**BPP Shops E-commerce Project Documentation**

* **Introduction**

Online shopping is the process whereby consumers directly buy goods, services etc. from a supplier interactively in real-time without an intermediary service over the internet. Online shopping is the process of buying goods and services from merchants who sells on the Internet. Since the emergence of the World Wide Web, merchants have sought to sell their products to people who surf the Internet. Shoppers can visit web stores from the comfort of their homes and shop as they sit in front of the computer. Consumers buy a variety of items from online stores. In fact, people can purchase just about anything from companies that provide their products online. Books, clothing, household appliances, toys, hardware, software, and health insurance are just some of the hundreds of products consumers can buy from an online store.

* **Purposes**

Online shopping tries to enhance access to care and improve the continuity and efficiency of services. Depending on the specific setting and locale, case managers are responsible for a variety of tasks, ranging from linking clients to services to actually providing intensive shopping and delivery services themselves.

* **Objectives:**

To shop while comforting of our own home without having to step out of the

door.

* Sell at lower rate due to less overhead.
* No wait to see the products if someone else are taking that.
* It saves time
* Advantages of our site:
* Saves time
* Saves energy
* More variety
* Easy to find the products
* Shop any time anywhere in world
* Reduces transaction costs
* Price comparison
* Privacy
* No crowds
* **System Analysis**
* Analyzing the system:
* This system is all about the converting the shopping-
* System from manual to online.
* Customer can buy products online after login to the site.
* Administrator is adding product to database.
* Administrator can edit or delete the products from the database.
* After buying and making payment the products are send to customer’s address.
* Customer can write feedback for the product or services.
* Admin can see daily sell and feedback given by customer.
* Administrator is adding the delivery report to the database.
* Both admin and customer can see the delivery report.
* **Functional Requirements:**
* The System must provide following functionalities—
* Keeping records of customers.
* Keeping the records of products.
* Keeping the daily sell.
* Storing the feedback given by the customer.
* Keeping details about the product it is delivered or not etc.
* Storing the items selected by the customer in the temporary storage.
* **Design Specification**
* System Design:

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design.

Specifications to performance specification. System design has two phases of development

* Logical design
* Physical design
* **Logical Design:** During logical design phase the analyst describes inputs, outputs, databases, procedures all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design.
* **Physical Design:** The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.
* **System Tools:**

A project development and an implementation technology can be mapped out using a project timeline. It is a process for defining designing, testing, and implementation of a software application or program. Acquisition of their party tools like dependency manager, database system all can be included for customizing the total system.

Tools that we have used to design and develop our system are as follows-

* **HTML**:

It is used to generate web page. HTML, an initialization of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on.

* **CSS**:

CSS stands for “Cascading Style Sheets” is a language for style and manipulate HTML Language. CSS is a style sheet language used for describing the look and formatting of a document written in a markup language.

* **Bootstrap**:

Bootstrap is free open-source front end web framework that is used to design websites and web applications. HTML and CSS is used to create Bootstrap framework. It makes the web interfaces more user friendly.

* **JavaScript**:

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. Java Script is used to create popup windows displaying different alerts in the system like “User registered successfully”,” Product added to cart” etc.

* **PHP (Laravel):**

PHP is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same.

* **MySQL**:

MySQL is a relational database management system (RDBMS) which has more than 6 million installations. MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases.

* Figure BPP Shops E-commerce E-R Diagram

* Figure: Use Case Diagram

shows the use case diagram for the abstract online store and contains different features of shopping process. In the following sections, a brief description, flow of events, actors, preconditions and post conditions for each feature of the use case diagram has been presented and that will help determine the requirements in the next step.

* Order Process

* Figure: Bpp Shops brand, category, and supplier diagram
* Figure: Add Product diagram
* Figure: BPP Shops user DFD 1
* Search product DFD
* URL DFD
* ERP Design
* Figure: Employee Management System
* Figure: Application Layer

Figure: Application Layer

* Figure: Data Store
* Figure Application Layer
* Figure: Product Search
* Figure: Payment Process
* Order API
* **Module**
* **Product Catalog Features**
* Unlimited pictures with automatic resizing
* Multiple pictures per product
* Zoom-in on product photos
* Watermark on product pictures
* Quick product search
* Customer comments on products
* Customer reviews
* Alias search
* Ajax search
* Double price display: with or without VAT
* Choice of number of products to display per page
* Out-of-stock e-mail notifications
* Display available quantities
* Sort products by relevance, price, alphabet, etc.
* Printable product pages
* Display products in the same category
* Export products to major search engines
* Multi-tier pricing
* Build product attributes
* Display cart content
* Display product categories
* Administrative Features
* Manage brand, categories, subcategories, and sub-subcategories
* Unlimited product characteristics
* Choose product states
* Create catalog mode
* Bulk uploads
* Select products unavailable
* Display unit prices on products
* Customizable favicon
* PDF invoices and delivery slips
* Supplier management
* Sell downloadable products: MP3s, software, etc.
* Display featured products on the homepage
* Insert advertisement banners
* Customize product displays
* Multiple recipient contact form
* Back office universal search
* Text editor
* Manage exotic typefaces
* SMTP E-mails: send e-mails via a contact form, etc
* **Marketing Feature:**
* Google Adsense
* Export product to major search engines with Shopping Flux
* Export product to Product List
* Automated follow-up email
* Newsletter subscriptions
* Loyalty programs
* Recently viewed products
* Coupons and vouchers
* Integrate Google AdWords
* Product presentation videos
* Promotional tools
* SMTP e-mails
* **Security Features**
* Multiple users
* Set security permissions for users
* Maintenance mode
* SSL compliant
* Set password expiration
* Password and cookie encryption
* Resistance to attacks such as XSS, CSRF, SQL injections, distant

inclusions, path transversal

* E-mail header injections blocked
* Password encryption in the database
* Cookie’s encryption
* **Shipping Features**
* PDF invoices and delivery slips
* Supplier management
* Display featured products on the homepage
* Insert advertisement banners
* Customize product displays
* Multiple recipient contact form
* Back office universal search
* Text editor
* Manage exotic typefaces
* SMTP E-mails: send e-mails via a contact form, etc.
* **Search Engine Optimization**
* Search engine friendly
* URL re-writes
* Dedicated URL for each product
* Google sitemap
* Auto-generated site map
* Automatic robot.txt files
* Automatic .htaccess files
* Permalinks
* Edit product tags
* Title tags, meta tags, meta descriptions, etc.
* Tag cloud
* Reduce the load time of products
* Canonical URL: preventing duplicate content
* **Checkout and payment features**
* One page checkout
* Renew an order in one click
* Customize order statuses
* Predefined order messages
* Moneybookers/Skrill pre-configured
* Google checkout
* SSL Commerce pre-configured
* Payment upon delivery (COD)

* **Features and Requirements**
* This chapter discusses the features and requirements of the system that can solve the problem. The first section considers what the mobile shopping process looks like in general. In this section, the use case diagram will show the required features in detail. In the second section, the functional and non-functional requirements will be addressed for the mentioned features while making it easy in the upcoming sections to refer to different requirements.

As the choice of development, the waterfall method was chosen since the project's plan or project requirements were unlikely to change and the testing results or customer's feedback were not supposed to affect the project. So, each model's phase (project planning, analysis, design, implementation, etc.) was accomplished before moving on to the next phase.

* **Use Cases**

The selected frameworks have been analyzed to discover the required system features. Since the online shopping process is almost always the same in a generic way, it could be generalized.

* According to the abstract web shop, the following features are necessary for the whole online shopping process from a mobile device.
* Login
* Browse category
* Search product
* View product description
* Add product to the cart
* Remove product
* Change quantity
* Checkout receipt
* Check item quantity
* Paying order
* Payment method

Figure: THE E-R DESIGN OF ONLINE SHOPPING SYSTEM

* To control the contents of the data in the website, a data entry model E-R is needed to build and generate data sheets. Therefore, according to the description above, along with the need to achieve the purpose of the site or the target. The product can be vulnerable to view through the shopping website. However, there are many systems database papers; recalling a default model explaining the relationship between the contents of the design of the commercial E-R diagram in Figure 38.
* **Login Use Case**

| Use Case Name | Login |
| --- | --- |
| Brief Description | The login process consists of getting the username and password of the customer and checking the authorization with the system. |
| Flow Of Events | * Customer should click the sign in button. * The system displays login dialog box. * Customer should enter username and password. * Customer should click the login button. |
| Actors | Customer |
| Preconditions | The customer’s account should exist on the system. |
| Post conditions | Customer will be able to use the other functionalities of the system. |
| Exception conditions | If the username and/or password are not correct, the login process is canceled and the customer would be notified. |

Figure: Login use case Description.

* **Browse Category**

| Use Case Name | Browse category |
| --- | --- |
| Brief Description | Customer should be able to view the category contents. |
| Flow Of Events | * Customer should click on a category name. * System displays contents of the selected category. |
| Actors | Customer |
| Preconditions | None |
| Post conditions | Customer would be able to browse other categories or view description of a product inside the selected category. |

Figure: Browse category use case description.

* **Search Products**

| Use Case Name | Search product |
| --- | --- |
| Brief Description | Customer could search products by a specific keyword. |
| Flow Of Events | * Customer should enter a keyword. * Customer should click the search button. * System would show the related search result. |
| Actors | Customer |
| Preconditions | None |
| Post conditions | Customer would be able to view description of a product from the returned result. |

Figure: Search products use case description.

* **View Product Description**

| Use Case Name | View product description |
| --- | --- |
| Brief Description | Customer could be able to view the product description such as product name, image, model, price and product details. |
| Flow Of Events | Customer should click on a product name.  System displays description of the selected product. |
| Actors | Customer |
| Preconditions | A category’s contents should have been viewed. |
| Post conditions | Customer would be able to add the viewed product to the items on the shopping cart. |

Figure: View product use case description.

* **Add Products to the Cart**

| Use Case Name | Add products to the cart |
| --- | --- |
| Brief Description | Customer could add a product to the shopping cart. |
| Flow Of Events | * Customer should click on add to cart button. * System must add the selected product to the items on the customer’s shopping cart. * System must display a notification stating that the item has been added to the shopping cart. |
| Actors | Customer |
| Preconditions | Customer must be logged in. Desired product quantity must be available. |
| Post conditions | Customer would be able to remove or change the quantity of the selected product in the shopping cart or customer could view the checkout receipt. |
| Exception conditions | If the product does not exist, the adding process is canceled and the customer would be notified. |

Figure: Add products to the cart use case description.

* **Remove Product from Cart**

| Use Case Name | Remove product from cart |
| --- | --- |
| Brief Description | Customer should be able to remove an already existing product on the shopping cart. |
| Flow Of Events | * Customer should click on the shopping cart button. * System must display the shopping cart contents. * Customer should click on the remove button of a product. * System should display a confirmation dialog box. * Customer could click on OK button to confirm the product’s removal from the shopping cart. * System must remove the product from the shopping cart. * System must display an updated shopping cart list. |
| Actors | Customer |
| Preconditions | Customer must be logged in.  The product should have been already added to the shopping cart. |
| Post conditions | None |

Figure: Remove the product from the cart use case description.

Figure: Activity Diagram for Online Shopping-Cart Application.

* **Change Quantity**

| Use Case Name | Change quantity |
| --- | --- |
| Brief Description | Customer should be able to change the quantity of an already existing product on the shopping cart. |
| Flow Of Events | * Customer should click on the shopping cart button. * System must display the shopping cart contents. * Customer should click on the change quantity button of a product. * System should display a dialog box to get the new quantity. * Customer could enter the new quantity and click on OK button to update the product’s quantity on the shopping cart. * System must change the product’s quantity on the shopping cart. * System must display an updated shopping cart list. |
| Actors | Customer |
| Preconditions | * Customer must be logged in. * The product should have been already added to the shopping cart. Desired product quantity must be available. |
| Post conditions | Customer would be able to remove or change the quantity of the selected product in the shopping cart or customer could view the checkout receipt. |
| Exception conditions | If the product does not exist in the desired quantity, the changing quantity process is canceled and the customer would be notified. |

Figure: Change quantity use case description.

* **Check Item Quantity**

| Use Case Name | Check item quantity |
| --- | --- |
| Brief Description | System should be able to check a product’s quantity on the system to make sure that the desired quantity of a product exists. |
| Flow Of Events | System checks the existing product’s quantity. |
| Actors | None |
| Preconditions | None |
| Post conditions | None |

Figure: Check item quantity use case description.

* **Checkout**

| Use Case Name | Checkout |
| --- | --- |
| Brief Description | Customer should be able to buy the products on the shopping cart. |
| Flow Of Events | * Customer should click on the shopping cart button. * System must display the shopping cart contents. * Customer should click on the checkout button. * System must display the checkout receipt. * Customer should confirm the order details. |
| Actors | Customer |
| Preconditions | Customer must be logged in.  The customer’s shopping cart should not be empty. |
| Post conditions | The customer would be able to do the payment. |
| Exception conditions | If the shopping cart is empty, the checkout process is canceled and the customer would be notified. |

Figure: Checkout use case description.

* **Payment Method**

| Use Case Name | Payment method |
| --- | --- |
| Brief Description | Customer should be able to do the payment. |
| Flow Of Events | * System must display the payment method’s authentication. * Customer should enter credentials require for payment. * System must give customer the opportunity to confirm or deny the payment. * Customer could confirm the payment. * System must display a confirmation that the order has been completed successfully. |
| Actors | Customer |
| Preconditions | Customer must be logged in.  The customer should have confirmed the order details. |
| Post conditions | None |
| Exception conditions | If the payment credentials are not valid, the payment process is canceled and the customer would be notified.  If the payment is not successful, the payment process is canceled and the customer would be notified. |

Figure: Payment method use case description.

**Architecture and Design**

The first section of this chapter describes the overall architecture and design which is the overview of the solution implementing the requirements in the abstract level. The second section includes details for PayPal (the payment method).

* **Overall Architecture and Design**

Different instances of web shops have more or less same functionalities on the abstract level. Dedicated plugins inheriting from an abstract plugin could be used to implement a general communication protocol. Thus, a client could communicate with these plugins and could also be easily instantiated in order to match a particular web shop.

The picked-up frameworks have similar architectures, and to show the server-side architecture and design, the general communication to an abstract web shop would be discussed.

* Now Figure 1.1 could be presented more precise as shown in ////.

Figure:Mobile’s client connection to a web shop’s plugin and database.

Instead of the Android's web browser, the Android's dedicated client connects to the Internet as shown in Figure 4.1 (arrow A). Using the HTTP Client methods, the client connects itself to the plugin which is installed on the server side (arrow B). The plugin which is a PHP module in this case, has the required functions to communicate with the MySQL database (arrow C).

* **Server Side**

On the server side, what matters the most is the communication between the plugin and the database. The plugin should have functions for running different queries on the database, so it could respond to the client's requests which have been fully explained in the features and requirements chapter. The PHP module will be explained in more details on the next chapter.

What follows next is an abstract web shop's database structure. It has been taken from the ECommerce framework's architecture, but satisfactorily demonstrates the other framework's database structures too.

Figure: The framework’s database structure.

There is usually a configuration table on the framework's database which contains the framework's configuration information. Examples of the data which this table could involve are the basic shop's information, installed modules, current setting's information and a lot more.

A bunch of tables usually belong to categories and a group of tables contain information about different products. These two groups of tables should be accessed during the “Browse category”, “Search product”, “View products description” and some more features.

Another group contains the customer's information, these tables along with the previously mentioned groups should be accessed during the “Add to cart” and some more features.

Tables related to orders, address book, taxes, zones and countries are usually needed to be modified during the “Checkout” processes.

* **Supplier**

Supplier are a part of the BPP shops ecommerce. A supplier provides the multiple products on the BPP Shops Ecommerce. BPP shops admin are set the supplier information. And set his role permission. Supplier are only seeing the product list. Supplier seeing the which product are show on the BPP shops e-commerce. And which product are selling. Supplier also see the his/her transaction history. Supplier also see the BPP shops return product history. It’s not non editable page. Supplier are seeing the product requisition. Product requisition are coming from BPP Shops admin panel. Supplier search his product on the product list are BPP shops product code or supplier product code.

* **Agent**

Agent is a part of the BPP Shops e-commerce. BPP Shops admin panel are register the agent details. And set the role permission on agent.

* **Financial Intermediaries and E-Payment Service Providers**

Financial intermediaries and E-Payment Service Providers provide payment services when they are commissioned by E-Commerce platforms/marketplaces/vendors.

The traditional role of financial intermediaries consists of the transfer of the payment from the consumer/buyer to the vendor. In the E-Commerce environment, where the consumer/buyer may have limited knowledge of the vendor and may fear identity theft and fraud, the security of the customer/purchaser’s bank data is of major concern. As a response, financial intermediaries have developed payment solutions that are only indirectly associated with the customer/purchaser’s bank account. These include secure debit cards, which mitigate the risk involved with the vendor storing credit card information, and online payment systems provided by specialized online payment service providers.

Typically, the vendor enters into an agreement with the payment service provider to facilitate transactions with consumers/buyers. Payment by consumers/buyers may be made directly to an “e-money” account with the payment service provider, or directly to the vendor’s bank account. The system is secured, and generally the vendor does not receive the bank, credit or debit card data of the consumer/buyer. The consumer/buyer may not always be required to have an account with the payment service provider.

During the payment process, the financial intermediary collects and stores data such as vendor and consumer/buyer account information (name, address and bank details) and the financial data of the transaction. In most cases, the financial intermediary does not collect information concerning the nature of the goods being sold or the place to which they are to be delivered. Depending on the information available with financial intermediaries and data privacy laws, possibilities for sharing such information with Customs and other government agencies within the framework of the relevant regulations could be explored.

* **OVERVIEW OF THE SYSTEM**

Visitors/Customers: The Individuals will be the end users of the website. They will be able to

perform the following tasks:

* **Home Page:** This will work as the welcome screen for the Visitors/Customers. Customers can get to know about different plastic accessories and products sold from the website from this page. Customer also can explore all the main pages.
* **About Us:** ‘About Us’ will provide information about the business-like inception of the business, how the business grew, etc.
* **Products:** Customers can find all the products on the website and can purchase the products from the same website itself. There can be different dropdown subpages or the product categories can be individual pages as well. That can be decided during design phase.
* **Contact Us:** A ‘contact us’ page typically contains your business contact information, address, a map and a form wherein your customers can submit queries, feedback, suggestions, comments, sales enquiry etc. These details are sent out to an email address selected by you.
* **Google Map on Contact Us Page:** We will integrate Google Map on your Contact us page, showing exact location of your Business on the map. This helps your target audience to locate your business or reach you easily. For us to be able to do this, your address should be traceable by Google in its map.
* **MULTIPLE BANNER STYLE:**

We have added multiple pre-made banner styles that you can select the ecommerce. You can also add your images to those banners. Banners are below:

* **Ad’s banner:**

Ad’s banner is show on the below the main slider. And they are the tree banner. Banner size are 612x150.

* **Top Banner:** Top banner are show on the right side of the main slider. And they are the two banners. Top banner size is 933X110.
* **Slider:** Slider are show on the below the top nav bar. And they are show on the multiple sliders. Sliders are size are 933X236.
* **Search Product:**
* Customer search for products on the website.
* Search module will use “like search” module to provide product suggestion as per the terms used to search.
* Customer can browse for different products on the websites on the home page as well as product page as well.
* Customer now can go to specific products’ details page on the website.
* When a customer clicks on a product, they will be redirected to the products’ details page where customers can select the quantity they want to order and they add the product to the cart.
* Cart and Checkout module is described in details later.
* To complete the purchase customer/visitor need to register for new user and login for existing user.
* **Filter & Search Products:**

Option to filter product in a list by a category, type, price, size, etc.

This will perform as an advance search.

* **Product Details**
* Description
* Image
* Price
* Quantity
* Size
* Etc.
* **Add to Cart:**
* The cart module will be incremented by one when an End-User starts adding products to the cart.
* On clicking cart icon, the system will redirect to the cart page with column for:
* Item Description
* Quantity
* Price
* Apply discount coupon
* Add product to wish list
* Sub Total
* Four buttons at the bottom: Add More, Delete, Continue Shopping and Place Order.
* Add More will give clients an option to increase the quantity and delete will remove the product from cart.
* Place Order will lead to Payment Process and Continue Shopping will allow users to redirected back to the products section so that they can purchase more.
* **Check out Module:** Check out process is divided into three parts:
* Billing & Shipping
* Review & Place Order
* Payment
* **Billing & Shipping Information**: It will have the required fields:
* First Name
* Last Name
* Email Address
* Phone
* Address
* Suburb/City
* State
* Post code

Shipping address will have the same fields.

If both the addresses are same, then on ticking the check box the shipping address fields will get automatically populated. Option for continue checkout. Non-Registered Customers need to provide the details for first time. After providing the details and continue they will get registered on the website and will receive their password on their email. For registered customers the billing and shipping address will get auto populated from their address book. If registered customer wants the order to be delivered to a different address, they need to check the box “Add a new address” and they need to provide the address. Existing User need to login during this stage.

* **Recently viewed products:**
* List of products which is viewed by customers recently are shown under recently viewed
* products tag.
* Customer will also get suggestion of most sold products.
* **Order Date and time:**

Customer set the order delivery date and time. It’s the mandatory field. Order delivery date current date to next 5 days and time start from morning 10.00 AM to 8.00 PM 1 hour’s gap. But when the customer order by the non 1.00 PM he/she doesn’t see the previous time.

* **Product Browsing**

Display feature product on home page: Customer can see few featured products on the home page set by the admin

* Multiple images per product (Up to 5 max)
* Product image zoom-in capability (Cursor Hovering Zoom)
* Product attribute
* Product review & ratings
* Add products to wish list
* Stock availability
* **PRODUCT MODAL WITH QUICK VIEW BUTTON:**

You can view the product by hovering over the mouse cursor. The hover process will let you zoom the product.

* **PRODUCT RATING AND REVIEW:**

You adding ratings and reviews of the products, so that customers can check the feedback of other customer.

* **Payment**:
* After entering the Review and Confirm, the next phase is Payment.
* Various mode of payment is available:
* Visa/Master Card/PayPal Etc. (Client needs to provide Payment gateway API)
* Stripe payment gateway to be used for payments (if required) using Cards.
* **Sign Up:** Customer can register from the home page of the website

The registration fields are as follows:

* Login with OTP
* Functional Requirements: \*\*
* Each service should be able to generate an OTP and each tuple order\_id tuple should be unique for a specified time period -- say a week
* User should be able to validate himself against that OTP
* User should not be able to enter the OTP for any other user.
* Non-functional requirements: \*\*
* Delivery latency should be low.
* Service should be consistent
* Service should be as available as possible
* Durability the generated OTPs should be available till the time user

Capacity estimation: \*\*

* Let’s say we have 100 million users of which 50 million are DAUs
* Let’s each DAU views 10 items on the website and 1% of all views gets converted into OTP generation
* So, number of OTP generated in a day 50 million100.01 = 5 million;
* Number of OTPs in a week = 75 = 35 million;
* Number of digits in our OTP = 8
* Number of OTPs per hour ~ 200K
* Number of OTPs per sec = 200/ (6060) < 1K

\*\*APIs for our service: \*\*

Generate OTP For Order (service\_id, user\_id, order\_id, channel);

Generates OTP for the particular service, user and order ids and sends it via the specified channel.

Validate OTP For Order (service\_id, user\_id, order\_id, OTP)

Checks if the OTP provided is for that particular user and service and for that particular order.

* Login with Facebook
* Login with Gmail
* **Customer Dashboard**
* My Profile:
* Customer can view/edit their profile information from My Profile. Profile is: customer name, email address, phone number, gender, upload image, date of birth, address.
* **My Cart:**
* Customer can see the entire product that they have added in their cart but haven’t completed the purchased procedure.
* **My Orders:**
* Customers can see all the orders they have placed.
* Customers can see all their past orders.
* Customers can download or see the invoices for their orders.
* Customers can also track their order from My Orders.
* Customer need to provide their Tracking Number to track the orders
* **Address Book:**
* User can add address as many as required.
* User can edit existing address & make any address as default address.
* When the customer add the address the he/she follow the street address (mandatory field), Select district, select area/thana, (mandatory field), floor no, apartment no, name, phone number.

**BACKEND MODULE:**

* **POS MANAGEMENT:**

We have introduced a POS system in the website which can handle sales monitoring and reporting, analysis, inventory tracking, mobile connectivity, customer data management, employee management, and robust integrations.

* **BRAND AND CATEGORY MANAGEMENT**

Firstly, admin set the multiple ecommerce, then the admin set the product brand name, product brand image, product category name, product category image, product category icon and after the set sub category name, sub category image and sub category icon, and finally set the product sub-sub category name, sub-sub category image and sub-sub category icon. When the admin Once add button is clicked, the web page sends this information to the server, which in turns stores it in the database.

* **PRODUCT MANAGEMENT:**

**Add Products:** In this section, you will manage the products you have various settings in this section you can add the product. Products can be different types: single product, variable product, product unit will be added in this section as well. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details including an image (thumbnail image size are 400\*400 and multiple image size are 1024\*1024). Firstly, admin set the multiple ecommerce, then the admin set the product brand name, product brand image, multiple product category name, product category image, product category icon and after the set multiple sub category name, sub category image and sub category icon, and finally set the product multiple sub-sub category name, sub-sub category image and sub-sub category icon. When the admin Once add button is clicked, the web page sends this information to the server, which in turns stores it in the database. Brand, category, sub category and sub-sub category are store the same table. And E-commerce name are static data. Overall total ecommerce number are 16. And the product table are store on the database are different table. And finally add product page are not modal. It’s a single web page.

**Edit Products:** Admin can edit or modify the products by clicking the Edit button. The web page is displayed a list available product from the database. The admin then chooses the item he/she wants to modify by clicking on particular item. A page is displayed letting the admin to modify all item information, the web page sends a message to the server, which updating the information from the database Delete Products: Administrator can delete the products based on the stock of that particular product. The admin starts this action by clicking on the Delete button. Once, delete button is clicked, the page is sent message to the server which in turns tells the database to remove the item.

**List Product:** Admin are seeing the list product. List product are showing on the: Multiple or single e-commerce name, product image, product code, supplier name, supplier product code, brand name, category name, product name, weight, color, size, purchase qty, available stock qty, buying price, selling price, discount price, discount start date, discount close date, product expire date, purchase date, QR code, Status, and action button.

* **EXPENSE MANAGEMENT:**

We have introduced expense management which is a system for processing expense reports approvals and a sum paid to cover the money that has been spent or lost. Through Expense management we can not only track employee spending but also determine how the organization will reimburse the costs incurred. It also applies the procedures and polices used to control this type of spending.

* **REPORT MANAGEMENT:**

**PROFIT/LOSS REPORT:**

We have introduced a profit and loss report that summarizes the revenues, costs and expenses incurred during a specified period, usually a fiscal quarter or year. This indicates how the revenue and transformed into net income or net profit. You can generate the report as per your requirements.

**SALES REPORT:**

Through the report, you come to know about the business sale and its activities for a selected date range of all the products or the specific selected product that we deal in. It also shows the trending sale items. This report helps you to assess how well the products are doing.

**CUSTOMER REPORT:**

A customer report always you to know what a customer has purchased from you and what a customer has paid for it. It included all detailed transaction made between you and the customer. Also show the current balance.

**STOCK REPORT:**

One of the most important reports that give you information about stock. This report tells you how much inventory/stock is present in our warehouse. You can also check the stock of the specific date by the filter. It gives the summary of the stock making sure to check.

**EXPENSE REPORT:**

From this report, you can track your business spending. It includes all types of expense you made from your business. With the help of this report, you can compare the current month expense with the previous month expense that will control your expense in a better way.

**PARALLAX BANNER:**

We have added parallax banners at the background images of the home pages. You can add your images if you want to change the default parallax banner.

**USER CONTROL PANEL:**

We have introduced multiple option of the users on the website that you can manage an account, add a profile picture and change the password.

**EMAIL TEMPLATES:**

A good email template can make or break your marketing, but designing and building a template from scratch can be a very time-intensive process. Instead of creating a template from the ground up, let us save you hours of precious time with free email template resources.

**USER ACCOUNT, CART, CHECKOUT**

All crucial page templates are designed and developed to ensure your online shop will provide as smooth and engaging a user experience as possible.

**CONTACT & NEWSLETTER FORMS**

Rawal templates are using dynamic contact and newsletter forms you can customize according to your desire.

**INVOICE FEATURE:**

In CMS we have introduced the Invoice Feature in which you can generate the invoice for one or multiple customers. All you need to do is add the Invoice Address, Invoice Email, Mobile number, Phone number, Invoice logo, Invoice Prefix, and invoice footer.

**BARCODE FEATURE:**

We have introduced the barcode setting for the products for POS. Following things are required for the barcode to work: The ID of the product, Name, Continuous-Feed/Rolls, Top Margin, Sticker Heights, Sticker Width, Paper width, and Paper Height. Sticker in one row, Sticker between 2 rows, Distance between 2 columns, Sticker per sheet, and Actions. Once you have filled in this information you will be able to Run POS.

**SEO CONTENT:**

In SEO content you can add SEO title, SEO Meta Tag, SEO Keywords and SEO Description. Once you have finalized the SEO Content then you can submit it.

**APPLICATION SETTINGS:**

You can manage the application from CMS rather than editing the source code. You can edit Styles for Header, Slider, Banners, Tab Style, Brand Sliders, Footer styles and many more.

**BANNER MANAGEMENT:**

In this section you can change the banners as per your requirements as we have made, you will have multiple options for all the banners that are on the website. All you need to do is click on the desired banner, upload a new image, and boom it’s done.

**WAREHOUSE MANAGEMENT:**

In this section you can add the information about your warehouse. You can add the information as warehouse name, Code of a warehouse, phone number, Email, and status if the warehouse is inactive or not.

**PAYMENT GATEWAYS:**

In this section we have introduced multiple build-in payment gateways that you can activate as per your region and requirements. All you need to do is activate the payment gateway that you want on your website and add its API credentials that you can get while creating the account on the Payment gateway website. We have also introduced a sandbox and live environment for the payment gateway to check before making it live on the website.

**SHIPPING METHODS:**

Local Pickup, Free Shipping, Shipping by weight, and flat rate. You can select any of the above shipping methods and select the shipping rate as per your requirements.

**TAX SETTINGS:**

In this section you can define the ratio of your tax ratio. You can add the tax name with the percentage that you are going to deduct.

**COUPON SETTINGS:**

There are two types of coupons in the coupon section, fixed and percentage. Users can add and remove any coupon at any given time. The fixed coupon allows you to give a fixed amount off whereas the percentage coupon gets a percentage off from the total order.

**QUOTATION MANAGEMENT:**

We have introduced a Quotation Management System which allows a company to create, submit and track quotes and invoices. The Quotation Management System is an open-source web-based application. You can add the following information on the website: Biller information, Purchaser information, Customer information, Warehouse, Quotation Status and, total.

**SALE RETURN MANAGEMENT:**

We have introduced Sale/Returns management that involves interfacing with customers who wish to return a product, and then collecting, organizing, and restocking inventory that has been returned or exchanged. Returns management goes beyond the final delivery and is not used for every customer order. You can add Sale ID, Customer name, Warehouse Name, Description, Quantity, Payable Amount, Paid Amount, Tax Amount, Sale Date, and Due amount.

**BILLER MANAGEMENT**

We have introduced Billing Management to handle time and billing tracking as well as invoicing customers for services and products. It helps in managing a chain of stores as well as multiple companies billing systems. It also provides recurring services and renting business billing solutions.

**ROLES/PERMISSION MANAGEMENT:**

In this section, you can introduce the roles of the management. You can give Admin privilege to other users, editor access, or limited access.

**BALANCE SHEET:**

We have introduced a Balance Sheet feature that displays the company’s total assets, how these assets are financed, through either debt or equity. It can also be referred to as a statement of net worth, or a statement of financial position.

**TRIAL BALANCE:**

We have introduced trial balance as a bookkeeping worksheet in which the balances of all ledgers are compiled into debit and credit account column totals that are equal.

* **Ecommerce API**

| Get | Post | Put | Delete |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* **Testing:**

Software testing is the process used to measure the quality of developed computer software. Usually, quality is constrained to such topics as correctness, completeness, security, but can also include more technical requirements as described under the ISO standard ISO 9126, such as capability, reliability, efficiency, portability, maintainability, compatibility, and usability. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

* **Black box testing:**

It treats the software as a black-box without any understanding as to how the internals behave. Thus, the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behavior), is the same as the expected value specified in the test case.

* **White box testing:**

It is when the tester has access to the internal data structures, code, and algorithms. For this reason, unit testing and debugging can be classified as white-box testing and it usually requires writing code, or at a minimum, stepping through it, and thus requires more skill than the black-box tester. If the software in test is an interface or API of any sort, white-box testing is almost always required.

* **Gray box testing:**

Grey box testing is the combination of black box and white box testing. Intention of this testing is to find out defects related to bad design or bad implementation of the system.it is used for web application.

* **Unit Testing:**

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

Unit testing is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a class, which may belong to a base/super class, abstract class or derived/child class.

* **Integration Testing:**

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integration testing is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision. There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

* **Validation testing:**

At the validation level, testing focuses on user visible actions and user recognizable output from the system. Validation’s testing is said to be successful when software functions in a manner that can be reasonably expected by the customer. Two types of validation testing-

* **Alpha testing:** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.
* **Beta testing:** comes after alpha testing. Versions of the software, known as beta version, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.