

ClickToCart: An E-Commerce Based Website

Submitted By

Vinit Shah

SEAT NO: 1689

Kevin Desai

SEAT NO: 1554

Shrishti Agarwal

SEAT NO: 1531

Supervised By

Mr. Mohamed Sohel Shaikh

Assistant Professor

BCA(SEM-VI) , CBPCC

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Computer Applications (BCA)



C.B. Patel Computer College (CBPCC),
Affiliated to Veer Narmad South Gujarat University(VNSGU)

Abstract

This project presents a full functional website for a medium sized business enterprise. It will provide convenient support for the parties relating to the business *e.g.*, customers, staff, owners etc. This application will also help the business personnel manage and distribute the product effectively. In addition, related information of product, customer support, and online transaction features are available here. The key intent of this project is to maintain a professional standard to compete in the current market of online business.

Letter for Acceptance

This Project entitled “ClickToCart: An E-Commerce Based Website” submitted by Vinit Shah(SEAT NO: 1689) , Kevin Desai (SEAT NO: 1554) and Shrishti Agrawal(SEAT NO: 1531) to the C.B. Patel Computer College (CBPCC), Veer Narmad South Gujarat University(VNSGU) University, Surat, Gujarat. It is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Computer Applications (BCA) for year 2021.

Acknowledgements

First, we are thankful and expressing our gratefulness to Almighty who offers us divine blessings, patience, mental and psychical strength to complete this project. The progression of this project could not possibly be carried out without the help of several people who, directly or indirectly, are responsible for the completion of this work. We are deeply indebted to our project supervisor Mr. Mohamed Sohel Shaikh. His scholarly guidance, especially for his tolerance with our team and unfailing support. He gives us the freedom to pursue aspects of reversible fault tolerant computing which we found interesting and compelling. This helped our project to achieve its desired goals.

We wish to thank the great people of C.B. Patel Computer College (CBPCC). A special thank goes to all faculties for their well-disposed instructions and Encouragements.

Finally, we would like to thank our friends and family. Their continued tolerance with our moods and tendency to disappear for weeks at a time gave us a much needed break from the world computing.

Table of Contents

Abstract	II
Letter of Acceptance	III
Acknowledgements	IV
 Chapter 1: Introduction	
1.1. Motivation	1
1.2. Key Features	2-3
1.3. Scope	3-4
1.4. Outline	4-5
1.5. Summary	5
 Chapter 2: Requirement Analysis	
2.1.Functional Requirements	6
2.2.Non-functional Requirements	7
2.3.Class Diagrams	8
2.3.1 Application Class Diagram	8
2.3.2 Class Diagram for Admin	9
2.4.Use Case	10
2.5.Activity Diagram	11
2.6. Software Requirements	12
2.7. Plugin Requirements	12
2.8. Summary	12
 Chapter 3: Application Functionalities	
3.1. User Registration and Verification	13-14

3.2. Throttled Login	14
3.3. Product Searching	14-15
3.4. Cart Functionalities	15-16
3.5. User Profile	16-17
3.6. Checkout	17
3.7. Receipt	17-18
3.8. User Feedbacks	18
3.9. Product Details	18
3.10. Related Products	19
3.10.1 Algorithm	19
3.11. Customer Support	19
3.12. Sale Management	20
3.13. Product Management	20
3.14. User Management	20
3.15. Summary	20

Chapter 4: Implementation Plan

4.1. A Quick Glance at Laravel Framework	21
4.1.1 Key Terms	22
4.2. Project Coordination	23
4.2.1 Version Controlling	23
4.2.2 Merging	23
4.3 Summary	23

Chapter 5: Technical Procedures

5.1 Ease of Use	24
5.1.1 Auto Suggestion	25
5.1.2 Search Result Sorting	25
5.1.3 Automatic Sale Deletion	25-26
5.2 Server-side validation	26
5.2.1 During Registration	27
5.2.2 In Cart Functionalities	27

5.2.2(a) Problematic Scenarios	27-28
5.2.2(b) Prevention	28
5.2.3 In Checkout	28-29
5.3 Redundancy Free Database	29
5.3.1 In Product Management	29
5.3.2 In Advanced Info for Users	30
5.3.3 In Receipts and Orders	30-31
5.4 Summary	31

Chapter 6: View Optimization

6.1 Using Service Providers	32
6.2 View Stock Availability	33
6.3 Search Bar	34
6.4 Summary	44

Chapter 7: Data dictionary

7.1 List of all tables	45
7.2 Table detail view	64

Chapter 7: User interface

8.1 Admin side	65
8.2 User side	79

Chapter 9: Conclusion

9.1 Learnings	80
9.2 Conclusion	81
9.3 Future Works	81

List of Figures

Figure 2.1: Class diagram of the application.	8
Figure 2.2: Class diagram for the Admin	9
Figure 2.3: Use Case Diagram	10
Figure 2.4: Activity diagram for users purchasing products	11
Figure 3.1: Auto complete suggestion	14
Figure 3.2: Cart Functionalities	16
Figure 3.3: View of a User's Profile	17
Figure 3.4: A Purchase Receipt	18
Figure 5.1: Deleting sales on ending date	26
Figure 5.2: Task scheduling for custom command	26
Figure 6.1: Categories (a) and Brands (b) on navigation bar fetched from database.	33
Figure 6.2: Stock availability indicator	33

Chapter 1

Introduction

ClickToCart is an e-commerce based website that makes the trade for buyers and sellers in few clicks. This website helps the business personnel (*e.g.*, administrators, managers) to easily manage users and products, apply or remove sales, keep track of orders etc. In addition, users can search and buy products through secure online transactions and see their order histories later. The scenario seems pretty basic, and the idea is to keep it that way through backend complex fail-safe design.

1.1 Motivation

For this project we considered building a full professional website because of two key factors:

- i. In recent years, e-commerce has been dominating the world of trades and business. Due to the ease-of accessibility of the internet and microcomputer devices along with the simple manner of usability, it has been a sector where more and more people are merging every day. We wanted to get a little insight of the practical events and issues related to this sector.
- ii. Our goal was to build a professional standard project, which would help us to get an idea about how things work in real world (*i.e.*, Website development firms). Working with Laravel has enabled us to become a little familiar to such kind of a workflow.

In short, mixing and exploring two of the largest sectors that not only keeps growing fast in different parts the globe but also shows a great deal of promise in India's near future as well- was the actual motivation behind our project.

1.2 Key Features

- **E-mail verification** for user registration
- **Proper user redirection** after different tasks
- **Fail safe** manner for user inputs
 - Front end warnings and restrictions
 - Strict **Server-side validation**
 - Strict **Client-side validation**
- **Free from data redundancy**
 - Thoroughly used Laravel's ORM (Object Relational Mapping)
 - Framework's **OOP** concept gives extra benefit in keeping the data safe and the security intact
- **Autocomplete suggestions** for search queries
- **Find products by Categories, Subcategories and Brands etc.**
- **Search** products conveniently
 - **Sort by attributes**
- **User Review** on products can be added
- Administrators can **add Products, Categories, Brands** and perform various other functionalities.
- **Product Checkout** is done safely and securely using stripe.
 - Account authentication is validated

- **Stock availability** is validated throughout multiple step of order
 - During Adding to Cart
 - During final checkout
- **Order status** can be monitored by managers/admins afterwards
 - Staff can change
- A **Printable Receipt** is given to the user after the checkout is completed
 - Includes **unique receipt number**, transaction date etc.
- Users' **Cart** info is saved in **Order** table inside database
 - Order includes payment_id, user_id etc.
- **User support**
 - **Contact** staff through messaging to get info
 - Create **tokens** for refund.
 - **Get help** from staff through email messages
- **Related Products**
 - Suggest related products

1.3 Scope

Guests:

(Unsubscribed users)

- Search products
- View product details
- Share product(on Facebook, Whatsapp, Instagram etc.)
- Add to cart (Cart access will result in a prompt to login/register)
- Subscribe to newsletter
- View blogs

Subscribers:

- Search products
- View product details
- Share product(on Facebook, Whatsapp, Instagram etc.)
- Add reviews on products

- Add to cart ○ Increment, decrement products by 1 ○ Discard a product from cart
- Add to wishlist ○ Process to product detail page ○ Discard a product from wishlist
- Delete cart
- Checkout
- Track orders
- Download invoices
- Check previous orders
- Change/Reset password
- Store/Change users' advanced info ○ View self-profile
- Contact staff

Administrator:

- Search/view reports by date, month, year.
- Create/update/delete categories, subcategories and brands
- Create/update/delete/view/(active/inactive) products
- Create/update/delete coupons
- Create/update/delete blogs
- Manage site settings(including T&C, disclaimer, privacy policy)
- Update SEO settings
- Manage newsletters' subscribers
- Manage Order(accept, decline or cancel)
- Return order approval
- Update product stock
- Give supports and solution to users via emails.
- Complete power over users (including subadmins) ○ Add/delete users ○ Change user roles etc.

1.4 Outline

The next chapter (Chap.2) discusses the requirement analysis of the project. Diagrams from different perspectives and software requirements are also shown in the chapter.

Chap.3 discusses the application's functionalities i.e. discussion of special and usability features in detail.

Chap. 4 discusses the implementation plan of the project, i.e. the approaches taken for completion and maintenance of individual and cooperative tasks.

Chap.5 shows the technical procedures for modules of the application. Some potential risk and problematic scenarios and their solutions, preventions and validation processes are discussed in the chapter.

Chap 6 discusses the view optimization in brief.

Chap. 7 finally discussed about Conclusions and Future work.

1.5 Summary

This chapter demonstrates motivations, features and scopes of this project. A brief elementary instructional text of remaining chapters of this project have also been described in outline.

Chapter 2

Requirement Analysis

Requirement Analysis, also termed as SRS (Software Requirements Specification) is the process of finding the requirements for developing a software, i.e., data-dependencies, software dependencies, time-constraints, feasible features to be applied etc. Usually this includes interviewing personnel from client or buyer side, who provides with the primary expectations of what the application might or should do. As we do not have any clients, we brainstormed to find out some functional requirements of the website.

2.1 Functional Requirements

- Users must be able to search for products
- Users must be able to view product details
- Users must be able to share product on various platforms.
- On registration, users should have to verify their e-mails
- Users must be able to add or delete products to cart
- Users must be able to add or delete products to wishlist
- Subscribers should be able to add review on products
- Subscribers should be able to view and edit their profile info
- Subscribers must be able to change/reset password
- Subscribers should be able to view order histories
- Admin must be able to add, update and delete products
- Admin must be able to add, update and delete brands, categories and subcategories etc.
- An interactive communication system should be set between admin, subadmins and subscribers for user support.
- Admin should be able to see and change orders' current statuses
- Admin must be able to carry out all the tasks a subadmins can do, additionally should have complete power over all the users.

2.2 Non-functional Requirements

- E-mail verification
- Server-side validation
- Proper routing
- Ease-of-use
- Stock validation
- Payment validation
- Redundancy free

2.3 Class Diagrams

2.3.1 Application Class diagram

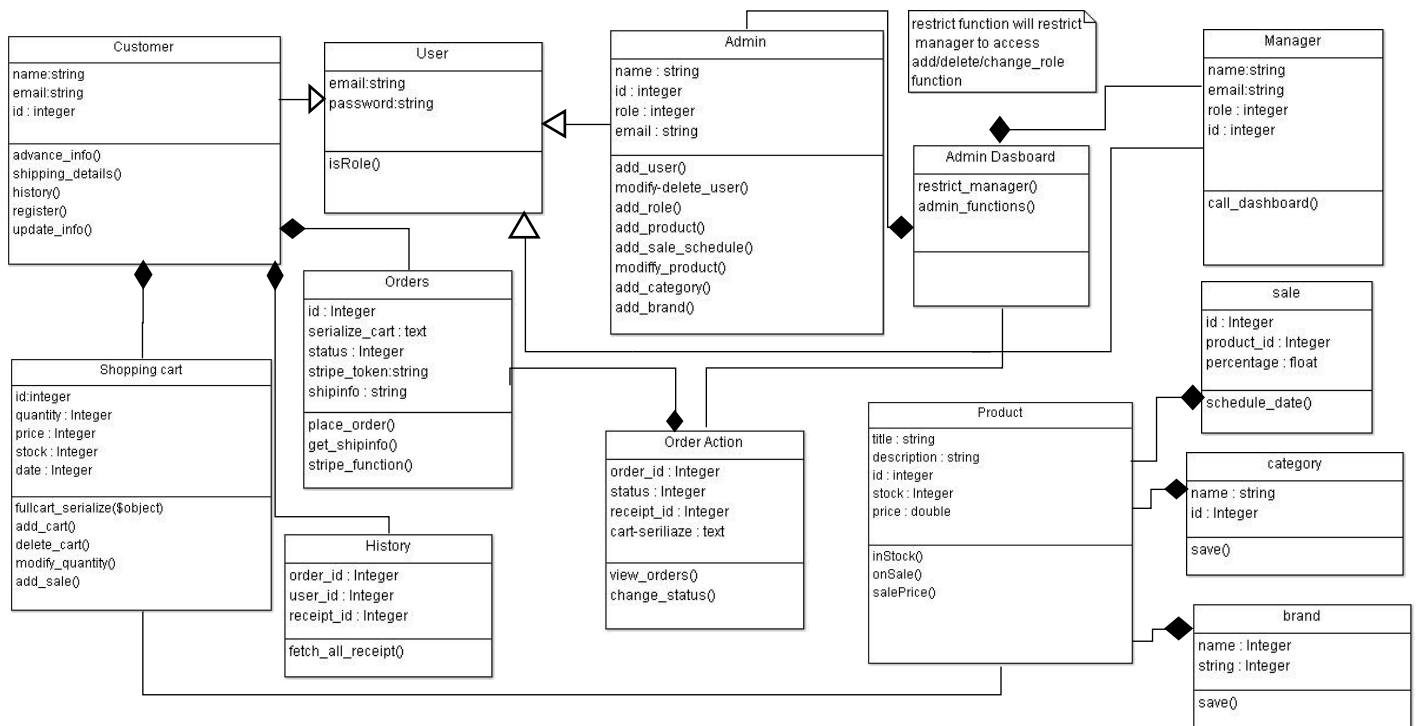


Fig 2.1: Class diagram of the application.

2.3.2 Class Diagram for the Admin

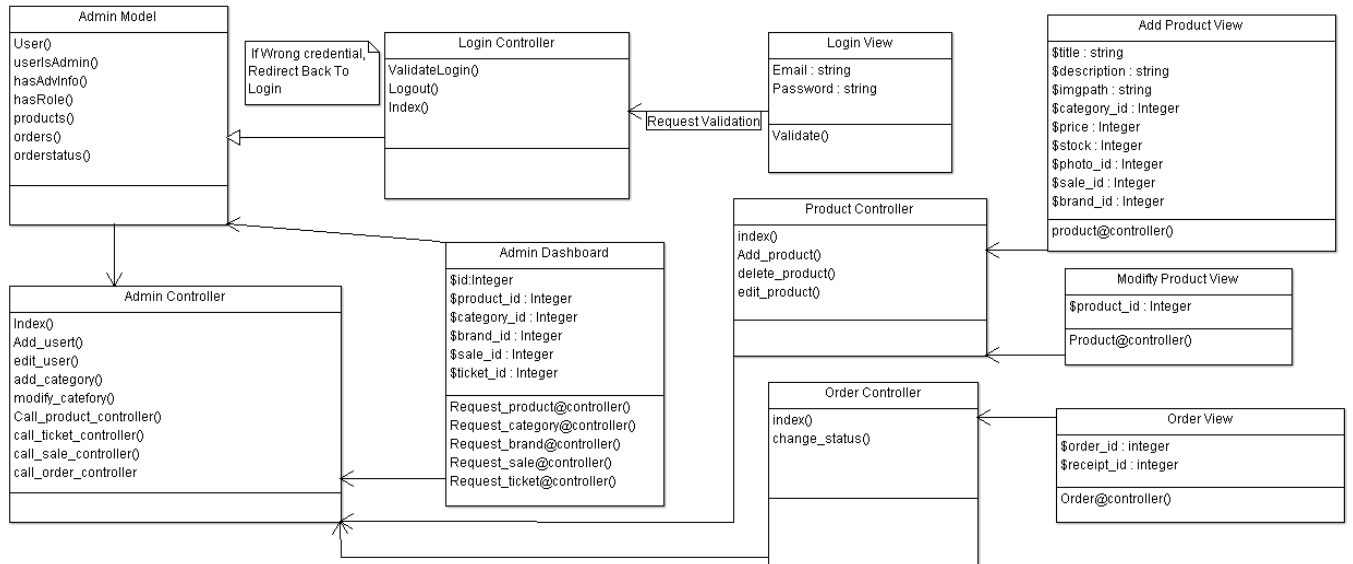


Fig. 2.2: Class diagram for the Admin.

2.4 Use Case Diagram

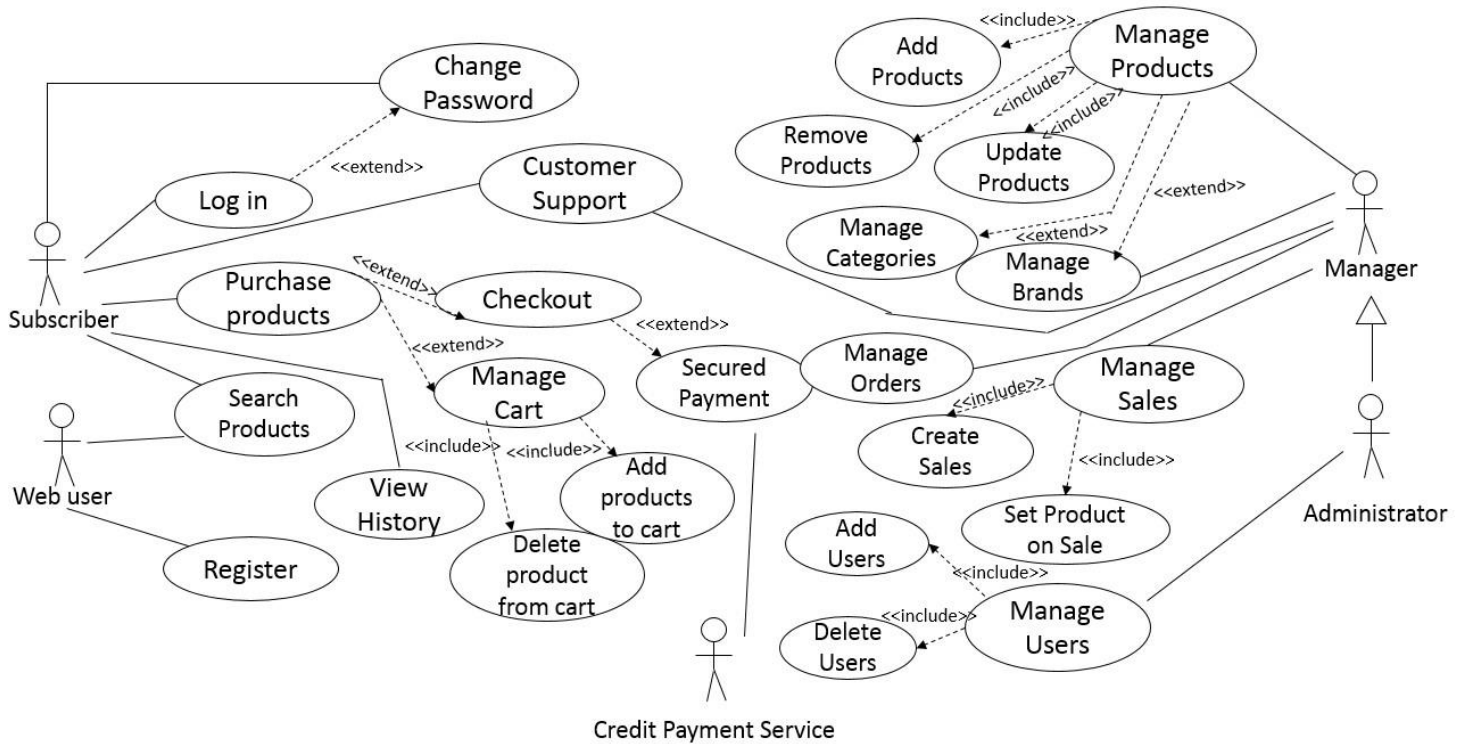


Fig. 2.3: Use Case Diagram

2.5 Activity Diagram

The activity diagram below shows the process of product purchasing using the application.

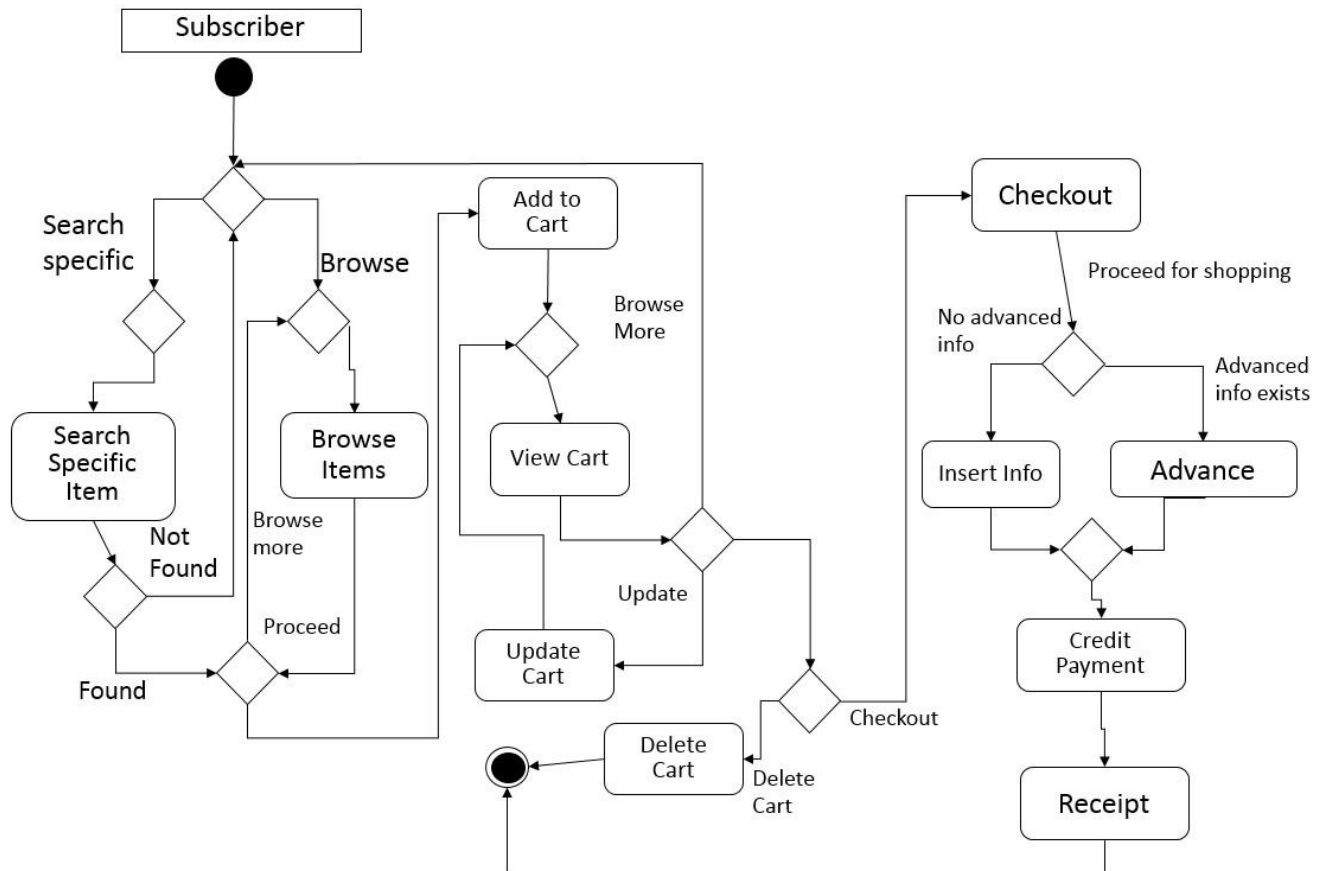
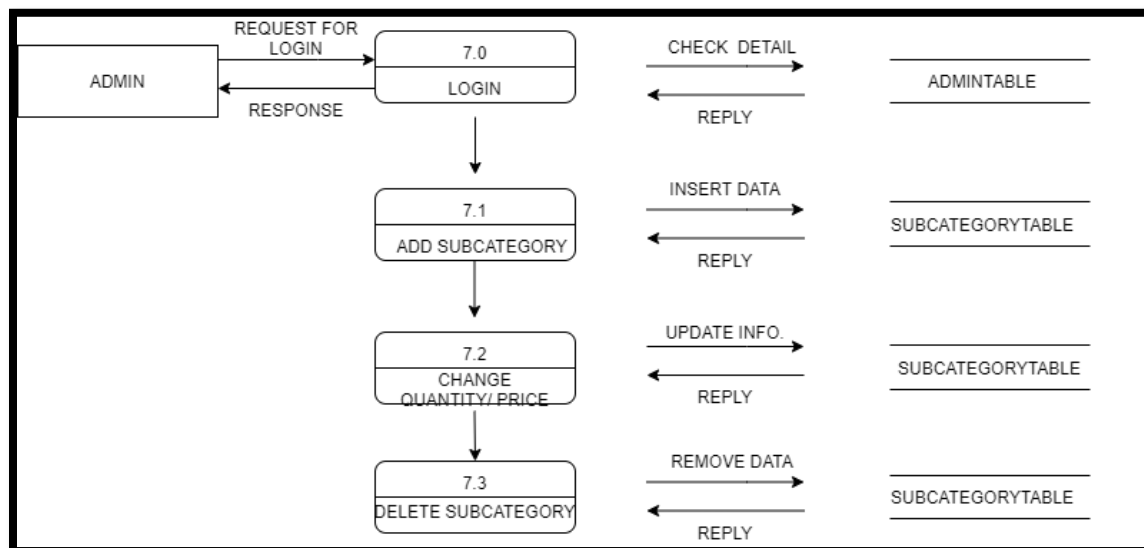
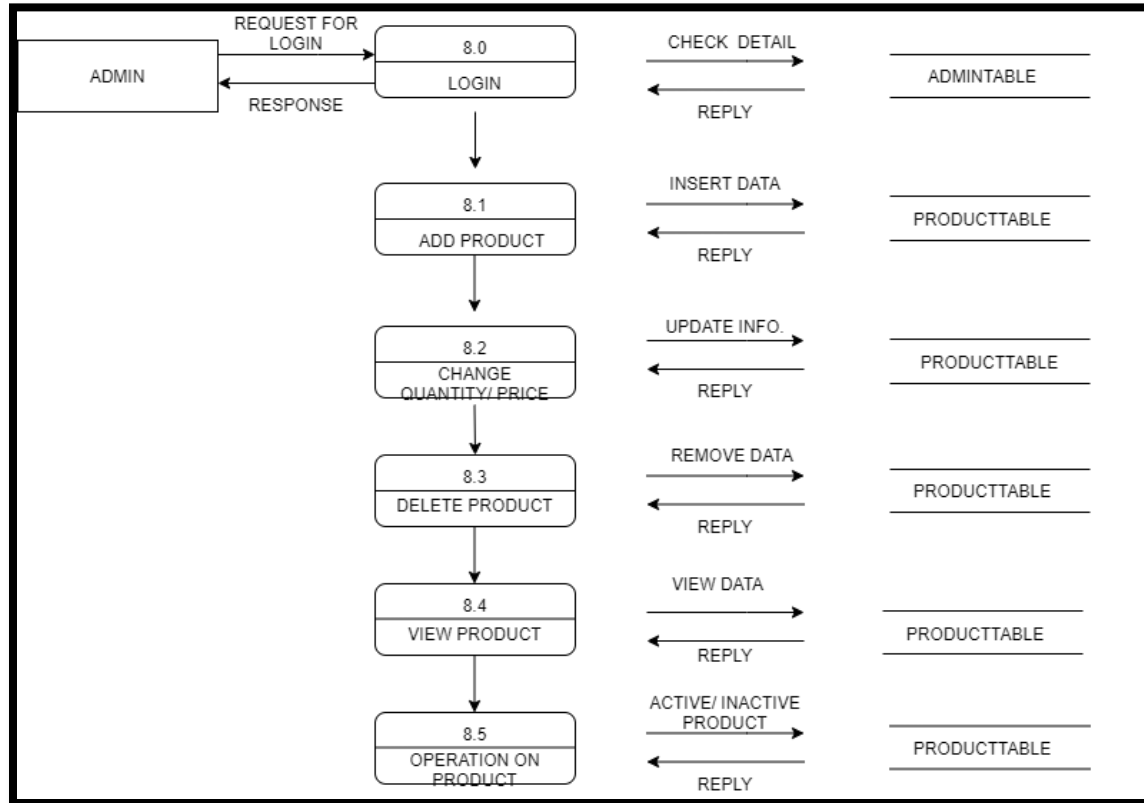
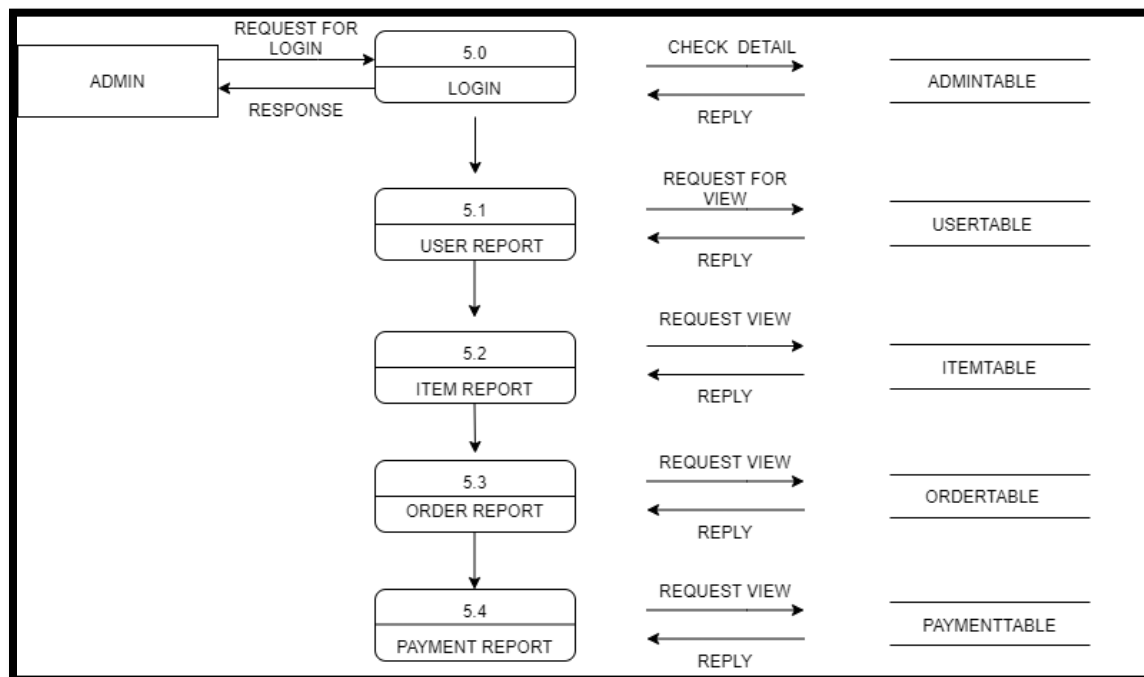
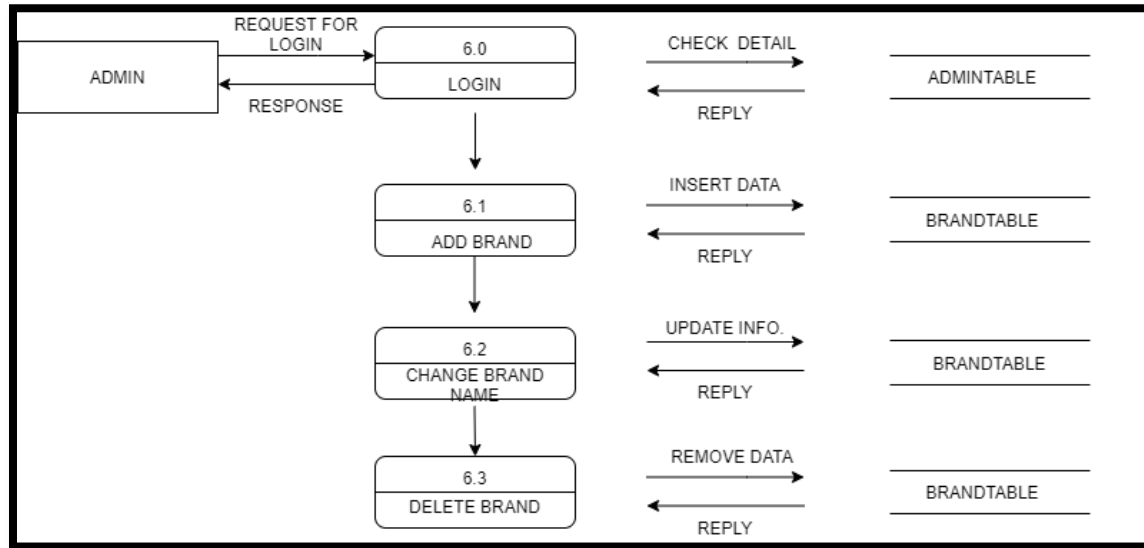
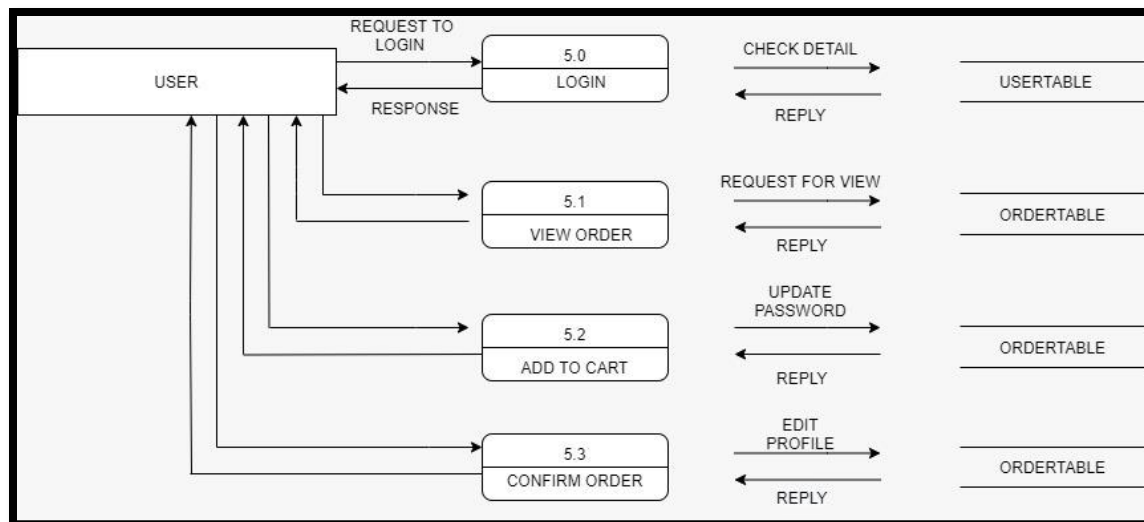
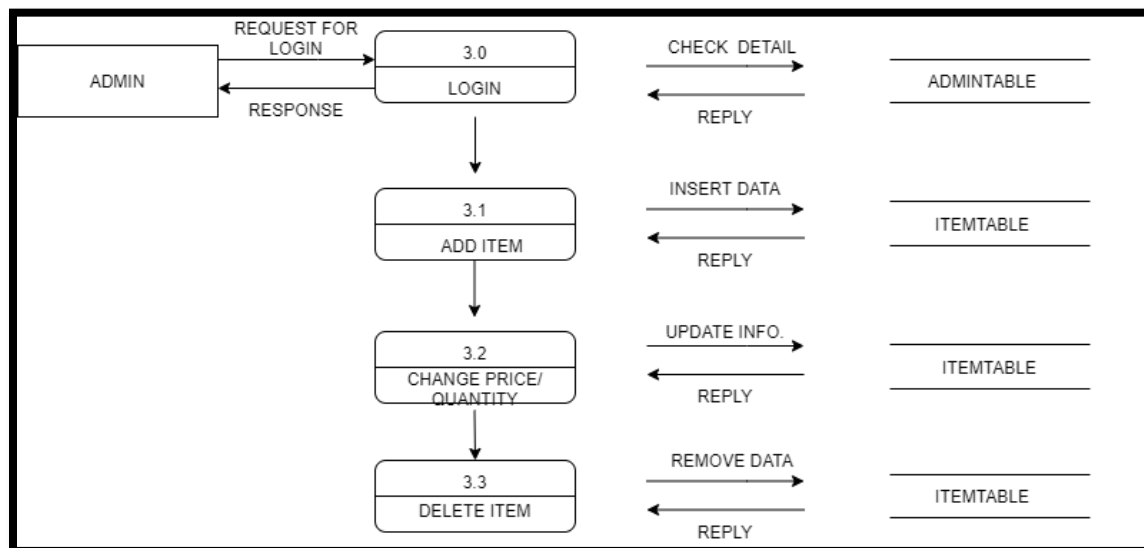
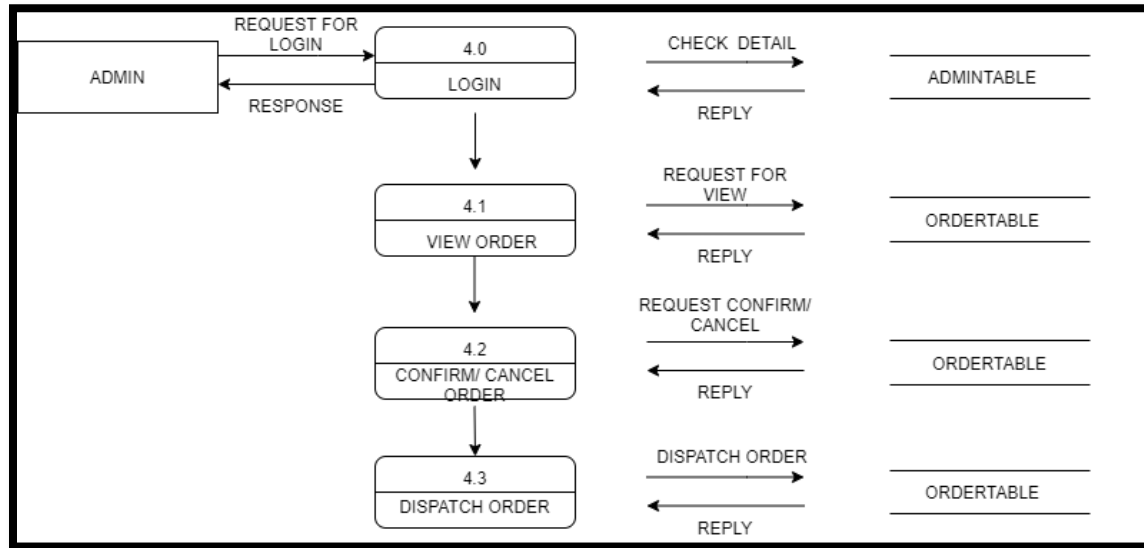


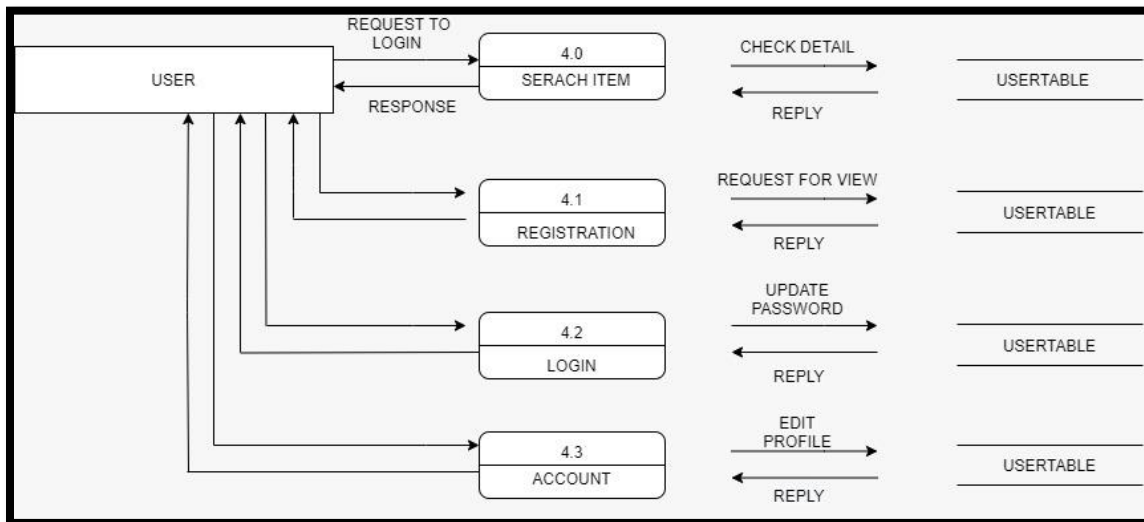
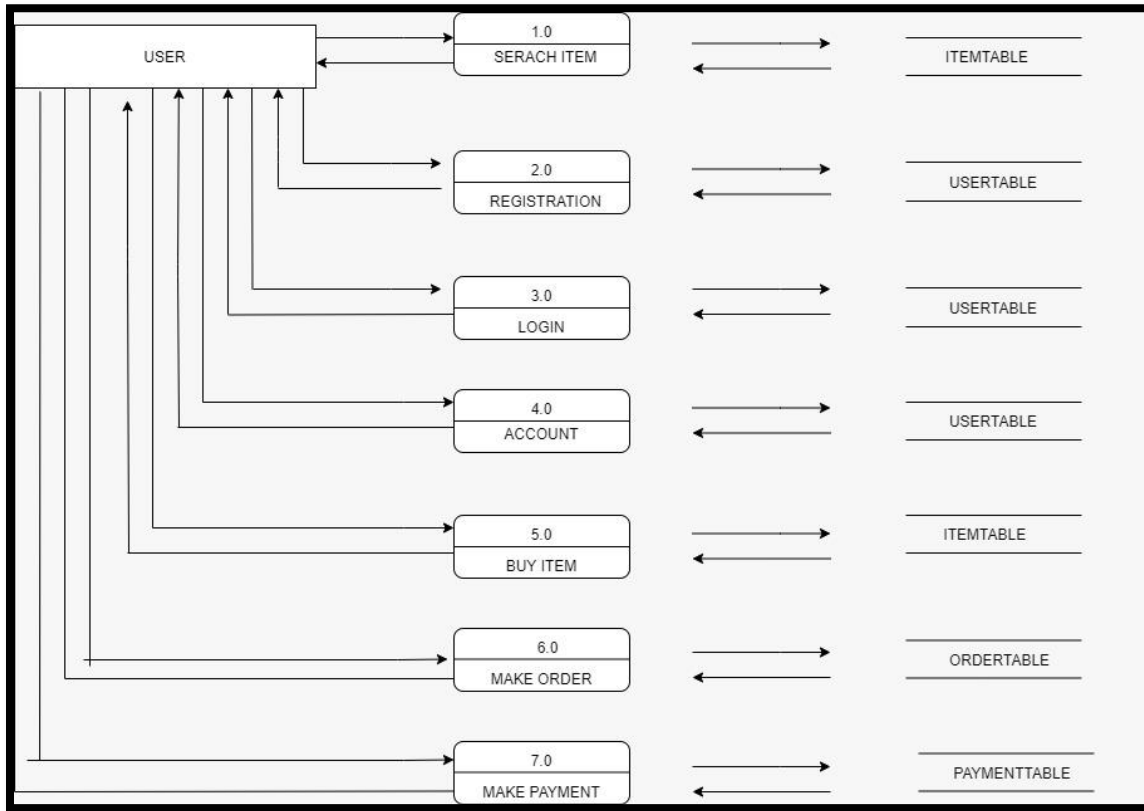
Fig. 2.4: Activity diagram for users purchasing products

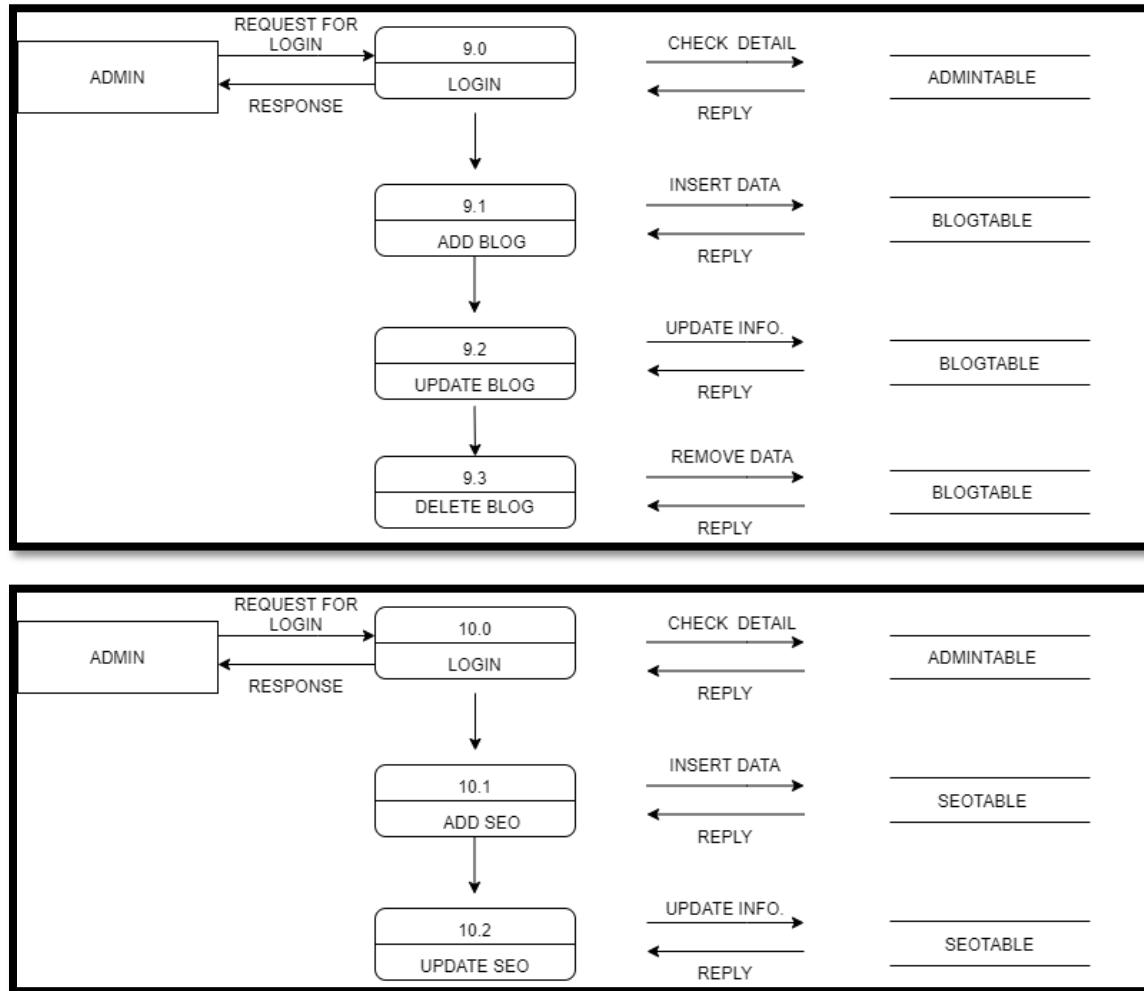
2.6 DFD Dataflow Diagram











2.7 Software requirements

For establishing the system we need these following software packages:

Web Server : **Apache** (Included in WAMP/XAMPP/LAMP)

DBMS : **MySQL** (Included in WAMP/XAMPP/LAMP)

Server-side Language : **PHP** (Included in WAMP/XAMPP/LAMP)

PHP Dependency Manager : **Composer** (<https://www.getcomposer.org>)

Version Control Software : **github**

IDE : Any text editor/IDE can be used as per user preference

2.8 Plugin Requirements

Stripe : API for payment (debit/credit card authenticity) validation.

Google : API for login.

Facebook : API for login.

Laravel Throttle : for limiting too many login attempts at once. (by Graham Campbell)

Laravel Invoices : for generating invoice.

Bum bum99 : for managing shopping cart.

AddThis : for sharing product.

Facebook Comment : for product review

Mailtrap : for sending mails in system.

2.9 Summary

This chapter discusses the basic structural and supplemental requirements for the application using software requirement analysis, sets the plot for a better understanding of the following chapters.

Chapter 3

Application Functionalities

The application offers diversified tasks for the users, different user groups carry out different tasks. From searching to purchasing products, from adding products to removing products- a moderately wide range of tasks are performed by subscribers and office staff. In this section we try to mention the primary functionalities of the web application.

3.1 User Registration and Verification

The website “ClickToCart” will let “guests” browse and search through the product details and add the products they want to purchase to the cart, but to access the cart and advance through the process, they have to login, if not- register on the website by verifying their e-mails.

The registration page will take users’ Full Name, Phone number, a unique E-Mail id, and confirmed password initially.

A confirmation e-mail is sent to the user-given e-mail address with a **randomly generated 40 character unique confirmation code**. The uniqueness is maintained by encrypting the current UNIX time using **SHA1** hash function.

User is kept inactive (email_verified = 0), hence unable to carry out subscriber tasks until the e-mail verification is confirmed.

After the registration is completed, the user can access the cart, edit their profiles, contact the customer support and advance through the purchase options.

3.2 Throttled Login

Limiting too many login attempts at once and hence negating the chance of getting hacked by any kind of brute-force attacks is a primary security feature an application should have. To provide this security feature we used **Laravel Throttle** plugin to make sure the application can record the false-requesting IP and block it from making farther requests for a given amount of time to prevent potential attacks.

3.3 Product Searching

Users can search for products in a more convenient way through features like-

□ Auto-Suggestion

Our application's search bar provide the users with auto-suggestions by matching the keywords to any product related info and shows it below the search bar. This piece of AJAX functionality is managed completely with plain and Javascript code. No jQuery or any other library was used for this part of the functionality to maintain simplicity.

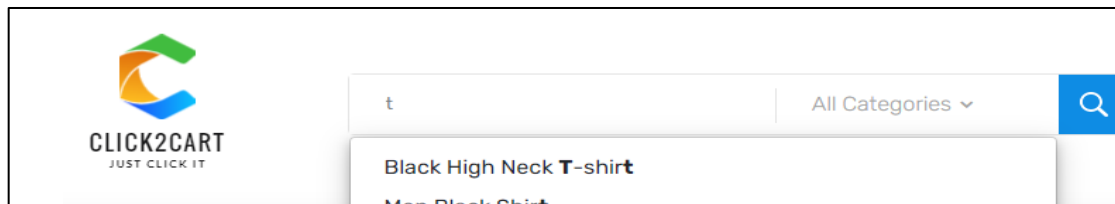


Fig. 3.1: Autocomplete suggestion

□ Sort Search Result

Users can sort the search result to their likings, sorting can be done considering products'- ■

- Name (Ascending/Descending)
- Price (High to Low/ Low to High)
- Categories
- Brand Names

3.4 Cart Functionalities

ClickToCart's cart system provides convenient features for users to help product purchasing more interactive.

□ Add to Cart

Users add their to-be purchased products to the cart, either from the partial views (e.g., Hotdeals section, Trending products section etc.) or from the product details view. They can select their desired quantity to be added to the cart but not more than the stock permits. This stock limit validation is done through multiple stages, which is discussed later in the “Application Implementation” part of this paper.

□ Addition, Deduction& Removal On-The-Go

After a user adds a certain number of products to a cart, he/she might come to a decision to add another piece of the product to the cart, or deduct the product by one, or the choice might even be to remove the product from the cart. This fine-tuning feature is implemented on the cart to increase customer satisfaction.

□ Product Count and Total Price

Throughout the whole browsing session of a user the total number of products added to the cart and accumulated price is shown beside the cart's button. This feature can be useful for users to get a better estimation of their purchase status.

❑ Delete Cart

Cart deletion can be performed to discard a cart full of products and continue with a new one.

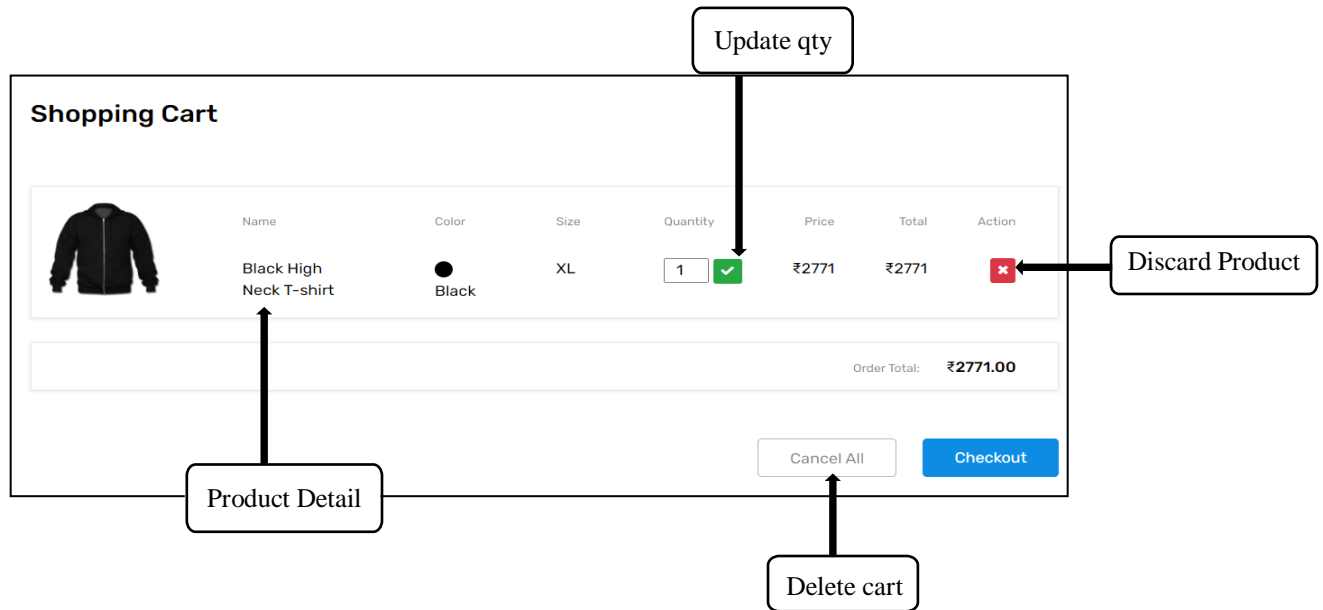


Fig. 3.2: Cart Functionalities

3.5 User Profile

Before a user can advance through to the checkout stage, the user's profile containing related advanced info(e.g., address, name, phone number etc.) must be submitted. The idea is to make sure of the availability of shipping address, phone number and such user contact-info beforehand, so the purchased products does not go to a wrong address. The profile info can later be viewed and edited from going to the "Edit profile" tab(/profile url) on the dropdown menu.

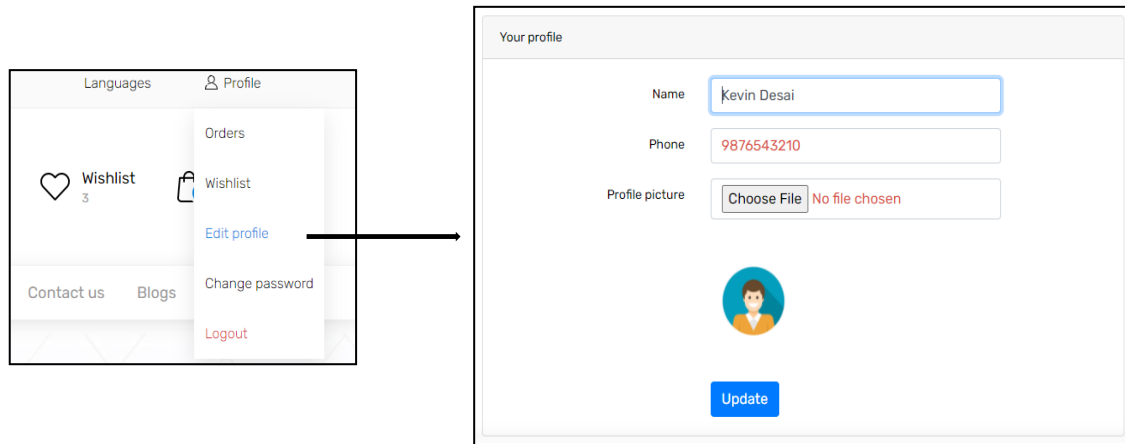


Fig. 3.3: View of a User's Profile

3.6 Checkout

Users can checkout only using valid payment cards. The validity and authentication is managed by **Stripe**- a well renowned, accredited API for handling payment procedures globally, and supporting user with by providing convenience.

3.7 Receipt

A printable receipt is given to the user after a purchase is successfully completed. These receipts can the viewed later by the user.



RECEIPT

Tax Invoice/Bill of Supply/Cash Memo

(Original for Recipient)

Invoice date: **07 June 2021**

Seller

ClickToCart Seller Pvt. Ltd.

Phone: 1800 3000 9009

GST Registration No:
27AAQCS4259Q1ZA

PAN No: AAQCS4259Q

Buyer

Kevin Desai

Phone: +919876543210

Shipping Details

Vinit Haresh Shah

Phone: +918866838023

Email: vinitshah2001@gmail.com

Address: 802/RatnaRekha apartment near
Gandhi Smruti hall, Nanpura, Surat-
395001

Country: India

State: Gujarat

City: Surat

Description	Qty	Price	Sub total
Black High Neck T-shirt	1	₹2771	₹2771
Total discount			₹138.55
Total taxes			₹50
Total shipping			₹120
Total amount			₹2802.45

Hey Kevin Desai thanks for shopping at ClickToCart Services Pvt. Ltd.

Fig. 3.4: A Purchase Receipt

3.8 User Review

Users can give their review on products, which can be removed later by the users at will.

3.9 Product Details

Product detail page shows a products detailed information about stock, price, brand, category etc. Stock interactively reduces/increases on the screen as the users purchase products or return them. Product details page accommodates a number of features in a packed space.

3.10 Customer Support

User support is managed by using an effective plug-in named “Mailtrap”. This communicative feature lets customers enquire or complain about products, discuss refund issues or give feedbacks about the service.

- Users create ticket about their enquiry
- Staff respond to the open tickets and marks as completed
-

3.11 Product Management

- Products can be added, deleted, edited and activated/deactivated by admins and managers
- Categories, Subcategories and Brands are to be created by the staff
- An useful feature is **addition of new brands on-the-go** (non-redundant approach)

3.12 User Management

Admins can add,delete and edit users manually, and assign or change roles of existing users. User data manipulation is not accessible to the managers.

3.13 Summary

An overview of mentionable key functionalities of the application has been shown in the chapter. For better understanding, figures and algorithms have been shown in the chapter also. Following chapters discuss the implementation and technical procedures of these functionalities.

Chapter 4

Implementation Plan

Before we move on to the next section where the technical implementations of our project is discussed, we would like to discuss some of the mandatory to know key-terms first.

Both Laravel, and PHP are products of open-source projects. Hundreds of skillful contributors are constantly working on both of them to make them more competent, more secure. In that sense, these two make the perfect, powerful duo to be used to make powerful, robust web applications.

We used this open-source duo to build our project along with HTML, CSS & Javascript. We used **bootstrap 4**, a web-styling framework for front-end designing.

Apache serves as the local-host, **MySQL** is the **DB Management System**. **PHP** is the serverside language. A package manager **Node.js** is used to manage gulp elixir.

4.1 A Quick Glance at Laravel Framework

Laravel is an open-source new generation web framework that uses a vast number of advanced features to make web application development secure, and time-saving for developers. Implementation of **MVC** pattern gives the developers a level ground to start their app development in a guided sense.

4.1.1 Key Terms:

- **MVC:** MVC is abbreviation for **Model-View-Controller** pattern in software development. Laravel framework uses this highly efficient pattern in the process of web development.
- **ORM:** ORM is short form of **Object Relational Mapping**. ORM Helps make an entity with unsupported data-type into an “Object”, which can be later manipulated using **OOP** (Object Oriented Programming) concept.
- **Eloquent:** Laravel’s eloquent simplifies the task of information retrieval by using “Models” to access the respective data relations. Hence simple functions can be implemented on the
- **Migrations:** Migrations are used for creating, deleting, and modifying databases and relations. The key convenience of migrations is that, the databases can be rolled back and forth of its state (e.g., remove a column that has just been inserted, rename a column back to its previous name) during development process at ease. Large chunks of data can be “seeded” into the database with significantly less effort.
- **Middleware:** Middlewares work as passages through which view requests go through to the controller to be manipulated. In a sense a “Middleware” controls the viewcontroller communication.
- **Service Provider:** Service providers are the start-up entities those register event listeners, middleware, routes and other variable at boot level.
- **.blade:** Naming convention of Laravel views follows adding a “.blade.php” extension (e.g., index.blade.php) at the end of the view filename.
- **Route:** All the existing routes in the web application must be declared in the “web.php” file inside the “Route” folder.

4.2 Project Coordination

We managed the coordination of the tasks for our project by constantly keeping each-other updated about the current events. We kept logs about the to-do tasks regularly and separated the task of implementing different modules between us. The coordination was mostly done using online chat sessions and/or screen-sharing applications.

4.2.1 Version controlling

We used github for our application's version controlling. Whenever a permanent change was done the files were pushed to the main repository of github where all our files were kept for better coordination.

4.2.2 Merging

Minor mergers were done by sharing particular files or through updating github repository. Major mergers were done however face-to-face under careful watch of our team.

4.3 Summary

Planning the implementation procedures and maintaining cooperation is a key task to be sorted out before going into project development. Proper cooperativeness leads to smooth mergers and easier issue-solving for a project.

Chapter 5

Technical Procedures

For the technical part of this project the issues we considered with utmost priority were-

- i. Ease of use
- ii. Secured server-side validation
- iii. Redundancy free database

5.1 Ease of Use

Maintaining usability of a website may seem negligible if not much thought is given into the matter, but if we try to look at it carefully, it becomes pretty clear that, this is one of the, if not, the only significant quality an application should have. It is because of the fact that, applications are built for the users particularly. If they do not feel comfortable using the application, then the sole purpose of building one fails. We took careful notes, and tried to build the application as much easy to use as we could within our extent.

Few features those were included to support our goal-

- Auto-suggestion in search
- Search result sorting

- Consistent fail-safe pop-ups
- Effective redirection to proper routes
- User-staff communication (Customer Support)

Fail-safe manner of the interactions makes sure that the users navigate through their browsing sessions inherently.

5.1.1 Auto-suggestions

Suggestions appear below the search bar as users start typing to search for a product. AJAX functionality manages to send the request to the controller instantly as the letters are being typed, the controller searches for products with the matching keyword and returns the data to the view to be received by the AJAX code. This listening process activates as soon as a user selects the search bar.

5.1.2 Search Result Sorting

Search results are primarily shown to the user in an alphabetical order. But those can be sorted later according to the users' needs. Sorting by Price both high-to-low and low-to-high are two useful options.

5.2 Server-side Validations

Proper server side validations are performed before fetching any data from the database or storing into it. The significance of server side validation is that even if someone manages to inject HTML codes through to the server, the server-side validation will check if the request is a legal/ non-violating one before the request actually reaches the database. This way any kind of harm to the database can be prevented effectively.

5.2.1 During Registration

- During the registration process users are prompted to use unique e-mail addresses. The process of checking the uniqueness of the user given e-mail is performed through server-side validation. Inside the file “RegisterController.php”- the controller that manages the user registration process, checks the validity of the uniqueness of the given e-mail.⁽³⁾ To make the request posting safe, a CSRF token is sent with the form request to the server.
- Password format and the correctness of the password is checked and confirmed in the controller.
- Registering users are instantly assigned the “Subscriber” role to their account.
- The confirmation code is generated by encrypting the current UNIX time and re-shuffling it before sending to the user’s e-mail address.

5.2.2 In Cart Functionality

Users can conveniently add products to the cart and deduct or remove them at will. This process of addition and subtraction may seem very simple but in reality, even a tiny little loophole can break the database during this process.

5.2.2(a) Problematic Scenarios

If proper server-side validation were non-existent in a project, it does not take a genius to tell that the system would break any moment after it launches, even if it manages not to fail to launch in the first place. But for the sake of example, let us assume that, the server-side validation were not thought very thoroughly

and were weak in implementation in the project. In that scenario, any tech-savvy user having intermediate knowledge about web can inject unreal and/or breakpoint values to manipulate the database and hence harming the transactions. I.e.,

- Create multiple cart images and send the requests altogether.
- Inject 0 to multiply with the total price and checking out with a huge amount of products
- Insert a garbage value to the cart quantity so that, on checkout it breaks the product stock in database

5.2.2(b) Prevention

To prevent these kind of failing scenarios firstly, we save our cart in the appointed session only. Users cannot use any other session to function as a cart. That is because the PHP sessions are cross-matched with a given id from the server, stored in the cookies, which is a hashed string and too hard to break; even it were to be broken, the session will fail to cross-match with the server and become invalid instantly. This process prevents the first problem from arising ever.

To prevent users from adding any invalid number of product that can dismantle the database, primarily the maximum number of products is set in the view file according to the product's available stock. The minimum number is set to 1 if the product is in stock. Both are set to 0 if the product is removed from cart.

As we discussed before only this setup alone is not sufficient for preventing the damage from happening. So the server-side controller takes out the cart data and checks if any invalid amount of quantity has been requested. Attempts to insert 0 as product quantity is checked and restricted, attempts to add negative values are also discarded.

One particular feature to be pointed here is that, though we are letting users add, remove products to/from the cart and also restricting them from

inserting invalid values, the actual database is nearly untouched. That is, the actual stock of products does not get affected until the user checks-out the cart.

5.2.3 In Checkout

As the user proceeds forward to checkout, the controllers will look for users advanced information. If the info are not available, the user cannot go forward, the user must fill out the information form to continue the purchase session. If advance info pre-existed, the user can choose to either proceed instantly or edit the appropriate info to continue. This process is also validated server-side. After the order is placed, the current stock is checked accessing the database before forwarding, if the product has been reduced meanwhile, the user is redirected back or else the checkout process forwards and updates the database accordingly.

There are quiet a few more places here and there inside the controllers to ensure the data validation, such as the 'ReviewController.php' checks if the review shown on a product are of the current user, if they are not, they cannot be deleted by the current user. Another mentionable one is inside the 'ProductController.php', where the staff can choose to add a new brand for the product on the go and it checks server-side if the requested brand exists or not (discussed later in 'Redundancy Free Database').Concisely, we considered any visible/potential loopholes as threats to the system and put efforts within our extent to remove them.

5.3 Redundancy Free Database

Throughout the whole project we carefully tried to maintain the non-redundant approach for our database system. From storing users' advanced

info to not updating the product stock until the checkout is completed, less-to-no redundancy was one of our top priorities.

5.3.1 In Product Management

During addition of new products to the database (by office staff) there might come a scenario where a product of the same name and same brand exists but the staff failed to notice that beforehand. In such case, adding a product having same name results in data redundancy.

To prevent this kind of potential record duplication, the application checks if a product having the same name and the same brand pre-exists in the database. If a matching product is pre-existent, the stock number for the product is increased adding new stock quantity to the previous one and price for the product keeps the lower amount as the updated price.

Another quite similar a scenario arises during addition of a new product, when a new brand name for the product has to be entered but the database does not contain the brand name. In such case the staff can simply insert the name of the brand instead of an existing one. Now, the controller receives the request, checks for a match if the brand pre-exists. If it does, the **existing brand_id** is assigned to the product. If the brand is a new one, the controllers inserts it into the database giving it a new id and saves it after assigning it to the product. This way any potential redundancy arising from duplicate data existence is avoided.

Note that, product's 'Categories' and 'Brands' are kept in different tables for the convenience of managing them in advanced fashion (e.g., use in navigation bars, subcategorize), or manipulating later in time by the staff. This keeps the option of adding extra attributes to categories/brands open for future use.

5.3.2 In Advanced Info for Users

For the advanced user info, we reduced redundancy by keeping the table corresponding to advanced user info ('advanceds') separated from the main users table. This way the users do not necessarily have to insert their

advanced info until they try to purchase something; so the database and the table is more likely kept relatively smaller. This approach also helps to minimize the database access cost when a user performs regular tasks like adding feedbacks, contacting customer support, watching histories etc., where the only entity the user needs is the ID to get access to those task initially. Had the other data been in the same table, more cost would have caused due to fetching bigger data chunks. On deletion of any user the advanced info also gets deleted.

5.3.3In Receipts and Orders

This is another section where we maintained less redundancy according to the scenario of the two entities. An 'Order' will keep the ordering user's id (user_id), the items bought (cart), the payment confirmation id (payment_id) and the status of the order (i.e., pending, delivered, returned) to keep track of the actual order.

The 'Receipt' will keep the user's id (user_id), the order's id (order_id), shipping address and phone number of the user. This way the products related to the receipt will be fetched from the order to show on view. One important point to notice here is the inclusion of shipping address and phone number in the 'receipts' table. Primarily it may seem that these two columns are used here redundantly, when it can simply fetch data from the user's advanced info relation. But if a user moves from the place he/she ordered from to another place and updates the current address in the 'advanced' table, the actual address where the shipping was made gets lost. Same happens for the phone number too. That is the reason why these two columns were kept with receipts intentionally. Any other visible redundancy were avoided.

Redundancy free database is a highly desirable characteristic for an application. We tried to keep our database as less redundant as possible for a better system.

5.4 Summary

Key background features and their implementation procedures have been discussed in the chapter. Ease-of-use, proper validation and the process of non-redundancy go hand in hand in an interactive project. Implementing them and making sure they all work together is a delicate task.

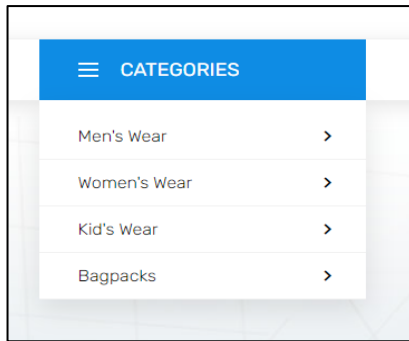
Chapter 6

View Optimization

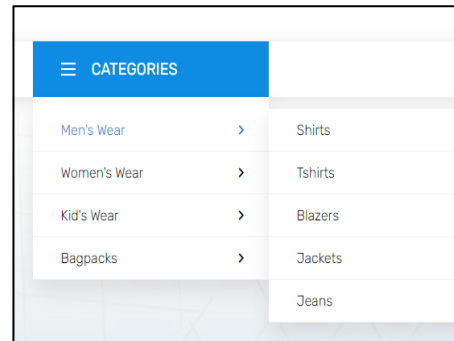
The UI (user interfaces), or ‘View’ as termed in MVC pattern is a significantly vital part of a website. That is because the users would likely have no interest in knowing what kind of features are used in the website, how fail-safe the website is or how less redundant the website’s database is. If the users do not like the view or if they feel uncomfortable using the website they would most likely lose their interest on the website. In this section we briefly mention a few of the procedures we undertook to optimize the website’s views.

6.1 Using Service Providers

Service providers helps load variables, routes, event listeners and such entities during the startup, before any views or controllers are loaded. We passed our frequently used models such as categories, brands to our main layout view using the ‘AppServiceProvide.php’ at the application’s booting time. The categories and brands are fetched from the database and sent to the main layout (app.blade.php) page during program initiation. This way, we maintain nonredundancy in our views too.



(a)



(b)

Fig. 6.1:Categories (a) and Subcategories (b) on navigation bar fetched from database.

6.2 View Stock Availability

We managed to show the users the stock availability in a virtual manner. Initially the stock is shown to the users according to the actual product stock that is derived from the database. If the product has stock, a green colored 'Available' text is shown beside the product's available quantity; if not, a red texted 'Out of Stock' is shown in the same place.

Stock : **Available**

(a)

Stock : **out of stock**

(b)

Fig. 6.2: Stock availability indicator

Now, if a user starts adding the product to the cart, the product's available quantity keeps reducing and shown accordingly to the user, and as long as the available quantity does not reach 0 the 'In Stock' text is kept shown. If the user adds all the available stock to the cart the text then shows 'Limit Reached' in red colored text. All of these mentioned functionalities are carried out without affecting the database

right away as we discussed in the earlier section, which adds a sense of virtuality to the view manipulation.

6.3 Search Bar

The search bar is built as a partial layout with its full functionality for the website. This enables the search-bar to be extended anywhere in any page as an external module. To add the search bar to view using a single line is (e.g., `@extends('layouts.search')`) enough. This way, we avoided using the search bar on places where it is not necessary (e.g. receipt, invoice etc. views).

6.4 Summary

Laravel provides developers with a subtle view management system to avoid any unnecessary work procedure. Idea of extending a main layout, and/or partial layouts in views is an effective feature for reducing overhead; hence increasing the throughput quality.

Chapter 7

Data Dictionary

7.1 List of all tables

Sr No	Table name
1	Admin
2	Brands
3	Categories
4	Cities
5	Colours
6	Contact
7	Countries
8	Coupons
9	Order
10	Newsletter
11	Password resets
12	Order_details
13	Post
14	Post_category
15	Products
16	SEO
17	Settings
18	Shipping
19	Site settings
20	Sizes
21	Subcategories
22	Users
23	Wishlist

7.2 Table's detail view

TABLE NAME: ADMIN

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Name	varchar	191	No
Phone	varchar	191	No
Email	varchar	191	No
Avatar	varchar	500	No
Email_verified_at	timestamp		No
Password	varchar	191	No
Remember_token	varchar	100	Yes
Reports	varchar	255	No
Categories	varchar	255	No
Products	varchar	255	No
Orders	varchar	255	No
Return_orders	varchar	255	No

Coupons	Varchar	255	No
blogs	varchar	255	No

TABLE NAME: BRANDS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Brand_name	varchar	191	No
Brand_logo	varchar	191	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: CATEGORIES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Category_name	varchar	191	No

Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: CITIES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	int	10	No
Name	varchar	191	No
State_id	int	11	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: COLOURS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No

Colour	varchar	191	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: CONTACT

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Name	varchar	191	No
Phone	varchar	191	No
Email	varchar	191	No
Message	text		No
Subject	varchar	255	Yes
Reply	text		Yes
Status	int	255	0

Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: COUNTRIES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	int	10	No
name	varchar	191	No
Sort_name	varchar	191	No
Phone_code	varchar	191	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: COUPONS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
-------------	-----------	------	------

Id	bigint	20	No
Coupon	varchar	191	No
Image	varchar	500	No
Discount	varchar	191	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: ORDER

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
User_id	varchar	191	No
Payment_type	varchar	255	Yes
Payment_id	varchar	191	Yes
Paying_amount	varchar	191	Yes
Balance_transaction	varchar	191	Yes
Stripe_order_id	varchar	191	Yes

Subtotal	varchar	191	Yes
Shipping	Varchar	191	Yes
Vat	Varchar	191	Yes
Total	Varchar	191	Yes
Status	Varchar	191	0
Return_order	Varchar	255	0
Created_at	Timestamp		Yes
Updated_at	timestamp		Yes

TABLE NAME: NEWSLETTER

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Email	varchar	191	No
Created_at	Timestamp		yes
Updated_at	timestamp		Yes

TABLE NAME: PASSWORD_RESETS

Column Name	Data Type	Size	Null
Email	varchar	191	No
Token	Varchar	191	No
Created_at	Timestamp		Yes

TABLE NAME: ORDER_DETAILS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Order_id	Int	11	No
Product_id	varchar	191	Yes
Product_name	varchar	191	Yes

Color	varchar	191	Yes
Size	varchar	191	Yes
Quantity	varchar	191	Yes
Single_price	varchar	191	Yes
Total_price	Varchar	191	Yes
Created_at	Timestamp		Yes
Updated_at	timestamp		Yes

TABLE NAME: POSTS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Category_id	Int	11	No
Post_title_english	varchar	191	No
Post_title_hindi	varchar	191	No

Post_title_gujarati	varchar	191	No
Post_image	varchar	191	No
Details_english	Text	191	No
Details_hindi	Text	191	No
Details_gujarati	Text	191	No
Created_at	Timestamp		Yes
Updated_at	timestamp		Yes

TABLE NAME: POST_CATEGORY

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Category_name_english	varchar	191	No
Category_name_hindi	varchar	191	No

Category_name_gujarati	varchar	191	No
Created_at	Timestamp		Yes
Updated_at	timestamp		Yes

TABLE NAME: PRODUCTS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
Category_id	Int	11	No
Subcategory_id	Int	11	Yes
Brand_id	Int	11	Yes
Product_name	varchar	191	No
Product_code	varchar	191	No

Product_quantity	varchar	191	No
Product_details	Longblob		Yes
Product_color	varchar	191	No
Product_size	varchar	191	No
Selling_price	Int	191	No
Discount_price	Int	191	Yes
Video_link	varchar	191	Yes
Main_slider	Int	11	Yes
Hot_deal	Int	11	Yes
Best_rated	Int	11	Yes
Mid_slider	Int	11	Yes
Hot_new	Int	11	Yes
Buyone_getone	Int	11	Yes
Trend	Int	11	Yes
Image_one	Varchar	191	Yes
Image_two	Varchar	191	Yes

Image_three	Varchar	191	Yes
Status	Int	11	Yes
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SEO

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Meta_title	Int	11	Yes
Meta_author	Int	11	Yes
Meta_tag	Varchar	191	Yes

Meta_description	Text		Yes
Google_analytics	Text		Yes
Bing_analytics	Text		Yes
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SETTINGS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Vat	Varchar	191	Yes
Shipping_charge	Varchar	191	Yes

Shopname	Varchar	191	Yes
Email	Varchar	191	Yes
Phone	varchar	191	Yes
Address	Varchar	191	Yes
Logo	Varchar	191	Yes
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SHIPPING

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Order_id	Varchar	191	No
Ship_name	Varchar	191	No

Ship_phone	Varchar	191	No
Ship_Email	Varchar	191	No
Ship_address	varchar	255	No
Ship_country	Varchar	255	Yes
Ship_state	Varchar	255	Yes
Ship_city	varchar	191	No
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SITESETTING

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Logo	Varchar	500	Yes
Favicon	Varchar	500	yes

Phone_one	Varchar	191	Yes
Phone_two	Varchar	191	Yes
Email	varchar	191	Yes
Company_name	Varchar	191	Yes
Company_address	Varchar	191	Yes
Facebook	varchar	191	Yes
Youtube	varchar	191	Yes
Instagram	varchar	191	Yes
Twitter	varchar	191	Yes
Disclaimer	longblob		Yes
Policy	longblob		Yes
Safe	longblob		Yes
Terms	longblob		Yes
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SIZES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Code	Varchar	191	No
Size	Varchar	191	No
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: STATES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Name	Varchar	191	No

Country_id	Int	11	No
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: SUBCATEGORIES

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	Bigint	20	No
Category_id	Int	11	No
Subcategory_name	Varchar	191	No
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: USERS

PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
-------------	-----------	------	------

Id	bigint	20	No
Name	Varchar	191	No
Phone	Varchar	191	Yes
Email	varchar	191	No
Avatar	varchar	255	Yes
Provider	varchar	255	Yes
Provider_id	varchar	255	Yes
Email_verified_at	Timestamp	191	Yes
Password	varchar	191	Yes
Remember_token	varchar	100	Yes
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

TABLE NAME: WISHLISTS

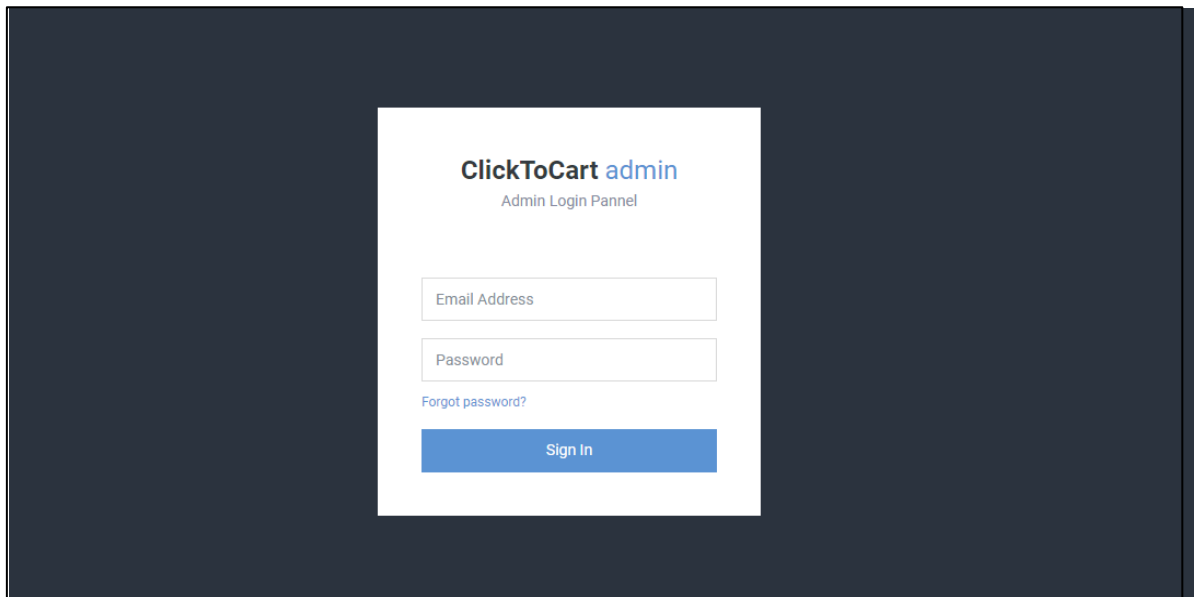
PRIMARY KEY: 'ID'

Column Name	Data Type	Size	Null
Id	bigint	20	No
User_id	Int	11	No
Provider_id	Int	11	No
Created_at	Timestamp		Yes
Updated_at	Timestamp		Yes

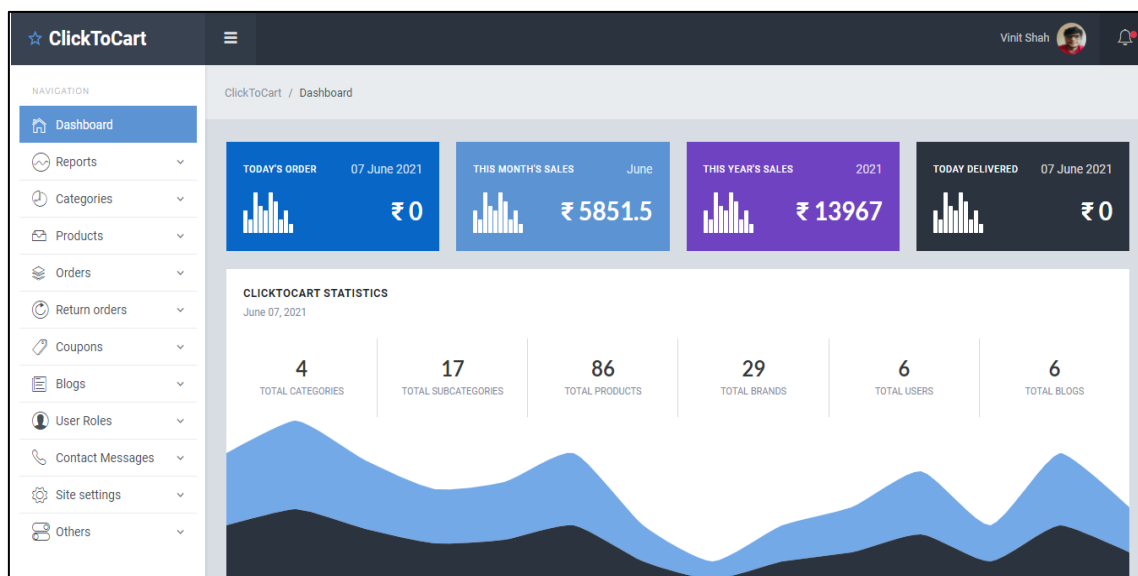
Chapter 8

User interface

8.1 Admin side



Admin login panel



Admin dashboard

ClickToCart

Vinit Shah

NAVIGATION

- Dashboard
- Reports
- Categories
- Products
- Orders
- Return orders
- Coupons
- Blogs
- User Roles
- Contact Messages
- Site settings
- Others

Edit profile

Email

admin@clicktocart.com

Name

Vinit Shah

Profile Photo

Choose file... **Browse**

Update **Close**

Edit Profile **Change Password** **Sign Out**

Admin edit profile

ClickToCart

Vinit Shah

NAVIGATION

- Dashboard
- Reports
- Categories
- Products
- Orders
- Return orders
- Coupons
- Blogs
- User Roles
- Contact Messages
- Site settings
- Others

ClickToCart / Change Password

Admin Change Password **← BACK TO DASHBOARD**

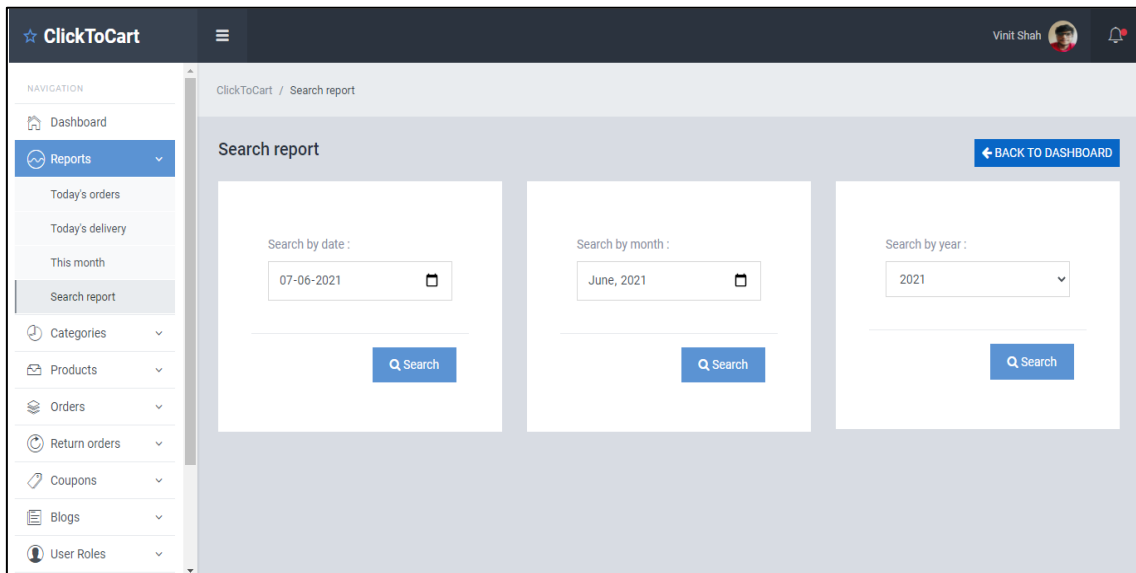
Old Password

New Password

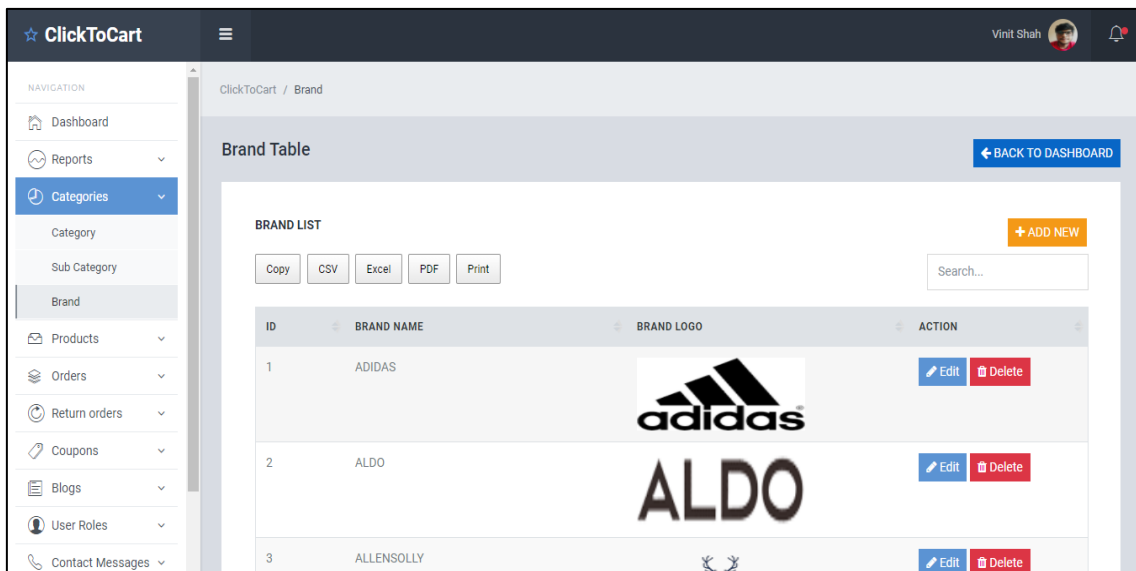
Confirm Password

Reset Password **Close**

Admin change password



Admin search report



Admin brands

ClickToCart

Vinit Shah

NAVIGATION

Dashboard

Reports

Categories

Products

Add Products

All Products

Orders

Return orders

Coupons

Blogs

User Roles

Contact Messages

Site settings

ClickToCart / Product

Product Table

PRODUCT LIST

Copy

CSV




Excel

PDF

Print

Search...

ADD NEW

SR	PRODUCT NAME	IMAGE	CATEGORY	BRAND	QUANTITY	STATUS	ACTION
1	double hoodie		Men's Wear	U.S. POLO	29	Active	<div></div> <div></div> <div></div> <div></div>
2	rip jeans 2		Men's Wear	U.S. POLO	5	Active	<div></div> <div></div> <div></div> <div></div>
3	clickto tshirt		Men's Wear	BEING HUMAN	11	Active	<div></div> <div></div> <div></div> <div></div>

Admin products

ClickToCart

Vinit Shah

NAVIGATION

Dashboard

Reports

Categories

Products

Orders

Return orders

Coupons

Blogs

User Roles

Contact Messages

Site settings

Others

ClickToCart / Coupon

Coupon Table

COUPON LIST

Copy

CSV



Excel

PDF

Print

Search...

ADD NEW

ID	COUPON CODE	IMAGE	DISCOUNT(%)	ACTION
1	SB15		5%	<div>Edit</div> <div>Delete</div>
2	HDFC9		9%	<div>Edit</div> <div>Delete</div>

Admin coupons

76

ClickToCart

Vinit Shah

NAVIGATION

Dashboard

Reports

Categories

Products

Orders

Return orders

Coupons

Blogs

Blog Category

Add Post

All Post

User Roles

Contact Messages

ClickToCart / Blog Post

Blog Post Table

BACK TO DASHBOARD

BLOG POST LIST

Copy

CSV

Excel

PDF

Print

Search...

ADD NEW

ID	POST TITLE	POST CATEGORY	IMAGE	ACTION
1	10 Lessons From Disney And Pixar Movies	Entertainment		<div>Edit</div> <div>Delete</div>
2	Tesla's Autopilot Can Be Tricked Into Working With...	Global		<div>Edit</div> <div>Delete</div>

Admin blogs

ClickToCart

Vinit Shah

NAVIGATION

Dashboard

Reports

Categories

Products

Orders

Return orders

Coupons

Blogs

User Roles

Create User

All users

Contact Messages

Site settings

ClickToCart / Admin

Admin Table

BACK TO DASHBOARD

ADMIN LIST

Copy

CSV

Excel

PDF

Print

Search...

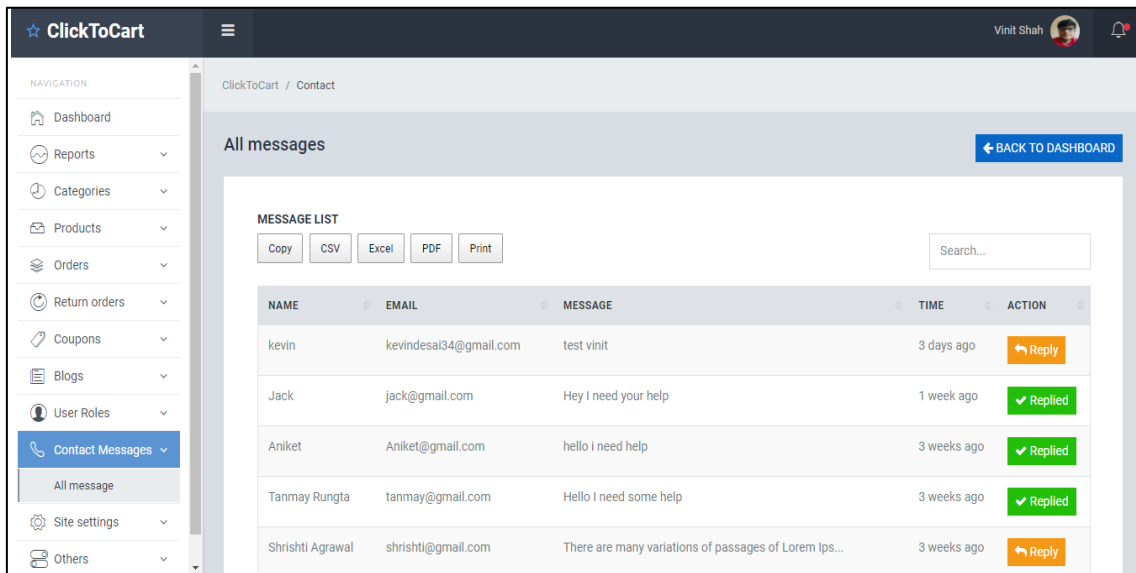
ADD NEW

NAME	PHONE	ACCESS	ACTION
Saroj Shah	+918866838023	<div>Coupons</div> <div>Contact messages</div>	<div>Edit</div> <div>Delete</div>
Divyesh Bhandari	+917990385903	<div>Reports</div> <div>Products</div> <div>Site settings</div> <div>Others</div>	<div>Edit</div> <div>Delete</div>
Shrishti Agrawal	+919898086962	<div>Categories</div> <div>Return orders</div> <div>User roles</div> <div>Site settings</div>	<div>Edit</div> <div>Delete</div>
Kevin Desai	+919558530802	<div>Reports</div> <div>Orders</div> <div>Coupons</div> <div>Contact messages</div> <div>Product comments</div> <div>Others</div>	<div>Edit</div> <div>Delete</div>

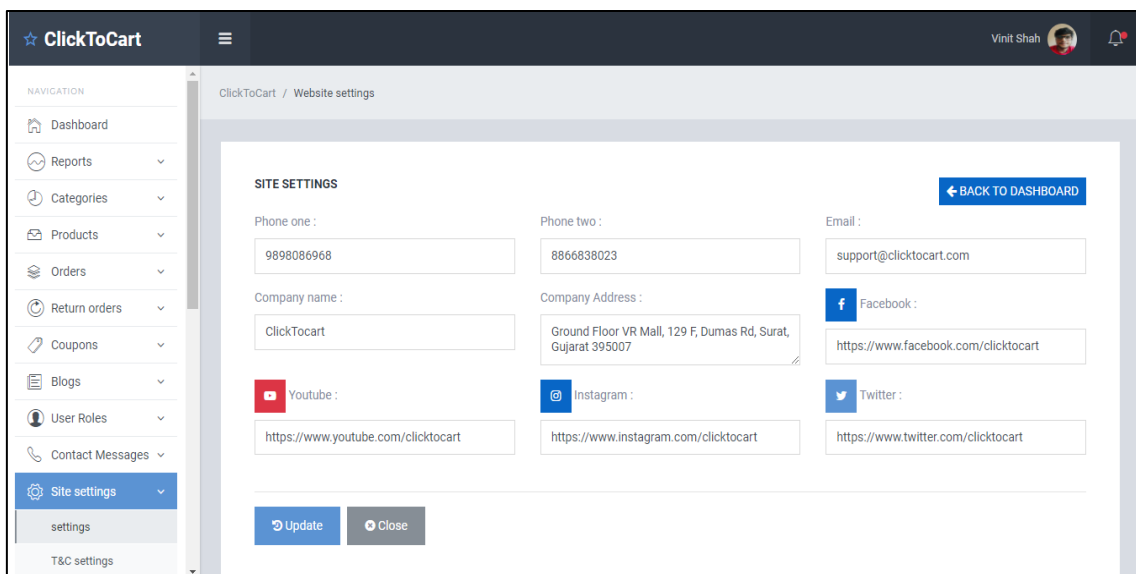
Showing 1 to 4 of 4 entries

Admin user roles

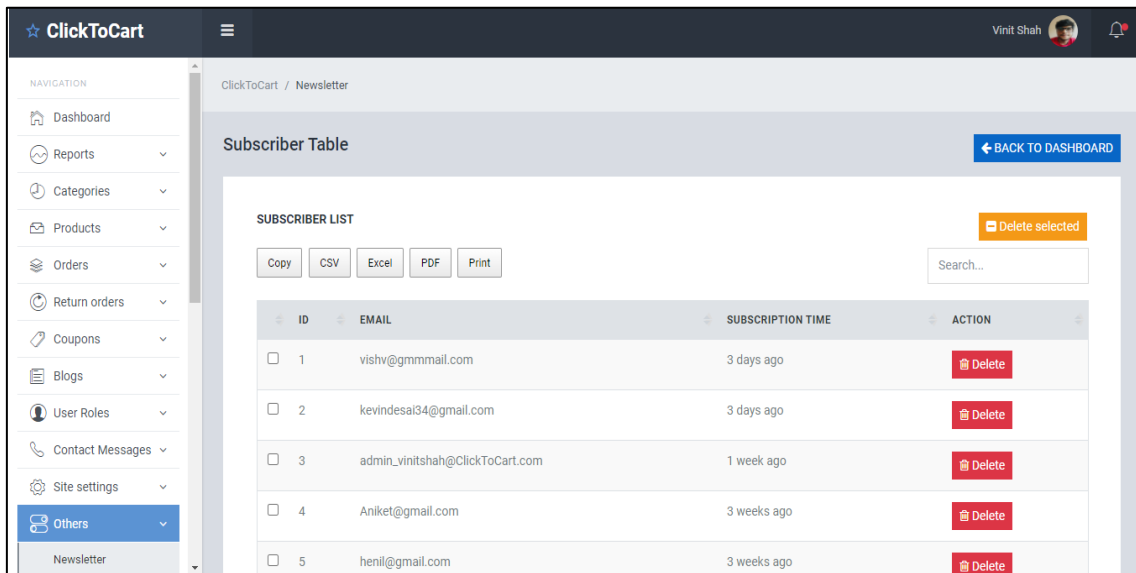
77



Admin contact messages

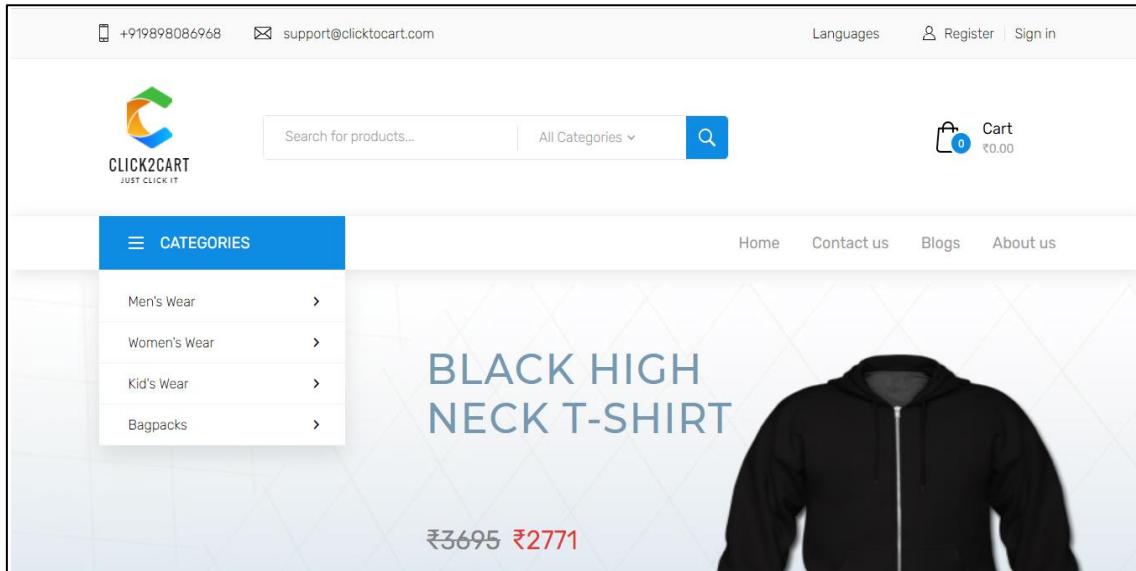


Admin site settings

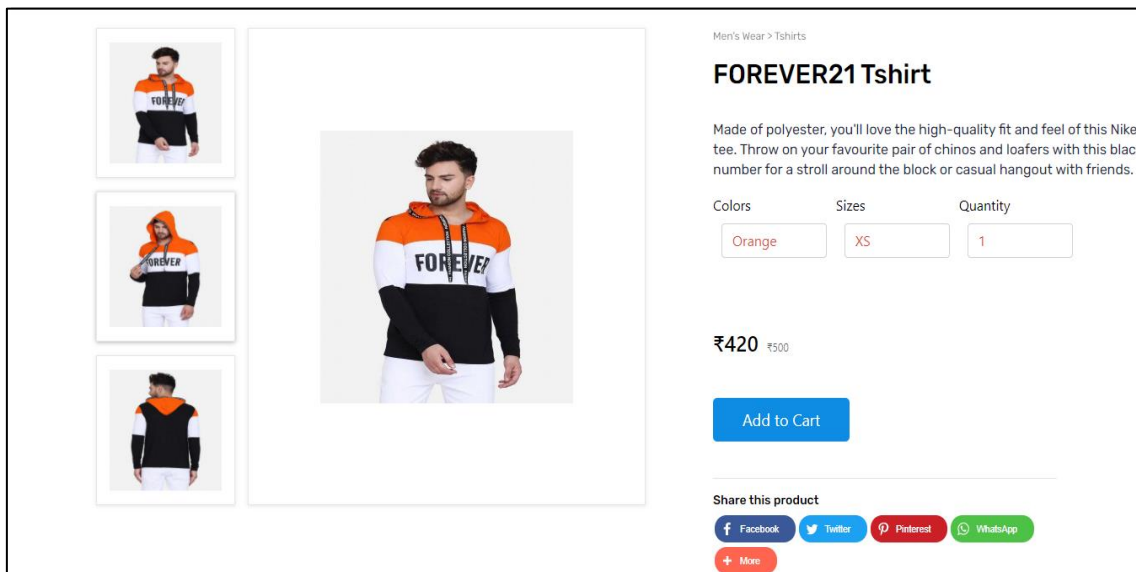


Admin newsletter

8.2 User side



User home page



User produt detail

Sign up

Full Name

Enter Your Name

Phone

Enter Your Phone Number

Email

Enter Your Email Address

Password

Enter Your Password

Confrim Password

Re-type Your Password

Register

[Already have an account ?](#)

User login panel

Sign in

Email or Phone

Password

Login

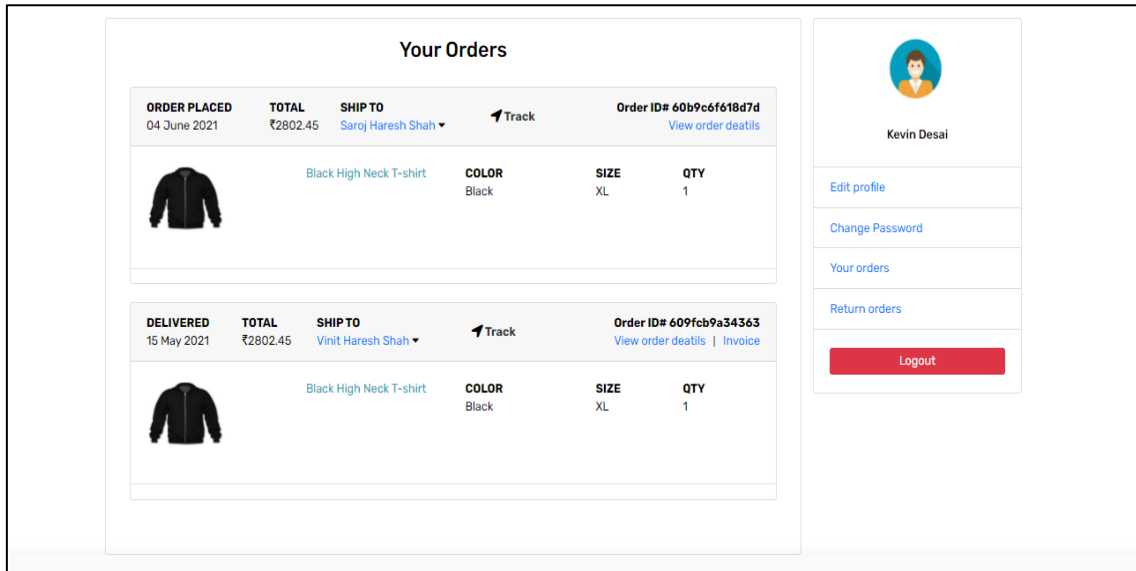
[Create new account](#)

[I forgot my password](#)

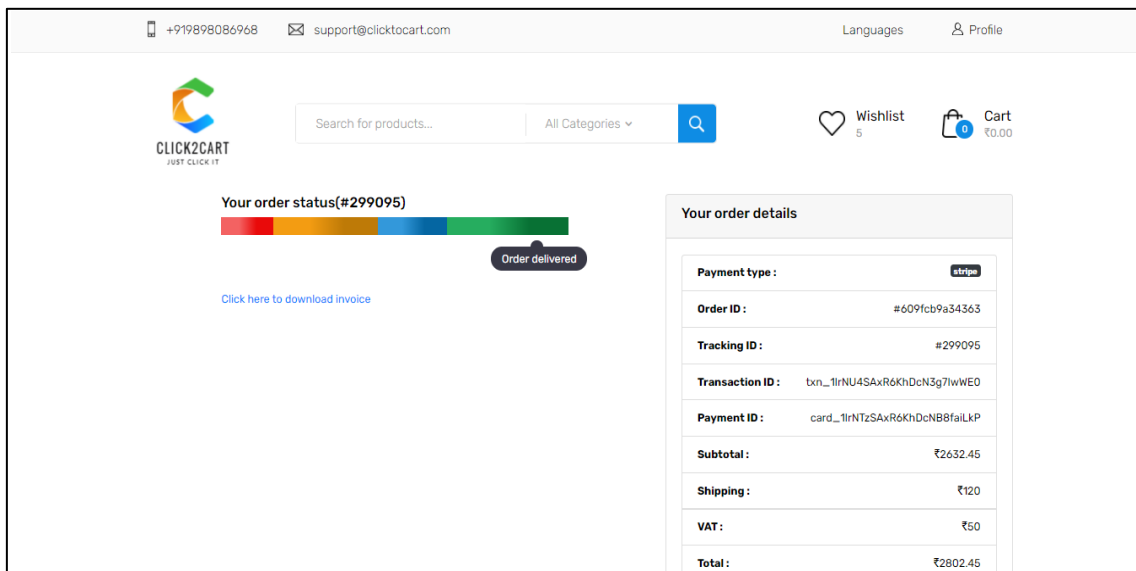
Continue with Facebook

Continue with Google

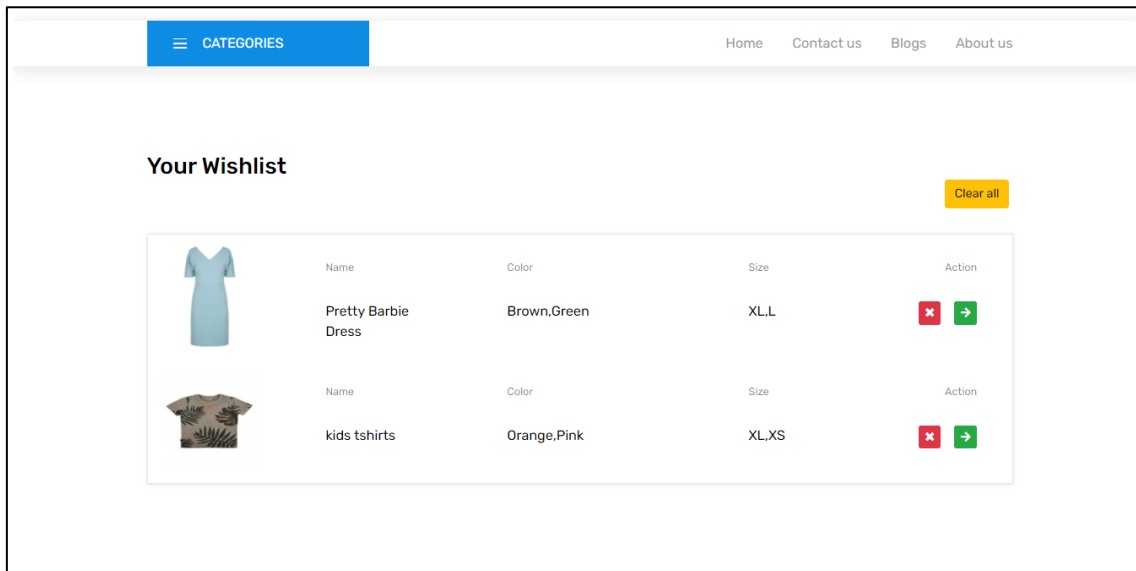
User register panel



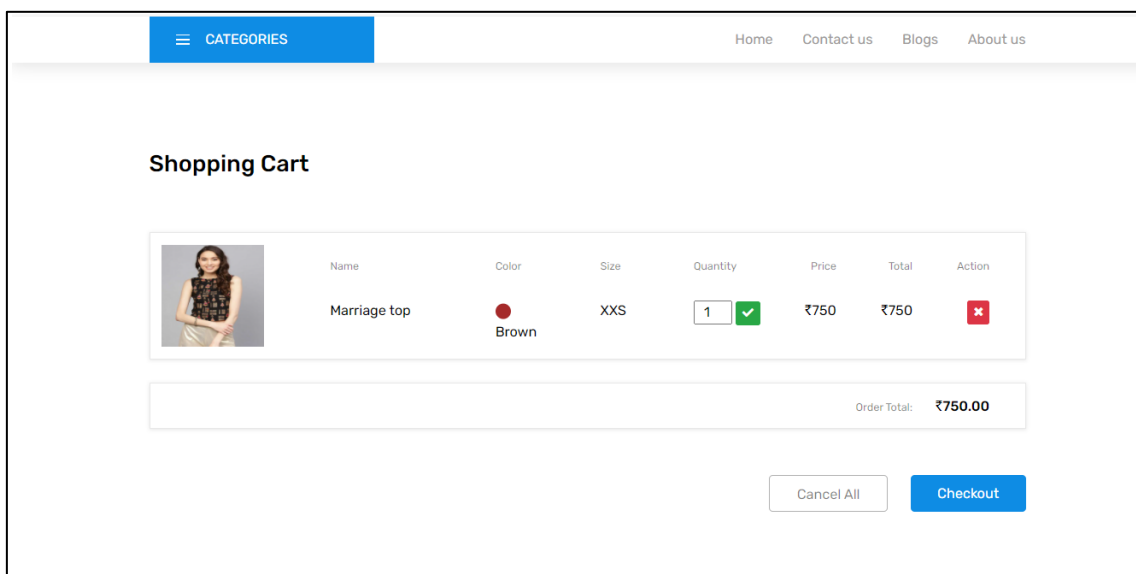
User order section



User order tracking



User wishlist



User cart

Chapter 9

Conclusion

9.1 Learnings

There are a few learnings worth mentioning those we learned working on the project. These realizations would surely help if kept in mind for future projects.

- Building a website is a continuous process
- Need for new modules will always persist
- Bug-free modules are products of countless re-testing
- Careful time-management is very crucial
- Regular team meeting during development is a must
- Maintaining good coding practices is necessary for bigger projects

9.2 Conclusion

In the project we explored and demonstrated a few advanced, user-friendly features to carry out the regular tasks. Working on this project, we came to a better understanding of how ecommerce and engineering techniques can work hand in hand to bring more advancement in the sector. We hope people interested in this sector would get a clear view on the workflow of these kind of businesses and take that extra step to explore the sector for the purpose of improvement.

9.3 Future Work

- A better looking UI
- Product rating feature
- Advance filter on products
- Home delivery
- Monolog
- SMS verification system (in progress)

Short Forms& Acronyms

MVC	:	Model-View-Controller
ORM	:	Object Relational Mapping
API	:	Application Programming Interface
OOP	:	Object Oriented Programming
SRS	:	System Requirements Specification
UI	:	User Interface
DBMS	:	Database Management Systems
IDE	:	Integrated Development Environment
SHA1	:	Secure Hash Algorithm 1

References:

- [01] <https://www.w3schools.com>
- [02] <https://en.wikipedia.org/wiki/Laravel/>
- [03]<https://laracasts.com/>
- [04] <http://stackoverflow.com/>
- [05] <http://www.laravel.io/>
- [06] <http://www.tutorialspoint.com/laravel/>
- [07] <http://www.youtube.com/>
- [08] <http://bootsnipp.com/>
- [09] https://en.wikipedia.org/wiki/Design_specification/
- [10] <http://www.getbootstrap.com/>
- [11] <https://github.com/thekordy/ticketit/>
- [12] <https://github.com/GrahamCampbell/Laravel-Throttle>
- [13] <https://stripe.com/>