Full-Stack Web Development Micro-credential

Capstone Project - June 2022

Problem Statement

During the pandemic, lots of businesses got affected, especially physical stores that started to move online to open new sales channels. You just landed a project to build an e-commerce web application for an apparel store that wants to offer its products online, this application will serve two main personas:

- Admin (the business owner): wants to be able to list their goods under specific categories on the application, and process customer orders.

- Customer: wants to be able to search for products, add them to their cart, and checkout to create an order successfully.

You are given a project scope with some mandatory requirements and nice to have requirements and are asked to develop this project within 6 weeks.

Table of Contents

Get Started

All illustrations by Dmitry Zubanov from Ouch!

Overview

The capstone project is the last milestone for you to successfully complete all the requirements of the “Full Stack Web Development Micro-credential”. This is a graduation project that **all learners have to complete to be able to earn the program certificate**.

The project is meant to give you the chance to put everything you learned throughout the program into practice and build an end to end application.

The project requirements section has all the mandatory features which all learners must submit. A bonus features section is also added to the end of the project, these are additional tasks that will grant you additional points if delivered (partially or fully).

As a learner, you might need to research and learn about additional topics listed within the project. This is a chance for you to learn how and where to look for information.

You are asked to deliver this project within 6 weeks.

High Level Requirements

The following are **mandatory requirements** that you need to implement in your project. For detailed information about each requirement, visit the Detailed Requirements page.

Admin

- An Admin account should be created by the developer.
- The admin should be able to log into the website.
- The logged-in admin should be able to create, list, update, and delete different product categories and subcategories.
- The logged-in admin should be able to create, list, update, and delete different products.
- The logged-in admin should be able to add and remove products from subcategories.
- The logged-in admin should be able to see a list of customer orders.
- The logged-in admin should be able to view the details of customer orders.
- The logged-in admin should be able to update any order's status.
- The logged-in admin should be able to log out of the website.

Customer

- The customer should be able to see a list of products on the website.
- The customer should be able to search for products by name.
- The customer should be able to filter products.
- The customer should be able to view a specific product's details.
- The customer should be able to register an account on the website.
- The customer should be able to log into their account on the website.
- The logged-in customer should be able to add a product to their shopping cart.
- The logged-in customer should be able to view the products added to their cart.
- The logged-in customer should be able to remove a product that was added to their shopping cart.
- The logged-in customer should be able to complete their order by entering their shipping information and confirming their order.
- The logged-in customer should be able to view a list of their orders and their statuses.
- The logged-in customer should be able to view the details of their orders.
- The logged-in customer should be able to log out of the website.

Other (non-functional) Requirements

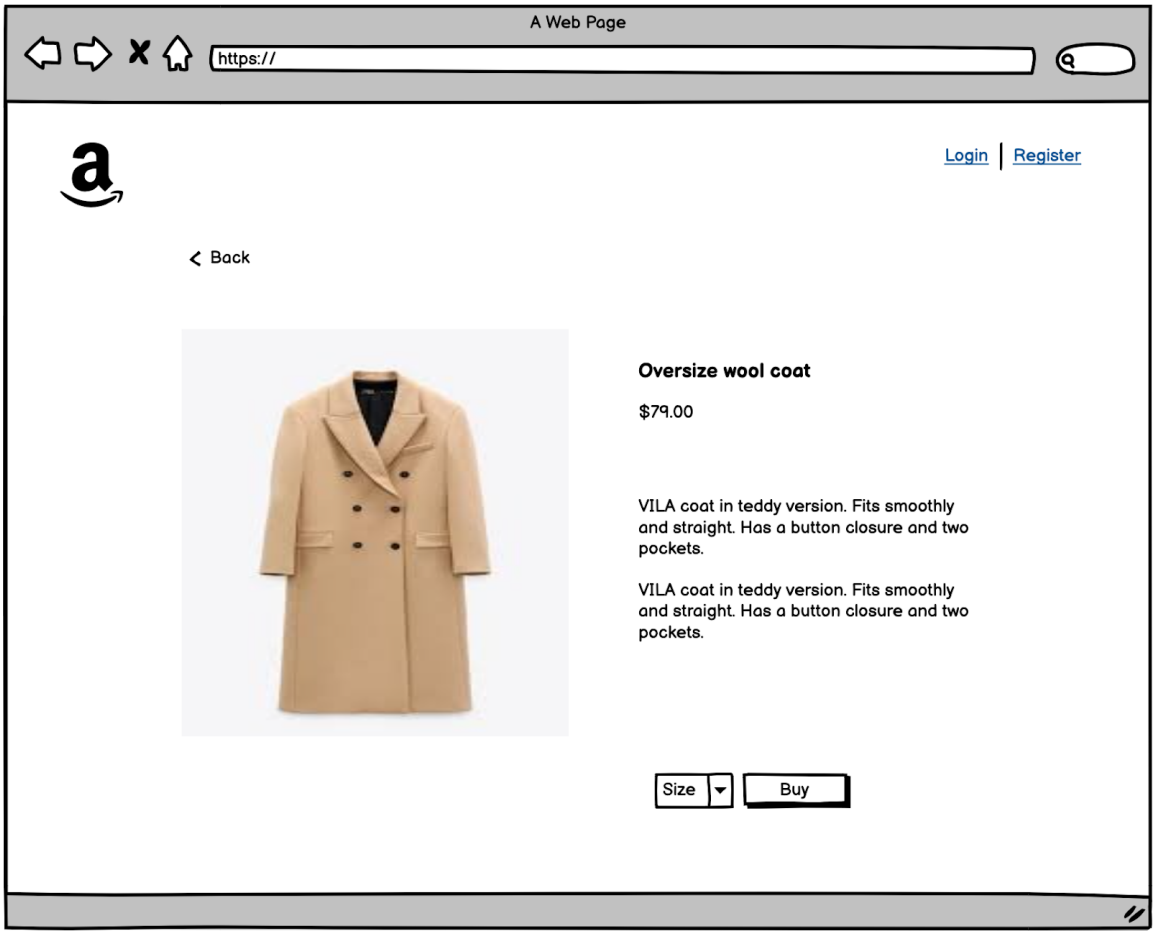
- Admins and Customers must have different interfaces (pages).
- Code should be under version control (using git)
- The application should be SEO friendly
    - Correct HTML tags should be used
    - Good page speed score
- The application should be accessible from the Internet
- The application should be secure
    - Safe against basic SQL injection attacks
    - Accessed through HTTPS
- The application should be fast, and page render time should be below 3 seconds
    - Optimized Database Queries
    - Clean code
- The website should look nice and be user friendly

Example Wireframes

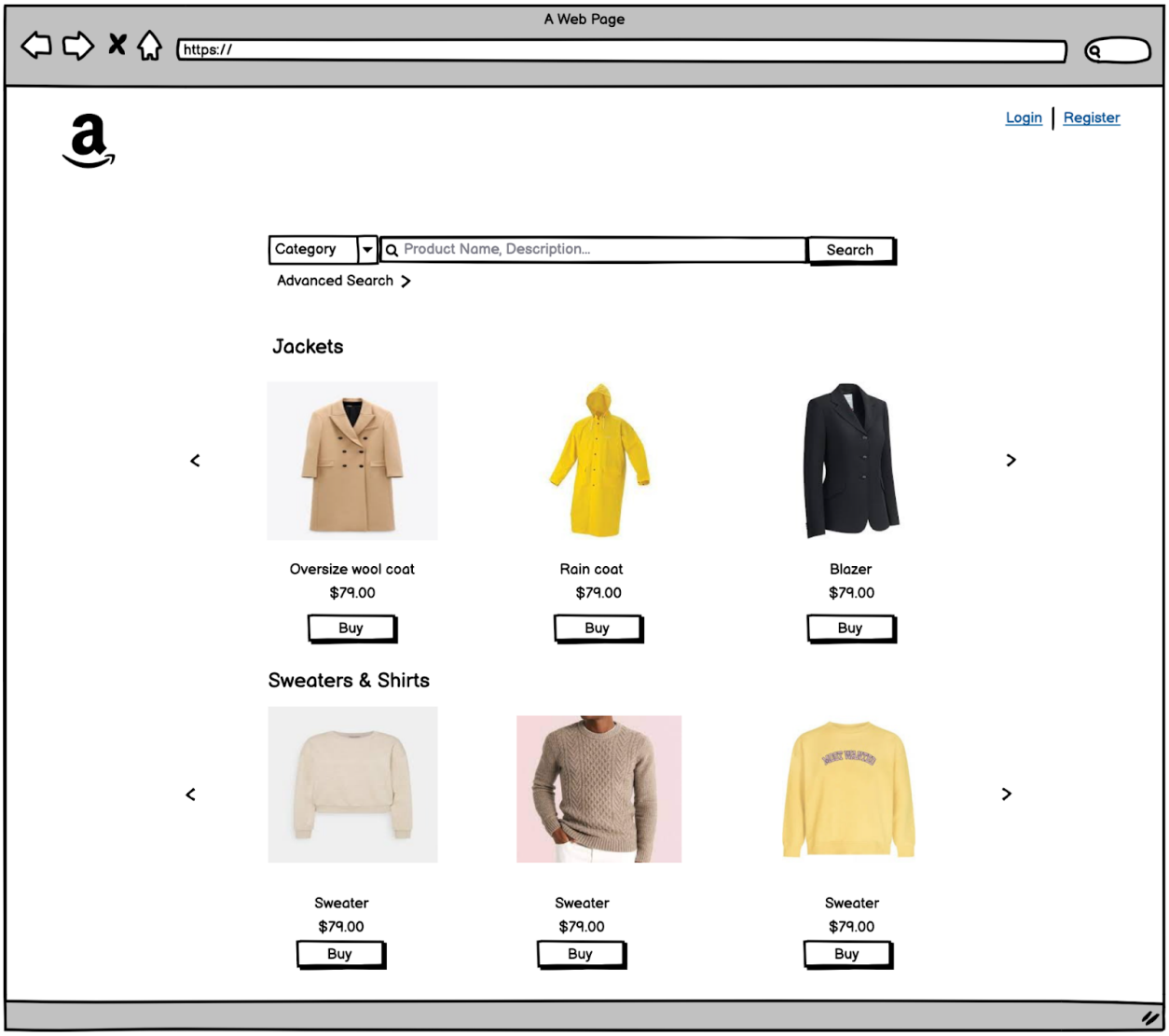
“A wireframe is a schematic or blueprint that is useful for helping you think and communicate about the structure of the software or website you're building.”

The following are examples of how you can create wireframes to help you imagine how your website will look like. These are **just examples, so you should create your own designs**.

For more information about wireframes, you can visit the following link, or check other resources online:
https://balsamiq.com/learn/articles/what-are-wireframes



Example of a View Product Page



Example of a List Products Page

Detailed Requirements (Stories)

This section contains more details for all the functional requirements for the Admin and Customer persona that are defined in the High Level Requirements section.

Make sure you pay attention to the description of each requirement, and the acceptance criteria to get an idea how each requirement should function.

It is very recommended that you create a checklist of all the requirements to help you keep track of which requirements you finish, and those that are still remaining.

Stories for the Admin Persona

An Admin account should be created by the developer.	
Description	The developer should create an admin account manually.
Acceptance criteria	- Email is: “admin@edraakmc.com”. - Password is: “edraakMC_admin”. - Account can do all admin actions.

The admin should be able to log into the website.	
Description	The admin should be able to input their email and password and log into the website.
Acceptance criteria	- A login page for admins should be created. - Proper frontend and backend validation should be carried out on the email and the password. - A proper error message should show on the frontend if the login credentials were wrong. - If login is successful, a session should be created for the user, and they should be redirected to the admin order list page.

The logged-in admin should be able to create, list, update, and delete different product categories and subcategories.	
Description	The logged-in admin should be able to access a page that allows them to view current categories and subcategories, add new ones, update existing ones, or delete them.

Acceptance criteria	- Categories and subcategories can be seen in a list.
	- New categories and subcategories can be created.
	  - A category has the following attributes:
	  - Name.
	  - A subcategory has the following attributes:
	  - Name.
	  - A category has many subcategories.
	- Categories and subcategories can be modified.
	- Categories and subcategories can be deleted.
	- Different categories cannot have the same name.
	- Different subcategories that belong to the same category cannot have the same name.
	- The following categories should be created:
	  - Women.
	  - Men.
	  - Kids.
	- The following subcategories should be created for each category:
	  - Jackets.
	  - Pants.
	  - Sweaters & Shirts.
	  - Shoes.
	  - Bags.
	  - Accessories.
	- Success messages should show after successful addition, update, or deletion.
	- Error messages should show after failed addition, update, or deletion.
	- A confirmation message should show upon attempting deletion.

The logged-in admin should be able to create, list, update, and delete different products.	
Description	The logged-in admin should be able to access a page that allows them to view current products, add new ones, update existing ones, or delete them.

Acceptance criteria	- Products can be seen in a list.
	- The list should be paginated, and should show 15 products per page.
	- New products can be created.
	  - A product has the following attributes:
	  - Name.
	  - Description.
	  - Price.
	  - Image.
	  - Size (optional for products that need size - values are: S, M, L, XL, or XXL).
	  - Return policy (optional, text field).
	  - A product can belong to many subcategories.
	- Products can be modified. If a product's price was updated, it should not affect the orders that were already created.
	- Proper frontend and backend validation should be carried out on the entered fields.
	- Products can be deleted. If a product is not used inside orders, it gets deleted. If it is used inside orders, it should be soft-deleted. Soft deletion means that the order will not be actually deleted, but will be marked as "disabled" and will no longer show up for customers.
	- Success messages should show after successful addition, update, or deletion.
	- Error messages should show after failed addition, update, or deletion.
	- A confirmation message should show upon attempting deletion.

The logged-in admin should be able to add and remove products from subcategories.	
Description	The logged-in admin should be able to add products to subcategories and remove them from them.
Acceptance criteria	- Products can be added to subcategories. - Products can be removed from subcategories. - A product can be added to more than one subcategory. - A subcategory can have more than one product. - A product cannot be added to the same subcategory more than once. - Success messages should show after successful addition, or removal. - Error messages should show after failed addition or removal.

The logged-in admin should be able to see a list of customer orders.	
Description	The logged-in admin should be able to access a page that allows them to see a list of customer orders that shows multiple order details.

Acceptance criteria	- Orders can be seen in a list.
	- The list should be paginated, and should show 15 orders per page.
	- The list shows the following details for each order:
	  - Order ID.
	  - Customer name.
	  - Order creation date.
	  - Number of items in order.
	  - Total order cost.
	  - Order status.

The logged-in admin should be able to view the details of customer orders.	
Description	The logged-in admin should be able to access a page that allows them to see a specific order's details.

Acceptance criteria	- The details should include:
	  - Order ID.
	  - Customer name.
	  - Order creation date.
	  - Number of items in order.
	  - Items in order, each with its price and quantity and total.
	  - Total order cost.
	  - Order status.

The logged-in admin should be able to update any order's status.	
Description	The logged-in admin should be able to update any order's status to one of specific statuses.

Acceptance criteria	- The allowed statuses are:
	  - Processing.
	  - Shipped.
	  - Delivered.
	  - Complete.
	  - Canceled.
	- The admin should select the status from a list, they are not supposed to type it.
	- A success message should show after a successful update.
	- An error message should show after a failed update.
	- A confirmation message should show if the admin attempts to cancel an order.

The logged in admin should be able to log out of the website.	
Description	The logged-in admin should be able to log out of the website to end their session.
Acceptance criteria	- Session ends after logging out. Admin will not be able to carry out any actions other than logging-in again.

Stories for the Customer Persona

The customer should be able to see a list of products on the website.	
Description	The customer should be able to access a page that allows them to view products.
Acceptance criteria	- Products can be seen in a grid or a list. - Each product has its name, picture, and price displayed. - The page should be paginated, and should show 15 products per page. The products page can be viewed even by non logged-in customers.

The customer should be able to search for products by name.	
Description	The customer should be able to type a product's name into a search bar and click a button to view products that match the search term.
Acceptance criteria	- The products page should have a search bar. - Searched for products should show after a search operation. - The results should be paginated, and should show 15 products per page. - If no products match the search term, the page should say that there are no products that match the search. - All products that have names that match or contain the search term should be retrieved. - Search should be performed on the backend. - SQL Injection must be avoided.

The customer should be able to filter products.	
Description	The customer should be able to filter products by category, subcategory, product price, and product size (if any).
Acceptance criteria	- The products page should have filters for category, subcategory, product price, and product size. - Filtered products should show after any filtering operation. - The results should be paginated, and should show 15 products per page. - If no products match the filters, the page should say that there are no products that match the filters. - The category filter selects products that belong to the selected category. - The subcategory filter selects products that belong to the selected subcategory. - The price filter consists of two values: the min price and the max price. It selects products that have prices between the selected min and max filters. - If multiple filters are selected together, only products that match all filters should be retrieved. - All filtering should be performed on the backend. - SQL Injection must be avoided.

The customer should be able to view a specific product's details.	
Description	The customer should be able to click on a product to view all of its details.

Acceptance criteria	- The product's information should include:
	  - Name.
	  - Description.
	  - Price.
	  - Image.
	  - Available sizes (if any).
	  - Return policy (if any).

The customer should be able to register an account on the website.	
Description	The customer should be able to fill their information and register an account on the website.

Acceptance criteria	- The customer should be able to access a registration page and enter the following information to register an account:
	  - First name.
	  - Last name.
	  - Email (must be a valid and unique email).
	  - Password
	  - At least 8 characters
	  - Must contain at least 1 special symbol
	  - Must contain at least 1 number
	- Proper frontend and backend validation should be carried out on the email and the password.
	- A proper error message should show on the frontend if any validations fail.
	- If registration is successful, a session should be created for the user (they should be automatically logged in), and they should be redirected to the customer products list page.
	- The password should be stored encrypted in the database.
	- SQL Injection must be avoided.

The customer should be able to log into their account on the website.	
Description	The customer should be able to input their email and password log into the website.
Acceptance criteria	- A separate login page for customers should be created. - Proper frontend and backend validation should be carried out on the email and the password. - A proper error message should show on the frontend if the login credentials were wrong. - If login is successful, a session should be created for the user, and they should be redirected to the customer products list page. - Logged-in customers should not be able to access admin pages or perform admin actions. - SQL Injection must be avoided.

The logged-in customer should be able to add a product to their shopping cart.	
Description	The customer should be able to add products to their shopping cart.
Acceptance criteria	- Each product on the product list page should have a button that says “Add to cart”. - The product details page should also show a button that says “Add to cart”. - Clicking the “Add to cart” button as a non logged-in user should redirect the user to the login page. - Clicking the “Add to cart” button as a logged-in user should add the product to the customer’s cart. If this product already exists in the cart, the product quantity in the cart should be incremented by 1.

The logged-in customer should be able to view the products added to their cart.	
Description	The customer should be able to view the products in their shopping cart.
Acceptance criteria	- Customers should be able to access a page that shows the items in their cart. - The page should display a list of the products in the cart, showing the product's name, picture, product size (if any), the quantity of this product in the cart, and the total price of this product. - The page should show the total price of all products in the cart. - If the cart is empty, the page should show a text that says that the cart is empty, along with a button that says "Add products now" that redirects the user to the customer products list page.

The logged-in customer should be able to remove a product that was added to their shopping cart.	
Description	The customer should be able to remove products from their shopping cart.
Acceptance criteria	- Each product in the cart should show a button that says "remove from cart". - Clicking the "remove from cart" button should remove this product from the cart if the product has quantity 1. If the product has quantity more than 1, clicking the button should decrement the product's quantity by 1.

The logged-in customer should be able to complete their order by entering their shipping information and confirming their order.	
Description	The customer should be able to click a “Checkout” button and enter their shipping address details then confirm and place their order.

Acceptance criteria	- If the customer’s cart has products in it, it should show a “Checkout” button.
	- Clicking the checkout button should ask the user to enter their shipping details:
	  - Address line 1 (required).
	  - Address line 2.
	  - City (required).
	  - State.
	  - Country (selected from a list - required).
	  - Postal Code (required).
	- Proper frontend and backend validation should be carried out on the required fields.
	- A proper error message should show on the frontend if any validation fails.
	- Once the customer fills their address, they should be able to click a button to confirm and place their order.
	- The customer should also be able to select their payment method. The store should support one payment method only called “Cash on Delivery”.
	- Placing an order should create a new order that belongs to the customer, and contains the products they ordered.
	- The status of all new orders should be “Processing”.
	- A success message should show after the successful placement of an order.
	- SQL Injection must be avoided.

The logged-in customer should be able to view a list of their orders and their statuses.	
Description	The customer should be able to access a page that shows a list of their orders and their statuses.
Acceptance criteria	- Orders can be seen in a list. - For each order in the list, the page should show the order number, its date of creation, its current status, and its total cost. - The list should be paginated, and should show 15 orders per page. - The list should contain the orders of the logged-in customer only. No orders of other customers should be displayed in it.

The logged-in customer should be able to view the details of their orders.	
Description	The logged-in customer should be able to access a page that allows them to see the details of a specific order they created.

Acceptance criteria	- The details should include:
	  - Order ID.
	  - Order creation date.
	  - Number of items in order.
	  - Items in order, each with its price and quantity and total.
	  - Total order cost.
	  - Order status.
	- The logged-in customer can only view the details of their own orders. They are unable to view orders of other customers.

The logged-in customer should be able to log out of the website.	
Description	The logged-in customer should be able to log out of the website to end their session.
Acceptance criteria	- Session ends after logging out. The customer will not be able to carry out any actions other than logging-in again. - The user should be redirected to the customer products list page after logging out.

Submitting Your Work

1) When you are done with your work, you need to upload it to a **private** repository on Github.

2) Add the Teaching Assistant of your cohort to the repository as a collaborator with read-only access. Your teacher assistant will provide you with their Github username so you could add them to your repository. If you do not add them, they will not be able to see your code and grade your work.

3) In addition to putting your code on a repository on Github, you need to deploy your website to the Internet so that other people can open it and use it.

4) Once you are done with the first 3 points, head to the Capstone Project on Edraak's platform and start the test that is called: "تسليم مشروع التخرج". Inside the textarea, you have to put 2 links:
    a) The link to your Github repository from point 1.
    b) The link to your website from point 3.

Recommendations

1. The first step you should do is read this document slowly and carefully, several times.

2. Before writing any line of code it is recommended that you:
    - View and browse online websites that are similar to what is required in this project. This will help you get a better idea of what the requirements are.
    - Draw some wireframes to publicize your ideas about how the screens in your website may look like and what the overall picture of the website will be. Do not tire yourself with drawing tools, you can draw frames using a pen and a paper if that is easier for you.
    - Prepare a checklist of the items you need to do. Keep that checklist up to date as soon as you finish and test any part you build.

3. Do not leave big chunks of changes untested. Test the changes you make as soon as possible to make sure you identify bugs early on.

4. If you feel that there are new terms that you cannot understand, the Internet is your dearest friend. Believe it or not, Software Engineers spend a good deal of time searching the internet for tutorials and guides to help them do their job in the best possible way.

Bonus Requirements

These requirements are not mandatory, but you can gain bonus points for implementing them after completing the mandatory requirements.

Let your imagination run wild and write your own stories for these requirements.

Bonus Admin Requirements

- Admins can see a report that shows multiple statistics. (E.g., “most popular product” or “total sales in a month” are some of many statistics that you may think of).
- Admins can see a list of users and the number of orders each user has made.
- Admins can ban user accounts. Banned user accounts cannot use the website, and will just get a message that says they were banned when they login.

Bonus Customer Requirements

- Customers are able to reset their password and receive a password reset email if they forgot their password.
- Customers are able to save their shipping addresses and reuse them in future orders.
- Customers are able to pay online instead of having to pay on delivery.
- Customers are able to report inappropriate products.
- Customers can see a list of recommended products based on the items they bought before.

The End