**Secure Banking Application Project:**

**Web Application (65%)**

Department of Computing, Atlantic Technological University Letterkenny, Donegal

by

Michelle Ogunade

L00176523

Lecturer: Maria Griffin

Table of Contents

[Introduction 2](#_Toc197034959)

[Creating the project 3](#_Toc197034960)

[Functionality 5](#_Toc197034961)

[Conclusion 6](#_Toc197034962)

[References 7](#_Toc197034963)

## Introduction

**Introduction to the report (6/7 lines), tell the reader what this report is about, explain that you are going to describe the application that you created and how you made sure it would be secure.**

This is my CA3 – Code Project for a Bank System

This report outlines the development of a Secure Banking Web Application created as part of the CA3 assignment for the Secure Programming module. The application was developed to simulate a simple banking system that prioritises secure coding practices. I had an easier choice to approach this but I opted to challenge myself to develop this bank system as a web application.

In this report, I will describe how the application was created, what technologies were used, and how secure development principles such as authentication, password hashing, and secure data handling were implemented and tested.

## Creating the project

**Tell the reader how you created the project, did you use VS Code? MySQL Developer? Git Hub? XAMPP? What language did you choose, Java, C#, Python, explain why.**

This secure banking system (web application) was developed using Visual Studio Code as my primary IDE and MySQL Server for managing the relational database.

**Languages & Tools Used:**

* JavaScript, Java (partially), HTML, CSS, Bootstrap
* MySQL Server, Node.js, Express.js
* GitHub for version control, GitHub Desktop

JavaScript because of more experience and enjoy it more

I primarily used the ATU laptop (DELL) and tested on a DELL laptop running Windows 11.

The system specifications were:

* **Processor:** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
* **Installed RAM:** 16.0 GB (15.4 GB usable)
* **System type:** 64-bit operating system, x64-based processor

The main part

Project developed using Visual Studio Code as my IDE,

MySQLServer:

Connected to the database using the Command Prompt/terminal as well to run NodeJS

Node.js v20.4.0.

**Any issues you had with your set up. Did you use your own laptop, ATU machines, Virtual Machine, Hard Drive. You can be specific with the versions on the language that you used and the spec of your laptop etc.**

To create my banking system, I had to plan what pages I wanted to include and display to the user and the workflow logic

**Hardware & Devices**

Switching between my laptop and the ATU PCs

Had some issues with the ATU PCs with regards to running Visual Studio Code

Had a persistent error that kept saying that it couldn’t locate the Main method (although it was present)

**Programming Languages & Libraries**

I made use of Java, JavaScript, jQuery, HTML, CSS,

Bootstrap

**Database & Servers**

For the database, I used MySQL Server and determined the necessary number of tables and their associated attributes based on the system's core features and data flow.

Initially, I opted to use Live Server instead of XAMPP or installing multiple Node.js modules, to keep the setup lightweight and reduce complexity. However, building the server logic purely with JavaScript proved challenging. I had become accustomed to using Express.js and Node.js to build custom servers in previous projects, and the absence of these tools made user registration and database interaction more time-consuming.

Despite trying to avoid node modules at first, my prior experience with Node.js and npm packages made me realize that using them would streamline development and improve functionality, especially when handling backend logic and database operations.

Changed the datatype of accountNo to integer

But then I also wanted to include the prefix BOI (concatenation)

**Version Control**

I used GitHub for version control throughout the development of this project. It allowed me to efficiently track changes, and roll back to previous versions when necessary.

**Navigation Bar**

What links to contain on the main navigation bar and how the layout looks different depending on whether a user Is logged in or out and/or their role

**Users**

I had to identify what users and their associated roles within the system would be

## Functionality

**Overview of how your project works. Details the technologies that you used. What version. Explain the basic features of your application…. Login, Account creation, balance checking, deposits, and withdrawals etc.**

On arriving to the Home page,

The user has the option of logging in (if they have already created an account previously),

On the Register Page, the user can create an account

On the Account Information page,

The customer can view their account details, check their balance, deposits and withdrawals

The administrator can view their customer account information details

**How robust is it?**

**Talk about how your application handles errors gracefully, maintains consistent behaviour under different conditions, and recovers from failures.**

**Include screen shots…**

* Prepared Statements
* Error handling and validation
* Alerts

**How have you included Secure Coding Practices, show where you used prepared statements, password hashing, input validation, and data sanitisation. Details with the screenshots the Static Analysis tool(s) that you used. How did static analysis help you with developing your code?**

**Detail how you implemented Security Features such as password hashing and multi-factor authentication. 2FA does not have to work but should be implemented in the code and you need to show how you would do it and explain how it works. Detail the vulnerabilities of the system, identify potential security vulnerabilities and how you would mitigate against them, including**

**Show how you tested the application, explain how you tested the security features *e.g. secure login***

**Detail your unit tests add screen shots.**

**Did you find any bugs, did you find any issues/holes in your security?**

Yes, it wasn’t exactly easy using JavaScript to build it, it took some time for a uswr to be registered into the database, as I was avoiding using node\_modules to begin with for less stress. But I’ve had enough experience with node\_modules in the past with previous modules, I believed it would help

## Conclusion

**What did you learn from doing this report, how secure do you think your application was overall, what would you change/do differently?**

I believe that my web application is very secure overall

That a lot of research and preparation advance needs to be done

I found the project both challenging and insightful overall

**What was the most challenging part. Did anything surprise you as being easier/harder, something that you didn’t think was needed, or anything that you did that you realised wasn’t necessary or could be done a different or better way?**

The most challenging part I think was understanding the workflow in chronological order

I was struggling to pick between JAV and JavaScript as my main

programming language to use for the project.

I feel more comfortable writing code in JavaScript as I have not practiced Java in years so it felt like a whole new language again

At the same time, I was only a little bit familiar with Unit Testing in Javascript so that was also something I had to learn, but I enjoy the learning aspect of it

I wish I had more time to read more into things or if I could find examples that were similar to my implementation

But other than that

Unnecessary features?

## References

Any references you might use should go here…