# CHAPTER ONE INTRODUCTION

## **Background of Study**

Internet has contacted our lives from such a large number of points. One of the critical zones of day by day lives where it is profoundly influenced is matter of fact. Online business is where Information Technology (IT) and business exercises are merging into one another.

In this virtual time, a relevant online business procedure is the way to long haul sustenance for any business. Today, pretty much each and every errand is done through on the web and shopping is no exemption. Having a web-based business site is no more a way to accomplish upper hand. It is a vital basic for business, while customers are now flourishing on the web (SAHA, 2018).

Internet based business is quick making progress as an acknowledged and utilized business worldview. More entrepreneurs executing web-based business usefulness and online exchange framework into their sites. So, it isn't difficult to state that into couple of decades web will be the typical for shopping any sort of item or administrations (SAHA, 2018).

webstore shopping has undergone a revolution in every retail sector, with noticeable changes in customer purchasing behavior driven by high-income growth, changing lifestyles, and cost-effective and efficient online and mobile technology. Consumers' need for ease has grown as they devote less time to shopping and more to other activities, and their attention has frequently been diverted to virtual shopping as an alternative medium. Thus, the rapid expansion of the internet is altering the way people browse for and purchase items and services, and it has quickly become a worldwide phenomenon. As a result, one of the primary motivators behind customer preferences for online purchasing has been convenience. Although still in its early stages, webstores is gaining pace and becoming increasingly incorporated into the everyday lives of customers in various areas. Despite the development and significance of online webstores, little is known about how individuals buy goods online. Khan et al. (2020)

No firm can afford to ignore the functionality and new methods of conducting business that e-commerce provides. This application made advantage of business-to-consumer marketing (B2C). Online e-Commerce websites such as eBay provide a valuable platform for users to purchase and sell things. As a result, I decided to develop a beneficial webstore platform that can be used by anyone and location bound free

## **Statement of the Problem**

The factors that influence the course of development of this work is the problem that customer encounter when they want to purchase products, customer have to practice in-store shopping, and some even go around with a lot of cash to purchase product which is very dangerous, all these problem motivated me to embark on this project work to eradicate the above problems mentioned. Thus, it is necessary to introduce an information system that would be used for the recording of events regarding and complications related to webstore shopping.

## **Aim and Objectives of the Study**

Is to develop a modern responsive ecommerce web application using HTM, CSS, Js & Node.js.

The objectives are:

1. An engaging and easy to use UI with good UX using Bootstrap.
2. The logic will be handled and the data will be stored using Node.js.
3. Evaluating and verifying the app.

## **Scope of the Project**

This will be a single page shop ecommerce app allowing user to upload multiple products with their corresponding inventory and price.

## **Limitation of the Study**

This will be only on a local host as the web application is design for an undergraduate program.

## **Significance of the Study**

The platform or web app stands to bring a responsive and user experience interface design for user to facilitate easy use of the platform.

## **Project Organization**

This project is made up of five chapters which will be organized in the following order: Chapter one summarizes the introductory study on E-commerce grocery business to customer application, including the background of study, statement of the problem, aim and objectives, significance of the study, scope of the study and limitation of the study. Chapter Two is an overview of related and relevant literature on the topic. Chapter Three describes the research methodology used in this project. That is, specific methods which were used in order to achieve the objectives of the system. Chapter Four displays the data analysis and gives concise details of how the system is to be implementedwhile Chapter Five contains the summary and conclusion.

**1.9 Definition of Terms**

1. **E-commerce**: E-commerce, or electronic commerce, refers to the buying and selling of goods or services over the Internet.
2. **Shopping Cart:** asoftware application that allows an online shopper to accumulate a list of items for purchase
3. **Merchant:** A merchant is a person or company that sells goods or services.
4. **B2C:** business-to-consumer, refers to a type of e-commerce transaction in which a business sells goods or services directly to a consumer.

# CHAPTER TWO LITERATURE REVIEW



## **Introduction**

A look at what the programming language and database used the project are and a review of related literature.

## **Programming Languages Used**

### **Angular**

Angular is an open-source, [JavaScript](https://www.simplilearn.com/tutorials/javascript-tutorial/introduction-to-javascript) framework written in [TypeScript](https://www.simplilearn.com/tutorials/typescript-tutorial/typescript-interview-questions). Google maintains it, and its primary purpose is to develop single-page applications. As a framework, Angular has clear advantages while also providing a standard structure for developers to work with. It enables users to create large applications in a maintainable manner (Deshpande, 2022).

#### **Why Do You Need a Framework?**

Frameworks in general boost web development efficiency and performance by providing a consistent structure so that [developers](https://www.simplilearn.com/how-to-become-complete-web-development-professional-article) don’t have to keep rebuilding code from scratch. Frameworks are time savers that offer developers a host of extra features that can be added to software without requiring extra effort (Deshpande, 2022).

#### **Why Angular?**

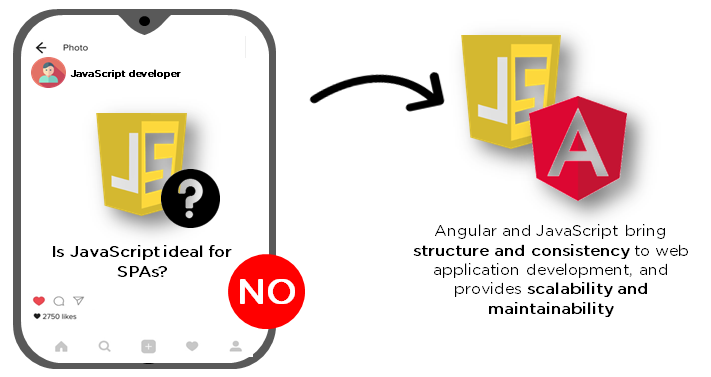


Figure 2.1: Angular (Deshpande, 2022)

[JavaScript](https://www.simplilearn.com/reasons-to-learn-javascript-article) is the most commonly used client-side scripting language. It is written into [HTML](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html) documents to enable interactions with web pages in many unique ways. As a relatively easy-to-learn language with pervasive support, it is well-suited to develop modern applications (Deshpande, 2022).

But is JavaScript ideal for developing single-page applications that require modularity, testability, and developer productivity? Perhaps not (Deshpande, 2022).

These days, we have a variety of frameworks and libraries designed to provide alternative solutions. With respect to [front-end web development,](https://www.simplilearn.com/how-to-become-a-front-end-developer-article) Angular addresses many, if not all, of the issues developers face when using JavaScript on its own (Deshpande, 2022).

#### **What are the Different Angular Versions:**

“Angular” is the catch-all term for the various framework versions out there. Angular was developed in 2009, and as a result, there have been many iterations (Deshpande, 2022).

First, there was the original Angular, called Angular 1 and eventually known as [AngularJS](https://www.simplilearn.com/angularjs-vs-angular-2-vs-angular-4-differences-article). Then came Angulars 2, 3, 4, 5, until finally, the current version, Angular 11, released on 11/11/2020. Each subsequent Angular version improves on its predecessor, fixing bugs, addressing issues, and accommodating increasing complexity of current platforms (Deshpande, 2022).

If you want to design apps better suited for mobile devices, and/or more complex apps, you had best to upgrade to its current version (Deshpande, 2022).

#### **Features of Angular**

##### **Document Object Model**

DOM (Document Object Model) treats an [XML](https://www.simplilearn.com/tutorials/programming-tutorial/what-is-xml) or HTML document as a tree structure in which each node represents a part of the document (Deshpande, 2022).

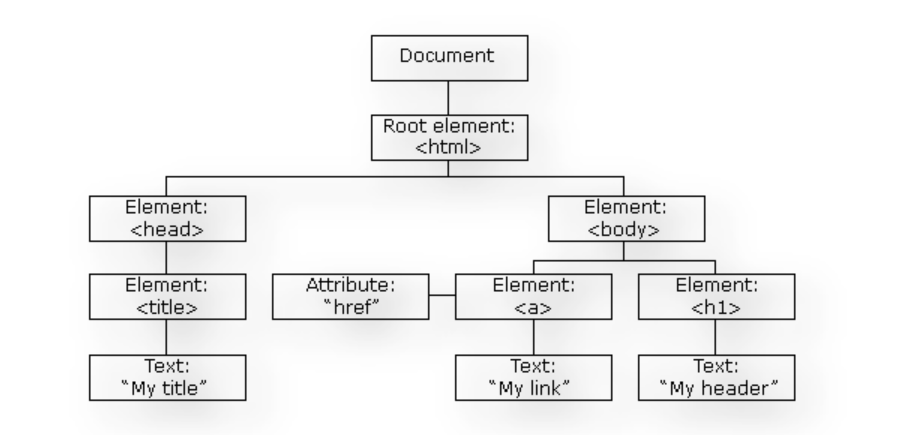


Figure 2.2: Document Object Model (Deshpande, 2022)

Angular uses regular DOM. Consider that ten updates are made on the same HTML page. Instead of updating the ones that were already updated, Angular will update the entire tree structure of [HTML tags](https://www.simplilearn.com/tutorials/html-tutorial/html-tags) (Deshpande, 2022).

##### **TypeScript**

TypeScript defines a set of types to JavaScript, which helps users write JavaScript code that is easier to understand. All of the TypeScript code compiles with JavaScript and can run smoothly on any platform. TypeScript is not compulsory for developing an Angular application. However, it is highly recommended as it offers better syntactic structure—while making the codebase easier to understand and maintain (Deshpande, 2022).



Figure 2.3: Typescript (Deshpande, 2022)

You can install TypeScript as an NPM package with the following command: npm install -g typescript (Deshpande, 2022).

##### **Data Binding**

[Data binding](https://www.simplilearn.com/tutorials/angular-tutorial/angular-data-binding) is a process that enables users to manipulate web page elements through a web browser. It employs dynamic HTML and does not require complex scripting or [programming](https://www.simplilearn.com/how-to-learn-programming-article). Data binding is used in web pages that include interactive components, such as calculators, tutorials, forums, and games. It also enables a better incremental display of a web page when pages contain a large amount of data (Deshpande, 2022).

Angular uses the two-way binding. The model state reflects any changes made in the corresponding UI elements. Conversely, the UI state reflects any changes in the model state. This feature enables the framework to connect the DOM to the model data through the controller (Deshpande, 2022).

##### **Testing**



Figure 2.4: Jasmine (Deshpande, 2022)

Angular uses the [Jasmine testing framework](https://en.wikipedia.org/wiki/Jasmine_(JavaScript_testing_framework)). The Jasmine framework provides multiple functionalities to write different kinds of test cases. Karma is the task-runner for the tests that uses a configuration file to set the start-up, reporters, and testing framework (Deshpande, 2022).

Now that you’re familiar with Angular’s basic features, you need to understand its architecture if you want to work with Angular daily. You can also expand your Angular knowledge by taking the [Angular Certification Training Course](https://www.simplilearn.com/angular-certification-training-course?source=GhPreviewCTAText) and learning concepts such as TypeScript, Bootstrap Grid System, dependency injections, SPA, forms, pipes, promises, observables, and Angular class testing (Deshpande, 2022).

#### **Angular Architecture**

Angular is a full-fledged [model-view-controller (MVC) framework.](https://www.simplilearn.com/tutorials/dot-net-tutorial/mvc-architecture) It provides clear guidance on how the application should be structured and offers bi-directional data flow while providing real DOM (Deshpande, 2022).

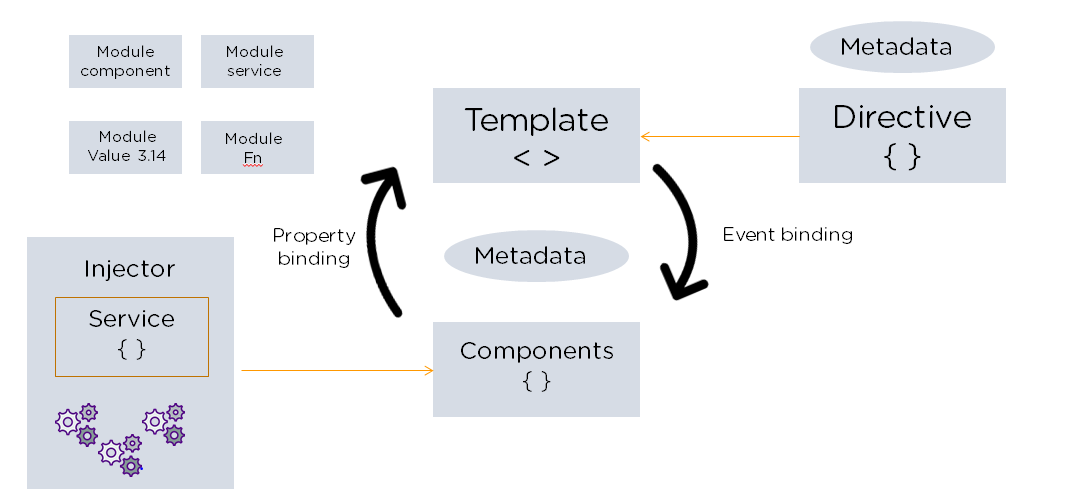


Figure 2.5: Angular Architecture (Deshpande, 2022)

### **Node.js**

Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser. Ryan Dahl developed it in 2009, and its latest iteration, version 15.14, was released in April 2021. Developers use Node.js to create server-side web applications, and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model (Sufiyan, 2022).

#### **Why Do We Use NodeJs?**

There are many reasons for which we prefer using NodeJs for the server side of our application, some of them are discussed in the following (Sufiyan, 2022):

1. NodeJs is built on Google Chrome’s V8 engine, and for this reason its execution time is very fast and it runs very quickly.
2. There are more than 50,000 bundles available in the Node Package Manager and for that reason developers can import any of the packages any time according to their needed functionality for which a lot of time is saved.
3. As NodeJs do not need to wait for an API to return data , so for building real time and data intensive web applications, it is very useful. It is totally asynchronous in nature that means it is totally non-blocking.
4. The loading time for an audio or video is reduced by NodeJs because there is better synchronization of the code between the client and server for having the same code base.
5. As NodeJs is open-source and it is nothing but a JavaScript framework , so for the developers who are already used to JavaScript, for them starting developing their projects with NodeJs is very easy.

#### Features of NodeJs

1. Asynchronous in Nature and Event driven: The servers made with the NodeJs never waits for the from an API. Without waiting for the data from the API, it directly moves to the next API. So all the APIs of NodeJS are totally non-blocking in nature. In order to receive and track all the responses of the previous API requests, it follows an event driven mechanism. Hence we can say that all the NodeJs API are non-blocking in nature (Sufiyan, 2022).
2. Single Threaded Architecture: With event looping, a single threaded architecture is followed by NodeJs and for this architecture makes NodeJs more scalable. In contrast to other servers, limited threads are created by them for processing the requests. Whereas for the event driven mechanism, the NodeJS servers reply in a non-blocking or an asynchronous manner and for this reason NodeJS becomes more scalable. If we compare NodeJs with other traditional servers like Apache HTTP servers, then we can say NodeJs handles a larger number of requests. A single threaded program is followed by NodeJS and this allows NodeJs to process a huge amount of requests (Sufiyan, 2022).
3. Scalable: Nowadays, scalable software is demanded by most of the companies. One of the most pressing concerns in Software Development is addressed by NodeJs and that is scalability. Concurrent requests can be handled very efficiently using NodeJs. A cluster module is used by NodeJs for managing the load balancing for all the active CPU cores. The most appealing feature of NodeJs is that it can partition the applications horizontally and this partition procedure is mainly achieved by it due to the use of child processes. Using this feature, the distinct app versions are provided to the different target audiences and also for customization it allows them for catering to the client preferences (Sufiyan, 2022).
4. Quick Execution time for code: V8 JavaScript runtime motor is used by NodeJs and this is also used by Google chrome. A wrapper is provided for the JavaScript by the hub and for that reason the runtime motor becomes faster and for this reason inside NodeJs, the preposition process of the requests also become faster (Sufiyan, 2022).
5. Compatibility on the cross platforms: Different types of systems like Windows, UNIX, LINUX, MacOS and other mobile devices can use NodeJs. For generating a self-sufficient execution, it can be paired with any appropriate package (Sufiyan, 2022).
6. Uses JavaScript: From an engineer's perspective, it is a very important aspect of NodeJs that this framework uses JavaScript Most of the developers are familiar with JavaScript, so for them it becomes very easier to grab NodeJs (Sufiyan, 2022).
7. Fast Data Streaming: The processing time of the data that have been transmitted to different streams takes a long time. Whereas for processing the data, NodeJs takes a very short amount of time and it does it at a very fast rate. NodeJs saves a lot of time because the files are processed and uploaded simultaneously by NodeJs. So as a result, the overall speed of data and video streaming is improved by NodeJs (Sufiyan, 2022).
8. No Buffering : The data is never buffered in NodeJs application (Sufiyan, 2022).

#### **Node.js Architecture**

Now that we established what is Node, let’s dig into its architecture. Node.js operates on a single-thread, allowing it to handle thousands of simultaneous event loops. Here’s a diagram, provided by Sinform.com, that best illustrates Node.js architecture (Sufiyan, 2022).

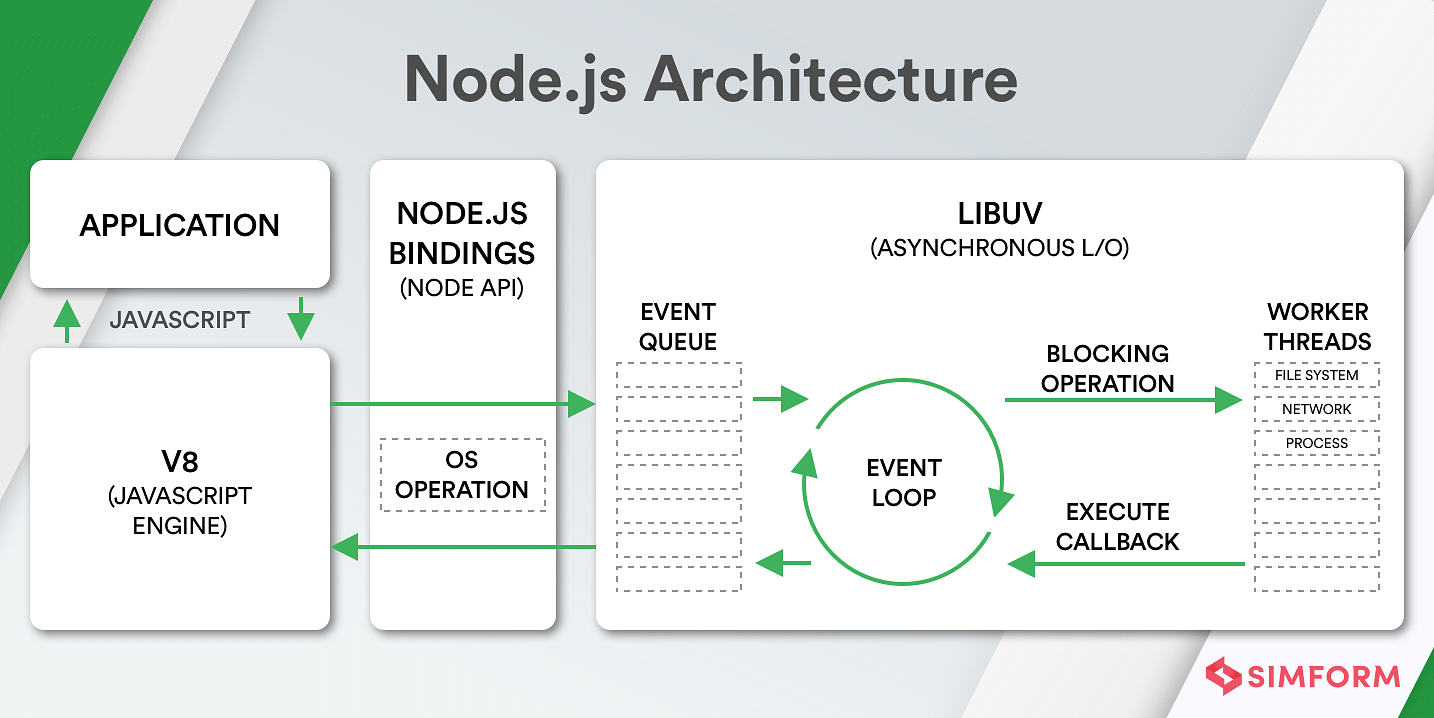


Figure 2.6: Architecture of Node.js (Sufiyan, 2022)

#### **Parts of Node.js**

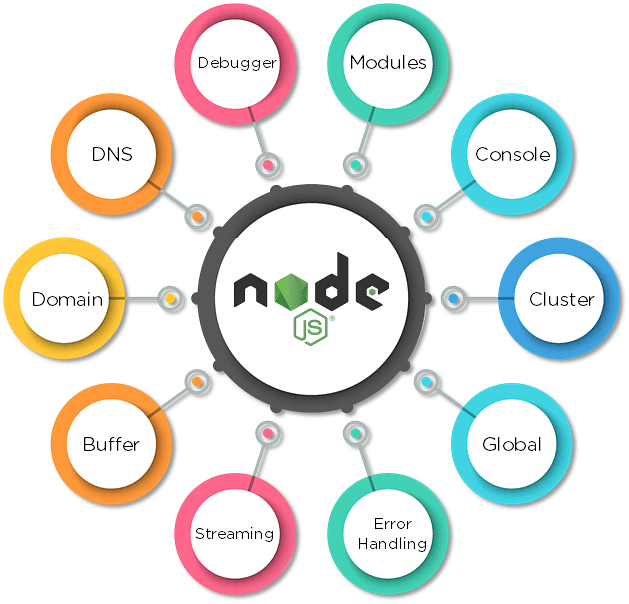


Figure 2.7: Parts of Node.js (Sufiyan, 2022)

## **Related Literature**

In this secion, literature from different sources (books, research paper and internet websites) on the ecommerce history, business model and comparative study has been discussed for better application development process. History of ecommerce says that web based shopping just wound up conceivable when the web was opened to general society in 1991. Amazon.com was one of the principal ecommerce destinations in the US to begin offering items on the web and a large number of organizations have pursued since. The tremendous fame of the web as of late has been powered generally by the possibility of performing business on the web. With the quick worldwide development in electronic trade, organizations are endeavoring to pick up an upper hand by utilizing internet business to interface with clients.

### **Definition of E-commerce**

There are numerous definitions for E-commerce which share a ton for all intents and purpose. By the by E-business has been characterized by researchers and specialists in various ways. Web based business alludes principally to the purchasing and offering exercises over the Internet, which incorporates exchanges, for example, setting orders, making installments, and following conveyance of requests on the Internet (Rodgers, 2002). The web based business is characterized as purchasing and offering of item administration or data through PC organizes mostly the web (Wen et al., 2001, as cited by SAHA, 2018) and individuals instantly consider shopper retail buys from organizations, for example, Amazon (Chaffey et al, 2006, as cited by SAHA, 2018). Be that as it may, internet business alludes to both monetary and educational electronically intervened exchanges between an association and any outsider it manages (Chaffey, 2006, as cited by SAHA, 2018). Further, Singh (2001, as cited by SAHA, 2018) upheld the definition by alluding web based business to an online administration offered to client to help their shopping knowledge over the web.

Online business is exponentially expanding the accessibility of data, giving clients access to more information, of preferable quality and quicker over previously. Mechanical headway drove an unstable improvement in electronic business, the reasons for that are the web end the World Wide Web (WWW), which are making electronic trade considerably more available. Web internet business incorporates electronic exchanging of physical merchandise and of intangibles, for example, data. This incorporates all the exchanging steps, for example, web based promoting, requesting, installment and support for conveyance (Trimmers, 2000, as cited by SAHA, 2018).

Online business is utilized wherever in regular daily existence. Presently a days it is used for everything from Visa approval, travel reservation over a system, wire support exchanges over the world, purpose of offer (POS) exchanges in retailing, electronic managing an account. It helps in create interest for the items and benefits and enhances arrange the executives, installment and other help capacities (Awad, 2004, as cited by SAHA, 2018). Thus, online business diminishes the expense of making, preparing, disseminating and recovering paper based data. Further encourages the advantages which incorporate enhanced picture, enhanced client benefit, streamlined procedures, packed cycle and conveyance time, disposing of printed material and expanded adaptability (Turban et al, 1999, as cited by SAHA, 2018).

Additionally, online business isn't just about purchasing and offering as it is likewise about electronically imparting, teaming up and finding data (Turban et al, 2004, as cited by SAHA, 2018). It can include the Internet, groupware programs, open email systems (Adams, 1994, as cited by SAHA, 2018), innovations, for example, electronic information trade (EDI) and electronic supports exchange (EFT) and all the more as of late, administrations related with cell phones and computerized TV (Voss, 1999, as cited by SAHA, 2018). E – Business has turned out to be exceptionally well known due to the advantages and accommodation it brings along as it is not any more an option, it is a goal (Wen et al, 2001, as cited by SAHA, 2018). E-trade is by and large dependent on financial exchange and e-administrations are the piece of electronic trade, as cited by (SAHA, 2018).

### **History of Ecommerce**

The historical backdrop of web based business begun 40 years prior and, right up 'til today, keeps on developing with new technological advancements, innovations, and a huge number of organizations entering the online market every year (SAHA, 2018).

Electronic Data Interchanges and mail order shopping during the 1970s made ready for the present day internet business store. The historical backdrop of online business is nearly entwined with the historical backdrop of the web. Web based shopping just ended up conceivable when the web was opened to the general population in 1991 (SAHA, 2018).

Web based shopping was designed and spearheaded in 1979 by Michael Aldrich in the United Kingdom. He associated an adjusted local TV by means of a phone line to an ongoing multi-client exchange handling PC (SAHA, 2018). The framework was promoted starting in 1980 and offered fundamentally business-to-business frameworks that were sold in the UK, Ireland, and Spain. One the most punctual purchaser shopping encounters was Book Stacks Unlimited, an online book shop made by Charles M. Stack in 1992 (SAHA, 2018). Stack's store started as a dial-up release board two years previously Amazon was established by Jeff Bezos. In 1994, Book Stacks Unlimited moved to the Internet as Books.com and was eventually acquired by Barnes & Noble (SAHA, 2018).

The main online exchange was, by a few reports, marijuana sold by Stanford students to MIT students by means of the Arpanet account at their artificial intelligence lab in 1972 (SAHA, 2018). In any case, the primary web based shopping exchange on the Internet occurred exactly 22 years after the fact. With the feature "The Internet is Open", the August 12, 1994, issue of New York Times chronicled the deal between two companions of a Sting CD. The Times said, “The team of young cyberspace entrepreneurs celebrated what was apparently the first retail transaction on the Internet using a readily available version of powerful data encryption software designed to guarantee privacy (SAHA, 2018).”

History of online business is unimaginable without Amazon and eBay which were among the principal Internet organizations to permit electronic exchanges. Because of their originators we presently have an attractive web based business area and appreciate the purchasing and offering points of interest of the Internet (SAHA, 2018).

Amazon is one of the primary web based business organizations to build up a partner promoting system, and these days the organization gets about 40% of its deals from associates and outsider venders who rundown and offer products on the site. In 2008 Amazon entered into the film and is right now supporting the film "The Stolen Child" with twentieth Century Fox (SAHA, 2018).

As per the exploration led in 2008, the space Amazon.com pulled in around 615 million clients consistently. The most well-known component of the site is the audit framework,

i.e. the capacity for guests to present their audits and rate any item on a rating scale from one to five stars. Amazon.com is additionally outstanding for its clear and user-friendly advanced search facility which empowers visitors to search for keywords in the full content of numerous books in the database (SAHA, 2018).

### **Different types of E-commerce**

The E-trade exchanges occur between opposite sides of an exchange. The sides of this exchange can be a customer, a business, inner or the administration. At the point when the transaction is between the business and the consumer, it is called B2C. The transaction between businesses is called B2B E-commerce (SAHA, 2018). For the same reason the when E-commerce happens between business and government, it is called B2G and C2C for consumer to consumer (SAHA, 2018).

The following table is a quick demonstration of different types of existed E-commerce/E- businesses (SAHA, 2018).

1. Not easy to classify
2. Many real models fall into multiple categories

TABLE 2.1: EXAMPLE OF DIFFERENT TYPE OF THE E-COMMERCE

|  |  |  |
| --- | --- | --- |
| **B2B**  Business to Business   1. PayPal (and B2C, B2B2C) 2. Optize (and B2C) 3. Alibaba Group | **B2C**  Business to consumer   1. Amazon 2. FreshDirect 3. Zynga (and C2C) | **B2G**  Business to Government   1. E-Procurement |
| **C2B**  Consumer to Business   1. ZonZoo 2. Fotolia 3. Google Adsense | **C2C**  Consumer to Consumer   1. Prosper (P2P) 2. eBay 3. Facebook | **C2G**  Citizen to Government   1. Agencia Tributaria (Tax agency online) |
| **G2B**  Government to Business   1. E-Government 2. AEPM 3. Certificado Digital | **G2C**  Government to Citizen   1. E-Government 2. eDNI 3. USA.gov (also G2B) | **G2G**  Government to Government   1. Government Gateway 2. Schengen Information System |

### **E-commerce Usage in Developed vs. Developing Countries**

Online business can possibly enhance proficiency and efficiency in numerous regions and, accordingly, has gotten critical consideration in numerous nations. Be that as it may, there has been some uncertainty about the significance of web based business for developing countries (SAHA, 2018). The nonappearance of satisfactory fundamental infrastructural, financial and the absence of government national ICT systems have made a huge hindrance in the adoption and development of online business in developing countries. For web based business in developing countries, social issues also should be considered (SAHA, 2018).

Nham (2022) outcome of the project based on this thesis is an e-commerce webstore prototype. The web app will allow people to buy and sell items that are relevant to their needs. The website also provides useful tools to help the owners managing their e-commerce store in handling orders, payments, and logistic. By implementing the serverless backend, the maintenance has become easier and easier, especially for the developers compared to the traditional backend technologies.

Wu et al. (2020) paper combs the concept of C2B according to the relevant literature research, through the comparative analysis of the decision relationship between B2C and C2B, discusses the essential connotation of C2B electronic commerce, and analyzes the C2B conceptual misunderstanding existing in the application of e-commerce combined with practical cases. On this basis, the C2B application platform and the main bottlenecks that may exist in the application of electronic commerce are discussed. Finally, the solution strategy of C2B application bottleneck and the future trend of C2B application are discussed.

E-commerce is mainly divided into three types, but (Xu & Chen, 2020) mainly analyzes the application of big data technology in B2C e-commerce precision marketing pattern. Firstly, the notion and features of B2C e-commerce as well as the concept and characteristics of precision marketing mode are described. Secondly, this paper further analyzes some challenges faced by B2C e-commerce in the times of large data. Finally, taking China Amazon B2C electronic commerce as an example, this paper briefly analyzes the ratio of B2C e-commerce in China’s online retail B2C market share in the third quarter of 2017, which is 4.1%. This shows that B2C e-commerce is developing rapidly under the background of large data era.

Qwaider (2020) introduces the concepts of e-commerce application framework based on cloud computing, the development trend of cloud computing that adapts to the problem of e-commerce and the storage and distribution of resources. A proposed structure allows organizations to reduce costs through the effective implementation of ecommerce activities and solve the problem of large companies to improve e-commerce applications through cloud computing.

Agrawal & Dhar (2021) reported in this paper primarily focuses on India’s top B2C organizations and the role of definitive user experience elements in shaping the success of these organizations. A heuristic-based comparative study was conducted among the top e-commerce Web applications in order to identify unique features that contribute to a sumptuous user experience. The analytical study was backed up by user studies that report the expectation of the customers and their frustrations. The collective insights were then correlated to each platform’s market ranking and share with an objective to identify and analyze features, design cues and elements that contribute to their standing in the Indian market. In a nutshell, the qualitative study explores connections between market ranking, usability heuristics, and user study insights in order to prescribe design features, cues, and elements that benchmark the user experience framework for B2C applications in an Indian context. The insights elaborate on the current trends, gaps, and opportunity areas for B2C applications.

Defiesta et al. (2020) discusses the development of an ecommerce platform with inventory integration for a mining equipment supplier companies. Because of e-commerce's low barrier of entry, many vendors and distributors sell directly to customers. With this, there is an opportunity to automate the business processes of the mid-size mining equipment supplier company to improve the service and sales performance. In turn, it can ease the management profit of the distributor channel. The system produces an e-commerce web application that can be overall access by the administrator. Likewise, distributors and customers' accounts can be partly access through the web application. This research includes modules to improve the system such as admin module, vendor module, distributor module, customer module and IT management system.

Nguyen (2021) The purpose of the thesis is to study the origins of Single-Page Applications and follow along the evolution of JavaScript and its ecosystem on how they redefine the norms of building modern web applications. To demonstrate the functionalities of the mentioned JavaScript tooling, the thesis will include examples of SPAs as subjects of research, and explore how they are being orchestrated in harmony under-the-hood by tools and libraries such as React.js, Next.js, Webpack.js, TypeScript. In ad-dition, different types of SPA will be taken into account, for example server-side ren-dering, client-side rendering and static site generation, to measure impact of JavaS-cript toolkit on the process of creating those applications.

**CHAPTER THREE**   
**METHODOLOGY AND DESIGN**

**3.1 Introduction**

A methodology is a formal study or research, particularly to unveil new facts or information; thus, research methodology should be good enough to enable the accomplishment of the specified objectives, which are accomplished through the use of specific components such as data collection and design procedures, and system modeling (use case, activity, and class diagrams).

**3.2 Methods of Data Collection**

Before constructing any system, it is necessary to collect data and facts about the existing system to comprehend what is going on. Two approaches were used in this study.

i. Observation of the Work Environment

ii. Documentation

**3.2.1 Observation of the Work Environment**

This strategy was used to collect information and data for this study by observing how the manual system functioned. Detailed inspection revealed the most glaring deficiencies in the present system. The setting in which the observation is made can be altered in a variety of ways when using the observational technique.

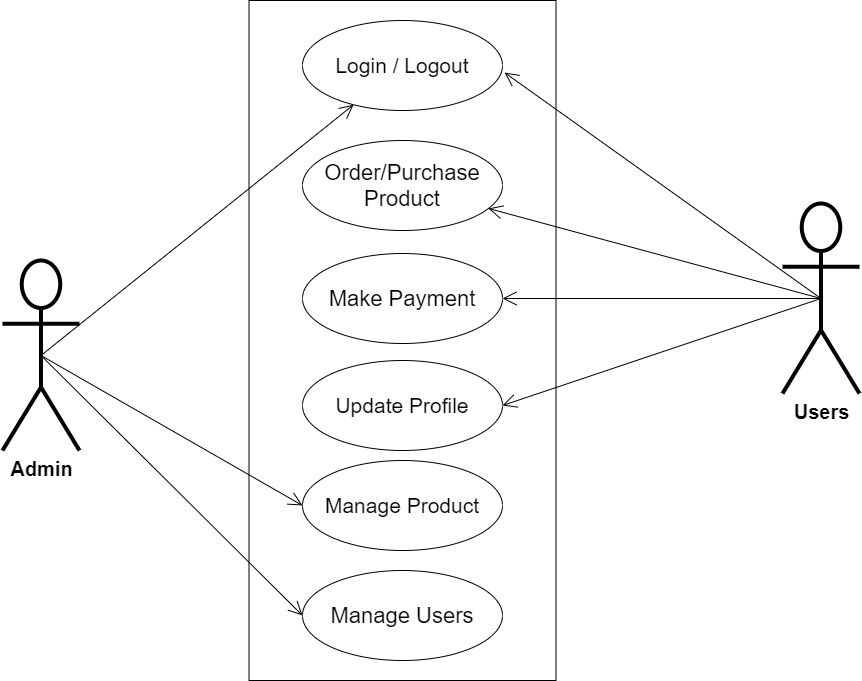
**3.2.2 Documentation**

Secondary data gathering includes documentation. Journals, books, previous work, publications, and other sources are used in this manner. This data-gathering technique is chosen because it allows for comparison with previous research. This includes the internet, which is a tool for data collection. The internet was utilized to study complex or unclear problems.

**3.3 System Modeling**

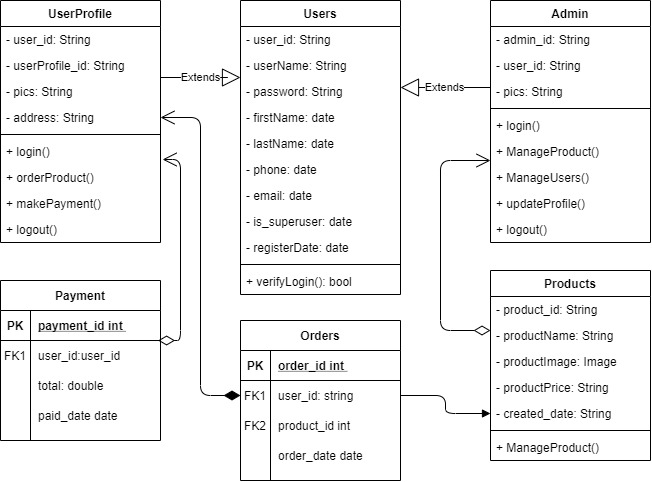
A system model is a conceptual model that characterizes and portrays a system as an outcome of system modeling. It is the connection of several components that collaborate to accomplish a shared goal. A collection of visual notation methods included in the Unified Modeling Language, which is utilized in the creation of this current system, may be used to generate visual models of object-oriented software-intensive systems. Use case diagrams, class diagrams, and activity diagrams are among the UML diagrams used in this new design.

**3.3.1 Use Case Diagrams**



**Fig 3.1 System Use Case Diagram**

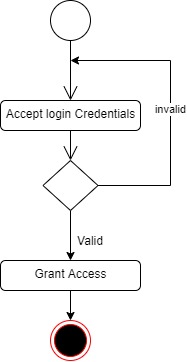
**3.3.2 Class Diagrams**



**Fig 3.2 System Class Diagram**

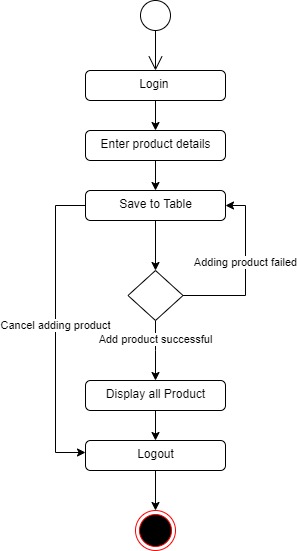
**3.3.3 Activity Diagrams**

**Login**

The process for gaining access to the system is depicted in the diagram below; the username and password must be accurate to gain access.

**Fig 3.3 System Login Activity Diagram**

**Create Product**

The process for adding product to the system is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the creation.

**Fig 3.4 Create Product Activity Diagram**

**3.4 Database Design**

The following are some of the input specifications used in this project work.

1. Users Table: contains the generic information of all system users.
2. Product Table: contains every grocery registered on the system.

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| username | Varchar | No |  | 20 | Unique name for users |
| password | Varchar | No |  | 128 | User Password |
| first\_name | Varchar | No |  | 20 | User first name |
| last\_name | Varchar | No |  | 20 | User last name |
| phone | Varchar | No |  | 11 | User phone number |
| email | Varchar | No |  | 100 | User email address |

**Table 3.2 Product Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| product\_id | Varchar | No | PK | 32 | Unique string for identifying citizens |
| productName | BigInt | No |  | 10 | Product name |
| productImage | BigInt | No |  | 60 | Image to identify the product |
| productPrice | Varchar | No |  | 60 | Price of the product |
| created\_date | Varchar | No |  | 60 | Date the product was registered |

**3.5 Output Design**

This declares and displays the outcome of the given input. This system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Users** **output design table**

**List of the System Registered Users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **Username** | **First\_name** | **Last\_name** | **Phone** | **Email** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

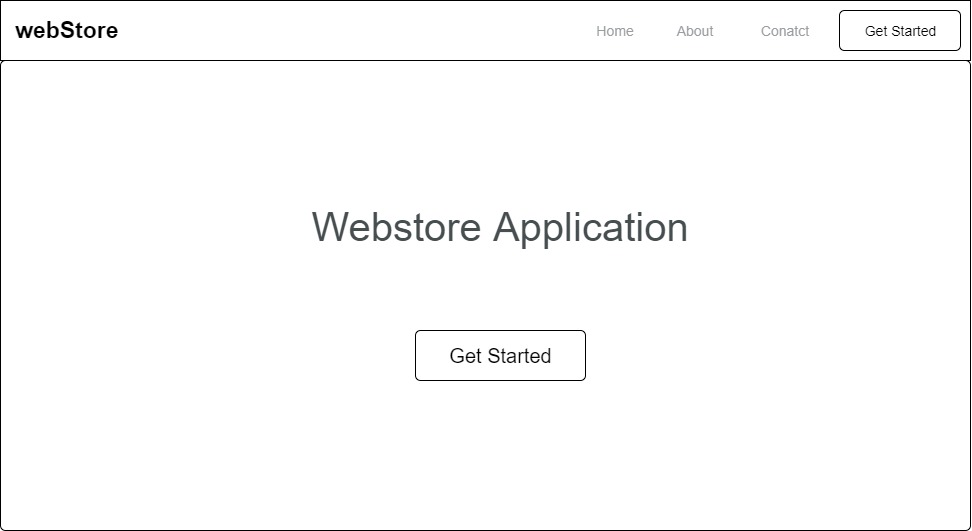
**Table 3.4 Product** **output design table**

**List of the System Registered Products**

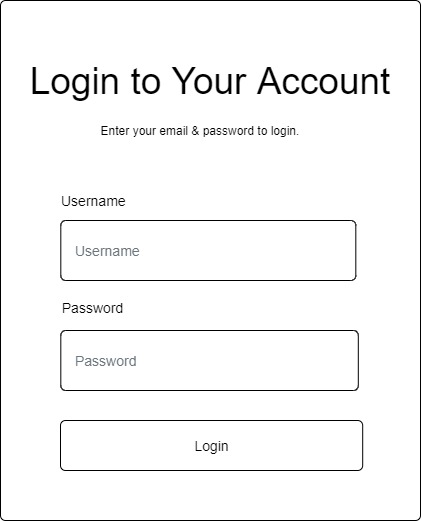
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product\_id** | **ProductName** | **ProductImage** | **ProductPrice** | **Created\_date** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

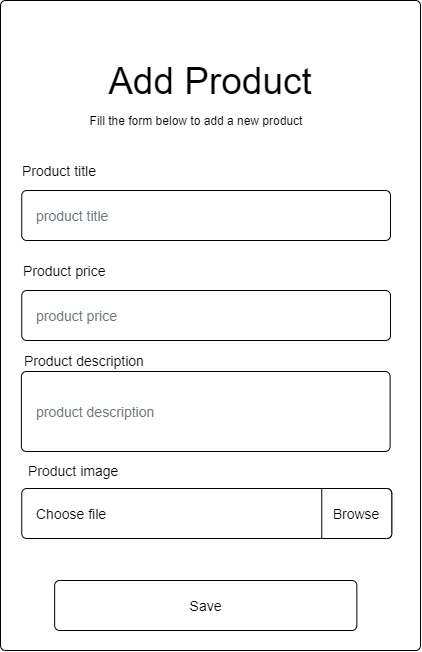
**3.6 Input & User Interface Design**

This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be appropriately safeguarded, thus authorization will be required to see certain levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.6 Home Page**



**Fig 3.7 Login Form**

**Fig 3.8 Add Product**

**3.7 System Requirement**

Every software system built has a stated system requirement on which it is meant to execute for the best performance. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system to work properly.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research project will be a developed using Angular & Node.js with a customizable backend Services. The combination of the above will help build a very robust webstore application that will be useful, fast, and handy.

**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION EVALUATION**

**4.1 Introduction**

This section describes in detail how the new system will be implemented in order to assure its efficacy. It illustrates instances of functional (new) systems as well as how the system will be implemented.

* 1. **System Testing and Evaluation**

The developed system should be tested for a variety of reasons. For example, only via testing will we be able to detect and address any problems in the new system. Unit and integration testing were used in this project to confirm the design's efficacy and efficiency, as well as to ensure the new system satisfies its functional requirements and is error-free.

**Unit Testing**

specific units or single components of the system are examined individually in this part to confirm that specific phases function properly and without problems.

**Integration Testing**

The program was tested via integration testing, in which all of the components were integrated and worked as one. The connection between the different components was examined to ensure that they are correctly integrated and that the units can function as a unit.

**4.3 System Installation**

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, pip, pipenv, and python3 or greater are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open project folder in Visual Studio Code
4. On the terminal run “pipenv install -r requirements.txt”
5. On the terminal run “python manage.py runserver”
6. Open any browser on the system example Chrome, Microsoft Edge, or Mozilla Firefox.
7. On the address bar, type http://127.0.0.1 and press the enter key the site should be loaded.

**4.4 Security Measures**

Since the scope of the application is public, literally all the information is made available to any user, but some functionalities are restricted to the admin, functionalities that have to do with adding products, managing user accounts, etc are restricted from the general user. The restriction is carried out by using passwords when the application is accessed.

**4.5 Sample Outputs**

These describe and give the pictorial representation of the program or software; it shows and gives clear understanding of the design, and displays all the interfaces

**Homepage**

The image provided illustrates the homepage, which serves as the initial page and serves as a gateway to navigate and explore the various sections and functionalities of the website

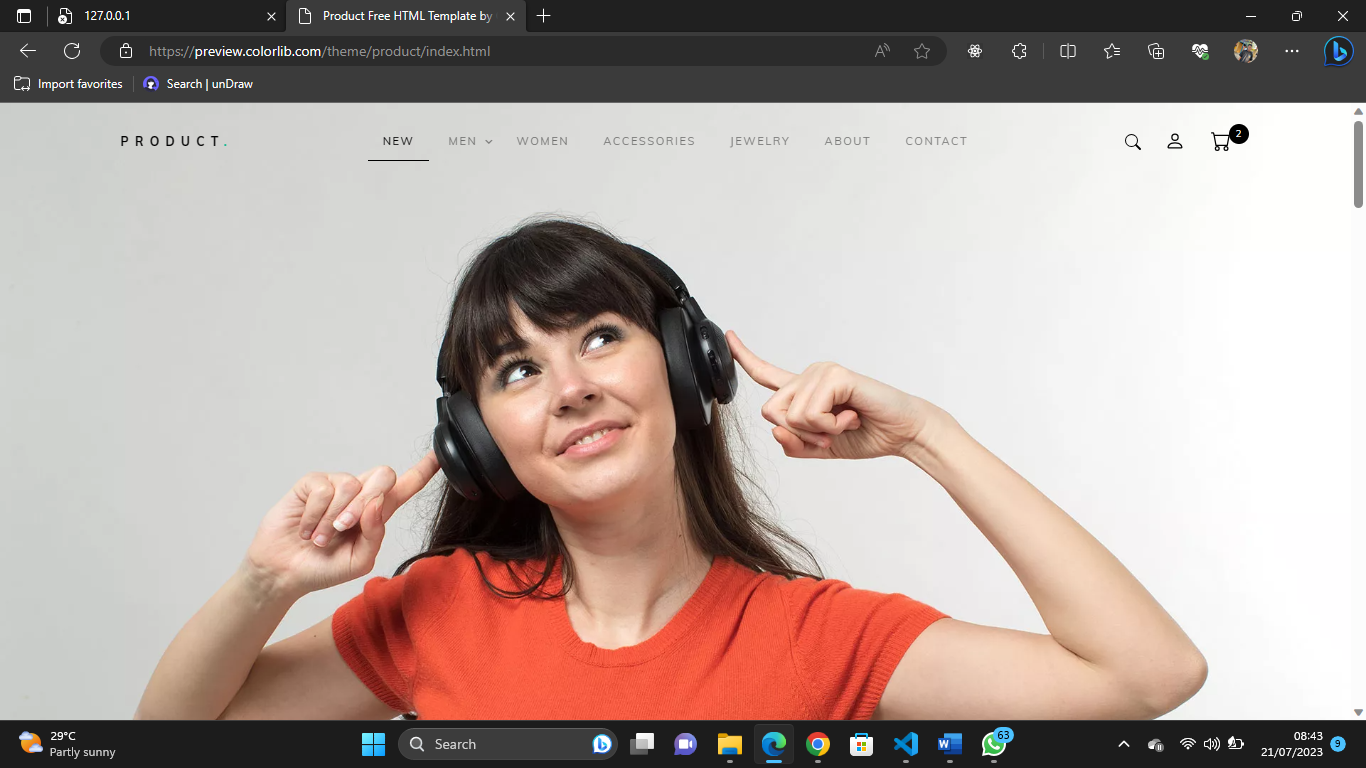


Fig 4.5.1 Homepage

**Products**

The page displays all the products available to the user for purchase

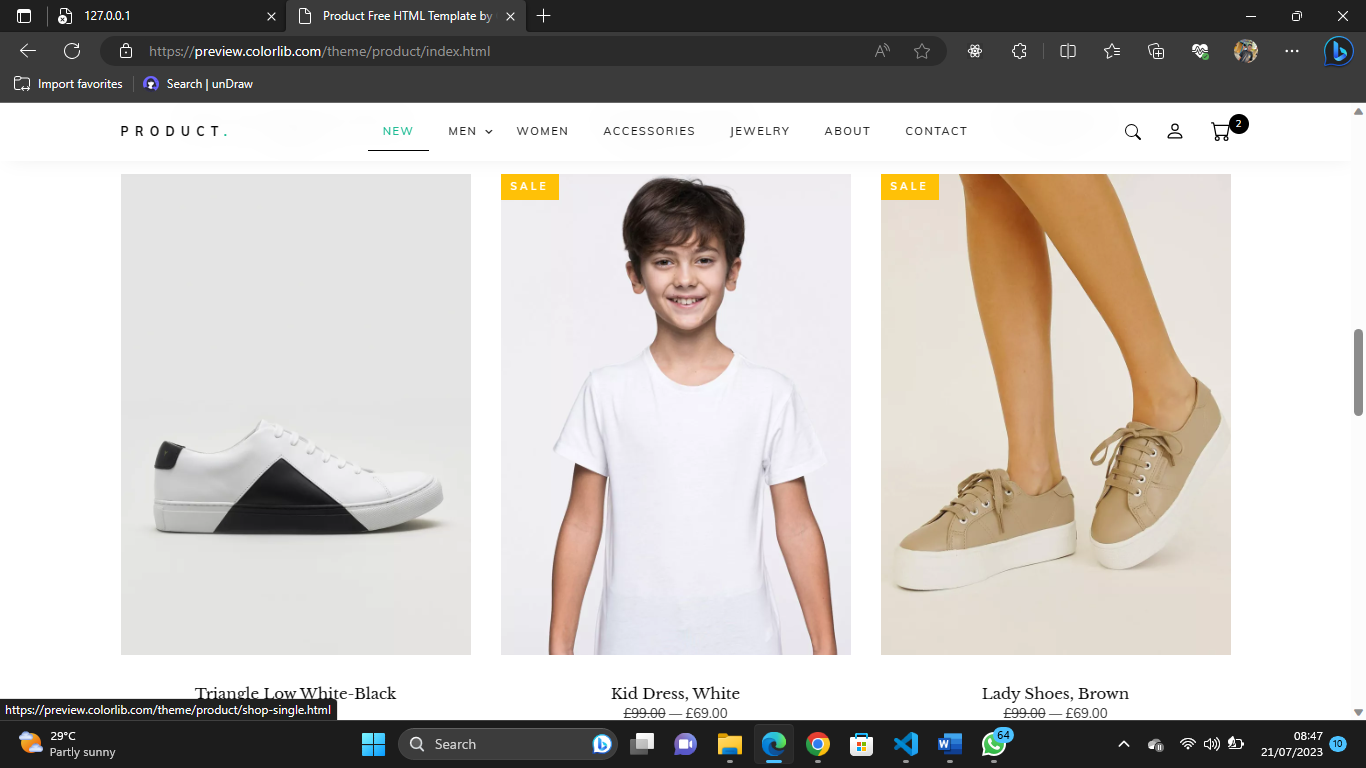


Fig 4.5.2 Products

**Product Categories**

This page highlights key categories of products that are available and users can click to purchase them.

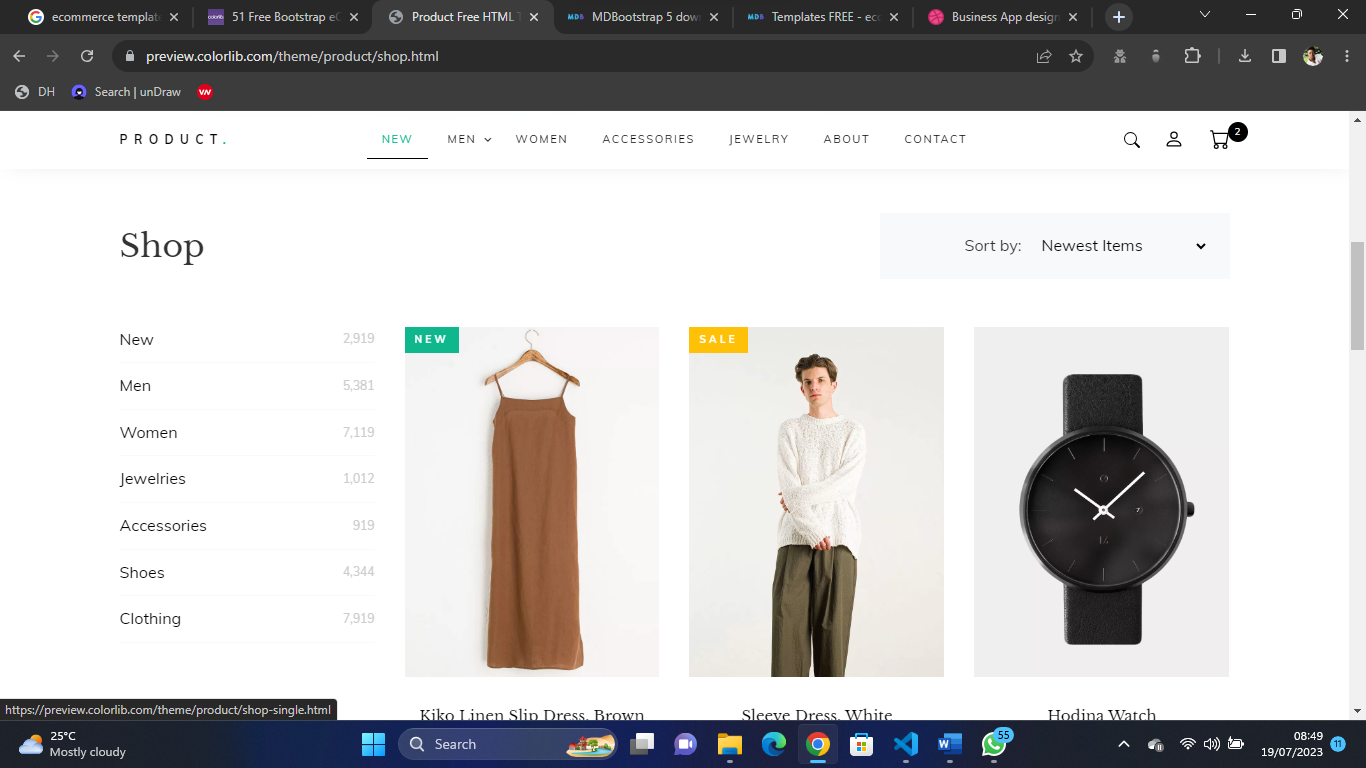


Fig 4.5.3 Product Categories

**Cart**

Authenticated users can view items in the cart, and increase or decrease the items before proceeding to checkout

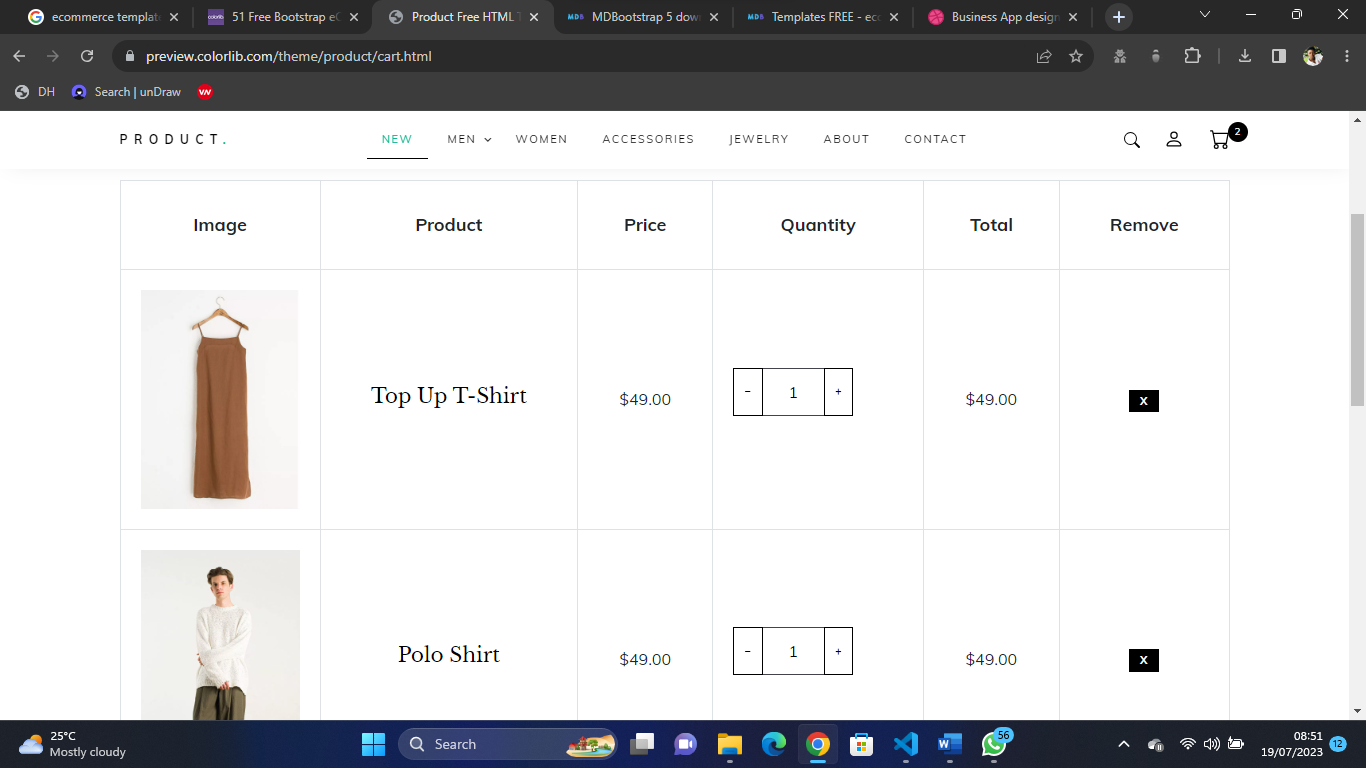


Fig 4.5.4 Cart

**About Us**

This page contains detailed reason why end-users should make use of our solution

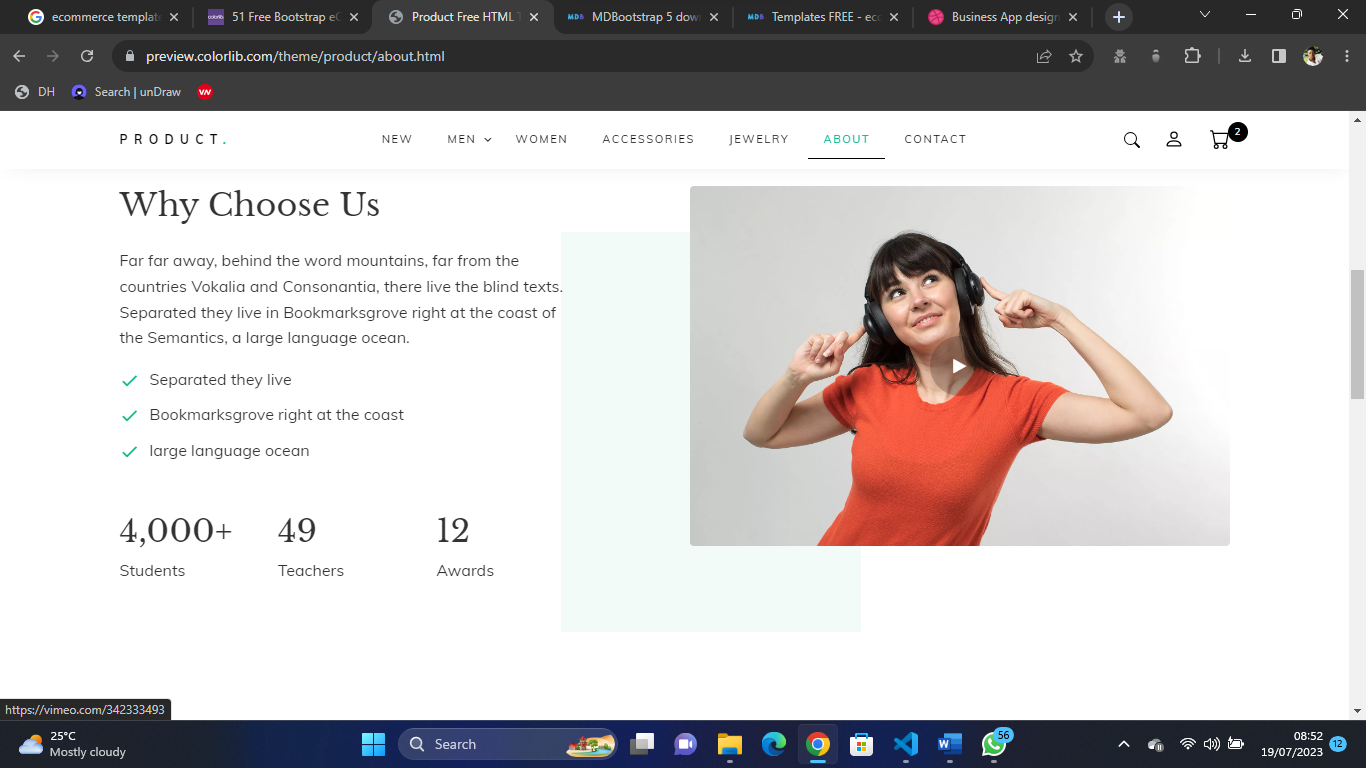


Fig 4.5.4 About Us

**Contact Us**

This is the screen where the user can contact the admin possibly because of any difficulty faced in using the application

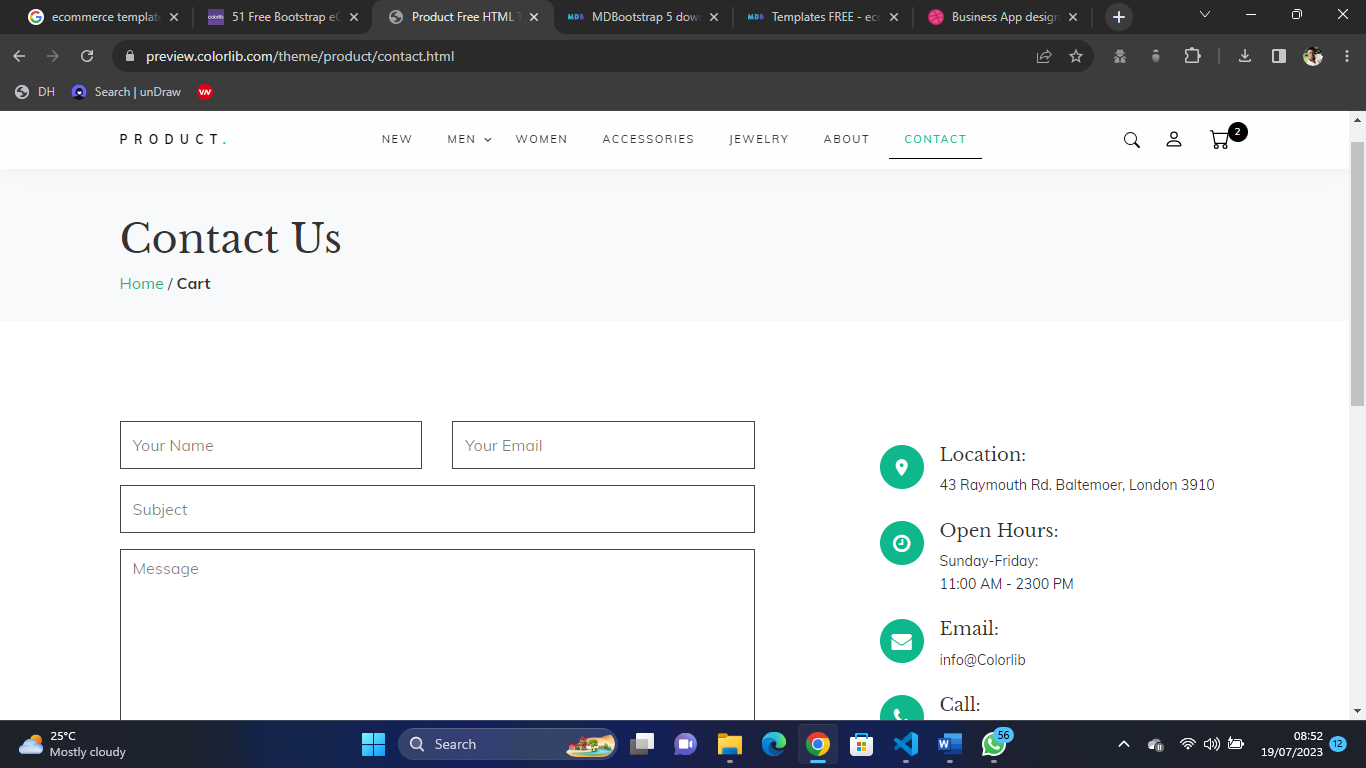


Fig 4.5.5 Contact Us

**Checkout Page**

This is the page where the user can checkout the placed order

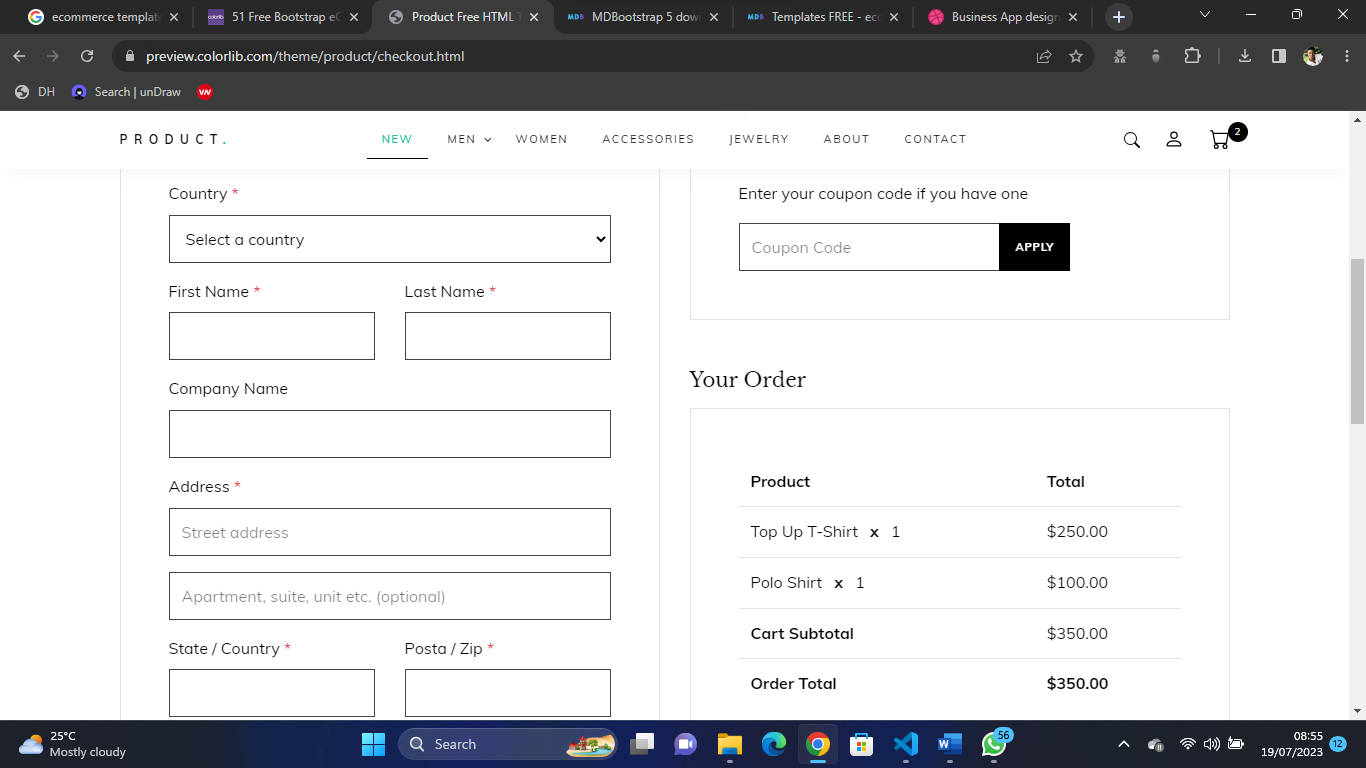


Fig 4.5.6 Checkout Page

**REFERENCES**

Agrawal, M., & Dhar, D. (2021). Enhancing user experience of e-commerce platforms—a case study of B2C applications in the Indian market. *Design for Tomorrow—Volume 1*, *1*, 503–516. https://doi.org/10.1007/978-981-16-0041-8\_42

Defiesta, M. L., Estrada, D. G., Veloso, A. N., Llesol, M. G., & Intal, G. L. (2020). B2B and B2C e-commerce platform for mining equipment suppliers. *Proceedings of the 2020 2nd International Conference on Management Science and Industrial Engineering*. https://doi.org/10.1145/3396743.3396772

Eisenman, B. (2015). *Learning react native*. O'Reilly Media.

Qwaider, W. Q. (2020). A Conceptual Model for E-Commerce Applications based a Cloud Computing. *International Journal of Computer Applications (0975 – 8887),* *176*(13), 18–22.

Saha, S. (2018). *Implementation Of An E-Shopping Website* (thesis). Daffodil International University, Daffodil Smart City.

Sufiyan, T. (2022, November 24). *What is node.js: A comprehensive guide*. Simplilearn.com. Retrieved December 24, 2022, from https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs

Wu, Q., Ma, J., & Wu, Z. (2020). Consumer-driven e-commerce: A study on C2B applications. *2020 International Conference on E-Commerce and Internet Technology (ECIT)*. https://doi.org/10.1109/ecit50008.2020.00019

Xu, G., & Chen, Q. (2020). On the application of large data technology in B2C E-Commerce Precision Marketing Mode. *Advances in Intelligent Systems and Computing*, 1197–1203. https://doi.org/10.1007/978-981-15-2568-1\_166

Deshpande, C. (2022, November 25). *What is angular?: Architecture, features, and advantages [2022 edition]*. Simplilearn.com. Retrieved December 24, 2022, from https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular

Nham, T. (2022). *Developing an E-commerce application prototype with ReactJS and Firebase* (thesis). Metropolia, Helsinki.

Nguyen, P. (2021). *How JavaScript ecosystem and open-source tooling enable a modern era of Single-Page Applications* (thesis). Metropolia, Helsinki.

**APPENDIX**

**Homepage**

{% extends 'base.html' %}

{% load static %}

{% block title %} Home{% endblock %}

{% block head %}

   {% include 'partials/head.html' %}

{% endblock %}

{% block body %}

<div class="hero\_area">

   <!-- header section strats -->

   {% include "partials/nav.html" %}

   <!-- end header section -->

   <!-- slider section -->

   <section class="slider\_section ">

      <div class="slider\_bg\_box">

         <img src="{% static 'images/slider-bg.jpg' %}" alt="">

      </div>

      <div id="customCarousel1" class="carousel slide" data-ride="carousel">

         <div class="carousel-inner">

            <div class="carousel-item active">

               <div class="container ">

                  <div class="row">

                     <div class="col-md-7 col-lg-6 ">

                        <div class="detail-box">

                           <h1>

                              <span>

                              Best Service

                              </span>

                              <br>

                              On Everything

                           </h1>

                           <p>

                             Delfak Nigeria Limited the sure construction material shop.

                           </p>

                           <div class="btn-box">

                              <a href="{% url 'auth:login' %}" class="btn1">

                              Get Started

                              </a>

                           </div>

                        </div>

                     </div>

                  </div>

               </div>

            </div>

            <div class="carousel-item ">

               <div class="container ">

                  <div class="row">

                     <div class="col-md-7 col-lg-6 ">

                        <div class="detail-box">

                           <h1>

                              <span>

                              Best Quality

                              </span>

                              <br>

                              On Everything

                           </h1>

                           <p>

                              Delfak Nigeria Limited the sure construction material shop.

                           </p>

                           <div class="btn-box">

                              <a href="{% url 'auth:login' %}" class="btn1">

                              Get Started

                              </a>

                           </div>

                        </div>

                     </div>

                  </div>

               </div>

            </div>

         </div>

         <div class="container">

            <ol class="carousel-indicators">

               <li data-target="#customCarousel1" data-slide-to="0" class="active"></li>

               <li data-target="#customCarousel1" data-slide-to="1"></li>

            </ol>

         </div>

      </div>

   </section>

   <!-- end slider section -->

</div>

<!-- footer start -->

{% include "partials/footer.html" %}

<!-- footer end -->

{% include "partials/script.html" %}

{% endblock %}

**Checkout page**

{% extends 'base.html' %}

{% load static %}

{% block title %} Checkout{% endblock %}

{% block head %}

{% include "partials/auth\_head.html" %}

{% include "partials/stripe.html" %}

{% endblock%}

{% block body %}

{% include "partials/preloader.html" %}

<div id="main-wrapper">

    {% include "partials/login\_nav.html" %}

    {% include "partials/sidebar.html" %}

    <div class="content-body">

        <div class="row page-titles mx-0">

            <div class="col p-md-0">

                <ol class="breadcrumb">

                    <li class="breadcrumb-item"><a href="{% url 'auth:dashboard' %}">Dashboard</a></li>

                    <li class="breadcrumb-item active"><a href="{% url 'auth:profile' user.id %}">Order Summary</a></li>

                </ol>

            </div>

        </div>

        <div class="container-fluid">

            {% include "partials/messages.html" %}

            <div class="row">

                <div class="col-lg-8 col-xl-8">

                    <div class="card">

                        <div class="card-body">

                            <h3 class="card-header mb-0">Checkout Form</h3>

                            <form class="card-body" method="post">

                                {% csrf\_token %}

                                <div class="row">

                                    <label for="admin\_card" class="col-12 col-form-label">Customer Details</label>

                                    <hr>

                                    <div class="col-md-6 col-sm-12">

                                        <label for="fullname" class=" col-form-label">Full Name</label>

                                        {{ form.fullname }}

                                        {% if form.fullname.errors %}

                                        <div class="alert alert-danger mt-2" role="alert">

                                        {{ form.fullname.errors }}

                                        </div>

                                        {% endif %}

                                        <div class="invalid-feedback">Please, enter your Full name</div>

                                    </div>

                                    <div class="col-md-6 col-sm-12">

                                        <label for="email" class=" col-form-label">Email</label>

                                        {{ form.email }}

                                        {% if form.email.errors %}

                                        <div class="alert alert-danger mt-2" role="alert">

                                        {{ form.email.errors }}

                                        </div>

                                        {% endif %}

                                        <div class="invalid-feedback">Please, enter your email</div>

                                    </div>

                                    <div class="col-md-6 col-sm-12">

                                        <label for="phone" class=" col-form-label">Phone Number</label>

                                        {{ form.phone }}

                                        {% if form.phone.errors %}

                                        <div class="alert alert-danger mt-2" role="alert">

                                        {{ form.phone.errors }}

                                        </div>

                                        {% endif %}

                                        <div class="invalid-feedback">Please, enter your Phone Number</div>

                                    </div>

                                    <div class="col-md-6 col-sm-12">

                                        <label for="address" class=" col-form-label">Address</label>

                                        {{ form.address }}

                                        {% if form.address.errors %}

                                        <div class="alert alert-danger mt-2" role="alert">

                                        {{ form.address.errors }}

                                        </div>

                                        {% endif %}

                                        <div class="invalid-feedback">Please, enter your address</div>

                                    </div>

                                </div>

                                <hr>

                                <button id="btnsubmit" class="btn btn-primary btn-lg btn-block waves-effect waves-light">Complete Sales</button>

                              </form>

                        </div>

                    </div>

                </div>

                <div class="col-lg-4 col-xl-4">

                    <div class="card">

                        <div class="card-header d-flex align-items-center justify-content-between">

                            <h5 class="mb-0">Your Cart</h5>

                            <small class="text-muted float-end">cart</small>

                        </div>

                        <div class="card-body">

                            <div class="basic-list-group">

                                <ul class="list-group">

                                    {% for order\_item in order.product.all  %}

                                    <li class="list-group-item d-flex justify-content-between align-items-center">{{ order\_item.product.title }} <span class="badge badge-primary badge-pill">{{ order\_item.quantity }}</span>

                                    </li>

                                    {% endfor %}

                                    <li class="list-group-item d-flex justify-content-between align-items-center"><b>Total</b> <span class="badge badge-primary badge-pill">₦{{ orders.get\_total }}</span>

                                </ul>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

    {% include "partials/login\_footer.html" %}

</div>

<script>

    // Custom styling can be passed to options when creating an Element.

    var style = {

      base: {

        // Add your base input styles here. For example:

        fontSize: '16px',

        color: '#32325d',

      },

    };

    // Create an instance of the card Element.

    var card = elements.create('card', {style: style});

    // Add an instance of the card Element into the `card-element` <div>.

    card.mount('#card-element');

    card.on('change', function(event) {

      var displayError = document.getElementById('card-errors');

      if (event.error) {

        displayError.textContent = event.error.message;

      } else {

        displayError.textContent = '';

      }

    });

    var form = document.getElementById('payment-form');

    // Create a token or display an error when the form is submitted.

    var form = document.getElementById('payment-form');

    form.addEventListener('submit', function(event) {

      event.preventDefault();

      stripe.createToken(card).then(function(result) {

        if (result.error) {

          // Inform the customer that there was an error.

          var errorElement = document.getElementById('card-errors');

          errorElement.textContent = result.error.message;

        } else {

          // Send the token to your server.

          stripeTokenHandler(result.token);

        }

      });

    });

    function stripeTokenHandler(token) {

    // Insert the token ID into the form so it gets submitted to the server

    var form = document.getElementById('payment-form');

    var hiddenInput = document.createElement('input');

    hiddenInput.setAttribute('type', 'hidden');

    hiddenInput.setAttribute('name', 'stripeToken');

    hiddenInput.setAttribute('value', token.id);

    form.appendChild(hiddenInput);

    // Submit the form

    form.submit();

  }

  </script>

{% include "partials/auth\_script.html" %}

{% endblock %}

Order Summary

{% extends 'base.html' %}

{% load static %}

{% block title %} Order Summary{% endblock %}

{% block head %} {% include "partials/auth\_head.html" %} {% endblock%}

{% block body %}

{% include "partials/preloader.html" %}

<div id="main-wrapper">

    {% include "partials/login\_nav.html" %}

    {% include "partials/sidebar.html" %}

    <div class="content-body">

        <div class="row page-titles mx-0">

            <div class="col p-md-0">

                <ol class="breadcrumb">

                    <li class="breadcrumb-item"><a href="{% url 'auth:dashboard' %}">Dashboard</a></li>

                    <li class="breadcrumb-item active"><a href="{% url 'auth:order\_summary' %}">Sales Summary</a></li>

                </ol>

            </div>

        </div>

        <div class="container-fluid">

            {% include "partials/messages.html" %}

            <div class="row">

                <div class="col-lg-8 col-xl-8">

                    <div class="card">

                        <div class="card-body">

                            <h3 class="card-header mb-0">Sales summary</h3>

                            {% include "partials/summary.html" %}

                        </div>

                    </div>

                </div>

                <div class="col-lg-4 col-xl-4">

                    <div class="card">

                        <div class="card-header d-flex align-items-center justify-content-between">

                            <h5 class="mb-0">Summary</h5>

                            <small class="text-muted float-end">cart</small>

                          </div>

                        <div class="card-body">

                            <div class="basic-list-group">

                                <ul class="list-group">

                                    {% for order\_item in order.product.all  %}

                                    <li class="list-group-item d-flex justify-content-between align-items-center">{{ order\_item.product.title }} <span class="badge badge-primary badge-pill">{{ order\_item.quantity }}</span>

                                    </li>

                                    {% endfor %}

                                </ul>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

    {% include "partials/login\_footer.html" %}

</div>

{% include "partials/auth\_script.html" %}

{% endblock %}