StationX Final Submission

Lily Trissel, Lavani Somesan, TerriLynn Hale, & Nash Henry

Capstone Project Fall 2021

The University of West Florida

12/3/2021

CIS4595 Capstone Project

Dr. Bernd Owsnicki-Klewe

Abstract

This paper will serve as the final update on the progress of StationX, a web application, created by It's Not a Bug. StationX was created to provide users a way to purchase game software and stay up-to-date on the latest gaming news all in one place. Over the course of this semester, It's Not a Bug has brought this web application to life. The project requirements, which were first laid out in the project plan, have not changed much in that time. The timeline that we were attempting to follow did change quite a bit due to unforeseen circumstances. The project results are what we expected to produce at this time. We have evaluated the security of our software as well as evaluated the application itself for bugs. Although we completed a lot of the project, if we had more time we would like to bring more ideas to life.

Table of Contents

1 Executive Summary

The web application we have created is a game distribution platform called StationX. The primary goal of our web application is to facilitate a secure method for users to purchase gaming merchandise on our platform and give the latest news/updates on gaming content happening around the world. Our application keeps our user's information private by encrypting passwords using a hashing algorithm as well as using API tokens to retrieve any user data and updating those API tokens whenever a user logs out. Users are able to create an account, view a wide array of merchandise, make purchases (soon to be added functionality), update their password/profile information, view news/updates on games/gaming content, etc. through our web application.

2 Project Requirements

Some of the initial requirements for the web application included functionalities that you would expect from a merchant web application. By this, we mean having a database, the ability to create and manage an account, and being able to manage a cart with products. There were also some initial security requirements that we attempted to implement regarding the general security of the web application. These included password encryption, session management, api tokens, and overall application security.

We were able to implement all of these initial requirements as well as some additions. One of the additions that we accomplished was an account settings page where users can manage their accounts. Another was to add a NewsFeed Service. Although we always thought the web application would someday get the NewsFeed service, we did not think that we could do it in time for the final submission so it was left out of the initial requirements.

3 Timeline

3.1 Initial Timeline

3.1.1 Initial Sprint 1

Weeks 9/20 - 10/4

- Work on Project Plan
- Familiarize ourselves with node js/any other language needed for this project
- Set up initial files for project
- (9/27) Meet with client to discuss project goals/structure
- Work on Individual Plan
- Get NodeJS server running on localhost
- Set up basic HTML pages for the website i.e. Home, Games, Merch, etc.
- Setup Login Page
- Setup Create Account Page
- Add Search Bar
- Security Testing on Application

3.1.2 Initial Sprint 2

Weeks 10/5 - 10/25

- Work on Technical Document
- Setup API (acts as intermediary between server and services)
- Setup user service
- Setup MongoDb and Mongoose and connected it to our services
- Add Create User Functionality
- Add Login/Logout Functionality
- Error Testing on Login
- Setup NewsFeed Page and populate
- Add Session Functionality
- Work on FrontEnd

- Work on Presentation 2
- Work on Group Report 1
- Refactor Code
- Security Testing on Application

3.1.3 Initial Sprint 3

Weeks 10/26 - 11/15

- Work on Group Report 2 and 3
- Add Inventory Service
- Add Shopping Cart Functionality
- Add/Remove from Cart Functionality
- Add Inventory to Database
- Populate games/merch pages with inventory
- (11/01) Met with client to discuss project status/update
- Work on FrontEnd
- Work on Error Handling
- Work on Search Functionality
- Refactor Code
- Security Testing on Application

3.1.4 Initial Sprint 4

Weeks 11/16 - 12/03

- Work on Individual Presentation/Report
- Add Purchase Service
- Add Purchase Functionality
- Add Purchase History Page/Functionality
- Work on FrontEnd
- Add Logging Service
- Make sure everything is up and running
- Security Testing on Application
- Work on Final Presentation/Demo

3.2 Actual Timeline

In our actual timeline, most of the functionality requirements were pushed back a few weeks. It took a long time for the team to get comfortable with coding in NodeJS, which was an unexpected bump in the road for us. There was also some functionality that had to be forgotten because we were so behind schedule, i.e. purchase service, logging service, etc. Although these functions were left out, we added some functionality that we did not originally plan for, such as an account settings page. Overall, we think that we have done a lot for the project, although the website is nowhere near ready for deployment we completed more than what we thought we would be able to do.

3.2.1 Final Sprint 1

Weeks 9/20 - 10/4

- Worked on Project Plan
- Familiarize ourselves with node js
- Set up initial files for project
- (9/27) Met with client to discuss project goals/structure
- Worked on Individual Plan
- Got nodejs server running on localhost
- Set up basic HTML pages for the website i.e. Home, Games, Merch, etc.
- Setup Login Page
- Setup Create Account Page

3.2.2 Final Sprint 2

Weeks 10/5 - 10/25

- Worked on Technical Document
- Setup Navigation Bar
- Worked on Routing
- Setup API (acts as intermediary between server and services)

- Setup user service
- Setup MongoDb and Mongoose and connected it to our services
- Added Create User Functionality
- Added account guide page
- Worked on front end
- Did security testing with current functionality
- Worked on Login functionality
- Worked on Presentation 2
- Worked on Group Report 1
- Changed structure of files

3.2.3 Sprint 3

Weeks 10/26 - 11/15

- Worked on Group Report 2 and 3
- Worked on Login Functionality
- Error Testing on Login
- Created API token that's stored with user
- (11/01) Met with client to discuss project status/update
- Added Session Functionality
- Added Flash Messages to notify user if an action they have done was successful or not
- Added Update API token functionality when user logs out
- Worked on FrontEnd
- Worked on Error Handling
- Added Inventory Service
- Added get games/merch functionality
- Added inventory to our database
- Populated pages with our inventory
- Worked on Profile Page
- Worked on frontend

3.2.4 Sprint 4

Weeks 11/16 - 12/03

- Worked on Individual Presentation/Report
- Added Shopping Cart page
- Added Shopping Cart Functionality
- Added Search by Title or Brand Functionality
- Added Account Settings Page
- Added Change Password, Update Email, Birthday, or Name Functionality
- Fixed Bugs and Refactored Code
- Added NewsFeed Service
- Added NewsFeed Page
- Worked on FrontEnd
- Added Delete Account Functionality
- Added Encryption to Password
- Added Toggle Button for Password Input
- Did Security Testing on Application
- Worked on Final Presentation/Demo

4 Project Results

4.1 Project Outcomes

The final expectations of our project results match up pretty well with our initial one. There are a few things that didn't make it such as having the user make a purchase, viewing purchases, having a logging service, etc. due to time constraints. We also exceeded some aspects of our project results such as having an account settings page where the user can change their password, update their profile, and delete their account.

4.2 Anticipated Project Problems/Solutions

1.) *Not being able to get MongoDB to work correctly with node js.*

- If we are unable to get MongoDB to work then we will switch to another database server like Amazon's AWS.
- We were able to use MongoDB for our web application with very few problems.

2.) *Getting stuck on a particular feature/function of the project.*

- If we get stuck on something we can consult Dr. O.K to give us advice on how to solve the problem or we may have to disclude the feature if we cannot get it to work properly. We will be using version control to make it easy to remove or add functionality when needed.
- We did get stuck on a few features but were able to work together to solve them.

3.) *The client does not like the direction we are headed in/wants something more or something removed from the web application.*

- The solution here is to listen to whatever the client wants, as they are the owner of the application. If needed we will head in a different direction or add/remove functionality as we go as dictated by the client.
- Our client was satisfied with the results of the project.

5 Evaluation Strategies

5.1 Security Testing

To test the security of our web application we used automated tools, such as DirBuster, and manual tools, such as Burp Suite. We wanted to ensure that our web application was as secure as possible and locate any potential vulnerabilities.

5.1.1 DirBuster

First, we used DirBuster which is an automated tool that scans for directories and files on a web/application server. We were able to see most of the files that were outward-facing with this tool but not all due to this being a dictionary attack. Files that were found during the scan were the login, about, css, games, home, search, and user pages. A few of the pages that we had present, such as the create account page, were not able to be detected by DirBuster. A way to keep files from being detected by DirBuster is to change the naming conventions of our files from simple short file names to something slightly longer but that still conveys what the file's purpose is. Another way to stop DirBuster from detecting files is to limit the number of requests a user is allowed to make to a server before a time out so that dictionary attacks can be rendered ineffective but that can also affect how other users are able to utilize the website.

5.1.2 Nikto

The second tool that we used was Nikto which scans a web/application server for vulnerabilities involving the HTTP header as well as files that could be used as an attack vector against the server. Nikto also does a directory scan just not as well as DirBuster so using these two tools together nets better results overall. The only two files that appeared when it came to the directory scan were the /home page and the /login page.

5.1.3 Burp Suite

Burp Suite was the main tool that we used for security testing. Burp Suite has many different functions such as different dictionary attack payloads as well as it gives us the ability to capture and manipulate HTTP responses. With Burp Suite we did a sniper dictionary attack with a MongoDB NoSQL word list to attempt a NoSQL injection attack to bypass the login screen, this ended in a failure. We also attempted to do some session walking with the response capture functionality, this ended in a failure due to how we have our cookies set up. It doesn't go in sequential order so it makes it improbable to do session walking but not impossible. If someone is logged in and an attacker happens to guess the proper values of their cookie they can steal their session.

5.2 Application Testing

For our application, we will be using User Acceptance Testing as well as functionality testing as we continue with the development of the application. For documentation, we will have assumptions of how the application will run before testing and a list of test cases.

We ended up using a series of Gherkin tests and user tests to test the web application for functionality. The Gherkin tests were created as a base to test how the web application should react versus how it is actually reacting. We used Gherkin testing for creating an account, logging in, and logging out functionalities. To test other aspects of the web application, we simply used testing methods such as error handling, input validation. An example of these testing methods include trying to route to non-existent pages, and inputting passwords that do not meet the length requirement.

6 Future Updates

We accomplished a lot on this project, however, we do think that in the future there need to be some changes made to the web application. One of the major changes that will likely be seen is adding more services, specifically logging, billing, and shipping services. There also needs to be some security improvements to the application as well. One thing we would like to add to help improve overall security is two-factor authentication. We also plan to refactor the code for improved readability when maintaining it.