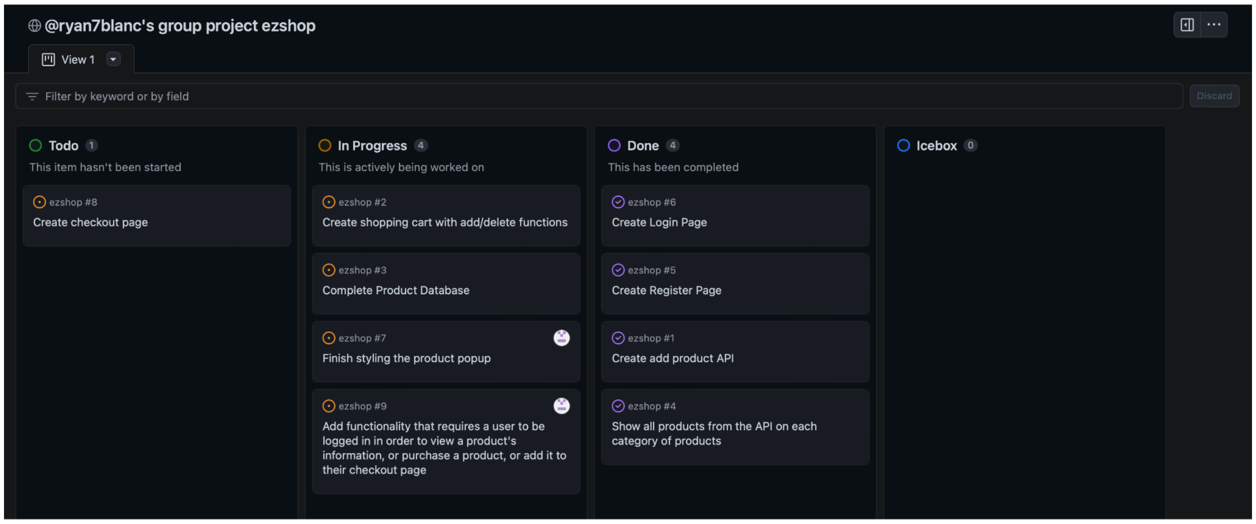
Title: EZShop

Who: Anna Soukhovei, Louis Blanc, Benji Burkhalter-Castro, Nathan Reed

Project description:
EZShop is an easy all in one shopping tool that provides users with the simplest way to find items they want and checkout with just a few clicks. With no ads, users are able to enjoy a clear and unobstructed view while shopping for their favorite products. Most of our project is done using EJS, with some JavaScript to handle the server functionality. We used git for source control, discord for communication and collaboration and Microsoft azure to host our application. The SQL database consisted of four tables, user, cart, product, and items. Each user and product has it's own individual ID to separate it from the other users and products. Using each ID, we created the cart table to store the user ID and product ID to keep track of what the user wants. At checkout, all the cart table entries will be displayed allowing the user to purchase or delete products. The item table came in when moving our front-end products to the database, as we have stored some products on the page that weren't stored in the back-end, until checkout. We also implemented a buy now option that selects an individual product allowing the user to purchase the product instantly. Once the user is ready to purchase, they can simply enter their credit card and address information. Once submitted, the user will be taken to a thank you page letting them know that their order is on the way.

Project Tracker: https://github.com/users/ryan7blanc/projects/3


@ryan7blanc's group project ezshop

View 1

Filter by keyword or by field

Discard

Todo 1
This item hasn't been started
- ezshop #8 — Create checkout page

In Progress 4
This is actively being worked on
- ezshop #2 — Create shopping cart with add/delete functions
- ezshop #3 — Complete Product Database
- ezshop #7 — Finish styling the product popup
- ezshop #9 — Add functionality that requires a user to be logged in in order to view a product's information, or purchase a product, or add it to their checkout page

Done 4
This has been completed
- ezshop #6 — Create Login Page
- ezshop #5 — Create Register Page
- ezshop #1 — Create add product API
- ezshop #4 — Show all products from the API on each category of products

Icebox 0

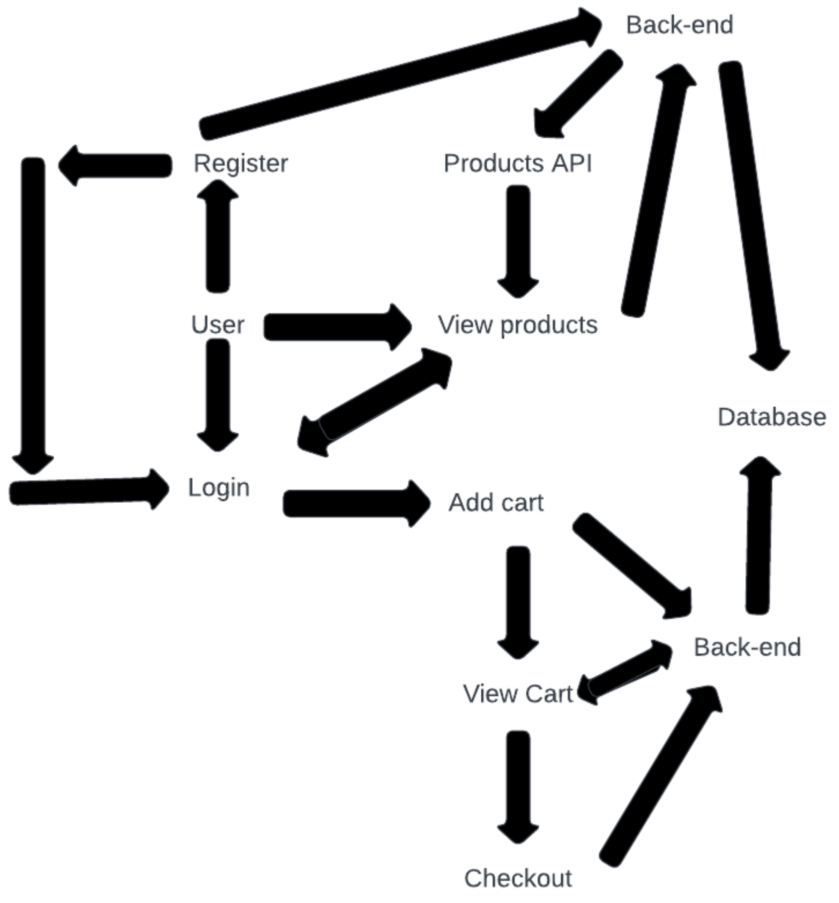
Video link:
https://drive.google.com/file/d/1RdDUl9zwk694Yb6AWOQWOLtBJ6sHNCPM/view?usp=sharing

VCS: https://github.com/ryan7blanc/ezshop.git

Contributions:

- Anna Soukhovei:
    - I worked mostly on the back-end of the project. I created the endpoints that extract data from Fake Products API to show products on the website, extracting data from the users, cart, items, and products table in SQL to show the items in the cart, and to count the number of items. I also added code to the front-end to show items from the user's cart and to show products from the API. I also implemented some error handling to make sure the user does not crash the website.
- Louis Blanc:
    - I cooperated on the creation of the SQL Database, Backend structure and templating. I planned out how we would make the table as well as how to structure the data sorting. Although we didn't use any of the cpp files at the end, this helped with how we wanted to execute our SQL table with our front-end data. I also made the Presentation and deployed the website on Azure.
- Benji Burkhalter-Castro:
    - I worked mostly on the front end of our website. Specifically within the checkout page. Initially we thought to use an Apple pay or Paypal API to handle our payments but because our website isn't a registered LLC it ended up not being possible. Alongside this I contributed to some the CSS elements and the project presentation and final report.
- Nathan Reed:
    - I worked on the partials and main look of the home page in CSS and ejs. I helped with creating the database for our product data and other features such as a completed purchase screen.

Use Case Diagram:



Test Cases: How 4 Features were tested:

Login:
User may enter their login credentials (username, password), and if username and password are found in the users table in the DB, the user may login.
If username or password are incorrect, the user is alerted with an error message and what action they can take. The test data that will be used to test the feature will be of 2 types: valid data and invalid data. Valid usernames will be found in the users table, not null and will be below 50 characters. Valid passwords will have a corresponding hash

value in the users table, are not null, have a minimum of 8 characters, and are below 60 characters long. The test environment will be the website on a server accessible to all user acceptance testers, and it will be focused on the login, register and home pages.
If a user successfully logs in with credentials found in the users table, that will be a positive test result. If users are unable to login with proper credentials, that will be a negative result. User acceptance testers can make new accounts using the register functionality, and then test if they can login successfully into their account.

Register:

A user may sign up for the website but they must make a username and password (minimum 8 characters). This information will be stored in the users table.
If a user tries to sign up with an invalid password (less than 8 characters or null), then they will be redirected to the registration page with an alert
that gives them feedback on what happened.
User acceptance testers will enter data that is either: valid, or invalid. Valid data will include non-null usernames under 50 characters
not found in the table, and non-null passwords that are between 9 to 59 characters long.
Invalid data will include null usernames and passwords,
usernames and passwords that violate size requirements, and usernames that are already found in the users table. The testing environment will be on a server hosting the website for multiple user acceptance testers to be working on simultaneously, so we can see how the website would function in real time and what bugs arise. The environment will take place on the register, home, and login pages. Positive test results will be that the user can successfully register an account with a valid username and password. A negative result would be that users will fail to sign up for the website, or that they will be able to sign up for the website with invalid credentials. User acceptance users will not be given anything except for access to the website and the ability to make various accounts on the site.

Adding product to cart:

A user may add 1 or more items to their cart and they may remove any added items. Products in the cart should appear across different sessions for the same user. In more depth, whenever a user clicks on the "add to cart" button on a given product page, that item should be added to that user's cart in the items table. In checkout, if a user clicks on "delete from cart" on a given product in their cart, then that button will call the delete endpoint. The test data will include the various types of products found in the products tables in our database. The test environment will be the website on a server accessible to all testers, and will be focused on the home page, product pages, cart page, and the checkout page.

Positive test results will involve products being properly added and removed from the cart, and every user has their own specific cart separate from other users. Negative test results will be the cart function breaking (not adding items properly, not removing items, invalid number of items like negative or null values). User acceptance testers will be each individually logged into an account found in the users table in the SQL database with an empty cart.

Viewing each category of products:

When a user clicks on a given category of product (Jewelry, Electronics, Mens clothing, Womens clothing), they will be met with a page that displays all available products on that page. Users can view product pages regardless of if they are logged in or out. However, unregistered users cannot interact with the cart features (adding to cart, clicking on the cart, or buying the product). If the API call to the Fake Products API fails, then an error page will display.

Results & Observations

Register: Users typically ignore the register button and opt to go straight into looking at and buying products. When an unregistered user goes to buy a product, they will be redirected to the registration page (successful in every case tested). In most cases, users did not know about the 8-character password requirement and typically went below it, and were not allowed to create an account. To solve this, we added a message that tells the user to make a password that is at least 8 characters long whenever they enter an invalid password.

Login: Users typically only use the login page right after registering, or when they load up the website as a pre-registered user. User acceptance testers were able to successfully login in all cases and access the website with their cart being saved across different sessions. Their behavior was consistent with the use case, so the login functionality served them well.

Adding product to cart:
Users enjoyed playing with the add to cart feature, which allowed them to add up to 10 products at a time. The “buy now” and “add to cart” features seemed intuitive for users, as that was what they were immediately trying to do upon entering the website. So, we had to include a safeguard that did not allow unregistered users to add items to their cart (because only registered users have a cart). In testing, the add to cart feature worked well. A user tried to store 101 items in their cart and there were no issues that stemmed from that (and this is assuming that a user will not try to buy more than 101 items, as that seems unrealistic for the intended consumer base). Users did deviate

from the use case, as they tried to add products to their cart without signing in, but after adding safeguards to the website, they could not deviate from the use case.

Viewing Products:
Thankfully in testing, the API that we sourced our products from never went down, so products were shown successfully and added successfully to the database. All information relevant to the product (price, image, name) was shown correctly. Viewing products was the most popular use of the website amongst user acceptance testers. This makes sense, as users are looking to use the website to fulfill their shopping needs. Users could not really deviate from the use case, as they had no choice but to see products when they clicked on a category of products.

Deployment:

http://recitation-15-team-3.eastus.cloudapp.azure.com:3000/
This website is deployed using Azure. This was deployed by following the steps of lab 13. This can also be run by taking the repository link and following the steps of lab 13 to run it in your own container.