Circuit Collective

Lab 11
Demo Presentation for April 4th

Team Members: ***William Wedemire, Bryce Gill, Nathan Aguiar, Tobenna Nnaobi, Jerry Yang***

Agenda for Lab 11 Meeting

For this meeting we will…

1. Present plans to finish, polish, and deliver the software.
2. Discuss any newly developed features since last meeting.
3. Discuss plans for documentation of the software.
4. Verify contentedness with the final feature list.
5. Showcase the results of the initialized test cases.
6. Present a live demonstration of the catalogue system.
7. Present the task board and Github Repository.
8. Present the overall Project Retrospective.

Timeline to Due Date:

- Project Due on April 6th
- One Task Remaining: Purchase Log (Est. 3 Days, 1 Day completed).
- Finalize Codebase (Final Report)
- Deploy Packaged Executable (Final Report)

Friday	***Saturday***	***Sunday***
Demo Purchase Log	Purchase Log Final Report	Final Report Create Executable

Documentation Plan

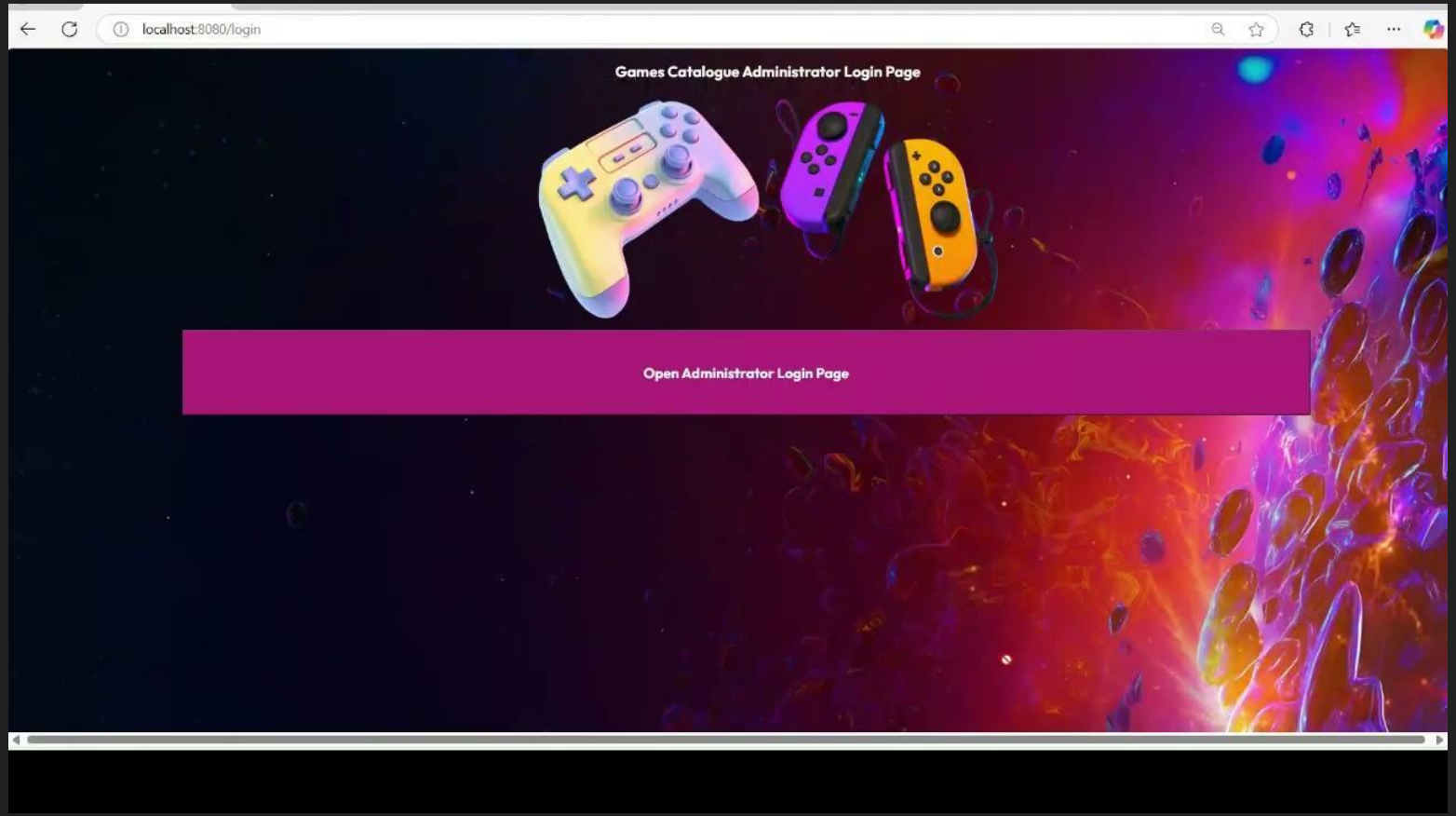
Developer Documentation on Github:

- Instructions on Dependencies and How to Build Project
- Describes The Project's Code Structure
- Explains How to Access The Interactive API Documentation.

User Documentation will be in form of demonstration videos to show functionality of the product.
Intended to be easy to distribute to new employees.

Recorded Software Demonstration

Software Demonstration Recording



Testing Plan Results

Clear Box Test Cases

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Testing Results
Unit Test	Nathan	insert()	Backend test that checks if new item is successfully added to database or not. The test will check what happen if input is different, such as null, or different item name, price, stock, etc..	true
Unit Test	Nathan	list()	Backend test that checks function used for getting list of item, the test will check what function will return with different input.	true
Integration Test	Nathan	Web API on localhost	Checks if web api can be accessed by javascript on frontend with given port, if success, JSON responses depend on method called by fetch() in javascript.	All API worked successful
System Test	Nathan	Logging	Check if user successfully logged in with correct username and password, and check is user directed to category page or not.	Login worked successful

Translucent Box Test Cases

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Testing Results
Unit Test	Nathan	search()	Backend test that checks if search function works successfully with different input, if success, function will return index table with given input as search result. Apache Lucene Search Algorithm is used and warped by custom function.	true
Integration Test	Jerry	Inserting item	Check if item info provided from frontend is being successfully added to database or not.	Insert worked successful
Integration Test	Jerry	Delete item	Check if item info provided from frontend is being successfully deleted from database or not.	Delete worked successful
Integration Test	Jerry	Search item	Given input from frontend, checks if search returned from backend is matching/approximate as input or not.	Search worked successful, result of search matched input
Integration Test	Jerry	Edit item	Given input with what attribute is being edited and what is changed, check if backend have successfully record new changes or not.	Edit worked successful

Opaque Box Test Cases

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Testing Results
Unit Test	Tobenna	fetch(`${api}/game/search?l=25&q=${query}`)	Calls from javascript, fetch data from backend using search API, if it success, check if result is reasonable or not.	true
Unit Test	Tobenna	fetch(`${api}/game`)	Call from javascript, creating a new item and insert it to database, if it successfully added to database. Result will return and display in category page.	true
Unit Test	Tobenna	fetch(`${api}/admin/game/${id}`, { method: "DELETE"})	Call from javascript, deleting item and removing it from database, if it successfully removed from database. Result will return and display in category page.	true
Unit Test	Tobenna	fetch(`${api}/admin/game/${id}`, { method: "POST"})	Call from javascript, changing attribute of item, if it is successfully changed and recorded in database. Result will return and display in category page.	true

Opaque Box Test Cases

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Testing Results
System Test	Jerry	Search bar	Typing string inside search bar, hit enter will filter out category list, only search result will be displayed in list. Different input will be test to check validity	Search bar worked successful and given reasonable result
System Test	Jerry	Website category item list editing	In web page, test checks inputting different item and using add to see if item is added into category list or not, use delete button checks if it is deleted from category list or not, and use edit to change item attribute checks if the attribute change is displayed inside category list or not.	All category function worked successful

Interesting Bugs That Were Discovered (All Solved)

1. ***Text Field's Text Not Resetting & Clearing***
   - After adding item, the text within the text field remains instead of clearing its contents.

2. ***Login Page Redirection***
   - After attempting to log in as an administrator, the login page does not direct to game category page.

3. ***Inserting Incorrect Data Types via JSON Parsing***
   - Inserting the wrong data type causes the javascript file to break and failing to fetch the data from the database.

4. ***Invalid Index ID***
   - Each initialized item has a random and/or incorrect ID instead of an organized index ID

Software Iteration Updates

Current Software Iteration 3 Updates

Search Bar Recommendation List - Done

Add functionality to the search bar to give recommendations based on the search

Priority: 30
Estimated Time: 4 days
Assigned: Nathan, Tobenna

Purchase Log - In Progress

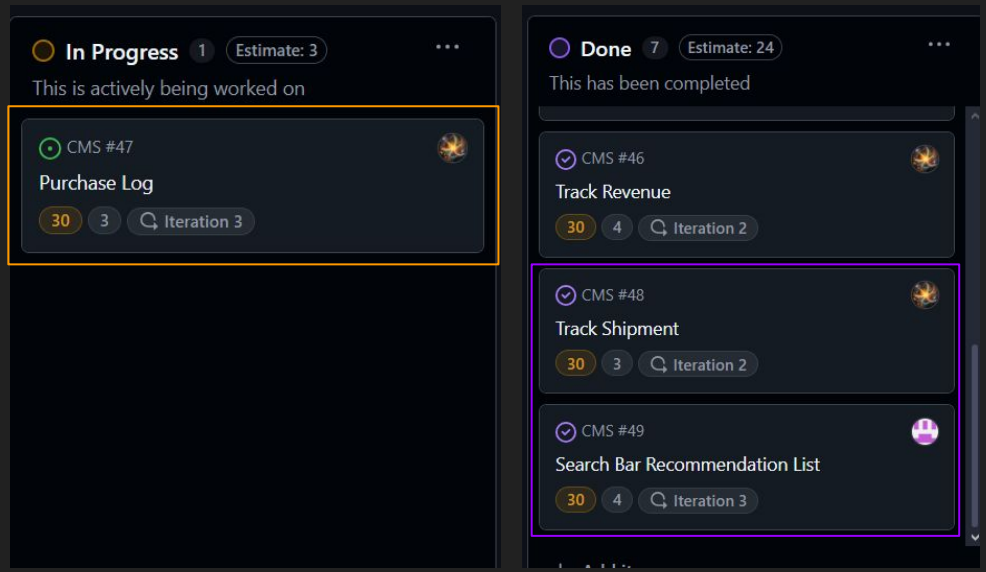
Add a purchase log so managers can manage customer purchase log

Priority: 30
Estimated Time: 3 days
Assigned: Nathan, Bryce, William

Track Shipment - Done

Allow managers to keep track of shipments.

Priority: 30
Estimated Time: 3 days
Assigned: Nathan, Bryce, William



Current Feature List

Search Bar Track Stock Login Password	Done Done Done
Game Tag Revenue Tracking	Done Done
Search Bar Recommendation List Purchase Logging Shipment Tracking	Done In Progress Done

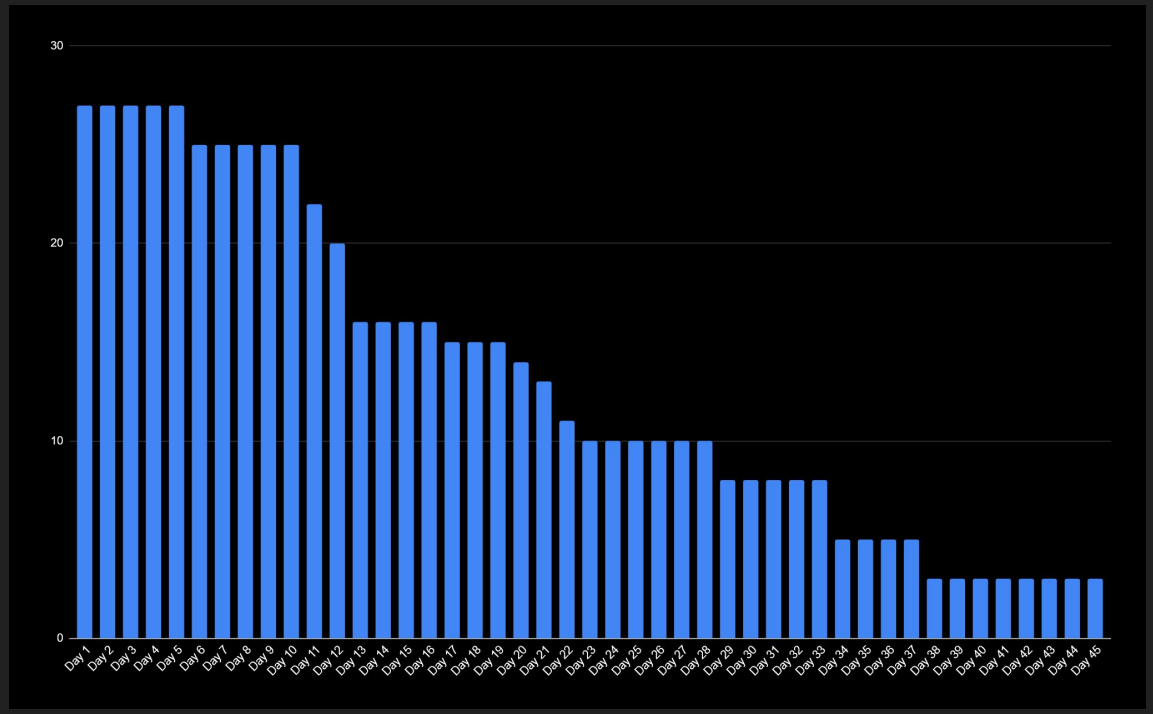
Final Velocity

Estimated Days: 10

Velocity: 0.7

Days Required: 15

Burn-down Chart



Iteration 3 Progress:

Search Bar Recommendation List - Done

Track Shipment - Done

Purchase Log - In Progress

Total: 8 Days

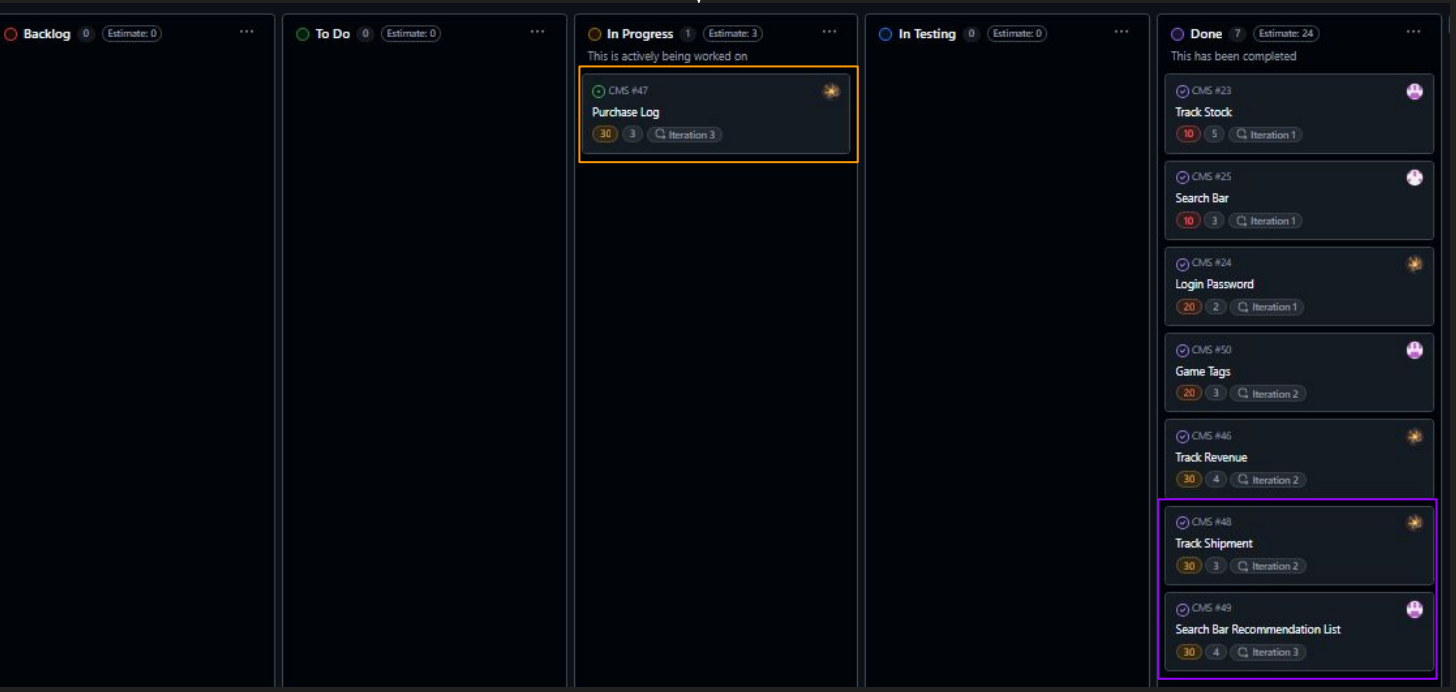
2 Days Remain

Task Board

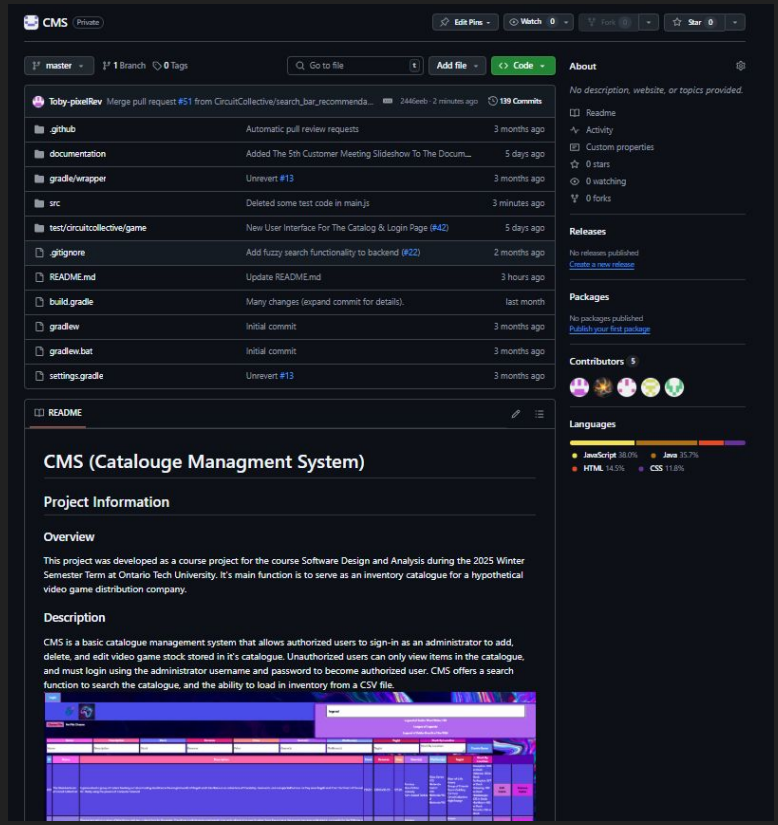
Moved to In Progress

Since Last Meeting

Finished
Track Shipment
Search Bar
Recommendation List



Repository Screenshot



https://github.com/CircuitCollective/CMS

Project Retrospective

What Went Well?

- Writing backend is a fun experience
- Writing frontend is enjoyable
- Meeting Consistently
- Peer programming
- Team dynamic

Lessons Learned?

- Time management in a group setting
- Navigating collaborative programming
- Leadership skills
- Adapting to peer dynamics
- Staying productive in a high stress environment

Most Important Takeaway?

- Software development is a team effort!
- (Fade to group riding off to a hallmark movie soundtrack)

Changes For Next time? (Process)

- Meeting earlier than Thursday night
- Better communication
- Clean up the Repository
    - Remove unnecessary branches
- Showing up to the lab before the start time

Changes For Next Time? (Technology)

- Ensure that the repository stays clean
    - Close/Merge unnecessary branches
    - Start documenting earlier in the project
    - Merge more frequently (Branch Conflicts issue)
- Have more frequent testing and validation of code

Questions & Feedback