



University of
East London

Pioneering Futures Since 1898

SCHOOL OF ARCHITECTURE, COMPUTING AND ENGINEERING

Department of Engineering and Computing

CN6000 Dissertation Project

Contactless Mobile Web Reward System to Enhance Customer Engagement

A report submitted in part fulfilment of the degree of

BSc (Hons) in *Computing for Business*

By

Kieron Oates

ID: 2034853

Supervisor: Dr. Hisham AbouGrad

15 October 2024

Abstract

Purpose – Many large supermarkets provide their customers with reward schemes to increase their customer repeat purchases and retention, small local businesses do not provide these same features. The reason to this might be thought of as larger supermarkets have an abundance of experts and resources to provide this reward system, however this report aims to explore the process of implementing a reward system that provides features that allow small businesses to also increase their customer retention and engagement.

Approach/Methodology – For the project to succeed in its purpose, It will include a literature review that outlines relevant studies and research to assist in the development of the system. To accompany this the project will follow a methodology and for this I chose agile, this is due to its focus on developing and testing prototypes to help the project reach its goal by a set deadline. A research approach is also followed, and this project includes quantitative research using an online survey to gain an understanding on shopping preferences and the effectiveness of implementing a reward system.

Findings - The findings and outcomes of this project will consist of the results of the survey and what these results suggest towards the objectives of the system, as well as the user testing performed on the system to evaluate its performance and ability to achieve its goals. This findings will be supported by analysis on the results as well as evaluating the limitations and challenges it faced.

Acknowledgments

I would like to pay acknowledgements to my supervisor of this project Dr. Hisham AbouGrad as well as my module leader Dr. Fadi Safieddine for supporting me by providing guidance and resources to aid me through the development of this project.

I would also like to pay acknowledgement to the participant in my online survey as they helped provide a better understanding and insight to the projects research aims and implementation.

C ontents

Abstract	2
Acknowledgments	3
Chapter 1: Introduction.....	6
1.1 Introduction and Research Rationale	6
1.1.1 Research Question.....	6
1.1.2 Research Aims	6
1.1.3 Research Objectives	7
1.1.4 Project Plan	7
Chapter 2: Literature Review.....	8
2.1 What is Near Field Communication technology?.....	8
2.2 NFC Security	9
2.3 Applications of NFC.....	10
2.3.1 Smart Cards.....	10
2.3.2 Payments.....	11
2.3.3 Item Tracking	11
2.3.4 Authentication	11
2.3.5 Automation.....	12
2.4 Customer Reward Systems.....	12
2.4.1 databases	14
2.5 Gaps in the Literature	14
Chapter 3: Project Methodology	15
3.1 Approach.....	15
3.2 Project Methodology and Methods.....	15
3.2.1 Agile Methodology	15
3.2.2 Method.....	16
3.2.3 User Experience.....	17

3.3	Research Approach.....	18
3.3.1	Quantitative Approach using a Survey	18
3.4	Challenges and Limitations	19
3.4.1	Implementation.....	19
3.4.2	Research.....	19
3.5	Ethical Considerations	19
3.6	Conclusion.....	19
Chapter 4:	Implementation.....	20
4.1	Backend	20
4.1.1	Frontend.....	23
Chapter 5:	Results.....	29
5.1	Survey Results	29
5.2	Conclusion.....	34
Chapter 6:	Evaluation.....	35
6.1	Implementation Evaluation.....	35
6.2	Research Evaluation	36
6.3	Limitations and Challenges	37
Chapter 7:	Conclusion	38
7.1	Overview.....	38
7.2	Key Findings	38
7.3	Lessons Learned.....	39
7.4	Further Work	39
7.5	Conclusion.....	40
	Reference List.....	41
	Appendix A - Initial Project Proposal	44
	Appendix B - Final Project Proposal.....	45
	Appendix C – Source Code	47

Chapter 1: Introduction

1.1 Introduction and Research Rationale

Many large supermarkets will have an in house reward system for their customers, helping to encourage them to come back and shop again. The same cannot be said for smaller local businesses, this is why I will be looking to implement a reward system that can be utilised by these small businesses. Offering better prices and incentives to return to the shop in hopes that it will increase customer retention and satisfaction. A system like this can help improve turnover and profitability, especially utilising the technology of NFC. Jain, G. (2015) defines Near Field Connection (NFC) as the wireless communication between devices by transmitting information in a localised area. Combining this with what Coskun et al. (2013) said about devices like smartphones and smart cards, they can use this technology for contactless payments. I will be adapting on this use case to create a process where customer information is placed onto the smart card ready for the user to utilise its perks within the store.

The system will consist of a backend database made in MongoDB, it will store the data of the customers as well as relating the tables to make the database dynamic and allow for manipulation of the data to provide the customer with details on spending, trends, and potential deals. This will be paired with a web based interactive front end which will provide the user with the tools and information to use and assess the data in efficient and effective ways to maximise their shopping experience. Data will be stored and transferred between the database and the user by utilising NFC cards accompanied with NFC readers within the shop. Customers will have their own personalised accounts which will be tailored specifically to them. This will enable customers to increase their shopping experience and be rewarded for their actions whilst not needing to worry about carrying the necessary receipts, coupons, or debit cards. This system can also be utilised by staff of shops as it will provide information on inventory levels, popularity of products and potential theft of stock. Customers will not be able to access this admin area of the system on their account. This will allow customers to ensure they can get the most out of their shopping experience without barriers to entry like access to mobile phones and cellular plans.

1.1.1 Research Question

Is it possible to create a reward system that can help a small business enhance their customer retention and engagement?

1.1.2 Research Aims

The project seeks to develop a working demo of a JavaScript web and mobile interface with a dynamic relational database system that will be used by local shops to increase customer repeat purchasing and customer retention by integrating Near Field Communication (NFC) technology.

1.1.3 Research Objectives

When the project has concluded, I will be able to:

1. To research the literature of using NFC for database systems
2. To research the effective aspects of an efficient customer reward system
3. To conduct a survey to collect habits and opinions on shopping preferences.
4. To analyse opinions and implement requested functionality using quantitative data gathered using surveys.
5. To design and implement a working demo database system with interactive Graphic User Interface
6. To test the functionality of the system in comparison to other methods
7. To evaluation the tools and techniques required to design and implement a customer reward system and database
8. To reflect on the limitations and success of the system and evaluate the findings.

1.1.4 Project Plan

As the project has a set deadline, it's important to ensure that all components of the project can be completed within this timeframe. To assist me in achieving this, I created a Gantt chart to list out the project elements with subtasks listed in these elements. Progress bars are then created on the calendar to display the time required for tasks to be completed by. When each task is completed, they will be checked off the chart and increasing the total progression percentage. Through some planning into the tasks ahead, the chart can display tasks that can start production in parallel to each other and any slack between new tasks beginning. This planning and slack can assist in delivering the project on time as any unplanned breaks in development can be counter balanced with this slack.

Being able to identify the scope of the project before it has begun also helps to plan deliverable dates more accurately, which is why I also created a use case diagram. This outlines the “actors” e.g. Customer, as well as their processes in the project. Creating a visual representation of the project.

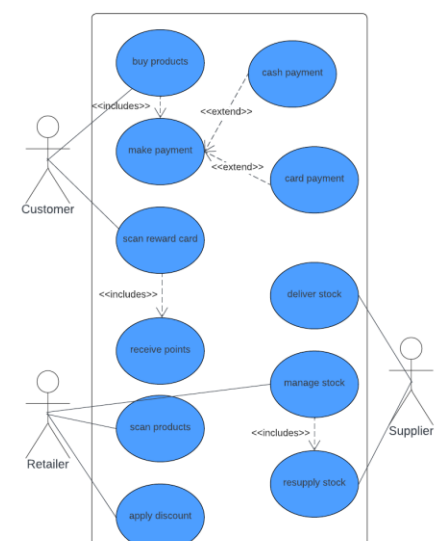


Figure 1 (above): Use case diagram showing scope of the system

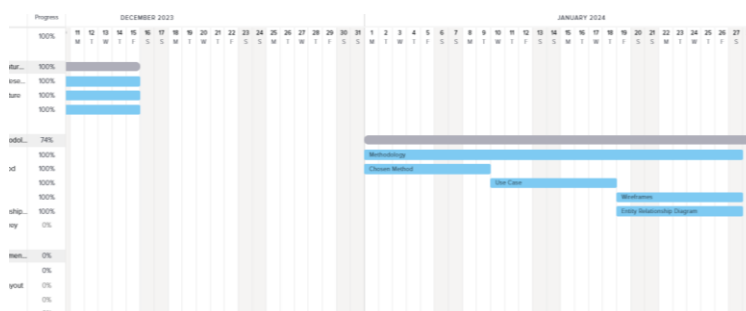


Figure 2 (beside): Gantt chart dated February 2024

Chapter 2: Literature Review

The utilisation of loyalty programs is becoming an increasingly popular technique to improve relationships with customers. This can be achieved through multiple approaches, but the outcome is the same, *“The companies that spend more time and effort in building strong brands and strong brand loyalty have proved to be more successful in the long run”* (Tabaku et al., 2015).

It's common practice for businesses to provide their customers with loyalty cards that give access to personalised accounts and the ability to use rewards granted for their loyalty (Demoulin and Zidda, 2008). These cards will use NFC technology to maximise efficiency and ease of use as well as allowing for easy communication with a backend stock management system.

2.1 What is Near Field Communication technology?

Near Field Communication (NFC) refers to how devices communicate through transmitting information in a localised area. nfc is the product of NXP Semiconductors and Sony Corporation and is the successor to RFID technology, Jain, G. (2015). NFC has many areas of implementation, Coskun, V et al, (2013) States that mobile phones, security locks, and smart cards can all make use of this technology as the electromagnetic waves emitted can be detected by their respective receiver when in close proximity, usually around 1 to 3 centimetres. The most common and prolific use case of nfc is contactless payments using a smart device, large service providers like Apple and Google have implemented this feature into their smartphones and allow its users to make these payments quicker and simpler. This is achieved through the active nfc source in the phone communication and transmitting the information to the receiver, active nfc is where a power source is used to power the technology, however this isn't necessary for NFC to work. NFC Chips are not required to have a power source or controller to work with the receiver, this is called passive nfc, MOU (2023). According to Blue Bite (2023), Passive nfc cannot receive information; it is only able to send information. Making it a lightweight and effective method of creating new ways of increasing customer engagement and loyalty opportunities. Blue Bite also mentions the components required in an NFC tag as well as the 5 types of nfc tags with their unique properties. They state *“In most cases, an nfc tag consists of an antenna + IC (chip), PET substrate, adhesive package and release liner”* indicating that minimal resources are required for nfc to work as well as the use cases are only limited by the implementation of the technology. Each type of nfc tag is defined by the total information storage limit as well as their ISO standard compatibility meaning each type will have greater performance based on the use case.

NFC can be seen as a continuation in the technology of RFID, which stands for Radio Frequency Identification, which works through communication and transmission of data through radio waves. It is stated by Coskun, V (2012) that RFID works similar to nfc as it uses passive or active tags that can communicate utilising electromagnetic waves to pass information to a receiver which can then interpret these signals and use the information. These tags have an integrated circuit and antenna inside them used for communication whilst not needing a dedicated power supply as they use radio frequency once in proximity with a reader. The application of RFID has been integrated into working life and has established its place as an extremely effective and simple way to enhance tasks that would previously be time consuming and have potential for error before its implementation. Areas in which RFID has proven its usefulness include public transport, inventory management and passports. This technology became prolific in these areas as RFID tags can have their own unique fingerprint that can be referenced and checked to initiate or complete a communication request.

2.2 NFC Security

The use of wireless communication will implement areas of security threat to personal data and the user involved, these threats and measures to protect against them will need to be considered and managed before NFC can be implemented into a system. Madlmayr (2008) states “*nfc is a bidirectional, proximity coupling technology based on the smart card standard ISO14443. The physical and data link layer of nfc peer-to-peer mode is already standardised in ISO18092*” meaning industry standard methods are put in place to ensure correct practices are followed and adhered to for ensuring security measures. nfc being peer to peer means data is exchanged between a client and a host, creating an opportunity for data to be intercepted or incorrectly exchanged. These ISO standards do not specify the need for data encryption to keep this data secure which means that this encryption will need to be implemented at the application level. Failure to implement this feature will lead to potential misuse of data and therefore present a major security risk.

According to Coskun (2012) and Haselsteiner (2006) , there are multiple types of risks that can occur with the use of NFC which include the following:

- Eavesdropping - where an attacker will try to replicate an nfc receiver that is capable of communicating with an nfc tag. This receiver is then brought in proximity to the target tag which will complete the peer to peer communication and can read the data from the tag.
- Data modification - the attacker generates an RF signal that overlaps the original signal emitted so the receiver does not receive the signal and the attacker can then modify bit values.
- Tag cloning - where data from a tag will be read and copied to be used by an attacker without permission from the target
- Data insertion - an attacker will add and insert extra data to the communication of nfc devices whilst they are communicating
- Data corruption - data is modified and manipulated to the point where the receiver is unable to understand the transmitted data and therefore denying the communication.

This is supported by Chattha (2014) as they mention the same methods, further emphasising how these are very common methods of security risks.

Haselsteiner (2006) also states some solutions and recommendations on how to combat these security risks, these solutions include:

- Eavesdropping - Passive and active data is always at risk of being eavesdropped due to the fundamental technology of nfc meaning that it is important to monitor NFC capable devices and the environment they are used in as an attacker needs to be in the local vicinity however creating a secure channel in which these communications can happen will help.
- Data modification - “*By using 106k Baud in active mode it gets impossible for an attacker to modify all the data transmitted via the RF link*”. This is not a perfect solution as some bits are still able to be modified but will majorly reduce the impact and possibility.
- Data insertion - attackers need to perform the attack while the communication happens so creating no delay between when the device answers will mean there is no time for the attacker as they cannot be faster than receiving the answer instantly which is impossible.

2.3 Applications of NFC

NFC as a technology is very simple but very powerful as an implementation into business actions due to its large range of use cases as well as its ability to perform many different tasks. Wireless communication and data transfer is the core of many potential applications that can be built upon and streamlined in a business. There are many examples of industries and applications where nfc has been implemented to enhance performance and ease of use which can be explored.

2.3.1 Smart Cards

An nfc smart card acts the same as an nfc tag as that is essentially what it is. The smart card contains all the same components needed for passive nfc communication. They are made into cards purely for beneficial reasons like it being easier to locate and store and being able to print branding/information on the card. Universal Smart Cards, (2021) have produced a blog post listing the industries that integrate smart cards into the infrastructure of that industry. Below is a table including the industry and how they implemented smart cards.

Industries using smart cards:

Industry	Implementation
Education	Student IDs to track attendance and securely store student information. Library cards to track ownership and duration of books.
Healthcare	Provide access to patient information and medical records. Enable easier access to doctor's offices, clinics, hospitals, and urgent care facilities.
Leasure and Entertainment	Used to purchase tickets, store admissions, and store rewards. Customers no longer must carry cash for transactions
Hotel sector	Guests access their rooms via an RFID keycard. Store information such as loyalty credits
Transport	Pay for public transportation tickets, access rental vehicles like bicycles
Retail	Supermarket loyalty programs store their data and transact with retailers. Storing details allowing the retailer to provide tailored offers to the customer. Also allow businesses to track customer spending patterns, to develop marketing strategies specific to the customer

Source: Universal Smart Cards, 2021

One common feature that all these implementations of smart cards have is that they increase the efficiency and reliability of a process within the field. If you were to measure the improvement made with an nfc card in only one instance, you wouldn't notice much of an increase, however its these small improvements that accumulate into a larger sense of security as these systems and processes allow for further implementation in other aspects surrounding the industry due to how fluent and efficient their small processing are running. This comes from a quote from a blog post on MRC saying *"NFC is primed to be the next big thing in technology. Not only is it extremely useful, it's also very easy to use... we will see more and more usage of NFC in everyday life"* (Stangarone, 2022).

2.3.2 Payments

Payments that would usually be done with a credit or debit card can be made contactlessly with a mobile phone due to the implementation of NFC. This is made possible by storing the relevant banking information and then transferring it to the NFC reader when making a payment. The phone and card machine communicate over a secure radio which is also authenticated by a biometric like a face or fingerprint scan, or secure pin, to exchange details and make the payment possible and keep it secure (Shobha et al, 2016). This is also backed up by Staxpayments.com (2023) as they explain that card data is encrypted when being transferred and for extra security “*the transaction information is relayed to the NFC terminal with a randomly generated one-time use code which is sent to the merchant instead of the customer’s debit card*”.

2.3.3 Item Tracking

Mass item storage is very common in most business practices and this storage will be managed using a barcode system. Each item will have a barcode that identifies that item and records the quantity of each item. This can also be achieved through using NFC tags instead of barcodes however rather than needing to scan a physical marker, that could get damaged and warped, the receiver is simply brought near the item, and it will communicate with the tag to receive the information about the item. (Thanapal, et al. 2017) States that using an NFC item tracking system will reduce human error and increase the efficiency of the workforce. This conclusion was made from a survey they conducted that shows that 62% of participants would swap to an NFC based item storage system, as well as 54% saying that the reason for swapping is due to the benefits NFC offers. Some participants also mentioned that NFCs increased security options was the reason for implementing this solution. However, they also state that participants did show distain to switching from a barcode system due to its incredibly cheap deployment over NFC tags. This indicates that the size of the item storage system required by the business plays a role in whether the business should deploy the change as the cost may outweigh the benefits that NFC can deliver.

2.3.4 Authentication

Typically, NFC tags are used for identification but there are NFC authentication tags that work differently. NFC identification tags will hold information about the tag and how to communicate with it whereas authentication tags are encrypted with a secret key, communication can only commence on the tag once the key has been matched and will not happen until then. Therefore, any receiver without the key will not be permitted to access the data it holds. A blog by Seritag, (2023), explains identification to be the ability to identify the model of a handbag but you cannot be certain that the information given by the tag is actually information about that specific handbag. Whereas authentication is the ability to add that extra layer of certification to ensure that what it says is legitimately what it is.

Seritag, (2023), also explains that “*when the NFC tag is scanned, the NFC chip within the NFC tag performs an encryption calculation based on two elements - the scan count (how many times the tag has been scanned) and the key.*” This combination creates the unique key needed by the tag meaning that by knowing one of the two elements needed, it would still not be possible to generate the access key needed. These encryptions are also not saved to storage which eliminate any security risks of these keys being visible.

ATM machines are a real-world use case of how NFC tags are used for authentication. (Mahansaria, et al. 2017) conducted research on secure authentication in ATM transfers which explained that with the implementation of NFC to complete these transactions, it would bypass the known issues of ATM machines being “skimming”. This is where the user’s PIN is copied once it is input at the machine, the card details are also copied meaning the card can be authenticated to make transactions that were not authorised by the owner of the card. To get around this, an NFC tag with the user’s encrypted details will be tapped to the receiver, eliminating the need to physically input a secret key that should be kept secure. An NFC tag doesn’t mean it’s impossible to retrieve the data being transferred however unlike a card and PIN, the sensitive information is encrypted so the attacker trying to steal this data would only have access to the data after it’s been encrypted. Making

the data useless to the attacker as they do not know the key necessary to decrypt it. Proving that with the same amount of authentication steps required by the user, the implementation of nfc can more greatly protect the user and also provide a more rigorous method of authenticating the user.

2.3.5 Automation

Due to the low cost of nfc tags accompanied by the large accessibility to smart phones equipped with nfc technology, its becoming very popular to automate tasks and applications around a house and not just in business practices. Looking at a blog post about home automation, (Canny, 2024) states a list of use cases where nfc can automate actions inside the house. These include:

- Light switches
- Home heating
- Door locks
- Wifi connection
- Music

By linking these actions to a smart system in the house, nfc tags can be placed in convenient places and then once they are scanned, they will transfer data holding the request to interact with the specific smart action. Some features can be accessed without a need of a smart system, the password to the wifi can be added to a tag near the entrance meaning when a guest enters, they can scan the tag and be connected straight away. Multiple tags can be used for a guest wifi and a main wifi, allowing for the option to connect to either much quicker than obtaining the password and inputting it manually.

NFC tags also have the ability to automate complex business operations even though nfc technology is very simple. Looking at a study by Oliveri, et al. (2024) called “An NFC application for the process mapping automation for SMEs” they looked into the implementation of an nfc-tracer toolkit. Its purpose is to map and measure the performance of manufacturing productions and the paper concluded that *“It could enable not only experts but also nontechnical personnel to immediately visualize the process flow of each individual product identified by the tag... Additionally, by utilizing an electronic spreadsheet (like Excel), many process information such as cycle times, transit times, and the duration of each phase could be evaluated, allowing for a clear understanding of the operations within the company”*. Showing how nfc can improve the processes in a business by reducing potential errors and increasing productivity by managing the process as well as providing performance data for analysis.

2.4 Customer Reward Systems

A customer reward system is a program put in place by a business with the intention to increase customer engagement and customer purchases in return for points that accumulate into discounts, (BigCommerce, 2023). These programs are put in place to benefit both the customer and the business, as the customer can save money on certain products and work towards a goal of getting a discount voucher while the business can increase their volume of sales and incentivise customers to buy stock that isn't as popular.

BigCommerce (2023), also state that there are 3 types of reward programs, each with their own benefits and methods of increasing their customer retention. These programs are a referral, points, and VIP programs. Referral programs incentivise customers to encourage other people, who do not shop with the business, to sign up to a program and in return both customers get a type of discount. This increases the business's total customer base, which in turn will increase their total sales through customers choosing to shop with this business when they previously would not have. A points program is essentially where items are worth points respective to their price, customers can collect these points and then use them to obtain specific deals based on their purchase history. This encourages customers to purchase items they did not initially intend to, as it would provide them with more points than usual therefore making it appear as the value of the deal is larger. This

program will also help increase customer loyalty, meaning that customer is more likely to return to the business after signing onto the program (Rizvi, 2021). Both these programs are not independent and can be used in unison for higher effectiveness.

A survey from Yotpo (Polkes, 2024) called *The state of Brand Loyalty 2021* gives insight into the opinions of customers in regards to their shopping trends with reward systems.

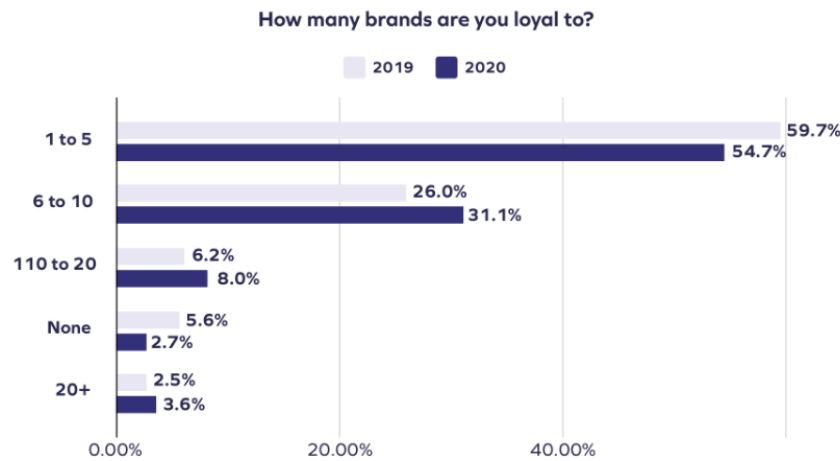


Figure 3: Survey results regarding brand loyalty. Source: Yotpo

Looking at the results of this question, in 2019 only 5.6% of participants did not have any brand loyalty, showing the impact of the reward programs. This is emphasised by that amount decreasing to 2.7% in 2020. This data also shows that after a year, more participants had between 5-10 brands rather than 1-5, implying that more brands are implementing these systems and are able to keep these customers loyal as customers with more than 10 brands only include <10% of participants.



Figure 4: survey results regarding customers habits towards loyal brands. Source: Yotpo

This graph shows the importance of building brand loyalty, as in 2019 roughly 60% of participants are willing to sign up to a brands reward system if they have loyalty for the brand. This increased by 7% by 2020. The biggest impact for brand loyalty is the increase of participants willing to shop with the brand even when the prices may be cheaper elsewhere. This increased by 20% in a year and reinforces the point that increasing the customers engagement will benefit the business.

2.4.1 databases

For a customer reward system to function, a database is required to store customer information as well as for stock management. There are many options for database management software that all provide different functionality to achieve the same goal. Software will have pros and cons based on the use case required for the database meaning its important to assess them before implementation.

Looking at a blog post by Appinventiv (Singh, 2024), they conducted a survey on Stack Overflow (coding forum) developers asking what they think the best database for web development is. It is apparent by this data that a database that uses Structured Query Language (SQL) is the best choice due to it being chosen by the majority. It also shows that the most popular database that is not run on SQL is MongoDB. Mongo runs on a modified version of SQL called MQL which stands for Mongoddb Query Language. This might be why MongoDB is so popular as MQL is “designed for easy use by developers” (MongoDB, 2024)

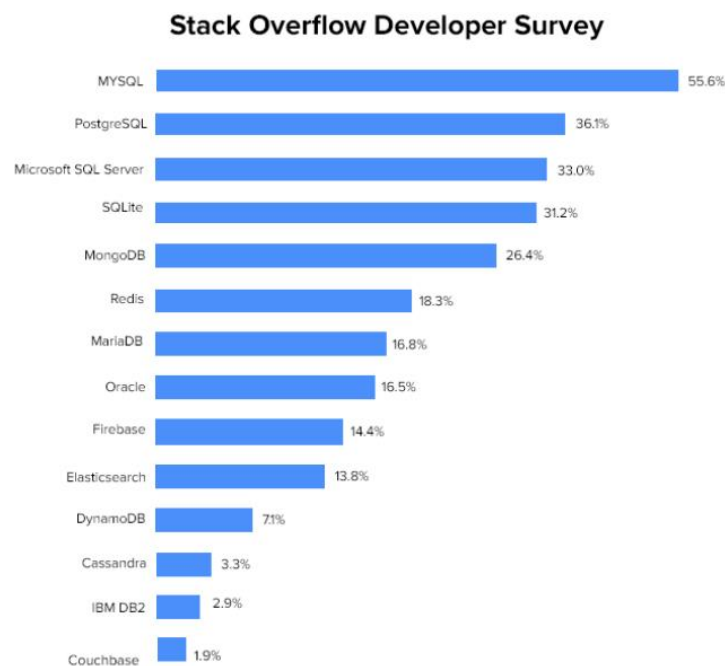


Figure 5: results of database choice by stack overflow developers. Source: Appinventiv

2.5 Gaps in the Literature

While there is a lot of research supporting the technologies that are used and related to implementing a reward system, it was evident while researching that there is less research providing insight into the effectiveness of the implementation of reward systems. Blog posts have been made about this however they do not provide evidence or sources to relating research, meaning it is hard to trust the legitimacy of the information. Due to the technologies needed being unchanged fundamentally for years, the research behind it has been present for enough time to ensure that the legitimacy and accuracy meet the required standards, unlike research into the effectiveness of reward systems.

Having a plethora of resources around the technologies while having less research into the effectiveness are not issues to the project. In fact, it creates an opportunity to take and build on what has already been published and add to it with my own findings and research. These gaps in the literature will not hinder the production of the project but will allow the project to fill these gaps and therefore increase the knowledge of reward systems.

Chapter 3: Project Methodology

This chapter will outline the methods of research and development of the project and how they will be implemented. Exploring the reasons and benefits of choosing these methods and how the project succeeds because of it. When creating a customer reward system, it is important to gather research that can aid in its development to allow the system to achieve the user requirements. This is why a research methodology and software development life cycle methodology will be implemented as this gives a clear structure and approach to achieve the goals set out.

3.1 Approach

Being able to investigate different methodologies and their benefits is essential to extending the lifespan of the project. Creating a clear path for the project to follow which allows for increased productivity by eliminating confusion and implementing a frame for the project. This is shown through my choice of quantitative research with the use of survey questions and pairing this with the development method of RAD otherwise known as Rapid Application Development. This is part of the Agile methodology which focuses on prototyping and reviewing the application until it meets the requirements. Being able to combine this with a dataset of opinions related to the application will allow me to create prototypes that focus on implementing features that the majority of users would be likely to use.

3.2 Project Methodology and Methods

Methodologies and methods are management guideline tools that are used as a structure for a project. Different methodologies will be more effective at increasing productivity than others so researching into them with a predefined scope of the project is important. This is because a methodology may seem effective in theory but its methods of approach to tasks do not fit the style and flow of the project, leading to delays rather than efficiency.

Looking at two blog posts from UpTech (Nikolaieva, 2024), and TatvaSoft (Sharma, 2020), they both agree on the most successful software development methodologies are Waterfall, Scrum, and Agile. I looked into these methodologies and what methods they use to increase efficiency and applied this to the scope of my project.

3.2.1 Agile Methodology

Agile was the methodology chosen to follow for this project due to its priority to deliver a project with its required features within a set timeframe and budget. This relates to my project as it had a definitive deadline as well as key features that are needed, showing how Agile was the best choice.

Traditionally Agile is implemented for development teams rather than individuals however the methods can be extracted and applied to this project in the same way. This is because Agile's main aim is flexibility as well as adaptability and how well a project can continuously improve with minimum down time. This is why prototypes are so effective as it creates the opportunity to implement without complication as small improvements are made and debugged before moving on to the next step.

When comparing Agile to other methodologies, it becomes apparent that making changes during the development process is increasingly more difficult. An example of this is Waterfall, another extremely popular methodology for project development, where it focuses on a more prolonged and rigid planning stage that covers everything that could come up before development starts and plans how to overcome it. This method is effective but may lead to a longer development time as unforeseen and unaccounted for issues may arise and put a stop in the production of the project. With the environment and requirements of my project, waterfall was not a suitable methodology to follow that would produce the best outcome possible.

3.2.2 Method

Rapid Application Development, RAD, is a method of agile that focuses on the techniques needed to enable constant and regular analysis of the progress of the project. Constant analysis of progress allows for many opportunities to make changes before moving to the next task, eliminating the need to review and tweak previous work as it is already at the required specification.

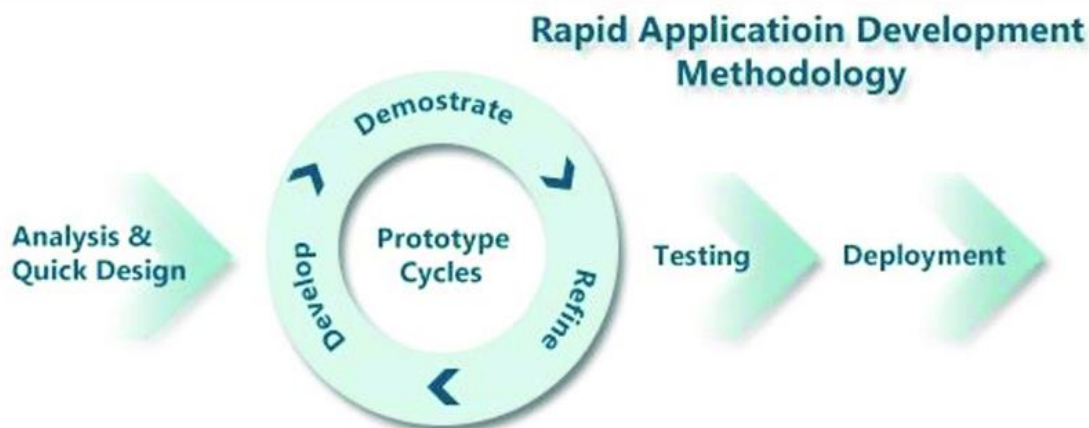


Figure 6: RAD methodology cycle diagram (Ghades et al 2015)

This figure shows the development cycle process that a project will take on, where a prototype of each task will be developed and tested before the development team is able to continue. This eliminates potential issues that can appear as well as creating a sturdy foundation that is worked on.

The periods of time where the project is being reviewed and tested may seem like progress is not being made; however, being able to create a better working environment will positively impact the progression of the project as it will reduce the amount of unwanted stoppages, which can extend the development period.

Applying this to my project, both the front end and the back end went through multiple stages of prototypes to successfully implement all the features that I found necessary for the project to succeed. These two sections of the project need to work in unison for the functionality of the reward system to be present, meaning working towards a final system entailed reviewing one end of the system while the other prototype is working as intended. However, after reviewing one end, this outlined changes and features needed elsewhere in the system. This is where following the RAD method was crucial to the project as this type of development heavily relied on the back and forth eventually leading to the successful final system.

3.2.3 User Experience

The analysis and design period of RAD outlines a section of the project dedicated to creating diagrams and visuals to help understand the flow of the interface and environment the user will be interacting with. From looking at Figure 1: use case diagram, an entity relationship diagram (ERD) can be created from this. An erd is a diagram of the tables “entities” included in the system, and inside these tables will be the data classes and data types. The relationships are defined by the links between the entities, representing how these tables will link together using a “key” from the tables. A key is a data class that relates to another class with identical data requirements.

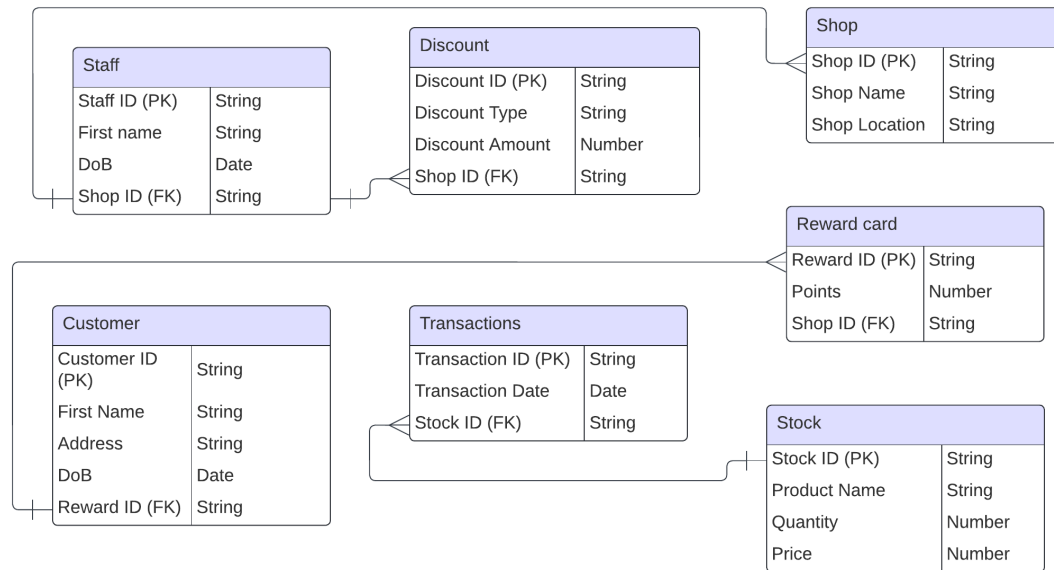


Figure 7: Entity Relationship diagram displaying tables and actions, with primary and foreign keys needed to relate tables.

These diagrams only map out the backend of the system so to understand what a user should see when interacting with the system, a wireframe can be created. Wireframes are illustrations of each webpage that include key sections, headings, and features required. Deciding on where elements should be located on a page can bring light to the flow of the system, it may become apparent when designing these wireframes that initial intentions for the system would not be beneficial and changes can be made before development starts. Helping to outline another advantage of using wireframes which is being able to increase development efficiency as naturally the changes and requirements needed to achieve the desired outcome will be made rather than having to review the code and make changes afterwards.

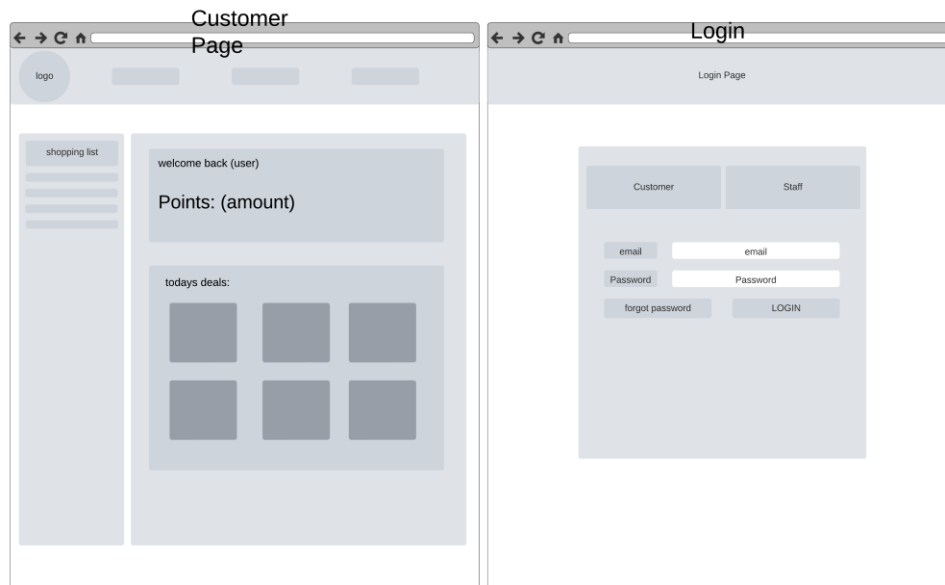


Figure 8: Wireframe diagram of desired login and customer page layout

3.3 Research Approach

Using past research provides insight into a project however this does not add to the total contribution of research. For this report to be involved in the further development of nfc technology and reward systems research, I will need to conduct research of my own that provides data and insight that can be used to answer the research questions of this project. There are two types of research that can be conducted, quantitative and qualitative. As the names suggest, the former focuses on quantity and the latter focuses on quality of the data collected. Quantitative is used for collecting opinions represented as a number, that can be used for creating graphs and predictions. These responses are easy to collect in large quantities but will include less accurate opinions. Qualitative on the other hand is used for collecting thoughtful feedback through communication, the questions will be more in depth therefore meaning the answers will need more thought and justification. These responses are tougher to collect in large quantities due to the time required to do so, but each response will hold more value in understanding the opinions of participants.

3.3.1 Quantitative Approach using a Survey

According to Grand Canyon University (2021), using a questionnaire is the most effective way to collect quantitative data and thus the reason why I will be using a questionnaire for my project. There are multiple types of questions that can be included, and I chose to use closed ended likert style questions. Likert is a 5 part scale used to indicate the position of an individual's opinion. An example of the scale is strongly disagree to strongly agree with a middle option of I'm not sure. This is important as it leads the individual to sway towards one side or the other whilst also having an option to stay in the middle if they are unsure. This helps with accuracy of data as; it's less likely for someone to change their real opinion to fit into what they think will be the average and there will be less responses that agree/disagree to a statement when in reality they do not.

Having responses on a scale also creates the option to conduct analysis through statistics and graphs, helping to visually understand the spread of data. These visualisations can then be used in combination with the development stage to better understand the importance of features and functionality, assisting in the delegation of time and resources to produce a system that will appeal to the highest number of users possible.

3.4 Challenges and Limitations

3.4.1 Implementation

The biggest challenge for my reward system implementation is accurately replicating the environment that a small business would be working in and how this system would integrate into that. As the project had a time constraint, it led me to extract the actions and use cases from a small business and create working demonstrations of how they would work in regards to my system.

Another limitation of this implementation is the cost of hosting a database and a domain for public use. As a demonstration the reward system works locally however to open it up to customers would create an immediate cost from subscribing to a platform as a service from a cloud service provider. This cost would then scale with the increase in users as both the domain and the database would need to accommodate the growing amount of traffic and data respectively.

3.4.2 Research

Even though quantitative research through questionnaires provides many attributes that are beneficial to the project, it still has its disadvantages that can affect the data collected as it does not provide the option to answer with explanation and feedback to the questions. These open ended responses can help with resolving specific issues and queries whereas showing agreement on a scale does not give this same option.

3.5 Ethical Considerations

When gathering opinions, it's important to ensure that it is not possible to identify an individual through visualisation and analysis of the data. This is why participants in my questionnaire will be anonymous and no sensitive data will be asked to retain their anonymity and increase the likelihood of someone participating in the survey. The database used for this project will not include customer information of real humans, all data of customers, staff, and products will be fabricated for the sole purpose of demonstration of the system. This choice was made to avoid any potential risks or implications with copyright or right to use for my own personal gain. Therefore, there are no ethical implications for this project that need to be assessed and managed.

3.6 Conclusion

This chapter outlined that the research method that I will be using is the Quantitative method, using a survey, and the methodology that will be followed to help increase efficiency is Agile. As well as this, RAD will be used to help improve the production speed of the implementation development. The limitations this project face were also outlined so that I can find a way to work around them and therefore provide insight into alternative methods of approach to these problems.

Chapter 4: Implementation

Like any project following the RAD method should, the system faced changes and redesigns along the development process to help accommodate the ability to implement all the features that was deemed essential to have an operational reward system. The database for the system was hosted on MongoDB and accessed through MongoDBCompass, the server for the system runs on Node.js and utilises compatible node modules for better features and functionality. The node package Mongoose was used to link the database to the server through a port, being 9999. The reward system is accessed through an internet browser at localhost using that port due to it being ran locally off my personal computer however the system can be hosted to a domain to allow for public assess.

4.1 Backend

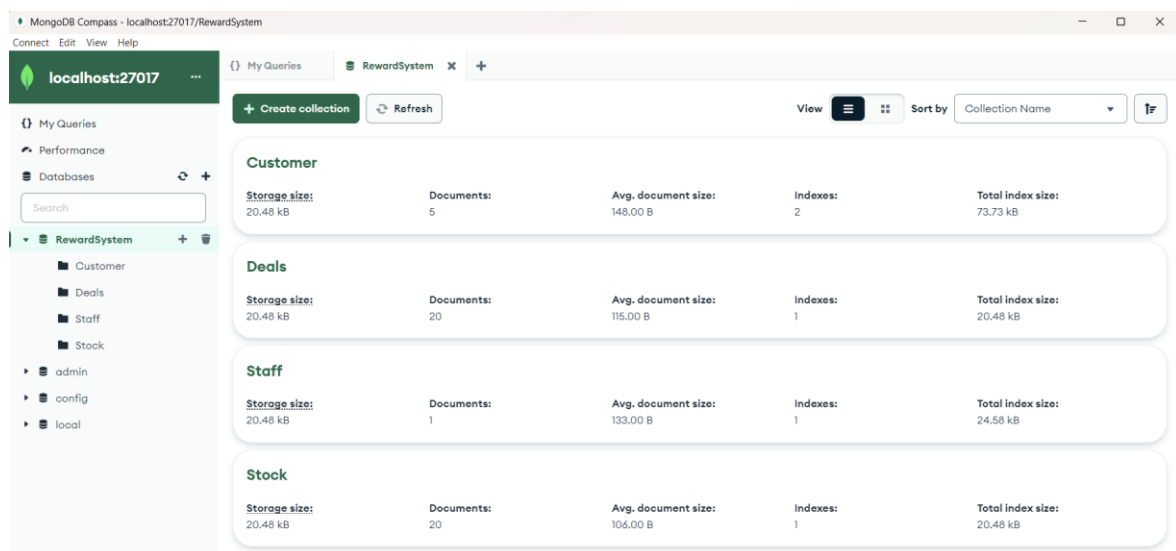


Figure 9: Using MongoDB Compass as an interface to access and modify the data for the reward system. Connection made on port 27017

The database has collections with fabricated data that is used and manipulated by the front end to show the functionality of the reward system. The database is now live and ready to be connected to the server that handles all the requests for the system. Here it will be linked to the rest of the project using a node package called “mongoose”. Mongoose is used to build the bridge between the database and the server so that data within the database can be accessed and changed in real time on the server.

```
const mongoose = require("mongoose");

const localDB = 'mongodb://localhost:27017/RewardSystem';

const connectDB = async () => {
  await mongoose.connect(localDB, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  });

  console.log("MongoDB Connected");
};

module.exports = connectDB;
```

Figure 10: code inside db.js file that handles the connection to the Mongodb database

```
ireVersion":21,"maxWireVersion":21},"outgoing":{"minWireVersion":21,"maxWireVersion":21},"isInternalClient":true}}
{"t":{"$date":"2024-04-29T19:20:53.675+01:00"},"s":"I", "c":"REPL", "id":5853300, "ctx":"initandlisten","msg":"curr
ent featureCompatibilityVersion value","attr":{"featureCompatibilityVersion":"7.0","context":"startup"}}
{"t":{"$date":"2024-04-29T19:20:53.677+01:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg":"Clea
ring temp directory"}
{"t":{"$date":"2024-04-29T19:20:53.683+01:00"},"s":"I", "c":"CONTROL", "id":6608200, "ctx":"initandlisten","msg":"Init
ializing cluster server parameters from disk"}
{"t":{"$date":"2024-04-29T19:20:53.684+01:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg":"Flow
Control is enabled on this deployment"}
{"t":{"$date":"2024-04-29T19:20:54.098+01:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Init
ializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2024-04-29T19:20:54.133+01:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Sett
ing new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2024-04-29T19:20:54.133+01:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Time
stamp monitor starting"}
{"t":{"$date":"2024-04-29T19:20:54.137+01:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening
on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2024-04-29T19:20:54.137+01:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting f
or connections","attr":{"port":27017,"ssl":"off"}}
{"t":{"$date":"2024-04-29T19:20:54.138+01:00"},"s":"I", "c":"CONTROL", "id":8423403, "ctx":"initandlisten","msg":"mong
od startup complete","attr":{"Summary of time elapsed":{"Startup from clean shutdown?":true,"Statistics":{"Transport lay
er setup":"0 ms","Run initial syncer crash recovery":"0 ms","Create storage engine lock file in the data directory":"0 m
s","Get metadata describing storage engine":"0 ms","Validate options in metadata against current startup options":"0 ms"
,"Create storage engine":"335 ms","Write current PID to file":"0 ms","Initialize FCV before rebuilding indexes":"24 ms",
"Drop abandoned idents and get back indexes that need to be rebuilt or builds that need to be restarted":"1 ms","Rebuild
indexes for collections":"0 ms","Load cluster parameters from disk for a standalone":"1 ms","Build user and roles graph
":"0 ms","Set up the background thread pool responsible for waiting for opTimes to be majority committed":"3 ms","Initia
lize information needed to make a mongod instance shard aware":"0 ms","Start up the replication coordinator":"8 ms","Sta
rt transport layer":"0 ms","_initAndListen total elapsed time":"830 ms"}}}}}
```

Figure 11: screenshot of the live database running, the server wouldn't be able to communicate without this running.

The database is now connected to the server, now some models need to be created to assist the server in fetching and sending data between them. These models will be an outline to how data will be packaged and sent so it can be interpreted and displayed in the reward system.

```

rewardsystem > model > JS User.js > ...
1  const Mongoose = require("mongoose");
2
3  const UserSchema = new Mongoose.Schema({
4    username: {
5      type: String,
6      unique: true,
7      required: true,
8    },
9    password: {
10     type: String,
11     minlength: 6,
12     required: true,
13   },
14   role: {
15     type: String,
16     default: "Basic",
17     required: true,
18   },
19   points: {
20     type: Number,
21     default: 500,
22     required: true,
23   },
24 }, { collection: 'customer' });
25
26 const User = Mongoose.model("User", UserSchema);
27
28 module.exports = User;
29

```

Figure 12: model for customer data

```

rewardsystem > model > JS StockItem.js > ...
1  const Mongoose = require("mongoose");
2
3  const StockItemSchema = new Mongoose.Schema({
4    name: {
5      type: String,
6      required: true
7    },
8    points: {
9      type: String,
10     required: true
11   },
12   price: {
13     type: Number,
14     required: true
15   },
16   quantity: {
17     type: Number,
18     required: true
19   },
20   stockId: {
21     type: Number,
22     required: false
23   },
24 }, { collection: 'stock' });
25
26 const Stock = Mongoose.model("stock", StockItemSchema);
27
28 module.exports = Stock;
29

```

Figure 13: model for stock data

Now that the database and server can communicate, its time to design the server.js file, where post and get requests are made so that the correct data and pages are displayed. Any page that needs to be accessed will have its requests and responses defined here.

```

app.get('/newuser', async (req, res) => {
  try {
    const allStock = await Stock.find({});
    const randomStock = await Deal.aggregate([ { $sample: { size: 4 } } ]);
    res.render('newuser', { allStock: allStock, randomStock: randomStock });
  } catch (err) {
    res.status(500).send(err.message);
  }
});

```

Figure 14: Example of GET request in server.js file.

To authenticate the users when they access the system I used a node package called JWT which stands for jsonwebtoken along side bcryptjs. This packages collect the user information in json format and and signs it with a special signature, being the key, to ensure that the data is not accessed until it reaches its destination and can be decrypted. This data is stored as a “token” and this token can be given an age, meaning after the token reaches its maximum age, it will be deleted from the browser and further increasing the security of the system. These packages also handle the authentication of a user when they login, as it will check whether the account matches the credentials to be an admin and if not, by default the basic role is given to the account.

This is achieved by creating a middleware, which stated as required in the server.js file, that acts as a “middle man” that will perform checks by first looking at the users role, based on this it will check the user credentials it received and match them to the browser token. If the results match then the user can enter the site, if the credentials do not match, the autherisation is denied.

The system also has two other methods of protecting the users data with including a maximum age for the token, and the user passwords are encrypted. The token created has been given a maximum time that it can be active and used for, this time was set to 3 hours but can be changed to fit the requirements. Having the token expire means that a users token cannot be used to unconcentially access a users account as the credentials used to validate the user will no longer be validated. On

top of this, the users passwords are encrypted using a jwt secret, this secret is unknown to everyone so if a user was to get ahold of a password, they could not decrypt it.

```
app.use("/api/auth", require("./Auth/route"));
```

Figure 15: Code line in server.js that mounts the middleware

```
rewardsystem > middleware > JS authjs > ...
1  const jwt = require("jsonwebtoken");
2  const jwtSecret =
3    "4715aed3c946f7b0a38e6b534a9583628d84e96d10fbc04700770d572af3dce43625dd";
4
5  exports.adminAuth = (req, res, next) => {
6    const token = req.cookies.jwt;
7    if (token) {
8      jwt.verify(token, jwtSecret, (err, decodedToken) => {
9        if (err) {
10          return res.status(401).json({ message: "Not authorized" });
11        } else {
12          if (decodedToken.role !== "admin") {
13            return res.status(401).json({ message: "Not authorized" });
14          } else {
15            next();
16          }
17        }
18      });
19    } else {
20      return res
21        .status(401)
22        .json({ message: "Not authorized, token not available" });
23    }
24  };
25  exports.userAuth = (req, res, next) => {
26    const token = req.cookies.jwt;
27    if (token) {
28      jwt.verify(token, jwtSecret, (err, decodedToken) => {
29        if (err) {
30          return res.status(401).json({ message: "Not authorized" });
31        } else {
32          if (decodedToken.role !== "Basic") {
33            return res.status(401).json({ message: "Not authorized" });
34          } else {
35            next();
36          }
37        }
38      });
39    } else {
40      return res
41        .status(401)
42        .json({ message: "Not authorized, token not available" });
43    }
44  };
```

Figure 16: Middleware code that authenticates users

4.1.1 Frontend

In the use case of the reward system, the first step is a customer registering to the system so this was the first page implemented and became the first prototype of the system. For a user to register they need to input their name as well as a password. The system may have security measures in place, but to eliminate the chance of exposing sensitive data, all customer account data is fabricated for the use of this project. When a customer registers, their data will be saved to the database in the Customer collection.

```
_id: ObjectId('661fe312571620cb89abe15b')
username: "Bob"
password: "$2a$10$T3ptgHpXIBVoKRJ7zBxhVeGEAiumNG4iQME0gRT1X/.Q4mJVuZqai"
role: "Basic"
```

Figure 17: Customer record in MongoDB, including encrypted password

When a customer enters the site, before they are able to register, they will be met with the home screen. This is defined under the “/” get request and renders the home page, this page will have links to either registering or logging in and the register page will link to the log in page to eliminate having to keep going back and forth between the pages.

```

rewardsystem > JS server.js > ...
1 //models and routes needed
2 const express = require("express");
3 const connectDB = require("./db");
4 const app = express();
5 const cookieParser = require("cookie-parser");
6 const { adminAuth, userAuth } = require("./middleware/auth.js");
7 const Stock = require("./model/StockItem");
8 const Deal = require("./model/Deal");
9 const User = require("./model/User");
10 const cors = require("cors");
11
12 //localhost port
13 const PORT = 9999;
14 //engine for views
15 app.set("view engine", "ejs");
16
17 connectDB();
18 //packaging moving and displaying data
19 app.use(express.json());
20 app.use(cookieParser());
21 app.use(express.static("public"));
22
23 //routes
24 app.use("/api/auth", require("./Auth/route"));
25
26 //pages
27 app.get("/", (req, res) => res.render("home"));
28 app.get("/user1", async (req, res) => {
29   try {
30     //fetch all stock
31     const allStock = await Stock.find();
32
33     // random stock
34     const randomStock = await Deal.aggregate([{$sample: { size: 4 } }]);
35
36     const user = await User.findOne({ username: 'Andy' });
37     // html template
38     res.render('user1', { allStock: allStock, randomStock: randomStock, points: user.points });
39   } catch (err) {
40     res.status(500).send(err.message);
41   }
42 });
43 app.get("/user2", async (req, res) => {

```

Figure 18: Section of the server.js file. Showing some GET requests.

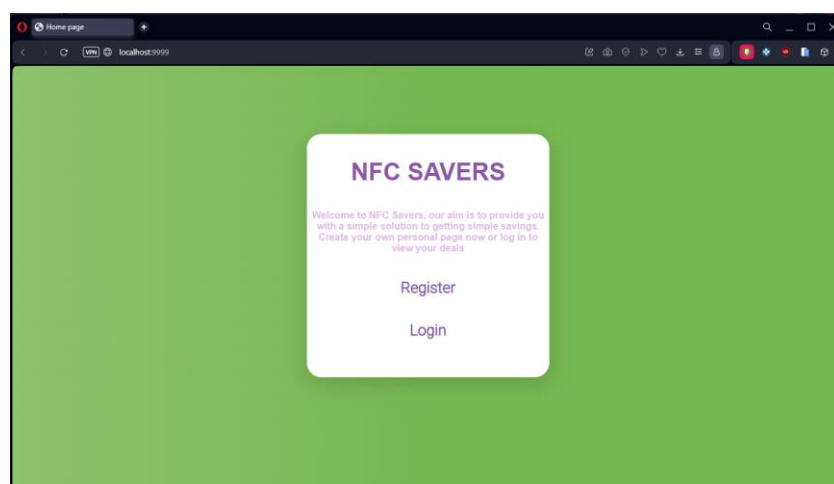


Figure 19: Home page of website, accessed through port 9999

On the home page the user will be met with options for either logging in or registering for a new account. Each of these pages will have a link at the bottom to take the user to the other page, the register page will also redirect the user once a successful registration is created.

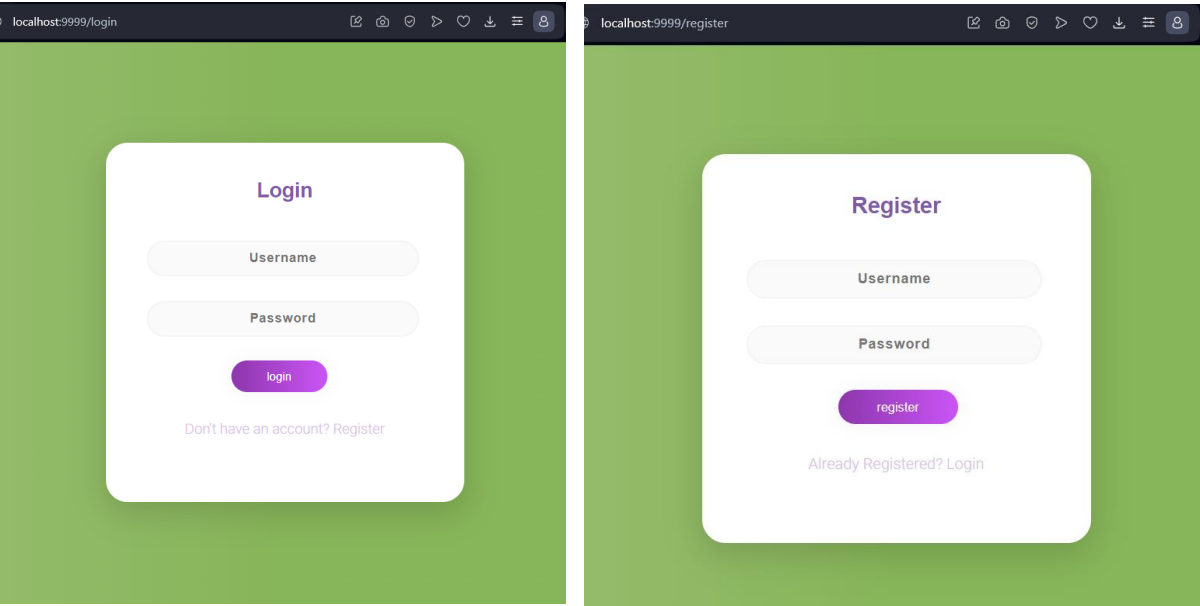


Figure 20, 21: Login and Register Pages

When a customer successfully logs in they are met with their personal page, each customer will have different deals offered to them available to use, their points will be visible and will dynamically update as they change.

NFC SAVERS

Log Out

Welcome, Andy

Points: 500

Check out your special deals as well as our stock, If you want to make a shopping list click [here](#)

Deals

Name	Points	Price £	Quantity	Deal
Butter	2	1.8	50	20% off
Bacon	3	2.5	50	20% off
Biscuits	1	1.3	100	3 for 2
Lemonade	1	0.79	100	3 for 2

Products

Name	Points	Price £	Quantity
Bread	1	1.2	50
Milk	2	1.5	50
Chocolate Bar	1	0.8	100
Apple	1	0.25	50
Banana	1	0.2	50
Grapes	2	2	50
Water	1	1	100
Cola	1	0.99	100

Figure 22: User page of “Andy” who is user 1 of the fabricated customers in the database

Deal	Deal	Deal
20% off	2 for 1	2 for 1
20% off	3 for 2	3 for 2
3 for 2	10% off	10% off
3 for 2	20% off	20% off

Figure 23, 24, 25: Deals offerered to 3 different users

```
<h1 class="right">Points: <%= points %></h1><h1 class="left">Welcome, <%= name %></h1>
```

Figure 26: Points and name are grabbed from the model and passed in the get response to be displayed dynamically to the user, rather than being a static image.

After customers have seen what deals are available, they are able to go to the shopping list page where they can input the items and quantity desired and then save it to an nfc tag. This is different to how it would happen if this system was deployed in a small business, however it is here to show the power of nfc and the use cases it can be applied to. The data on the nfc card doesn't include any of the users information, keeping them secure, and will stay there until the data is required by the admin. This data is also sent to the admin page without being needed to be saved on the nfc card.

This functionality would work on a mobile device, allowing the user to place the card on their phone to transfer the data. That data can now be transferred elsewhere without needing to login or access the reward system, meaning an internet connection is not required and neither is a device to access the system. Increasing the accessibility of the system due to its requirements of largely accessible and transportable technologies like nfc cards.

Product	Quantity	Delete
Banana	1	Delete
Water	4	Delete
Soda	2	Delete

Figure 27: User adding items to their list, option to delete from the list if necessary

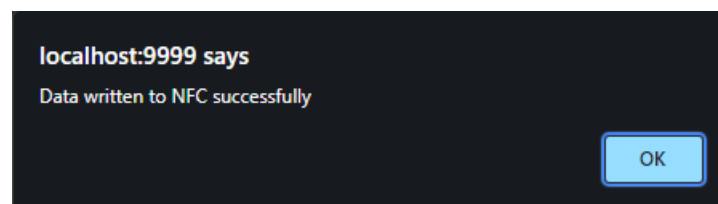


Figure 28: Action on press of save list

```

let shoppingLists = []; // store shopping lists

app.post('/api/shopping-lists', (req, res) => {
  const { list } = req.body;
  shoppingLists.push(list); // store list
  res.status(200).send('List received');
});

//send list
app.get('/api/shopping-lists', (req, res) => {
  res.json({ lists: shoppingLists });
});

```

```

// save function
function saveList() {
  const table = document.getElementById('tbl1');
  let data = '';
  for (let i = 1; i < table.rows.length; i++) {
    const product = table.rows[i].cells[0].innerText;
    const quantity = table.rows[i].cells[1].innerText;
    data += `${product},${quantity}\n`;
  }
}

// nfc write
fetch('/write-nfc', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({ data })
})
.then(response => response.text())
.then(result => {
  alert(result);
  // send data to admin
  return fetch('/api/shopping-lists', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ list: data })
  });
});
}
.catch(error => console.error('Error:', error));

```

Figure 29, 30: code required to save the list and write it to the nfc card



Figure 31: nfc card held to reader to receive data

I created an admin page that theoretically could be used by the shop owner to manage the accounts as well as the lists created, the page can edit the roles of the users as well as delete them, it will also show any lists that have been created

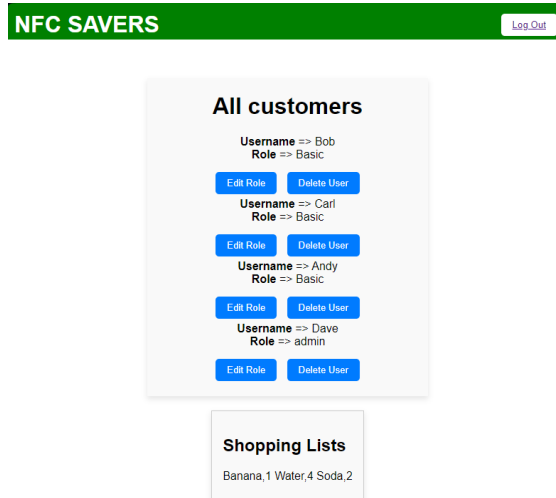


Figure 32: Admin page

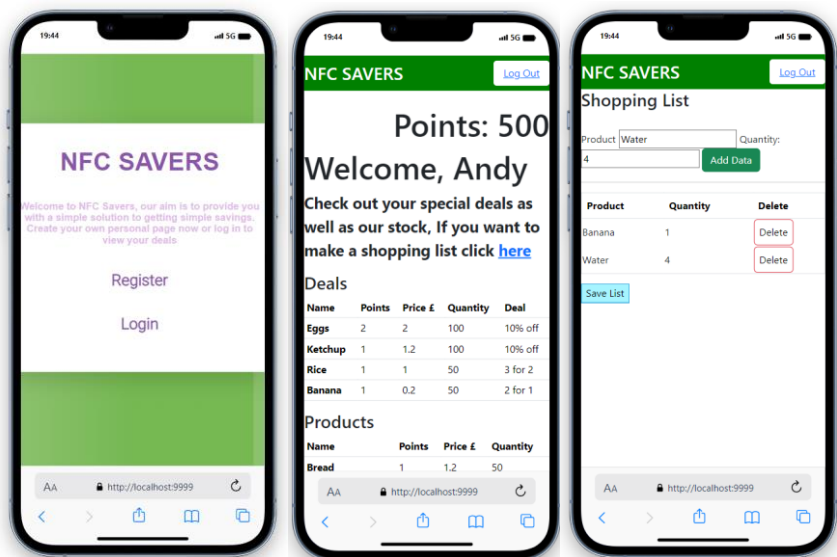


Figure 33, 34, 35: Screenshots of reward system accessed through browser extension emulating a smartphone

It was important to ensure that none of the functionality of the system is lost when accessing it through a device that is not a desktop. This meant that when designing the style of the page in the CSS files, it was important to be aware of the size proportions to ensure content wasn't using unnecessary space, and in turn ensuring that customers can get to the features they need in a smooth and quick process.

Chapter 5: Results

5.1 Survey Results

As I stated before, a survey was conducted as my research approach to gain a better understanding on the viewpoints of individuals in the field of rewards whilst shopping. This survey was published on 28th March 2024 and has been active for 1 month gathering 23 responses in that time. This amount is slightly under the target of the survey however it is still enough to show where the majority of opinions reside.

These responses have been collected and analysed using tools within Microsoft Excel after being imported from Microsoft Forms. Allowing for the creation of diagrams and graphs to better analyse and understand the data to then gain a better view of what features the potential users of the reward system require.

Below is a list of figures, showing the questions asked, the spread of answers and what the answers represent in terms of the system:

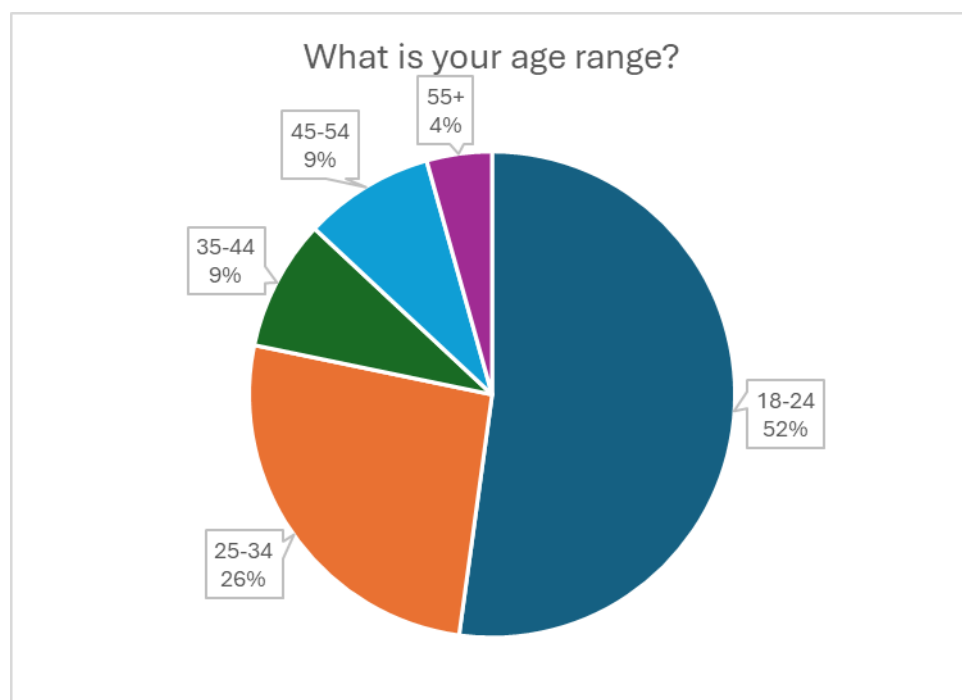


Figure 36: Pie chart of age range percentage

This figure shows over 75% of participants are the target audience of the system.

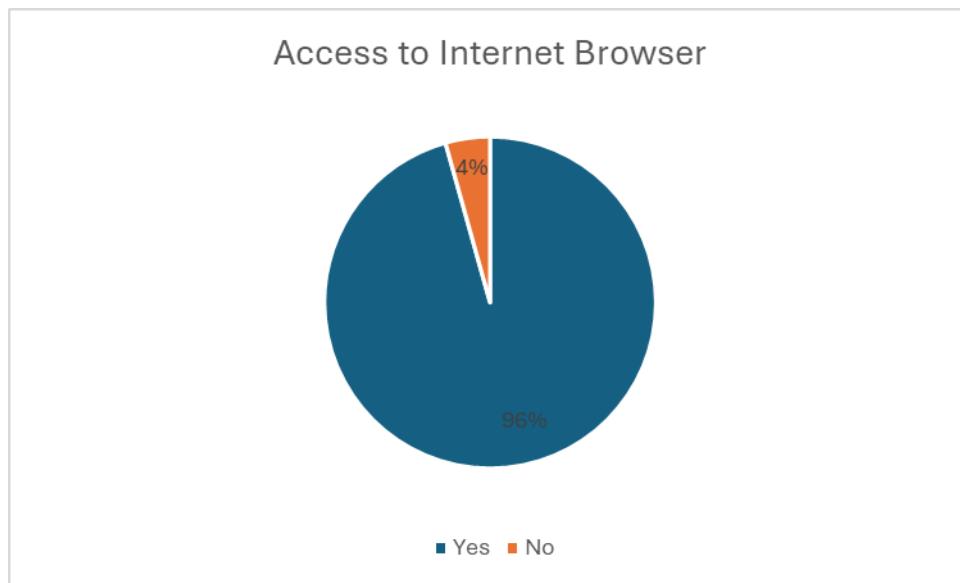


Figure 37: Pie chart of participants access to internet browser

This pie chart showing the large majority of participants having access to an internet browser means that having my reward system accessible through browser helps ensure there are fewer barriers to entry on the system



Figure 38: Bar chart showing frequency of shopping habits

The majority will shop for food one or more time in a week, meaning this system will need to be easily accessible to not detour customers.



Figure 39: Pie chart showing split of using online services to purchase food goods

In this figure there is an even split of participants who implement online shopping into their shopping habits and no one who solely shops online. This indicates that a mobile and portable solution to a reward system is required.

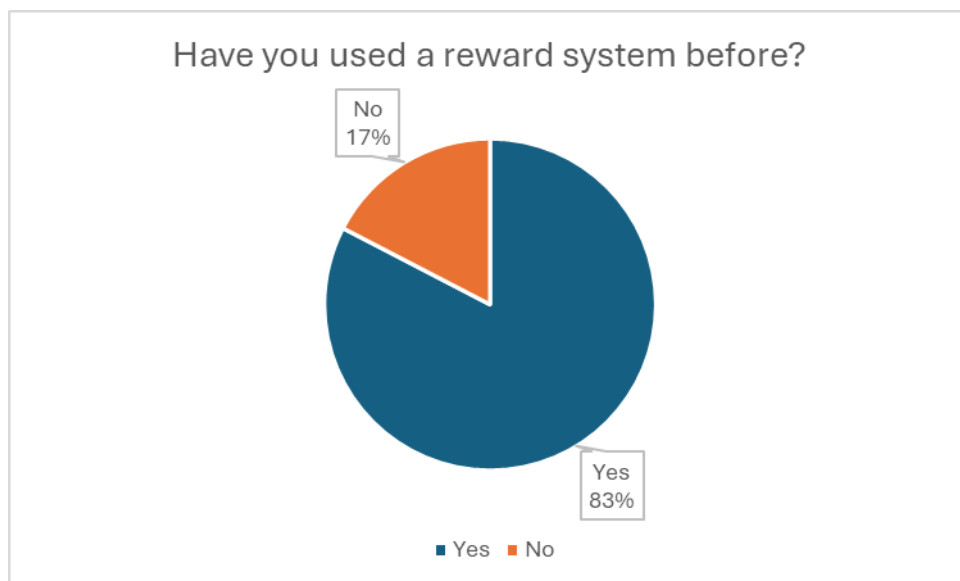


Figure 40: Pie chart of participants who have used another reward system before

From looking at the dataset of the results, participants who answered no also answered options that are not in the majority of other questions, further emphasising that using a reward system is a big part of shopping habits.

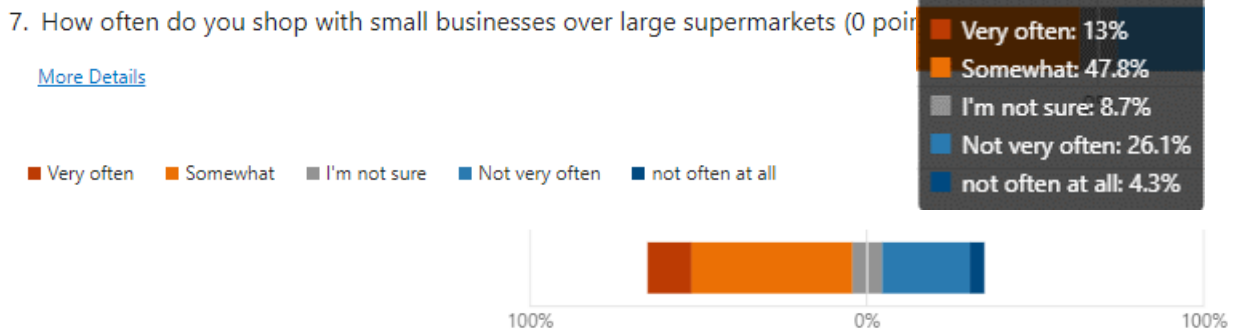


Figure 41: Stacked bar chart of how often participants shop with small businesses

Nearly 50% of participants choose to shop with a small business over a large supermarket somewhat often already, implementing rewards for doing such with a reward system could increase that majority that choose to do this as well as change their frequency to very often.

8. Large supermarkets have better incentives to shop with them over small businesses (0 point)

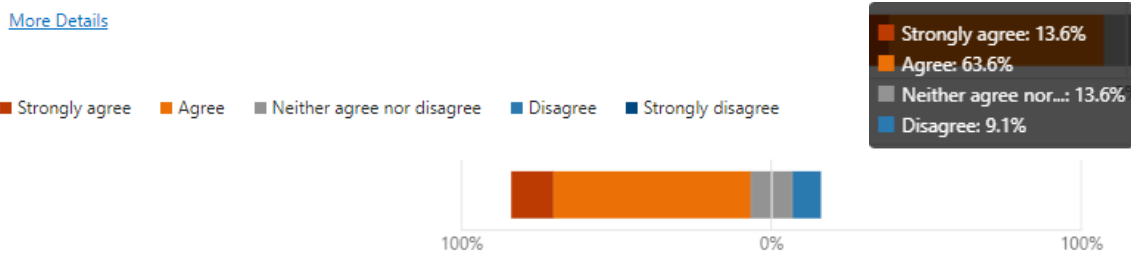


Figure 42: Stacked bar chart showing opinion on whether large supermarkets offer better rewards than small businesses

Over 75% agreed with the statement and yet the majority tend to shop with small businesses more often, this even further reinforces that after implementation of a reward system, frequency and satisfaction from shopping with small businesses would increase.

9. It is important to shop with small businesses (0 point)

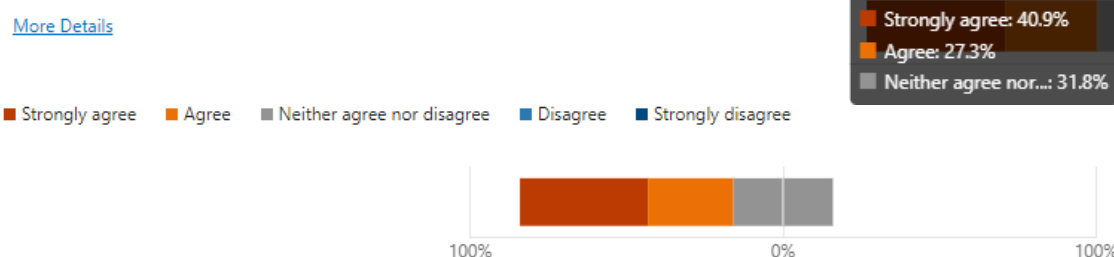


Figure 43: Stacked bar chart showing agreeance to the importance of shopping with small businesses

10. Receiving deals and vouchers for money off will influence my shopping trends (0 point)

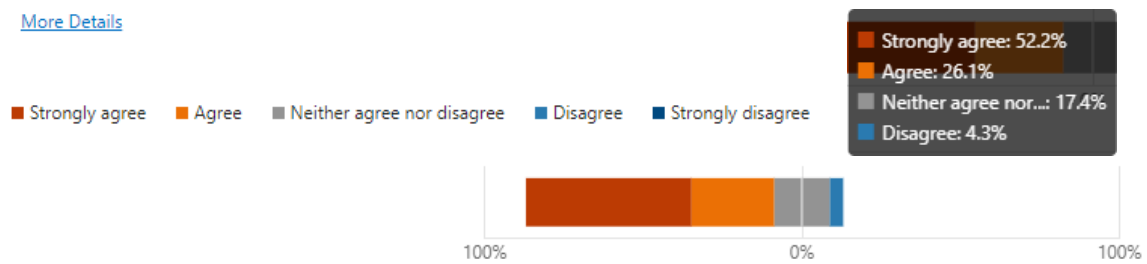
[More Details](#)

Figure 44: Stacked bar chart showing whether receiving deals and vouchers would influence shopping habits.

Figures 34 and 44 show that offering deals to customers could influence their decision to whether they shop with a small or large business.

11. How likely are you to return to a small business that uses a reward system? (0 point)

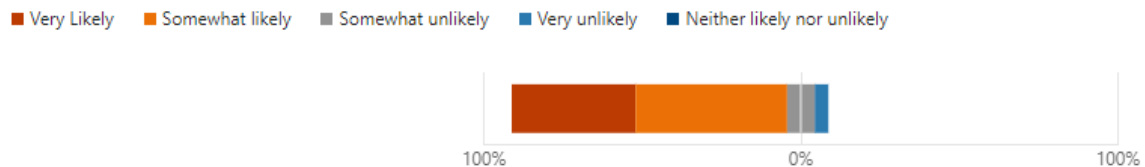
[More Details](#)

Figure 45: Stacked bar chart showing likelihood of participants returning to a small business if they offered rewards

12. How likely are you to subscribe to a reward system? (0 point)

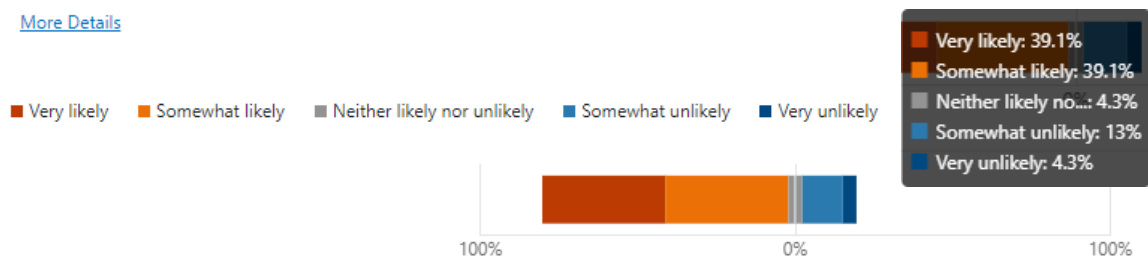
[More Details](#)

Figure 46: Stacked bar chart showing likelihood of participants subscribing to a reward system

13. How likely are you to use a web application that manages purchases and deals? (0 point)

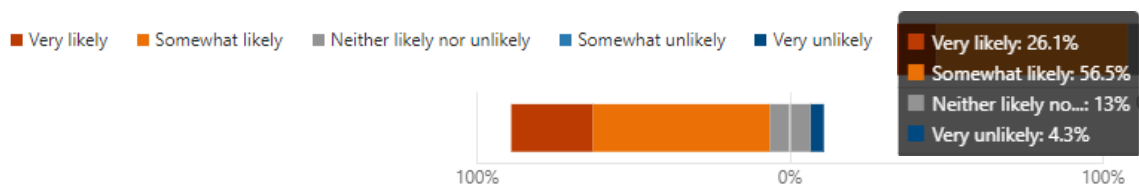
[More Details](#)

Figure 47: stacked bar chart showing likelihood of using web app that manages deals

This shows that almost all of the participants would willing to use the front end of my reward system. It also shows that the system needs functionality of managing deals and purchases

14. how familiar are you with NFC (Near Field Communication) technology (0 point)

[More Details](#)

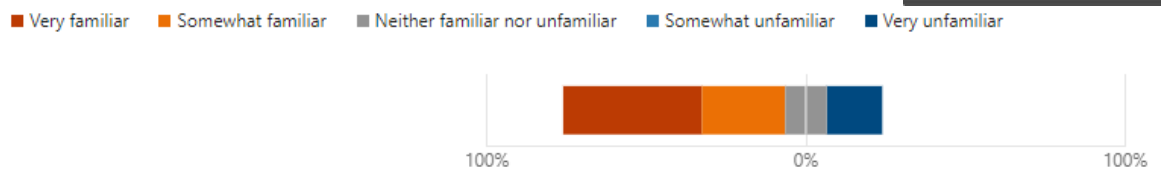


Figure 48: Stacked bar chart showing familiarity of NFC technology

NFC will be utilised in the reward system as the way of storing and transferring each customers unique data about their deals and purchases. Having users already familiar with the technology is helpful for better succession of implementing it.

15. NFC is beneficial to your day to day life (0 point)

[More Details](#)

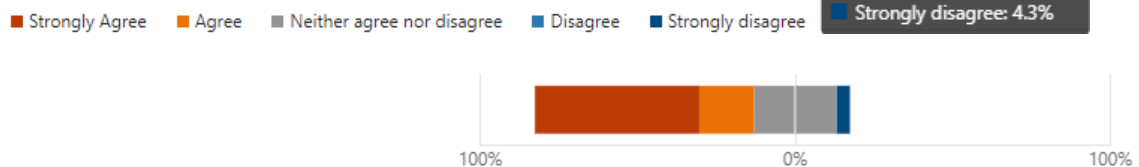


Figure 49: stacked bar chart showing whether participants agree nfc is beneficial to day to day life or not

16. A small business using a reward system with NFC cards would improve your overall shopping experience (0 point)

[More Details](#)

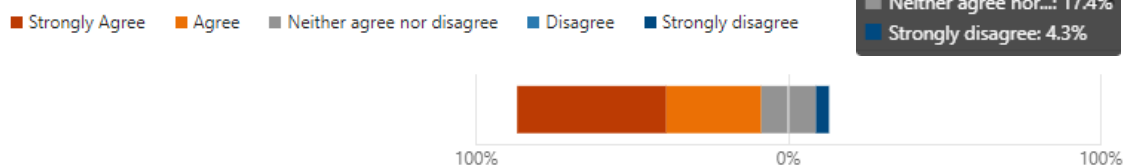


Figure 50: Stacked bar chart showing whether participants agree that small businesses using reward systems would improve shopping experience or not

5.2 Conclusion

To conclude with a summary of the results, the majority of participants agreed with the statements of implementing a reward system and nfc technology into a small business would help increase their retention with the business. These results help answer this projects research questions as the data suggests a clear indication to whether implementing a reward system would increase customer engagement. By reviewing this research its clear to see that the survey achieved its goal of providing further research to supplement my project as well as bringing greater insight into the literature of this project.

Chapter 6: Evaluation

This chapter will allow me to determine the success of the project, by putting the result of the implementation through testing, and by seeing if the research was able to answer its research question. This evaluation is very valuable not only for the lessons on how to improve, but also for carrying these lessons into new projects. Giving me the opportunity to constantly build on my knowledge and skillset and improve myself.

6.1 Implementation Evaluation

When developing a project, it may seem like its behaving as it should however this may not be the case once the implementation is deployed and given the opportunity to be tested by significantly more people. This is why it's good to plan out some tests to check the rigidity of the system, especially as the reward system is designed to hold sensitive customer data. Throughout the development of this project, I would follow a method of testing called smoke testing. This is where the application is run after the implementation of any changes. This is to locate the source of any errors quickly as they come up and outline the area of the project that needs to be adjusted.

As well as this, the reward system should be secure as it will be handing customer information. This is why I added some error checking features within the code. To stop the creation of duplicate accounts, I implemented a check that looks at the user data against records already in the database. If there is a match, the user will not be able to create the account. A similar check is performed when a user is logging in, it will check that there is a record in the database that matches what the user input. The username and password are checked separately too so the system can check whether the username or password is incorrect, allowing the user to amend this quickly. A simple security measure that is important is ensuring a user enters a secure enough password, so the system will check if the password is longer than 6 characters. Looking back at this, I should have also implemented a check for at least 1 special character and a 1 capital letter as this would have made the password more secure.

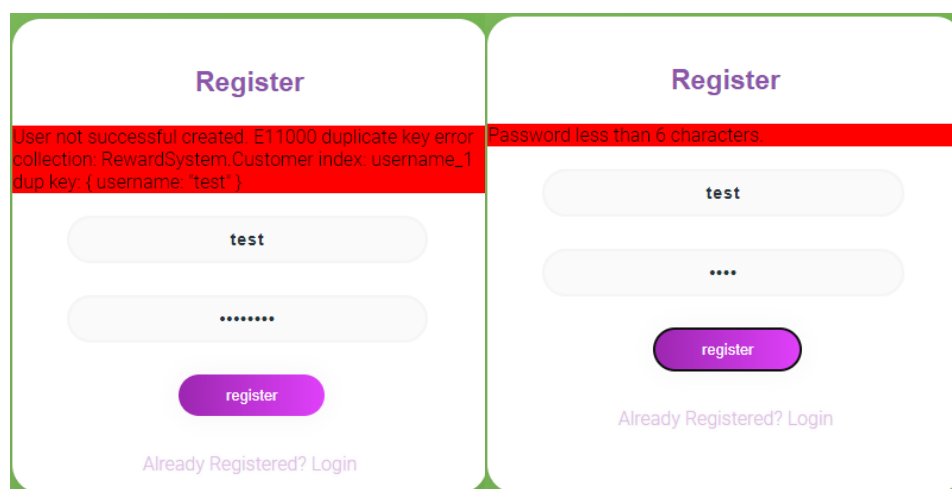


Figure 51: Duplicate record error

Figure 52: Weak password error

```

_id: ObjectId('66300db8e90c111267b86bf4')
username : "test"
password : "$2a$10$MIdMzI/thDhLht1HZlD/u6u2S5XhE/MDVxj.MrFFVhVYnyfSlg4a"
role : "Basic"
points : 500

```

Figure 53: Object of test account made in database. The password was “testtest” showing effect of the key

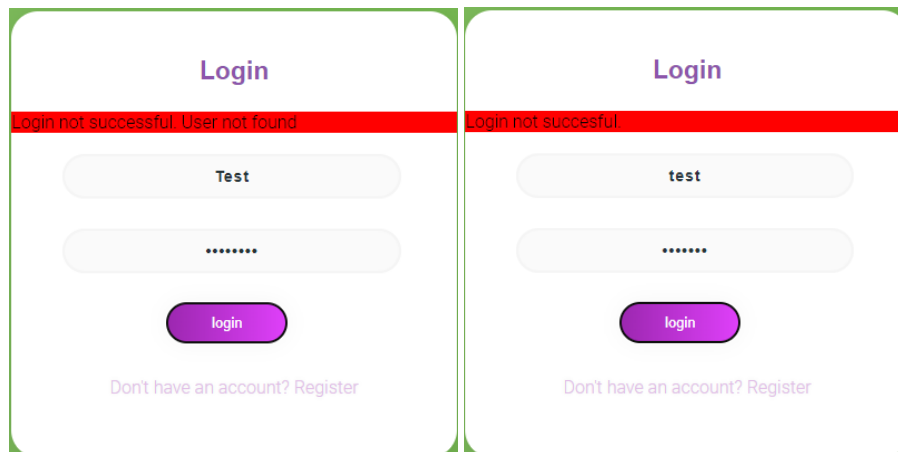


Figure 54: Incorrect username error

Figure 55: Incorrect password error

Comparing the wireframes to the result of the system, its clear to see that I didn't implement the side bar feature. This should be where the shopping list section should be instead of a link to click. This would make the list more accessible to the user and would complement the flow of the page better.

NFC SAVERS				
Welcome, Andy				
Check out your special deals as well as our stock, If you want to make a shopping list click here				
Deals				
Name	Points	Price £	Quantity	
Butter	2	1.8	50	2
Bacon	3	2.5	50	2
Biscuits	1	1.3	100	3
Lemonade	1	0.79	100	3
Products				
Name	Points	Price £	Quantity	
Bread	1	1.2	50	
Milk	2	1.5	50	
Chocolate Bar	1	0.8	100	
Apple	1	0.25	50	
Banana	1	0.2	50	
Grapes	2	2	50	
Water	1	1	100	
Cola	1	0.99	100	

Figure 56: Implementation page without sidebar

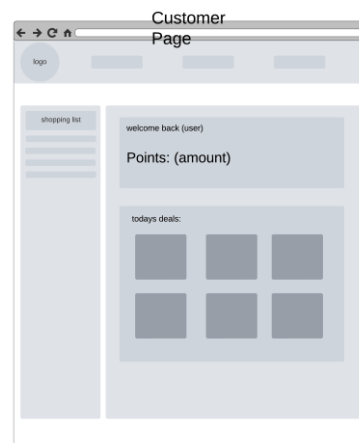


Figure 57: Wireframe with sidebar

6.2 Research Evaluation

The research survey that I conducted was able to provide valuable information and insight into the implementation of the system, however as it was a quantitative survey that focuses on acquiring data on a sort of scale, it is not able to collect true opinions with explanations and justifications like qualitative research can. Being able to collect explanations to small details relevant to the project that would be missed with a survey can provide information that not even myself could brainstorm as it allows the participant to make justifications and stipulations to their answers to better paint a visual of their true options. Being able to combine the two types of research would better assist the

collection of opinions as it would be able to cover a large sample size as well as gathering detailed feedback from a smaller focus group. Conducting this research comes with drawbacks however and is the reason it was not including in this project due to it having a strict deadline.

Looking back on the survey, I believe that I should have including screenshots of the prototype system and collect feedback on the usability and important of features. Having this extra information as well as the survey that I conducted would have produced a better insight into creating a better reward system. Knowing a potential user of the systems opinion on the functionality of features that will be included is very useful for resource management. It can outline where development and research time should be split before major development starts to optimise the production of the system and ultimately produce a better system in the same amount of time. I also could have taken my survey to a small business that would have been a likely candidate for a system like this and collect information and opinions from then to better develop my implementation further. This could have outlined key aspects that are critical to the life of the business that I would not have accounted for, helping with producing a more rigorous and prepared foundation of a system that could be implemented to a small business.

With this being said, this survey was able to provide this project with insight that was not previously available beforehand meaning that this report was able to add to the total research behind nfc reward systems. A research questions was asked in the introduction of this project that asked, “Is it possible to create a reward system that can help a small business enhance their customer retention and engagement?” and by conducting this survey this question can be answered. This was the goal of the research, to answer the question set, and it achieved that goal meaning it was a success.

6.3 Limitations and Challenges

A limitation to this project that was not originally defined would be the writing to the nfc card feature, as when the first prototype of this implementation was tested, it was not functional due to the low support for web based nfc support. Looking at a site called caniuse.com (2024), it has a section called Web NFC that displays coverage of support for desktop and mobile browsers. Adding all the coverage for the 6 desktop browsers with the most market share totals less than 1% global coverage whereas Chrome for Android alone has 42% supported global usage. This didn't stop the implementation of nfc support however as Node.js, the engine used to run the server of the project, has an extension called nfc-pcsc that adds the requirements needed for the browser to act like a mobile version and interact with the nfc card reader connected to the computer. This fix is specific to this project as replicating the shopping experience found some difficulties too as the inbuilt technology of nfc in mobile phones can be used to easily utilise this system but due to the project running locally on my personal computer, it could not be utilised.

Having to replicate the environment of a small business and its shopping experience was the reason for other limitations like not having the ability to test the application of saving data to the nfc card and then take it to a receiver at the shop that uses the data to manipulate the prices on the receipt. So, in theory the project supplies the user with all the tools needed to increase their engagement however I did not have the ability to take this functionality to an environment where I could confirm customer engagement would increase.

Chapter 7: Conclusion

7.1 Overview

This chapter will wrap up the overview and analysis of my project with some final words on the major findings from the research, the lessons I learned throughout the process, and what happens after the project ends. I'll explain what I would have done differently if I was to undertake the project again, as well as what changes and improvements I will make to this system from now on. This project and report have been increased my skillset in creating a complex project and seeing its success, skills that I can transfer to other aspects of my career to produce work of extreme quality.

7.2 Key Findings

Key Findings are the most notable and valuable pieces of information that I can take from this experience, whether this be information through research or discoveries through development and testing.

One of the key findings from this project is the ability to improve and better assist applications through nfc implementation. This was discovered through conducting research and made it apparent that there are systems that we use sometimes on a daily basis that can be upgraded with nfc to supply features that improve the users experience. This relates back to objective 4 of my project objectives *"To analyse opinions and implement requested functionality using quantitative data gathered using surveys"*, I tasked myself with collecting data regarding functionalities requested in the system and through this came the development of writing the users data to an nfc card. This shows the importance of surveying participants who fit the criteria of potential users of the system as the survey results showed clear indication of nfc technology increasing the users experience. As surveying participants who might not share the same views as users of the system can lead to misunderstanding the importance of prototype features and therefore not including them when in fact it would benefit the system by implementing the nfc technology compatibility.

The complexity that comes with planning and developing a relational database, that's equipped with the functionality needed to handle and manipulate data for a reward system, is also one of the key findings of the project as the development process of the project made it apparent that any issues and oversights in the database can lead to wrong information being written and displayed. Meaning before any development on the database can start, all documentation and planning diagrams need to be made and evaluated to ensure every data type, table, and entity is accounted for before creating the framework for the database. During my planning I originally missed out a key data column needed for displaying the customers data. When this wasn't present in the database, the systems true functionality of having personalised details of points and rewards was not present. This meant that modifications were needed in the database to accommodate for the systems code.

When looking to develop this project, it was apparent that there are multiple options for software utilities that can implement this system. Including database hosting options as well as website view engine options, it was clear that I had to evaluate each of these options and conclude which solution would be the most applicable to my scenario and which would provide me with the most appropriate features that I need for implementing the reward system. Some of these choices were made through familiarity from having used the software before but some choices were made due to it providing the features I need to develop the plans I have made. For example, I chose to use Node.js which is described on the Nodejs.org, (2024) website as *"designed to build scalable network applications"*, however according to EpamAnywhere, (2024) other options to execute the same functionality are: Python, ASP.NET, Java, and Ruby on Rails. These alternatives can all

achieve the same success of being able to implement my reward system but as I am very familiar with Node.js after using it for tasks and projects previously, it was clear that making this choice would reduce errors and issues and give me the ability to produce a higher quality system. This can be said about all other aspects of implementation of the system, alternative database hosting systems are also available, but these can come with subscription prices to host them as well as being forced to use the graphical interface that is included in the database. These could not provide features needed for my system or they could include features that are not relevant to me, causing confusion and incompatibility issues that would not be present by choosing a more appropriate solution to deploying a database. This is why I chose Mongo as it allows for locally hosted databases running off my machine rather than me connecting to a server located elsewhere. Hosting the database locally meant that I can chose when the database is active which eliminates the chance for the database to be unreachable due to outages or system failures that I cannot access myself. This does not mean that a cloud hosted database would not be a good option for this system as it could scale with the demand of the system, reducing installation and management costs due to these features being managed by a third party. Proving how the key finding from this is to analyse the project scope and to use appropriate solutions according to this scope.

7.3 Lessons Learned

While its useful to look at all the aspects that went well in a project, it's just as useful to assess the issues of the project and understand why those issue arose and therefore learning how to not let similar issues appear in other projects. These lessons taken from a finished project can provide better insight into a project than previous research can provide due to the relevancy of the problems to yourself and how you can work towards combatting them.

The first lesson I learned from this project is the importance of nfc technology inside mobile phones, with the lack of support for nfc in desktop browsers means that it becomes very important to fully utilise the technology available in a large range on phones. If I was to undertake this project again, I would focus my development on a mobile application rather than a web based application. This mobile app would then link to the desktop site instead of using both for mobile and desktop. With this change it would open up the opportunity to dive deeper into the use cases of nfc and maximise its potential in enhancing a users experience.

The other lesson that I learned from this project is that with the right amount of planning and research, it's possible to produce a project that can achieve its goals and objectives. By combining software utilities and intertwining them in the necessary way, a solution to an unsolved issue can be developed. The process of collecting previous research and adding to it with innovation is the sole reason to any kind of breakthroughs happening in all fields of information, and this is proven with this project. Any problems that stopped the development just became opportunities to overcome and improve the project rather than grinding the production to a stop This doesn't mean that there won't be an issue that cannot be overcome without further development of supporting technologies, but with the rate in which technologies like nfc are improving and expanding, it's unlikely that this will be the reason to a solution being unrealistic.

7.4 Further Work

This solution to a reward system isn't limited to the completion of this report, this project simply creates a foundation to be studied and adapted by myself or others with ideas on how to further improve its effectiveness. The project was created using all free to use software meaning the possibility of replicating this system is high, allowing for less limitations when implementing other resources and features that were not covered or mentioned in this report.

If I was to continue this project after the conclusion of this report, I would like to work towards the development of mobile application support for the system. This seems to me like the obvious next step for the system as it's in line with the aim of the project which is to increase customer engagement.

7.5 Conclusion

To conclude, this project report set out initially to provide a system for local businesses to increase customer retention and engagement through research and development, and this report has documented the process on how this project delivered on that. I have acquired the skills and knowledge required for this project to achieve its goals in the hopes of creating foundation that is used for further development of bringing a reward system to small business owners and customers.

Reference List

- Jain, G. and Dahiya, S., (2015). NFC?: Advantages, Limits and Future Scope. vol, 4, pp.1-12.
- Coskun, V., Ok, K. and Ozdenizci, B., (2013). Professional NFC application development for android. John Wiley & Sons.
- Frew, J. (2023) 9 cool ways to use NFC that'll Impress your friends, MUO. Available at: <https://www.makeuseof.com/tag/9-awesome-ways-use-nfc-thatll-impress-friends/#:~:text=NFC%20technology%20enables%20contactless%20payments,Google%20Pay%20and%20Apple%20Pay.> (Accessed: 15 November 2023).
- Coskun, V., Ok, K. and Ozdenizci, B., (2013), Near Field Communication (NFC): From Theory to Practice
- G. Madlmayr, J. Langer, C. Kantner and J. Scharinger, "NFC Devices: Security and Privacy," 2008
- Haselsteiner, E. and Breitfuß, K., (2006), July. Security in near field communication (NFC). In Workshop on RFID security
- Chattha, N.A., (2014), June. NFC—Vulnerabilities and defense. In 2014 Conference on Information Assurance and Cyber Security (CIACS) (pp. 35-38).
- Shen, W., Remédios, D., Sousa, L., Barata, M. and Osório, L., (2006). NFC technologies in mobile phones and emerging applications. In Information Technology For Balanced Manufacturing Systems: IFIP TC5, WG 5.5 Seventh International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, Niagara Falls, Ontario, Canada, September 4–6, 2006 (pp. 425-434). Springer US.
- Ramanathan, R. and Imtiaz, J., (2013), July. NFC in industrial applications for monitoring plant information. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-4).
- Shobha, N.S.S., Aruna, K.S.P., Bhagyashree, M.D.P. and Sarita, K.S.J., 2016, November. NFC and NFC payments: A review. In 2016 international conference on ICT in business industry & government (ICTBIG) (pp. 1-7).
- Staxpayments.com, NFC Mobile Payments: What Are They And How Do They Work? Available at: [https://staxpayments.com/blog/what-are-nfc-mobile-payments-and-how-do-they-work/#:~:text=Quick%20FAQs%20about%20NFC%20Mobile%20Payments&text=NFC%20\(Near%20Field%20Communication\)%20mobile](https://staxpayments.com/blog/what-are-nfc-mobile-payments-and-how-do-they-work/#:~:text=Quick%20FAQs%20about%20NFC%20Mobile%20Payments&text=NFC%20(Near%20Field%20Communication)%20mobile) (Accessed 21 November. 2023).
- Tabaku, E. and Zerellari, M., (2015). Brand loyalty and loyalty programs; a literature review. Romanian Economic and Business Review, 10(2), p.87.
- Demoulin, N.T. and Zidda, P., (2008). On the impact of loyalty cards on store loyalty: Does the customers' satisfaction with the reward scheme matter?. Journal of retailing and Consumer Services, 15(5), pp.386-398.
- Abd Ghadas, Zuhairah & Wan Ismail, Wan Nur Syahida & Aziz, Azwa & Harun, N.A. & Jusop, Mariana & Rahman, C.A.. (2015). LAFAMS: Account management system for Malaysian small legal firms. 23. 239-250.

Grand Canyon University (2021). The Most Effective Quantitative Data Collection Methods. [online] GCU. Available at: <https://www.gcu.edu/blog/doctoral-journey/most-effective-quantitative-data-collection-methods>

caniuse.com. (2024). *Web NFC | Can I use... Support tables for HTML5, CSS3, etc.* [online] Available at: <https://caniuse.com/webnfc> [Accessed 30 Apr. 2024].

Top 5 node.js alternatives (2024) EPAM Anywhere. [online] Available at: <https://anywhere.epam.com/en/blog/top-nodejs-alternatives> (Accessed: 01 May 2024).

Node.js - about node.js® (2024) Node.js -. [online] Available at: <https://nodejs.org/en/about> (Accessed: 01 Feb 2024).

Nikolaieva, A. (2024) 10 best software development methodologies: Uptech. Available at: <https://www.uptech.team/blog/software-development-methodologies> (Accessed: 03 Feb 2024).

Sharma, K. (2020) Top 12 software development methodologies, TatvaSoft. Available at: <https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/> (Accessed: 03 Feb 2024).

Universal Smart Cards (2021) How different industries use smart cards, 'Unlocking the Potential: How Various Industries Utilize Smart Cards - USmartCards'. Available at: <https://www.usmartcards.co.uk/news-blog/how-different-industries-use-smart-cards> (Accessed: 06 Feb 2024).

Stangarone, J. (2022) What is NFC, and how can it help your business?, mrc's Cup of Joe Blog. Available at: <https://www.mrc-productivity.com/blog/2012/09/what-is-nfc-and-how-can-it-help-your-business/> (Accessed: 06 Feb 2024).

Thanapal, P., Prabhu, J. and Jakhar, M., (2017), November. A survey on barcode RFID and NFC. In IOP Conference Series: Materials Science and Engineering (Vol. 263, No. 4, p. 042049). IOP Publishing.

Seritag (2023) NFC tag authentication explained, Seritag NFC Tags. Available at: <https://seritag.com/learn/using-nfc/nfc-tag-authentication-explained> (Accessed: 06 Feb 2024).

Mahansaria, D. and Roy, U.K., (2019), October. Secure authentication for ATM transactions using NFC technology. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-5). IEEE.

Oliveri, L.M., D'Urso, D., Trapani, N. and Chiacchio, F., (2024). An NFC application for the process mapping automation for SMEs. *Procedia Computer Science*, 232, pp.298-307.

BigCommerce (2023) What is a customer rewards program?, BigCommerce. Available at: <https://www.bigcommerce.co.uk/glossary/customer-rewards-program/> (Accessed: February 2024).

Rizvi, J. (2021) The perks of a great loyalty program, Forbes. Available at: <https://www.forbes.com/sites/jiawertz/2021/09/26/the-perks-of-a-great-loyalty-program/> (Accessed: February 2024).

Polkes, A. (2024) The state of Brand Loyalty 2021: Global Consumer Survey, Yotpo. Available at: <https://www.yotpo.com/blog/the-state-of-brand-loyalty-2021-global-consumer-survey/> (Accessed: March 2024).

MongoDB (2024) Comparing the differences - mongodb vs mysql, MongoDB. Available at: <https://www.mongodb.com/compare/mongodb-mysql#:~:text=Why%20is%20using%20MongoDB%20better,applications%20more%20efficiently%20at%20scale.> (Accessed: February 2024).

Singh, S. (2024) Top 15 databases to use in 2022 and beyond, Appinventiv. Available at: <https://appinventiv.com/blog/top-web-app-database-list/> (Accessed: March 2024).

Appendix A - Initial Project Proposal

Project (CN6000)

Initial Proposal Form

Programme: Computing for business

Year: 2023

Student Number: 2034853

Proposed Title: Contactless mobile web reward system to interact and engage with customers.

Proposed Aim:

To Develop a working demo of a JavaScript web and mobile interface with a dynamic relational database system that assists and rewards customers that shop using integrated Near Field Communication (NFC) technology.

Rationale:

The system will consist of a backend database made in MongoDB, it will store the data of the customers as well as relating the tables to make the database dynamic and allow for manipulation of the data to provide the customer with details on spending, trends, and potential deals. This will be paired with a web based interactive front end which will provide the user with the tools and information to use and assess the data in efficient and effective ways to maximise their shopping experience. Data will be stored and transferred between the database and the user by utilising NFC cards accompanied with NFC readers within the shop. Customers will have their own personalised accounts which will be tailored specifically to them. This will enable customers to increase their shopping experience and be rewarded for their actions whilst not needing to worry about carrying the necessary receipts, coupons, or debit cards. This system can also be utilised by staff of shops as it will provide information on inventory levels, popularity of products and potential theft of stock. Customers will not be able to access this admin area of the system on their account. This will allow customers to ensure they can get the most out of their shopping experience without barriers to entry like access to mobile phones and cellular plans.

Facilities required: NFC reader and tags, MySQL or MongoDB, Visual Studio Code IDE, Windows 11

Supervisor: Dr .Hisham AbouGrad

Appendix B - Final Project Proposal

Project (CN6000)	Final Proposal Form
Programme: Computing for Business	Year: 2023
Student Number: 2034853	
Proposed Title: Contactless mobile web reward system to enhance customer engagement.	
Proposed Aim: The project seeks to develop a working demo of a JavaScript web and mobile interface with a dynamic relational database system that will be used by local shops to increase customer repeat purchasing and customer retention by integrating Near Field Communication (NFC) technology	
Objectives: 1. To research the literature of using NFC for database systems 2. To research the effective aspects of an efficient customer reward system 3. To conduct a survey to collect habits and opinions on shopping preferences 4. To analyse opinions and implement requested functionality using quantitative data gathered using surveys 5. To design and implement a working demo database system with interactive Graphic User Interface 6. To test the functionality of the system in comparison to other methods 7. To evaluation the tools and techniques required to design and implement a customer reward system and database 8. To reflect on the limitations and success of the system and evaluate the findings	
Rationale: Many large supermarkets will have a built in reward system for their customers, helping encourage them to come back and shop again. The same cannot be said for smaller local businesses, this is why I will be looking to implement a reward system that can be utilised by these small businesses. Offering better prices and incentives to return to the shop in hopes that it will increase customer retention and satisfaction. A system like this can help improve turnover and profitability, especially utilising the technology of NFC. Jain, G. 2015 defines Near Field Connection (NFC) as the wireless communication between devices by transmitting information in a localised area. Combining this with what (Coskun, V et al, 2013) said about devices like smartphones and smart cards, they can use this technology for contactless payments. I will be adapting on this use case to create a process where customer information is placed onto the smart card ready for the user to utilise its perks within the store The system will consist of a backend database made in MongoDB, it will store the data of the customers as well as relating the tables to make the database dynamic and allow for manipulation of the data to provide the customer with details on spending, trends, and potential deals. This will be paired with a web based interactive front end which will provide the user with the tools and information to use and assess the data in efficient and effective ways to maximise their shopping experience. Data will be stored and transferred between the database and the user by utilising NFC	

cards accompanied with NFC readers within the shop. Customers will have their own personalised accounts which will be tailored specifically to them. This will enable customers to increase their shopping experience and be rewarded for their actions whilst not needing to worry about carrying the necessary receipts, coupons, or debit cards. This system can also be utilised by staff of shops as it will provide information on inventory levels, popularity of products and potential theft of stock. Customers will not be able to access this admin area of the system on their account. This will allow customers to ensure they can get the most out of their shopping experience without barriers to entry like access to mobile phones and cellular plans

Facilities required: NFC reader and tags, MongoDB, Visual Studio Code IDE, Windows 11

Supervisor:

Appendix C – Source Code

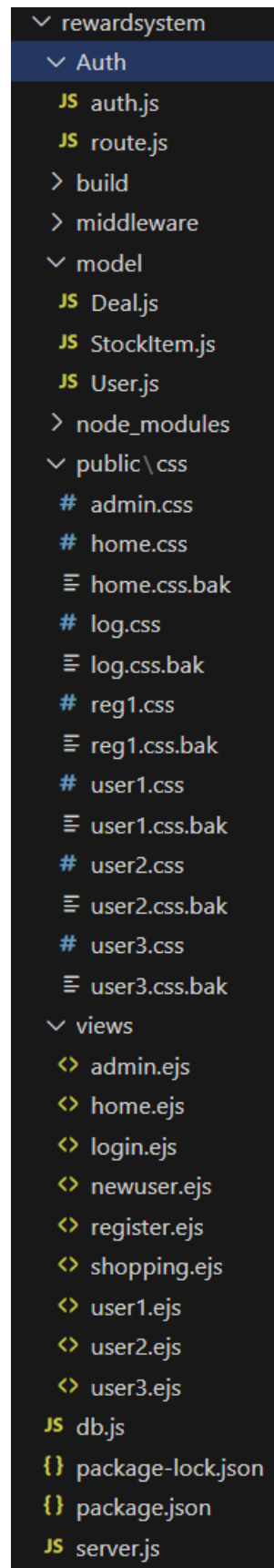


Figure 58: File structure

```
//models and routes needed
const express = require("express");
const connectDB = require("./db");
const app = express();
const cookieParser = require("cookie-parser");
const { adminAuth, userAuth } = require(
  "../middleware/auth.js");
const Stock = require("../model/StockItem");
const Deal = require("../model/Deal");
const User = require("../model/User");
const cors = require("cors");

//local host port
const PORT = 9999;
//engine for views
app.set("view engine", "ejs");

connectDB();
//packaging moving and displaying data
app.use(express.json());
app.use(cookieParser());
app.use(express.static("public"));

//routes
app.use("/api/auth", require("./Auth/route"));

//pages
app.get("/", (req, res) => res.render("home"));
app.get("/user1", async (req, res) => {
  try {
    //fetch all stock
    const allStock = await Stock.find({});

    // random stock
    const randomStock = await Deal.aggregate([
      $sample: { size: 4 } ]]);

    const user = await User.findOne({ username: 'Andy'
    });
    // html template
    res.render('user1', { allStock: allStock,
      randomStock: randomStock, points: user.points });
    } catch (err) {
      res.status(500).send(err.message);
    }
  });
  app.get('/user2', async (req, res) => {
    try {
      const allStock = await Stock.find({});
      const randomStock = await Deal.aggregate([
        $sample: { size: 4 } ]]);
      const user = await User.findOne({ username: 'Bob'
      });
      res.render('user2', { allStock: allStock,
        randomStock: randomStock, points: user.points});
    } catch (err) {
      res.status(500).send(err.message);
    }
  });
  app.get('/user3', async (req, res) => {
    try {
      const allStock = await Stock.find({});
      const randomStock = await Deal.aggregate([
        $sample: { size: 4 } ]]);
      const user = await User.findOne({ username: 'Carl'
      });
      res.render('user3', { allStock: allStock,
        randomStock: randomStock, points: user.points});
    } catch (err) {
      res.status(500).send(err.message);
    }
  });
  app.get('/newuser', async (req, res) => {
    try {
      const allStock = await Stock.find({});
      const randomStock = await Deal.aggregate([
        $sample: { size: 4 } ]]);
      res.render('newuser', { allStock: allStock,
        randomStock: randomStock});
    } catch (err) {
      res.status(500).send(err.message);
    }
  });
  app.get("/register", (req, res) => res.render(
    "register"));
  app.get("/login", (req, res) => res.render("login"));
  app.get("/shopping", (req, res) => res.render(
    "shopping"));
  app.get("/logout", (req, res) => {
    //checks token
    res.cookie("jwt", "", { maxAge: "1" });
    res.redirect("/");
  });
  //using Auth check
  app.get("/admin", adminAuth, (req, res) => res.render(
    "admin"));
  app.get("/user", userAuth, (req, res) => res.render(
    "user"));
  //nfc write
  app.post('/write-nfc', (req, res) => {
    const { data } = req.body;
    res.send('Data written to NFC successfully');
  });
  let shoppingLists = []; // store shopping lists
  app.post('/api/shopping-lists', (req, res) => {
    const { list } = req.body;
    shoppingLists.push(list); // store list
    res.status(200).send('List received');
  });
  //send list
  app.get('/api/shopping-lists', (req, res) => {
    res.json({ lists: shoppingLists });
  });
  const server = app.listen(PORT, () =>
    console.log('Server Connected to port ${PORT}'))
  );
  process.on("unhandledRejection", (err) => {
    console.log('An error occurred: ${err.message}');
    server.close() => process.exit(1);
  });
};
```

Figure 59: server.js

```
exports.update = async (req, res, next) => {
  const { role, id } = req.body;
  // verify role
  if (role && id) {
    // verify if admin
    if (role === "admin") {
      // Finds the user with the id
      await User.findById(id)
        .then((user) => {
          if (user.role !== "admin") {
            user.role = role;
            user.save((err) => {
              //Mongoadb
              if (err) {
                return res
                  .status("400")
                  .json({ message: "An error occurred"
                  }, error: err.message );
                process.exit(1);
              }
              res.status("201").json({ message:
                "Update successful", user });
            });
          } else {
            res.status(400).json({ message:
              "User is already an Admin" });
          }
        })
        .catch((error) => {
          res
            .status(400)
            .json({ message: "An error occurred", error
            : error.message });
        });
      } else {
        res.status(400).json({
          message: "Role is not admin",
        });
      }
    } else {
      res.status(400).json({ message:
        "Role or Id not present" });
    }
  }
  //delete users function
  exports.deleteUser = async (req, res, next) => {
    const { id } = req.body;
    await User.findById(id)
      .then((user) => user.remove())
      .then((user) =>
        res.status(201).json({ message:
          "User successfully deleted", user })
      )
      .catch((error) =>
        res
          .status(400)
          .json({ message: "An error occurred", error:
            error.message })
      );
  };
  //get users function
  exports.getUsers = async (req, res, next) => {
    await User.find({})
      .then((users) => {
        const userFunction = users.map((user) => {
          const container = {};
          container.username = user.username;
          container.role = user.role;
          container.id = user._id;

          return container;
        });
        res.status(200).json({ user: userFunction });
      })
      .catch((err) =>
        res.status(401).json({ message: "Not successful"
        }, error: err.message )
      );
  };
};
```

Figure 60: Auth/auth.js (1/2)

```
const User = require("../model/User");
const bcrypt = require("bcryptjs");
const jwt = require("jsonwebtoken");

//password secret
const jwtSecret =
  "4715aed3c946f7b0a38e6b534a9583628d84e96d10fbc84700770d
  572af3dce43625dd";

exports.register = async (req, res, next) => {
  const { username, password } = req.body;
  if (password.length < 6) {
    //password length error
    return res.status(400).json({ message:
      "Password less than 6 characters" });
  }
  //accept and create
  bcrypt.hash(password, 10).then(async (hash) => {
    await User.create({
      username,
      password: hash,
    })
      .then((user) => {
        //token 3hour max age
        const maxAge = 3 * 60 * 60;
        const token = jwt.sign(
          { id: user._id, username, role: user.role },
          jwtSecret,
          {
            expiresIn: maxAge,
          }
        );
        res.cookie("jwt", token, {
          httpOnly: true,
          maxAge: maxAge * 1000,
        });
        res.status(201).json({
          message: "User successfully created",
          user: user._id,
          role: user.role,
        });
      })
      .catch((error) =>
        res.status(400).json({
          message: "User not successful created",
          error: error.message,
        })
      );
  });
  exports.login = async (req, res, next) => {
    const { username, password } = req.body;

    // check user and pass
    if (username || !password) {
      return res.status(400).json({
        message: "Username or Password not present",
      });
    }
    try {
      const user = await User.findOne({ username });

      if (user) {
        res.status(400).json({
          message: "Login not successful",
          error: "User not found",
        });
      } else {
        // compare pass
        bcrypt.compare(password, user.password).then(
          function (result) {
            if (result) {
              const maxAge = 3 * 60 * 60;
              const token = jwt.sign(
                { id: user._id, username, role: user.role
                },
                jwtSecret,
                {
                  expiresIn: maxAge,
                }
              );
              res.cookie("jwt", token, {
                httpOnly: true,
                maxAge: maxAge * 1000, // 3hrs in ms
              });
              res.status(201).json({
                message: "User successfully logged in",
                user: user._id,
                role: user.role,
              });
            } else {
              res.status(400).json({ message:
                "Login not successful" });
            }
          });
      } catch (error) {
        res.status(400).json({
          message: "An error occurred",
          error: error.message,
        });
      }
    }
  };
};
```

Figure 61: Auth/auth.js (2/2)

```

const express = require("express");
const router = express.Router();

const { register, login, update, deleteUser, getUsers } = require("../auth");
const { adminAuth } = require("../middleware/auth");

router.route("/register").post(register);
router.route("/login").post(login);
router.route("/update").put(adminAuth, update);
router.route("/deleteUser").delete(adminAuth, deleteUser);
router.route("/getUsers").get(getUsers);

module.exports = router;

```

Figure 62: Auth/route.js

```

const jwt = require("jsonwebtoken");
const jwtSecret =

"4715aed3c946f7b0a38e6b534a9583628d84e96d10fbc04700770d
572af3dce43625dd"
;

exports.adminAuth = (req, res, next) => {
  const token = req.cookies.jwt;
  if (token) {
    jwt.verify(token, jwtSecret, (err, decodedToken) => {
      if (err) {
        return res.status(401).json({ message:
"Not authorized" });
      } else {
        if (decodedToken.role !== "admin") {
          return res.status(401).json({ message:
"Not authorized" });
        } else {
          next();
        }
      }
    });
  } else {
    return res
      .status(401)
      .json({ message:
"Not authorized, token not available" });
  }
};

exports.userAuth = (req, res, next) => {
  const token = req.cookies.jwt;
  if (token) {
    jwt.verify(token, jwtSecret, (err, decodedToken) => {
      if (err) {
        return res.status(401).json({ message:
"Not authorized" });
      } else {
        if (decodedToken.role !== "Basic") {
          return res.status(401).json({ message:
"Not authorized" });
        } else {
          next();
        }
      }
    });
  } else {
    return res
      .status(401)
      .json({ message:
"Not authorized, token not available" });
  }
};

```

Figure 63: middleware/auth.js

```
const Mongoose = require("mongoose");

const StockItemSchema = new Mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  points: {
    type: String,
    required: true
  },
  price: {
    type: Number,
    required: true
  },
  quantity: {
    type: Number,
    required: true
  },
  stockId: {
    type: Number,
    required: false
  },
}, { collection: 'Stock' });

const Stock = Mongoose.model("stock", StockItemSchema);

module.exports = Stock;
```

Figure 64: model/StockItem.js

```
const Mongoose = require("mongoose");

const DealSchema = new Mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  points: {
    type: String,
    required: true
  },
  price: {
    type: Number,
    required: true
  },
  quantity: {
    type: Number,
    required: true
  },
  deal: {
    type: String,
    required: true
  },
  stockId: {
    type: Number,
    required: false
  },
}, { collection: 'Deals' });

const Deal = Mongoose.model("deals", DealSchema);

module.exports = Deal;
```

Figure 65: model/Deal.js

```
const Mongoose = require("mongoose");

const UserSchema = new Mongoose.Schema({
  username: {
    type: String,
    unique: true,
    required: true,
  },
  password: {
    type: String,
    minlength: 6,
    required: true,
  },
  role: {
    type: String,
    default: "Basic",
    required: true,
  },
  points: {
    type: Number,
    default: 500,
    required: true,
  },
}, { collection: 'Customer' });

const User = Mongoose.model("user", UserSchema);

module.exports = User;
```

Figure 66: model/User.js

```

@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.home {
  width: 300px;
  padding: 0 0 0;
  margin: auto;
}
body {
  background: #f0f0f0; /* fallback for old browsers */
  background: rgb(141,194,111);
  background: linear-gradient(90deg, rgb(141,194,111) 49%, rgb(118,184,82) 50%);
  font-family: 'Roboto', sans-serif;
  -webkit-font-smoothing: antialiased;
  -ms-oss-font-smoothing: grayscale;
}

.main {
  background-color: #ffffff;
  width: 400px;
  height: 400px;
  margin: 7px auto;
  border-radius: 1.5em;
  box-shadow: 0px 1px 3px 2px rgba(0, 0, 0, 0.14);
}

.homepage {
  padding-top: 40px;
  color: #8C55AA;
  font-family: 'Ubuntu', sans-serif;
  font-weight: bold;
  font-size: 40px;
}

.para {
  font-family: 'Ubuntu', sans-serif;
  color: #E1BEE7;
  font-weight: bold;
}

.links {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #8C55AA;
  padding-top: 15px;
  font-size: 25px;
}

# {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #8C55AA;
  font-weight: bold;
  text-decoration: none
}

@media (max-width: 600px) {
  .main {
    border-radius: 0px;
  }
}

```

Figure 67: public/home.css

```

body {
  margin: 0;
  font-family: Arial, sans-serif;
}

.navbar {
  overflow: hidden;
  background-color: green;
  padding: 10px;
  transition: 0.4s;
  z-index: 1000;
  width: 100%;
  position: fixed;
  top: 0;
  display: flex;
  justify-content: space-between; /* Align items to the
  box-sizing: border-box;
}

.navbar-left {
  display: flex;
  align-items: center;
}

.navbar-right {
  display: flex;
  align-items: center;
}

.navbar h1 {
  color: white;
  margin: 0;
  flex-shrink: 0; /* Prevent shrinking of the Logo */
}

.navbar button {
  background-color: white;
  color: green;
  border: none;
  padding: 8px 16px;
  cursor: pointer;
  border-radius: 5px;
}

.navbar button:hover {
  background-color: #ddd;
}

.sticky {
  position: fixed;
  top: 0;
  width: 100%;
}

.right {
  float: right;
  font-size: 50px;
}

.left {
  float: left;
  font-size: 50px;
}

.welcome {
  font-size: 25px;
  font-weight: bold;
}

```

Figure 68: public/user.css

```

@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.register {
  width: 300px;
  padding: 0 0 0;
  margin: auto;
}
body {
  background: #f0f0f0;
  background: rgb(141,194,111);
  background: linear-gradient(90deg, rgb(141,194,111) 49%, rgb(118,184,82) 50%);
  font-family: 'Roboto', sans-serif;
  -webkit-font-smoothing: antialiased;
  -ms-oss-font-smoothing: grayscale;
}

.main {
  background-color: #ffffff;
  width: 400px;
  height: 400px;
  margin: 7px auto;
  border-radius: 1.5em;
  box-shadow: 0px 1px 3px 2px rgba(0, 0, 0, 0.14);
}

.logpage {
  padding-top: 40px;
  color: #8C55AA;
  font-family: 'Ubuntu', sans-serif;
  font-weight: bold;
  font-size: 40px;
}

.username {
  width: 70%;
  color: rgb(14, 50, 50);
  font-weight: 700;
  font-size: 14px;
  letter-spacing: 1px;
  background: rgb(136, 126, 126, 0.04);
  padding: 10px 20px;
  border: none;
  border-radius: 20px;
  outline: none;
  box-shadow: border-box;
  border: 2px solid rgba(0, 0, 0, 0.02);
  margin-bottom: 50px;
  margin-left: 40px;
  text-align: center;
  margin-bottom: 27px;
  font-family: 'Ubuntu', sans-serif;
}

form:focus {
  padding-top: 40px;
}

.password {
  width: 70%;
  color: rgb(14, 50, 50);
  font-weight: 700;
  font-size: 14px;
  letter-spacing: 1px;
  background: rgb(136, 126, 126, 0.04);
  padding: 10px 20px;
  border: none;
  border-radius: 20px;
  outline: none;
  box-shadow: border-box;
  border: 2px solid rgba(0, 0, 0, 0.02);
  margin-bottom: 50px;
  margin-left: 40px;
  text-align: center;
  margin-bottom: 27px;
  font-family: 'Ubuntu', sans-serif;
}

.username:focus, .password:focus {
  border: 2px solid rgba(0, 0, 0, 0.18) !important;
}

.submit {
  cursor: pointer;
  border-radius: 5px;
  color: #fff;
  background: linear-gradient(to right, #8C2780, #E69600);
  border: 0;
  padding-left: 40px;
  padding-right: 40px;
  padding-bottom: 10px;
  padding-top: 10px;
  font-family: 'Ubuntu', sans-serif;
  margin-left: 10px;
  font-size: 16px;
  box-shadow: 0 0 20px 3px rgba(0, 0, 0, 0.04);
}

.log_in {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #E1BEE7;
  padding-top: 15px;
}

# {
  text-shadow: 0px 0px 3px rgba(117, 117, 117, 0.12);
  color: #E1BEE7;
  text-decoration: none
}

@media (max-width: 600px) {
  .main {
    border-radius: 0px;
  }
}

```

Figure 69: public/log.css

```

const Mongoose = require("mongoose");

const localDB = 'mongodb://localhost:27017/RewardSystem';

const connectDB = async () => {
  await Mongoose.connect(localDB, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  });

  console.log("MongoDB Connected");
};

module.exports = connectDB;

```

Figure 69: db.js

```

body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh; /* Use full height to center vertically */
  margin: 0;
  font-family: Arial, sans-serif;
  display: flex;
  flex-direction: column; /* Stacks children vertically */
  align-items: center;
}

.navbar {
  overflow: hidden;
  background-color: green;
  padding: 10px 10px 0 10px;
  transition: 0.4s;
  z-index: 1000;
  width: 100%;
  position: fixed;
  top: 0;
  display: flex;
  justify-content: space-between; /* Align items to the left and right */
  box-sizing: border-box;
}

.navbar-left {
  display: flex;
  align-items: center;
}

.navbar-right {
  display: flex;
  align-items: center;
}

.navbar h1 {
  color: white;
  margin: 0;
  flex-shrink: 0; /* Prevent shrinking of the Logo */
}

.navbar button {
  background-color: white;
  color: green;
  border: none;
  padding: 8px 16px;
  cursor: pointer;
  border-radius: 5px;
}

.navbar button:hover {
  background-color: #ddd;
}

.sticky {
  position: fixed;
  top: 0;
  width: 100%;
}

.right {
  float: right;
  font-size: 50px;
}

.left {
  float: left;
  font-size: 50px;
}

.welcome {
  font-size: 25px;
  font-weight: bold;
}

.content {
  text-align: center;
  width: 50%; /* Adjust width as needed */
  margin-top: 0;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1); /* Optional: adds shadow for better
  background-color: #f9f9f9; /* Light grey background */
}

ul {
  list-style-type: none; /* Removes bullet points from list */
  padding: 0;
}

li {
  margin-bottom: 10px; /* Adds space between list items */
}

button {
  margin: 5px;
  padding: 8px 16px;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

.shopping-lists {
  margin-top: 20px;
  padding: 15px;
  border: 1px solid #ccc;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  background-color: #f9f9f9;
}

```

Figure 70: public/admin.css

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/admin.css">
  <title>Admin page</title>
</head>

<body>
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"> <a href="/logout">Log Out</a></button>
    </div>
  </div>

  <!-- display -->

  <div class="display content">
    <h1>All customers</h1>
    <ul></ul>
  </div>

  <div class="shopping-lists">
    <h2>Shopping Lists</h2>
    <ul id="shoppingLists"></ul>
  </div>

  <script>
    function fetchShoppingLists() {
      fetch('/api/shopping-lists')
        .then(response => response.json())
        .then(data => {
          const ul = document.getElementById('shoppingLists');
          data.lists.forEach(list => {
            const li = document.createElement('li');
            li.textContent = list; // Assuming list is a simple string
            ul.appendChild(li);
          });
        })
        .catch(error => console.error('Error fetching shopping lists:', error));
    }

    fetchShoppingLists();
  </script>

  <script>
    const ul = document.querySelector('ul')
    const display = document.querySelector('.display')
    // get all users
    const getUsers = async () => {
      const res = await fetch('/api/auth/getUsers')
      const data = await res.json()
      data.user.map((mappedUser) => {
        //show user if not admin
        if(mappedUser.username !== 'admin'){
          let li = <li><b>Username</b> => ${mappedUser.username} <br> <b>Role</b> => ${mappedUser.role}
          </li> <button class="edit">Edit Role</button> <button class="delete">Delete User</button>
          ul.innerHTML += li
        }else{
          return null
        }
      })
      const editRole = document.querySelectorAll('.edit')
      const deleteUser = document.querySelectorAll('.delete')
      // edit role
      editRole.forEach((button, i) => {
        button.addEventListener('click', async () => {
          display.textContent = ''
          const id = data.user[i+1].id
          const res = await fetch('/api/auth/update', {
            method: 'PUT',
            body: JSON.stringify({ role: 'admin', id }),
            headers: { 'Content-Type': 'application/json' }
          })
          const dataUpdate = await res.json()
          if (res.status === 400 || res.status === 401) {
            document.body.scrollTop = 0
            document.documentElement.scrollTop = 0
            return display.textContent = `${dataUpdate.message}. ${dataUpdate.error ? dataUpdate.error : ''}`
          }
          location.assign('/admin')
        })
      })
      // delete user
      deleteUser.forEach((button, i) => {
        button.addEventListener('click', async () => {
          display.textContent = ''
          const id = data.user[i+1].id
          const res = await fetch('/api/auth/deleteUser', {
            method: 'DELETE',
            body: JSON.stringify({id}),
            headers: { 'Content-Type': 'application/json' }
          })
          const dataDelete = await res.json()
          if (res.status === 401){
            document.body.scrollTop = 0
            document.documentElement.scrollTop = 0
            return display.textContent = `${dataUpdate.message}. ${dataUpdate.error ? dataUpdate.error : ''}`
          }
          location.assign('/admin')
        })
      })
    }
    getUsers()
  </script>
</body>
</html>

```

Figure 71: views/admin.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/home.css">
  <title>Home page</title>
</head>
<body>
  <div class="main">
    <p class="homepage" align="center">NFC SAVERS</p>
    <p class="para" align="center">
      Welcome to NFC Savers, our aim is to provide you with a simple solution to getting simple savings.
      Create your own personal page now or log in to view your deals
    </p>
    <p class="links" align="center"><a href="/register">Register</a></p>
    <p class="links" align="center"><a href="/login">Login</a></p>
  </div>
</body>
</html>

```

Figure 72: views/home.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/log.css">
  <title>Login Page</title>
</head>
<body>
  <div class="main">
    <p class="loginpage" align="center">Login</p>
    <form>
      <div class="error" style="background-color: red;"></div><br>
      <input type="text" id="username" class="username" placeholder="Username" align="center"
      required/><br>
      <input type="password" id="password" class="password" placeholder="Password" align="center"
      required/><br>
      <input type="submit" id="login" value="login" class="submit" align="center"><br>
    </form>
    <p class="log_in" align="center"><a href="/register">Don't have an account? Register</a></p>
  </div>

  <script>
    const form = document.querySelector('form')
    const username = document.querySelector('#username')
    const password = document.querySelector('#password')
    const display = document.querySelector('.error')

    form.addEventListener('submit', async (e) => {
      e.preventDefault()
      display.textContent = ''
      try {
        const res = await fetch('/api/auth/login', {
          method: 'POST',
          body: JSON.stringify({ username: username.value, password: password.value }),
          headers: { 'Content-Type': 'application/json' }
        })
        const data = await res.json()
        if (res.status === 400 || res.status === 401) {
          return display.textContent = `${data.message}. ${data.error ? data.error : ''}`
        }
        //user already redirect
        if (data.user === "662d28c291845bc48c33b49e") {
          location.assign('/user1');
        } else if (data.user === "661fe312571620cb89abe15b") {
          location.assign('/user2');
        } else if (data.user === "661fe31a571620cb89abe15d") {
          location.assign('/user3');
        } else if (data.user === "662d22489dad60e241ce45ee") {
          location.assign('/admin');
        } else {
          // new user redirect
          location.assign('/newuser');
        }
      } catch (err) {
        console.log(err.message)
      }
    })
  </script>
</body>
</html>

```

Figure 73: views/login.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NFC Savers</title>
  <link rel="stylesheet" href="css/user1.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JhJ6yG6W4ALEwIH" crossorigin="anonymous">
</head>
<body>
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"> <a href="/logout">Log Out</a></button>
    </div>
  </div>

  <br>

  <h1 class="right">Points: 500</h1><h1 class="left">Welcome</h1>

  <br>
  <br>
  <br>

  <!-- create and view shopping list -->
  <div>
    <p class="welcome">
      Check out your special deals as well as our stock, If you want to make a shopping list click <a href="/shopping">here</a></p>
    </div>
    <!-- deals table -->
    <div>
      <h1>Deals</h1>
      <table class="table table-sm">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Points</th>
            <th scope="col">Price £</th>
            <th scope="col">Quantity</th>
            <th scope="col">Deal</th>
          </tr>
        </thead>
        <tbody>
          <{{randomStock.forEach(deal => {
            <tr>
              <th scope="row"><%= deal.name %></th>
              <td><%= deal.points %></td>
              <td><%= deal.price %></td>
              <td><%= deal.quantity %></td>
              <td><%= deal.deal %></td>
            </tr>
          <}}%>
          </tbody>
        </table>
      </div>

      <div>
        <h1>Products</h1>
        <table class="table table-sm">
          <thead>
            <tr>
              <th scope="col">Name</th>
              <th scope="col">Points</th>
              <th scope="col">Price £</th>
              <th scope="col">Quantity</th>
            </tr>
          </thead>
          <tbody>
            <{{allStock.forEach(stock => {
              <tr>
                <th scope="row"><%= stock.name %></th>
                <td><%= stock.points %></td>
                <td><%= stock.price %></td>
                <td><%= stock.quantity %></td>
              </tr>
            <}}%>
            </tbody>
          </table>
        </div>

      <script>
        window.onscroll = function() {myFunction()};

        var navbar = document.getElementById("navbar");
        var sticky = navbar.offsetTop;

        function myFunction() {
          if (window.pageYOffset >= sticky) {
            navbar.classList.add("sticky");
          } else {
            navbar.classList.remove("sticky");
          }
        }
      </script>
    </body>
  </html>

```

Figure 74: views/newuser.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/reg1.css">
  <title>Register Page</title>
</head>
<body>
  <div class="main">
    <p class="regpage" align="center">Register</p>
    <!-- register form -->
    <form>
      <div class="error" style="background-color: red;"></div><br>
      <input type="text" id="username" class="username" placeholder="Username" align="center" required
    /><br>
      <input type="password" id="password" class="password" placeholder="Password" align="center"
    required><br>
      <input type="submit" id="register" value="register" class="submit" align="center"><br>
    </form>
    <p class="log_in" align="center"><a href="/login">Already Registered? Login</p>
  </div>

  <script>
    // use data
    const form = document.querySelector('form')
    const username = document.querySelector('#username')
    const password = document.querySelector('#password')
    const display = document.querySelector('.error')

    form.addEventListener('submit', async (e) => {
      e.preventDefault()
      display.textContent = ''
      try {
        const res = await fetch('/api/auth/register', {
          method: 'POST',
          //make data json
          body: JSON.stringify({ username: username.value, password: password.value }),
          headers: { 'Content-Type': 'application/json' }
        })
        const data = await res.json()
        if(res.status === 400 || res.status === 401){
          return display.textContent = `${data.message}. ${data.error ? data.error : ''}`
        }
        data.role === "admin" ? location.assign('/user') : location.assign('/login')
      } catch (err) {
        console.log(err.message)
      }
    })
  </script>
</body>
</html>

```

Figure 75: views/register.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NFC Savers</title>
  <link rel="stylesheet" href="css/user3.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-QktK2yjpPEjISvSuaRU90FeRpk6YctnYmDr5pNlyT2BRJXh03MhJy6NwALEwIH" crossorigin="anonymous">
</head>
<body>
  <!-- navbar -->
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"><a href="/logout">Log Out:</a></button>
    </div>
  </div>
  <!-- list -->
  <div>
    <h1>Shopping List</h1>

    <form method="POST" name="shoppingList">
      <br>
      Product <input type="text" name="product">
      Quantity: <input type="number" name="quantity">
      <!-- <br><br> run function on click -->
      <input type="button" name="add" value="Add Data" onclick="addList();" class="btn btn-success"><br>

      <table id="tbl" class="table" border="1">
        <thead>
          <th>Product</th>
          <th>Quantity</th>
          <th>Delete</th>
        </thead>
        <tbody>
          <tr>
            <td><input type="text" name="product"></td>
            <td><input type="number" name="quantity"></td>
            <td><input type="button" name="del" value="Delete" onclick="delList(this);" class="btn btn-danger"></td>
          </tr>
        </tbody>
      </table>
      <input type="button" value="Save List" name="save" onclick="saveList();" class="btn btn-info">
    </form>

    <script type="text/javascript">
      // shopping list function
      function addList() {
        var product = document.shoppingList.product.value;
        var quantity = document.shoppingList.quantity.value;
        var tr = document.createElement('tr');
        // parent and child to fill table
        var td1 = tr.appendChild(document.createElement('td'));
        var td2 = tr.appendChild(document.createElement('td'));
        var td3 = tr.appendChild(document.createElement('td'));

        td1.innerHTML=product;
        td2.innerHTML=quantity;
        td3.innerHTML=
        <input type="button" name="del" value="Delete" onclick="delList(this);" class="btn btn-danger">

        document.getElementById("tbl").appendChild(tr);
      }

      // delete function
      function delList(list){
        var s=list.parentNode.parentNode;
        s.parentNode.removeChild(s);
      }

      // save function
      function saveList() {
        const table = document.getElementById('tbl');
        let data = '';
        for (let i = 1; i < table.rows.length; i++) {
          const product = table.rows[i].cells[0].innerText;
          const quantity = table.rows[i].cells[1].innerText;
          data += `${product},${quantity}\n`;
        }
        // nfc write
        fetch('/write-nfc', {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json',
          },
          body: JSON.stringify({ data })
        })
        .then(response => response.text())
        .then(result => {
          alert(result);
          // send data to admin
          return fetch('/api/shopping-lists', {
            method: 'POST',
            headers: {
              'Content-Type': 'application/json',
            },
            body: JSON.stringify({ list: data })
          });
        })
        .catch(error => console.error("Error:", error));
      }
    </script>

    <script>
      // navbar stays at top on scroll
      window.onscroll = function() {sticky()};

      var navbar = document.getElementById("navbar");
      var sticky = navbar.offsetTop;

      function sticky() {
        if (window.pageYOffset >= sticky) {
          navbar.classList.add("sticky");
        } else {
          navbar.classList.remove("sticky");
        }
      }
    </script>
  </div>
</body>
</html>

```

Figure 76: views/shopping.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NFC Savers</title>
  <link rel="stylesheet" href="css/user1.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh8JmHjY6hW+ALEwIH" crossorigin="anonymous">
</head>
<body>
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"> <a href="/logout">Log Out</a></button>
    </div>
  </div>

  <br>

  <h1 class="right">Points: <%= points %></h1><h1 class="left">Welcome, Andy</h1>

  <br>
  <br>
  <br>

  <!-- create and view shopping list -->
  <div>
    <p class="welcome">
      Check out your special deals as well as our stock, If you want to make a shopping list click <a href="/shopping"
      >here</a></p>
    </div>

    <!-- deals table -->
    <div>
      <h1>Deals</h1>
      <table class="table table-sm">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Points</th>
            <th scope="col">Price E</th>
            <th scope="col">Quantity</th>
            <th scope="col">Deal</th>
          </tr>
        </thead>
        <tbody>
          <%= randomStock.forEach(deal => {%>
            <tr>
              <th scope="row"><%= deal.name %></th>
              <td><%= deal.points %></td>
              <td><%= deal.price %></td>
              <td><%= deal.quantity %></td>
              <td><%= deal.deal %></td>
            </tr>
          <%><%>
          </tbody>
        </table>
      </div>

      <!-- stock table -->
      <div>
        <h1>Products</h1>
        <table class="table table-sm">
          <thead>
            <tr>
              <th scope="col">Name</th>
              <th scope="col">Points</th>
              <th scope="col">Price E</th>
              <th scope="col">Quantity</th>
            </tr>
          </thead>
          <tbody>
            <%= allStock.forEach(stock => {%>
              <tr>
                <th scope="row"><%= stock.name %></th>
                <td><%= stock.points %></td>
                <td><%= stock.price %></td>
                <td><%= stock.quantity %></td>
              </tr>
            <%><%>
            </tbody>
          </table>
        </div>

      <!-- navbar stay at top -->
      <script>
        window.onscroll = function() {myFunction()};

        var navbar = document.getElementById("navbar");
        var sticky = navbar.offsetTop;

        function myFunction() {
          if (window.pageYOffset >= sticky) {
            navbar.classList.add("sticky");
          } else {
            navbar.classList.remove("sticky");
          }
        }
      </script>
    </body>
  </html>

```

Figure 77: views/user1.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NFC Savers</title>
  <link rel="stylesheet" href="css/user2.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QMTKZyypPEjISV5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JZh0JY6hWwALEwIh" crossorigin="anonymous">
</head>
<body>
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"> <a href="/logout">Log Out</a></button>
    </div>
  </div>

  <br>

  <h1 class="right">Points: <%= points %></h1><h1 class="left">Welcome, Bob</h1>

  <br>
  <br>
  <br>

  <!-- create and view shopping list -->
  <div>
    <p class="welcome">
      Check out your special deals as well as our stock, If you want to make a shopping list click <a href="/shopping"
>here:</a></p>
    </div>
    <!-- deals table -->
    <div>
      <h1>Deals</h1>
      <table class="table table-sm">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Points</th>
            <th scope="col">Price </th>
            <th scope="col">Quantity</th>
            <th scope="col">Deal</th>
          </tr>
        </thead>
        <tbody>
          <%= randomStock.forEach(deal => { %>
            <tr>
              <th scope="row"><%= deal.name %></th>
              <td><%= deal.points %></td>
              <td><%= deal.price %></td>
              <td><%= deal.quantity %></td>
              <td><%= deal.deal %></td>
            </tr>
          <%= %>
          </tbody>
        </table>
      </div>

      <div>
        <h1>Products</h1>
        <table class="table table-sm">
          <thead>
            <tr>
              <th scope="col">Name</th>
              <th scope="col">Points</th>
              <th scope="col">Price </th>
              <th scope="col">Quantity</th>
            </tr>
          </thead>
          <tbody>
            <%= allStock.forEach(stock => { %>
              <tr>
                <th scope="row"><%= stock.name %></th>
                <td><%= stock.points %></td>
                <td><%= stock.price %></td>
                <td><%= stock.quantity %></td>
              </tr>
            <%= %>
            </tbody>
          </table>
        </div>

      <script>
        window.onload = function() {myFunction()};

        var navbar = document.getElementById("navbar");
        var sticky = navbar.offsetTop;

        function myFunction() {
          if (window.pageYOffset >= sticky) {
            navbar.classList.add("sticky");
          } else {
            navbar.classList.remove("sticky");
          }
        }
      </script>
    </body>
  </html>

```

Figure 78: views/user2/ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NFC Savers</title>
  <link rel="stylesheet" href="css/user3.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QMTKZyypPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh8JMHjY6hW+ALEwIH" crossorigin="anonymous">
</head>
<body>
  <div class="navbar" id="navbar">
    <div class="navbar-left">
      <h1>NFC SAVERS</h1>
    </div>
    <div class="navbar-right">
      <button class="logout"> <a href="/logout">Log Out</a></button>
    </div>
  </div>

  <br>

  <h1 class="right">Points: <% points %></h1><h1 class="left">Welcome, Carl</h1>

  <br>
  <br>
  <br>

  <!-- create and view shopping list -->
  <div>
    <p class="welcome">
      Check out your special deals as well as our stock, If you want to make a shopping list click <a href="/shopping"
>here</a></p>
    </div>
    <!-- deals table -->
    <div>
      <h1>Deals</h1>
      <table class="table table-sm">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Points</th>
            <th scope="col">Price E</th>
            <th scope="col">Quantity</th>
            <th scope="col">Deal</th>
          </tr>
        </thead>
        <tbody>
          <%=RandomStock.forEach(deal => {%)
            <tr>
              <th scope="row"><%= deal.name %></th>
              <td><%= deal.points %></td>
              <td><%= deal.price %></td>
              <td><%= deal.quantity %></td>
              <td><%= deal.deal %></td>
            </tr>
          <%=}%>
        </tbody>
      </table>
    </div>

    <div>
      <h1>Products</h1>
      <table class="table table-sm">
        <thead>
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Points</th>
            <th scope="col">Price E</th>
            <th scope="col">Quantity</th>
          </tr>
        </thead>
        <tbody>
          <%=AllStock.forEach(stock => {%)
            <tr>
              <th scope="row"><%= stock.name %></th>
              <td><%= stock.points %></td>
              <td><%= stock.price %></td>
              <td><%= stock.quantity %></td>
            </tr>
          <%=}%>
        </tbody>
      </table>
    </div>

    <script>
      window.onscroll = function() {myFunction()};

      var navbar = document.getElementById("navbar");
      var sticky = navbar.offsetTop;

      function myFunction() {
        if (window.pageYOffset >= sticky) {
          navbar.classList.add("sticky");
        } else {
          navbar.classList.remove("sticky");
        }
      }
    </script>
  </body>
</html>

```

Figure 79: views/user3.ejs