AN ELECTRONIC COMMERCE WEBSITE FOR A BOOK SHOP USING EXPRESS.JS SERVER WITH CUSTOMER CHURN PREDICTION USING NEURAL NETWORK

Name: Chan Khai Shen

Coventry Student ID: 11323943

Supervisor Name: Dr. Hing Ratana

Bachelor of Science with Honours in Computer Science, 3+0 in Collaboration with Coventry University, UK

School of Computing, INTI International College Penang

21st June 2024

6001CEM Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Date: 21 June 2024

Please complete all fields.

First Name:	Khai Shen
Last Name:	Chan
Student ID number	11323943
Supervisor Name	Dr. Hing Ratana

This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.

Acknowledgement

I would like to thank my supervisor, Dr. Hing Ratana for guidance given when I am conducting my project.

Table of Contents

1. INTRODUCTION 1

1.1. Introduction 1

1.2. Problem Statement 2

1.3. Detailed Research Question 2

1.4. Objective 3

1.5. Client, Audience and Motivation 3

1.6. Scope 4

2. LITERATURE REVIEW 5

2.1. Electronic Commerce 5

2.1.1. Definition of Electronic Commerce 5

2.1.2. Business-to-consumer (B2C) electronic commerce 5

2.1.3. Contributing Factors for Electronic Commerce 6

2.1.4. Advantages of Electronic Commerce 6

2.2. Electronic Commerce Website 7

2.2.1. Existing Works 7

2.2.2. Process of Electronic Commerce Website 9

2.2.3. Comparison of Existing Works and the Proposed System 9

2.3. Customer Churn Prediction in Electronic Commerce using Neural Network 10

2.3.1. Customer Churn Prediction in Electronic Commerce 10

2.3.2. Existing Works 10

2.3.3. Artificial Neural Network 13

3. METHODOLOGY 16

3.1. Iterative Model 16

3.2. Gannt Chart 16

3.3. Use Case Diagrams 18

3.4. The Customer Churn Prediction Model 19

3.4.1. Training of the Prediction Model 20

3.4.2. Way of Getting the Attributes for Prediction 20

3.5. Entity Relationship Diagram 24

3.6. Components Involved in the Proposed System 26

3.6.1. Express.js, cross origin resource sharing and “nodemon” 26

3.6.2. MongoDB using “mongoose” 27

3.6.3. Environment Variables using “dotenv” 27

3.6.4. Top Navigation Bar 27

3.6.5. Login Authentication using Basic Authentication 28

3.6.6. Hashing Password using PBKDF2 28

3.6.7. HTML Form and the “axios” module 28

3.6.8. HTML Table 29

3.6.9. Uploading, Previewing and Displaying Image 29

3.6.10. Dumping of Dataset from CSV file into MongoDB Database 30

3.6.11. Training, Saving and Loading of the Model using TensorFlow.js 30

3.6.12. HTML Get Query Parameter from URL 31

3.6.13. Grid Display of List Items 31

3.6.14. Promise and Promise.all 32

4. IMPLEMENTATION 33
4.1. Customer Churn Prediction Feature 33
4.1.1. Analyzing the Training Dataset 33
4.1.2. Dumping of Training Datasets into the Database 34
4.1.3. Training of the Prediction Model 36
4.1.4. Save Predictors 38
4.1.5. Get Predictors 41
4.1.6. Make predictions 43
4.1.7. Churn Predictions Page 44
4.2. Other features 44
4.2.1. Inventory Page 44
4.2.2. Login 48
4.2.3. Home Page 50
4.2.4. Customer Register Page 52
4.2.5. Item Details Page 54
4.2.6. Cart Page 56
4.2.7. Orders Page 58
5. TESTING 61
5.1. Unit Testing 61
5.2. Black box Testing 66
5.3. Testing the Customer Churn Prediction Model 67
6. DISCUSSIONS 71
7. CONCLUSION 72

REFERENCES 73

List of Table

Table 2.1 Comparison of features in the existing works (electronic commerce website) and the proposed system 9

Table 2.2 Comparison of existing works (churn prediction) 13

Table 3.1 Explanation of the attributes used for churn prediction 21

Table 3.2 Customers Collection 25

Table 3.3 Items Collection 25

Table 3.4 Administrators Collection 25

Table 3.5 Orders Collection 26

Table 3.6 Training Datasets Collection 26

Table 5.1 Unit testing of get predictors 61

Table 5.2 Unit testing of customer register page 62

Table 5.3 Unit testing of add item page 64

Table 5.4 Unit testing for edit item page 65

Table 5.5 Unit testing for cart page 65

Table 5.6 Unit testing for order details page 66

Table 5.7 Black box testing of save predictor function 66

Table 5.8 Black box testing of update stock function 67

List of Figure

Figure 2.1 Process of Electronic Commerce Website 9

Figure 2.2 Internal activity of neuron (Dongare, et al., 2012) 14

Figure 2.3 Activity of feed-forward neural network (Dongare, et al., 2012) 15

Figure 3.1 Iterative model 16

Figure 3.2 Gannt Chart 17

Figure 3.3 Screenshot of GitHub commit 17

Figure 3.4 Administrator Use Case Diagram 18

Figure 3.5 Customer Use Case Diagram 19

Figure 3.6 Entity Relationship Diagram 24

Figure 4.1 Dataset information 33

Figure 4.2 Dataset minimum and maximum values 34

Figure 4.3 Dumping of training dataset to database 35

Figure 4.4 Storing the training dataset in MongoDB database 36

Figure 4.5 Loading of training dataset from database 37

Figure 4.6 Training of the prediction model 38

Figure 4.7 Save logout time stamp 39

Figure 4.8 Save interaction time stamp 40

Figure 4.9 Save transaction time stamp and amount 41

Figure 4.10 Get predictors (part 1 of 2) 42

Figure 4.11 Get predictors (part 2 of 2) 42

Figure 4.12 Make predictions 43

Figure 4.13 Churn predictions page 44

Figure 4.14 Inventory page 45

Figure 4.15 Add item page (part 1 of 2) 45

Figure 4.16 Add item page (part 2 of 2) 46

Figure 4.17 Add item success page 46

Figure 4.18 Edit item page (part 1 of 2) 47

Figure 4.19 Edit item page (part 2 of 2) 47

Figure 4.20 Edit item success page 48

Figure 4.21 Login 49

Figure 4.22 Handle login 50

Figure 4.23 Home page 51

Figure 4.24 Check login 51

Figure 4.25 Home page (logged in) 52

Figure 4.26 Customer register page 53

Figure 4.27 Customer register success page 53

Figure 4.28 Item details page 54

Figure 4.29 Login prompt upon clicking on "Add To Cart" button 55

Figure 4.30 Add to cart success page 55

Figure 4.31 Cart page 56

Figure 4.32 Cart page (Empty cart) 57

Figure 4.33 Place order success page 57

Figure 4.34 Orders page (administrator side) 58

Figure 4.35 Orders page (customer side) 59

Figure 4.36 Orders page (no past orders) 59

Figure 4.37 Order details (administrator side) 60

Figure 4.38 Order details (customer side) 60

Figure 5.1 Test prediction model script (part 1 of 2) 68

Figure 5.2 Test prediction model script (part 2 of 2) 69

Figure 5.3 Test prediction model result 69

Figure 5.4 Confusion matrix 70

List of Equation

Equation 1 Summing function (Dongare, et al., 2012) 14

Equation 2 Cost function (Seymen, et al., 2023) 15

Equation 3 Accuracy (Ismail, et al., 2015) 70

List of Abbreviation

Abbreviation	Full form
CRUD	Create, Read, Update and Delete
CSV	Comma-Separated Value
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
JSON	JavaScript Object Notation
PBKDF2	Password-Based Key Derivation Function 2
ReLU	Rectified Linear Unit
RMSProp	Root Mean Square Propagation
SHA-256	Secure Hash Algorithm - 256

Abstract

When there is no online book shop, sometimes when the customer visits the physical shop, the shop is closed. By having an electronic commerce book shop, the customer can browse through the items sold at their convenience and is not restricted to the opening hours of the physical shop. So, to solve this problem, an electronic commerce book shop using Express.js is developed. The electronic book store allows the administrator to update stock and view orders. On the other hand, the customer can browse through items selling on the website, choose item and place order. Customer churn is a condition when a customer stop buying a product or service. Customer churn can cause significant revenue lost to the company. To reduce churn, customer retention is an approach which costs less than customer acquisition. To conduct targeted retention on the customer who is likely to churn in near future, customer churn prediction is needed. Customer churn prediction is a prediction of whether a customer will interact with the online book shop in the next month or not. So, as a solution, churn prediction based on neural network is implemented on the electronic commerce book shop. The churn prediction model is pre-trained and saved on the server by using TensorFlow.js. The training dataset is obtained from Kaggle. The input attributes are shortlisted into only 12 attributes. The attributes are selected based on the ability of the proposed system to capture the attributes. The logout time stamp, user-application interaction (view item and add to cart) time stamp and transaction time stamp and amount, of each customer, are recorded. The 12 input attributes are computed based on the aforementioned recorded attributes. The churn probability of each customer is displayed in a table to the administrator. In the testing of the customer churn prediction model, the model is tested with the training dataset and the number of true positives, the number of false positives, the number of true negatives and the number of false negatives is calculated. The confusion matrix is drawn. The accuracy of the customer churn prediction model is 0.861 or 86.1%. The accuracy of the churn prediction model is not too high and not too low compared with the accuracy of the reviewed works, which is ranged between 0.740 and 0.976.

1. INTRODUCTION

1.1. Introduction

Lately, there is tremendous growth in electronic commerce sector. The turnover of electronic commerce sector has reached 3 trillion USD per year globally, while the growth of the fastest growing market, Asia, is about 30% (Pondel, et al., 2021). From this information-rich sentence, it can be seen that nowadays how large the electronic commerce sector is, especially in Asia. In Malaysia, the value of the electronic commerce industry is about $ 2 billion in 2020 and the predicted value in 2025 is $ 3.6 billion (Tang & Ya'acob, 2023). The growth of electronic commerce sector is supported by the government policies, as the government believes that the electronic commerce sector can be the creator of new job opportunities and the driving factor of economic growth (Tang & Ya'acob, 2023). Due to the fact that electronic commerce involves websites and smartphone applications, the amount of customer data collected in the database of companies were increased. This provides sufficient data for companies to implement customer analytics. Common customer analytics covers customer segmentation, lifetime value modelling and churn analysis (Seymen, et al., 2023).

Churn prediction is one of the activities under customer analytics. A churned customer is a customer who discontinued the product or service offered by a company. In electronic commerce sector, the definition of customer churn is slightly different. Fridrich & Dostal (2022) defines predicting customer churn in an electronic commerce website as predicting whether a customer will interact with the website in the next month or not. Customer churn can result in significant revenue lost (Tang & Ya'acob, 2023). Severe customer churn can impact the chance of survival of a company (Tariq, et al., 2021), So, churn analysis is an important indicator of a company's performance. A study found that the customer churn rate in online retail sector is around 21% (Seymen, et al., 2023). Another study found that the customer churn rate in electronic commerce sector in Malaysia is about 20% (Tang & Ya'acob, 2023). Both of these statistics show that customer churn is an issue which needs concern from companies that ventures in electronic commerce retail.

Researches had found that customer acquisition costs more than customer retention. A study by Amy Gallo (2014) found that acquiring a customer is anywhere 5-25 times more expensive than retaining an existing one. Another study found that capturing a customer is ten times more costly than retaining an existing one (Pondel, et al., 2021). These statistics shows how important customer retention is. According to Ismail, et. al. (2015), conducting customer retention program on the whole customer base is expensive, so to simplify the problem,

companies can carry out customer retention only on specific groups of customers who have risk of churning. One way to identify customer who is likely to churn is by conducting churn prediction. So, in this project, an electronic commerce website will be developed and a churn prediction model, which ingests the customer behavioral data collected by the proposed system, will be built.

1.2. Problem Statement

One problem of physical shop is sometimes when customer go to the shop, the shop is closed. Comparing to online shop, there is no restriction of opening hours, as online shop can be opened 24 hours. So, by having online book shop, the customer can browse through and select items at their convenience, and is not restricted to the opening hours of the book shop. In addition, by having an online presence, the book shop will stand an extra chance to gain a new group of customers who might not be visiting the physical shop. Besides that, having an online book shop can reduce human error because the ordering process is automated. As a solution, an online book shop website, which allows the book shop owner to upload books and for customers to view the books and place order on the website, will be built.

Customer churn is a condition where customer, who currently buying certain product, chooses not to continue buying or subscribing the product. Customer churn can cause a significant revenue loss to the company (Tang & Ya'acob, 2023). The customer churn rate in online retail sector is around 21% (Seymen, et al., 2023). This shows that customer churn is an issue that needs attention from the company. A study by Gallo (2014) found that customer acquisition is 5-25 times more expensive than customer retention. So, to save cost, it is better to try to retain the existing customer than capturing a new one. To identify those customers who have the potential to churn in near future, a customer churn prediction model will be built and integrated into the online book shop website, so that predictions can be made based on the customer generated data on the website. The probability of churn of each customer will be displayed to the bookstore administrator in table form. The administrator can then take appropriate actions after gaining more insights from the predictions.

1.3. Detailed Research Question

1. How can an electronic commerce website for a book shop be built?

2. How can customer churn prediction model for electronic commerce based on neural network be built and integrated into the website?

1.4. Objective

1. To develop an electronic commerce website for a book shop using Express.js, which allows the administrator to create stock and upload item description, and the customer to choose and place order online
2. To build a customer churn prediction model for electronic commerce based on neural network and integrate the model into the online book shop website, by displaying the predictions for each customer in a table, to the administrator

1.5. Client, Audience and Motivation

The two main clients of the proposed system are:

- Book shop customer
- Book shop owner

The book shop customer can shop for the books on the proposed system at their convenience. They are not restricted by their physical location and the opening hour of the book shop. In other words, the online book shop will be available at anywhere and anytime. This allows the customer to shop whenever they have free time and at any place.

The book shop owner can upload the books onto the proposed system for the customer to shop for. This gives the book shop opportunity to reach a larger community of customers beyond the physical location of the shop. This will bring in more business for the book shop.

In addition, the proposed system provides customer churn prediction for the book shop owner. This allows the proposed system to predict the probability of churning of each customer. This provides insights to the book shop owner, which allows the book shop owner to arrange strategy to retain certain group of customers. For example, the book shop may be carrying out targeted promotions on certain books.

1.6. Scope

The scope of the proposed system is limited to one book shop only. The proposed system consists of a server that renders the website for all users to use the proposed system. The proposed system has two types of users, namely customer and administrator.

The customer can view the list of items and the details of each item. Login is not needed for the aforementioned functionalities. Login is needed for the following functionalities. The customer needs to register before login. The customer can add an item to the cart and view the items in the cart. From cart, the customer can place order.

Login is needed to access the administrator page. The administrator is pre-registered from the coding of the proposed system. The administrator can upload image, description, price and quantity for each item. The proposed system only allows the administrator to upload one image for each item. The administrator can view list of orders and view the details of an order.

The proposed system will save the customer's logout timestamp, user-application interaction and transaction information to generate the attributes used to predict customer churn. The customer churn is defined as whether the customer will interact with the website in the next month. The administrator can view the prediction of churning of each customer in a table.

2. LITERATURE REVIEW

2.1. Electronic Commerce

This subchapter discusses about definition, business-to-consumer electronic commerce, and advantages of electronic commerce.

2.1.1. Definition of Electronic Commerce

Electronic commerce, or e-commerce, is the process of buying and selling, which involves the transfer of ownership or right to use goods or services, over a computer-mediated network (Gupta, 2014). Electronic commerce can take place through websites, smartphone applications or social media. The definition of electronic commerce has not been agreed upon and there are different descriptions from different point of view.

From communication point of view, electronic commerce involves the transmission of information, product or service through telephone line or computer networking (Jain, et al., 2021). Electronic commerce provides a different way of the exchange of information, which is through network, instead of the traditional way, which is through face-to-face meeting.

From business process point of view, electronic commerce promotes the automation of the transaction and workflow of a company (Jain, et al., 2021). Electronic commerce provides the facility that supports one-time and automated entry of transaction information, which eliminates the traditional way of manual, paper and pen record of transaction.

In a nutshell, the further definition of electronic commerce is the utilization of electronic communication and digital information processing techniques to create, change and redefine the value creation among companies, or between company and individual (Gupta, 2014). In other words, electronic commerce uses information and communication technology (ICT) to create new way of doing business among companies or between company and end consumer.

2.1.2. Business-to-consumer (B2C) electronic commerce

Electronic commerce has several types, namely business-to-business (B2B) and business-to-consumer (B2C). B2B electronic commerce is the electronic commerce between companies. B2B electronic commerce typically involves trading between producer and wholesale company (Jain, et al., 2021). On the other hand, B2C electronic commerce is the

electronic commerce between consumer and company. B2C electronic commerce involves the business of retailer to end consumer (Gupta, 2014).

The type of electronic commerce in the proposed system is B2C electronic commerce. B2C electronic commerce increases the ease of customer getting information because information of each item is readily available on the website. This allows the customer to compare prices across different websites.

2.1.3. Contributing Factors for Electronic Commerce

One contributing factor of electronic commerce is the penetration of internet (Jain, et al., 2021). Internet is essential for the users to access the electronic commerce website. With internet, anyone can access the online shop at anywhere and anytime. Oppositely, without internet, the online shop cannot be accessed. The increase in internet subscribers also means the increase in potential customers to the online shop. So, the growth in the internet community contributes to the growth of electronic commerce industry.

Another contributing factor of electronic commerce is the emerge of digital payment gateways. The availability of online banking, credit card and debit card contribute to the success of electronic commerce (Jain, et al., 2021). In other words, non-cash payment gateway is an important part of electronic commerce as it provides a method for the customer paying without going to the physical shop.

2.1.4. Advantages of Electronic Commerce

One advantage of electronic commerce for the consumer is the availability of the shop. The consumer can shop online at anywhere at anytime. In other words, consumer can view the items at the online shop at their convenience and is not restricted by the opening hours of the shop.

Another advantage for the consumer is the availability of information of goods. The consumer can read the detailed information of each item on the website before they decide to buy the item.

An advantage for both the consumer and the company is the reduction of error through automation of process. The process of selling and buying is automated, and the entry of information is done once, so the probability of human error is reduced.

One advantage for the company is expansion of business regardless of location. Through online shop, the company can reach a larger community of consumer, who might not be visiting their physical shop because of not living near to their physical shop.

2.2. Electronic Commerce Website

This subchapter reviews related works, analyses processes of an electronic commerce website, and compares related works and the proposed system.

2.2.1. Existing Works

Sharma (2021) developed and designed the customer-facing side of an electronic commerce website using Hyper Text Markup Language (HTML), Cascading Style Sheet (CSS) and JavaScript. The website sells a variety of phones (Sharma, 2021). There are several pages, such as home, category, blog, contacts, cart, product information tab, etc. (Sharma, 2021). The website allows user to choose items and add those items to cart, then proceed to payment (Sharma, 2021). There are also search bar for user to search for a product (Sharma, 2021). The website allows administrator to add events, such as summer sales, winter festival, etc. (Sharma, 2021). Sharma (2021) had made the website responsive by using HTML, CSS, JavaScript and images.

Naidu, et. al. (2021) developed an electronic commerce website by using MongoDB, Express.js framework, React.js library and Node.js platform (MERN). The website sells t-shirts (Naidu, et al., 2021). The customer can go to the home page to access the items on sale, and the price and description is available on this page (Naidu, et al., 2021). Then, the customer can add the selected item to cart (Naidu, et al., 2021). The website also allows the customer to contact the administrator if they have any issue (Naidu, et al., 2021). The administrator has a dashboard page which allows them to add items (Naidu, et al., 2021). Naidu, et. al. (2021) used MongoDB as the database, Express.js library for routing and React.js library for user interface.

AbdAlameer (2014) built and developed a reliable online retail website based on electronic commerce theories. The website sells software and hardware (AbdAlameer, 2014). AbdAlameer (2014) had recognized that search engine optimization, proper debit card and credit card payment gateway and good implemented security features are important aspects for an online retail website. The website allows customer to choose category from the home

page and the list of items in that category will be displayed (AbdAlameer, 2014). The customer can add any item to the shopping cart and the selected items, together with price, and quantity, which will be listed on the shopping cart page (AbdAlameer, 2014). From the shopping cart page, the customer can click on the proceed to check out button to go to the billing information page where the customer's name, address, total price, account number, email address and phone number will be displayed (AbdAlameer, 2014). As the final step, the customer can click on the place order button to proceed and the delivery information will be displayed (AbdAlameer, 2014). The website used PayPal as the payment gateway (AbdAlameer, 2014). AbdAlameer (2014) used PHP and MySQL to build the website.

Ariyani, et. al. (2015) developed an electronic commerce website for Wiga Art, a handicraft shop in Pacitan, Indonesia. The customer can view the images, price and description of all goods at the home page (Ariyani, et al., 2015). The website applied multiple prices, which means the price is different for member and non-member, as well as price range manager, which means the price is different for buying one item and 10 items (Ariyani, et al., 2015). The website also applied auto calculation of shipping cost and multiple currencies (Ariyani, et al., 2015).

Soegoto & Suripto (2018) designed a prototype for an online shopping website. In this website, the process starts when the administrator uploads the goods to the website, then when a customer chooses an item, the administrator creates a list of reservations and passes the list to the warehouse, and if the items are available, the warehouse creates duplicate notes, which is passed to the customer and to the archive (Soegoto & Suripto, 2018). After that, the administrator gives a memorandum of payment to the customer, then the customer pays to the shop's bank account and provides a proof of payment (Soegoto & Suripto, 2018). After the checking and confirming the proof of payment, the administrator proceeds with the delivery of the items (Soegoto & Suripto, 2018).

Tenzin, et. al. (2022) designed and developed an electronic commerce website for an on-campus cooperative store to manage ordering and home delivery of goods. Tenzin, et. al. (2022) used Laravel framework and followed rapid application development (RAD) methodology during development of the website. The website allows both registered and non-registered customer to view the goods available, but only permits registered customer to place order (Tenzin, et al., 2022). The customer can also manage their profile, view their past records and get notification about their orders (Tenzin, et al., 2022). On the other hand, the administrator and staff can make stock entry, view orders, view monthly or annual report and get notification when the stock is low or there is a new order (Tenzin, et al., 2022).

2.2.2. Process of Electronic Commerce Website

After reviewing the existing works, it can be concluded that the general process of an electronic commerce website consists of a few steps, namely upload item, view item, add to cart and check out for payment, as depicted in Figure 2.1. In the upload item step, the administrator uploads the item to the website. Under this step, the administrator is able to do the create, read, update and delete (CRUD) operations on the items. In the view item step, the customer views the item on the website. Under this step, the customer is also allowed to search and sort the items. Optionally, there are also promotion events or categorization of the items. In the add to cart step, the customer adds the item he wants to the shopping cart. Under this step, the item wanted by the customer are listed and the total price of the items is displayed. In the check out for payment step, the customer pays for the item using online payment method, such as PayPal or Stripe. One of websites reviewed used the cash on delivery option which does not require online payment.

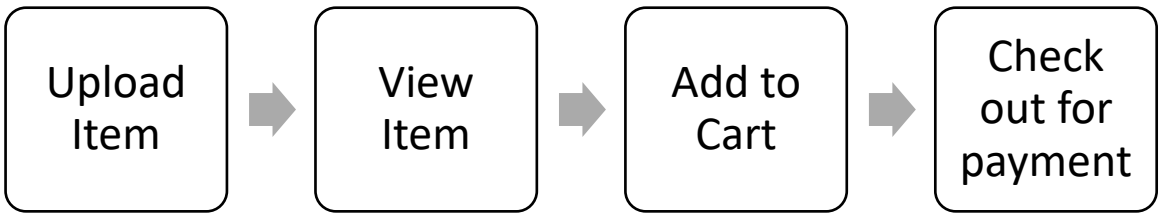


Figure 2.1 Process of Electronic Commerce Website

2.2.3. Comparison of Existing Works and the Proposed System

Table 2.1 shows the comparison of existing works and proposed system based on features.

Table 2.1 Comparison of features in the existing works (electronic commerce website) and the proposed system

Author	**Add, edit and delete item**	**Browse items**	**Add item to cart**	**Check out for payment**	**Customer churn prediction**
(Naidu, et al., 2021)	Yes	Yes	Yes	Yes (Stripe Payment)	No
(AbdAlameer, 2014)	Yes	Yes	Yes	Yes (PayPal)	No
(Ariyani, et al., 2015)	Yes	Yes	Yes	No	No

(Sharma, 2021)	Yes	Yes	Yes	No	No
(Soegoto & Suripto, 2018)	Yes	Yes	No	No	No
(Tenzin, et al., 2022)	Yes	Yes	No	No	No
The proposed system	Yes	Yes	Yes	No	Yes

Based on Table 2.1, all of the existing work has add, edit and delete item and browse item. Most of the existing work (4 out of 6) has add item to cart. Some of the existing works (2 out of 6) has check out for payment. None of the works reviewed has customer churn prediction. The customer churn prediction feature is the key contribution of the proposed system. Besides that, the proposed work has the features which most of the existing works have (add, edit and delete item, browse item and add item to cart).

2.3. Customer Churn Prediction in Electronic Commerce using Neural Network

In this subchapter, three subtopics are discussed, namely churn prediction in electronic commerce, existing works and artificial neural network.

2.3.1. Customer Churn Prediction in Electronic Commerce

Customer churn is a condition when a customer stopped buying a product or service. Fridrich & Dostal (2022) defines a churned customer in electronic commerce sector as a customer who will not interact with the website in the following month. In electronic commerce retail sector, a customer can compare between many websites, so it is very easy for the customer to switch to a different site. A study by Amy Gallo (2014) found that acquiring a customer is anywhere 5-25 times expensive than retaining an existing one. So, customer churn analysis becomes important because it helps in customer retention.

2.3.2. Existing Works

Fridrich and Dostal (2022) built machine learning model for customer churn prediction in electronic commerce sector by using logistic regression, support vector machine, multi-layer perceptron (neural network), random forest and gradient boosting machine. Fridrich & Dostal

(2022) defined churn as whether the customer will interact with the website in the next month or not. The models used attributes formed around user-item interactions, which is data generally available to any electronic commerce retail store (Fridrich & Dostal, 2022). In the experiment, Fridrich and Dostal (2022) found that logistic regression, support vector machine and gradient boosting machine are almost on par in the accuracy and F-1 score. The accuracy for neural network (multi-layer perceptron) is 0.8762 (Fridrich & Dostal, 2022).

Tariq, et al. (2021) built a customer churn prediction model for the telecommunication sector using 2-dimentional convolutional neural network. The model has 8 filter with size (2, 2) with Rectified Linear Unit (ReLU) as the activation function and a dropout layer (Tariq, et al., 2021). The model achieved accuracy of 0.963 and validation loss of 0.004 (Tariq, et al., 2021).

Seymen, et. al. (2023) compared the customer churn prediction models for electronic commerce sector built based on convolutional neural network and artificial neural network. The prediction models predicted customer churn based on the customers' demographic information, such as age, marital status and job, as well as the customers' behavioral information, such as number of purchasing and spendings (Seymen, et al., 2023). Convolutional neural network is actually used to process image. Seymen, et. al. (2023) converted the 567 input features into 21 * 27 matrix so that it can be ingested by the 2-dimentional convolutional layer. The experiment found that convolutional neural network has higher accuracy (0.9762) than that of artificial neural network (0.96) (Seymen, et al., 2023).

Pondel, et. al. (2021) built and compared neural network models for customer churn prediction in the electronic commerce sector by using different configurations. The prediction models predicted the customer churn based on the customers' previous transactions, which means the buying record of each product category of each customer (Pondel, et al., 2021). The experiment is conducted on 16 sets of models, with and without recurrent layer, with and without a dropout layer and with Rectified Linear Unit (ReLU) and with Leaky ReLU (Pondel, et al., 2021). The experiment found that one-layer recurrent model without dropout and with ReLU performs better than one-layer dense model with 30% dropout and Leaky ReLU (Pondel, et al., 2021). Based on the experiment results among dense models, it can also be interpreted that one-layer dense model without dropout with ReLU has higher accuracy (0.737) than that of two-layer dense model with dropout with Leaky ReLU (0.733) (Pondel, et al., 2021).

Ismail, et. al. (2015) built and compared customer churn prediction models for the telecommunication sector based on neural network, multiple regression and logistic

regression. The prediction models used customer demographic data, customer relationship data, billing data and usage data to predict customer churn (Ismail, et al., 2015). The models were compared based on accuracy, sensitivity (proportion of predicted churn to actual churn) and specificity (proportion of predicted non-churn to actual non-churn) (Ismail, et al., 2015). The experiment found that neural network (multi-layer perceptron) achieved overall accuracy of 91.28%, which is better than multiple regression (78.84%) and logistic regression (75.19%) (Ismail, et al., 2015).

Tang & Ya'acob (2023) built and compared customer churn prediction model for the electronic commerce sector based on decision tree, random forest, Naïve Bayes, logistic regression, K-Nearest Neighbors and extreme gradient boosting. The evaluation metrics are accuracy, precision, recall and F-1 score (Tang & Ya'acob, 2023). The research found that attributes such as amount of consumption, number of days since last purchase, frequency of purchasing and enjoyment / ratings are among the important attributes for predicting customer churn in electronic commerce (Tang & Ya'acob, 2023).

Jaeyalakshmi, et. al. (2020) built and compared customer churn prediction model for the electronic commerce sector based on support vector machine, random forest, K-Nearest Neighbors, logistic regression, gradient boosting and extreme gradient boosting. The experiment found that both random forest and gradient boosting achieves the highest accuracy (0.96).

Shaaban, et. al. (2012) built and compared customer churn prediction models for the telecommunication sector based on decision tree, support vector machine and neural network, then clustered the churned customers into three clusters using k-means technique. Shaaban, et. al. (2012) defined churned customer as a customer who left the service provider. Attributes such as number of minutes of use, number of messages and number of customer care calls as the input attributes of the prediction model (Shaaban, et al., 2012). The comparison of the three prediction models found that neural network and support vector machine has better performance than that of decision tree, both achieving accuracy of 83.7% (Shaaban, et al., 2012).

Sabbeh (2018) built and compared customer churn prediction model for the telecommunication sector based on random forest, adaptive boost, multi-layer perceptron (neural network), stochastic gradient boosting, support vector machine, k-nearest neighbor, decision tree, Naïve Bayes, logistic regression and linear discriminant analysis. The models used attributes such as number of customer service calls and number of day minutes as the input attributes (Sabbeh, 2018). The experiment found that random forest and adaptive

boosting has highest performance, both achieving accuracy of 0.963964, whereas multi-layer perceptron (neural network) achieved accuracy of 0.944 (Sabbeh, 2018).

Domingos, et. al. (2021) built customer churn prediction models for the banking sector using neural network and compared the models based on different combinations of activation function, different batch size and different optimizers. Domingos, et. al. (2021) defined churn as the closure of bank account and used parameters such as age, balance, has credit card, etc. as the input of the prediction model. The experiment found that the accuracy of multi-layer perceptron is highest when using sigmoid activation function in the hidden layer and the output layer and when using Root Mean Square Propagation (RMSProp) optimizer (Domingos, et al., 2021). The experiment also found that the accuracy dropped when the batch size grows near to the size of the test set data (Domingos, et al., 2021).

Table 2.2 compares the existing work in terms of the sector of customer churn prediction, best accuracy and machine learning technique of best accuracy.

Table 2.2 Comparison of existing works (churn prediction)

Author	**Sector**	**Best accuracy**	**Machine learning technique of best accuracy**
(Fridrich & Dostal, 2022)	Electronic commerce	0.889	Logistic regression, Gradient boosting machine
(Tariq, et al., 2021)	Telecommunication	0.963	Convolutional neural network
(Seymen, et al., 2023)	Electronic commerce	0.976	Convolutional neural network
(Pondel, et al., 2021)	Electronic commerce	0.740	Recurrent neural network
(Ismail, et al., 2015)	Telecommunication	0.913	Neural network
(Tang & Ya'acob, 2023)	Electronic commerce	Not disclosed	Not disclosed
(Jaeyalakshmi, et al., 2020)	Electronic commerce	0.960	Random forest, Gradient boosting
(Shaaban, et al., 2012)	Telecommunication	0.837	Neural network, Support vector machine
(Sabbeh, 2018)	Telecommunication	0.964	Random forest, Adaptive boosting
(Domingos, et al., 2021)	Banking	0.869	Deep neural network

2.3.3. Artificial Neural Network

Artificial neural network, or multi-layer perceptron, is a machine learning algorithm inspired from the working methodology of the neural network in human brain. The neural

network in human brain consists of a connected network of neurons, helping human to recognize objects and make classifications (Dongare, et al., 2012). Each neuron has receivers which receive signals from other neurons, a cell body which processes the input signals and effectors which emit signals to other neurons (Dongare, et al., 2012). In an artificial neural network, each node carries out computation with a label or weight, which amplifies or diminishes an input value, and each connection passes the result of computation to one node and another (Dongare, et al., 2012). In other words, the artificial neural network mimics the human neural network, in the way how a single neuron receives, processes and emits signals and in the way how the network of neurons sends the signals to each other.

Figure 2.2 depicts the internal activity of a single neuron. The neuron receives p number of inputs and each input is multiplied with a weight. The products of input and weight, of all inputs, are summed up and passed to the activation function to produce the output of the neuron.

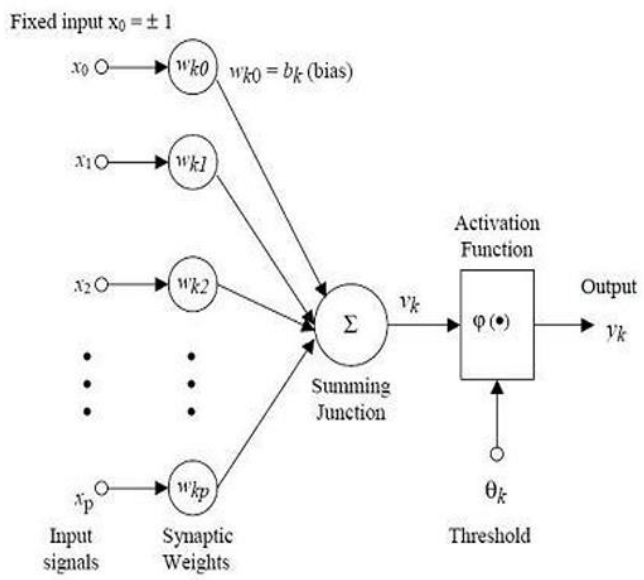


Figure 2.2 Internal activity of neuron (Dongare, et al., 2012)

The equation of the summing function is as in Equation 1. As depicted in Figure 2.2, v_k is the sum of the modified inputs, p is the number of inputs, w_k is the weight of the input and x is the input.

Equation 1 Summing function (Dongare, et al., 2012)

$$v_k = \sum_{j=1}^{p} w_{kj} x_j$$

<u>Feed forward neural network</u>

A feed forward neural network is a type of neural network which the connection is allowed only to the next layer (Dongare, et al., 2012). In the proposed work, feed forward neural network is used. Figure 2.3 depicts the activity of a feed forward neural network. The neural network depicted in Figure 2.3 has 1 input layer with 3 neurons, 1 hidden layer with 2 neuron, 1 hidden layer with 3 neurons and 1 output layer with 2 neurons.

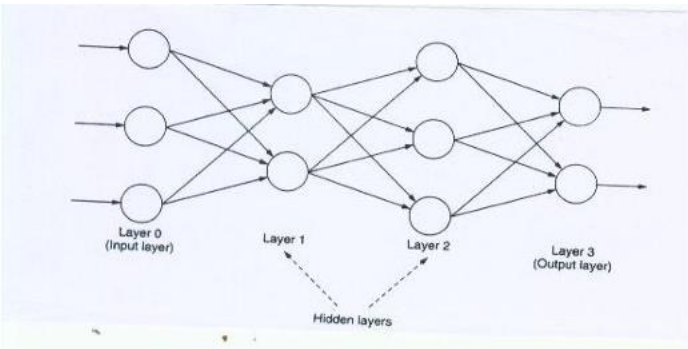


Figure 2.3 Activity of feed-forward neural network (Dongare, et al., 2012)

<u>Cross-entropy cost function</u>

The cross-entropy cost function calculates how far the predicted result differ from the actual result (Seymen, et al., 2023). The larger the value of the cost function, the further the predicted result differ from the actual result. During training of the neural network prediction model, the cost function is calculated and the purpose of training is to minimize the cost (Seymen, et al., 2023). Equation 2 is the cost function where m is the number of training samples, y is the actual output and $\hat{y}$ is the predicted output.

Equation 2 Cost function (Seymen, et al., 2023)

$$E = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\hat{y}^{(i)}) - (1 - y^{(i)})\log(1 - \hat{y}^{(i)})]$$

3. METHODOLOGY

3.1. Iterative Model

This project is developed based on iterative model. In the iterative model, the life cycle starts by specifying and implementing one part of the software and repeat the process by producing a new version of the software at the end of each iteration (Tutorials Point, n.d.). Figure 3.1 depicts the iterative model.

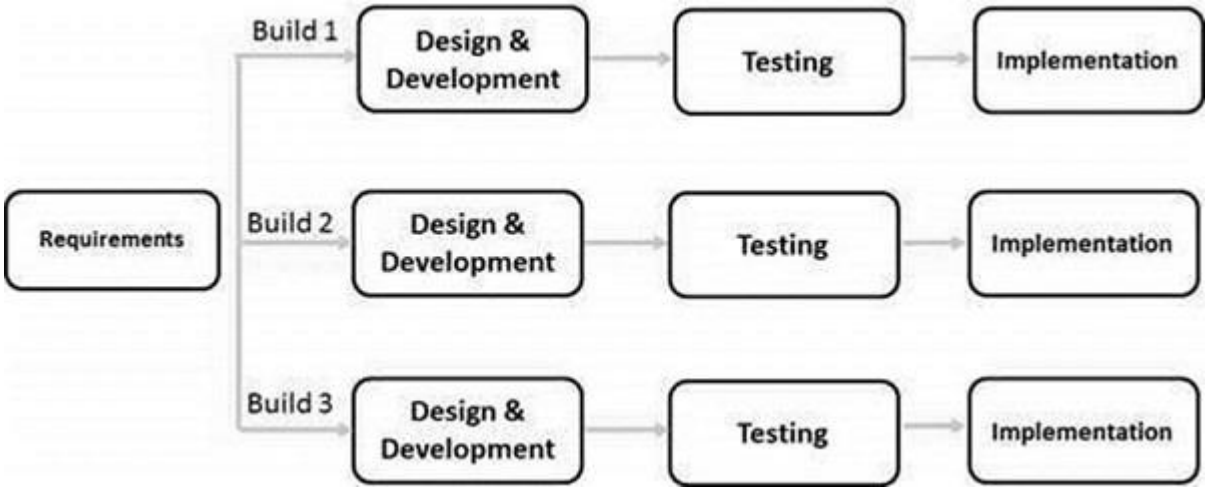


Figure 3.1 Iterative model

In this project, in the first iteration, administrator login feature, the inventory page, add item, edit item and delete item were implemented. In the second iteration, the work was focused on finding the training dataset for customer churn prediction and analyzing the training dataset. In the third iteration, the work was focused on training the customer churn prediction model and predict customer churn based on a few dummy data. In the fourth iteration, the work was focused on the save predictors function, get predictors function and the churn prediction page. In the fifth iteration, the browse item page, view item details, customer registration and add item to cart were implemented. In the sixth iteration, the view cart, reduce item quantity, increase item quantity and place order were implemented. In the seventh iteration, the orders and order details page were implemented.

3.2. Gannt Chart

Figure 3.2 is the Gannt chart of the whole project.

Item	18/4 – 2/5	3/5 – 8/5	9/5-15/5	16/5-20/5	21/5-28/5	29/5-4/6	5/6-13/6
Administrator Login	/						
Inventory page	/						
Add Item	/						
Edit Item	/						
Delete Item	/						
Find dataset		/					
Analyse dataset		/					
Dump dataset to database			/				
Train model			/				
Predict dummy data			/				
Churn predictions page				/			
Browse item page					/		
View item details					/		
Customer registration					/		
Add item to cart					/		
View cart						/	
Reduce/increase item quantity						/	
Place order						/	
Orders page							/
View order details							/

Figure 3.2 Gannt Chart

Version Control

The version control of the proposed system is done by using GitHub. Figure 3.3 is the screenshot of GitHub commits.

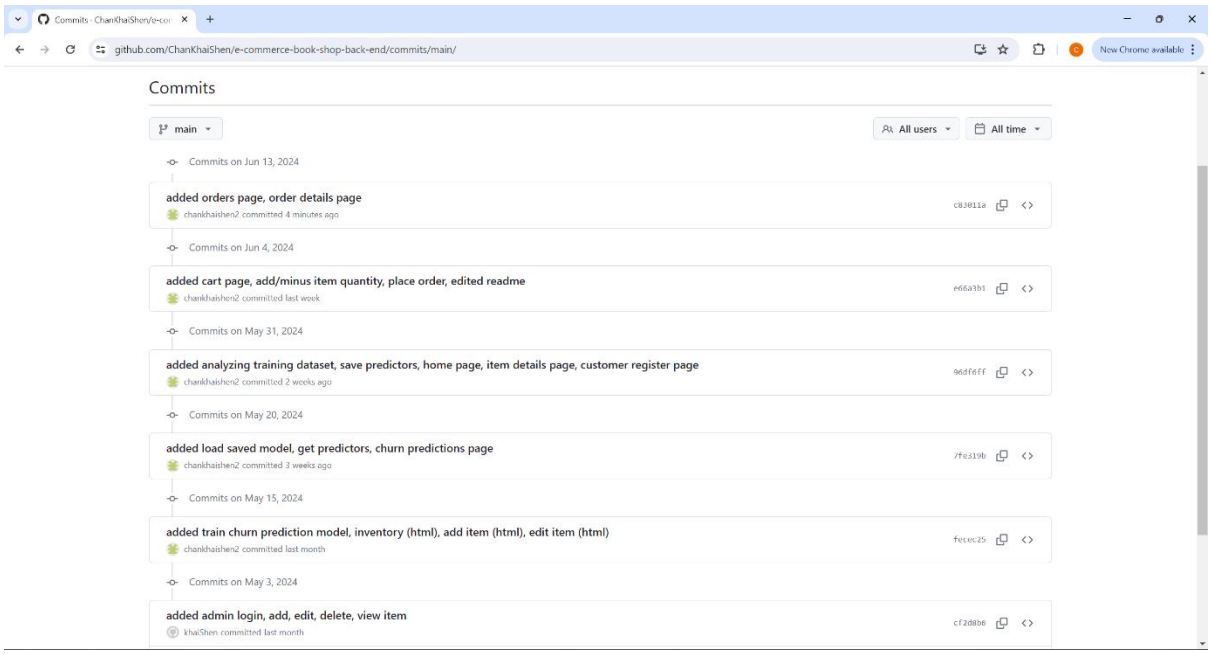


Figure 3.3 Screenshot of GitHub commit

3.3. Use Case Diagrams

Figure 3.4 is the use case diagram for administrator. After login, the administrator can go to inventory page, orders page and churn predictions page. On the inventory page, the administrator can view list of items. The administrator can delete item, edit item or add item. For adding item, the administrator needs to fill in item name, description, weight, price and quantity. Optionally, the administrator can upload image. For editing item, the administrator can edit description, weight, price or quantity. The administrator can also upload image. On the orders page, the administrator can view list of orders. The administrator can view details of one order. On the churn predictions page, the administrator can view list of customers with predictor attributes and prediction result.

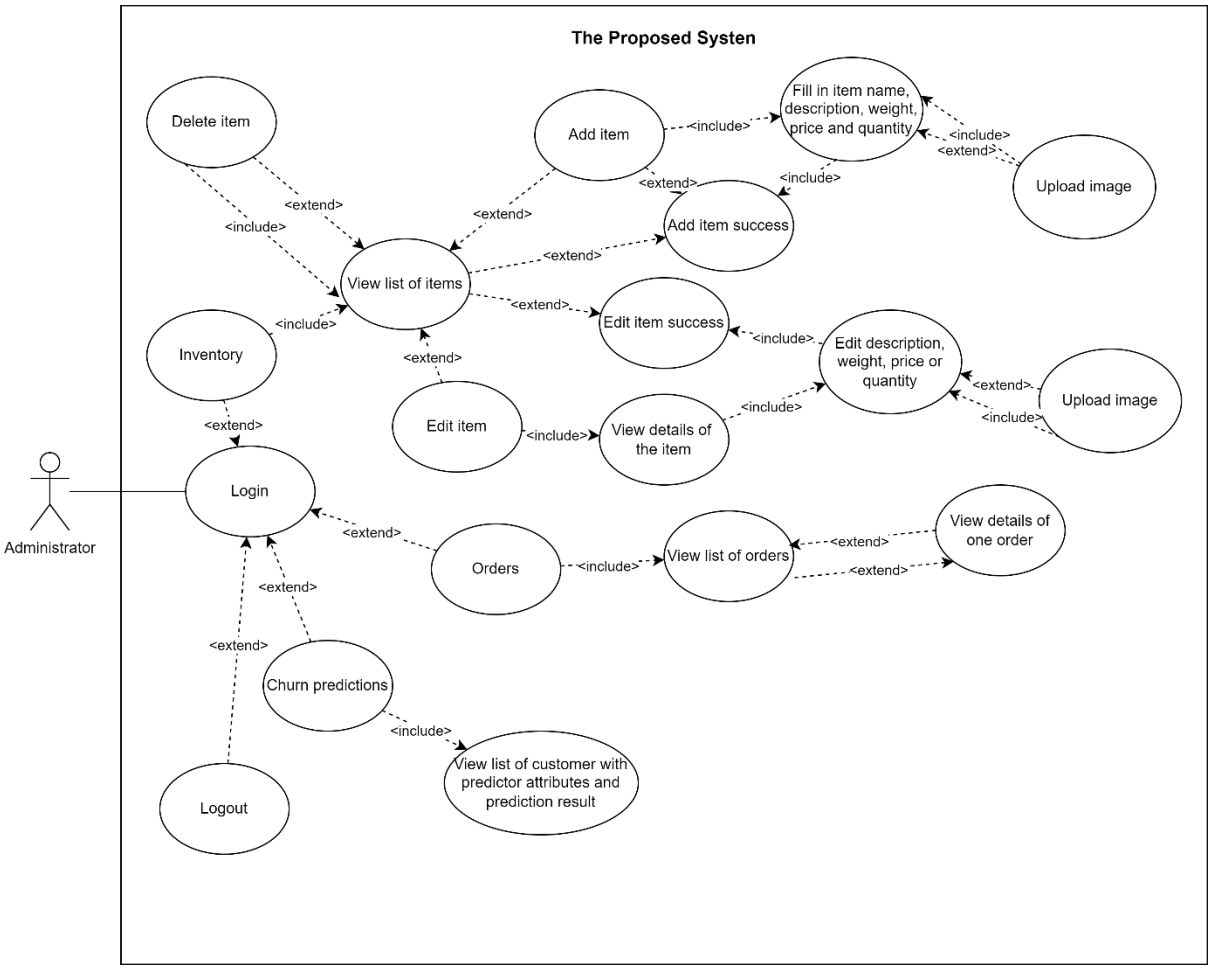


Figure 3.4 Administrator Use Case Diagram

Figure 3.5 is the use case diagram for customer. Without login, the customer can access the home page. On the home page, the customer can browse through list of items. The customer can view details of one item or register himself. During registration, the customer needs to fill in user name and password. On view item details page, the customer can add to

cart. During adding to cart, the customer needs to login. Upon login, the customer can go to the cart page and the orders page. On the cart page, the customer can view list of items and total amount. The customer can opt to reduce item quantity, increase item quantity or place order. On orders page, the customer can view list of past orders. The customer can view details of one order.

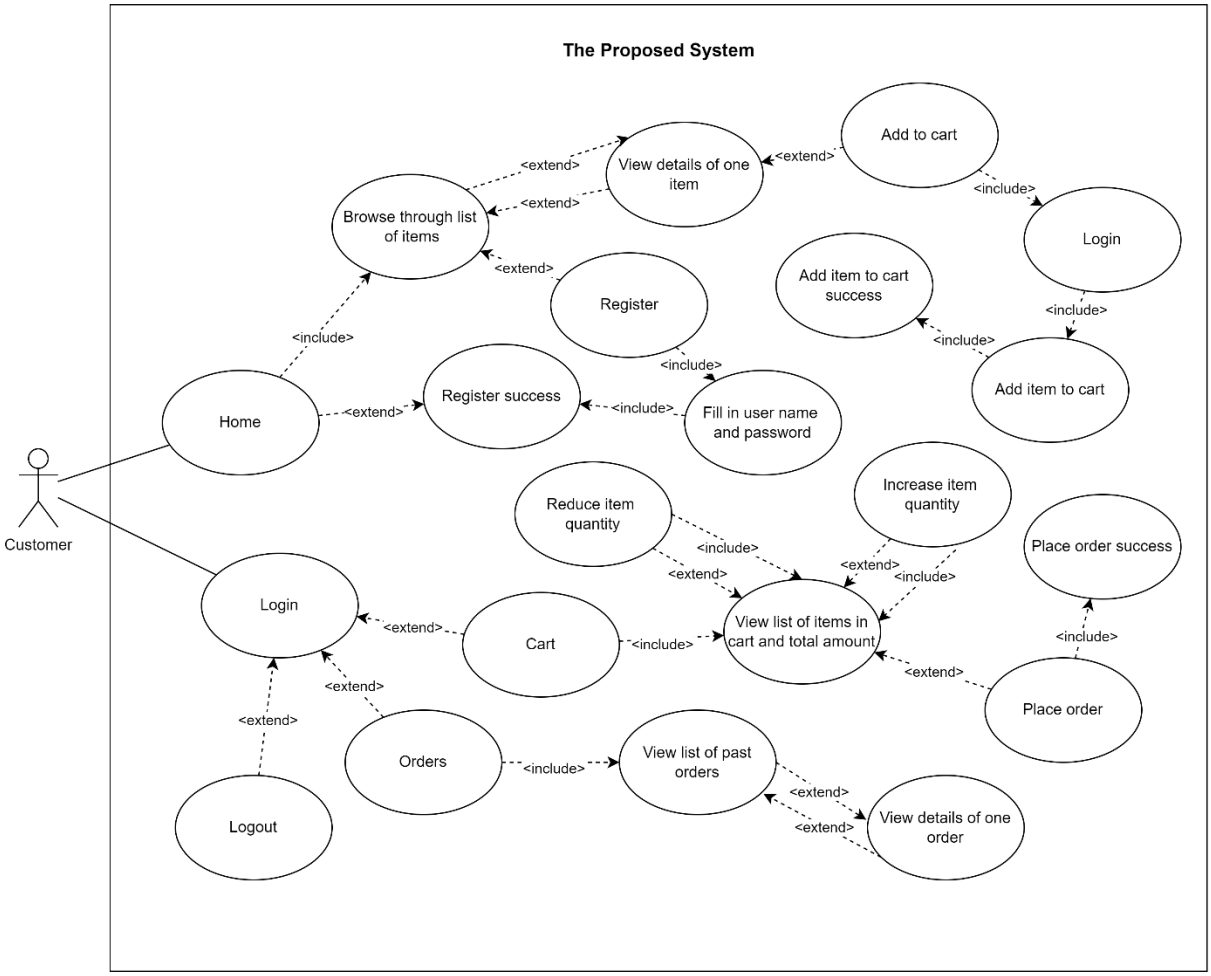


Figure 3.5 Customer Use Case Diagram

3.4. The Customer Churn Prediction Model

In this subchapter, the training of the prediction model and the way of getting the attributes for prediction are discussed.

3.4.1. Training of the Prediction Model

The prediction model is trained by using the dataset retrieved from the work of Fridrich & Dostal (2022). The training dataset has 49358 rows of non-null values. The input attributes of the prediction model are formed around user-application interaction, which is generally available for any electronic commerce retail store (Fridrich & Dostal, 2022). In the proposed system, the attributes are shortlisted to only include those attributes that can be obtained through the functionalities available in the proposed system. The detailed explanation of those input attributes and the way which the data can be obtained are discussed in subtopic 3.4.2. In the proposed system, a customer with high churn probability is a customer who has a high probability of not interacting with the website in the next month.

The prediction model is trained based on neural network. An experiment conducted by Ismail, et. al. (2015) found that multi-layer perceptron (neural network) has better performance compared to logistic regression and multiple regression. The prediction model is trained by using the TensorFlow.js library. The detailed explanation of the library is in subtopic 3.6.11. The prediction model has 1 dense input layer with 4 nodes using ReLU activation function, 1 dense hidden layer with 4 nodes using ReLU activation function and 1 dense output layer with 1 node using sigmoid activation function. An experiment conducted by Pondel, et. al. (2021) found that the test accuracy of 1 dense hidden layer without dropout with ReLU activation function is higher (0.736) than that of 2 dense hidden layers with 30% dropout with Leaky ReLU activation function (0.733). The prediction model is trained by using root mean square propagation (RMSProp) optimizer, binary cross entropy loss function, batch size 100, number of epochs 200 and validation split 30%. An experiment conducted by Domingos, et. al. (2021) has found that RMSProp optimizer achieved best accuracy for multi-layer perceptron (dense neural network).

The training dataset is uploaded from the comma-separated value (CSV) file to the MongoDB database. During training, the dataset is loaded from the database. The prediction model was pre-trained on the server and was saved on the server. The prediction model is then loaded when needed.

3.4.2. Way of Getting the Attributes for Prediction

Table 3.1 explains the attributes used for churn prediction. The training dataset is retrieved from the work of Fridrich & Dostal (2022). Only input attributes that can be obtained from the functionalities of the proposed system are shortlisted. The detailed explanation of the

attribute and the way of getting the attribute in the proposed system are explained in Table 3.1 and Pseudocode 3.1.

Table 3.1 Explanation of the attributes used for churn prediction

Attribute name	**Variable name**	**Range in training dataset**	**Explanation**	**Way of getting the variable in the proposed system**
Session recency	ses_rec	0.0 to 31.0	Difference in days between the last user session and the second last user session in the past month	$Date\ of\ last\ logout - Date\ of\ second\ last\ logout$
Average session recency	ses_rec_avg	0.0 to 99.0	Average days between sessions	Refer to Pseudocode 3.1
Standard deviation in session recency	ses_rec_sd	0.0 to 47.5	Standard deviation of days between sessions	Refer to Pseudocode 3.1
Coefficient of variation in session recency	ses_rec_cv	-1.0 to 11.5	Ratio of standard deviation of days between sessions to average days between sessions	$\frac{Standard\ deviation\ in\ session\ recency}{Average\ session\ recency}$
User maturity	user_rec	0.0 to 99.0	Difference in days between the last user session and the first user session	$Date\ of\ last\ logout - Date\ of\ first\ logout$
Session frequency	ses_n	2 to 475	Session count	$Number\ of\ logouts$
User-app interaction frequency	int_n	2 to 5549	User-application interaction (view / add to cart / buy) count	$Number\ of view\ and\ add\ to\ cart + Number\ of\ transaction$

Relative user-app interaction frequency	int_n_r	1.0 to 59.0	Ratio of user-app interaction frequency to session frequency	$\frac{User\ app\ interaction\ frequency}{Session\ frequency}$
Transactional frequency	tran_n	0 to 461	Transaction count	$Number\ of\ transaction$
Relative transactional frequency	tran_n_r	0.0 to 5.0	Ratio of transactional frequency to session frequency	$\frac{Transactional\ frequency}{Session\ frequency}$
Transactional revenue	rev_sum	0.0 to 5971796.0	Total revenue	$Sum\ of\ transaction\ amount$
Relative transactional revenue	rev_sum_r	0.0 to 158748.0	Ratio of total revenue to session frequency	$\frac{Total\ revenue}{Session\ frequency}$

In Table 3.1, all of the attribute originates from the training dataset built in the work of Fredrich & Dostal (2022). The contribution in the proposed work is trying to integrate and apply the prediction model built from the dataset in an electronic commerce website to predict customer churn. To get the attribute needed, in the proposed system, every time when the customer logs out, the time stamp will be saved in the database. The logout time stamp represents the end of a session and is analogous with the session time stamp in the training dataset. Besides that, the time stamp will also be saved in the database every time when the customer views an item or when the customer adds an item to cart. These records represent user-application interaction in the training dataset. The time stamp and the amount of the customer placing order are analogous to transaction time stamp and amount in the training dataset.

Pseudocode 3.1 depicts in the proposed system how each input attribute of the customer churn prediction model for a single customer will be computed from the database records as mentioned above. In Pseudocode 3.1, “math.mean()” and “math.std()” the JavaScript’s “math.js” module is used to calculate the mean and standard deviation respectively.

Pseudocode 3.1

```
logoutTimeStamps = GET_FROM_DATABASE
logoutTimeStamps.sort( function (a, b) {return a-b} )
if (logoutimeStamps.length < 2) {
        ses_rec = 31
}
```

```
else {
        ses_rec = logoutTimeStamps[ logoutTimeStamps.length – 1 ] –
        logoutTimeStamps[ logoutTimeStamps.length – 2 ]
        if (ses_rec > 31) {
                ses_rec = 31
        }
}
listOfDayBetweenSessions = [ ]
if (logoutTimeStamps.length < 2) {
        listOfDayBetweenSessions.push(99)
}
else {
        for (j = 0; j < logoutTimeStamps.length - 1; j++) {
                dayBetweenSessions = logoutTimeStamps[ j + 1 ] – logoutTimeStamps[ j ]
                if (dayBetweenSessions > 99) {
                        listOfDayBetweenSessions.push(99)
                }
                else {
                        listOfDayBetweenSessions.push( dayBetweenSessions )
                }
        }
}
ses_rec_avg = math.mean( listOfDayBetweenSessions )
ses_rec_sd = math.std( listOfDayBetweenSessions )
ses_rec_cv = ses_rec_sd / ses_rec_avg
if (logoutTimeStamps.length < 2) {
        user_rec = 99
}
else {
        user_rec = logoutTimeStamps[ logoutTimeStamps.length – 1 ] –
        logoutTimeStamps[ 0 ]
        if (user_rec > 99) {
                user_rec = 99
        }
}
ses_n = logoutTimeStamps.length === 0 ? 1 : logoutTimeStamps.length
viewAndAddToCartTimeStamps = GET_FROM_DATABASE
transactions = GET_FROM_DATABASE
int_n = viewAndAddToCartTimeStamps.length + transactions.length
int_n_r = int_n / ses_n
tran_n = transactions.length
tran_n_r = tran_n / ses_n
rev_sum = 0
for (i = 0; i < transactions.length; i++) {
        rev_sum = rev_sum + transactions[ i ].amount
}
rev_sum_r = rev_sum / ses_n
```

3.5. Entity Relationship Diagram

Figure 3.6 is the entity relationship diagram of the proposed system. There are five main collections, namely items, orders, administrators, customers and training datasets. Each item is created by only one administrator and each administrator can create zero or many items. Each customer can make zero or many orders and each order is made by only one customer. In the items array in the orders collection, each item belongs to only one item in the items collection and each item in the items collection is can be either in or not in the items array in the orders collection. In the cart items array in the customers collection, each item belongs to only one item in the items collection and each item in the items collection is can be either in or not in the cart items array in the customers collection.

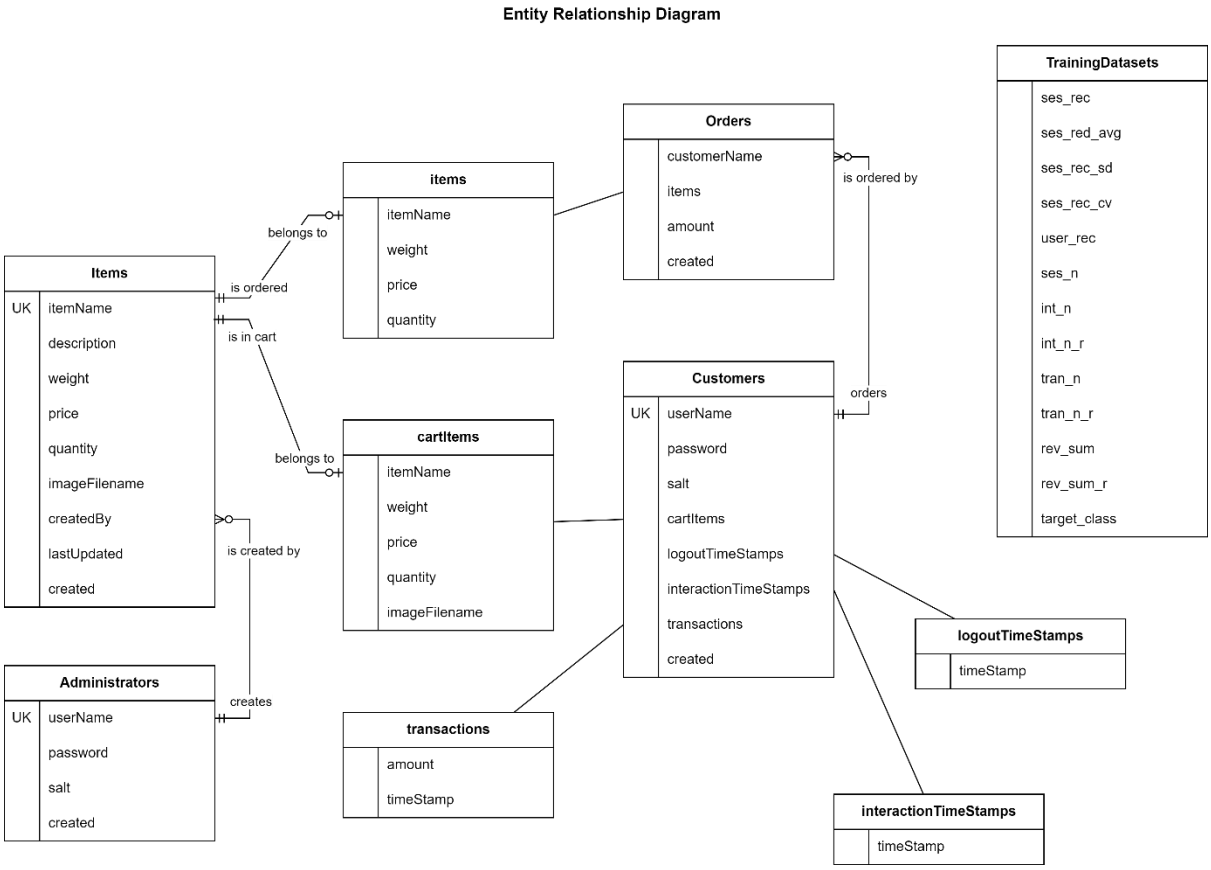


Figure 3.6 Entity Relationship Diagram

The customers collection contains the information of each customer. The fields in the customers collection are explained in Table 3.2.

Table 3.2 Customers Collection

Field	**Explanation**
userName	String, maximum length 20, unique key
password	String
salt	String
cartItems	Array **Field** / **Explanation** itemName / String price / Number weight / Number quantity / Number imageFilename / String
logoutTimeStamps	Array **Field** / **Explanation** timeStamp / Date
interactionTimeStamps	Array **Field** / **Explanation** timeStamp / Date
transactions	Array **Field** / **Explanation** amount / Number timeStamp / Date
created	Date

The items collection contains each item sold on the proposed system. The fields in the items collection are explained in Table 3.3.

Table 3.3 Items Collection

Field	**Explanation**
itemName	String, maximum length 80, unique key
description	String
weight	Number
price	Number
quantity	Number
created	Date

The administrators collection contains the information of each administrator. The fields in the administrators collection are explained in Table 3.4

Table 3.4 Administrators Collection

Field	**Explanation**
userName	String, maximum length 20, unique key
password	String
salt	String

The orders collection contains each order in the proposed system. The fields in the orders collection are explained in Table 3.5.

Table 3.5 Orders Collection

Field	Explanation
customerName	String
items	Array Field: Explanation itemName: String price: Number weight: Number quantity: Number
amount	Number
created	Date

The training datasets collection contains the training dataset used to train the customer churn prediction model in the proposed system. The training datasets collection has no connection to other collections. The fields in the training datasets collection are explained in Table 3.6.

Table 3.6 Training Datasets Collection

Field	Explanation
ses_rec	Number
ses_rec_avg	Number
ses_rec_sd	Number
ses_rec_cv	Number
user_rec	Number
ses_n	Number
int_n	Number
int_n_r	Number
tran_n	Number
tran_n_r	Number
rev_sum	Number
rev_sum_r	Number
target_class	Number

3.6. Components Involved in the Proposed System

In this subchapter, the components involved in the proposed system are discussed.

3.6.1. Express.js, cross origin resource sharing and “nodemon”

Express.js library is used to create the server for the proposed system. To be able to run and test the server on local host, cross origin resource sharing has to be enabled. The “cors” module is a library used to configure the server so that it allows cross origin resource sharing This can be done by calling “app.use(cors)” (dougwilson & troygoode, 2018). On the

other hand, the “nodemon” module is a library used to automatically restart the application when there are changes in file (remy, 2024). This can be done by changing “node app.js” to “nodemon app.js” (remy, 2024).

3.6.2. MongoDB using “mongoose”

The “mongoose” module is an object modelling tool in Node.js designed for MongoDB database (MDN Web Docs, n.d.). MongoDB is a No-SQL database that uses document-oriented model (MDN Web Docs, n.d.). In MongoDB, a collection of documents is analogous to a table of rows in a relational database (MDN Web Docs, n.d.). A database schema can be defined by using the “Schema” object in “mongoose” module (Mongoose, n.d.). The data type of a field can be string, number, date, array, etc. and is defined using the “type” attribute in each field (Mongoose, n.d.). After the schema being defined, the database model can be created by specifying the collection name and the schema name (Mongoose, n.d.). The database model can be used to work with find, delete, save and update operations.

3.6.3. Environment Variables using “dotenv”

Environment variables are variables that might change when the program is loaded and run on a different computer. One example of environment variable is local host port number. “dotenv” is a JavaScript module that loads environment variables dynamically from a separate file named “.env”, which contains the definition of each environment variable (jcblw, et al., 2024). In Express.js server, in the coding, the environment variable can be used by writing “require(‘dotenv’).config()” and specifying “process.env” together with the variable name (jcblw, et al., 2024).

3.6.4. Top Navigation Bar

In the top navigation bar, the “list-style-type” attribute of the<ul> component is set to “none” to remove the dot of the list of links and the “display” attribute of the <li> component is set to “inline-block” to place the list of links in the same line. The “text-decoration” of the <a> component is set to “none” to remove the line under the link and the “color” is set to intended text color as well.

3.6.5. Login Authentication using Basic Authentication

The proposed system uses basic authentication for both customer and administrator. When the user wants to access protected resources, the user is required to login. Basic authentication can be done by the server responding the request with status “401” and specifying the “WWW-Authenticate” header as “Basic” (IBM, 2024). The web client will handle the response by requesting user name and password from the user (IBM, 2024). After the user name and password are obtained, the web client resends the request and specifies the “Authorization” header in the format “Basic [userid]:[password]” (IBM, 2024). The web client retains the user name and password and reuses it for the site (IBM, 2024).

3.6.6. Hashing Password using PBKDF2

In the proposed system, the password is hashed using password-based key derivation function (PBKDF2) before being saved to the database. When user login, the input password is hashed and compared with the hash stored in the database. PBKDF2 hashing algorithm is a slow hash function which makes password cracking attacks harder and is sufficient to protect the password (Defuse Security, 2021). PBKDF2 hashing algorithm can be applied by calling pbkdf2() method in “crypto” module. The pbkdf2() method takes password, salt, number of iterations, key length, digest algorithm and a callback function as input parameters (Geeks For Geeks, 2021). In the proposed system, digest algorithm secure hashing algorithm-256 (SHA-256) and key length of 512 are used. The callback function takes error and derived key as the input parameters. The derived key can then be converted to string by specifying [derived_key].toString(‘base64’) (Geeks For Geeks, 2021).

The salt should be different for every user to impede reverse look up table attack on the salted password (Defuse Security, 2021). The salt can be randomly generated using randomBytes() method in “crypto” module. The crypto.randomBytes() method takes size and callback function (optional) as the input parameter and returns a random generated buffer (Geeks For Geeks, 2024). The random generated bytes can be converted to string specifying [buffer_name].toString(‘base64’).

3.6.7. HTML Form and the “axios” module

In the proposed system, HTML form is used for the customer registration, add item and edit item functionality. Each field of the item are included as the input field of the form. In order

to submit the form data through a post request, the “action” attribute of the form can be set as the destination route and the “method” attribute can be set as “post” (W3 Schools, n.d.). For sending image file, the “enctype” attribute can be set as “multipart-formdata”.

In order to fetch the data through get method and deleting a selected item through delete method, the “axios” module is used. “axios” is a HTTP client for node.js and browser (Axios, n.d.). It can be installed on the browser by using unpkg. A get request can be performed by calling “axios.get()” and passing the route as the input parameter (Axios, n.d.). The get request response will be returned by the axios.get() method upon a successful get request (Axios, n.d.).

3.6.8. HTML Table

In HTML table, each row of the table is specified in the <tr> tag (W3 Schools, n.d.). For the header, each column is specified in the <th> tag, whereas for the data, each column is specified in the <td> tag (W3 Schools, n.d.). To make a table cell to span across multiple columns, the “colspan” attribute of the <td> tag can be specified according to the number of columns that the cell should be spanned across (W3 Schools, n.d.).

3.6.9. Uploading, Previewing and Displaying Image

By specifying the “type” attribute in <input> tag in a HTML form as “file”, the user can upload a file from their device storage (Mdn Web Docs, n.d.). This allows the user to upload image file to the server. The “accept” attribute is a comma separated string which specifies the type of file the input should accept (Mdn Web Docs, n.d.). The “accept” attribute can be tuned to only accept image file by specifying “image/*”.

The image preview can be built by using <img> tag. The image preview element can be set to display the image selected by using the readAsDataUrl() method in the FileReader object (Mdn Web Docs, n.d.). The input file, which can be accessed by specifying “<input element>.files[0]”, is passed as the input parameter for the readAsDataUrl() method (Mdn Web Docs, n.d.). An event listener, which specifies a callback function which sets the result of the readAsDataUrl() method as the “src” attribute of the image preview element, can be added to the FileReader object (Mdn Web Docs, n.d.). The readDataAsUrl() method will return a URL in the form of “data:<file type>; base64,<base64 encoded string>” (Mdn Web Docs, n.d.).

Upon form submission, the form data can be submitted in a post request by specifying the type as “multipart/formdata” (Jorge, 2023). The file can be processed by using the “multer” module. In the “storage” attribute of multer can be configured to store the file in a specified directory and with a specified file name (Jorge, 2023). This can be done by specifying the “destination” and “filename” attributes in multer.diskStorage() method, then specify the configured disk storage as the “storage” attribute in multer. The file name can be saved in the database and will be used to access the image file in the future. By specifying “<middleware name>.single(<”name” attribute of file input element>)” in the route’s callbacks, the file can be accessed by specifying “req.file” (Jorge, 2023).

3.6.10. Dumping of Dataset from CSV file into MongoDB Database

In the proposed system, the training dataset of the customer churn prediction model originates from the work of Fridrich &Dostal (2022). The dataset is in comma-separated value (CSV) file format. The reading of the data in the dataset can be done by using “fs” module and parse() function in “csv-parse” module. The path of the .csv file is passed as the input parameter of the fs.createReadStream() function and this will create a read stream of the .csv file and send it to the pipeline (Gandhi, 2024). The data in the pipeline can be parsed by specifying “.pipe(parse()) behind the “fs.createReadStream()” (Gandhi, 2024). The parse option can be specified by setting the “delimiter” as “,” and “from_line” as 2, so that the data in each column will be parsed starting from line 2. Event handler on data loaded can be specified by adding “.on(‘data’, function(row))”, where the “row” is an array of the data in the current row read by the program (Gandhi, 2024).

The data in each row represent the record of one customer and it will be converted into JavaScript Object Notation (JSON) form by specifying the column name and the column value. The JSON form of the row data is then saved to the database by using “mongoose” module.

3.6.11. Training, Saving and Loading of the Model using TensorFlow.js

In the proposed system, the training of the customer churn prediction model is done on the Express.js server by using “@tensorflow/tfjs-node” module, which is imported as “tf”. Once the training dataset is loaded from the database, the JavaScript Object Notation (JSON) form of each row is converted to array form. When the array is passed into tf.tensor() method, a tf.Tensor object is formed from the array (Tensor Flow, n.d.). A tf.Tensor object is the central unit of data in TensorFlow.js (Tensor Flow, n.d.).

In TensorFlow.js, a sequential model can be created by using the tf.sequential() method by specifying the layers inside the “layer” parameter (Tensor Flow, n.d.). A dense layer can be created by using the tf.layers.dense() method and specifying the input shapes, number of neuron units and activation function (Tensor Flow, n.d.). A model named “model” can be compiled by calling “model.compile()” by specifying the optimizer, the loss function and the list of metrics (Tensor Flow, n.d.). The model can be trained by calling “model.fit()” by specifying the number of epochs and batch size (Tensor Flow, n.d.).

The model named “model” can be saved to the native file system by calling “model.save()” and specifying the input of the “save()” method as the concatenation of “file://” and the path to the destination directory (Tensor Flow, n.d.). The command will save two files inside the directory, which are a text JSON file named “model.json” and a binary file named “weights.bin” (Tensor Flow, n.d.).

Once saved, the model can be loaded by calling “tf.loadLayersModel()” method and specifying the input of the method as the concatenation of “file://” and the path to the “model.json” file (Tensor Flow, n.d.). The pre-trained prediction model will be returned by the method.

3.6.12. HTML Get Query Parameter from URL

In HTML, the query parameter of the URL can be got by using URLSearchParams. If the URL is “http://localhost:3000/?name=sai”, then the query parameter of “name” is “sai”. The query parameters can be got by writing “params = new URLSearchParams(window.location.search)”, then specifying “params.get(‘name’)” to get the “name” field (HTML Hints, 2020).

3.6.13. Grid Display of List Items

In the proposed system, the items in the customer home page are displayed in grids. In HTML, to display the child elements in grids, the “display” attribute of the parent element should be configured as “grid” (W3 Schools, n.d.). The “grid-template-columns” attribute of the parent element can be configured to specify the number of columns and width of each column in the grid display (W3 Schools, n.d.). For example, if the “grid-template-columns” attribute is specified as “auto auto auto auto”, then the grid will have 4 equal size columns (W3 Schools,

n.d.). In the proposed system, grid display is used in customer home page for rendering the list of items.

3.6.14. Promise and Promise.all

Promise.all() method in JavaScript takes an array of promises as the input and returns a single promise, which fulfils if all promises are fulfilled and rejects if any of the promises is rejected (MDN Web Docs, n.d.). In the proposed system, promise.all() is used to handle multiple asynchronous functions for checking and updating quantity of each item in the customer's order.

4. IMPLEMENTATION

4.1. Customer Churn Prediction Feature

In this subchapter, the implementation details related to the customer churn prediction feature in the proposed system are discussed.

4.1.1. Analyzing the Training Dataset

The dataset is analyzed using Jupyter notebook. Figure 4.1 shows the dataset information running in Visual Studio Code. As shown in Figure 4.1, the training dataset has 49358 rows with non-null values.

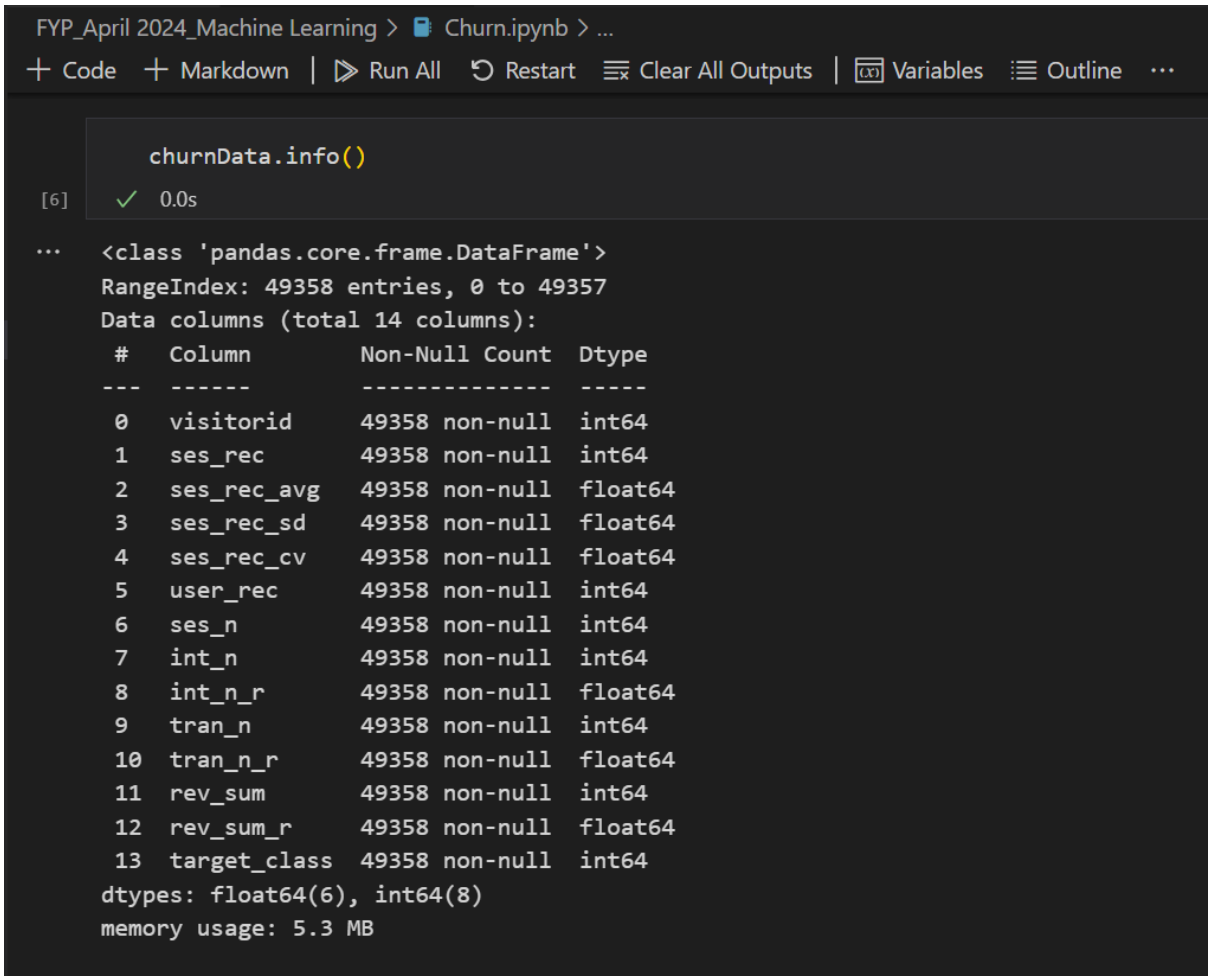


Figure 4.1 Dataset information

The minimum and maximum values of each column of the training dataset are computed and listed in Table 3.1 in subtopic 3.4.2. Figure 4.2 shows the computation of the minimum and maximum values.

FYP_April 2024_Machine Learning > Churn.ipynb > ...

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...

```python
minMaxData = {
    "column": [],
    "minimum": [],
    "maximum": []
}

for column in churnData.columns:
    minimum = churnData[column].min()
    maximum = churnData[column].max()

    minMaxData["column"].append(column)
    minMaxData["minimum"].append(minimum)
    minMaxData["maximum"].append(maximum)

minMaxDataframe = pd.DataFrame(minMaxData)
print(minMaxDataframe)
```

[7] ✓ 0.0s

```
...          column  minimum       maximum
0       visitorid     37.0  1.407573e+06
1         ses_rec      0.0  3.100000e+01
2     ses_rec_avg      0.0  9.900000e+01
3      ses_rec_sd      0.0  4.750000e+01
4      ses_rec_cv     -1.0  1.152512e+01
5        user_rec      0.0  9.900000e+01
6           ses_n      2.0  4.750000e+02
7           int_n      2.0  5.549000e+03
8         int_n_r      1.0  5.900000e+01
9          tran_n      0.0  4.610000e+02
10       tran_n_r      0.0  5.000000e+00
11        rev_sum      0.0  5.971796e+07
12      rev_sum_r      0.0  1.587480e+06
13   target_class      0.0  1.000000e+00
```

Figure 4.2 Dataset minimum and maximum values

4.1.2. Dumping of Training Datasets into the Database

The training dataset is loaded from a comma-separated value (CSV) file, line by line, and each line is converted to JavaScript Object Notation (JSON) format and saved to the

MongoDB database as one record. Figure 4.3 is the code for dumping the training dataset from .csv file to MongoDB database.

```
const fs = require('fs');
const {parse} = require('csv-parse');

const datasetPath = './ChurnTrainingDataset/ecom-user-churn-data.csv';
const parseOption = parse({delimiter:',', from_line:2});

fs.createReadStream(datasetPath)
.pipe(parseOption)
.on("data", async (row)=>{

    const trainingDataset = new TrainingDatasetModel(
        {
            ses_rec : row[1],
            ses_rec_avg : row[2],
            ses_rec_sd : row[3],
            ses_rec_cv : row[4],
            user_rec : row[5],
            ses_n : row[6],
            int_n : row[7],
            int_n_r : row[8],
            tran_n : row[9],
            tran_n_r : row[10],
            rev_sum : row[11],
            rev_sum_r : row[12],
            target_class : row[13]
        }
    );

    await trainingDataset.save().then(response=>{
        //
    }).catch(error=>{
        console.log(error);
    });
})
.on("error", (error)=>{
    console.log(error);
})
.on("end", ()=>{
    console.log('Finished read file.');
});
```

Figure 4.3 Dumping of training dataset to database

Figure 4.4 shows one record (one row) of the training dataset stored in MongoDB database, viewed from MongoDB Atlas.

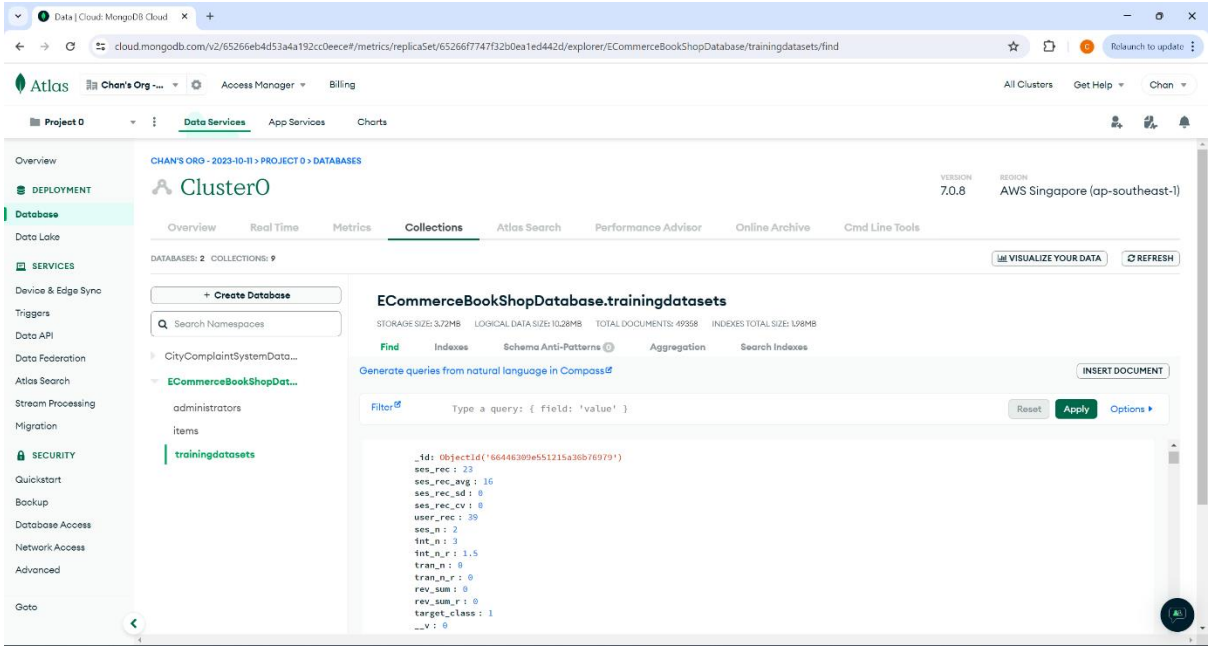


Figure 4.4 Storing the training dataset in MongoDB database

4.1.3. Training of the Prediction Model

When training the prediction model, the training dataset is loaded from the MongoDB database and converted from JavaScript Object Notation (JSON) into an array. All predictors in each row formed the inner array named “x” and all rows in the dataset formed the outer array named “xs”. The target class of each row formed the inner array named “y” and all rows in the dataset formed the outer array named “ys”. The “xs” and “ys” array is converted into a Tensor object. Figure 4.5 is the code for implementing the aforementioned process.

```
// Load dataset from MongoDB

TrainingDatasetModel.find().then(results=>{
    console.log(results.length);

    const xs = [];
    const ys = [];

    for (var i = 0; i < results.length; i++) {
        const x = [
            results[i].ses_rec,
            results[i].ses_rec_avg,
            results[i].ses_rec_sd,
            results[i].ses_rec_cv,
            results[i].user_rec,
            results[i].ses_n,
            results[i].int_n,
            results[i].int_n_r,
            results[i].tran_n,
            results[i].tran_n_r,
            results[i].rev_sum,
            results[i].rev_sum_r
        ]

        const y = [
            results[i].target_class
        ]

        xs.push(x);
        ys.push(y);
    }

    //console.log(xs[0], ys[0]);

    const x_tensor = tf.tensor(xs);
    const y_tensor = tf.tensor(ys);
```

Figure 4.5 Loading of training dataset from database

The customer churn prediction model is built with one dense input layer with ReLU activation function with 4 neurons, one hidden layer with ReLU activation function with 4 neurons and one output layer with sigmoid activation function with 1 neuron. The model is trained with RMSProp optimizer, binary cross entropy loss function, batch size 100, number of epochs 200 and validation size 30%. After the model is trained, the model is saved to the server for future use. Figure 4.6 is the code for training the prediction model. As shown in Figure 4.6, the training accuracy of the model is 0.8610 and the validation accuracy is 0.8613.

```
108     const model = tf.sequential({
109         layers: [
110             tf.layers.dense({inputShape: [12], units: 4, activation: 'relu'}),
111             tf.layers.dense({units: 4, activation: 'relu'}),
112             tf.layers.dense({units: 1, activation: 'sigmoid'})
113         ]
114     });
115
116     model.compile({
117         optimizer: 'rmsprop',
118         loss: 'binaryCrossentropy',
119         metrics: ['accuracy']
120     });
121
122     model.fit(x_tensor, y_tensor, {
123         batchSize: 100,
124         epochs: 200,
125         validationSplit: 0.3
126     }).then(info=>{
127         console.log('Train arccuracy', info.history.acc[199]);
128         console.log('Validation accuracy', info.history.val_acc[199]);
129
130         const prediction = model.predict(tf.randomNormal([5, 12]));
131         prediction.print();
132
133         // Save the model
134
135         model.save('file://' + path.join(__dirname + '/ChurnModel')).then(response=>{
136             console.log('Saved', response);
137
138         }).catch(error=>{
139             console.log(error);
140         });
141
142
143     }).catch(error=>{
144         console.log(error);
145     });
146
147 }).catch(error=>{
148     console.log(error);
149 });
150
```

```
PROBLEMS    TERMINAL    DEBUG CONSOLE    OUTPUT    PORTS

1959ms 57us/step - acc=0.861 loss=0.828 val_acc=0.861 val_loss=0.849
Train arccuracy 0.8610129952430725
Validation accuracy 0.8612911701202393
Tensor
```

Figure 4.6 Training of the prediction model

4.1.4. Save Predictors

The predictors of the prediction model are formed around user sessions, user-application interactions and transactions. In the proposed system, these fields are stored in the customers collection. Figure 4.7 is the code for saving customer's logout time stamp, which is analogous with user session. The logout time stamp is saved every time when the customer logs out.

```
const CustomerModel = require('../Database/CustomerModel');

function saveLogoutTimeStamp(req, res, next) {
    const authorization = req.header('Authorization');

    if (authorization == null) {
        console.log('No authorization');
        next();
        return;
    }

    const [_, credentials] = authorization.split(' ');
    const decodedCredentials = atob(credentials);
    const [userName, password] = decodedCredentials.split(':');

    console.log('user name:', userName);

    CustomerModel.findOne({userName: userName}).then(customer=>{
        console.log('Found customer', customer);

        if (customer == null) {
            console.log('No such user name');
            next();
            return;
        }

        const logoutTimeStamps = customer.logoutTimeStamps == null ? [] : customer.logoutTimeStamps;
        logoutTimeStamps.push({
            timeStamp: new Date(Date.now())
        });

        CustomerModel.findByIdAndUpdate(customer._id, {logoutTimeStamps: logoutTimeStamps}).then(response=>{
            console.log('Updated logout time stamp', response);

            next();

        }).catch(error=>{
            console.log('Update logout time stamp error');
            next();
        })

    }).catch(error=>{
        console.log('Find customer error', error);
        next();
    })
}

module.exports = saveLogoutTimeStamp;
```

Figure 4.7 Save logout time stamp

Figure 4.8 is the code for saving interaction time stamp, which is analogous to user-application interaction. The interaction time stamp is saved every time when the customer views the details of an item or adds an item to cart.

```
const CustomerModel = require('../Database/CustomerModel');

function saveInteractionTimeStamp(req, res, next) {
    const authorization = req.header('Authorization');

    if (authorization == null) {
        console.log('No authorization');
        next();
        return;
    }

    const [_, credentials] = authorization.split(' ');
    const decodedCredentials = atob(credentials);
    const [userName, password] = decodedCredentials.split(':');

    console.log('user name:', userName);

    CustomerModel.findOne({userName: userName}).then(customer=>{
        console.log('Found customer', customer);

        if (customer == null) {
            console.log('No such user name');
            next();
            return;
        }

        const interactionTimeStamps = customer.interactionTimeStamps == null ? [] : customer.interactionTimeStamps;
        interactionTimeStamps.push({
            timeStamp: new Date(Date.now())
        });

        CustomerModel.findByIdAndUpdate(customer._id, {interactionTimeStamps: interactionTimeStamps}).then(response=>{
            console.log('Updated interaction time stamp', response);

            next();

        }).catch(error=>{
            console.log('Update interaction time stamp error');
            next();
        })

    }).catch(error=>{
        console.log('Find customer error', error);
        next();
    })
}

module.exports = saveInteractionTimeStamp;
```

Figure 4.8 Save interaction time stamp

Figure 4.9 is the code for saving the customer's transaction time stamp and amount. The transaction time stamp and amount are saved every time when the customer places order.

```
const CustomerModel = require('../Database/CustomerModel');

function saveTransactions(req, res, next) {
    const authorization = req.header('Authorization');

    if (authorization == null) {
        console.log('No authorization');
        next();
        return;
    }

    const [_, credentials] = authorization.split(' ');
    const decodedCredentials = atob(credentials);
    const [userName, password] = decodedCredentials.split(':');

    console.log('user name:', userName);

    CustomerModel.findOne({userName: userName}).then(customer=>{
        console.log('Found customer', customer);

        if (customer == null) {
            console.log('No such user name');
            next();
            return;
        }

        if (customer.cartItems == null || customer.cartItems.length === 0) {
            console.log('No item in cart');
            next();
            return;
        }

        var amount = 0;
        for (var i = 0; i < customer.cartItems.length; i++) {
            amount += customer.cartItems[i].price * customer.cartItems[i].quantity;
        }

        const transactions = customer.transactions == null ? [] : customer.transactions;
        transactions.push({
            timeStamp: new Date(Date.now()),
            amount: amount
        });

        CustomerModel.findByIdAndUpdate(customer._id, {transactions: transactions}).then(response=>{
            console.log('Updated transaction', response);

            next();

        }).catch(error=>{
            console.log('Update transaction error');
            next();
        })

    }).catch(error=>{
        console.log('Find customer error', error);
        next();
    })
}

module.exports = saveTransactions;
```

Figure 4.9 Save transaction time stamp and amount

4.1.5. Get Predictors

The predictor fields that are used to predict churn for each customer are formed from logout time stamps, interaction time stamps and transactions. The equations and pseudocode for getting the predictors are already explained in subtopic 3.4.2. Figure 4.10 and Figure 4.11

are the code for getting the predictors, which is logically equivalent to Pseudocode 3.1 in subtopic 3.4.2.

```
// get predictor values
function getPredictors(logoutTimeStamps, viewAndAddToCartTimeStamps, transactions) {

    logoutTimeStamps.sort((a, b)=>{return new Date(a.timeStamp).getTime() - new Date(b.timeStamp).getTime();});

    //console.log(logoutTimeStamps);

    var ses_rec;
    if (logoutTimeStamps.length < 2) {
        ses_rec = 31;
    }
    else {
        //console.log(logoutTimeStamps[logoutTimeStamps.length - 2].timeStamp, new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime())

        const differenceInMilliseconds = new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime() - new Date(logoutTimeStamps[logoutTimeStamps.length - 2].timeStamp).getTime();
        const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

        if (differenceInDays > 31) {
            ses_rec = 31;
        }
        else {
            ses_rec = differenceInDays;
        }
    }

    const listOfDayBetweenSessions = [];
    if (logoutTimeStamps.length < 2) {
        listOfDayBetweenSessions.push(99);
    }
    else {
        for (var j = 0; j < logoutTimeStamps.length - 1; j++) {
            const differenceInMilliseconds = new Date(logoutTimeStamps[j + 1].timeStamp).getTime() - new Date(logoutTimeStamps[j].timeStamp).getTime();
            const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

            if (differenceInDays > 99) {
                listOfDayBetweenSessions.push(99);
            }
            else {
                listOfDayBetweenSessions.push(differenceInDays);
            }
        }
    }

    const ses_rec_avg = mathjs.mean(listOfDayBetweenSessions);

    const ses_rec_sd = mathjs.std(listOfDayBetweenSessions);

    const ses_rec_cv = ses_rec_sd / ses_rec_avg;
```

Figure 4.10 Get predictors (part 1 of 2)

```
    var user_rec;
    if (logoutTimeStamps.length < 2) {
        user_rec = 99;
    }
    else {
        const differenceInMilliseconds = new Date(logoutTimeStamps[logoutTimeStamps.length - 1].timeStamp).getTime() - new Date(logoutTimeStamps[0].timeStamp).getTime();
        const differenceInDays = differenceInMilliseconds / (1000 * 3600 * 24);

        if (differenceInDays > 99) {
            user_rec = 99;
        }
        else {
            user_rec = differenceInDays;
        }
    }

    const ses_n = logoutTimeStamps.length === 0 ? 1 : logoutTimeStamps.length;

    const int_n = viewAndAddToCartTimeStamps.length + transactions.length;

    const int_n_r = int_n / ses_n;

    const tran_n = transactions.length;

    const tran_n_r = tran_n / ses_n;

    var rev_sum = 0;
    for (var i = 0; i < transactions.length; i++) {
        rev_sum = rev_sum + transactions[i].amount;
    }

    const rev_sum_r = rev_sum / ses_n;

    return [ses_rec, ses_rec_avg, ses_rec_sd, ses_rec_cv, user_rec, ses_n, int_n, int_n_r, tran_n, tran_n_r, rev_sum, rev_sum_r];
}
```

Figure 4.11 Get predictors (part 2 of 2)

4.1.6. Make predictions

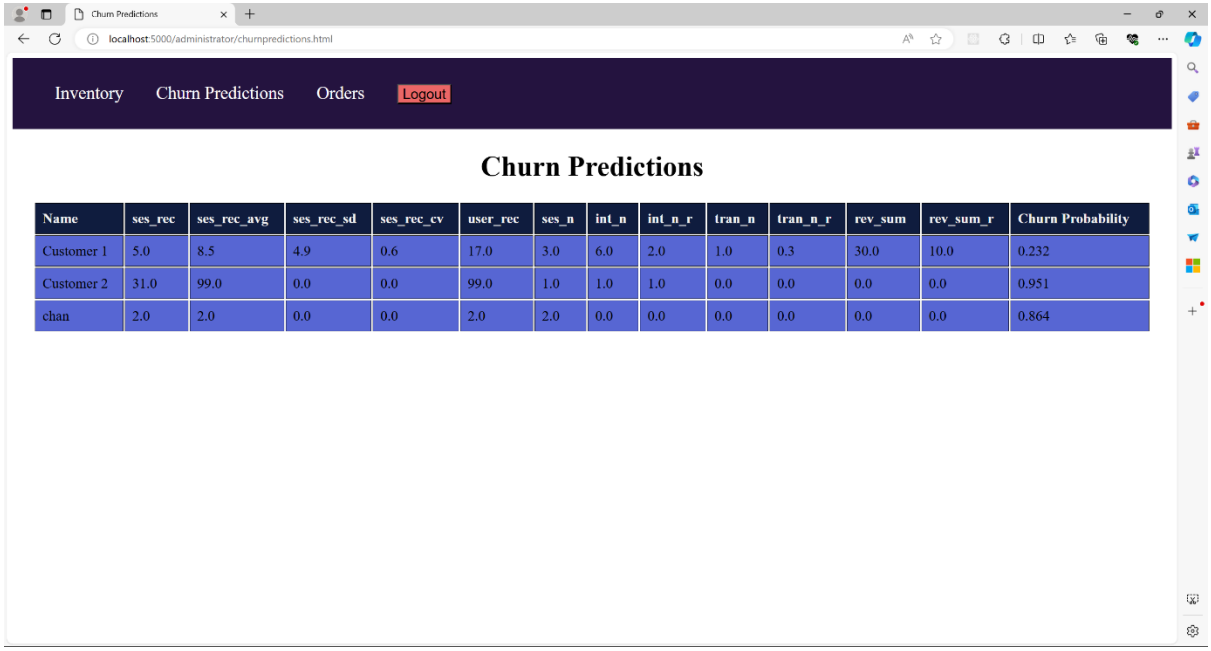
When making predictions, the logout time stamps, interaction time stamps and transactions of all customers are loaded and passed, one by one, into the get predictors function (details in subtopic 4.1.5). The output of the get predictors function is the input parameters for the prediction model to predict the churn for one customer. An “arrays in an array” is formed, where each inner array is for one row or one customer and the outer array contains all rows or all customers. The array is converted to Tensor object. The pre-trained and saved prediction model is loaded and the Tensor array is passed into the prediction model. The prediction results are then downloaded. The downloaded prediction results are in array format. Figure 4.12 is the code for making predictions.

```
// Predict customer churn
app.get('/predictchurn', adminLogin, (req, res)=>{

    CustomerModel.find().then(results=>{
        console.log('Find customer success', results);

        const predictionList = [];
        const xs = [];

        for (var i = 0; i < results.length; i++) {
            const logoutTimeStamps = results[i].logoutTimeStamps == null ? [] : results[i].logoutTimeStamps;
            const viewAndAddToCartTimeStamps = results[i].interactionTimeStamps == null ? [] : results[i].interactionTimeStamps;
            const transactions = results[i].transactions == null ? [] : results[i].transactions;

            const [ses_rec, ses_rec_avg, ses_rec_sd, ses_rec_cv, user_rec, ses_n, int_n, int_n_r, tran_n, tran_n_r, rev_sum, rev_sum_r] = getPredictors(logoutTimeStamps, viewAndAddToCartTimeStamps, transactions);

            predictionList.push({
                name: results[i].userName,
                ses_rec: ses_rec,
                ses_rec_avg: ses_rec_avg,
                ses_rec_sd: ses_rec_sd,
                ses_rec_cv: ses_rec_cv,
                user_rec: user_rec,
                ses_n: ses_n,
                int_n: int_n,
                int_n_r: int_n_r,
                tran_n: tran_n,
                tran_n_r: tran_n_r,
                rev_sum: rev_sum,
                rev_sum_r: rev_sum_r
            });

            xs.push([ses_rec, ses_rec_avg, ses_rec_sd, ses_rec_cv, user_rec, ses_n, int_n, int_n_r, tran_n, tran_n_r, rev_sum, rev_sum_r]);
        }

        const xs_tensor = tf.tensor(xs);

        tf.loadLayersModel('file://' + path.join(__dirname + '/ChurnModel/model.json')).then(model=>{
            console.log('Loaded model');

            const predictions = model.predict(xs_tensor);

            predictions.data().then(values=>{
                console.log('Downloaded predicted value', values);

                for (var j = 0; j < values.length; j++) {
                    predictionList[j].churnProbability = values[j];
                }

                res.status(200).json({
                    predictionList: predictionList
                });

            }).catch(error=>{
                console.log('Download predicted value error', error);
                res.sendStatus(500);
            });

        }).catch(error=>{
            console.log('Load model error', error);
            res.sendStatus(500);
        });

    }).catch(error=>{
        console.log('Find customer error', error);
        res.sendStatus(500);
    });
});
```

Figure 4.12 Make predictions

4.1.7. Churn Predictions Page

The churn predictions page is only accessible by the administrator. The churn predictions page displays the churn probability of each customer in table form, together with the values of each input variable of the prediction model. Figure 4.13 is the churn predictions page which shows the churn predictions for 3 dummy customers. As shown in Figure 4.13, the churn probability of “Customer 1” is 0.232 or 23.2%, which is more likely “not to churn”; whereas the churn probability of “Customer 2” is 0.951 or 95.1%, which is more likely “to churn”.



Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	2.0	2.0	0.0	0.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.864

Figure 4.13 Churn predictions page

4.2. Other features

In this subchapter, the features available in the proposed system, other than the customer churn prediction feature, are discussed.

4.2.1. Inventory Page

The inventory page is only accessible by the administrator. The inventory page has the create, read, update and delete (CRUD) operations for the items selling on the proposed system. On the inventory page, all items selling on the proposed system are listed in a table. Figure 4.14 is the inventory page.

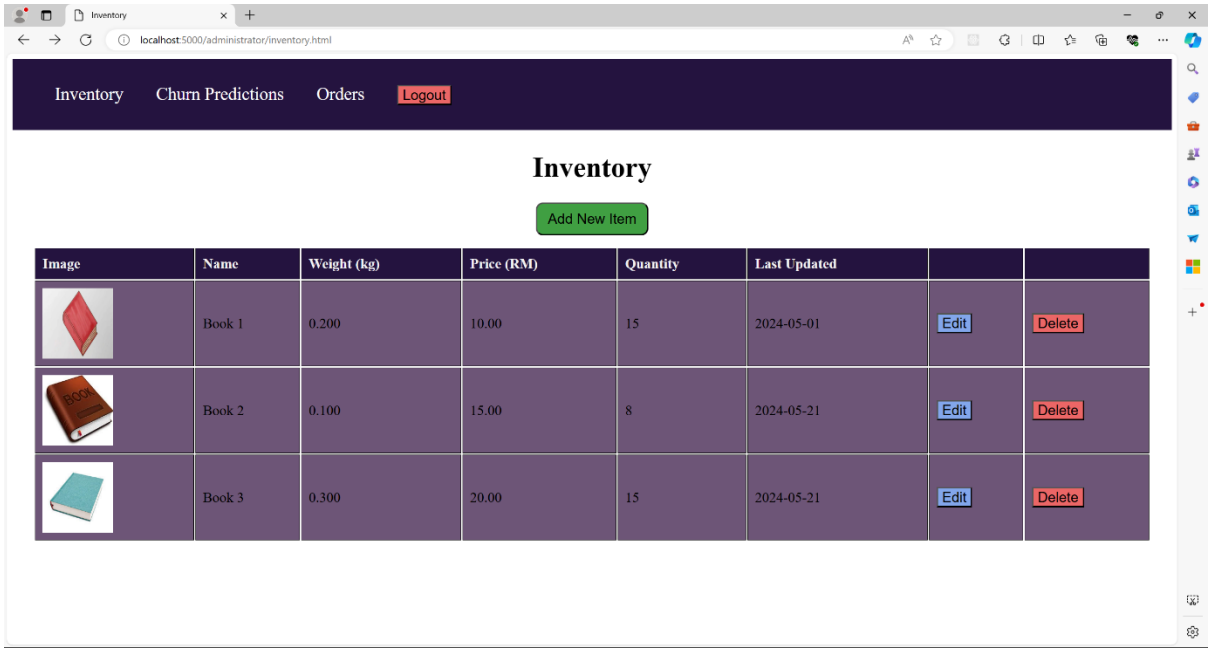


Figure 4.14 Inventory page

By clicking the green “Add New Item”, button, the administrator will be navigated to the add item page. On the add item page, the administrator can fill in the item name, description, weight, price and quantity, as well as attaching an image for the item. Figure 4.15 and Figure 4.16 are the add item page.

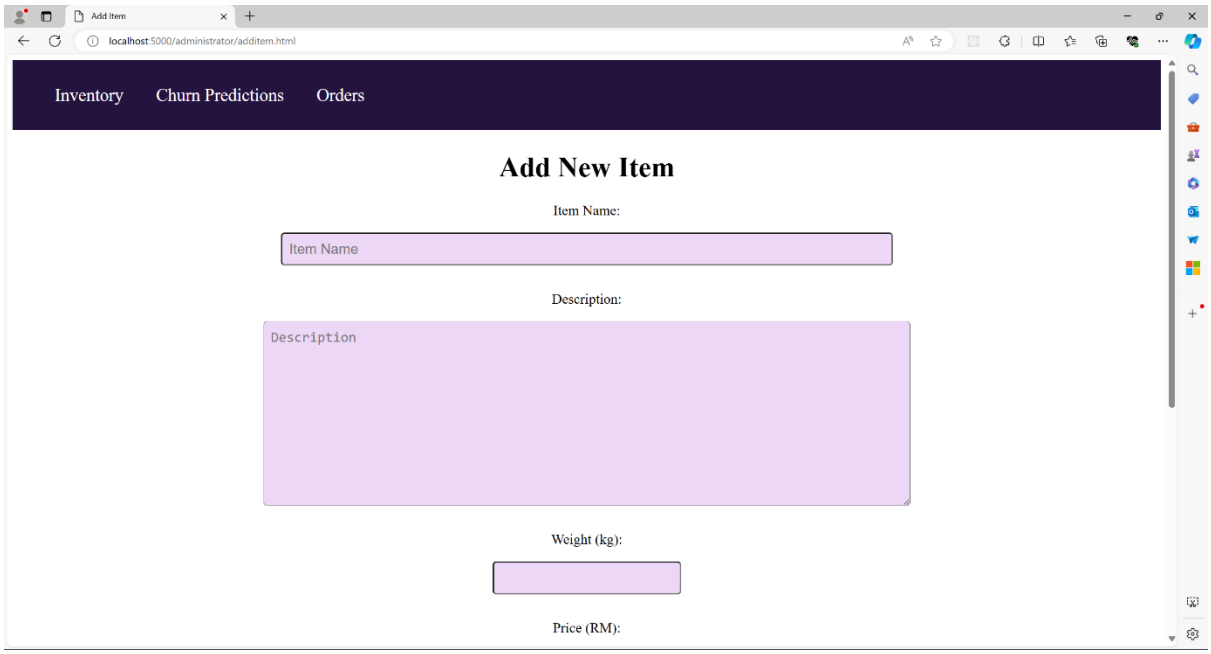


Figure 4.15 Add item page (part 1 of 2)

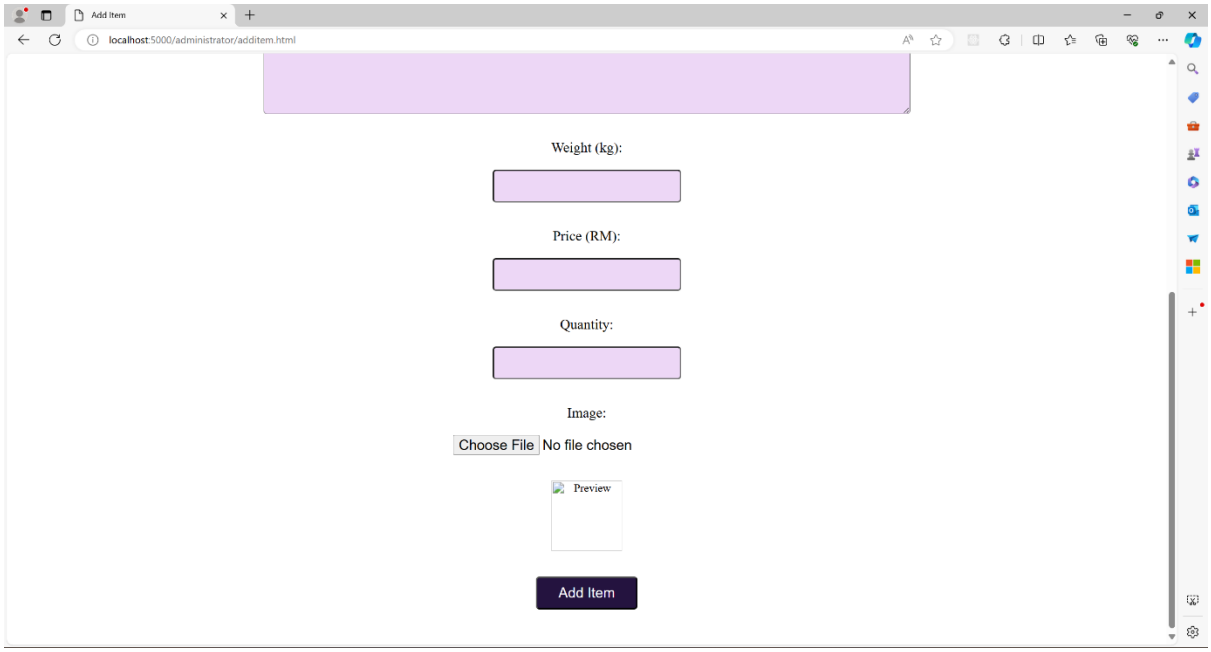


Figure 4.16 Add item page (part 2 of 2)

By clicking on the purple “Add Item” button at the bottom of the add item page, the item will be added and saved to the database. If the item is successfully added, the administrator will be navigated to the add item success page, where the administrator can opt to go back to the inventory page or go to the add item page again to add another item. Figure 4.17 is the add item success page.

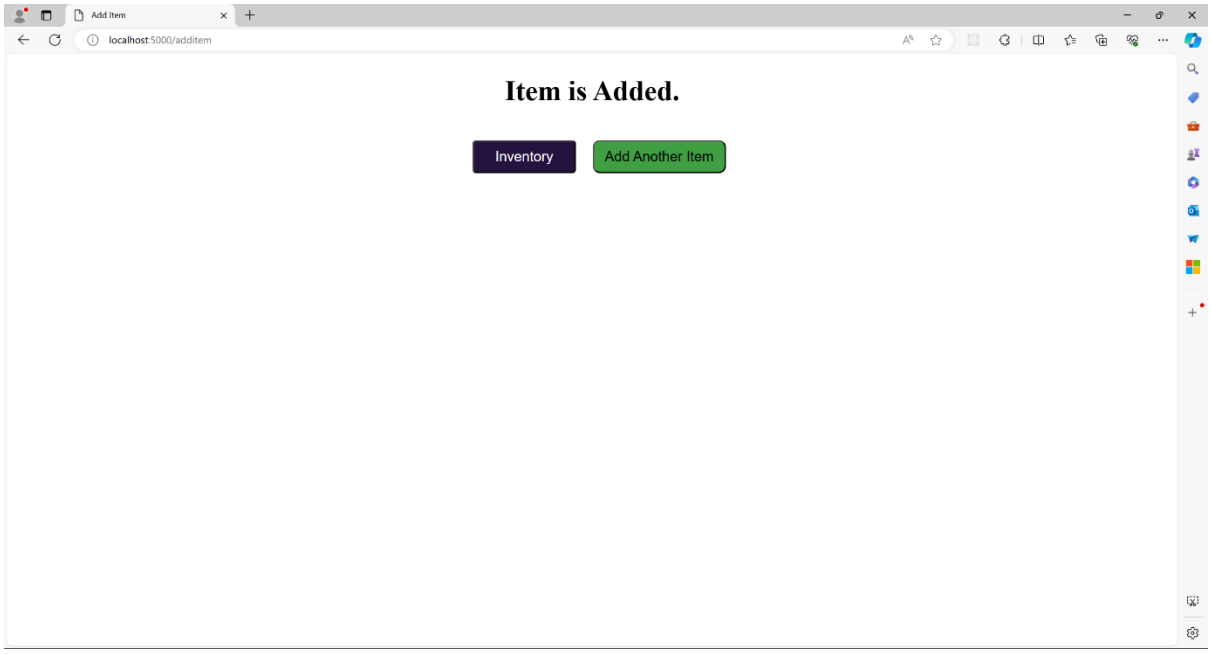


Figure 4.17 Add item success page

On the inventory page (Figure 4.14), by clicking on the blue “Edit” button, the administrator will be navigated to the edit item page, where the item name, description, weight,

price, quantity and image of the selected item are displayed. The administrator can edit any field except the item name. Figure 4.18 and Figure 4.19 are the edit item page.

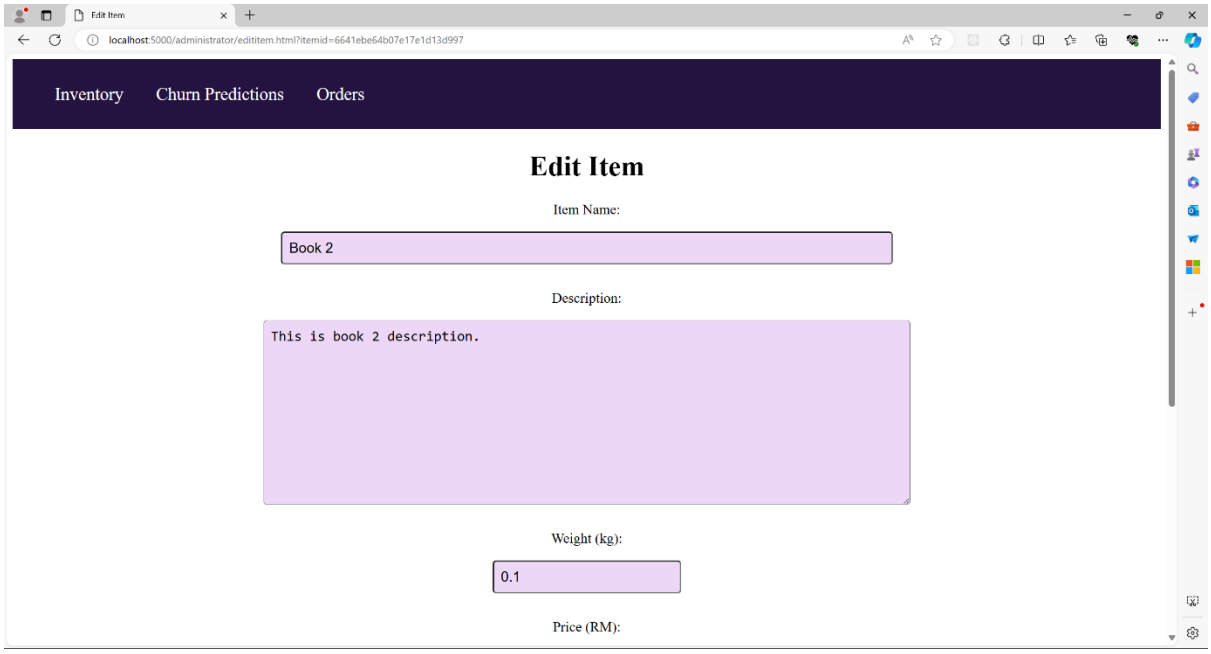


Figure 4.18 Edit item page (part 1 of 2)

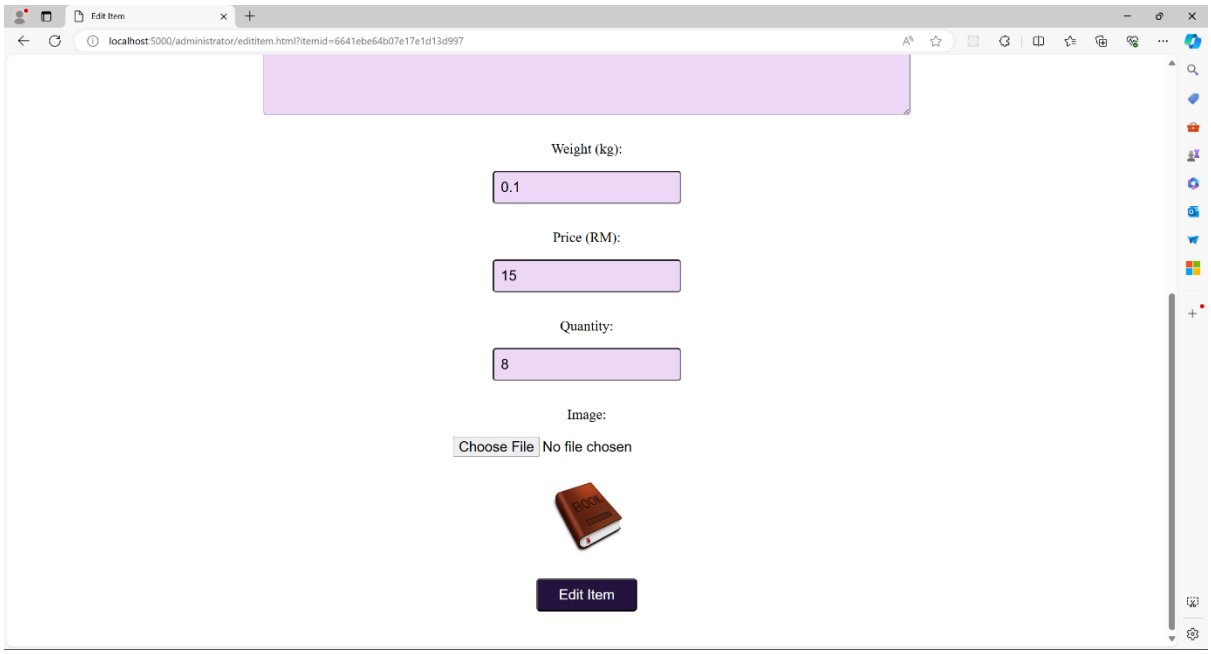


Figure 4.19 Edit item page (part 2 of 2)

On the edit item page, by clicking on the purple “Edit Item” button on the bottom of the page, the updated version of the item will be saved to the database. If the update was successful, the administrator will be navigated to the edit item success page, where the administrator can click on the “Inventory” button to go back to the inventory page. Figure 4.20 is the edit item success page.

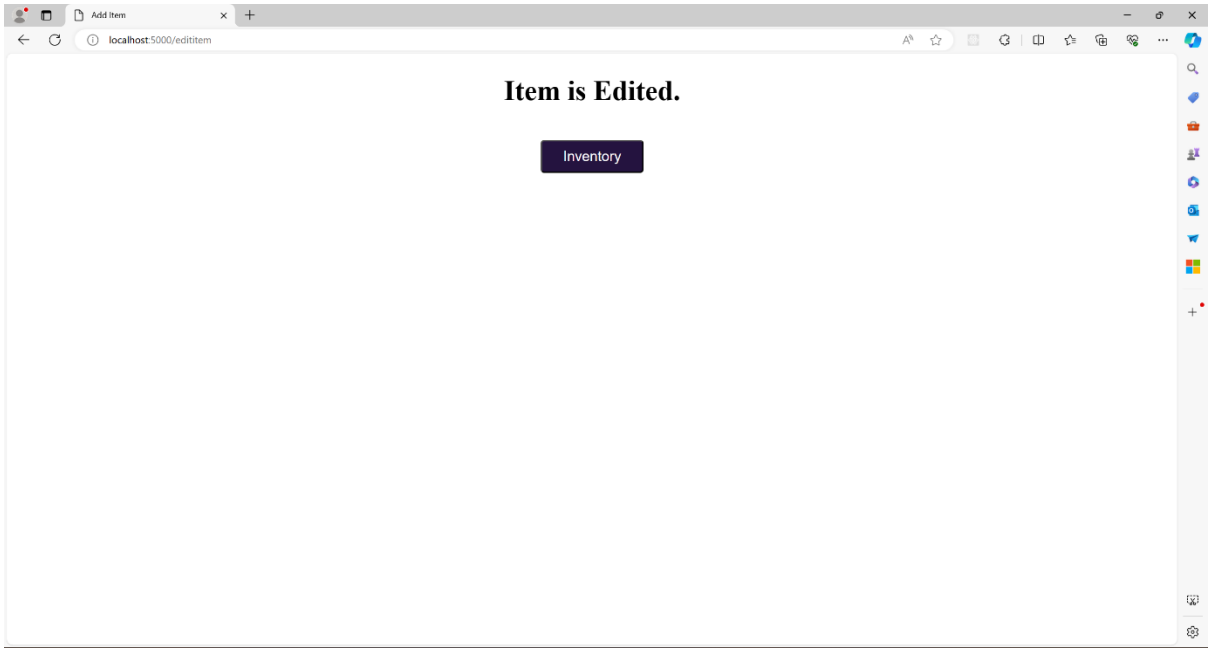


Figure 4.20 Edit item success page

On the inventory page (Figure 4.14), the administrator can click on the red “Delete” button to delete the selected item.

4.2.2. Login

The proposed system uses basic authentication for administrator and customer login. The server sends a response with status “401” and specifies the “WWW-Authenticate” header as “Basic” to trigger the basic authentication process. Figure 4.21 is the basic authentication login.

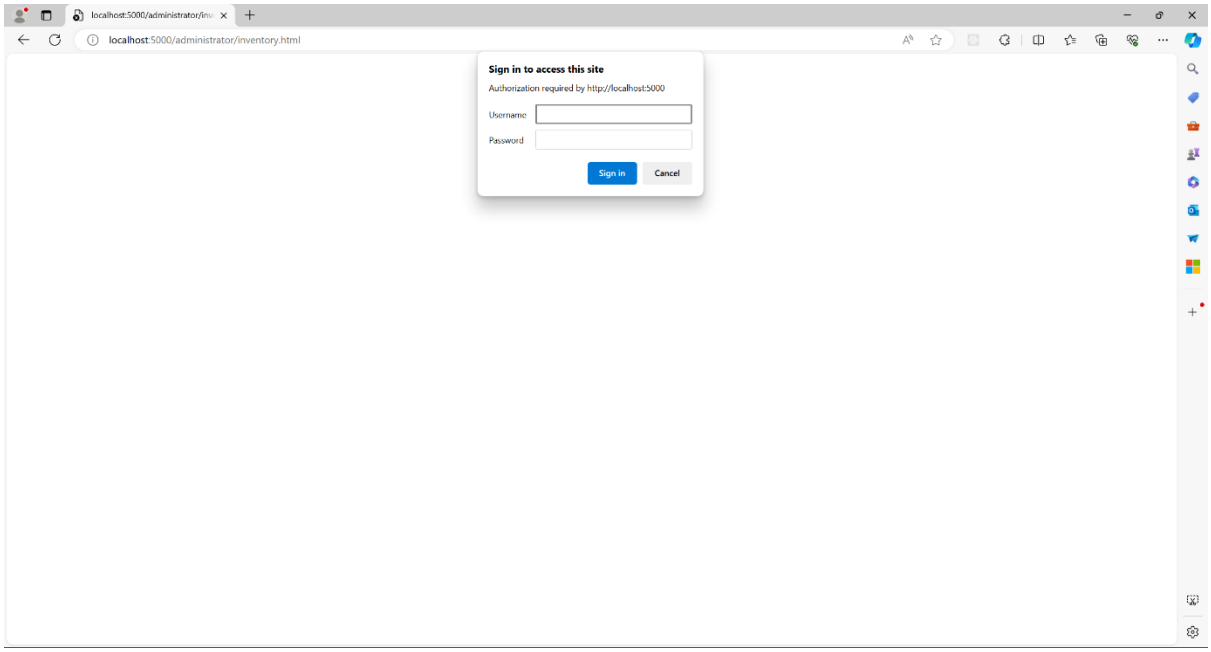


Figure 4.21 Login

The user name and password will be sent in the “Authorization” header in the request. After getting the user name and password, the server will hash the password using PBKDF2 algorithm and the stored salt then compare the hashed password with the stored hash. If the hashed password is same as the stored hash, then the administrator or the customer is granted access. Figure 4.22 is the code for handling login.

```
const crypto = require('crypto');

const AdministratorModel = require('../Database/AdministratorModel');

function adminLogin(req, res, next) {
    const authorization = req.header('Authorization');

    if (authorization == null) {
        console.log('No authorization');
        res.setHeader("WWW-Authenticate", "Basic");
        res.sendStatus(401);
        return;
    }

    const [_, credentials] = authorization.split(' ');
    const decodedCredentials = atob(credentials);
    const [userName, password] = decodedCredentials.split(':');

    console.log('user name:', userName, 'password:', password);

    AdministratorModel.findOne({userName: userName}).then(administrator=>{
        console.log('login found administrator', administrator);

        if (administrator == null) {
            console.log('no such username');

            res.setHeader("WWW-Authenticate", "Basic");
            res.sendStatus(401);

            return;
        }

        crypto.pbkdf2(password, administrator.salt, 100, 512, 'sha-256', (error, derivedKey)=>{
            if (error != null) {
                console.log('administrator login error', error);
                res.sendStatus(500);
                return;
            }

            const derivedPassword = derivedKey.toString('base64');
            console.log('password:', derivedPassword);

            if (derivedPassword === administrator.password) {
                req.userName = userName;
                next();
            }
            else {
                console.log('administrator login fail: password not match');

                res.setHeader("WWW-Authenticate", "Basic");
                res.sendStatus(401);
            }
        });
    }).catch(error=>{
        console.log('Find administrator error', error);
        res.sendStatus(500);
    });
}

module.exports = adminLogin;
```

Figure 4.22 Handle login

4.2.3. Home Page

The home page is accessible by any type of user, but it is meant for the customer to browse the items selling on the proposed system. On the home page. The image, name and price of each item is displayed in grid form. Figure 4.23 is the home page.

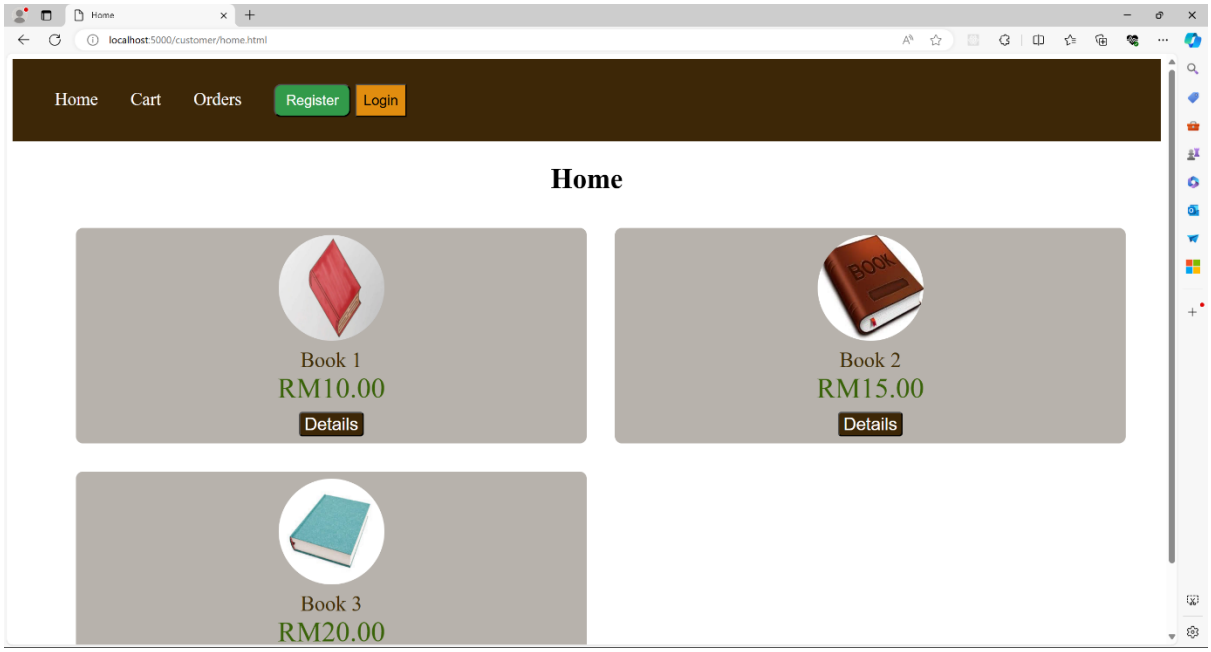


Figure 4.23 Home page

On the home page, the customer can click on the brown “Details” button on any item to view the details of each item. The item details page will be explained in subtopic 4.2.5. The customer can also opt to register by clicking the green “Register” button or login by clicking on the orange “Login” button. The customer register page will be explained in subtopic 4.2.4.

When the customer is logged in, red “Logout” button will appear on the home page instead of “Register” button and “Login” button. The HTML script can know whether the customer is logged in or not by sending a get request to the route “/checklogin”. The server will check whether the “Authorization” header is empty or not and reply in text form, as the message attribute. Upon receiving response, the HTML script will render the buttons accordingly. Figure 4.24 is the code for handling the “/checklogin” route.

```
app.get('/checklogin', (req, res)=>{
    const authorization = req.get('Authorization');

    console.log('authorization', authorization);

    if (authorization == null) {
        res.status(200).json({
            message: 'Not logged in'
        });
    }
    else {
        res.status(200).json({
            message: 'Logged in'
        });
    }
});
```

Figure 4.24 Check login

Figure 4.25 is the home page when the customer is logged in (note the red “Logout” button), whereas Figure 4.23 is the home page when the customer is not logged in (note the green “Register” button and the orange “Login” button).

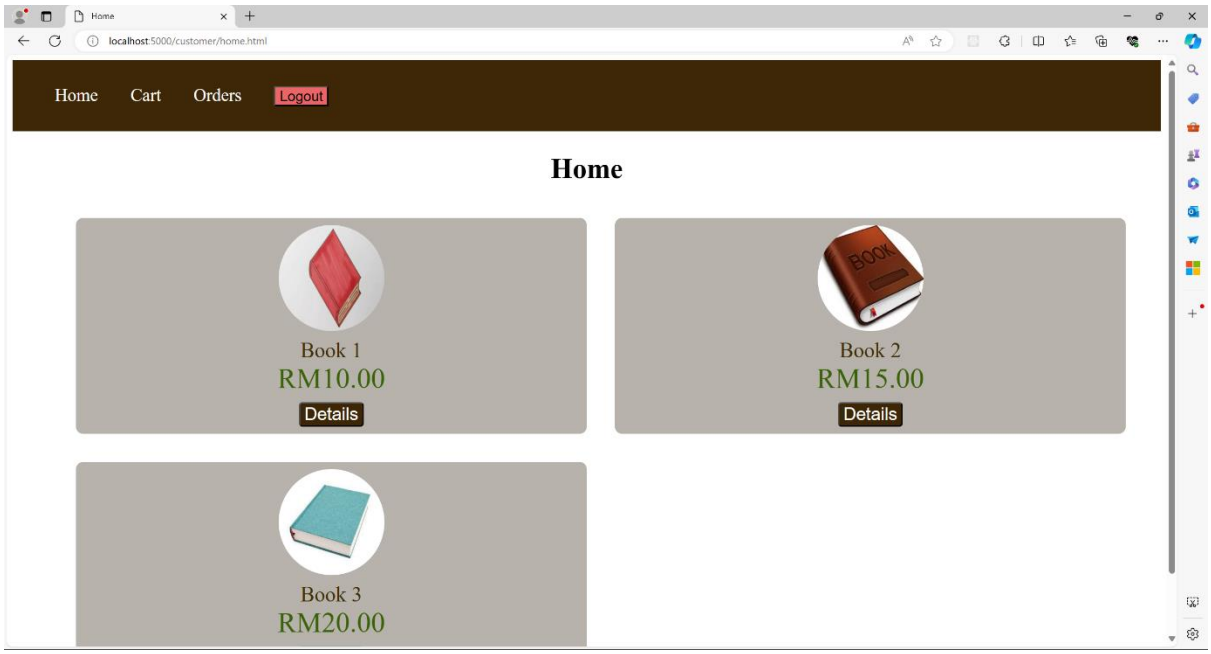


Figure 4.25 Home page (logged in)

4.2.4. Customer Register Page

The customer register page is accessible by any type of user, but it is meant for the customers to register themselves. The customer needs to register before they can login. Without login, the customer can only browse the items on the home page, and cannot add item to cart, place order and access the cart page and orders page. On the customer register page, the customer needs to fill in the user name and the password. The customer needs to repeat the password to confirm that he remembers the password. Figure 4.26 is the customer register page.

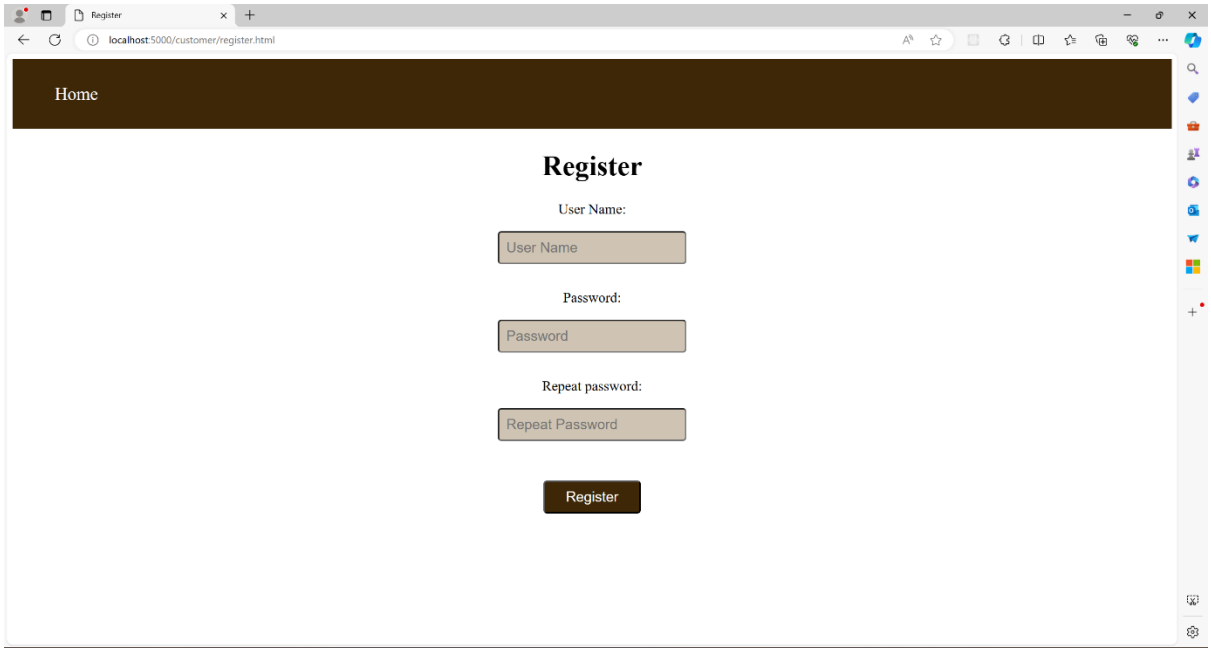


Figure 4.26 Customer register page

When the customer clicks on the brown “Register” button, the user name and password will be saved on the database. If the registration is successful, the customer will be navigated to the customer register success page, where the customer can opt to go back to the home page. Figure 4.27 is the customer register success page.

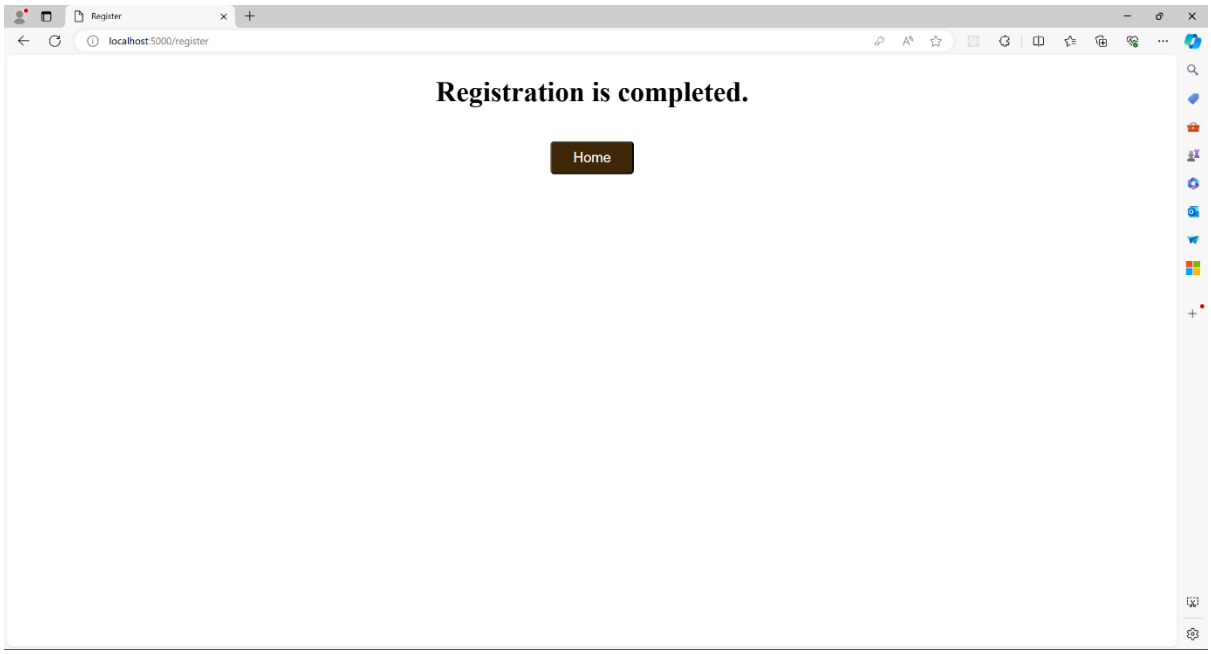


Figure 4.27 Customer register success page

4.2.5. Item Details Page

The item details page is accessible for anyone but it is meant for the customer. On the item details page, the image, name, price, weight and description of the item are displayed. Figure 4.28 is the item details page.

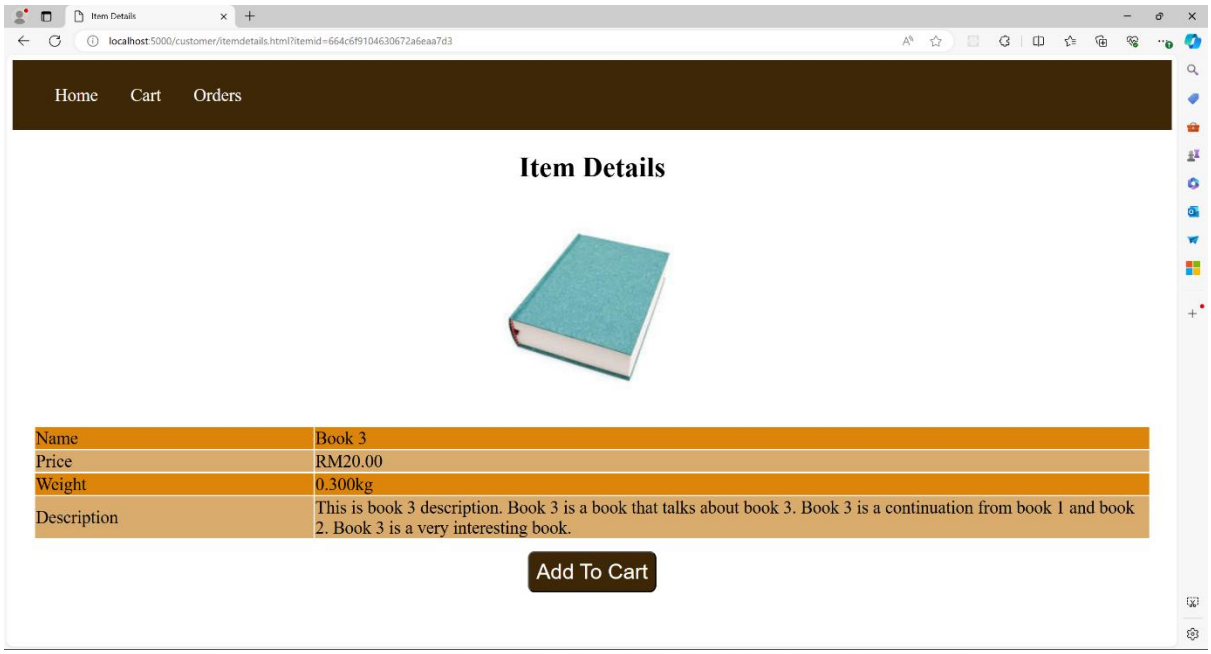


Figure 4.28 Item details page

When the customer clicks the brown “Add To Cart” button, if the customer is not logged in the customer will be asked to login. Figure 4.29 shows the customer is prompted for login upon clicking on the “Add To Cart” button.

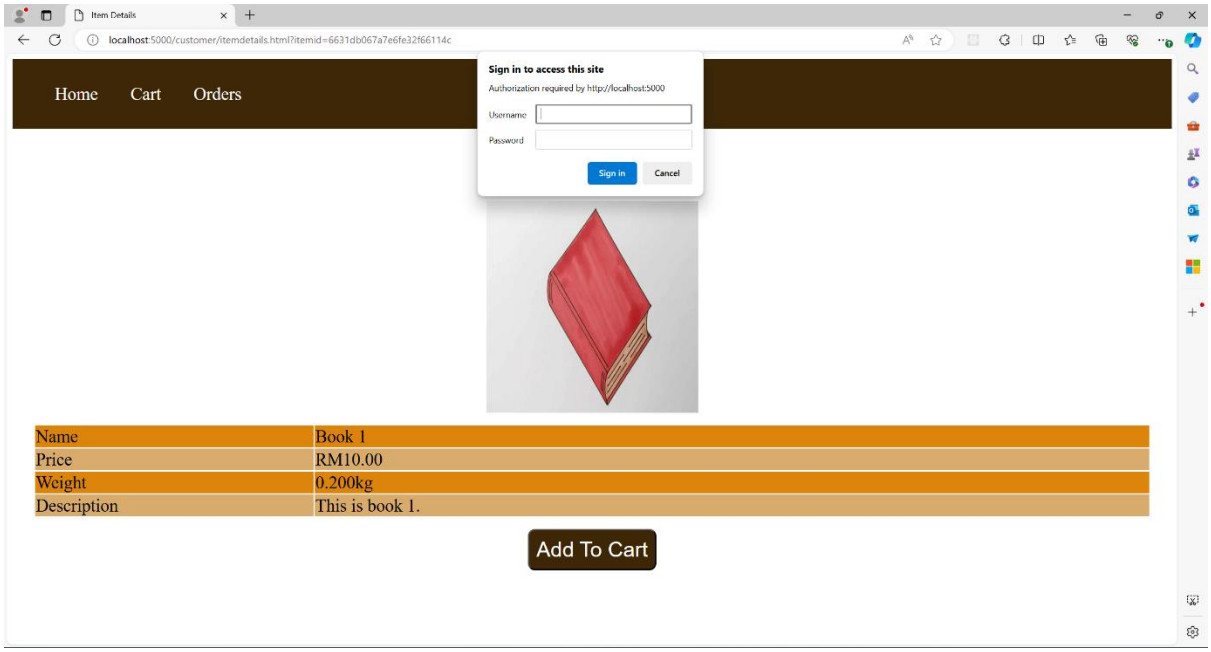


Figure 4.29 Login prompt upon clicking on "Add To Cart" button

After the customer login, then the item is saved to the customer's cart. If the item is successfully added to the cart, then the customer will be navigated to the add to cart success page. Figure 4.30 is the add to cart success page.

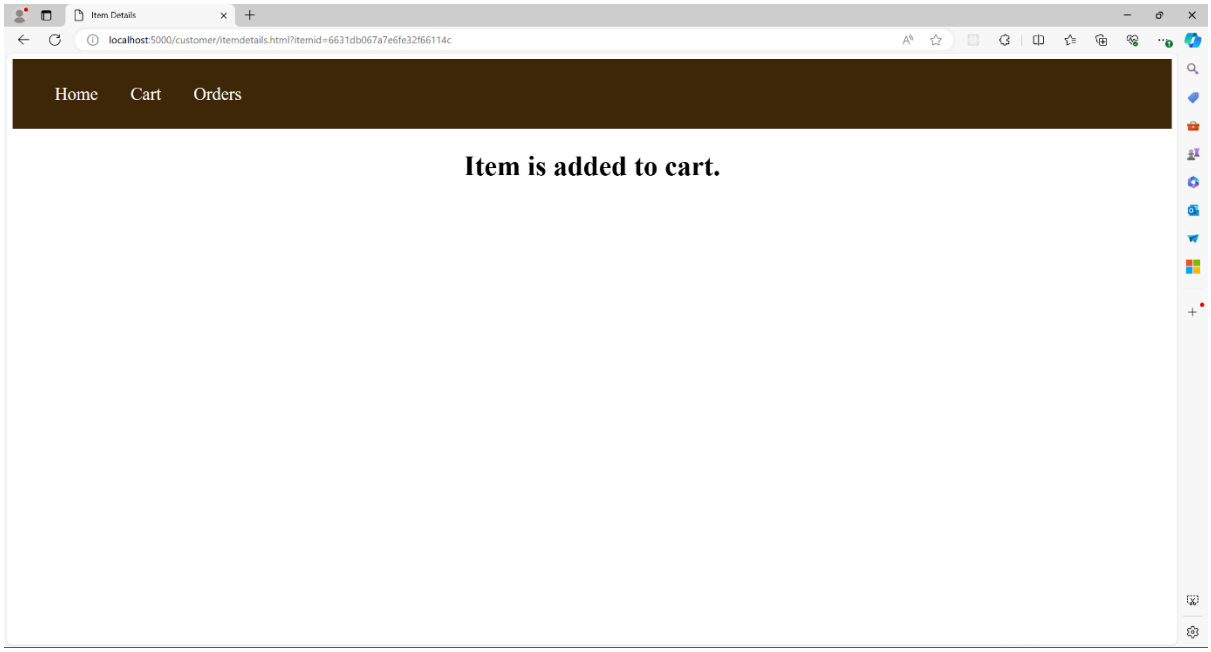


Figure 4.30 Add to cart success page

4.2.6. Cart Page

The cart page is only accessible to the customer. On the cart page, the items in the customer's cart are listed in table and the total amount is displayed. Figure 4.31 is the cart page.

Figure 4.31 Cart page

On the cart page, the customer can add the quantity of any item by clicking the green "Add One" button. The customer can also minus the quantity of any item by clicking the red "Minus One" button.

If the customer's cart is empty, then the message "No item in cart" is displayed instead of the cart table. Figure 4.32 is the cart page when cart is empty.

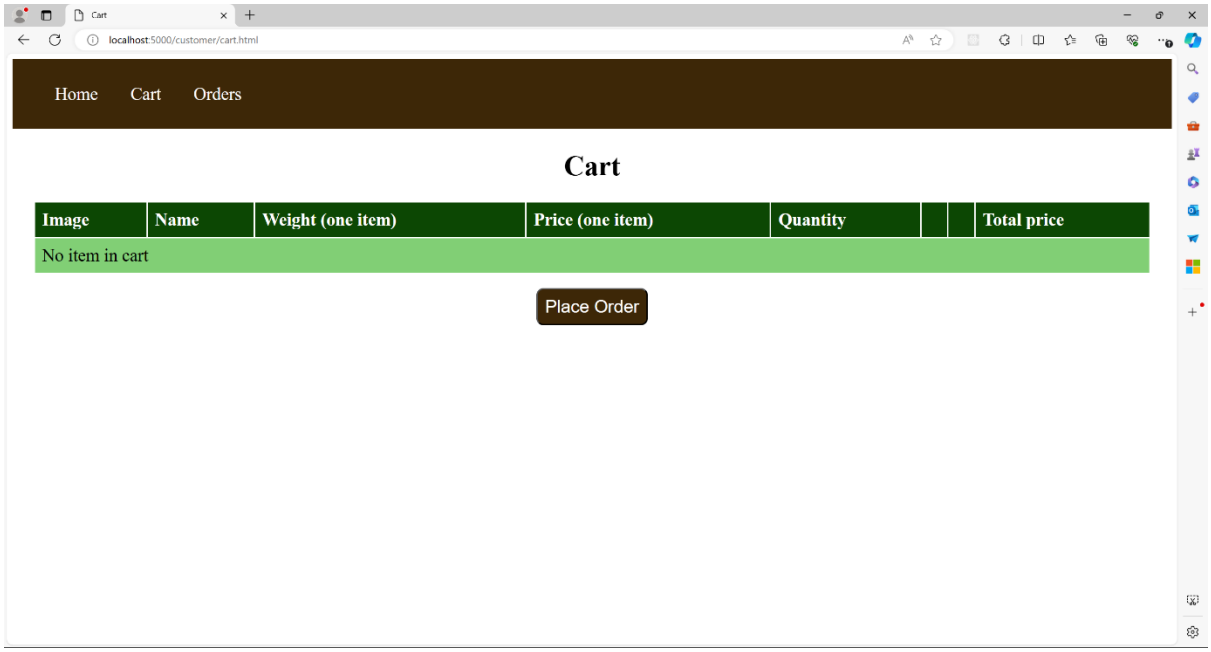


Figure 4.32 Cart page (Empty cart)

On the cart page, if the cart is not empty (Figure 4.31), when the customer clicks the brown “Place Order” button, an order will be created based on the items in the cart. Then, the cart will be emptied. If the order is placed, then the customer will be navigated to the place order success page. Figure 4.33 is the place order success page.

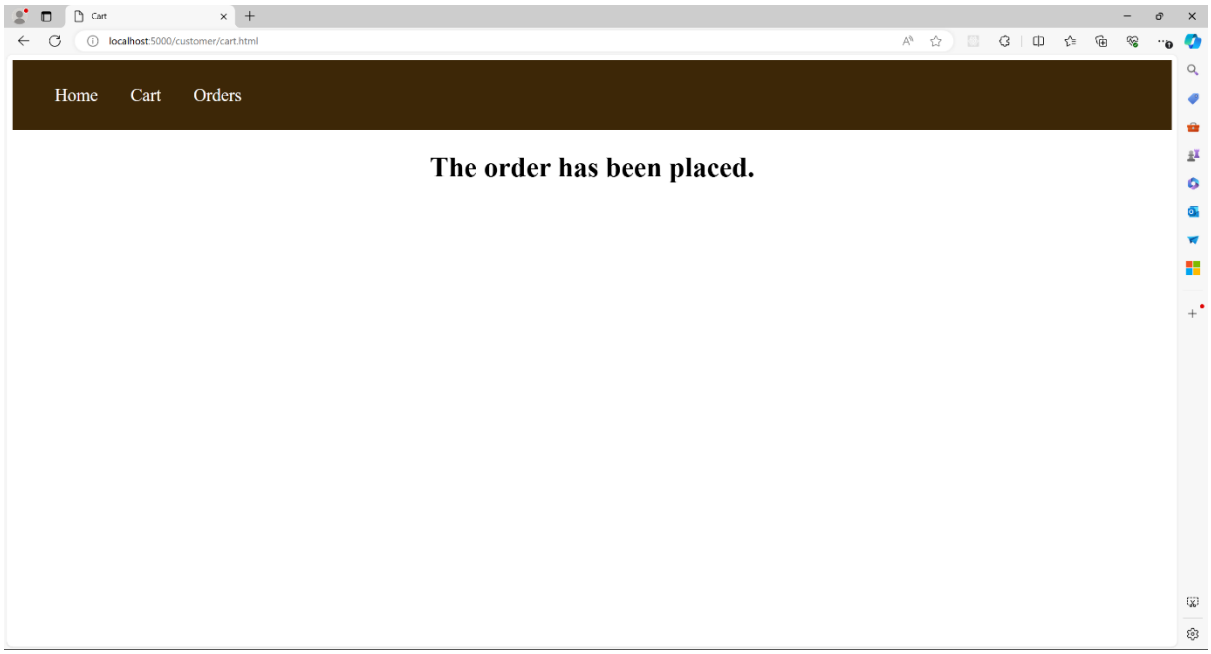


Figure 4.33 Place order success page

4.2.7. Orders Page

The orders page is accessible to both customer and administrator. On the orders page, the administrator can see all orders in the proposed system, whereas the customer can only see his own past orders. Figure 4.34 and Figure 4.35 are the orders page from administrator side and customer side respectively. On administrator side, the order ID, date, customer name and amount of the orders are displayed, whereas on customer side, the order ID, date and amount of the orders are displayed.

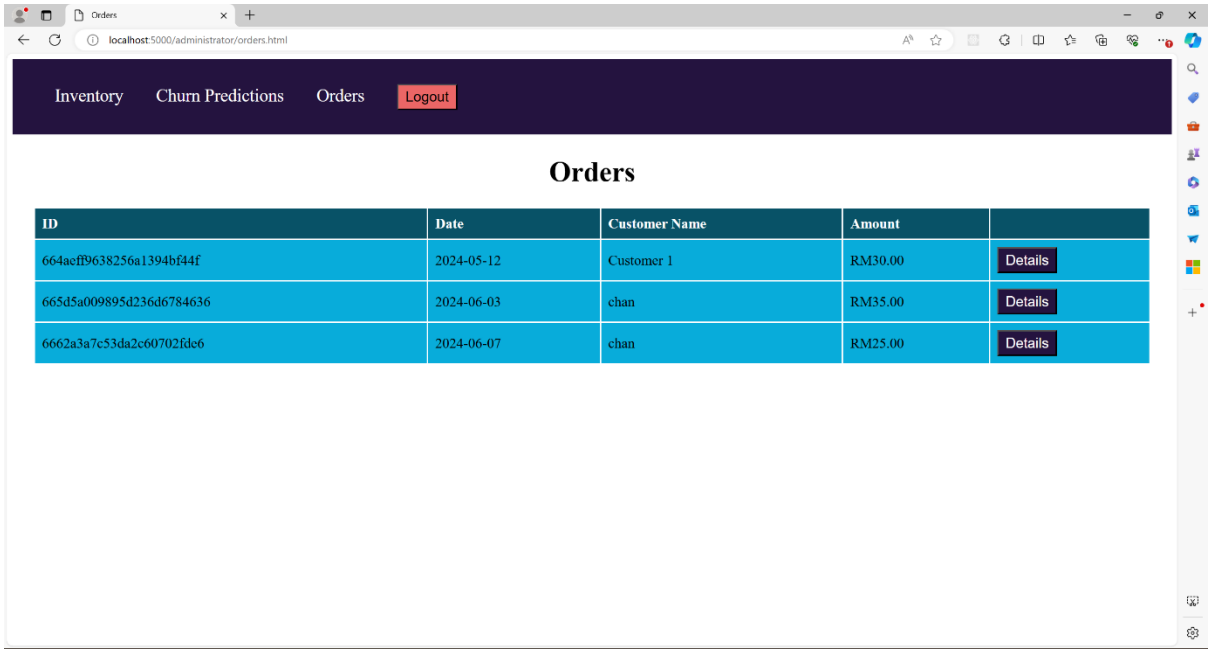


Figure 4.34 Orders page (administrator side)

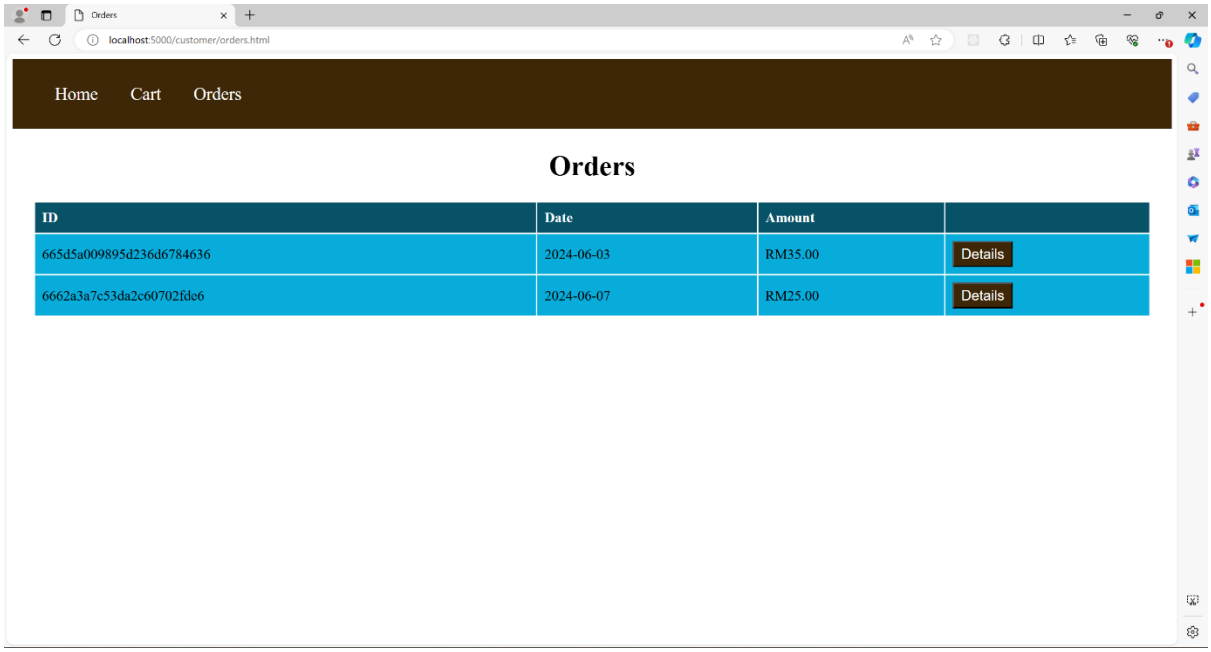


Figure 4.35 Orders page (customer side)

If there are no past orders, “No past order” will be displayed on the orders page, as shown in Figure 4.36.

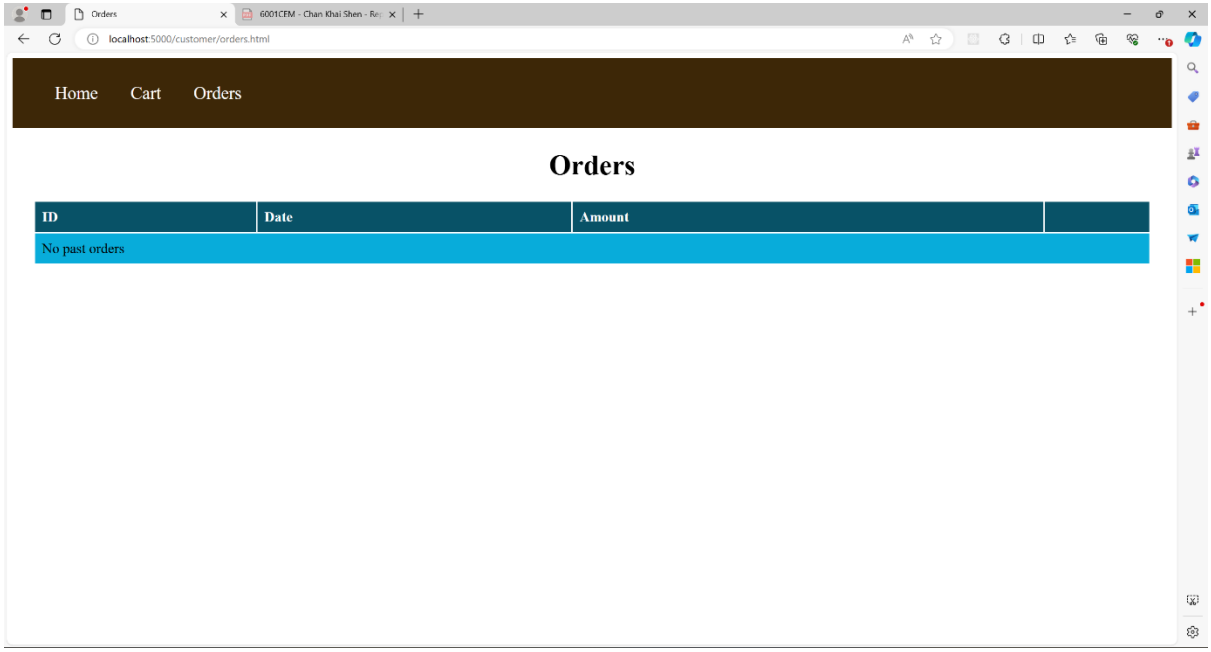


Figure 4.36 Orders page (no past orders)

When the administrator clicks on the purple “Details” button (Figure 4.34), the administrator will be navigated to the order details page. When the customer clicks on the brown “Details” button (Figure 4.35), the customer will be navigated to the orders details page. On the orders details page, date and amount of the order, as well as item name, weight, unit price, quantity and total price of each item, are displayed. On administrator side, the customer

name is displayed, whereas on customer side, the customer name is not displayed. Figure 4.37 and Figure 4.38 are the orders details page from administrator side and customer side respectively.

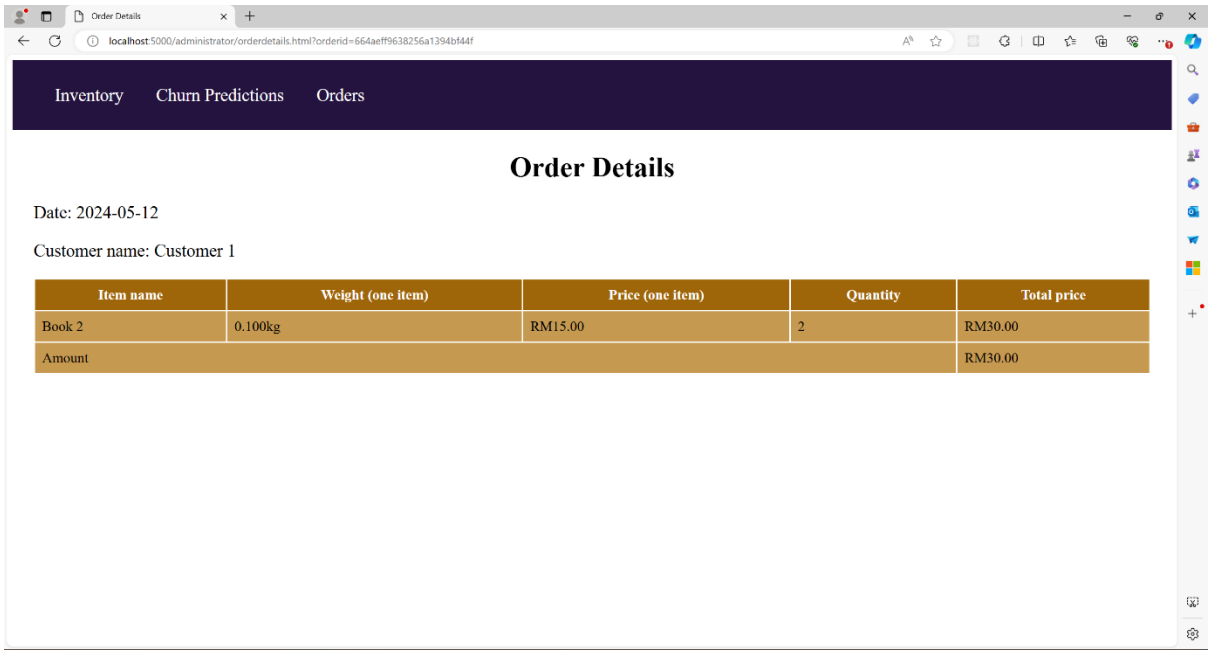


Figure 4.37 Order details (administrator side)

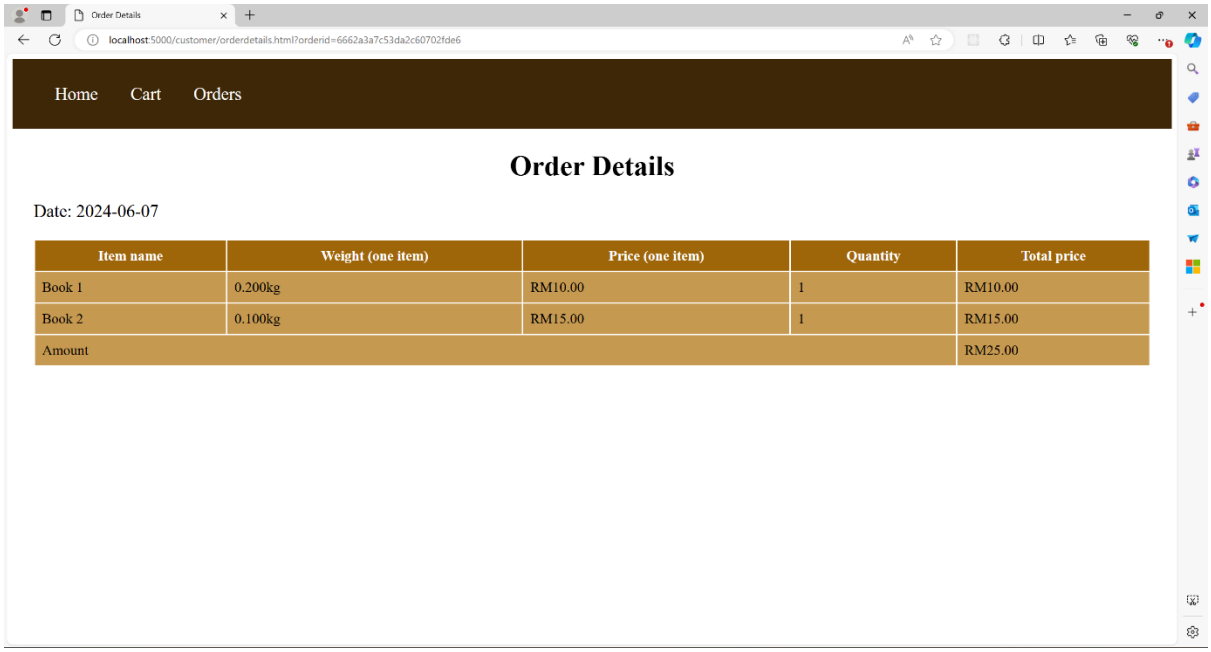


Figure 4.38 Order details (customer side)

5. TESTING

5.1. Unit Testing

Table 5.1 lists out the test cases of unit testing of get predictors function.

Table 5.1 Unit testing of get predictors

Test	Expected output	Actual output	Code

Input logout time stamps as [{timeStamp: new Date('2024-05-01')}]	ses_rec = 31, ses_rec_avg = 99, user_rec = 99	test 1 ses_rec 31 ses_rec_avg 99 user_rec 99	var ses_rec;
			if (logoutTimeStamps.length < 2) {
			ses_rec = 31;
			}
			const listOfDayBetweenSessions = [];
			if (logoutTimeStamps.length < 2) {
			listOfDayBetweenSessions.push(99);
			}
			const ses_rec_avg = mathjs.mean(listOfDayBetweenSessions);
			var user_rec;
			if (logoutTimeStamps.length < 2) {
			user_rec = 99;
			}

Input logout time stamps as [{timeStamp: new Date('2024-05-01')}, {timeStamp: new Date('2024-01-01')}]	ses_rec = 31, ses_rec_avg = 99, user_rec = 99	test 2 ses_rec 31 ses_rec_avg 99 user_rec 99	if (differenceInDays > 31) {
			ses_rec = 31;
			}
			if (differenceInDays > 99) {
			listOfDayBetweenSessions.push(99);
			}
			const ses_rec_avg = mathjs.mean(listOfDayBetweenSessions);
			if (differenceInDays > 99) {
			user_rec = 99;
			}

Input logout time stamp as [{timeStamp: new Date('2024-05-01')}, timeStamp: new Date('2024-05-11')}]	ses_rec = 10, ses_rec_avg = 10, user_rec = 10	test 3 ses_rec 10 ses_rec_avg 10 user_rec 10	else {
			ses_rec = differenceInDays;
			}
			else {
			listOfDayBetweenSessions.push(differenceInDays);
			}
			const ses_rec_avg = mathjs.mean(listOfDayBetweenSessions);
			else {
			user_rec = differenceInDays;
			}

Input logout time stamp as []	ses_n = 1	test 4 ses_n 1	const ses_n = logoutTimeStamps.length === 0 ? 1 : logoutTimeStamps.length;

Table 5.2 lists out the test cases of unit testing of customer register page.

Table 5.2 Unit testing of customer register page

Test	Expected output	Actual output	Code

Input “nnnnnnnnnnnnnnnnnnnn” at user name field	Alert message “User name cannot be longer than 20 characters.” pops out.	Appendix A1	`if (document.getElementById('userName').value.length > 20) {`
			`alert('User name cannot be longer than 20 characters.');`
			`}`

Input “chan” at user name field	Alert message “User name already exists.” pops out.	Appendix A2	`else {`
			`const userName = document.getElementById('userName').value;`
			
			`axios.get('http://localhost:5000/checkduplicate?username=' + userName).then(response=>{`
			`console.log(response);`
			
			`if (response.data.message === 'Duplicate') {`
			`alert('User name already exists.');`
			`}`
			
			`}).catch(error=>{`
			`console.log(error);`
			`});`
			`}`
			
			`app.get('/checkduplicate', (req, res)=>{`
			`const userName = req.query.username;`
			`console.log('check duplicate', userName);`
			
			`CustomerModel.findOne({userName: userName}).then(customer=>{`
			`console.log('Found customer', customer);`
			
			`if (customer == null) {`
			`res.status(200).json({`
			`message: 'No duplicate'`
			`});`
			`}`
			`else {`
			`res.status(200).json({`
			`message: 'Duplicate'`
			`});`
			`}`
			`}).catch(error=>{`
			`console.log(error);`
			`res.sendStatus(500);`
			`});`
			`});`

Input “Myyyyyyyyyy” at password field (no digits)	Alert message “Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet.” pops out.	Appendix A3	`const password = document.getElementById('password').value;`
			
			`if (!/\d/.test(password) || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || password.length < 8) {`
			`alert('Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet.');`
			`}`

Input “M111111111111” at password field (no	Alert message “Password must be at least 8	Appendix A4	`const password = document.getElementById('password').value;`
			
			`if (!/\d/.test(password) || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || password.length < 8) {`
			`alert('Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet.');`
			`}`

lower-case alphabet)	characters long and contain number, lower-case alphabet and upper-case alphabet." pops out.		

Input "n2222 22222" at password field (no upper-case character)	Alert message "Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet." pops out.	Appendix A5	`const password = document.getElementById('password').value;`
			`if (!/\d/.test(password) || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || password.length < 8) {`
			`  alert('Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet.');`
			`}`

Input "My123" at password field (less than 8 characters)	Alert message "Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet." pops out.	Appendix A6	`const password = document.getElementById('password').value;`
			`if (!/\d/.test(password) || !/[a-z]/.test(password) || !/[A-Z]/.test(password) || password.length < 8) {`
			`  alert('Password must be at least 8 characters long and contain number, lower-case alphabet and upper-case alphabet.');`
			`}`

Input "My$Password123" at password field and input "MyPassword123" at repeat password field	Alert message "Password and repeat password not match." pops out.	Appendix A7	`if (document.getElementById('repeatPassword').value !== document.getElementById('password').value) {`
			`  alert('Password and repeat password not match.')`
			`}`

Table 5.3 lists out the test cases of unit testing of add item page.

Table 5.3 Unit testing of add item page

Test	Expected output	Actual output	Code

Input “0” at weight field	Alert message “Weight cannot be less than 0.001.” pops out.	Appendix A8	`if (document.getElementById('weight').value < 0.001) {`
			`  alert('Weight cannot be less than 0.001.');`
			`}`

Input “0” at price field	Alert message “Price cannot be less than 0.01.” pops out.	Appendix A9	`if (document.getElementById('price').value < 0.01) {`
			`  alert('Price cannot be less than 0.01.');`
			`}`

Input “-1” at quantity field	Alert message “Quantity cannot be less than 0.” pops out.	Appendix A10	`if (document.getElementById('quantity').value < 0) {`
			`  alert('Quantity cannot be less than 0.');`
			`}`

Input “0.2” at quantity field	Alert message “Quantity must be integer.” pops out.	Appendix A11	`else if (!/^\d+$/.test(document.getElementById('quantity').value.toString())) {`
			`  alert('Quantity must be integer.');`
			`}`

Create a new item with item name “Book 1”	Navigated to the page which displays “Item name already exists.”.	Appendix A12	`if (error.code === 11000) {  //Duplicate error`
			`  console.log('duplicate item name')`
			`  res.status(400).send(`
			`  <html lang='en'>`
			`  <head>`
			`  <title>Add Item</title>`
			`  <head>`
			`  <style>`
			`  .Content {`
			`  margin: 30px;`
			`  text-align: center;`
			`  }`
			`  .Content h1 {`
			`  font-size: 40px;`
			`  }`
			`  </style>`
			`  <body>`
			`  <div class="Content">`
			`  <h1>Item name already exists.</h1>`
			`  </div>`
			`  </body>`
			`  </html>`
			`  );`
			`  return;`
			`}`

Table 5.4 lists out the test cases of unit testing of edit item page.

Table 5.4 Unit testing for edit item page

Test	Expected output	Actual output	Code

Input "-1" at weight field	Alert message "Weight cannot be less than 0.001." pops out.	Appendix A13	`if (document.getElementById('weight').value < 0.001) {`
			`  alert('Weight cannot be less than 0.001.');`
			`}`

Input "-1" at price field	Alert message "Price cannot be less than 0.01." pops out.	Appendix A14	`if (document.getElementById('price').value < 0.01) {`
			`  alert('Price cannot be less than 0.01.');`
			`}`

Input "-3" at quantity field	Alert message "Quantity cannot be less than 0." pops out.	Appendix A15	`if (document.getElementById('quantity').value < 0) {`
			`  alert('Quantity cannot be less than 0.');`
			`}`

Input "0.2" at quantity field	Alert message "Quantity must be integer." pops out.	Appendix A16	`else if (!/^\d+$/.test(document.getElementById('quantity').value.toString())) {`
			`  alert('Quantity must be integer.');`
			`}`

Table 5.5 is the unit testing for cart page.

Table 5.5 Unit testing for cart page

Test	Expected output	Actual output	Code

Login as "chan2". Go to cart page. The cart is empty. Click "Place Order" button.	Alert message "No item in cart." pops out.	Appendix A17	`if (customer.cartItems == null || customer.cartItems.length === 0) {`
			`  console.log('No item in cart');`
			`  res.status(400).json({`
			`  message: 'No item in cart.'`
			`  });`
			`  return;`
			`}`
			`if (error.response.status === 400) {`
			`  alert(error.response.data.message);`
			`}`

Login as “chan”. Go to cart page. On item “Book 2”, add quantity until “9”. The stock for “Book 2” is “8”. Click “Place Order” button.	Alert message “Book 2 is out of stock.” pops out.	Appendix A18	`const leftQuantity = itemQuantities[j].quantity - cartItems[i].quantity;`
			
			`if (leftQuantity < 0) {`
			`console.log(cartItems[i].itemName, 'is out of stock.');`
			
			`res.status(400).json({`
			`message: cartItems[i].itemName + ' is out of stock.'`
			`});`
			
			`return;`
			`}`
			
			`if (error.response.status === 400) {`
			`alert(error.response.data.message);`
			`}`

Table 5.6 is the unit testing for order details page.

Table 5.6 Unit testing for order details page

Test	Expected output	Actual output	Code

Go to orders page. Click on the order with id “665d5a009895d236d6784636”. On the URL bar, edit the order id to “665d5a009895d236d6784637”. Reload the page.	Alert message “No such order.” pops out.	Appendix A19	`if (order == null) {`
			
			`res.status(400).json({`
			`message: 'No such order.'`
			`});`
			
			`return;`
			`}`
			
			`if (error.response.status === 400) {`
			`alert(error.response.data.message);`
			`}`

5.2. Black box Testing

The test cases of black box testing in Table 5.7 are formed around the save predictors function (explained in subtopic 4.1.4). Customer actions that will invoke the save predictor function, such as logout, view item, add item to cart and place order, are carried out. Then, the churn prediction table on the administrator’s churn predictions page is observed before and after the action. The test cases and the observations are documented in Table 5.7.

Table 5.7 Black box testing of save predictor function

Test	Expected output	Actual output (before)	Actual output (after)

Go to customer home page. Login as “chan”. Then, logout.	On the churn predictions page, the number of sessions (ses_n) of “chan” increases from “2” to “3”.	Appendix A20	Appendix A21
Go to customer home page. Login as “chan”. Click “Details” button for “Book 1”. Then, click “Add To Cart” button. Then, go back to home. Then, logout.	On the churn predictions page, the number of user interactions (int_n)s increases from “1” to “3” because of 2 interactions (view, add to cart).	Appendix A22	Appendix A23
Login as “chan”. Go to cart page. In the cart, there is “Book 1” with quantity “2” and “Book 2 with quantity “1”. The amount is RM35.00. Click “Place Order” button.	On the churn predictions page, the number of user interactions (int_n) increases from “5” to “6”, the number of transactions (tran_n) increases from “0” to “1” and the revenue (rev_sum) increases from “0” to “35”.	Appendix A24	Appendix A25

Table 5.8 is the black box testing of update stock function which is expected to run when the order is placed.

Table 5.8 Black box testing of update stock function

Test	**Expected output**	**Actual output (before)**	**Actual output (after)**
Login as “chan”. Go to cart page. In the cart, there are “Book 1” with quantity “2” and “Book 2” with quantity “1”. Click “Place Order” button.	On the inventory page, the quantity of “Book 1” reduces from “15” to “13” and the quantity of “Book 2” reduces from “8” to “7”.	Appendix A26	Appendix A27

5.3. Testing the Customer Churn Prediction Model

The testing of the customer churn prediction model was carried out by passing the training dataset into the prediction model and comparing the predicted output to the actual output. As the result of the testing process, the counts of rows with true positive, false positive,

true negative and false negative are computed respectively. Figure 5.1 and Figure 5.2 are the code for conducting the test on the prediction model.

```
TrainingDatasetModel.find().then(results=>{

    const xs = [];
    const ys = [];

    for (var i = 0; i < results.length; i++) {
        const x = [
            results[i].ses_rec,
            results[i].ses_rec_avg,
            results[i].ses_rec_sd,
            results[i].ses_rec_cv,
            results[i].user_rec,
            results[i].ses_n,
            results[i].int_n,
            results[i].int_n_r,
            results[i].tran_n,
            results[i].tran_n_r,
            results[i].rev_sum,
            results[i].rev_sum_r
        ];

        xs.push(x);
        ys.push(results[i].target_class);
    }

    const xs_tensor = tf.tensor(xs);

    tf.loadLayersModel('file://' + path.join(__dirname, '/ChurnModel/model.json')).then(model=>{

        const predictions = model.predict(xs_tensor);

        predictions.data().then(predicted=>{

            var truePositive = 0;
            var falsePositive = 0;
            var trueNegative = 0;
            var falseNegative = 0;
```

Figure 5.1 Test prediction model script (part 1 of 2)

```
            for (var j = 0; j < ys.length; j++) {

                if (predicted[j] >= 0.5) {
                    if (ys[j] === 1) {
                        truePositive += 1;
                    }
                    else {
                        falsePositive += 1
                    }
                }
                else if (predicted[j] < 0.5) {
                    if (ys[j] === 0) {
                        trueNegative += 1;
                    }
                    else {
                        falseNegative += 1;
                    }
                }

            }

            console.log('True positive:', truePositive);
            console.log('False positive:', falsePositive);
            console.log('True negative:', trueNegative);
            console.log('False negative:', falseNegative);

        }).catch(error=>{
            console.log('Download predicted values error', error);
        });

    }).catch(error=>{
        console.log('Load model error', error);
    })

}).catch(error=>{
    console.log('Load training dataset error', error);
});
```

Figure 5.2 Test prediction model script (part 2 of 2)

Figure 5.3 is the result of testing the prediction model. The threshold for the prediction is 0.5, which means if the result of the prediction is larger or equal to 0.5, it is considered as “churn”, else, it is considered as “not churn”. Among the 49358 rows, the number of true positives, number of false positives, number of true negatives and number of false negatives are 41589, 4725, 922 and 2122 respectively.

```
PROBLEMS    TERMINAL    DEBUG CONSOLE    OUTPUT    PORTS

Listening from port 5000
Loaded saved model.
Connected to database.
True positive: 41589
False positive: 4725
True negative: 922
False negative: 2122
```

Figure 5.3 Test prediction model result

Figure 5.4 is the confusion matrix of the prediction model, drawn from the number of true positives, number of false positives, number of true negatives and number of false negatives.

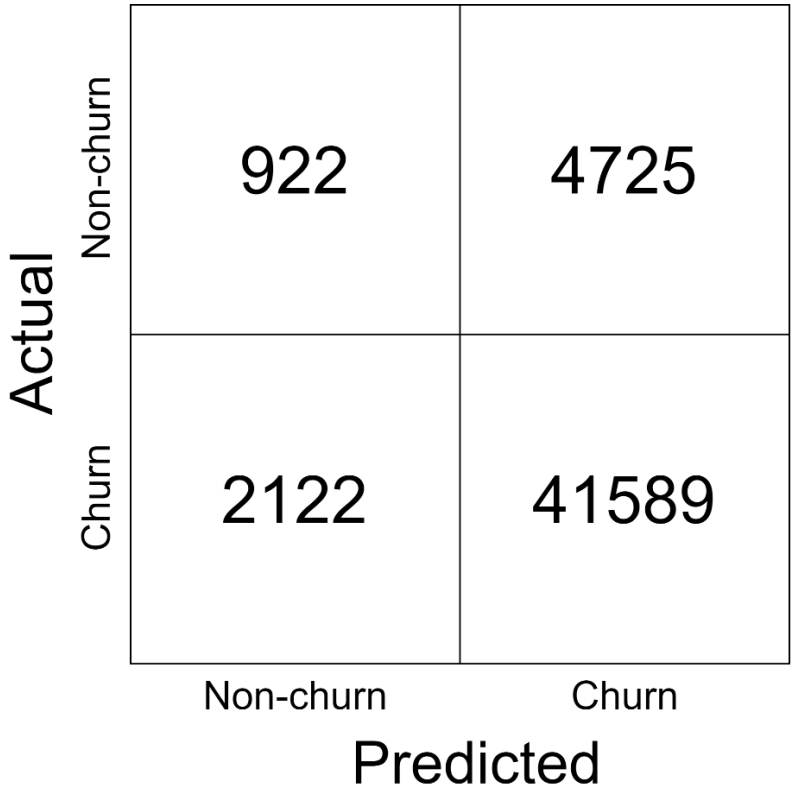


Figure 5.4 Confusion matrix

Based on the confusion matrix, the accuracy of the prediction model is 0.8613 or 86.13%. The equation for accuracy is in Equation 3.

Equation 3 Accuracy (Ismail, et al., 2015)

$$Accuracy = \frac{True\ positive + True\ negative}{True\ positive + False\ positive + True\ negative + False\ negative}$$

6. DISCUSSIONS

In the proposed system, an electronic commerce book shop website using Express.js is developed. After the literature review of existing electronic commerce websites, it is concluded that an ordinary electronic commerce website comprises of upload item, view item, add to cart and checkout for payment. The proposed system consists of the first three components, which is upload item, view item and add to cart. The proposed system will allow the book shop administrator to view the orders to prepare and pack the items and the customer might need to collect the items and pay at the book shop. For future enhancement, the checkout for payment component shall be considered to be added to the proposed system so that the customer does not need to go to the book shop for payment and collection of items.

In the proposed system, a customer churn prediction model based on neural network is trained on the server based on the training dataset originated from the work of Fridrich & Dostal (2022). The 12 predictor variables in the training dataset are formed around user session, user-application interaction and transaction. The proposed system is modified so that it can capture customer logout (user session), view and add to cart (user-application interaction) and place order (transaction). There is only one predicted variable, which is churn or not churn. The model is trained with one dense input layer with ReLU activation function, one dense hidden layer with ReLU activation function and one dense output layer with sigmoid activation function. The model is tested with a threshold of 0.5, which means if the predicted variable is larger than or equal to 0.5, then the customer is likely to churn, else the customer is not likely to churn. The accuracy of the model is 0.861 or 86.1%. This is actually slightly lower than the highest accuracy achieved in the work of Fridrich & Dostal (2022), which is 0.889. Comparing to the accuracies of the existing works reviewed which ranged between 0.740 and 0.976, the accuracy achieved in this project is not too high and not too low. In future research, the configuration of neural network can be fine tuned so that the prediction model is able to achieve a higher accuracy.

7. CONCLUSION

In this project, the proposed system allows the administrator to create stock and view orders. The proposed system also allows the customer to browse items, choose items and place order. So, the first objective is achieved. On the other hand, in this project, a customer churn prediction model based on neural network is trained. The predictors of model can be formed from customer logout time stamp, view item or add to cart time stamp and transaction. In the proposed system, the churn prediction model predicts the churn probability for each customer and displays the churn probability in a table, which is accessible to the administrator. So, the second objective is also achieved.

REFERENCES

AbdAlameer, E. M. T., 2014. Building and Developing E-commerce Website. *International Journal of Science and Research (IJSR),* 3(9), pp. 1419-1425.

Ariyani, W., Hanantjo, D. & Purnama, B. E., 2015. E-Commerce Web Development in Wiga Art. *International Journal of Science and Research (IJSR),* 4(5), pp. 379-384.

Axios, n.d. *Getting Started.* [Online] Available at: https://axios-http.com/docs/intro [Accessed 16 May 2024].

Axios, n.d. *Minimal Example.* [Online] Available at: https://axios-http.com/docs/example [Accessed 16 May 2024].

Defuse Security, 2021. *Salted Password Hashing - Doing It Right.* [Online] Available at: https://crackstation.net/hashing-security.htm [Accessed 16 May 2024].

Domingos, E., Ojeme, B. & Daramola, O., 2021. Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction in the Banking Sector. *Computation,* 9(34).

Dongare, A. D., Kharde, R. R. & D. Kachare, A., 2012. Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology (IJEIT),* 2(1), pp. 189-194.

dougwilson & troygoode, 2018. *cors.* [Online] Available at: https://www.npmjs.com/package/cors [Accessed 19 May 2024].

Fridrich, M. & Dostal, P., 2022. User Churn Model in E-Commerce. *Scientific Papers of the University of Pardubice, Series D: Faculty of Economics and Administration ,* 30(1), pp. 1478-1489.

Gallo, A., 2014. *The Value of Keeping the Right Customers.* [Online] Available at: https://hbr.org/2014/10/the-value-of-keeping-the-right-customers [Accessed 17 May 2024].

Gandhi, G., 2024. *How to Read *.CSV File in JavaScript?.* [Online] Available at: https://www.naukri.com/code360/library/how-to-read-csv-file-in-javascript [Accessed 15 May 2024].

Geeks For Geeks, 2021. *Node.js crypto.pbkdf2() method.* [Online] Available at: https://www.geeksforgeeks.org/node-js-crypto-pbkdf2-method/ [Accessed 15 May 2024].

Geeks For Geeks, 2024. *Node crypto.randomBytes() Method.* [Online] Available at: https://www.geeksforgeeks.org/node-js-crypto-randombytes-method/ [Accessed 15 May 2024].

Gupta, A., 2014. E-Commerce: Role of E-Commerce in Today's Business. *International Journal of Computing and Corporate Research,* 4(1).

HTML Hints, 2020. *Get Query Parameters From a URL in JavaScript.* [Online] Available at: https://www.htmlhints.com/article/get-query-parameters-from-a-url-in-javascript/112
[Accessed 23 May 2024].

IBM, 2024. *HTTP Basic Authentication.* [Online] Available at: https://www.ibm.com/docs/en/cics-ts/5.4?topic=concepts-http-basic-authentication
[Accessed 15 May 2024].

Ismail, M. R., Awang, M. K., Rahman, M. N. A. & Makhtar, M., 2015. A Multi-Layer Perceptron Approach for Customer Churn Prediction. *International Journal of Multimedia and Ubiquitous Engineering,* 10(7), pp. 213-222.

Jaeyalakshmi, M. et al., 2020. Prediction of Customer Churn on e-Retailing. *International Journal of Recent Technology and Engineering (IJRTE),* 8(6), pp. 5541-5545.

Jain, V., Malviya, B. & Arya, S., 2021. An Overview of Electronic Commerce (e-Commerce). *Journal of Contemporary Issues in Business and Government,* 27(3).

jcblw, scottmotte, motdotla & motdotenv, 2024. *dotenv 16.4.5 ReadMe.* [Online] Available at: https://www.npmjs.com/package/dotenv?activeTab=readme
[Accessed 26 April 2024].

Jorge, L., 2023. *A Comprehensive Guide to Uploading and Saving Images with Node.js, Express.js, and JavaScript.* [Online] Available at: https://medium.com/@livajorge7/a-comprehensive-guide-to-uploading-and-saving-images-with-node-js-express-js-and-javascript-69137e227c4d
[Accessed 3 May 2024].

Mdn Web Docs, n.d. *<input type="file">.* [Online] Available at: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file
[Accessed 3 May 2024].

MDN Web Docs, n.d. *Express Tutorial Part 3: Using a Database (with Mongoose).* [Online] Available at: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose
[Accessed 19 May 2024].

Mdn Web Docs, n.d. *FileReader: readAsDataUrl() method.* [Online] Available at: https://developer.mozilla.org/en-US/docs/Web/API/FileReader/readAsDataURL
[Accessed 3 May 2024].

MDN Web Docs, n.d. *Promise.all().* [Online] Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/all
[Accessed 31 May 2024].

Mongoose, n.d. *Schemas.* [Online] Available at: https://mongoosejs.com/docs/guide.html
[Accessed 19 May 2024].

Naidu, N. D. et al., 2021. E-Commerce Web Application By Using MERN. *International Journal for Modern Trends in Science and Technology,,* 7(05), pp. 1-5.

Pondel, M. et al., 2021. Deep Learning for Customer Churn Prediction in E-commerce Decision Support. *24th International Conference on Business Information Systems (BIS 2021) .*

remy, 2024. *nodemon.* [Online] Available at: https://www.npmjs.com/package/nodemon [Accessed 19 May 2024].

Sabbeh, S. F., 2018. Machine-Learning Techniques for Customer Retention: A Comparative Study. *International Journal of Advanced Computer Science and Applications (IJACSA),* 9(2), pp. 273-281.

Seymen, O. F. et al., 2023. Customer Churn Prediction Using Ordinary Artificial Neural Network and Convolutional Neural Network Algorithms: A Comparative Performance Assessment. *Gazi University Journal of Science,* 36(2), pp. 720-733.

Shaaban, E., Helmy, Y., Khedr, A. & Nasr, M., 2012. A Proposed Churn Prediction Model. *International Journal of Engineering Research and Applications (IJERA),* 2(4), pp. 893-897.

Sharma, A. K., 2021. BIG BUY (E- Commerce Website) By Using Frontend Web. *International Journal for Modern Trends in Science and Technology,* 7(11), pp. 201-208.

Soegoto, E. S. & Suripto, A., 2018. Design of E-commerce Information System on Web-based Online Shopping. *IOP Conf. Series: Materials Science and Engineering.*

Tang, H. Y. & Ya'acob, S., 2023. E-Commerce Customer Churn Prediction for the Marketplace in Malaysia. *Open International Journal of Informatics (OIJI),* 11(2), pp. 58-66.

Tariq, M. U., Babar, M., Poulin, M. & Khattak, A., 2021. Distributed Model for Customer Churn Prediction using Convolutional Neural Network. *Journal of Modelling in Management.*

Tensor Flow, n.d. *Models and Layers.* [Online] Available at: https://www.tensorflow.org/js/guide/models_and_layers [Accessed 15 May 2024].

Tensor Flow, n.d. *Save and Load Models.* [Online] Available at: https://www.tensorflow.org/js/guide/save_load [Accessed 15 May 2024].

Tensor Flow, n.d. *Tensors and Operations.* [Online] Available at: https://www.tensorflow.org/js/guide/tensors_operations [Accessed 15 May 2024].

Tensor Flow, n.d. *Training Models.* [Online] Available at: https://www.tensorflow.org/js/guide/train_models [Accessed 15 May 2024].

Tenzin, S., Lhamo, T. & Dorji, T., 2022. Design and Development of E-Commerce Web Application for Cooperative Store. *International Research Journal of Engineering and Technology (IRJET),* 9(2), pp. 843-847.

Tutorials Point, n.d. *SDLC - Iterative Model.* [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm [Accessed 19 May 2024].

W3 Schools, n.d. *CSS grid-template-columns Property.* [Online] Available at: https://www.w3schools.com/cssref/pr_grid-template-columns.php [Accessed 24 May 2024].

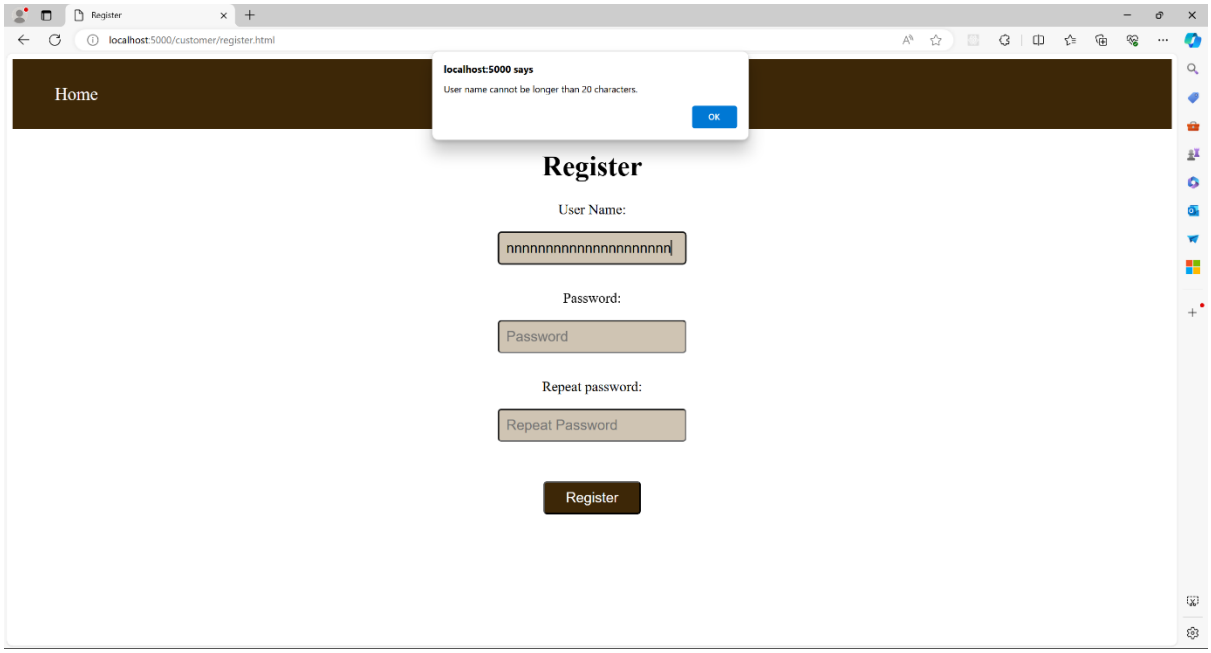
W3 Schools, n.d. *HTML Form Attributes.* [Online] Available at: https://www.w3schools.com/html/html_forms_attributes.asp [Accessed 16 May 2024].

W3 Schools, n.d. *HTML Table.* [Online] Available at: https://www.w3schools.com/html/html_tables.asp [Accessed 19 May 2024].

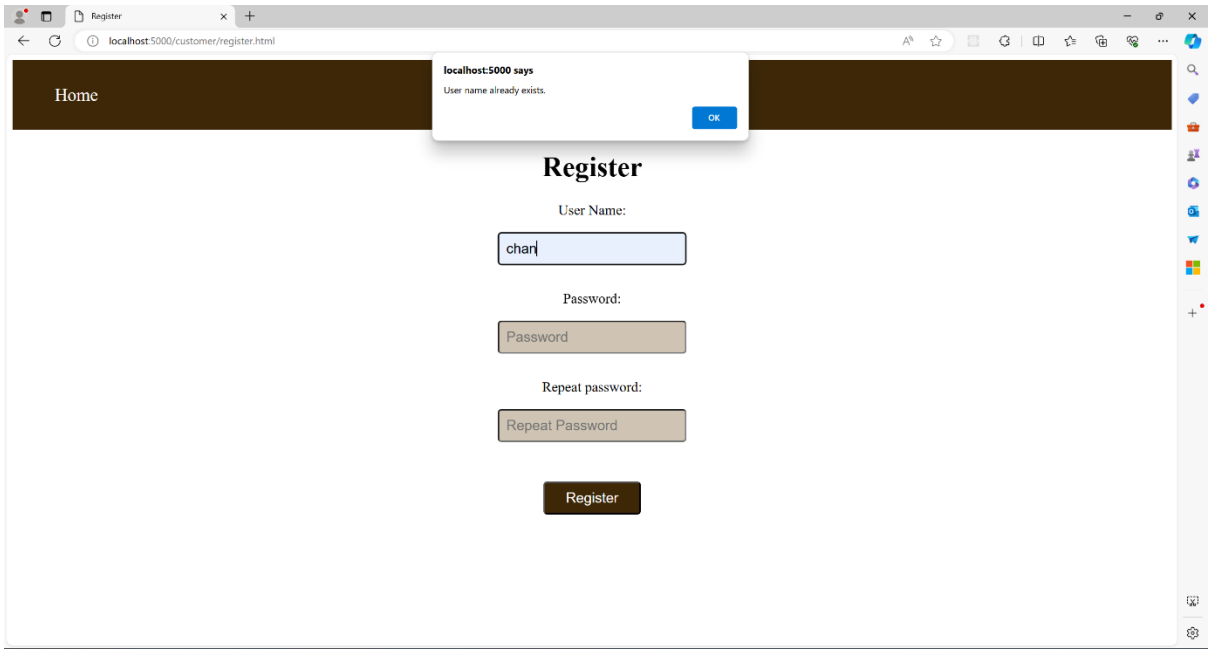
W3 Schools, n.d. *HTML Table Colspan & Rowspan.* [Online] Available at: https://www.w3schools.com/html/html_table_colspan_rowspan.asp [Accessed 31 May 2024].

APPENDICES

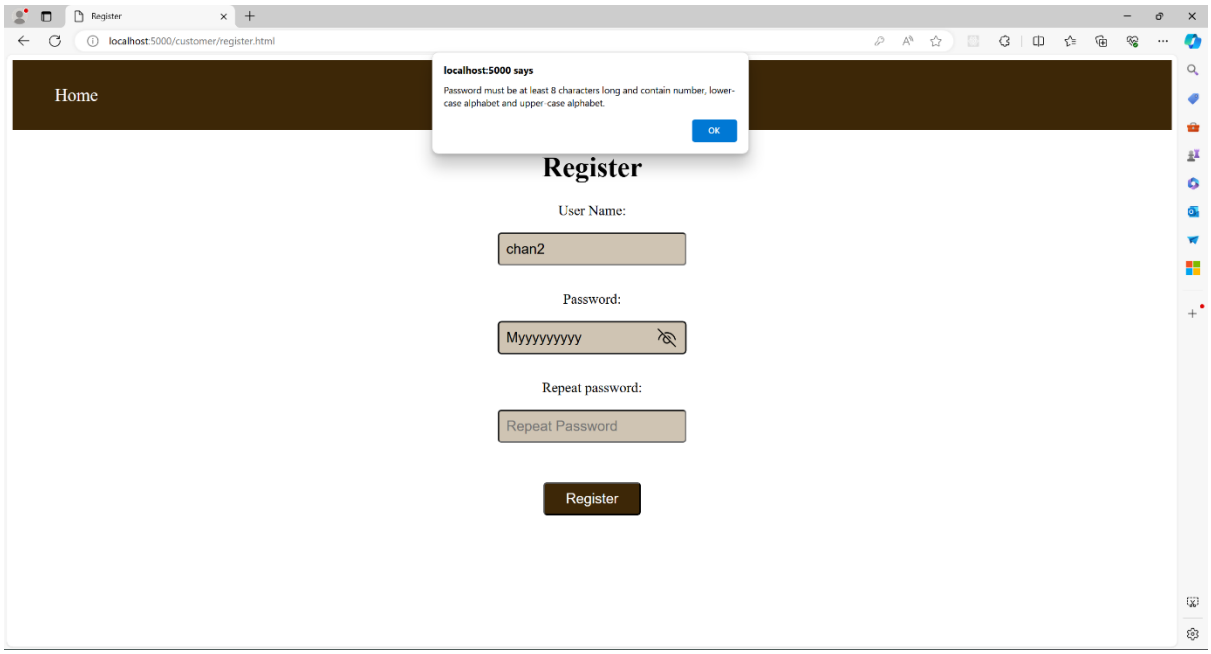
Appendix A1



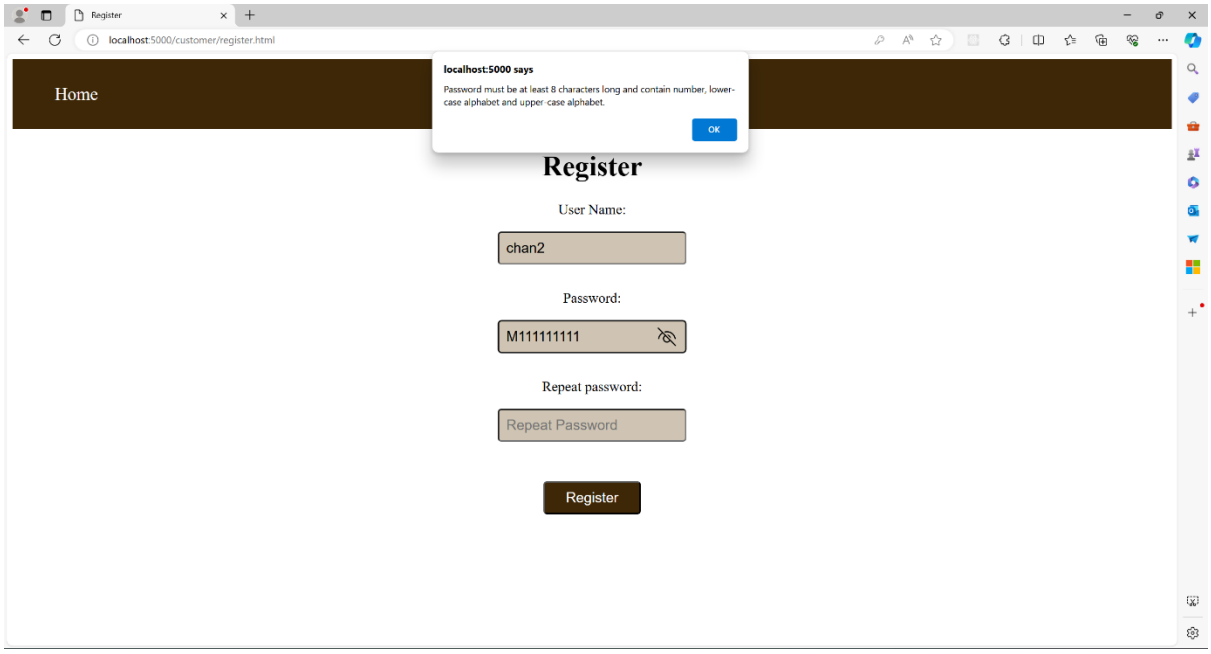
Appendix A2



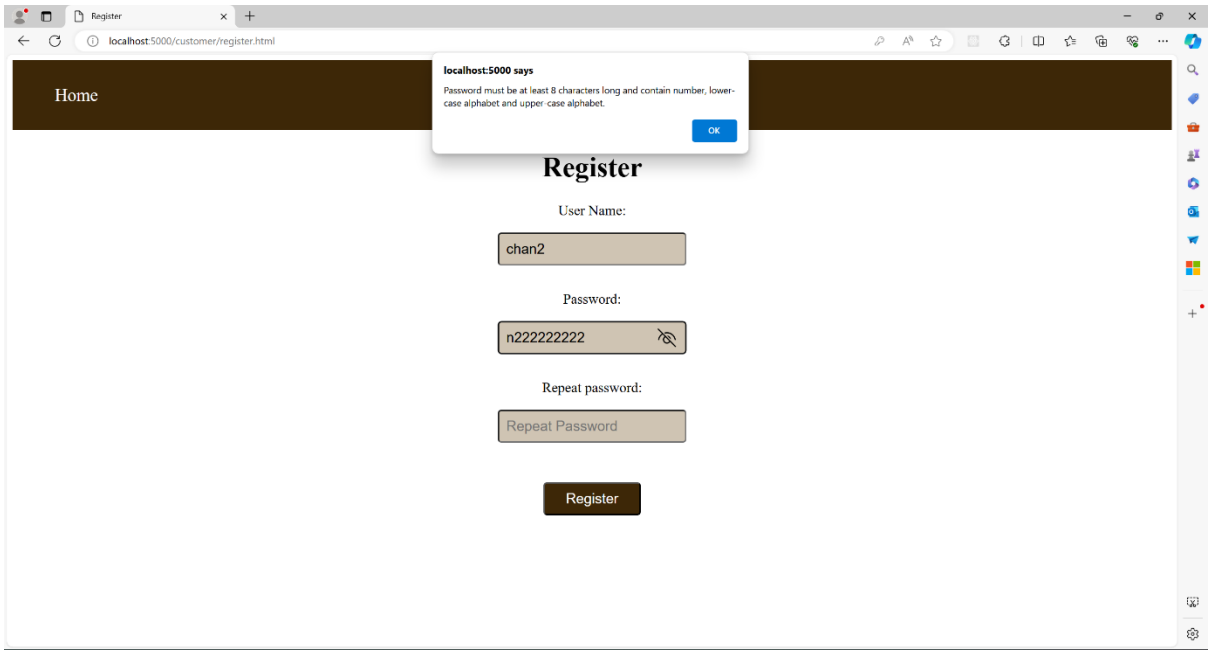
Appendix A3



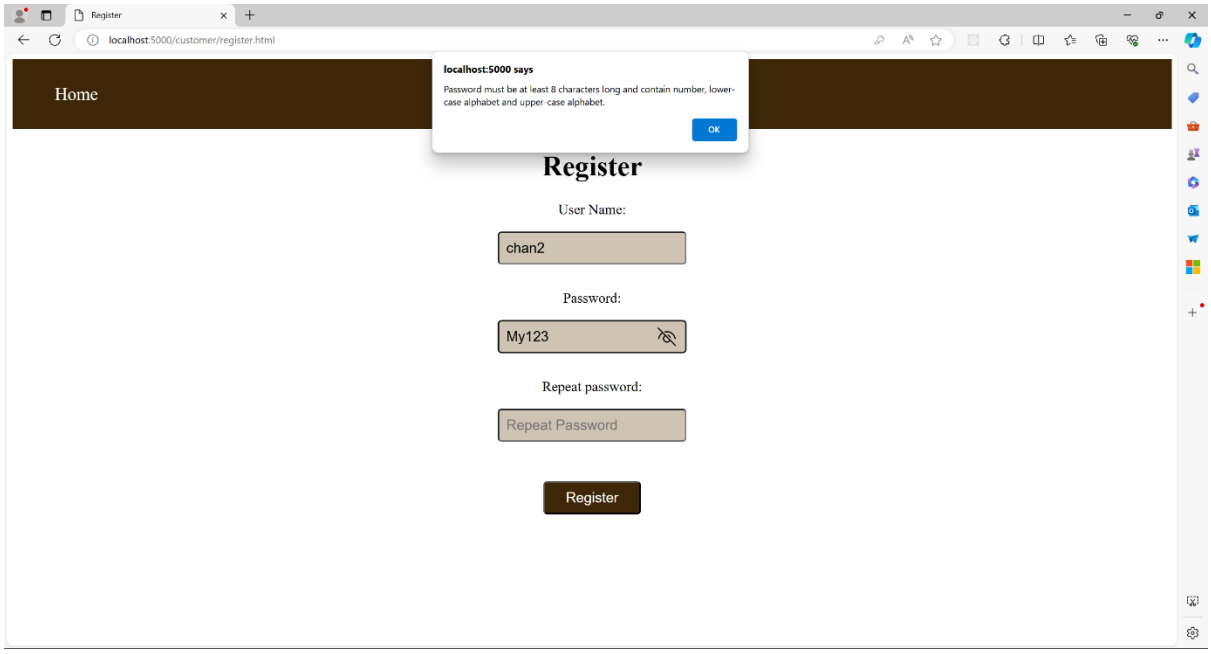
Appendix A4



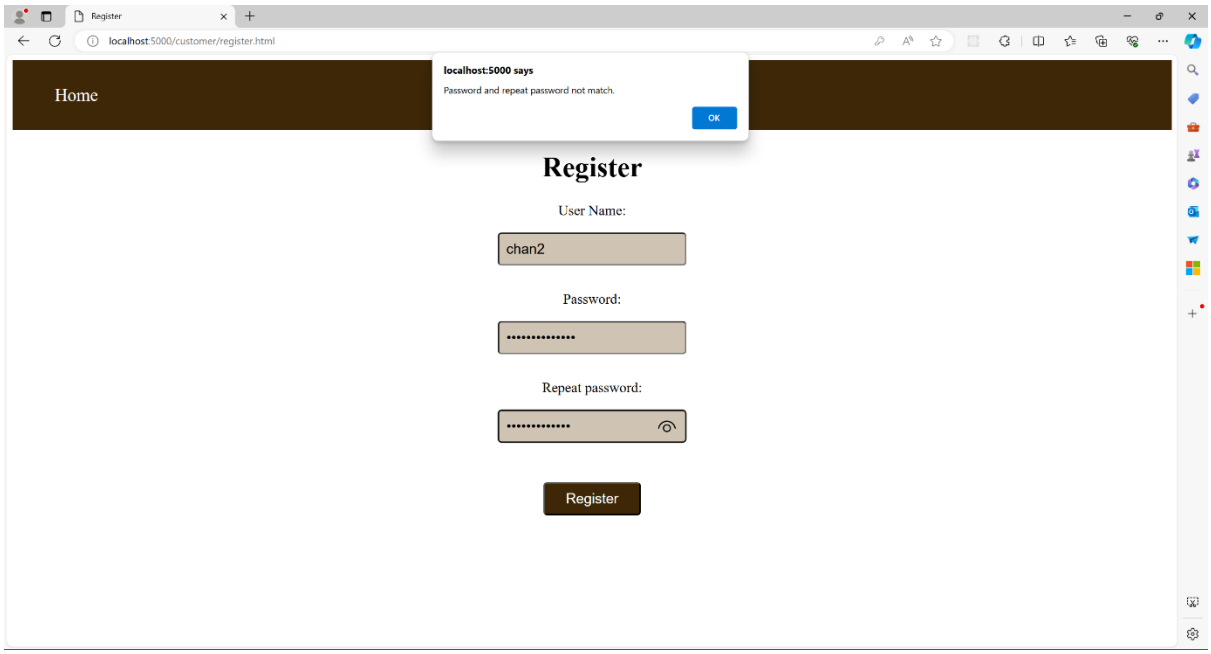
Appendix A5



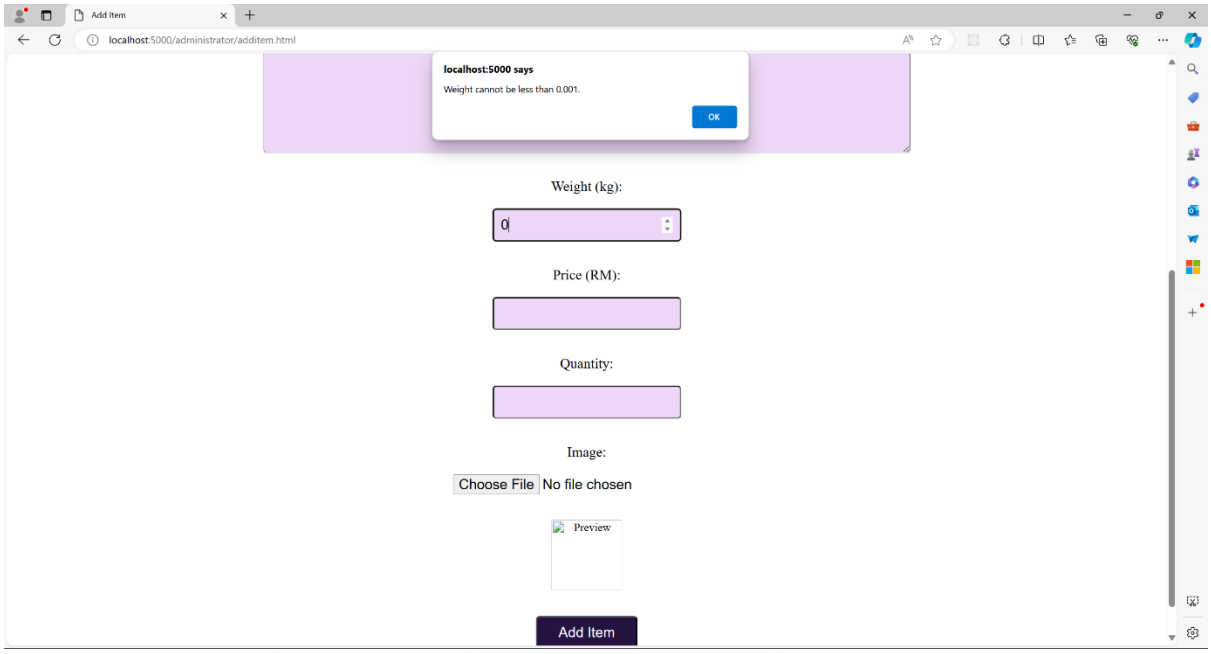
Appendix A6



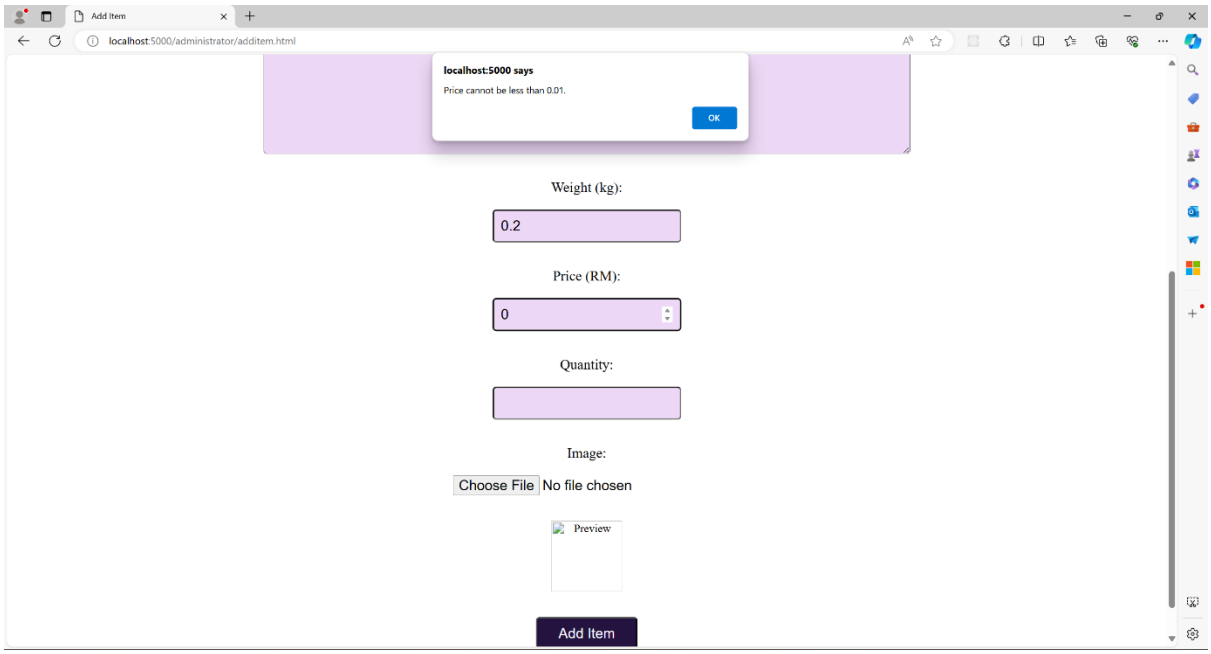
Appendix A7



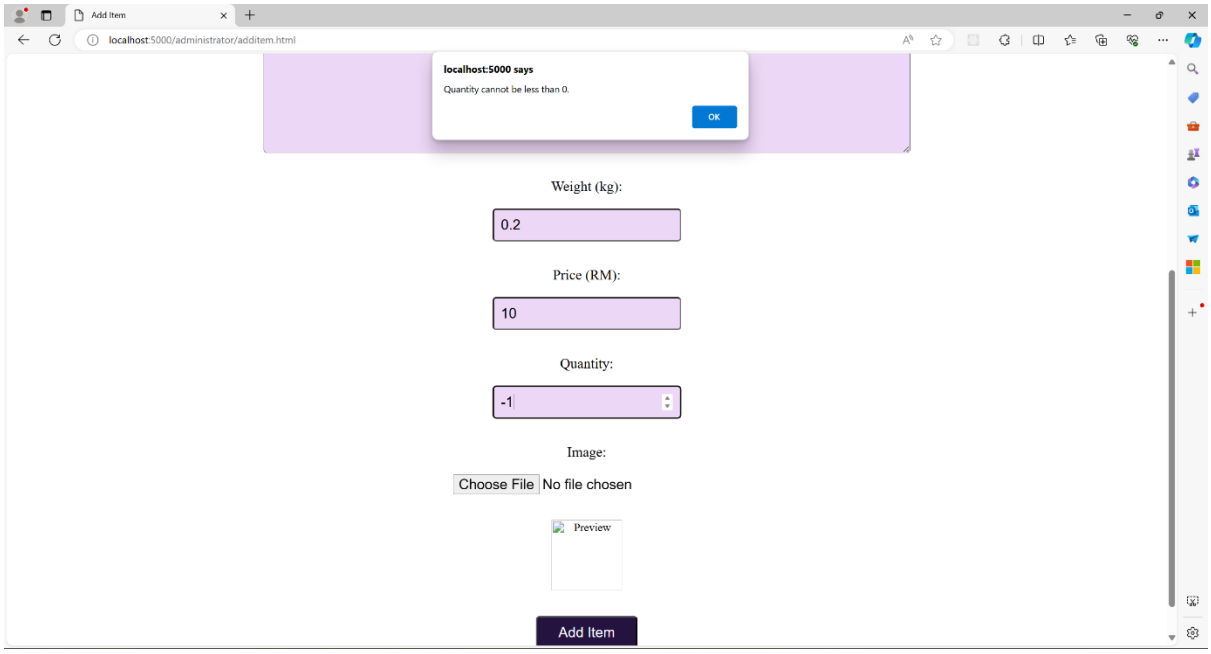
Appendix A8



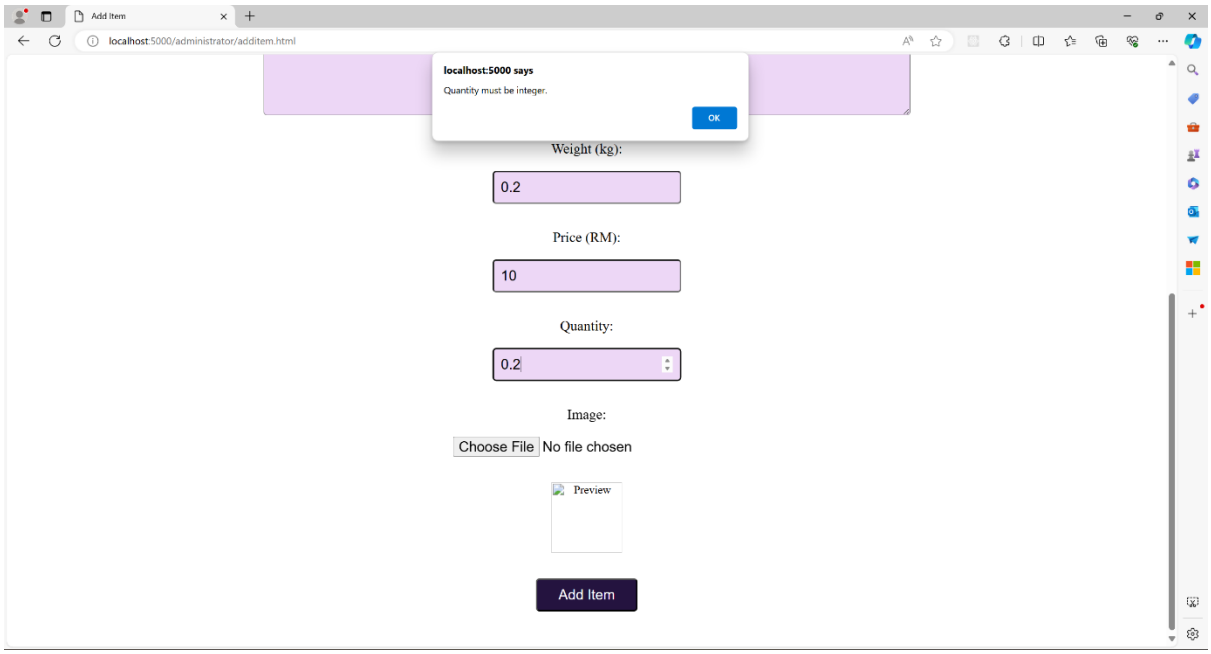
Appendix A9



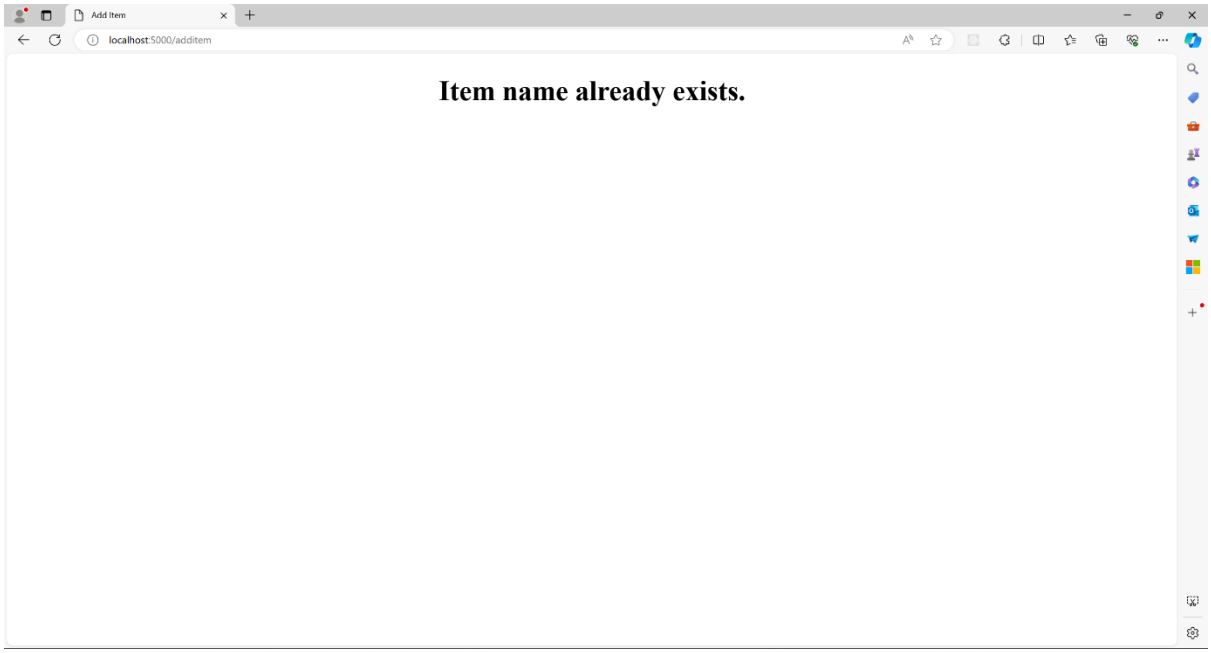
Appendix A10



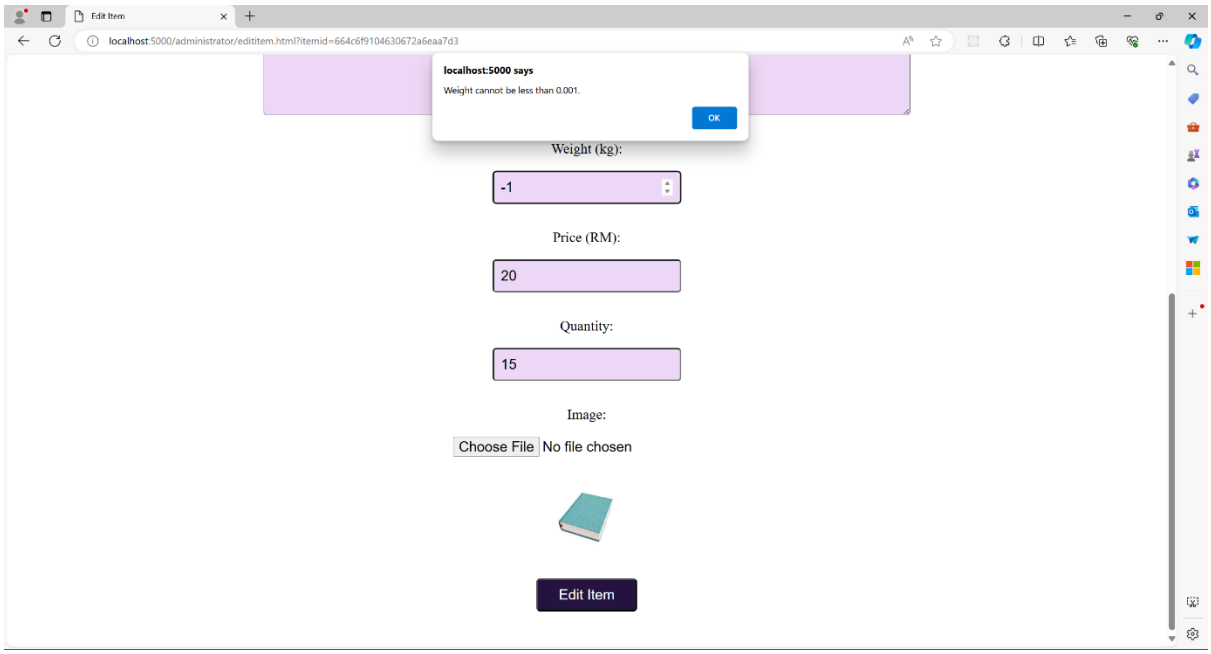
Appendix A11



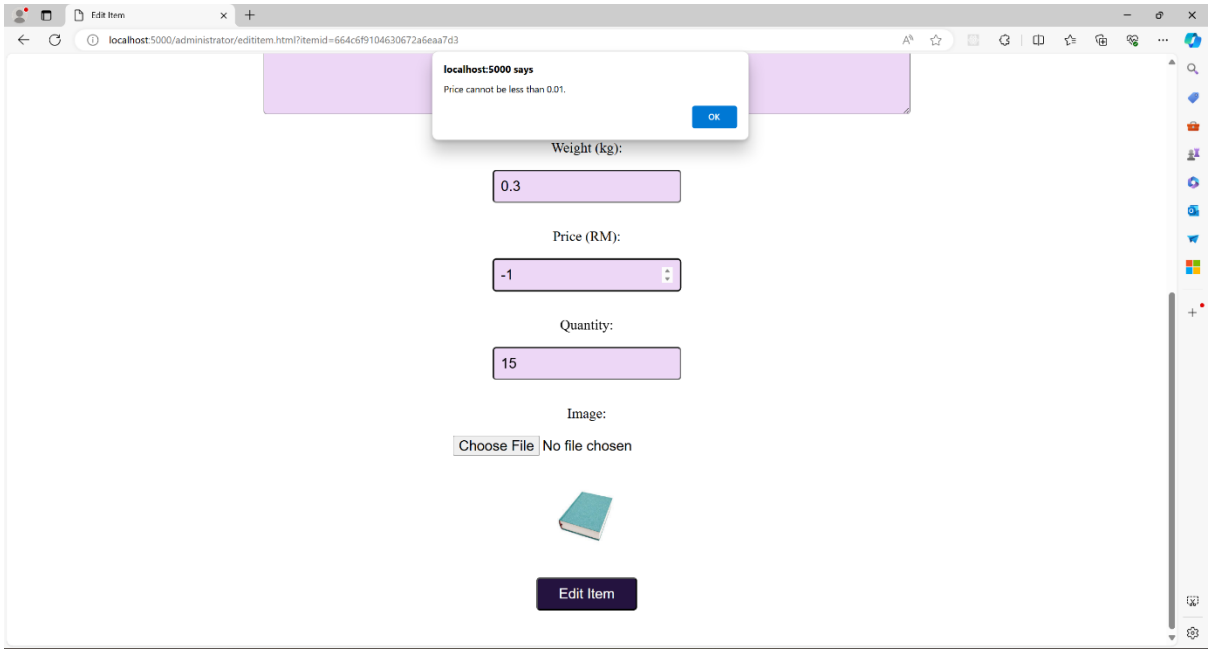
Appendix A12



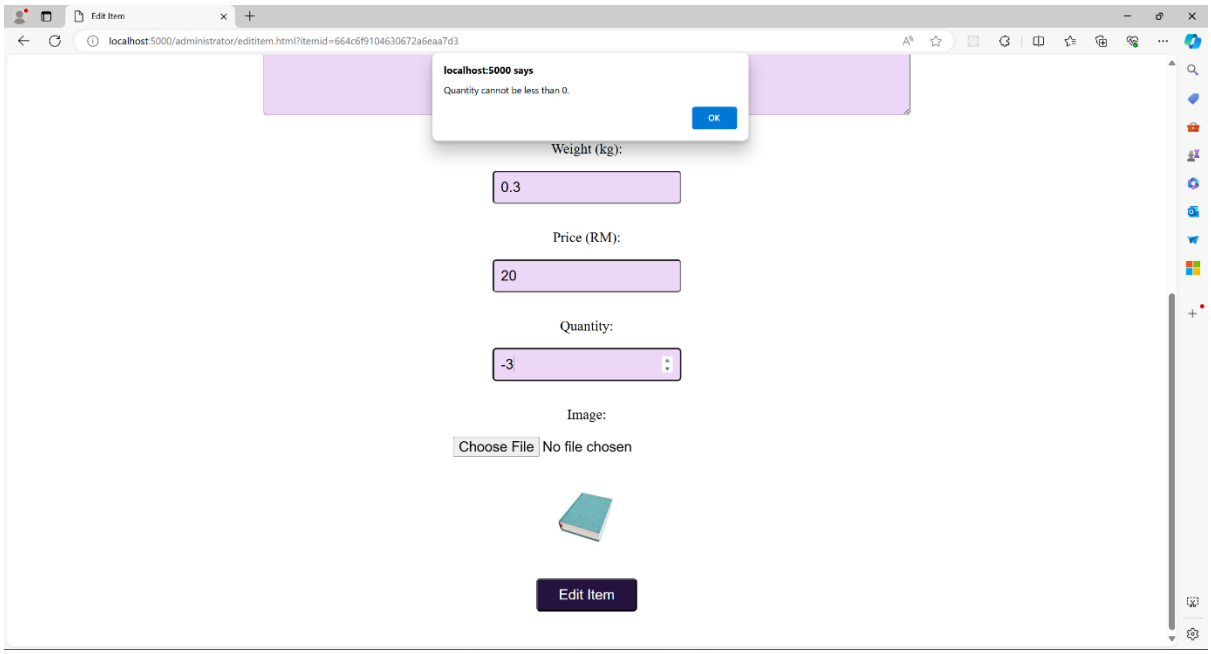
Appendix A13



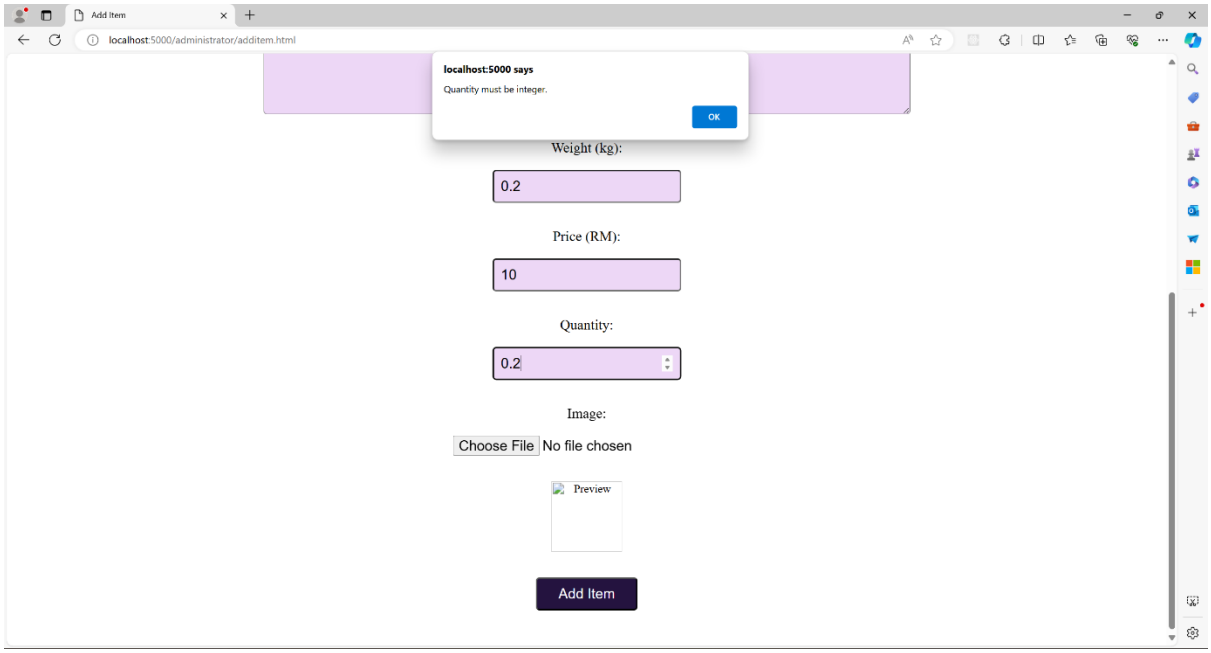
Appendix A14



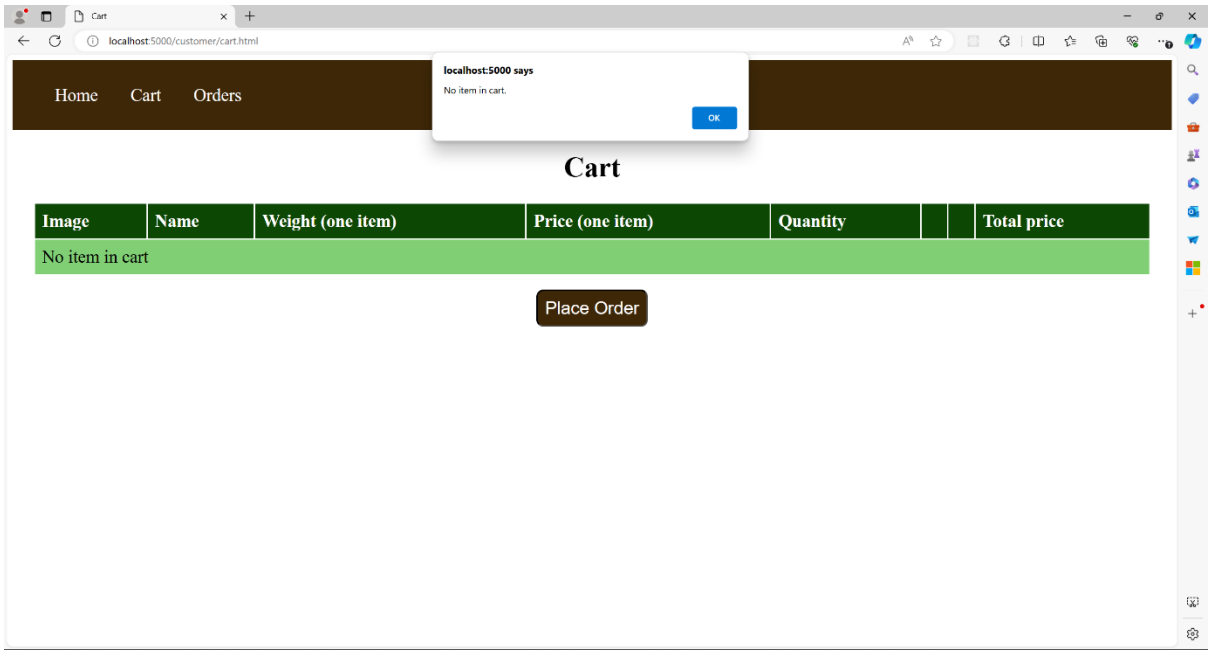
Appendix A15



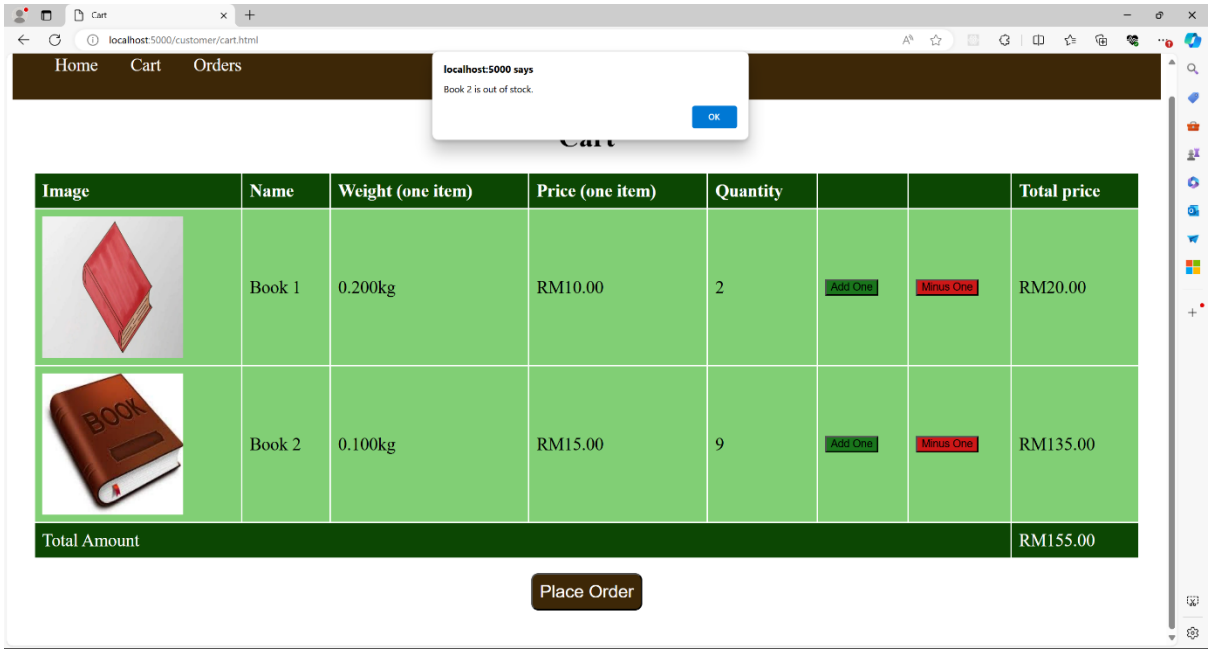
Appendix A16



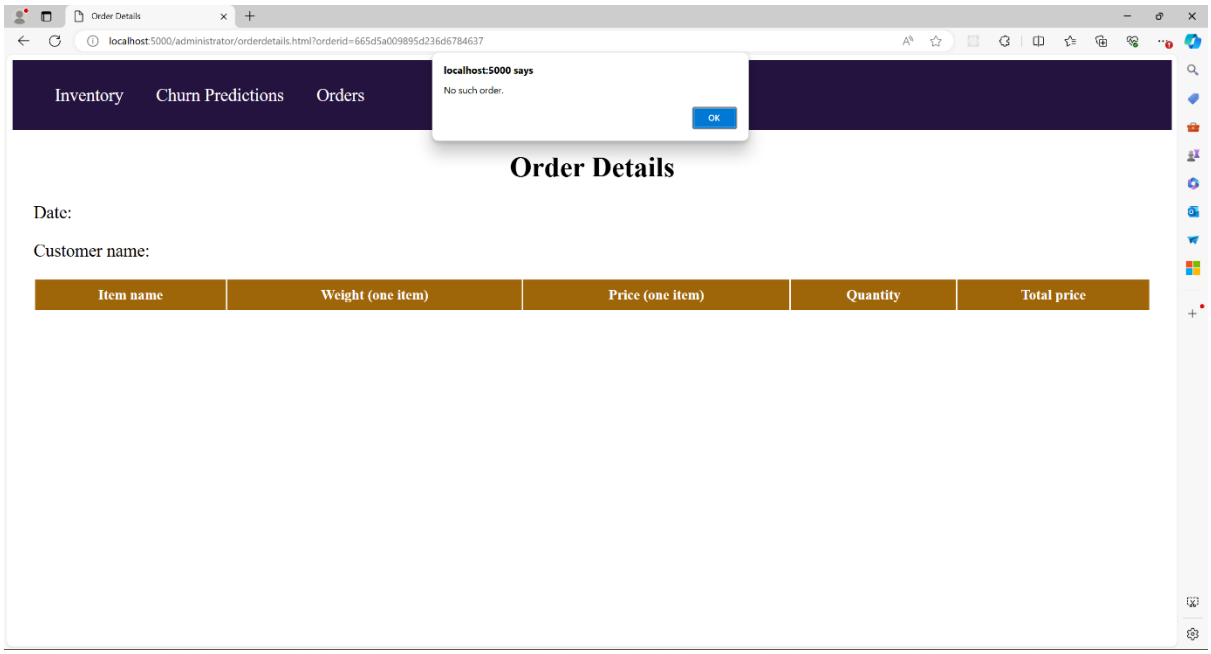
Appendix A17



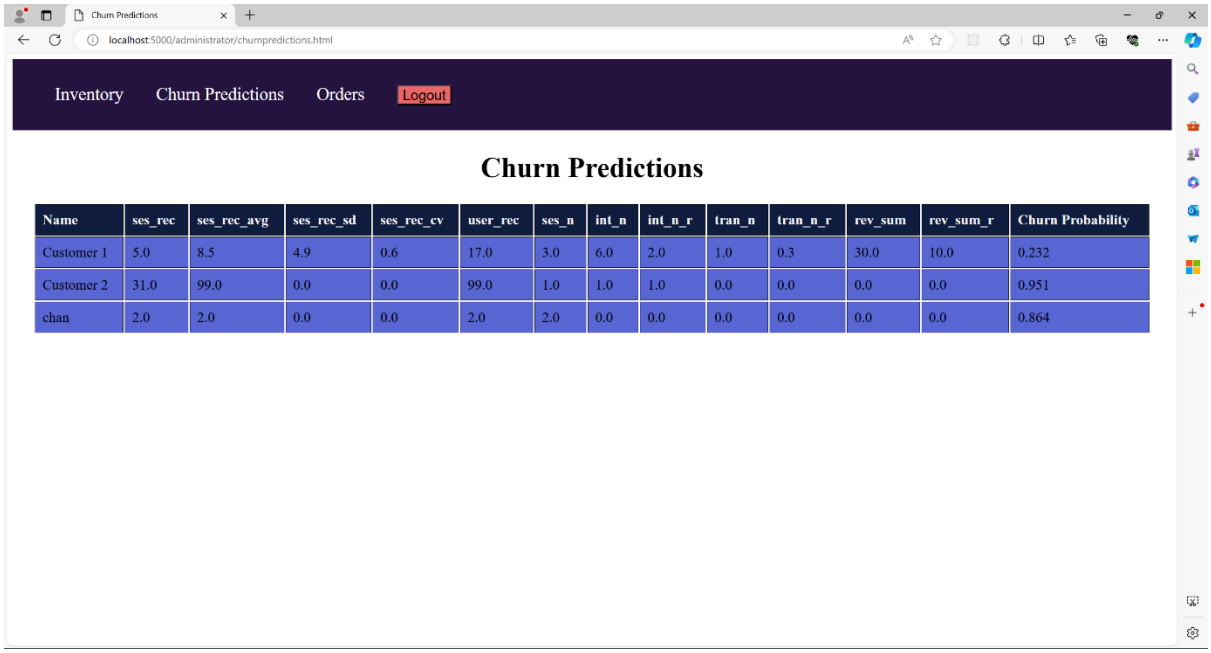
Appendix A18



Appendix A19

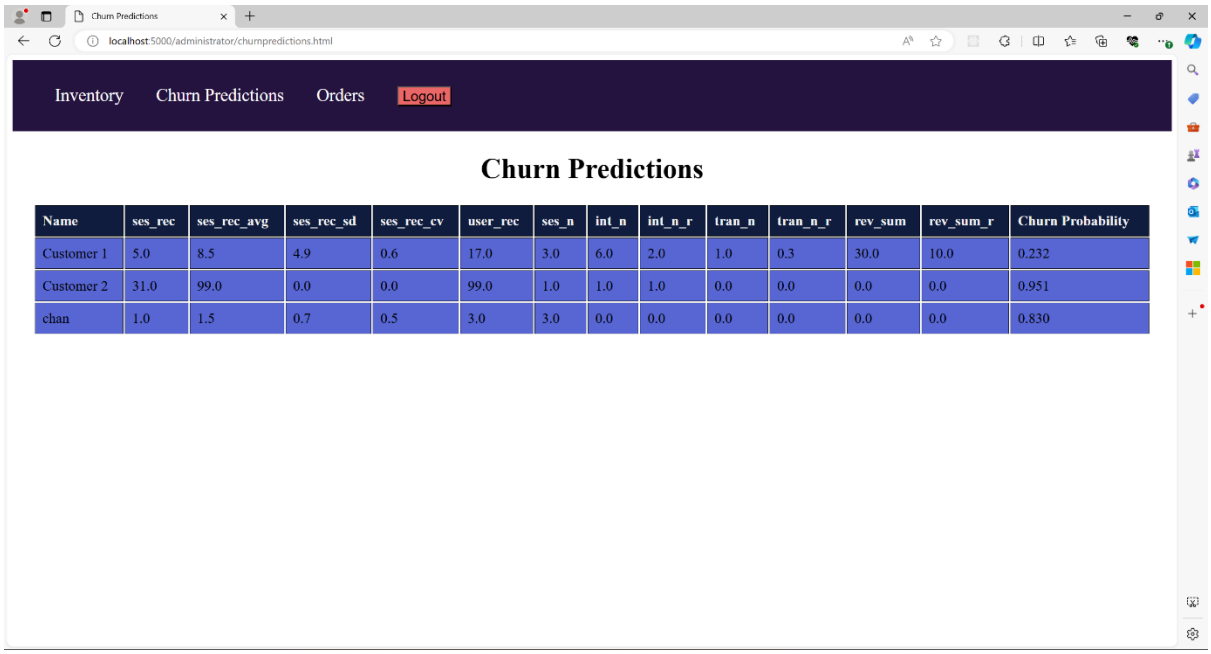


Appendix A20



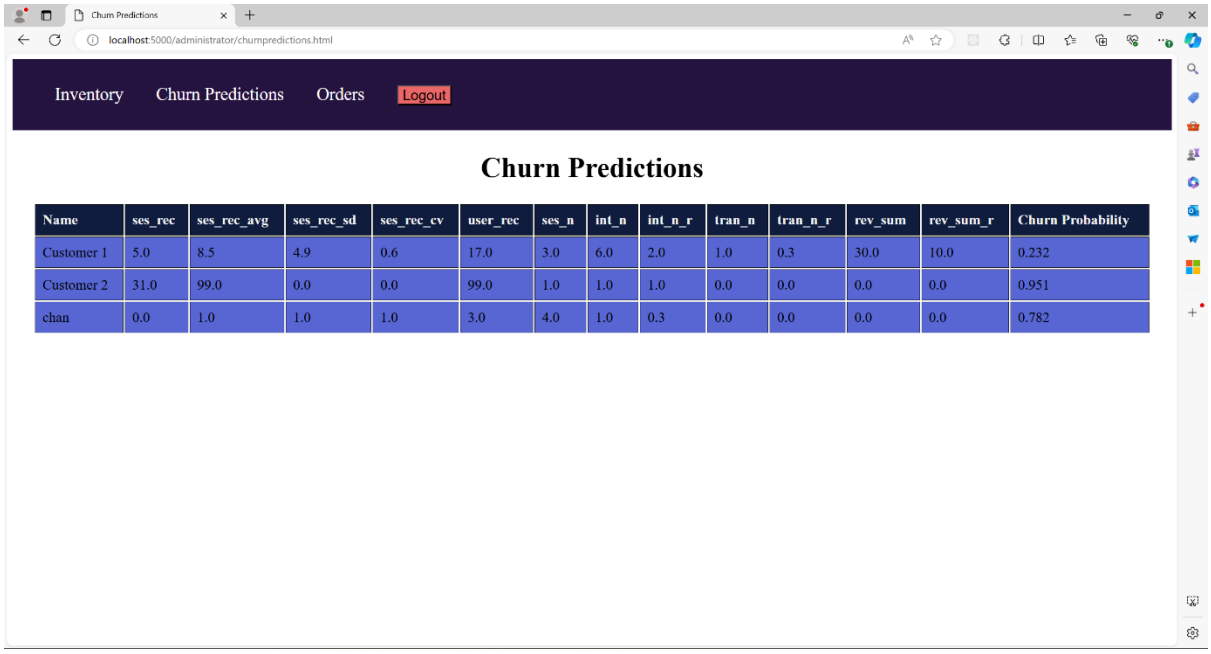
Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	2.0	2.0	0.0	0.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.864

Appendix A21



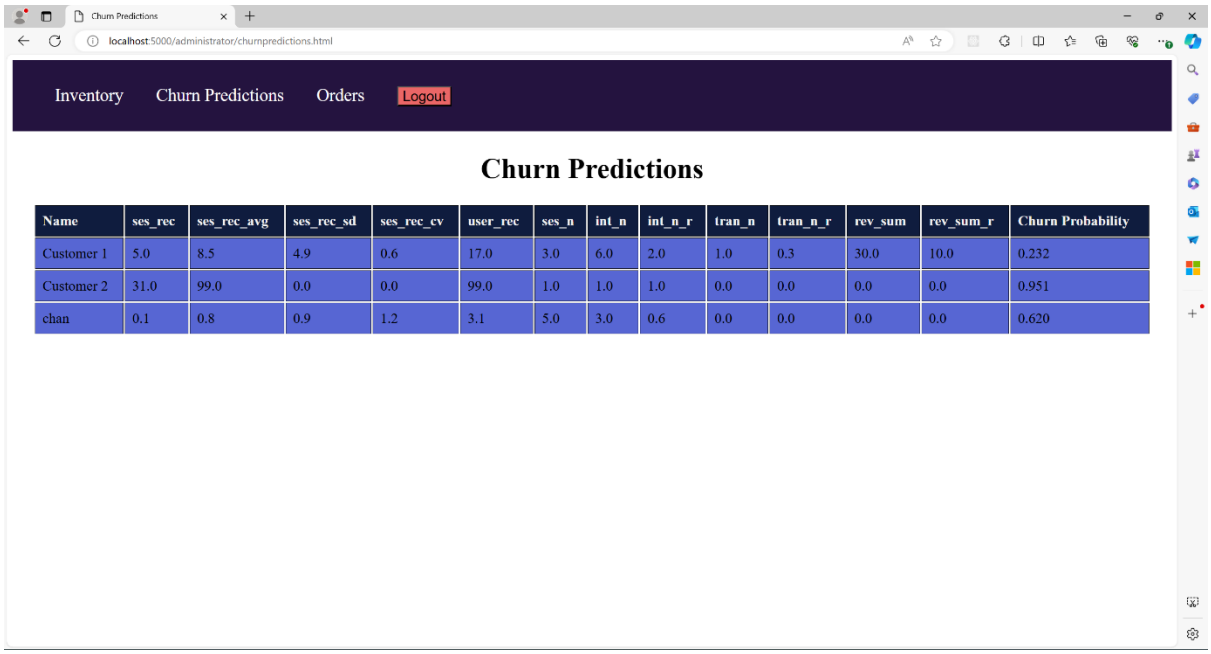
Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	1.0	1.5	0.7	0.5	3.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.830

Appendix A22



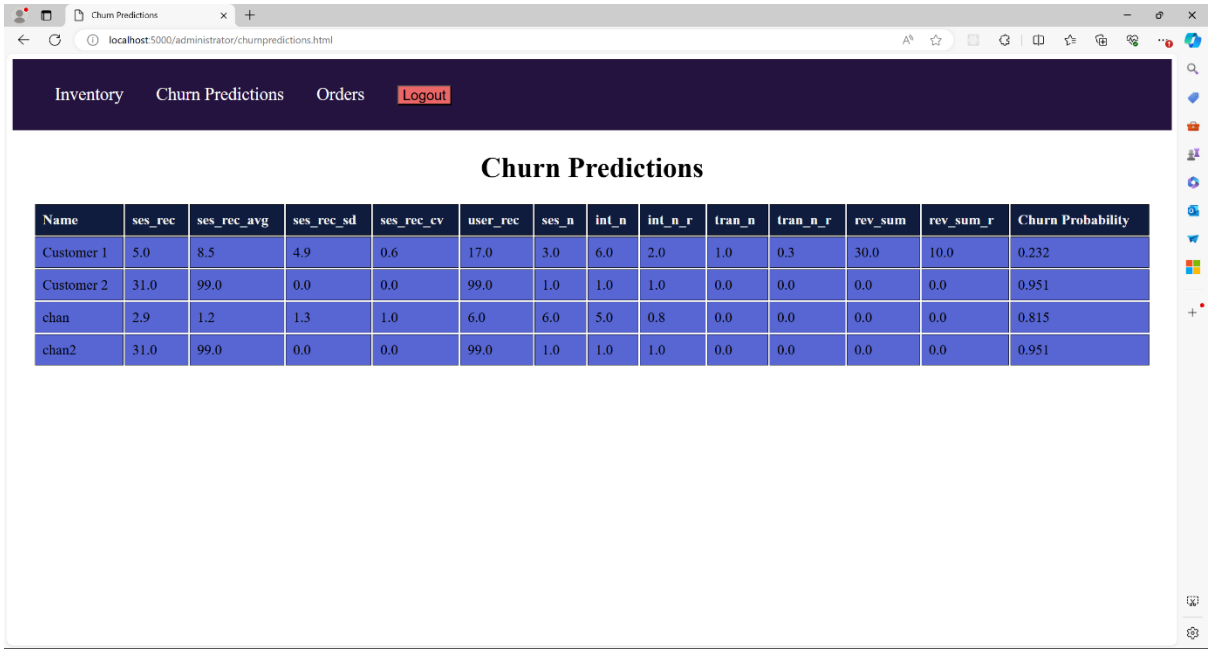
Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	0.0	1.0	1.0	1.0	3.0	4.0	1.0	0.3	0.0	0.0	0.0	0.0	0.782

Appendix A23



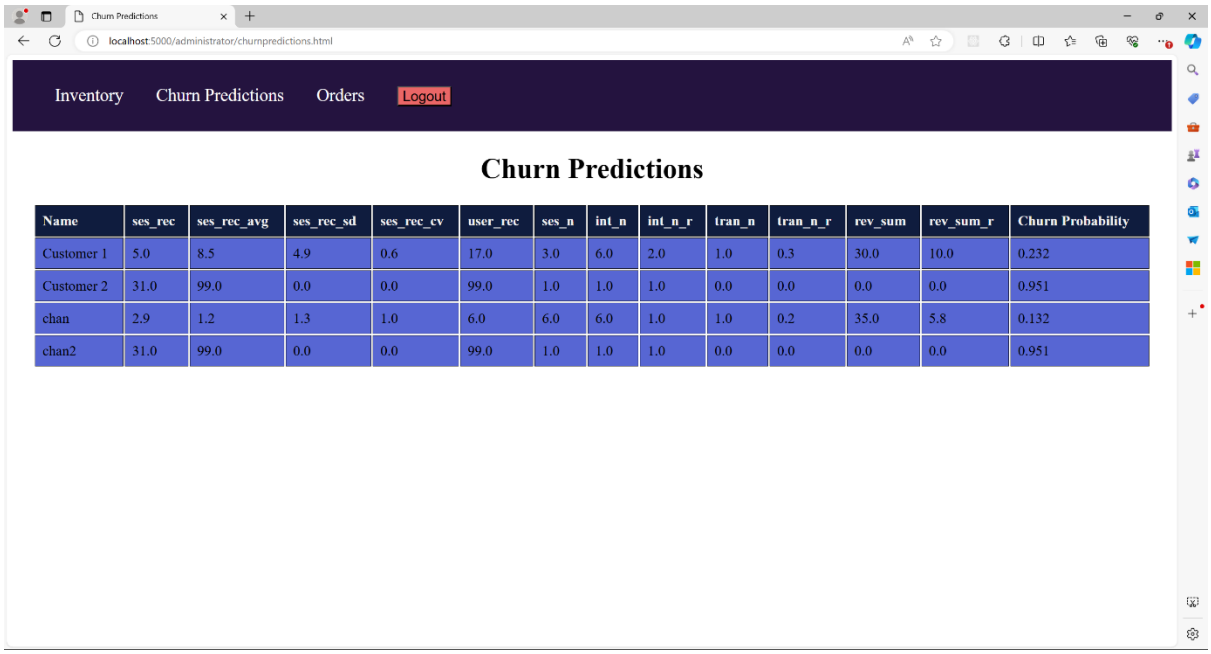
Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	0.1	0.8	0.9	1.2	3.1	5.0	3.0	0.6	0.0	0.0	0.0	0.0	0.620

Appendix A24



Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	2.9	1.2	1.3	1.0	6.0	6.0	5.0	0.8	0.0	0.0	0.0	0.0	0.815
chan2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951

Appendix A25



Name	ses_rec	ses_rec_avg	ses_rec_sd	ses_rec_cv	user_rec	ses_n	int_n	int_n_r	tran_n	tran_n_r	rev_sum	rev_sum_r	Churn Probability
Customer 1	5.0	8.5	4.9	0.6	17.0	3.0	6.0	2.0	1.0	0.3	30.0	10.0	0.232
Customer 2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951
chan	2.9	1.2	1.3	1.0	6.0	6.0	6.0	1.0	1.0	0.2	35.0	5.8	0.132
chan2	31.0	99.0	0.0	0.0	99.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.951

Appendix A26

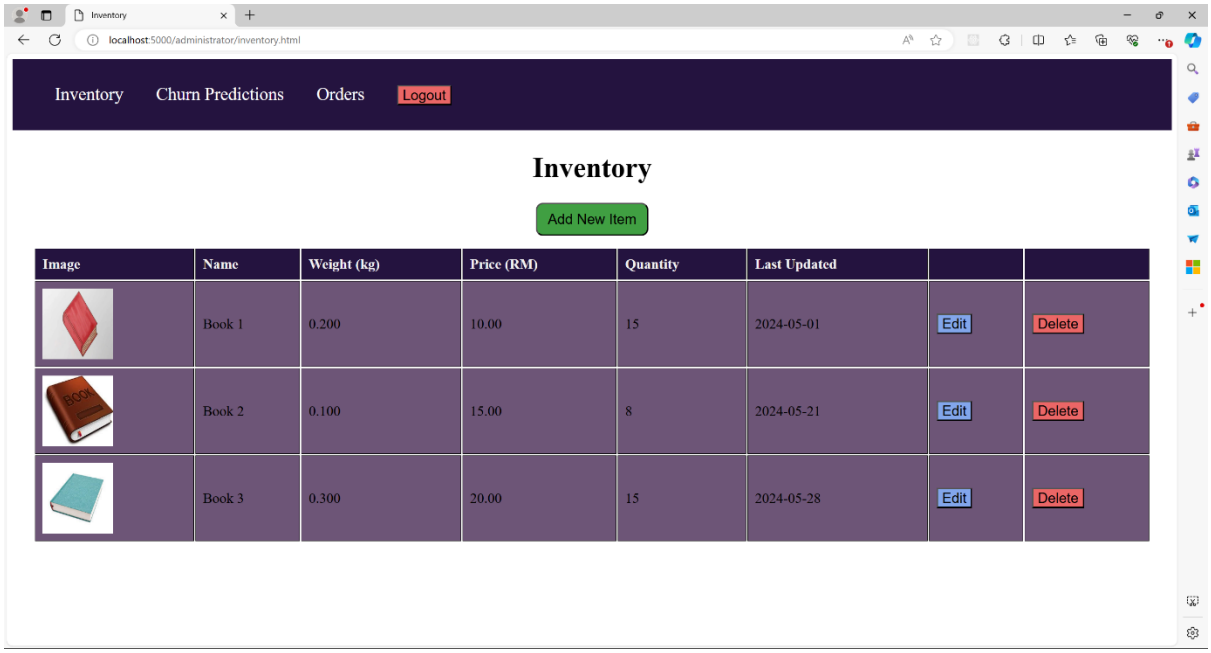
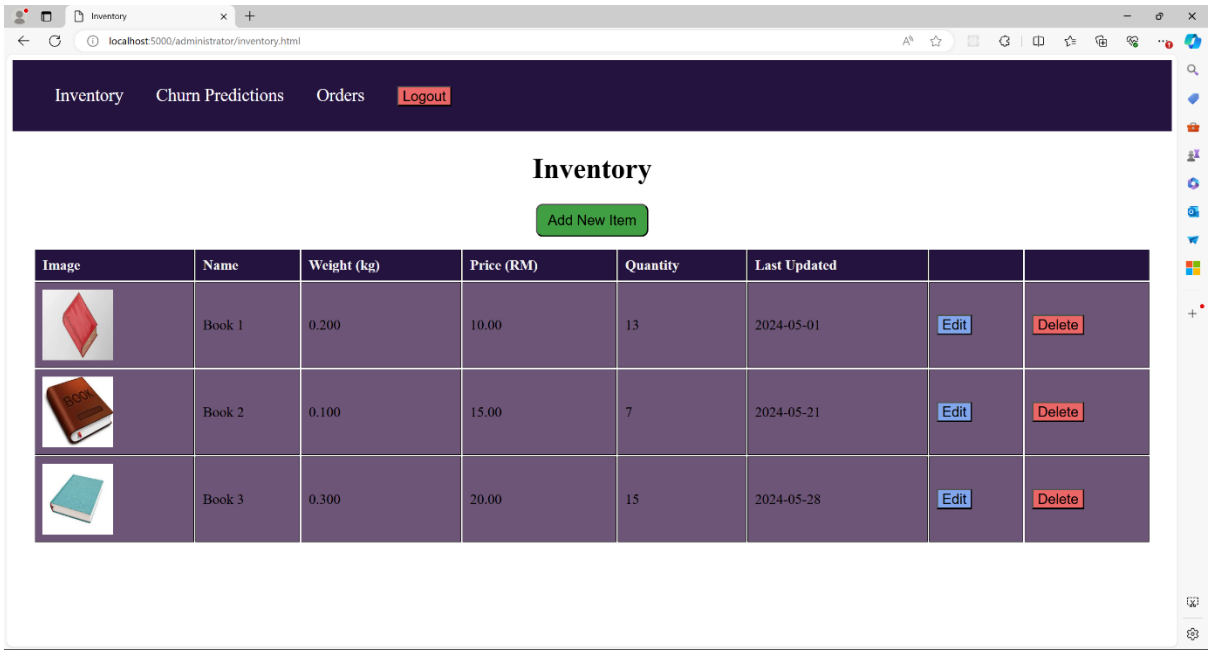


Image	Name	Weight (kg)	Price (RM)	Quantity	Last Updated		
	Book 1	0.200	10.00	15	2024-05-01	Edit	Delete
	Book 2	0.100	15.00	8	2024-05-21	Edit	Delete
	Book 3	0.300	20.00	15	2024-05-28	Edit	Delete

Appendix A27

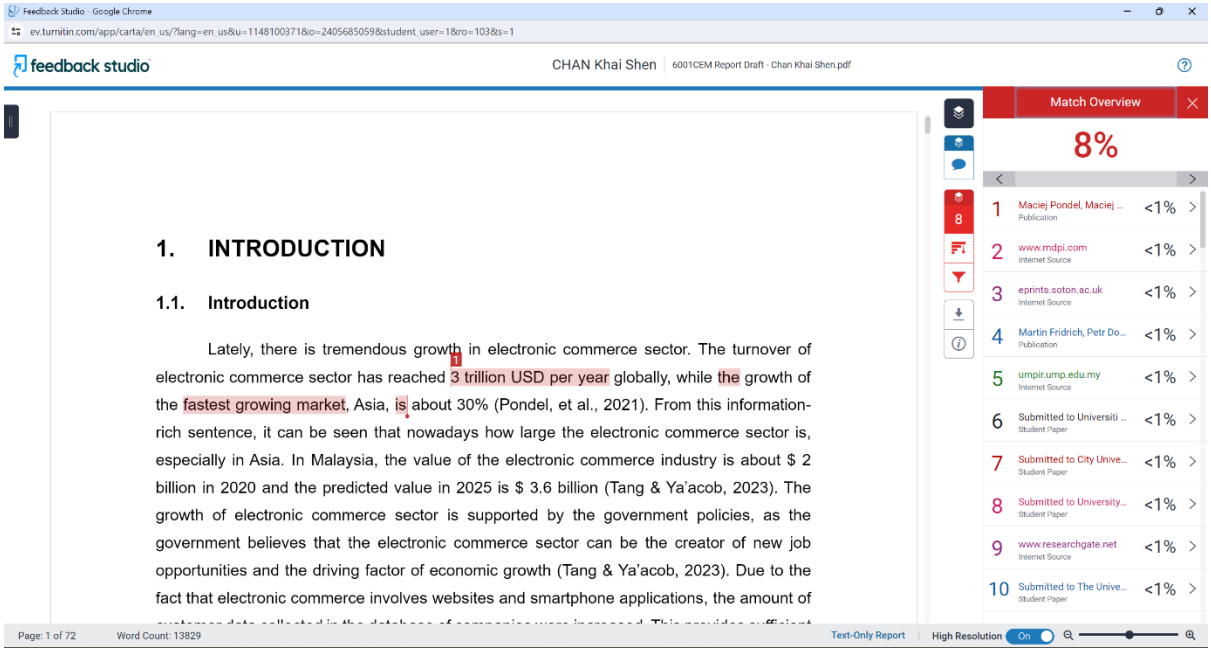


Appendix B

Training dataset source: https://www.kaggle.com/datasets/fridrichmrtn/user-churn-dataset

GitHub repository: https://github.com/ChanKhaiShen/e-commerce-book-shop-back-end

Appendix C



Feedback Studio - Google Chrome

ev.turnitin.com/app/carta/en_us/?lang=en_us&u=1148100371&o=2405685059&student_user=1&ro=103&s=1

feedback studio | CHAN Khai Shen | 6001CEM Report Draft - Chan Khai Shen.pdf

1. INTRODUCTION

1.1. Introduction

Lately, there is tremendous growth in electronic commerce sector. The turnover of electronic commerce sector has reached 3 trillion USD per year globally, while the growth of the fastest growing market, Asia, is about 30% (Pondel, et al., 2021). From this information-rich sentence, it can be seen that nowadays how large the electronic commerce sector is, especially in Asia. In Malaysia, the value of the electronic commerce industry is about $ 2 billion in 2020 and the predicted value in 2025 is $ 3.6 billion (Tang & Ya'acob, 2023). The growth of electronic commerce sector is supported by the government policies, as the government believes that the electronic commerce sector can be the creator of new job opportunities and the driving factor of economic growth (Tang & Ya'acob, 2023). Due to the fact that electronic commerce involves websites and smartphone applications, the amount of

Match Overview

8%

1	Maciej Pondel, Maciej ... Publication	<1%
2	www.mdpi.com Internet Source	<1%
3	eprints.soton.ac.uk Internet Source	<1%
4	Martin Fridrich, Petr Do... Publication	<1%
5	umpir.ump.edu.my Internet Source	<1%
6	Submitted to Universiti ... Student Paper	<1%
7	Submitted to City Unive... Student Paper	<1%
8	Submitted to University... Student Paper	<1%
9	www.researchgate.net Internet Source	<1%
10	Submitted to The Unive... Student Paper	<1%

Page: 1 of 72 Word Count: 13829 Text-Only Report High Resolution On