

Capstone Project 1:

E-Commerce App

Hey!

Welcome to the first Capstone Project for this training. Hope you had a great time in the Front-End Web Development training and are ready to apply what you've learned so far in creating real-world applications.

This project is meant to test your Front-end web development (HTML, CSS, Bootstrap, and JavaScript) skills. You'll be applying what you've learned so far in this training to create an e-commerce application.

In this Project, you'll be required to create the front end (User Interface, UI) and the back end of a full-blown E-commerce application. We will work on the front-end part right now, and you'll work on the back-end for the same at the end of the next training.

For now, please work on the front end and submit it for review, and you can submit the application again when you work on its back-end functionality.

Part 1 – Front end Development (HTML, CSS, Bootstrap 5, JavaScript)

Problem Statement

Create the UI design of the e-commerce website, which the users will be using to browse and purchase the products. This is the first part of the product. Since we are only working on the front-end part of this project, you don't need to create the UI dashboard for the admin side as of now.

Build an E-commerce website UI with the following features/pages:

1. **Header:** A header with 2 parts – the first part contains the logo, search box, and the cart and login buttons. The second part is the navigation bar that contains the links to the products and the categories.
2. **Footer:** A footer that again lists the categories and links, with a copyright message at the end. The header and footer should be common for all the pages.
3. **Home Page:** A home page that contains a cover image with the company's slogan right below the header and a slider that lists the featured products (right above the footer).
4. **Product Pages:** The following product pages:
 - a. All products page that lists all the products in a grid format.
 - b. Women's category page with the following sub-categories
 - i. All products
 - ii. Dresses
 - iii. Pants
 - iv. Skirts
 - c. Men's category page with the following sub-categories:
 - i. All products
 - ii. Shirts
 - iii. Pants
 - iv. Hoodies
 - d. Kids category page that lists all the kid clothes.
5. **Contact Page:** A decent-looking contact us page that asks for the user's name, e-mail, and message.
6. **Login Page:** A login page that asks for the user's E-mail and password
7. **Cart Page:** A cart page that lists all the products which we have added to our cart, with a "+ and –" button to increase and decrease the product quantity and a delete button to

delete the product from the cart. You'll make these buttons and features work later. For now, just create the design.

Specifications

1. Header

The header needs to be done in 2 parts. The first part of the header will contain a logo to start with. Use any logo you want. You can create a quick one on [canva.com](https://www.canva.com) if you'd like, design one from scratch using Photoshop, or just download a free logo from a website like [pixabay.com](https://www.pixabay.com).

Get creative, but make sure your logo's color and theme match the overall theme of the website. The logo goes to the far left of the page, and in the middle, place a search box with a search button. You can design it any way you want, but again, maintain a color theme.

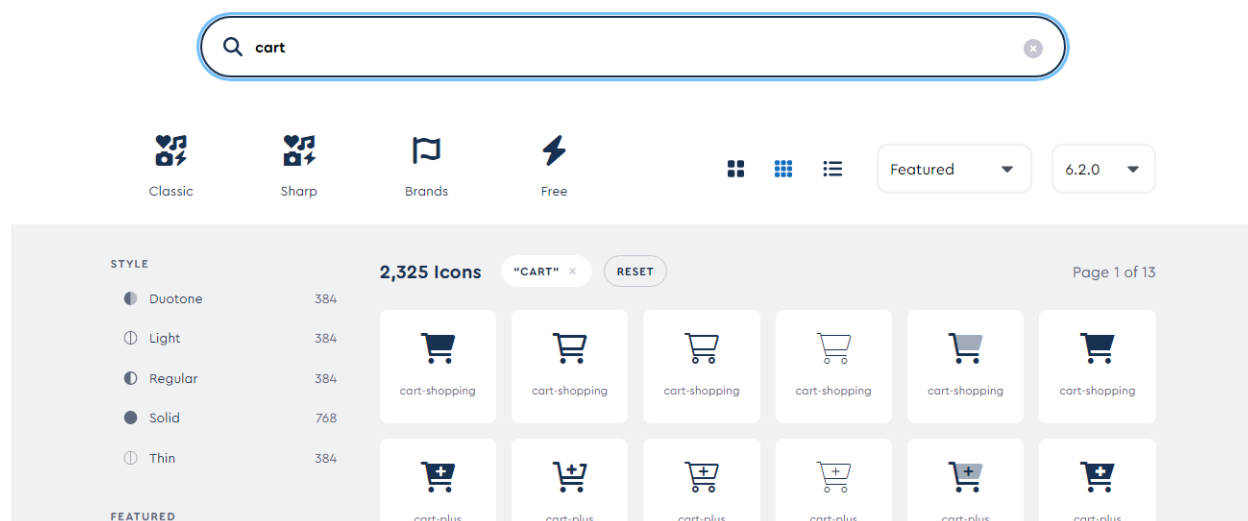
Finally, at the very end of the header, place your login and cart buttons. The login button can be a normal, but customized bootstrap button, but make the cart a font-awesome icon, and customize it further to increase its size (font size) and color.

Let me explain how to use font-awesome in your projects.

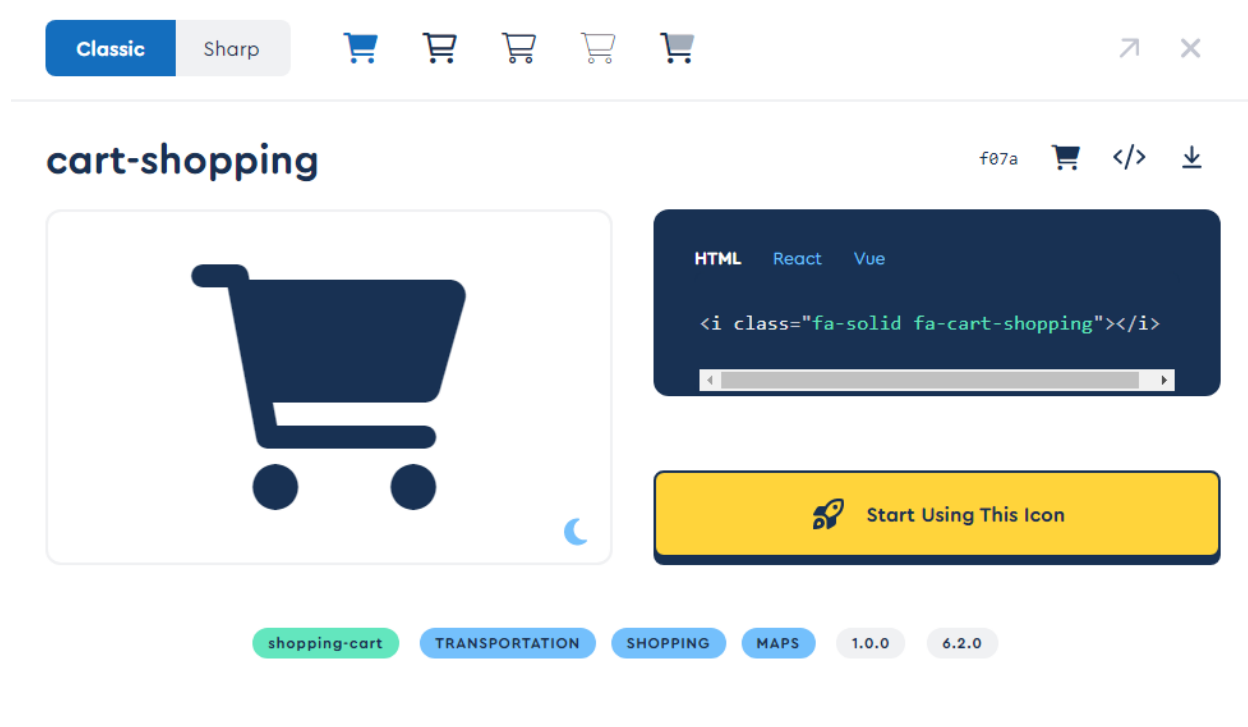
Include the following line of code in your HTML file, right above the link to your CSS stylesheets.

```
<!-- Font Awesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Then, you can get the icon you want from fontawesome.com. Go to their website and press on the search icon. In the search box, type 'cart', and look at the results, like this:



I'm choosing the 'cart-shopping' icon. You can choose anyone you want. Click on the icon you've chosen, and you'll see a box pop up, like this:

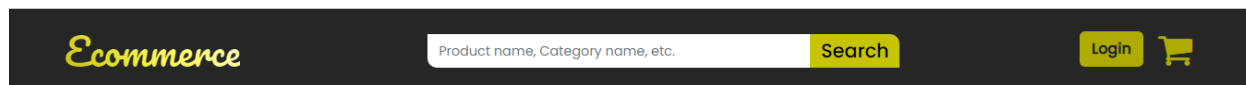


Copy the HTML code and use it in your code. Generally, font icons are used in place of button texts or link texts, like in my example right here:

```
<li class="nav-item">
  <a class="nav-link" href="/pages/cart.html">
    <i class="cart fa fa-shopping-cart"></i>
  </a>
</li>
```

Make sure to give an eye-catching 'hover' theme for the 3 buttons as well.

The first part of your header should look almost like this:



Now, for the second part of your header, create a very simple navigation bar (look up the nav bar tutorial in Bootstrap's documentation to get inspiration).

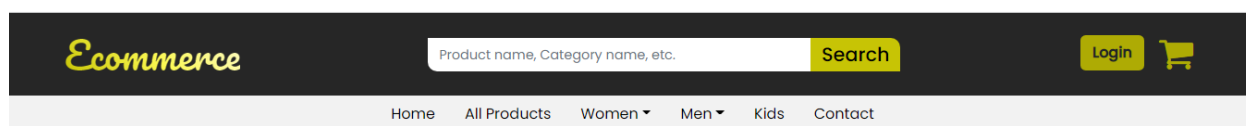
Below is the list of the items which you need to link to their respective pages:

1. Home
2. All Products
3. Women
 - a. All products
 - b. Dresses
 - c. Pants
 - d. Skirts
4. Men
 - a. All products
 - b. Shirts
 - c. Pants
 - d. Hoodies
5. Kids
6. Contact

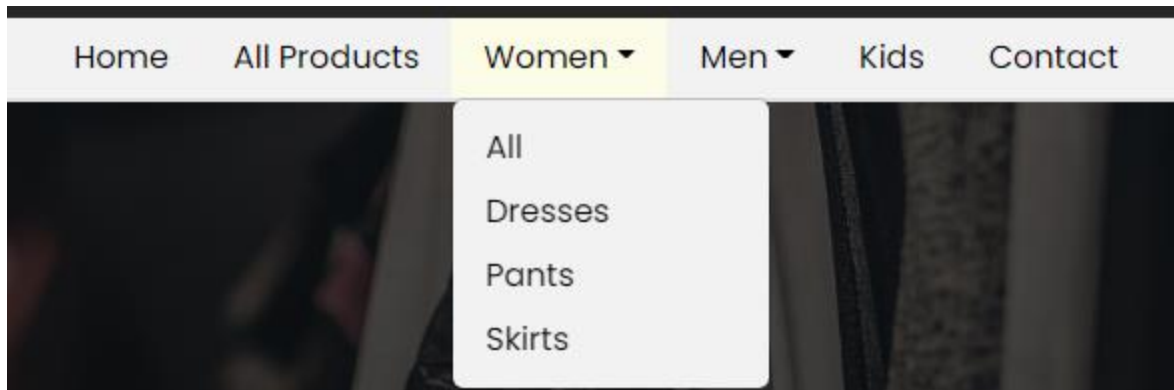
Link these to their relevant pages (create them beforehand). Design the navigation bar with a complimentary background color (to your header) and make the sub-categories a part of a dropdown menu.

Make the dropdown menu clickable on mobile devices, but they should drop down on hovering over them on large devices.

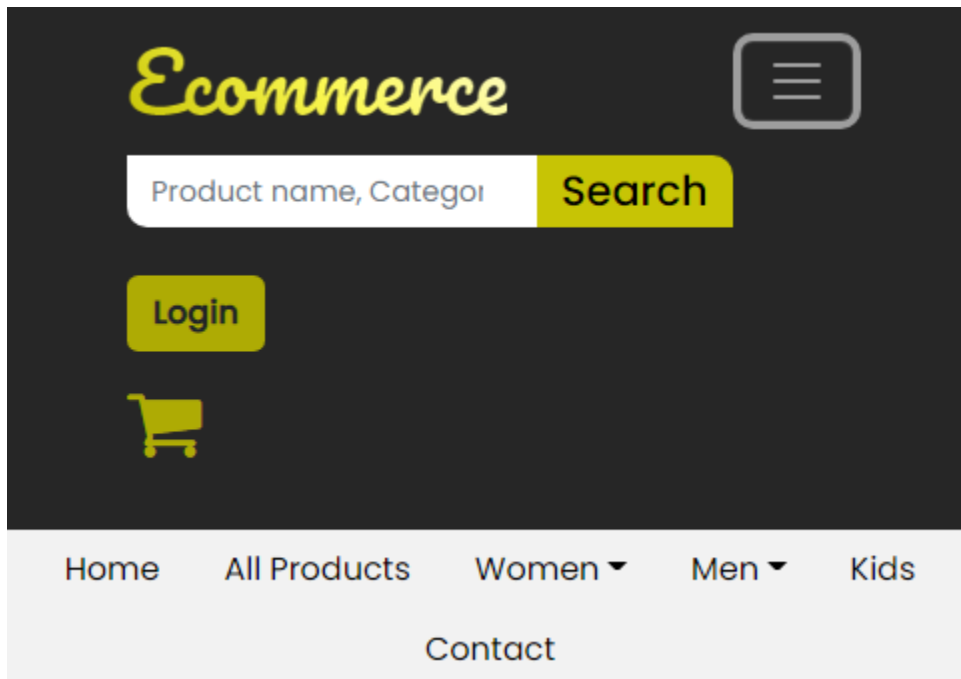
The final header should look something like this, please check the below image:



The dropdowns should look like this:



Make the entire header responsive. On small devices, they'll look like this:

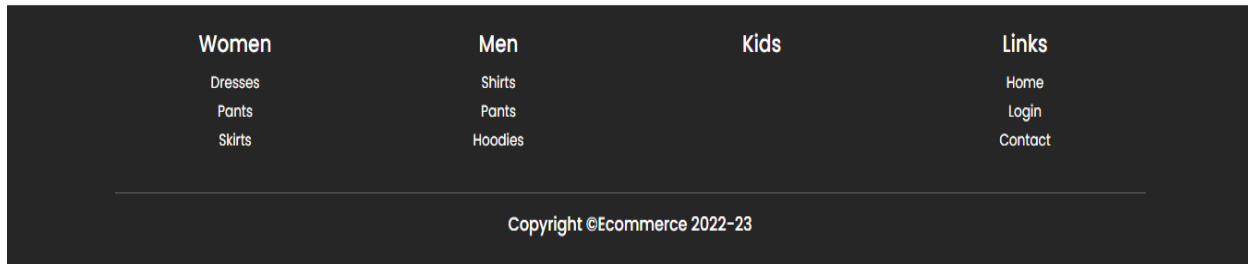


2. Footer

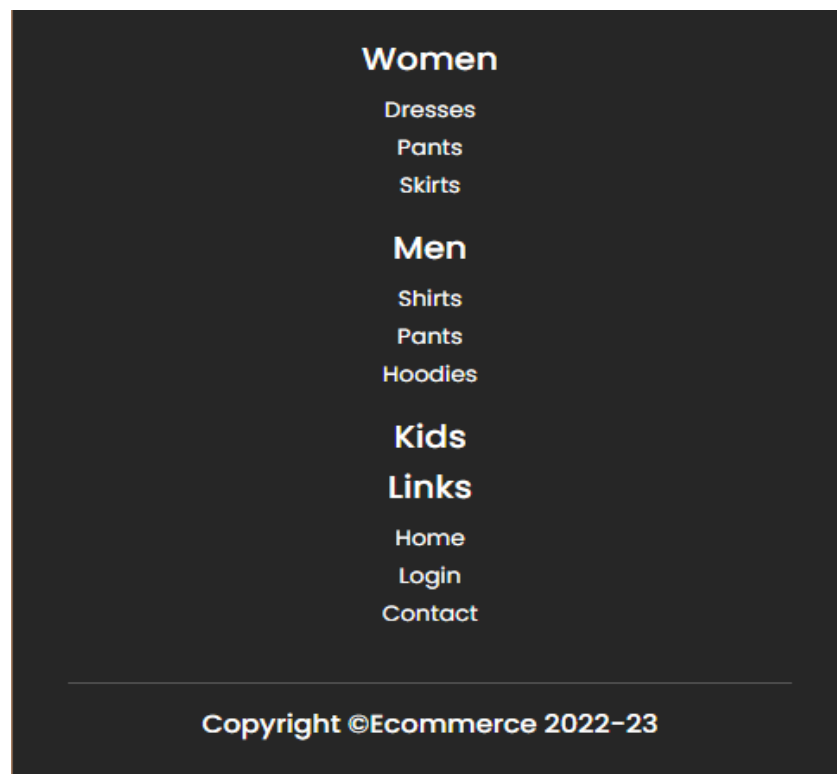
The footer can be quite simple. Create a simple row grid that lists all the categories in equally spaced columns, and then, in the final column, list the links. So, you should create 4 columns: 1 for Women, 1 for Men, 1 for Kids, and the final one for Links, which list the Home, Contact, and Login page links.

Create a divider between these links and the copyright message that should be at the very bottom of the page.

The footer should look something like this, please check the below image:



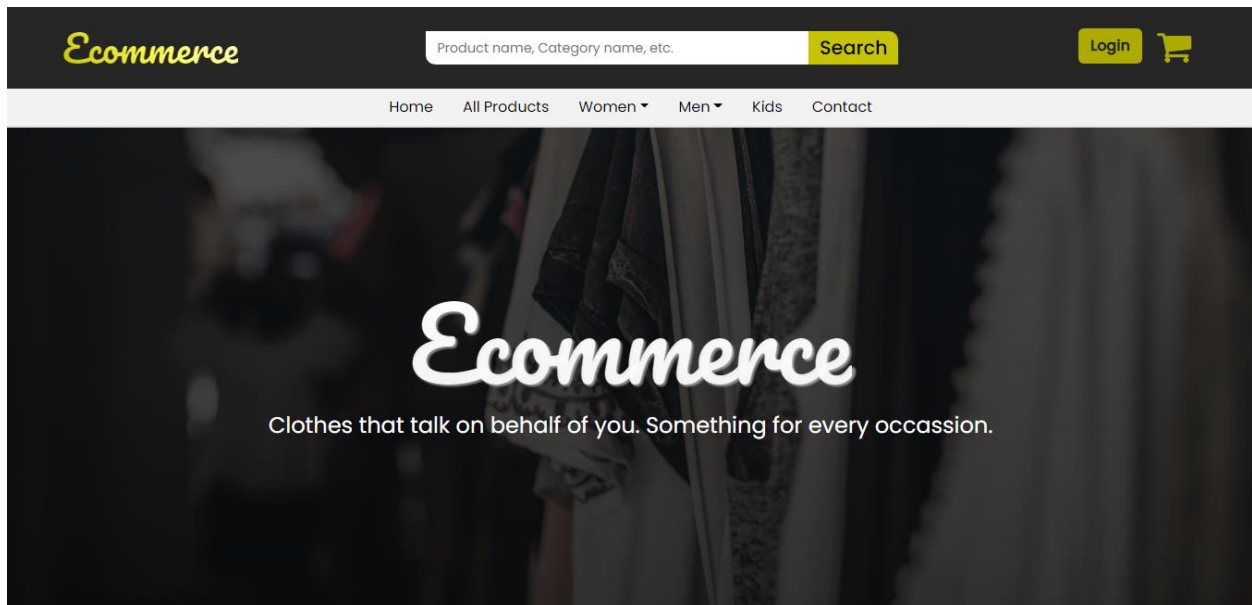
The footer should also be responsive, and the responsive footer design should look something like this, please check the below image:



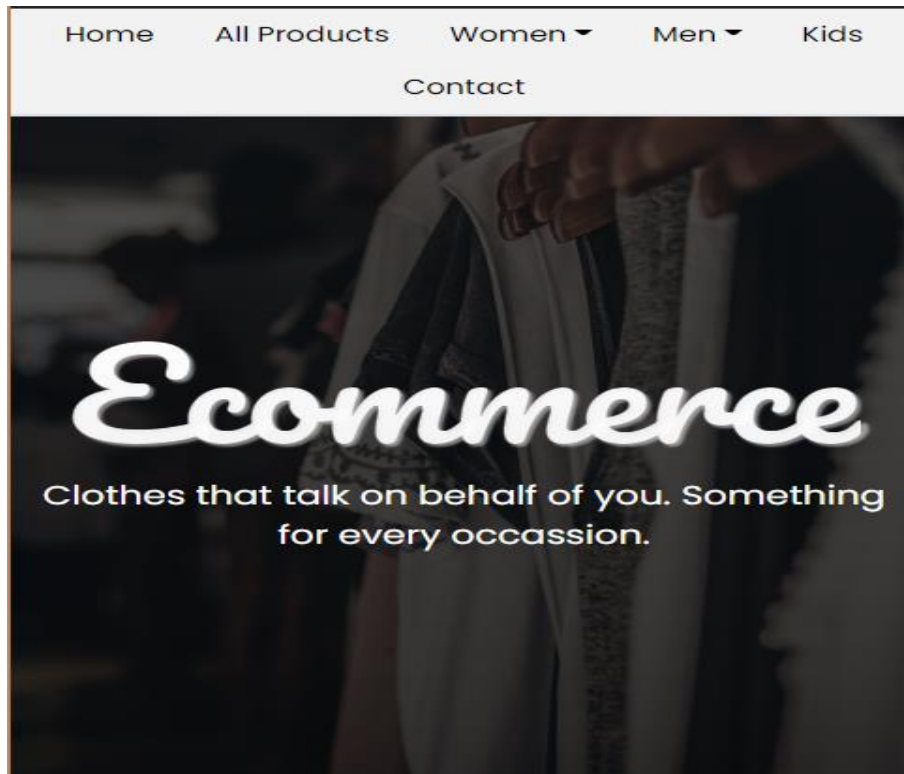
Note: Make the header and footer dynamic. Make sure you dynamically inject the header and footer code into every page (using JavaScript) to save time and avoid the repetition of code.

3. Home page

The home page has 2 parts, apart from the header and footer. The first is the cover image, with the website's logo and slogan written on it. Make the image cover the entire width of the page, and make sure the logo and slogan are properly aligned, designed (shadows), and spaced, they should look something like this, please check the below image:

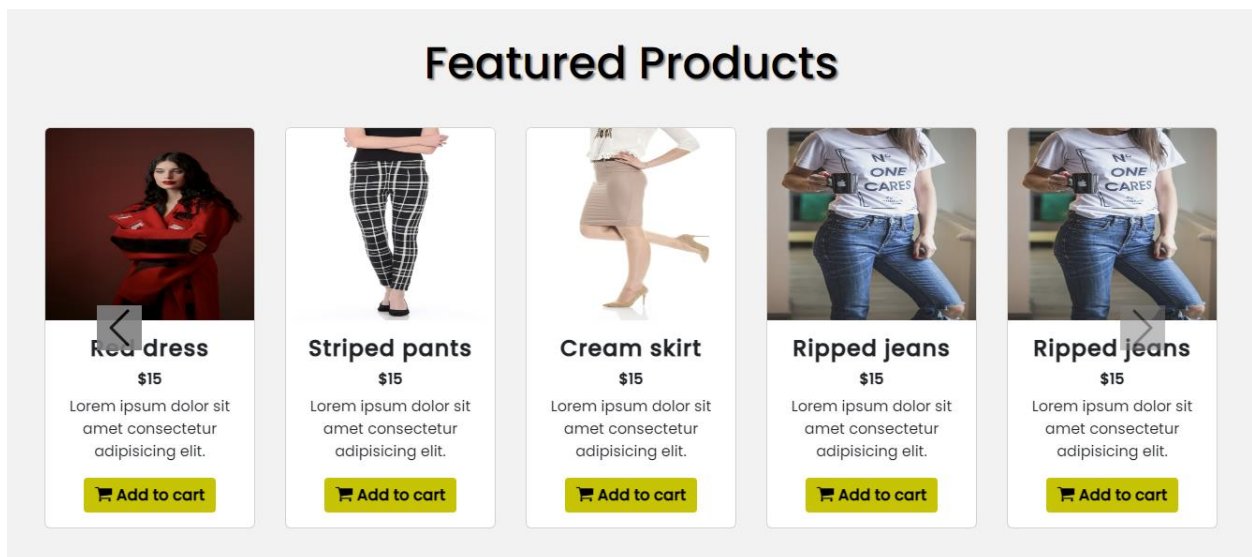


On mobile devices, it should look like this:



Next comes the featured products section. You can create a slider for this. Look at the Carousel section in the bootstrap official documents to get inspiration on how to create a 3-slide Carousel. Inside it, create 4 cards each for every slide, and design the cards appropriately, in accordance with the website's color theme.

The final slider should look something like this:



The slider must be responsive though, and in mid-sized devices, it should look like this:

Featured Products



Princess dress

\$15

Lorem ipsum dolor sit
amet consectetur
adipiscing elit.

 Add to cart



Green frock

\$15

Lorem ipsum dolor sit
amet consectetur
adipiscing elit.

 Add to cart

In small-sized devices, it should look like this:


Featured Products



Red dress

\$15

Lorem ipsum dolor sit amet
consectetur adipiscing elit.

 Add to cart

Hint: Just remove 2 of the cards from every slide for mid-sized devices, and 3 of the cards from every slide (leaving only 1 per slide) for small-sized devices. This would make the slides fit in smaller devices.

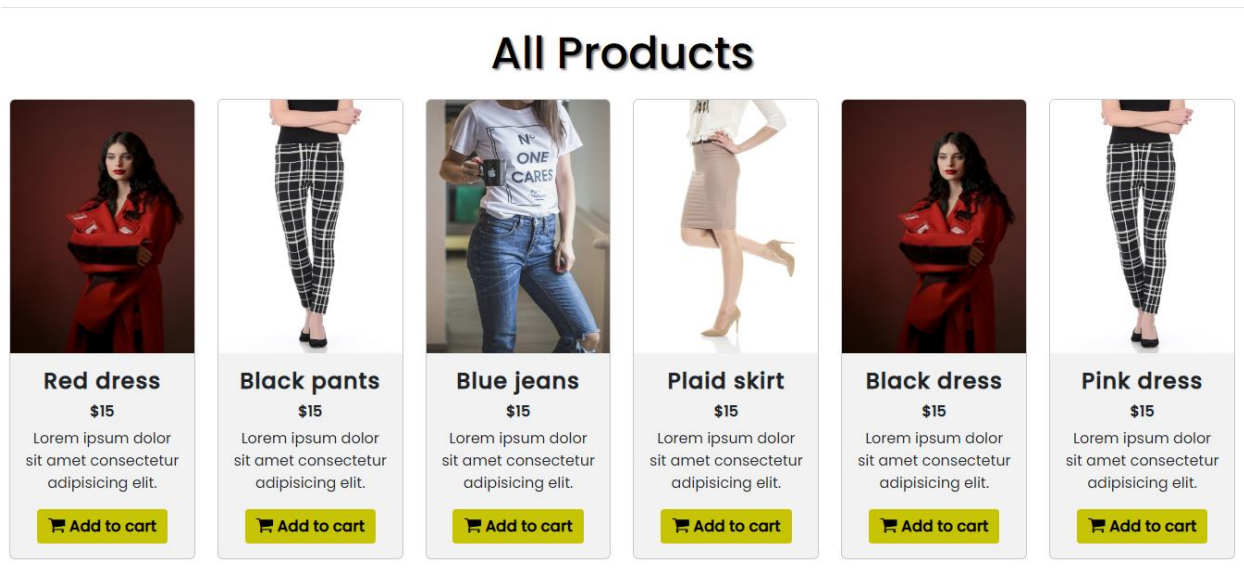
Alternatively, you can just create new slides for mid and small-sized devices, and just place 2 (and 1) cards respectively for every slide and increase the number of slides in the Carousel. That'll ensure that the user gets to browse through all the featured products, regardless of the device size.

4. Product pages

Create grids for all the product pages. Later, when you get into React and Node, you'll get the products' details from the database and dynamically create the rows and columns in the product grid.


For now, do it manually, to see how they'll finally look. You can create a grid for one of the pages (All products) and replicate them for the rest of the pages (from categories like 'Women' to sub-categories like 'skirts').

The final output should look like this on large devices:




On medium devices, it should look like this:

All Products




Red dress
\$15
Lorem ipsum dolor sit amet consectetur adipiscing elit.

[Add to cart](#)



Black pants
\$15
Lorem ipsum dolor sit amet consectetur adipiscing elit.

[Add to cart](#)




Blue jeans
\$15
Lorem ipsum dolor sit amet consectetur adipiscing elit.

[Add to cart](#)


This is how it should look on small devices:

All Products



Red dress
\$15
Lorem ipsum dolor sit amet consectetur adipiscing elit.

[Add to cart](#)



Black pants
\$15
Lorem ipsum dolor sit amet consectetur adipiscing elit.


[Add to cart](#)

5. Contact page

Make the contact form simple. Create a single row with 2 columns. The first column contains an image that represents your business. The second is the form, with the relevant input boxes, designed appropriately with Bootstrap and CSS.

The final output should look like this:

Contact Us



Name:

E-mail:

Message:

Submit

Make sure the contact page is responsive (use container fluid or container class).

6. Login page

The login page should just be a form, with a heading on top of it. It should look like this:

Login

E-mail:

Password:

Login

Make sure the login page is responsive (use container class). Also, on the login page, after entering the email address and password, if the email address is admin@admin.com and the password is 123456, display an alert saying, "Login successful". Otherwise, display an alert saying, "Incorrect email or password". Use JavaScript for this. You can do this quickly with a string equivalent (===) check.

7. Cart page

The cart page should contain 2 parts, in a grid. The first part should list the items, and the second part should be a summary of the price, shipping, and total price, with a checkout button at the bottom.

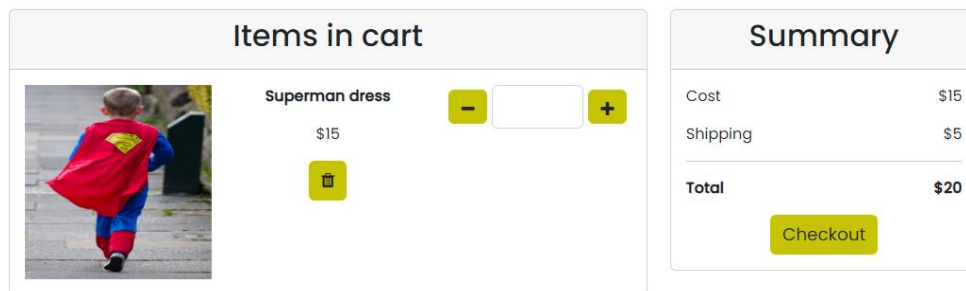
Inside the first grid, create another 3-column grid:

1. The first column should have the image.
2. The second column should list the product's name, and price, and should include a delete button (use font awesome's trash icon).
3. The third column should contain the form where the user can increase or decrease the quantity of the product, with relevant icons on either side of the form (+ and – icons).

The summary section should be very simple. It should contain a single card, with a

1. Card header (Summary)
2. Product price (total)
3. Shipping with its cost
4. A horizontal line that separates both parts
5. The total cost and finally,
6. The checkout buttons.

Finally, the cart should look something like this:



Make sure the cart is also responsive and designed beautifully for smaller devices.

Evaluation

The capstone project will be evaluated based on the rubric given below. It contains the weightage of marks allotted to each section of the project. Make sure you adhere to the rubric to score well on the project.

#	Evaluation	Score
1	Create all the given pages with relevant specifications as mentioned	30%
2	The design looks like the ones shown in the screenshots – you can try to make	25%

	it a bit more creative, but it should look like a professional website	
3	Make everything mobile responsive for both medium and small devices	25%
4	The header and footer should be dynamically generated (JavaScript)	10%
5	All the code files are neat, with structured folders, and comments added to segregate the sections	10%
Total Marks		100

Submission

- Add all the project files to a folder and compress that folder. You are required to submit this compressed(.zip) folder.
- Make sure you submit the correct files for the project. There will be no resubmission allowed if you submit the wrong files.
- In any case, if the instructor asks you to resubmit your project, you are required to resubmit the correct project within **5 days** of the resubmission request. After this, you will not be able to resubmit and your initial submission will be evaluated and considered as your final submission.