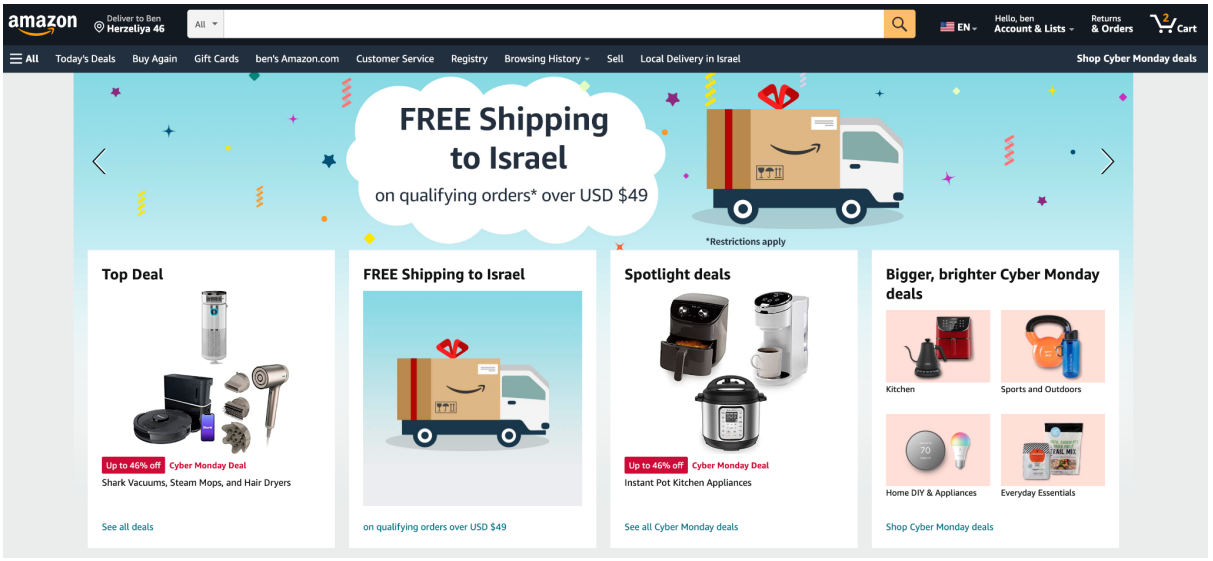
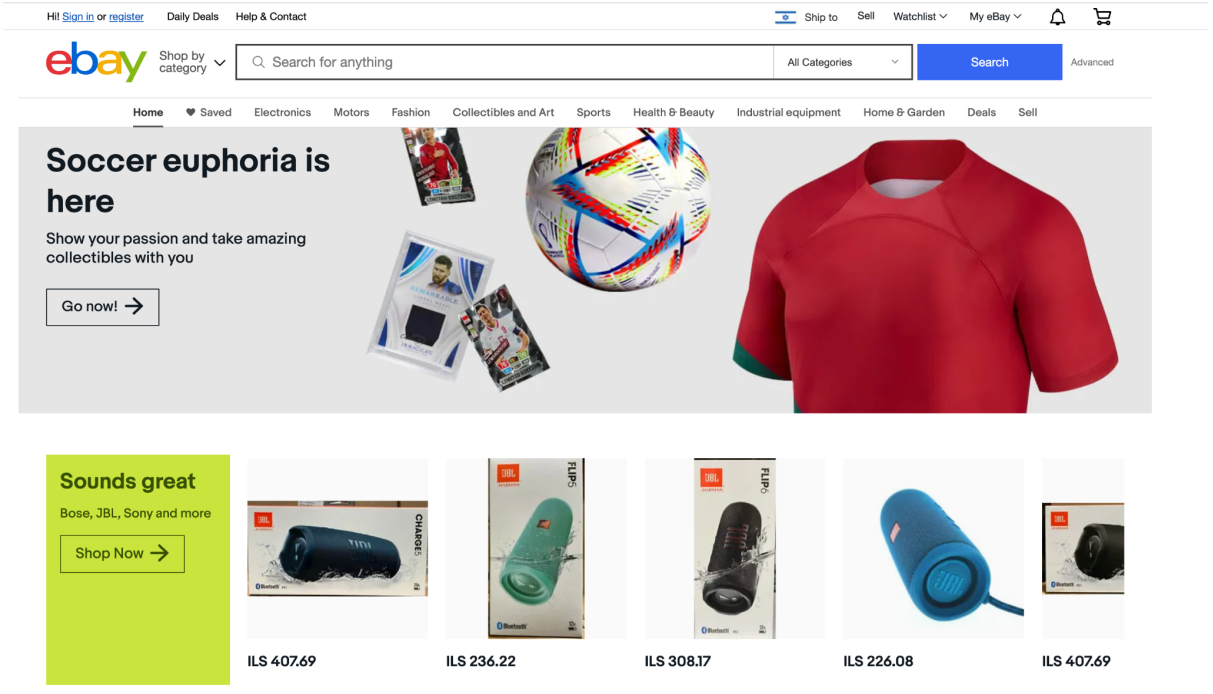
Final Project - Shopping Website

Instructions:

1. Your mission is to create a full end to end shopping website
   You can take examples from familiar websites such as Amazon, Ebay, Shufersal, Rami Levi, etc…





2. Your website should implement the following:

    a. **Main page** → The main page is the first page your users will see.
    This page should have the following ui components:
        - Main navbar → logo, order list button, favorite items button, login / sign up button, logout button
        - Main heading → Your shopping website name and picture
        - Searching area
        - Available items grid

    b. **Show available items list** → Each item should be shown in the main screen and should have:
        - Title of the item
        - Photo of the item
        - Price in USD
        - How much items available in stock

        You should have at least 10 different items in your website

    c. **Show search bar** → The search bar should send an API call to search specific items by name, the search result should show all the items that have the requested name in their name.
    For example: searching for the word sun, can return sunglasses
    If your search didn't find any item you should notify the user about it.

    d. **Login process** → Each user should login to your website by simple authentication process. If a user doesn't have an account yet, he should be able to create one with a simple create account form.
    You should save the following data on each user account:
        - User ID
        - User first name
        - User last name
        - User email
        - User phone
        - User address (country & city)

- User name
- User Password

User should have the ability to delete himself from the website, if a user deleted his account you should also delete anything that was associated with that user.

e. **Favorite list** → Each user could create a favorite list page, the user can add any item he wants to this list and once he goes to that list he could see all the favorite items he added.
The user can add items and remove items to this list. Each item should appear in this list only once.
The user should see the favorite items the same way they appear in the main page.
Favorite list should be saved so even when the user logged out he should see his favorite list again when he logged back in.
Only when a user is logged in he should have the ability to access to favorite items list and add / remove items from this list.

f. **Stock management** → For each item in your website your system should handle the stock. Each item should have available items in stock. If a user orders an item this number should decrease.
Users should not be able to order more items than what you have in stock.
If an item is not available in the stock anymore it should appear in the main page and in the favorite list page with “0 items left in stock” message.

g. **Order list** → Your system should have the ability to handle orders.
For each order you should save:
    - Order id
    - Order user id (The user that created the order)
    - Order date (The date the temp order was create)
    - Order shipping address
    - Order total price

- Order status (TEMP, CLOSE)

Your system should show for each login user his order list. This list should show all the orders that this user created and also temp order if this user has one.
If a user clicks on the order with the temp status he
should see the order process page. In that page he could add / remove items from his temp order.
If a user clicks on order with a close status he should just see a page with the order details (list of the items in the order, price, address ext..) but he could not make any changes on that page.

h. **Order process page** → Each login user can have the ability to add any item to his pending order. Once a user adds the first item to the order he is automatically creating a new order with pending status and he should see this order in the order list page.
The pending order should always appear first in the order list and should be styled differently from the closed orders.
Once a user clicks on the pending order in the order list he should see the order process page.
This page should show a list of all items that already were added to his order. The user should have the ability to add / remove items from the order.
The order page should show:
   - List of all items in the order (With title, photo, price)
   - Total price
   - User shipping address
   - Payment button

   Once a user press on the payment button the order should be saved again with CLOSE status and should be added to the close orders in the user order list. In that case the user could create a new pending order again.
   The pending order should be saved even after the user logged out.

If the user removed all items from the list you should delete his pending order completely.
Each user should have only 1 order in status pending.

3. For your implementation you should use the following stack:
    - **For the Backend side** → Java & Spring Boot framework, H2 DB
      Notice that you are building this service as we learned in the course (MVC, Configuration, JDBC, Mappers, Enums, proper naming, ext …)
    - **For the Frontend side** → Javascript & React library
      Notice that you are building this service as we learned in the course (Components, States, Configuration, Routing, Material-UI, proper naming, ext…)

The entire project should be uploaded to github as two separate repositories for the Backend side and for the Frontend side.

Each repository should have a professional read me file that describe the project, the project logic and the technology stack.

Good Luck

ECOM SCHOOL

המכללה למקצועות הדיגיטל וההייטק

ECOM SCHOOL

המכללה למקצועות הדיגיטל וההייטק