Bobcat C.L.A.W.S.

Prototype Report

Group Members: Abigail De Rousselle, Dillon Hughes, Dustin Bruce, Garrett Dipalma, Jawad Abu-Gabal, Jess Stevenson, Kevin Garcia, Rayyan Khan, & Rebekah Hardsand

Contents:

1. Intro
2. Webpage - FrontEnd
    2.1. Completed Issues
    2.2. In Progress Issues
    2.3. Failures – Resoultions
    2.4. Out-Of-Scope
    2.5. QC notes
3. Database – BackEnd
    3.1. Completed Issues
    3.2. In Progress Issues
    3.3. Failures – Resoultions
    3.4. Out-Of-Scope
    3.5. QC notes
4. Server
    4.1. Base
        i. Completed Issues
        ii. In Progress Issues
        iii. Out-Of-Scope
        iv. QC notes
    4.2. Data Collection
        i. Parsehub
            1. Completed Issues
            2. In Progress Issues
            3. Failures – Resoultions
            4. Out-Of-Scope
            5. QC notes
        ii. Bluecart & RedCircle APIs
            1. Completed Issues
            2. In Progress Issues
        iii. CLAWS Scraper
            1. Completed Issues

2. In Progress Issues
3. Failures – Resoultions
4. Out-Of-Scope

iv. Coupon

1. Completed Issues
2. In Progress Issues
3. Failures – Resoultions
4. Out-Of-Scope

5. Updated Schedule
6. Supporting Images

1. Intro

We have been working on our C.L.A.W.S. project for the past 3 months, in this report we will be introducing our working prototype and discussing what features we will be continuing work on and adjusting our expectations for the final project that we will be presenting on Nov. 30th. While we belive that our project is in line with our original vision, we have had to adjust some of our scope to acount for how and what data we are able to gather. We will also discuss what may need to be adjusted if this project is to continued to be developed. Finally we have included some quality control notes about our testing and maitenence procedures.

WHAT IS THAT?

2.Webpage

2.1 Completed Issues

Several tasks have been accomplished for this prototype. Firstly, the implementation of the Bobcat CLAWS button in the header now allows users to navigate back to the homepage. Additionally, a subcategory dropdown menu has been successfully integrated into the header. The dropdown menu enchances user interaction by mapping each subcategory name to a number, facilatating easy queries. Subroutines for each category have been implemented into the header component aswell. When a user selects a category the user is taken to a page which is loaded with products of that category. Users can click on any product and be taken to a dedicated products page which also has a similar products feauture.

2.2 In Progress Issues

Our team is currently focusing on several critical aspects of the project. The product comparison page is under development, with the goal to display one of each product for easy comparison by the users. We are also working on integrating a price filtering feature on this page. This feature will allow users to adjust a price slider, updating the product page to reflect products within the selected price parameters.

Another significant task in progress is the enhancement of the search bar functionality. We are developing subroutines that will allow the search bar to generate results based on keywords entered by the users. This feature will greatly improve the site's usability and user experience.

2.3 Failures – Resoultions

Price filtering: Price filtering displays products within the lower and upper bounds of user selected price, but when prices of produtcs are below min bound page is not sorted. Price filter is breaking the menu component currently so it is not implemented in prototype version.

Resolution:

Search Functionality: The keywords have just been inserted into the database. Front end has not had enough time to add the search functionality to the webpage yet. The search functionality will search through the keywords peramters for each of the products stored in our database. The database should return all products in an array to the save-product service which will be able to display to the products page.

Resolution: The keywords that are generated and stored in the database must be relevant. The front end must recieve an array with the corresponding products to from the keyword search.

2.4 Out-Of-Scope

Since the query service has just been set up, front end is still being developed and we have not yet experienced any out of scope tasks.

2.5 QC notes

The product comparison page is being created off of similar priced options. We are not comparing the same product accross different stores. The filtering is still being developed.

3. Database

The database stores the product data formatted so that the flow is easy to follow for the front-end and back-end processes. It is sorted by categories and subcategories. We have 3 categories and 16 subcategories. All data we currently have has been collected using Walmart and Target APIs and scrapping Best Buy.

3.1 Completed

During this development phase, we have finished readjusting our database design and implementation and have shifted focus to data insertion and data querying.

The database tables are:

- Category
- Sub_category
- Product
- Store
- Product_Store_Price
- Store_location
- Coupon
- Coupon_Product

- Coupon_Store

We readjusted the product, subcategory, and category tables to fit our data needs better. We limited the foreign keys in the product table to just the sub_category and category IDs and changed these IDs to integers. (See DB Flow in Supporting Images)

We have created a full script to insert new product data into the database or replace the current one if a product is already in the database. This script first checks stores, categories, and subcategories, inserts product data, and finally, the product store and price table.

We have a script to insert coupon data.

We have inserted and categorized actual initial data into the database for all categories and sub categories.

We have created a script to generate keywords for products in the database. These keywords will be used to implement a search functionality on the front end.

We have created stored procedures to get data based on sub category IDs, category IDs, or all available products. These procedures are used by the front end to query the database for different pages.

3.2 In progress

In progress, we currently have the development of additional stored procedures according to front-end needs, such as querying products by price ranges and searching by keywords. We also have started the testing of insertion scripts and stored procedures.

3.4 Out-of-Scope

We were unable to create a single entry for a product and map it to different stores because we did not get enough consistent product data. Since most of the products did not have a UPC available and products have different names and descriptions across different stores. So, instead of comparing products by store, we shifted to comparison by price ranges.

3.5 QC Notes

We are currently testing and maintaining insertion scripts and stored procedures. To maintain data integrity and quality, we are also checking the data types in the data being inserted and converting it to the necessary ones. We are also checking for duplicate products for the same store during insertion to avoid duplicate entries. There is also a script that creates database backups to ensure data recoverability. For post-semester plans or if time permits, there should be mechanisms for additional data validation, data cleaning, data completeness, and security.

4. Server

4.1. Completed

In our recent server development phase, we have achieved several milestones. We've established a secure and efficient web hosting environment on a server running Linux Mint. This includes the development of a webpage using Angular, Node.js, and npm. We have successfully installed and configured Nginx and pm2. In terms of security, we've implemented a reverse proxy for port 8080 and introduced a self-signed SSL certificate using OpenSSL. Additionally, we have acquired a valid SSL certificate from TXST CFA.

4.1.i In Progress

Currently, we are working on expanding the website's accessibility. The website is accessible only on the TXST Network, and there are plans to expand this accessibility through either self-hosting or a commercial hosting solution after the semester ends.

4.1.ii Out-Of-Scope

We have encountered a significant challenge in our development process. Port forwarding has proven to be infeasible due to TXST server constraints for students, which limits some aspects of our server's functionality. There are options to port-forward given we pass vulnerability scan and Firewall exception requests from ISO. These requests can only be made by faculty members.

4.1.iii QC Notes

We are currently working on refining our maintenance processes. This includes integrating essential scripts into the logFiles folder for automated server management. It is crucial to ensure that the security measures, such as the self-signed SSL certificate and reverse proxy, are thoroughly reviewed to align with best practices and address potential vulnerabilities. Also, regular monitoring of the performance of Nginx and pm2 is essential to ensure reliability and efficiency. And finally, the plans for expanding website accessibility post-semester should be monitored and reviewed to ensure a smooth transition.

4.2. Data Collection

4.2.i. Parsehub

4.2.i.1. Completed Issues

The Parsehub program is intended to be used to scrape the data from any eCommerce site that does not have its own product data API. The finished program now consists of two main components. First, a trained model called a 'project' that scrapes relevant product data from a given URL, to a particular store's search results page, then returns a JSON file of scraped data. So far, only one project has been created, it scrapes bestbuy.com's search results and product details pages. The second component is a script that automates the process of getting all URLs, running the corresponding projects on them and then saving a JSON file containing all scraped data from every run. The script uses an API key, project tokens, and other sensitive data hidden in a .env file to complete its tasks.

All the URLs are generated and validated by the script itself. The script reads in search terms from the searchTerm.txt file and concatenates it to a URL template that is specific to a store page and indicates that the products being scraped should be available to the San Marcos location.

New functions have been implemented in the script to catch and attempt to fix any errors and mistakes that were made while scraping data. These include a retry mechanism, checking all fields are being correctly scraped, filling any fields that were not scraped, putting the fields in the correct order, and removing duplicate products. There is also now a function that sends an error message to the developer's email if there is an error the script is unable to fix itself.

4.2.i.2. In Progress Issues

Tests using the pytest library are still being written for the Parsehub script. Tests have been made for all the functions except run_proj and main.

4.2.i.3. Failures – Resolutions

It was discovered that Parsehub is unable to scrape hidden elements on a webpage, and so the project for bestbuy.com was unable to scrape the UPC and Description data. The original intention of scraping the UPC was to use it to match products across stores but since none of the other stores were able to pull the product UPCs either it became unnecessary, and its value was set to a placeholder string. The Description value was set to be the product's name since most product names were already descriptive of the product.

The original idea to get URLs to scrape from was to use the URLs for the store's categories that matched up with the search terms in searchTerm.txt and save them to a .txt file. This way only the products relevant to the search term would be scraped and the script wouldn't have to regenerate the URLs each time it ran. The categories, though, were either too specific or too general to match up with all the search terms. Instead, the script would have to generate all the URLs using a URL template. A URL template is the URL to a search result page with a missing search value. This way a complete URL could be made by tacking the search term on to the template. This meant that irrelevant products, loosely related to the search term, would appear in the search results as well. The Parsehub project was able to be altered to at least ignore package deals in the results.

4.2.i.4. Out-of-scope

The Parsehub program can take several hours to completely scrape even one search term. Parallelization is needed to make scraping faster, but it is not possible to accomplish in the time left.

It was originally planned to have the Parsehub program work on more than just bestbuy.com but time constraints prevent this. A new project would need to be created for any additional stores and there are a few changes to the script needed to make it more generalized.

4.2.i.5.QC Notes

As the tests are being written small changes are being made to the script to make the code more concise, readable and easier to test in more ways.

4.2.ii.Bluecart & RedCircle APIs

4.2.ii.1. Completed Issues

BlueCart and RedCircle APIs offer the ability to collect data from Walmart.com and Target.com, respectively, based on a subscription service, scraping the results of 1 page of search results for a specific search term. This service was used to create an initial data population from this brand. Due to cost constraints of development, only the initial free plan was used. **After the development of the Bobcat CLAWS platform, it is recommended that this service be set up on the monthly plan for $15/month, per API**. These subscriptions provide 500 API requests per month, and additional requests beyond this number at $.06 per request. With 500 requests per month, this allows the platform to update the prices and availability of items once per day across all 15 sub-categories offered, but only for the first page of search results. Scraping the data across subsequent search pages will require additional costs.

The scripts provided have implemented the following features:

- Isolation of credentials
- Transformation: providing data in the correct format to be used with the database injection tools
- Scheduling: The scripts are set up to be schedulable by a shell script at a user defined interval
- Targeted Product Data: Tools iterate through a current list of sub-categories defined by the scope of this project (all 15 sub-categories)

4.2.ii.2. In Progress Issues

The only issue with these scripts as provided is they currently contain an API key that is linked to a free account with limited requests. It is recommended that a subscription be obtained as part of the data collection framework.

4.2.ii.3. Failures – Resolutions

The BlueCart and RedCircle APIs and the provided scripts provide all the necessary functionality, currently only limited by the small number of requests provided to the current API key.

4.2.iii. CLAWS Scraper

4.2.iii.1. Completed Issues

The CLAWS Scraper was able to meet some of the requirements set out in the project proposal, as a tool to "scrape" data from Walmart.com to provide a baseline for data collection. The limitation is that the program requires a user to be available to click through a "Human Challenge" verification that occurs roughly every minute at current. The current implementation has the current features:

- Accessing Walmart.com after an initial "Human Challenge" verification.
- Searching: By passing a search parameter to the tool, a user is able to customize searches on the Walmart.com website
- Scraping: the tool is currently able to iterate through the containers on the search results page of Walmart.com, identifying products, clicking on their link, pulling relevant data from the page, and, using the browser "Back Button", return to the search results page.

- Pagination: As long as a user is present to assist with the "Human Challenge", the tool is able to scrape all the products on a page, and continue on to the next page of search results
- Transformation: The tool actively writes data to a JSON in the format needed for insertion into the database.
- Filtering: The tool is currently able to utilize filtering and sorting method from Walmart.com
- Error Handling: The tool is currently able to handle all but the most extreme errors: When the tool recognizes an error from the browser, the most common strategy is to simply skip that item.

4.2.iii.2. In Progress Issues

- Speed: The tool is currently set up to only access the site with waits in between button clicks and other interactions. This is to avoid the bot detection.
- User Involvement Necessary: Even with the inbuilt speed limitations, the tools session on the Walmart server will still regularly get flagged as a bot and be sent to another "Human Challenge" verification page. This requires a user to be able to click through the challenge to return to the Seach results page.
- Domain Specific: At current, the tool is only designed to be used with Walmart.com. To rekey this tool to work with other vendors would require an intimate inspection of the relevant website to identify reusable structures that can be identified by the tool.

4.2.iii.3. Out-Of-Scope

Due to the prevalence, ease of use, and success of other scraping techniques, namely the BlueCart and RedCircle APIs, further development of this tool is out of scope for this project. If necessary, this tool can be set up as an exploratory tool that can provide a small dataset for research and business consideration.

4.2.iv. Coupon API

4.2.iv.1. Completed

The development and upload of the Coupon API code have been successfully completed. This marks a significant achievement in line with the project's initial proposal, which emphasized the creation of a user-friendly coupon application, particularly focusing on Walmart and Best Buy products. The completion of this phase signifies a crucial step in ensuring the application's foundational functionality and sets the stage for further development and integration efforts.

4.2.iv.2. In Progress

Currently, the integration of the Coupon API with application products is underway. This phase is essential for the application's overall functionality, aiming to provide customers with easy access to discounts on various products. The progress in this area is a critical component of the project, as it directly relates to the user experience and the application's effectiveness in delivering value through coupons.

4.2.iv.3. Failure

The project has encountered a significant setback in its inability to find coupons for all products. This challenge represents a deviation from the initial objective of the project, which was to develop an application that seamlessly integrates coupons for specific items. The failure to secure applicable coupons for all selected products has necessitated a shift in focus. The project now aims to use the coupon API to obtain coupon codes for entire websites and stores, rather than individual products. This change in strategy highlights the difficulties faced in the coupon-sourcing process and the need to adapt to these unforeseen challenges.

4.2.iv.4. Out-Of-Scope

The exploration of the Honey application for coupons was determined to be out-of-scope for this project. Additionally, the original project proposal did not include a contingency plan for situations where certain coupons were unavailable or did not apply to the chosen products. This oversight has become more pronounced given the recent challenges in sourcing applicable coupons. The absence of a fallback strategy in the initial planning stages has emerged as a significant issue, underscoring the need for flexibility and adaptability in project management.

5. Updated Schedule

- 11/14 - 11/16 (3-day Sprint)
  - !!Continue isolation of relevant credentials.!!
  - Blue Circle/Red Cart API Requirements
    - Focus on completing Error Handling.
    - Have api scripts write to jsons or insert directly into Database
  - In-House Scraping – Walmart
    - Resolve compatibility issues with Chrome Driver and switch to GeckoDriver and Firefox.
  - Coupon Team
    - Continue Integration of Coupon API, Do unit testing, have API data inserted into databse
  - Quality Control
    - Start implementing tests for tAPI.py and wmAPI.py.
  - Server Team
    - Try to implement Self-Hosting Webpage
    - Work on Maintenance Scripts in logFiles Folders
  - Database Team
    - Testing: Insert scripts.
    - Testing: Stored Procedures.
    - Start creating stored procedures based on keywords for searching functionality.
  - Front End Team
    - Fix refresh bug issues
    - Enhance search bar functionality.
- 11/17 - 11/20 (4-day Sprint)
  - In-House Scraping – Walmart

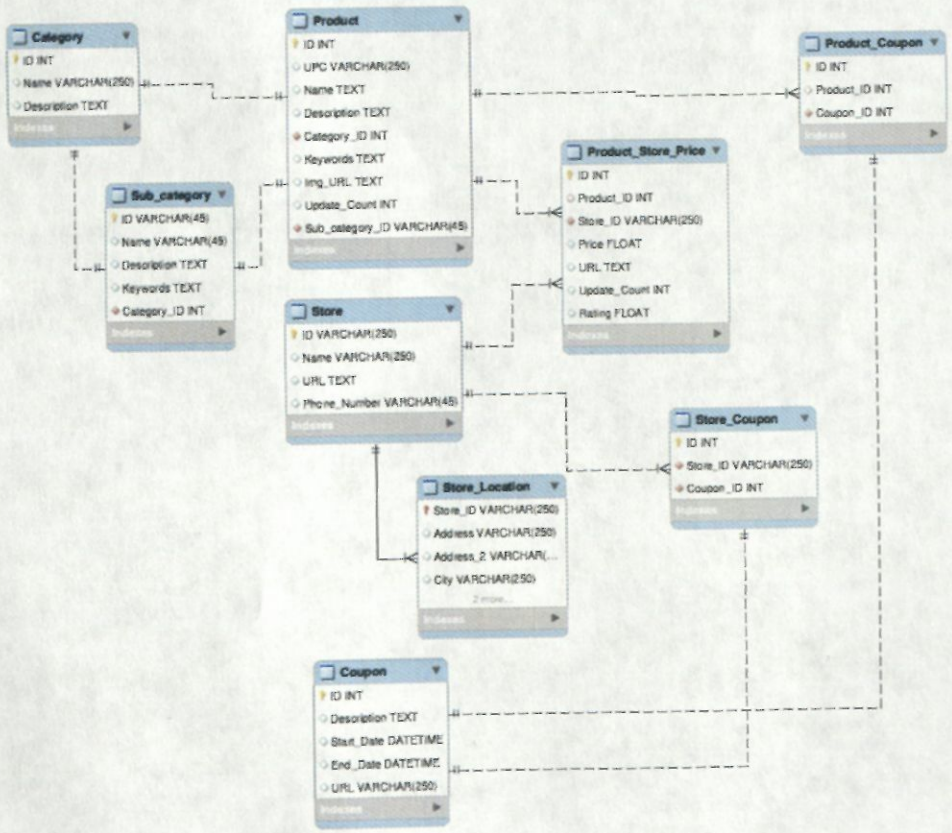
        - Implement custom "waiting" configurations.
    - Coupon Team
        - Aim to complete Integration of Coupon API script and inserting coupon data into database.
        - Run unit testing on both the API call and Inserting into database
    - Quality Control
        - Continue testing all APIs.
        - Have a template for documenting unit tests
    - Server Team
        - Finalize Maintenance Scripts.
    - Database Team
        - Continue creating and testing subroutines.
        - Start working on stored procedures for homepage products.
    - Front End Team
        - Complete product comparison page.
            - Web page is accurate and reactive
            - Filter sidebar includes more filters and sorting methods
        - Finalize search bar functionality.
- 11/21 - 11/23 (3-day Sprint)
    - In-House Scraping – Walmart
        - Aim to complete all scraping functionalities.
    - Quality Control
        - Finalize testing of all APIs.
        - Have a collated document of all unit tests, used for final documentation
    - Server Team
        - Ensure all server functionalities are stable.
        - Test Backups
        - Have an after-semester plan for repo and server
    - Database Team
        - Finish up subroutines for searching and homepage products.
        - Make sure Database schema has correct FK, PK, and contraints for security
        - Look at opening Database port to have a GUI implementation, like phpmyadmin
    - Front End Team
        - Ensure all functionalities are integrated and working smoothly.
        - Work on UI fixes
        - Try to implement AJAX (Scope extension)
- 11/24 - 11/27 (4-day Sprint)
    - General
        - Focus on integration testing across all teams. Make sure scripts execute correctly, make sure backups work, make sure major bug issues are fixed
        - Begin preparation for final presentation. Prepare documentation for Final deliverable.
        - Address any remaining minor issues.
        - Discuss after-semester plan for project

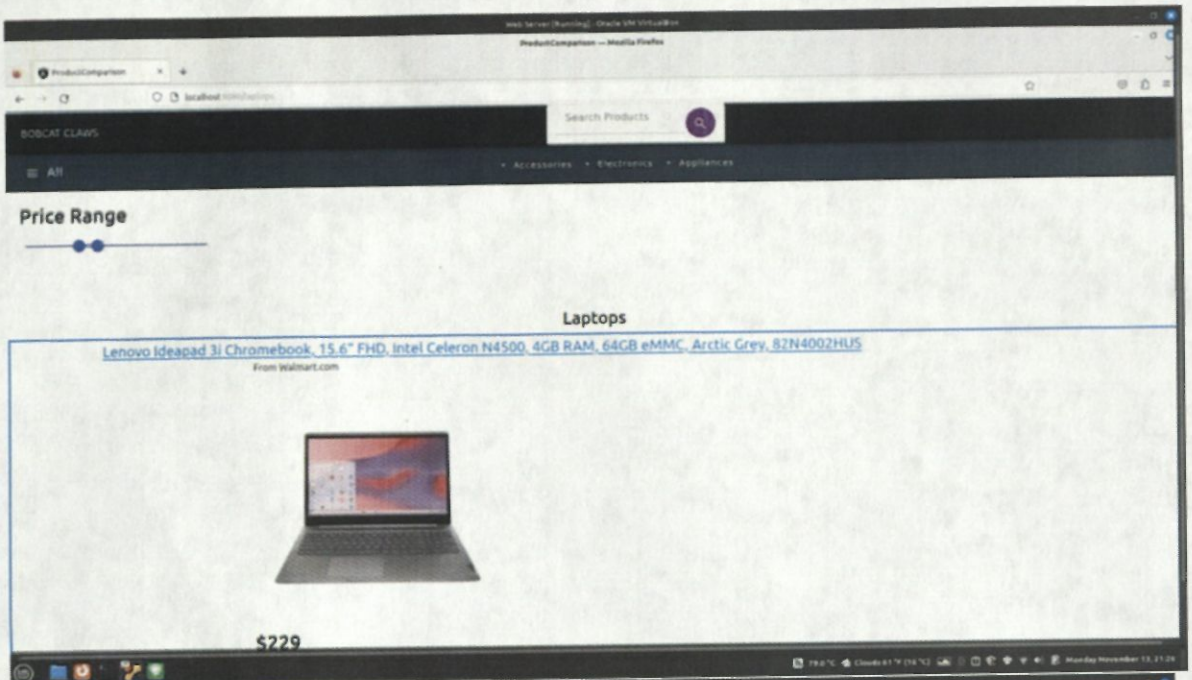
- Database Team
  - Finalize any remaining database functionalities.
- Front End Team
  - Polish and finalize all front-end components.

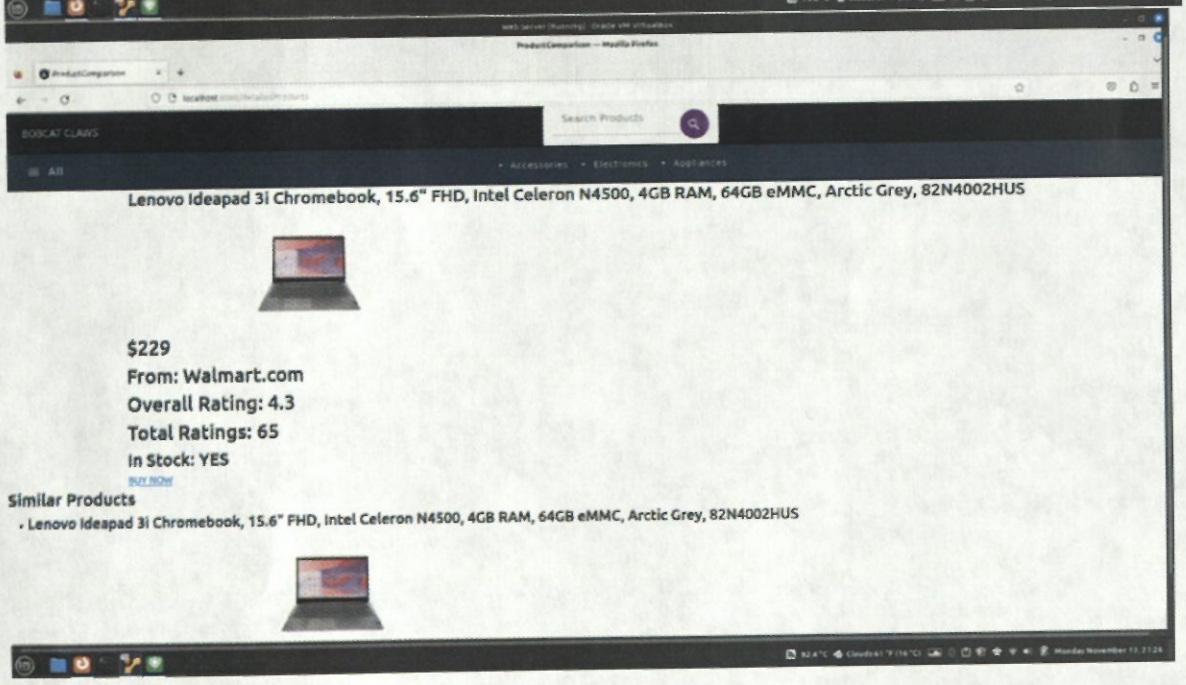
- 11/28 - 11/29 (2-day Sprint - FINAL)
  - General
    - Final checks and testing across all teams.
    - Preparation for project presentation.
    - Address any last-minute issues.
  - All Teams
    - Ensure documentation is complete and up-to-date.
    - Prepare for Final Project Deliverable. A Report, Presentation, and *Working* Product.

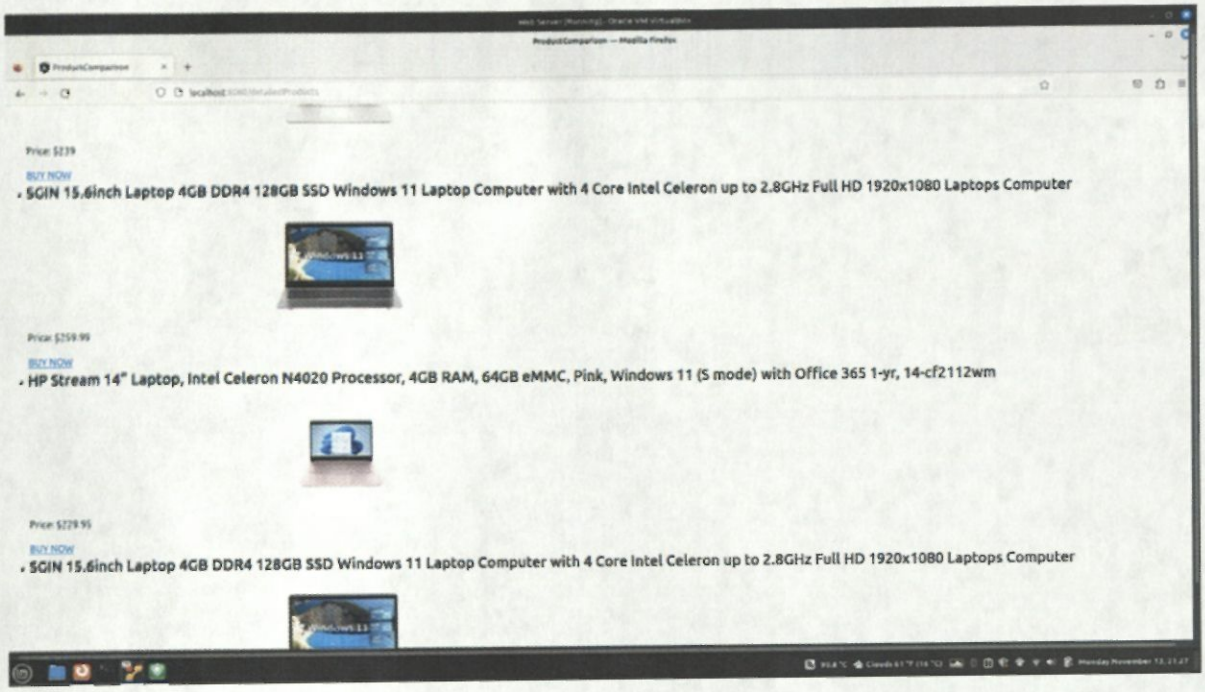
6. Updated Images

- DB Flow




BOBCAT CLAWS
Search Products
All
Accessories
Electronics
Appliances
Price Range
Laptops
Lenovo Ideapad 3i Chromebook, 15.6" FHD, Intel Celeron N4500, 4GB RAM, 64GB eMMC, Arctic Grey, 82N4002HUS
From Walmart.com
$229


BOBCAT CLAWS
Search Products
All
Accessories
Electronics
Appliances
Lenovo Ideapad 3i Chromebook, 15.6" FHD, Intel Celeron N4500, 4GB RAM, 64GB eMMC, Arctic Grey, 82N4002HUS
$229
From: Walmart.com
Overall Rating: 4.3
Total Ratings: 65
In Stock: YES
BUY NOW
Similar Products
Lenovo Ideapad 3i Chromebook, 15.6" FHD, Intel Celeron N4500, 4GB RAM, 64GB eMMC, Arctic Grey, 82N4002HUS


ProductComparison — Mozilla Firefox
ProductComparison
Price: $239
BUY NOW
• SGIN 15.6inch Laptop 4GB DDR4 128GB SSD Windows 11 Laptop Computer with 4 Core Intel Celeron up to 2.8GHz Full HD 1920x1080 Laptops Computer
Price: $259.99
BUY NOW
• HP Stream 14" Laptop, Intel Celeron N4020 Processor, 4GB RAM, 64GB eMMC, Pink, Windows 11 (S mode) with Office 365 1-yr, 14-cf2112wm
Price: $229.95
BUY NOW
• SGIN 15.6inch Laptop 4GB DDR4 128GB SSD Windows 11 Laptop Computer with 4 Core Intel Celeron up to 2.8GHz Full HD 1920x1080 Laptops Computer