

Project Brief

**CS2TP teaching team:** Abinaya Sowriraghavan,

Alexandros Giagkos, Chloe Barnes, Shereen Fouad, Tharindu Ranasinghe

Your team is approached by a business owner that wants you to design and deploy an e - commerce platform to sell their products. Their business has grown rapidly over the last few years, and their current solution is clunky and slow. The ecommerce platform needs to be hosted and must handle customers’ data, manage inventories, and fulfillment customers’ orders. The user interface must be friendly and easy to interact with. The business needs a database that can manage their list of customers and items. Some employees of the business have some experience with Laravel so would prefer a solution developed with [Laravel framework,](https://laravel.com/) however you may use whichever language or web development framework you wish. You are also expected to host their solution for them. You can choose your preferred hosting solution; however, it is recommended that you use the University server provided. They also would like an inventory management system that can be used to manage their inventory of products, and the incoming and outgoing orders. The system will help the business manage and view their stock to better meet customer demand and avoid the expense of overstock or loss of business with stock outs.

# The business (of which you must decide on the name, vision, scope, logo, and product line) requires at least:

* A home page. **– DONE**
* An “About Us” page (or similarly named) to state the vision of the business. **– DONE**
* A page with contact details for the business, with a contact us form to allow customers to automatically send requests to admin. **– NOT DONE**
* A page to display the products that the company sells, with a product search functionality by name or by category. **– DONE (IN TERMS OF SEARCH BY CATEGOGRY SECTION)**
* A page to sign up to use the website as a customer or admin, with password change functionality after login. **– DONE**
* A page to log in if they already have an account (customer or admin). **– DONE**
* A page to display the user’s current basket. **– DONE**
* A page to display the user’s previous orders. **– DONE**
* A page for customers checkout to complete **dummy** payment and submit the order. **– DONE**

# This also gives rise to the following requirements:

* The company must have a brand including logo, colour scheme, consistent font, etc. (consider this in the design phase of your project); **– DONE**
* The home page must display the categories of products, and clicking one of these categories will take the user to the products page which will only display that category of products. **– DONE**
* There must be at least five different categories of products listed within the platform, with each

category having at least five different products listed. - **– DONE**

* The products page must include a way to search and filter the products, depending on the category or other filters. **– DONE**
* The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur. **– DONE**
* The website must be hosted online and must always be in a functional state. **– NOT DONE**
* The website must be secure to protect customers’ data. **– DONE**
* You may decide in your teams an additional functionality that you can offer to the customer.

There must be at least two types of users who each have a different view of the website and the functionality available to them – an admin and a customer. **– DONE**

# The customers must be able to:

* sign up for the first time and use the website as a customer, with password change functionality after first login. **– DONE**
* change their password for future security. **– DONE**
* search and filter products by names of categories and/or by price range. **– DONE**
* place an order, which submits their basket and registers it as an order in the database, along with the total price (it is NOT necessary to include any method of payment, either by taking details or through online payment systems); **– DONE**
* return a product which they have already purchased in previous orders. **– DONE**
* add, update, or remove items to/from their basket. **– DONE**
* view, add, delete and update their details (personal information). **– DONE**
* check the status of their past orders. **–DONE**
* rate and review individual products and the overall service provided by the website. **–DONE**
* you may decide in your teams an additional facility/service to customers to improve their online shopping experience. Use the creativity of your team!

# The admin must be able to:

* sign up for the first time and use the website as a customer, with password change functionality after first login. **– NOT DONE**
* change their password for future security. **– NOT DONE**
* process an order, by checking customers transactions and processing shipments. **– NOT DONE**
* view, add, delete and update customers’ details. **– NOT DONE**
* effectively operate the inventory management system (explained below)

The web application will be supported by an inventory management system, to help manage website requests asynchronously. This system will be used by admin ONLY to manage stock levels and other associated actions. The application will include several features that are essential to the business including, (but not limited to):

* A product listing must include an image of the product, a price, a description, an indication of the stock level (either units available, or “in stock/low stock/out of stock” (where you decide the thresholds), or your own choice) and any further information if needed (such as if selling clothing or shoes, the individual sizes – which must each **– DONE**

have a stock level tracked in the back end);

* Inventory alert system: If an item goes out of stock, or under a specified threshold this must be automatically communicated to the user in an effective and friendly way. **– DONE**
* Whenever an incoming or outgoing product is entered or removed, the stock number will be automatically updated. **– NOT DONE**
* The application should be able to generate reports in real-time that will be displayed on the website about current stock levels as well as incoming and outgoing orders for all products. **– NOT DONE**
* The system will allow authorized users (admin) to search, filter and view the status of selected products and orders. **– NOT DONE**
* The system will allow authorized users (admin) to add, edit or remove products to/from the inventory through the website. **– NOT DONE**
* The system will allow authorized users (admin) to initiate and process an incoming order in a friendly way. Following this entry, the stock level will be automatically updated. **– NOT DONE**
* You may decide in your teams an additional functionality for the inventory management system, on top of the above features. Use the creativity of your team!

The web application must be deployed in a working state and accessible at all times. The code for the website must also be professionally managed with a version control system – GitHub/GitLab – and its use must be documented in a README file for accessibility and maintainability.