|  |
| --- |
| **Advanced Database COM526 Application Report** |
|  |
|  |
| |  |  | | --- | --- | | Name: Filipe Quartin  GitHub link: <https://github.com/fmilheir/inventory-tracking>  Cyclic: <https://jade-clear-turtle.cyclic.app/> |  | | Stn: Q15287599 |  | |
|  |

#### 

# **Introduction**

Managing a business is a complex and challenging task, one of the most critical aspects of which is the accurate tracking and management of inventory. The ability to know exactly what products are in stock, in what quantity and when they are sold is crucial for the smooth operation and success of any business. The problem is that this task can be overwhelming and time-consuming, especially for larger enterprises. My application aims to solve this problem by providing businesses with a powerful tool to effortlessly and accurately track and manage their inventory. From monitoring inventory levels, to tracking stock movements, my application is designed to help businesses streamline their operations, improve their efficiency and ultimately achieve greater success.

***System Overview***

This application uses a variety of technologies, in order to make it clear and simple for me, I decided to use the MVC model (Model, View, Controller), it is a very organized way and a verry good practice when dealing with full stack applications, making it simple and organized.

About the technologies I used for this project, for the front end I used ejs, css and bootstrap, and for the backend node and JavaScript connecting to MongoDB Atlas One of the main reasons why I decide to use Bootstrap is its responsive design feature which is enabled by its grid system. This allows developers to create websites that automatically adjust to different screen sizes and devices. Bootstrap also provides a wide range of pre-built UI components such as forms, buttons, navigation, and modals, which can be easily customized to match a website's design.

My application is divided in 2 main status:

Page with no login, where a talk about the application, and give the user to create an account or login, talking as well why this application was build, and the costs that it has.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

the logo was designed using canva.com

The second part of this application is the pos-login, where the user has access to the database, here he stores the items, in order to keep track of the uses used I created a class of sold items.

This class simply stores the items deleted by the user so it can be seen how may item have been used.

I enable the user to search for the type(stock, sold) a filter and as well to search the item by his name using the search bar.

The filter have the option of display all as well.

The logo is a button for the home page, if its logged in directs to the /visualization, else to the /index.

Graphical user interface, text, application, email

Description automatically generated

Icon

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated Graphical user interface, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Diagram

Description automatically generated

***Key Design Decisions***

For this application I crated two different collections:

* Users
* Datas

The user collection is responsible of storing the user data(username and password), the password is encrypted using BCrypt. BCrypt is a slow algorithm which is good for security as it increases the time it takes to brute force the password.  
It also stores the time of creation and the last time of modification.

The Datas collection is responsible of storing the items that each user wats to keep track of. It consist of an Item, a quantity and a type, the type is just a simple Boolean that tells me If the Item is in stock or if it was sold.

The Datas collection was suppose to take as a parameter for column the user\_id, in order to just show the items related to the user, unfortunately I was not able to make it work.

In the security measures, I tried to create a session to validate the user, but I was getting the error(const user = await User.findById(req.session.userID);), so I left it commented in the code.

Two of the biggest improvements I could ve made beside the sessions for the user, would’ve be using https instead of regular http connections, and give as referred in class the login responsibility to a third party company like google.

***Challenges***

I faced my self with many challenges while building this app:

.first I had an extension called balckbox, this extension was influencing the submissions of my forms, making my app to crash, I spend several days to find out about this problem.

I was receiving the following error:

Uncaught TypeError: Cannot read properties of null (reading 'CodeMirror') at <anonymous>:7:17

. the second error that I faced was because of my internet connection DNS, giving me a MongoDB connection error.  
 I spend a lot of time as well in order to figure out wat error was as well since It could be caused for several reasons.  
In order to solve this error I was forced to use my mobile data connection to receive a different DNS.

I was receiving the following error:

Error: MongooseError: Operation `users.insertOne()` buffering timed out after 10000ms

***Evaluation***

I had big hopes for this project and was really hoping to ace it. But things didn't go as planned and I hit some roadblocks that really demotivated me and made me waste a lot of time trying to fix them. Once I got past those problems, I got my mojo back but there wasn't enough time left to finish the project the way I wanted, like adding all those cool visualizations I had in mind.

# **Conclusion**

This project's conclusion fell short of my expectations since I was unable to meet several of my objectives, leaving me unsatisfied. Despite this, I feel the project's concept has potential, and there may be space for development by identifying more practical methods to help specific circumstances. My inability to correctly implement the cyclic (sometimes does not work) functionality was a huge setback because it is one of the funniest parts of the programme since you are able to access it from anywhere . To develop the programme further, I would concentrate on finishing the sessions, ensuring that users only have access to their saved products, and generating visuals to highlight changes in inventory and sales.