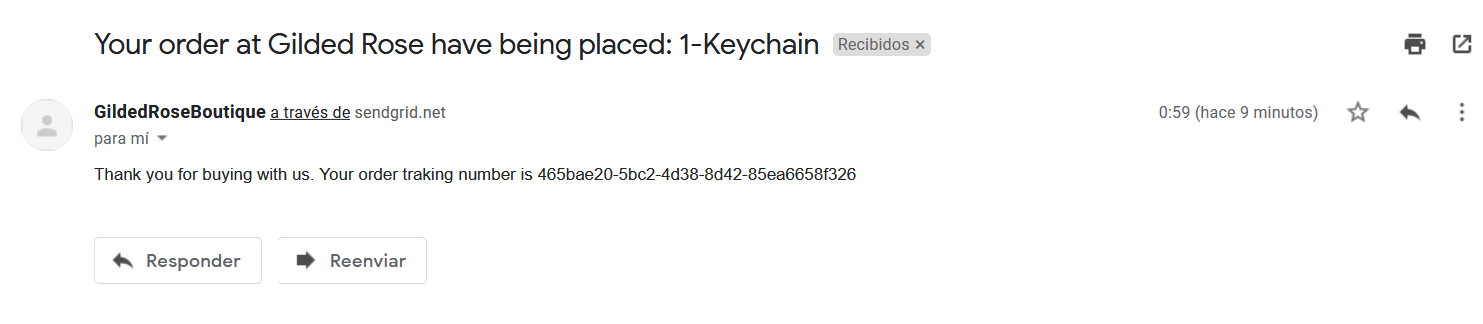




It is possible to check the Order state. In future implementations it is necessary to link this view with more meaningful information related to the order, the user and the product:



The user receives an email with details regarding the order and its different states. In the future this email must use a pretty template with all the required information:

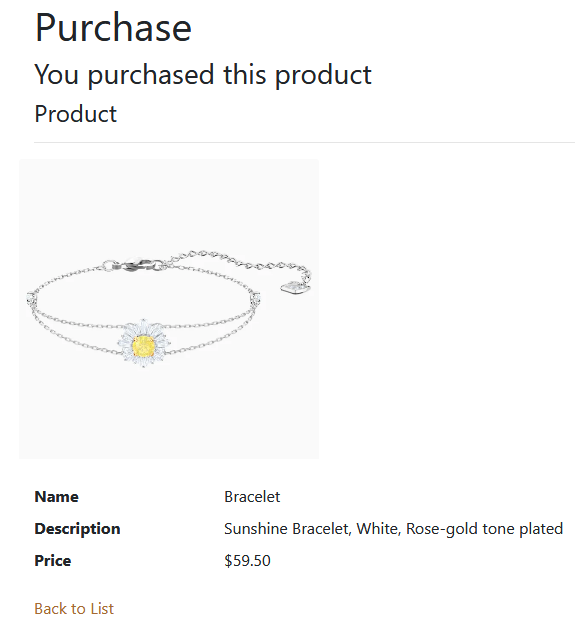


**Purchasing the last item of a product**

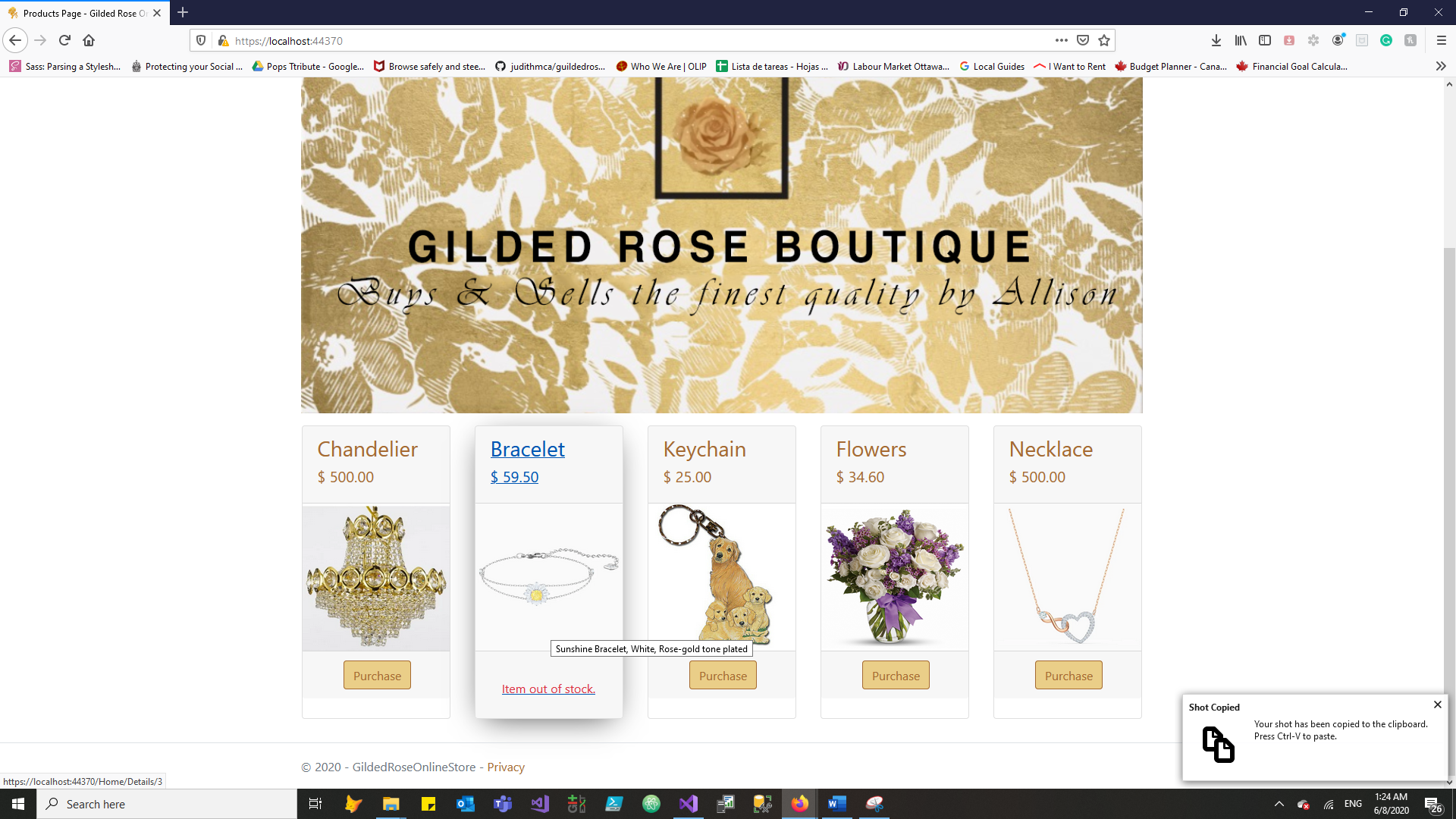
In the next picture we can see that the Quantity of the bracelet is 1. This means that after purchasing this item, the quantity will be 0 and will be not possible to buy it any more until it is restocked:



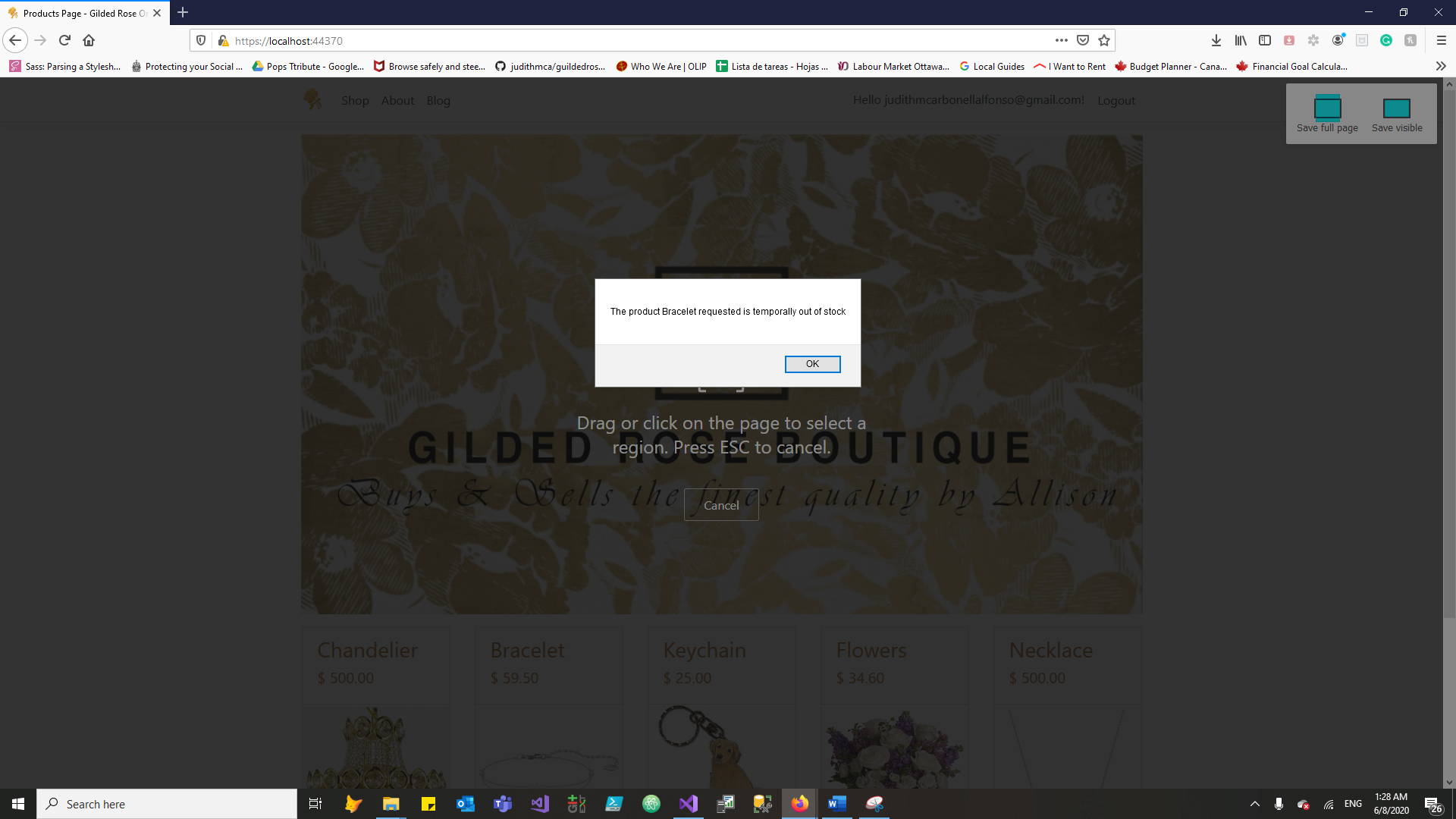
The purchase was effective, the item was depleted and as we can see in the right picture, the quantity is 0 now and is not possible to buy the item, because it does not have a button for triggering that action:



In the home page we can also notice that the Item out of stock sign is on the bracelet and there is no button for purchasing:



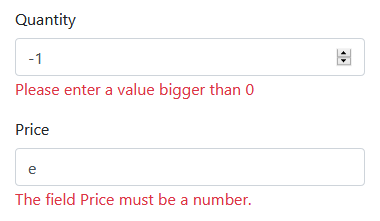
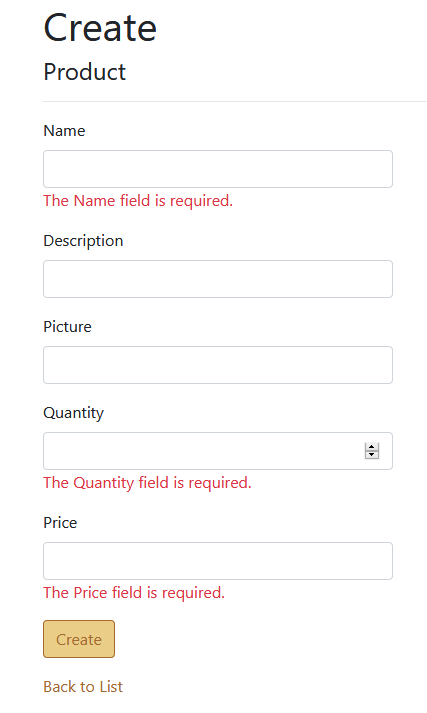
Just in case some user hard code the URL for purchasing the item with no need of clicking on a button, there is the next validation, with the corresponding alert:



In the future this alert will be using Bootstrap style or something else to make it looks more appealing to the user.

**Create product**

The Create Product form has different validations implemented for the Required fields, for the numeric fields type and range, as it shows in the next pictures:



In the future it is necessary to modify the Picture field to an upload file input to be able to upload the pictures and linked it to the model.