Tribhuvan University

Faculty of Humanities and Social Science

Shop Swiftly (Online shopping system)

A PROJECT REPORT

Submitted To

Department of Computer Application

Shahid Smarak College

In partial fulfillment of the requirements of the Bachelors in Computer Application

Submitted by: -

Amir Maharjan (26202002)

December 2024

Tribhuvan University

Faculty of Humanities and Social Science

Shahid Smarak College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by **Amir Maharjan** entitled **"Shop Swiftly"** in partial fulfillment of the requirements for the degree of Bachelor of computer Application is recommended for the final evaluation.

_____

Hari Lal Chalise

SUPERVISOR

Shahid Smarak College

Kirtipur, Nayabazar

Tribhuvan University

Faculty of Humanities and Social Science

Shahid Smarak College

Letter of Approval

This is to certify that this project prepared by **Amir Maharjan** entitled **"Shop Swiftly"** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

_____ Hari Lal Chalise Supervisor Shahid Smarak College Kirtipur, Nayabazar	_____ Mr Sanjay Jnawali Coordinator Shahid Smarak College Kirtipur, Nayabazar
_____ Internal Examiner	_____ External Examiner

Acknowledgement

I extend my heartfelt gratitude to the supervisor Hari Lal Chalise who played a pivotal role in the successful completion of the Shop Swiftly Project. His unwavering support and contributions have been instrumental in bringing this project to fruition.

I am deeply thankful to my project supervisor for his guidance, mentorship and invaluable insights throughout the project's lifecycle. His expertise has been a guiding light that shaped the project's direction. I also want to acknowledge our families, whose unwavering support, understanding, patience were the pillars sustained throughout this project's journey.

Lastly, this project has been collaborative effort, and I am proudly grateful to everyone who contributed in any capacity. The knowledge gained and lessons learned during this project will guide me in my future.

Yours Sincerely,

Amir Maharjan

Abstract

This project presents a comprehensive e-commerce website designed for both users and administrators. On the admin side, administrators can efficiently manage products by adding details such as prices, stock levels, media, and pages. They have the ability to accept or reject user orders and monitor order statuses, including pending, canceled, or completed. Additionally, administrators can generate reports over various time frames—daily, weekly, or monthly.

Users benefit from a user-friendly dashboard where they can view their order history and track the status of their orders. They also have the capability to update their usernames and email addresses.

The project integrates the YouTube API to incorporate videos into the website, enhancing user engagement. To improve interactivity and user experience, third-party React libraries such as Chart.js and Font Awesome were utilized to render visual data representations like pie and bar charts. For payment processing, the system employs Esewa, a popular digital wallet in Nepal.

During the development of this project, it was noted that many e-commerce platforms offer built-in Instagram integration rather than YouTube integration. While the YouTube API is freely accessible, obtaining access to the Instagram API requires contacting the company directly, which can lead to delays.

At the end of this project, an interactive and user-friendly website will be ready. Where users can order their desired products, view their prices and watch videos related to them. It also has its own CMS from where admins will be able to manage their contents like adding products, viewing reports, accept or reject user orders.

***Keywords**: E-commerce, YouTube API, Chart.js, Payment Gateway*

Table of Contents

List of Figures

List of Appendices

List of Tables

List of Abbreviations

Adobe XD	Adobe Experience Design
CASE	Computer Aided Software Engineering
CSS	Cascading Style Sheet
DFD	Data Flow Diagram
DOM	Document Object Model
ER	Entity Relationship
HTML	Hyper Text Markup Language
MySQL	My Structured Query Language
PHP	HyperText Preprocessor
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language
API	Application Programming Interface
AJAX	Asynchronous JavaScript and XML
KISS	Keep It Simple, Stupid
DRY	Don’t Repeat Yourself
CMS	Content Management System
SCSS	Sassy Cascading Style Sheets

Chapter 1: Introduction

1.1 Introduction

This project was developed in response to the growing demand for customized websites equipped with its own CMS. Users increasingly prefer customized solutions to pre-defined templates for reasons such as personalized configuration, scalability, optimized performance, flexibility, control, security, and ownership

This project is a new eCommerce platform with a unique CMS that allows efficient management of all website content. Administrators can easily view products, prices, images, categories, and pages directly from the CMS interface. This centralized system empowers users to create customized shopping experiences that meet their specific needs.

Designed with the user experience in mind, this project offers a seamless interface that improves the navigation and simplifies the buying process. Features such as advanced search functionality and self-service recommendations ensure that users can quickly find what they need. Additionally, the platform provides scalability for businesses of all sizes, allowing them to adapt to changing market demands while using built-in analytics to make data-driven decisions

1.2 Problem Statement

When we look into other CMS such as WordPress and Joomla, they use plugins and extensions to create an e-commerce website. The biggest disadvantage of this method is the dependency on third party applications, in the case of WordPress, Woo-commerce plugin.

This project looks to eradicate this exact problem by integrating e-commerce in the website directly without the use of third-party applications and extensions.

The problems that need to be addressed are:

- Dependency on third-party applications
- Performance overhead
- Security risks

1.3 Objectives

The main objectives of this project are listed below:

- To create an attractive CMS
- To create an environment where admin can easily create, manage and organize their contents.
- To create a system that will allow users to compare two products

1.4 Scope and Limitation

Scope

- People with minimum knowledge about website can also use this website.
- Can become similar to amazon
- User-friendly interface for booking.

Limitation

- Relies on internet connectivity for real-time tracking and updates.
- Initial implementation and training may require time and resources.

1.5 Report Organization

This report is organized into five chapters, each serving a distinct purpose:

- **Chapter One: Introduction**

  This chapter introduces the project, outlining its objectives and problem statement.
- **Chapter Two: Background Study**

  This chapter provides a comprehensive overview of existing systems, discussing their features, advantages, and disadvantages.
- **Chapter Three: System Analysis and Design**

  This chapter presents the analysis and design of the system, including a detailed requirements analysis and feasibility study to assess the project's viability.
- **Chapter Four: Implementation**

  This chapter covers the implementation process of the system, detailing the methods used for testing and debugging to ensure functionality and performance.
- **Chapter Five: Conclusion and Future Work**

  This final chapter summarizes the findings of the report, discusses its limitations, and suggests potential areas for future enhancement.

Chapter 2: Background study and Literature Review

2.1 Background study

During the development of this project, I researched and studied a couple of existing systems, they are listed below:

WordPress: It has been in operation for over 20 years. Almost 43% of all websites have been made using WordPress. I studied this to understand how data flows from one component to another and tried to do the same for user friendliness as many users are already familiar with how WordPress works [1].

Shopify: This is an e-commerce platform much similar to our own. This platform is built for businesses that want to create online stores. But there is one major flaw, to use the best features, users have to pay a certain amount of money which increases the user's budget [2].

Joomla: It is one of the most well-known CMS in the world. It has been in operation for almost 20 years. It can be pretty complex especially if someone is used to WordPress, which has thousands of available themes and plugins that extend the core functionality. There can be some compatibility issues as well [3].

2.2 Literature

Author in [4], talks about what a CMS is, its importance, etc. The author failed to mention many import aspects of a CMS such as role and capability of user that allows and provides required permission to edit, publish and delete a certain post.

Author in [5] talks about the challenges that companies face with CMS, the best CMS to use, etc. The author even talks about which is better, use a custom CMS or use a popular system. The author clarifies that the usage depends on the client's requirements but he fails to mentions the advantages and disadvantages of a custom CMS and an already existing system.

Author in [6], writes about the benefits, types of CMS, etc. The author fails to mention about how the CMS should look, the coding standards to follow and maintain. Disadvantages of using an already existing system was also not discussed in this article.

Author in [7], used programming languages like PHP, HTML, CSS, SCSS were used. It is one of the best-selling CMS templates in Envato market which is the best market place for these types of templates. As for database, author most likely used phpMyAdmin as they have used php. It is very attractive to look at but not as efficient, as user are most likely to be confused while using it.

Author In [8], have used more than 9 backend languages like Laravel and Node. As for frontend languages, HTML, react, angular, etc. were used. They used KISS methodology that emphasizes simplicity in various fields, including software development and project management. DRY methodology was also used.

Author in [9], have used Vue 3, Next.js 14 and Bootstrap 5 for frontend and Laravel, Dot Net core, etc. for backend. They provide ultra-responsive design that performs seamlessly in desktops, tablets and smartphones. They have provided detailed documentation, so even if the users are confused, they can read the documentation to solve their issues. Most feedbacks are about how amazing their support is.

Author in [10], have provided their code in GitHub which most authors don't do. Its users have complained about the presence of useless codes, bugs and poor loading pages in AJAX implementation. It has issues when developing with Vue Js and react. As for the good side, their codes are clean and very readable.

Chapter 3: System Analysis and Design

3.1 System Analysis

During the development of this project Waterfall model is used. It is one of the most commonly used methodologies as well as the oldest. Gantt chart was used to track the progress through each phase. Reviews were conducted at the end of each phase to ensure that all the expectations were met before proceeding. A considerable amount of time was spent in the requirement gathering phase to ensure all aspects were covered before moving forward. The figure of waterfall model is given below.

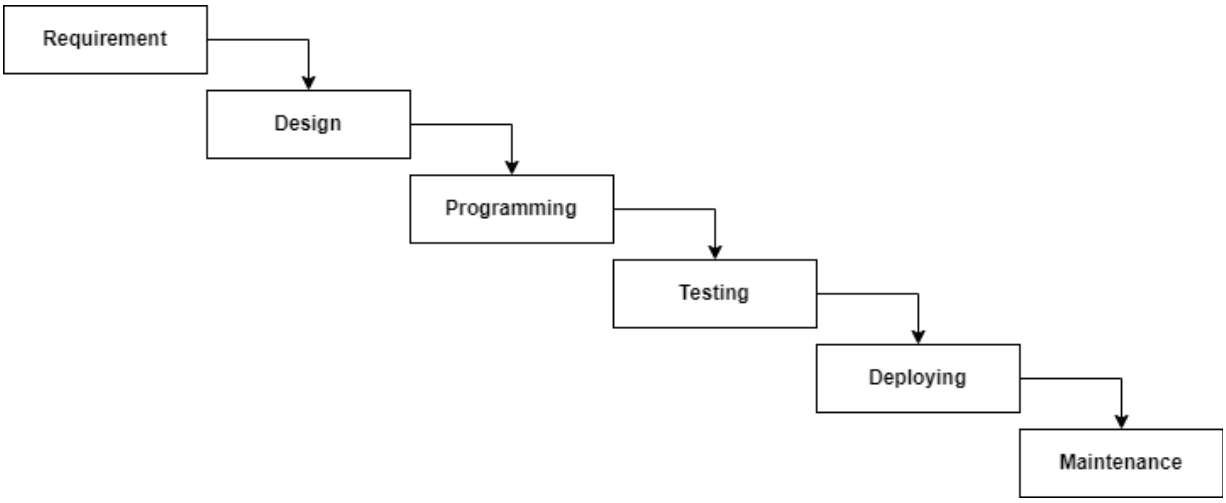


Figure 3.1 Waterfall Model

3.1.1 Requirement Analysis

Before beginning any project, the requirements should be analyzed. There are two ways to analyze the requirements, they are given below:

3.1.2 Functional Requirement

Functional requirements are requirement that make up our entire website. For example: a registration form, when a user fills it, the data that is submitted has to be stored in the database in a secure manner.

The system provides the following functionalities:

- Add to cart
- User request to cancel their booking
- Light mode / Dark mode
- Attractive user interface

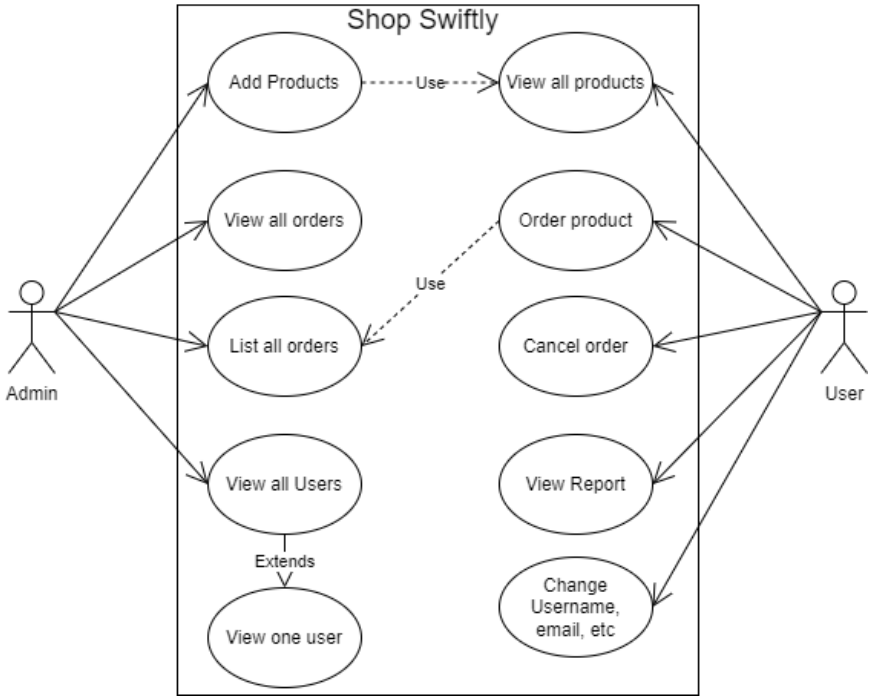


Figure 3.2: Use Case Diagram

In the diagram, user can interact with “view all products”, “order product”, “cancel order”, “view report”, “change username, email” and admin can interact with “Add products”, “view report”, “Add media”, “view all users”, “view one user”

3.1.3 Non-Functional Requirement

- **Reliability:** System will run 24/7. The data that is submitted won’t be made public or lost when the user logs in next time.
- **Security:** Personal data and information of the user will be stored in a secure manner so that these data and information won’t be hacked or misused.
- **Availability:** The system can be accessed by anyone, anytime and anywhere. What a user’s needs to access the system is a smartphone or a computer or a laptop and a internet connection, that is all.
- **Performance:** Every system needs to perform well in order to be liked by the users. The system is fast, interactive and is easy to use.

3.1.4 Feasibility Analysis

Feasibility means to check if the system can be done conveniently or not. A feasibility study is carried out to determine whether the project should (proceed) be done or not. The feasibility of a project is determined by the following: -

- **Technical:** Technically speaking, there are no obstacles because all that is needed for the development stage is a functional computer and a reliable internet connection.
- **Operational:** There are no operational difficulties as well. To operate this system, all it needs is good database connection and a good host.
- **Economic:** Overall budget can be affected a little bit because of the purchase of database and a host.
- **Schedule:** Probably the most important aspect in feasibility study. If a project cannot be completed in time, the project should not even be in consideration. It should be rejected immediately.

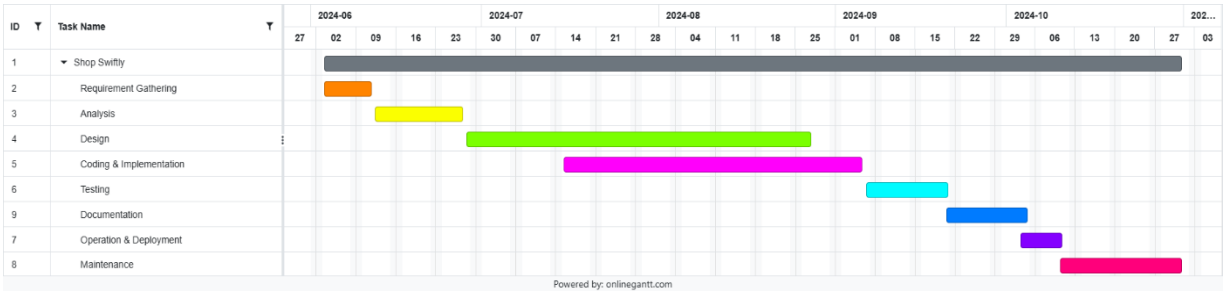


Figure 3.3: Gantt chart

3.1.5 Data-Modeling (ER Diagram)

A data model is a mechanism that provides abstraction for database application. Data models define how data is connected to each other and how they are processed and stored in a system. ER diagram is a database structure that show the relationship between entity and their attributes connect with each other via primary key and foreign key.

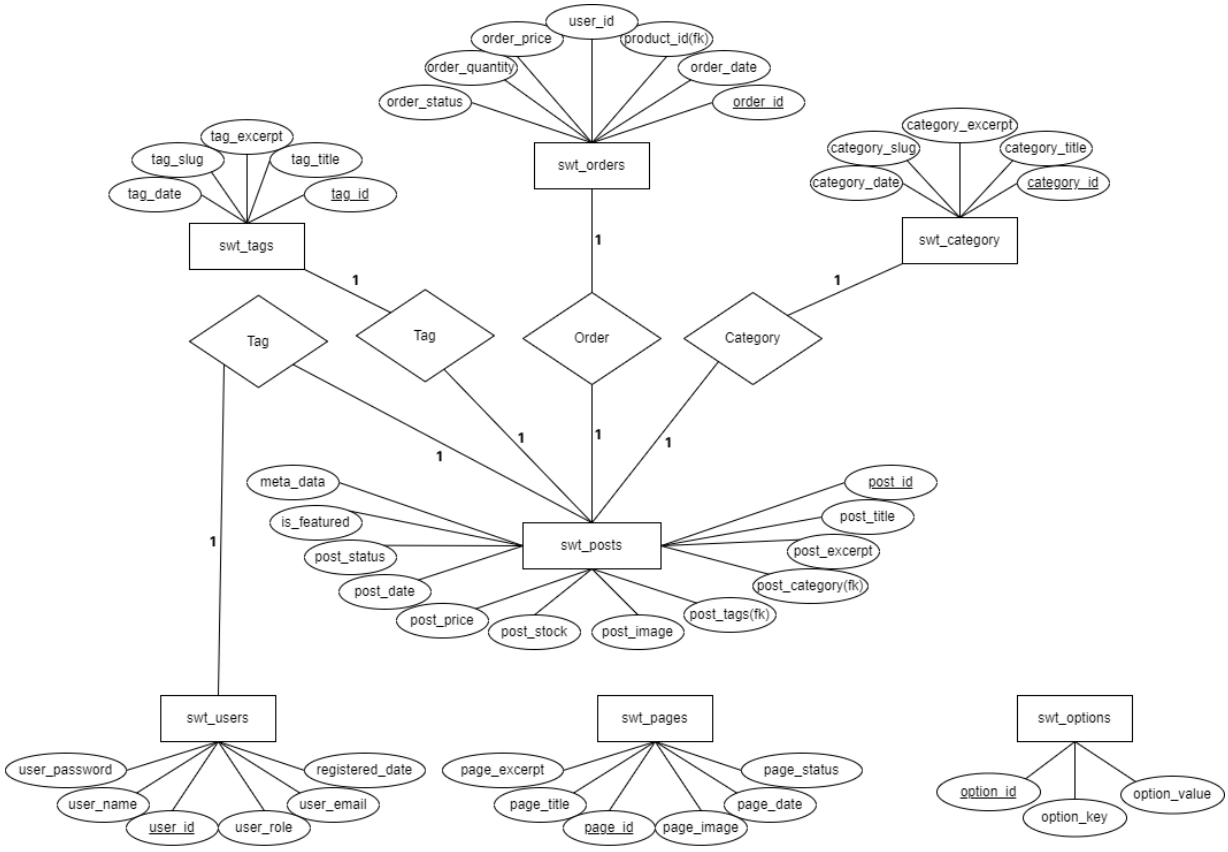


Figure 3.4: ER Diagram

3.1.6 Process Modeling (DFD)

DFD is a graphical representation of how the data flows in the proposed or an already existing system. The DFD is also known as context diagram or bubble chart. Its main purpose is to help understand how the data flows in the system. The figure below is a basic data flow diagram of "Shop Swiftly".

DFD level 0 is a graphical representation of the system without much detail. It just gives a basic overview of how the data flows in the proposed or an already existing system. Below is a DFD level 0 figure:

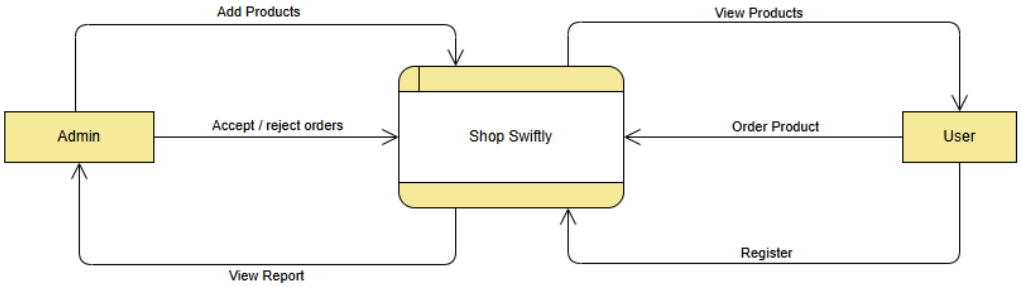


Figure 3.5: DFD Level 0

DFD level 1 is another graphical representation of the system that has more detail of how the data flows in the system. It has more details than the DFD level 0. A DFD level 1 is given below:

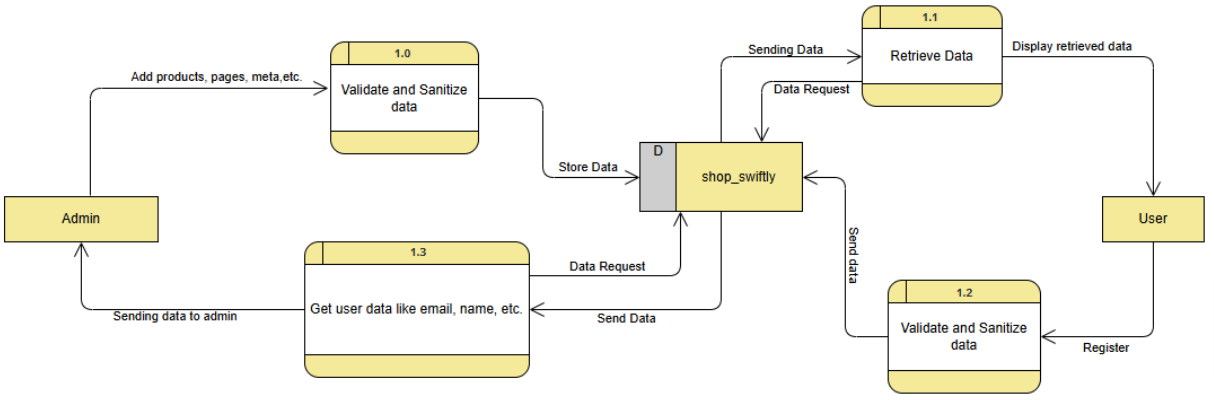


Figure 3.6: DFD Level 1

DFD level 2 has the most details of how the data flows in the system than any other level. It tries to display as much detail of data flows as possible. A figure of DFD level 2 is given below:

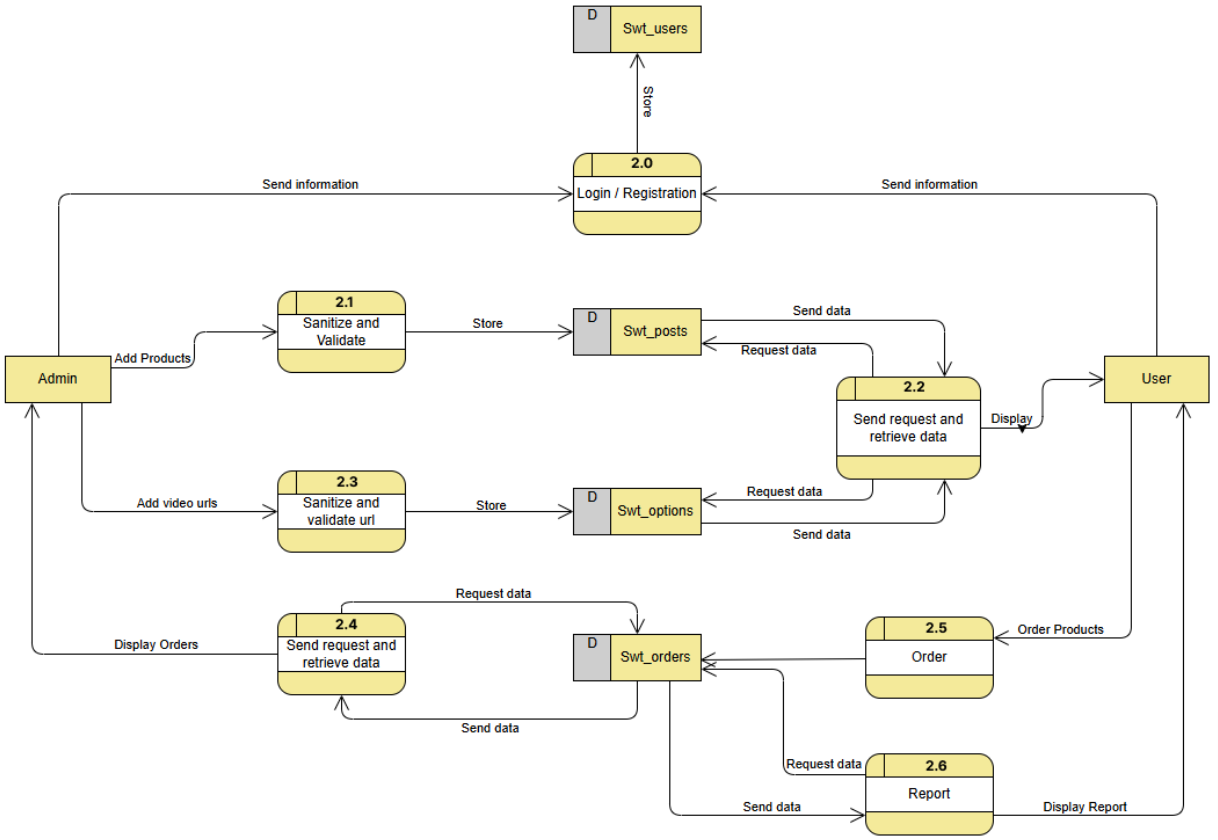


Figure 3.7: DFD Level 2

3.2 System Design

The process of meeting the end-user requirements by designing the architecture, components and interfaces for the system is called system design. The system must be designed in such a way that it meets all of the end-user's requirements.

3.2.1 Architectural Design

The purpose of this phase is to design how the users will see the system, how the system deals with user inputs and how the data gets stored in the database. All these things are planned and carried out in detail without missing out on anyone of these things.

Presentation login: how the users will see the system.

Business logic: how the user's inputs will be handled.

Database driver: how the user's inputs are stored in the database.

The architecture of the application is shown in the figure below.

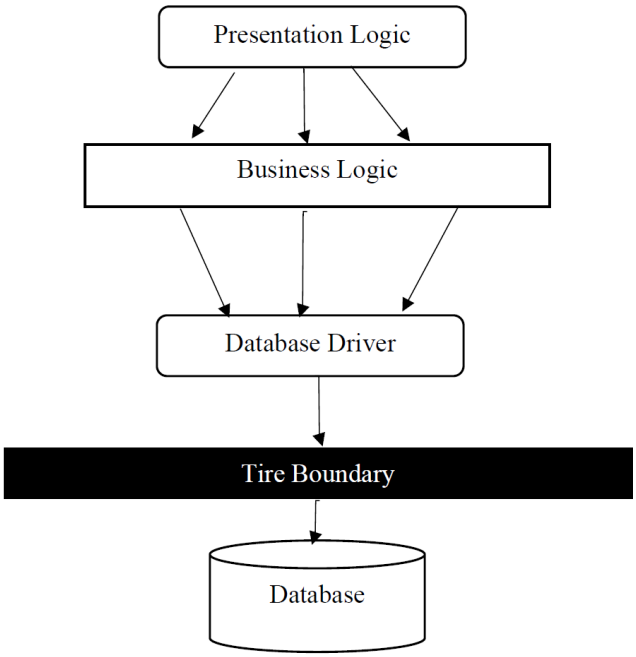


Figure 3.8: Architectural Design

3.2.2 Database Schema Design

According to dictionaries, schemas are a representation of a plan or theory in the form of an outline or model. Database schema is the skeleton structure that represents the logical view of the entire database. It shows the relationship between different tables and its attributes. Two

tables are linked with each other by the use of primary key and foreign key. It is designed by database designers.

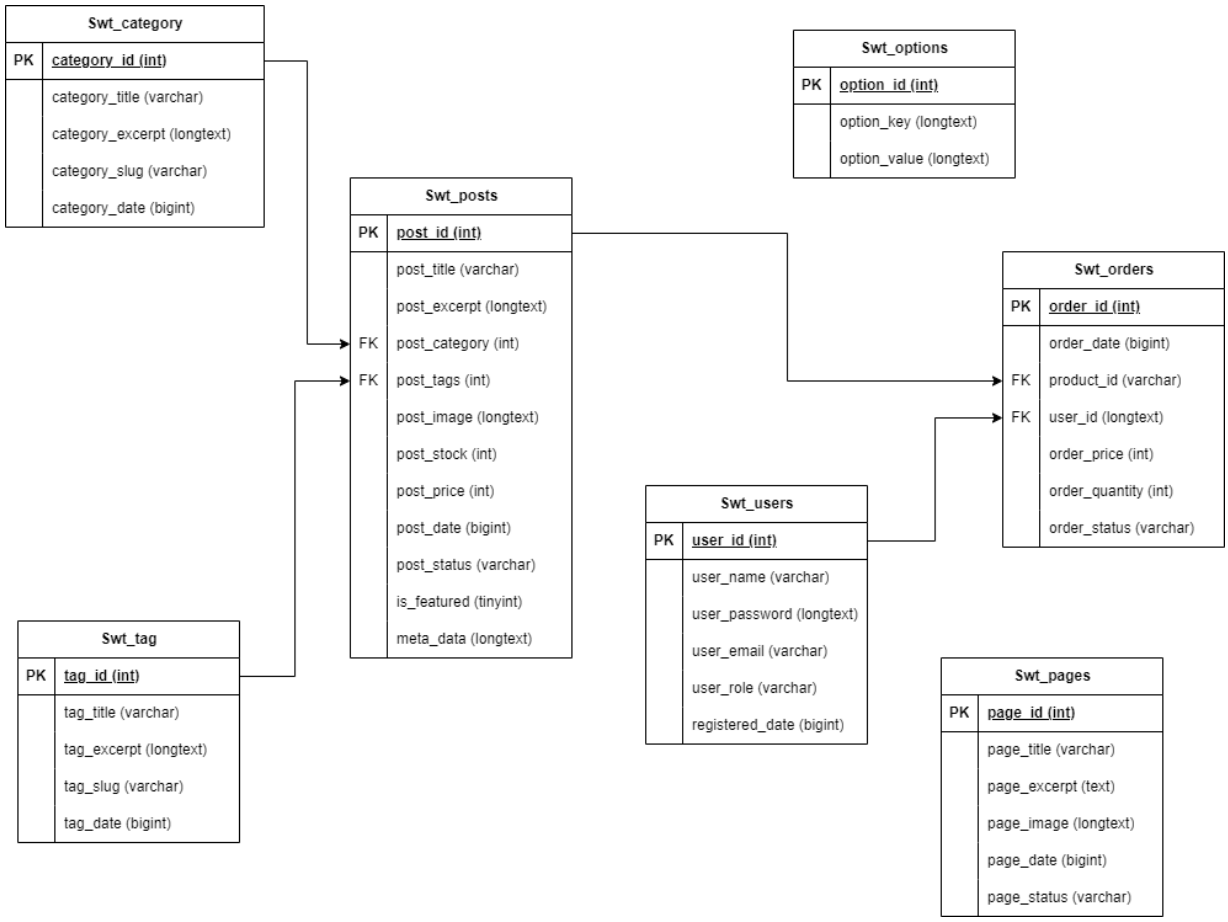


Figure 3.9: Database Schema

3.2.3 Interface Design

Figma have been used in the design of the application's or system's UI. Both are user-interface tools for producing, modifying, and working with raster and vector pictures. The figures below represent some of the features of the system like theme mode, login system, registration system, review section, etc.

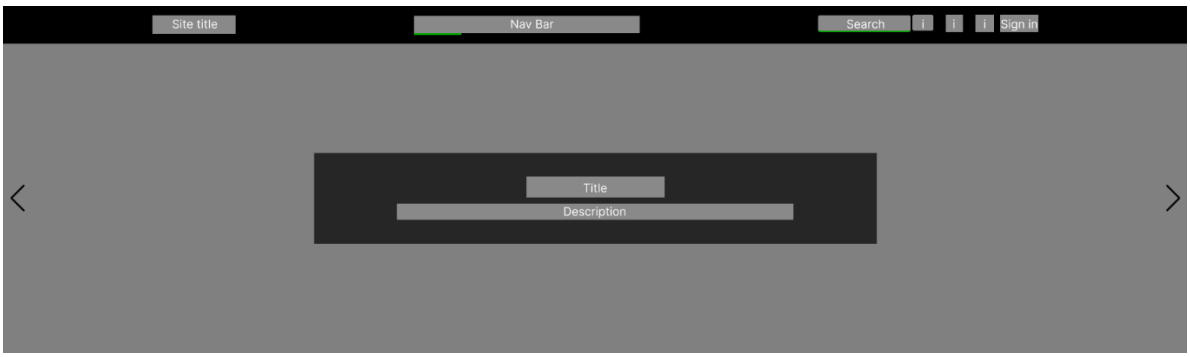


Figure 3.10: Header & Banner

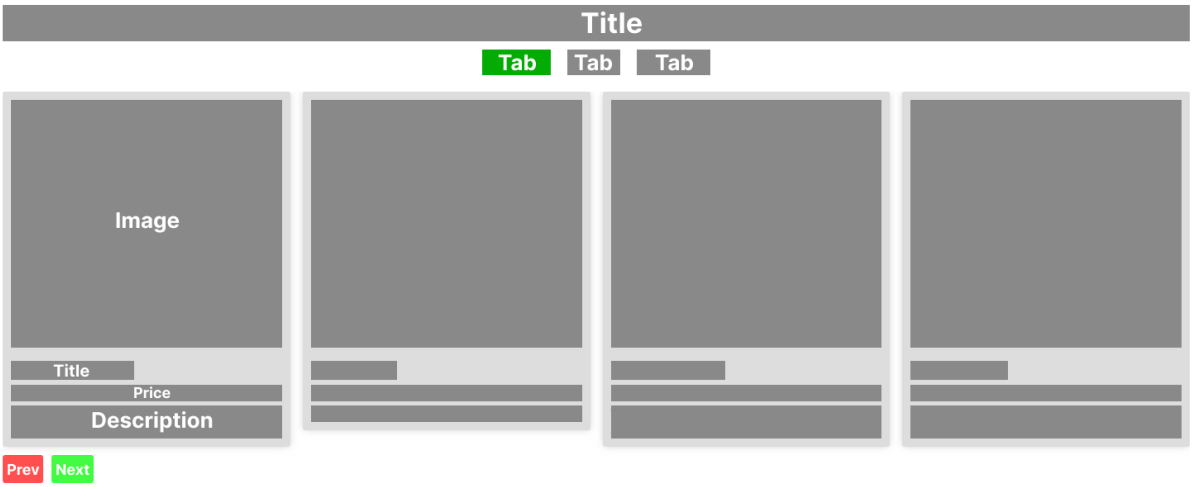


Figure 3.11: Trending Products

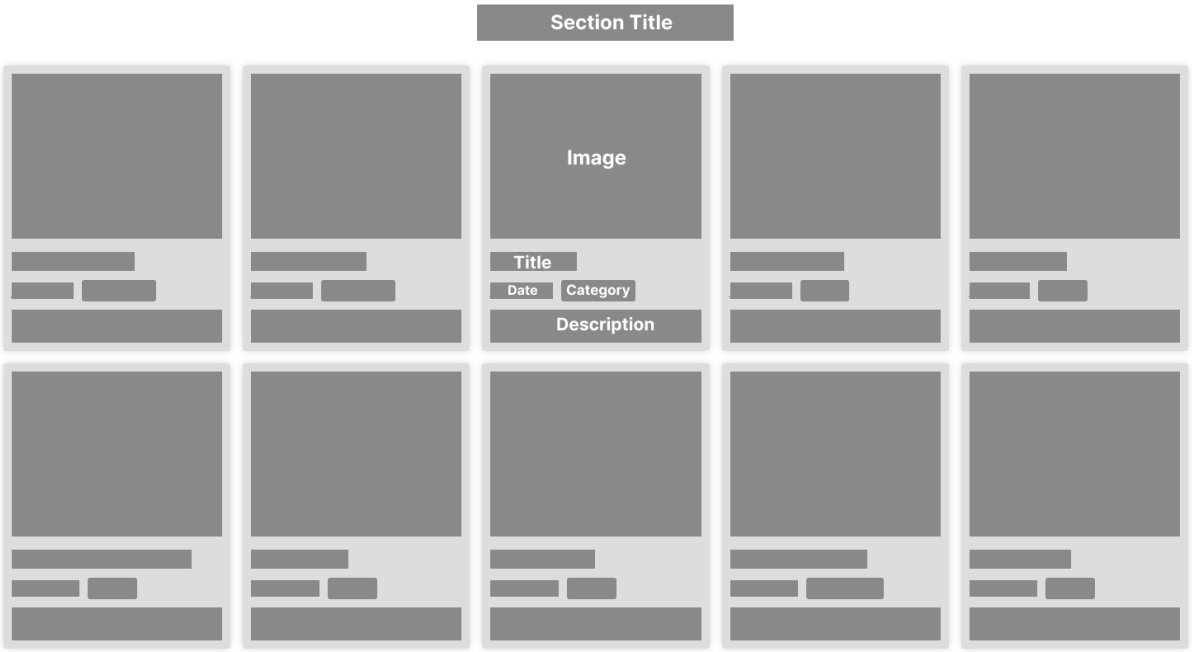


Figure 3.12: Recently Added

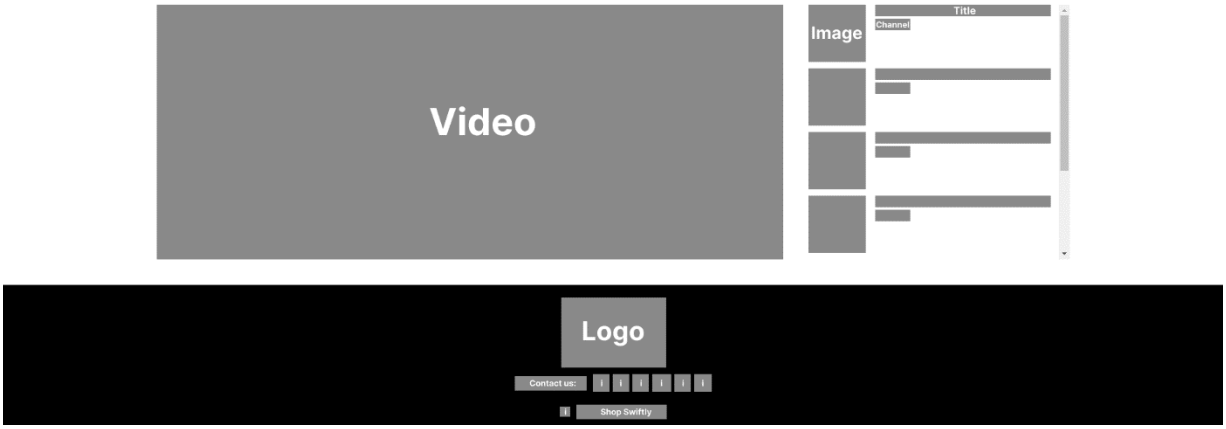


Figure 3.13: YouTube & Footer

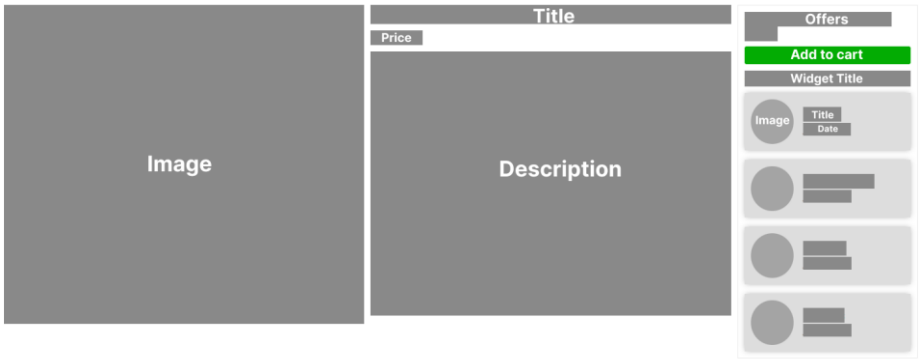


Figure 3.14: Single

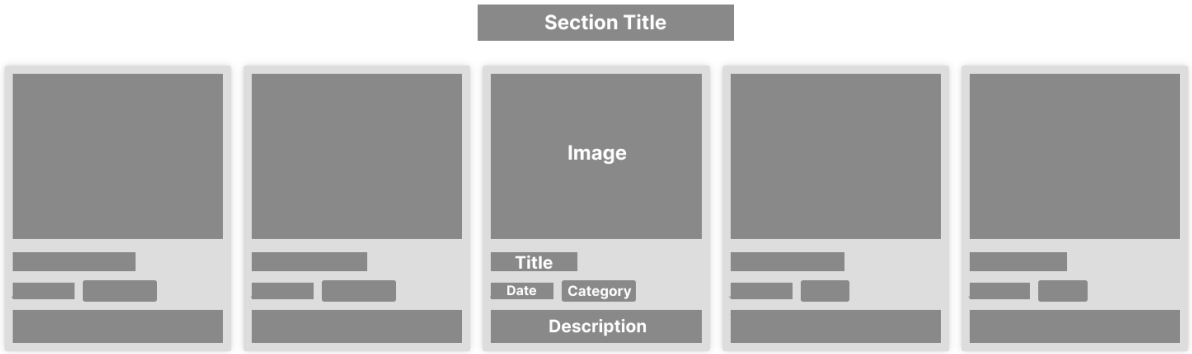


Figure 3.15: Related Products

3.3 Algorithms

- **Sorting Algorithm:** This algorithm is being used in the dashboard and website of this project. In the dashboard side, it is used to filter the report based on month, week and day. While on the website end, it is being used to filter the latest, featured and best-selling products.
- **Pagination Algorithm:** This algorithm is being used in the website's grid layout to load next or previous set of data.
- **Live Search Algorithm:** This algorithm filters out the data that matches the users search query in real time. Users don't have to wait for the page to load. The data will load as they type.

Chapter 4: Implementation & Testing

4.1 Implementation

During the implementation phase, design specifications such as database schema, DFD, interface design were transformed into a functioning software application.

The testing phase begins after the implementation phase has ended. During this phase, codes were tested. Whether or not the modules were implemented as planned during the system design and analysis phase.

4.1.1 Tools used

This section describes about the tools used during this project. The tools used are categories into three groups.

4.1.1.1 Programming Languages

Programming languages is further divided into two categories frontend and backend which have different set rules and implementation.

i. **Frontend**

- **HTML**

  HTML is the structural foundation of this project. About 0.7% of all codes in this project are purely HTML only. It is used to define layouts and organize the contents. Also used with CSS and JS to create interactive and visually attractive UI.

- **CSS**

  CSS plays the fundamental role in defining how the website looks. All the designing were done using CSS. It covers about a quarter, exactly 22% of all codes used in the project. It is also used to make the system responsive, create grids and list layouts.

- **React**

  Instead of using pure JavaScript, React, a library of JavaScript, was used because it has a strong community support and it is platform independent. It is used to communicate with database using its fetch functions. JS libraries such as Chart Js and Font awesome were used for user friendliness. It covers 69% of all codes in the project.

ii. **Backend**

- **PHP**

Instead of using backend frameworks like Django and Node.js with Express, PHP was used because I was most familiar with it. PHP covers exactly 7.7% of all codes used in this project. It was used to manage and run queries in the database.

4.1.1.2 Database Platforms

- **phpMyAdmin**

phpMyAdmin was used as the database for this project. It stores all the information that the users and admins provide. It enforces data integrity through constraints and validation rules. This reduces errors.

4.1.1.3 Libraries used

- **Swiper Js**: A library that help add sliders in our projects.
- **Font Awesome**: A simple JavaScript Library that provides icons to use in our projects
- **Chart Js**: A library that helps render charts like pie, bar chart, etc.

4.1.2 Implementation details of modules

Following the completion of the planning, analysis and design phase, we go on to the implementation phase, where the entire system is broken down into modules in order to reduce effort and find any new faults or issues. The report cannot describe every element of the modules, but it will document the major parts of them. Among the modules are a few of these:

- LOGIN

```
// on login button click
const handleOnSubmit = ( event ) => {
     setFormErrors( formValidate( rawInputs ) )
     if( Object.keys( formValidate( rawInputs ) ).length === 0 ) {
           setValidatedInputs( rawInputs )
           const { admin_username, admin_password } = rawInputs
           let whereClause = 'user_password="'+admin_password+'" AND '
           +'user_email="' + admin_username + '"'
           const FORMDATA = new FormData()
           FORMDATA.append( 'action', 'select_where' )
           FORMDATA.append( 'table_identity', 'user' )
           FORMDATA.append( 'where_clause', whereClause )
           fetch('http://localhost/shopswiftly/src/components/admin/inc/data
           base/index.php', {
               method: 'POST',
               body: FORMDATA
           })
           .then(( response ) => response.json() )
           .then(( data ) => setUserData( data ))
           event.preventDefault()
     } else {
           event.preventDefault()
     }
   }
```

This module runs when the user submits their user credentials such as email and password in the login form. It also validates the user inputs. The inputs are submitted using formdata and fetch api. The data argument in then() returns whether or not the user is a verified user. If the user is verified then they are logged in.

- **Registration**

```
/* Handle submitting registration form */
const handleRegistrationFormSubmit = ( event ) => {
   const FORMDATA = new FormData()
   FORMDATA.append( 'action', 'insert' )
   FORMDATA.append( 'table_identity', 'user' )
   FORMDATA.append( 'post_type', 'user' )
   FORMDATA.append( 'user_name', username )
   FORMDATA.append( 'user_password', password )
   FORMDATA.append( 'user_email', email )
   FORMDATA.append( 'user_role', role )
   FORMDATA.append( 'registered_date', Date.now() )
   fetch('http://localhost/shopswiftly/src/components/admin/inc/database/inde
     x.php', {
     method: 'POST',
     body: FORMDATA
   })
   .then(( result ) => result.json())
   .then(( data ) => setIsRegistered( data.length > 0 ? true : false ))
   event.preventDefault()
}
```

This module runs when the user submits their user credentials such as email, name and password in the registration form. Formdata class is used to submit the user inputs. Fetch api is used to communicate with the database using the post method. If the returned result has no value, then the registration fails.

- **Add to cart**

```
/* Add to cart */
export const AddToCartButton = ( props ) => {
    const { productId: ID } = props
    const { loggedIn, parsedProductDetails } = useSession()
    /* Handle click event */
    const handleClick = () => {
        if( loggedIn && ! parsedProductDetails.includes( ID ) ) {
            sessionStorage.setItem( 'productDetails', JSON.stringify([
...parsedProductDetails, ID ]) )
        }
    }
    return <div className='add-to-cart'>
        <button onClick={ handleClick }>{ 'Add to cart' }</button>
    </div>
}
```

This module renders the add to cart button in the frontend side which has a click event binded to it. When clicked the products id is stored in the session to display the product in the cart component.

- **PDF download**

```
/* Download pdf button */
const DownloadButton = () => {
    const context = useContext( REPORTCONTEXT )
    const { reportHTML } = context
    /* On click download pdf of sales report */
    const downloadPdf = () => {
        const doc = new jsPDF( 'l', 'mm', 'a4' );
         doc.setFontSize(20);
        // Using html method of jsPDF to directly convert HTML to PDF
        doc.html( reportHTML.current, {
            callback: function (doc) {
                doc.setFontSize(10);
                doc.save('report.pdf'); // Save the PDF
            },
            margin: [10, 10, 10, 10], // Optional margins
            html2canvas: {
              scale: 0.5, // 0.5 to reduce zoom
            }
        });
    }
    return <button className='download-button' onClick={ downloadPdf }>
        <FontAwesomeIcon
            icon = { faDownload }
            className = 'download-icon'
        />
        <span className='download-label'>{ 'Download PDF' }</span>
    </button>
}
```

This module renders the download pdf button in the dashboard. It has a click event binded to it. When clicked, a file called report.pdf is downloaded which contains the orders of the current user. The click event uses useContext hook of react and a third-party library called jsPDF which allows for the download. The font-size, margin, etc. are all defined here.

4.2 Testing

Now that the system has been designed, examined, and put into operation, it is time to test it for flaws and faults. The testing step of the system development process is essential since it reduces errors and aids in system troubleshooting. Although there are many various kinds of tests that can be done, for the sake of this report, I will mostly concentrate on the system's functionality, performance, and security.

4.2.1 Test case for unit testing

- **User Login Test Case**

Table 4.1: Login Test Case for User

ID	Test Case Description	Test data	Expected Result	Actual Result	Pass / Fail
1	Wrong username	Username: test@1234 Password: Amir12345	Username does not exist	As expected	Pass
2	Wrong password	Username: Amir12345 Password: test@1234	Password does not match	As expected	Pass
3	Valid username and password	Username: Amir12345 Password: Amir12345	Login	As expected	Pass

- **User Registration Test Case**

Table 4.2: Registration Test Case for User

ID	Test Case Description	Expected Result	Actual Result	Pass / Fail
1	All form fields filled and valid	Successfully registered	As expected	Pass
2	Empty one or more form fields	Please fill all the required fields	As expected	Pass
3	Already used username	Username already in use	As expected	Pass

- **Admin Login Test Case**

Table 4.3: Login Test Case for Admin

ID	Test Case Description	Expected Result	Actual Result	Pass / Fail
1	Wrong Username	Email do not exist	As expected	Pass
2	Wrong Password	Password does not match	As expected	Pass
3	Login using subscriber details	Redirect to user dashboard	As expected	Pass

4.2.2 Test case for system testing

- Verify that the password is stored in the database via md5 or password hashing.
- Verify that phone numbers are 10 characters long or not and are numeric or not.
- Verify that the emails are valid or not by filtering the emails
- Verify that for better UI interface and user satisfaction, form fields contain dropdowns, radio

buttons, checkboxes, sliders, etc.

- Verify that on form submit data are stored in the database properly.
- Verify that the session variables are cleared after logout
- Verify that all required form fields are filled.

Chapter 5: Conclusion & Future Recommendation

5.1 Lesson Learnt / Outcome

This project taught me many things on personal and professional level. On personal level, I learned to work with patience, improved my problem-solving skill and programming logics. On the professional side, I learned that there are many aspects to keep in consideration during the development of the project like accessibility (people with physical disability should also be able to use website), internationalization (website should be free of any language barriers), security, etc. In order to receive a positive response from the customer, a project needs to be finished by the specified timeframe. Communicate with users on a regular basis to learn about their needs and requirements.

5.2 Conclusion

My goal was to create a system that would enable users to order their desired products whenever they wanted, from anywhere. The users can view various products, read about them, see their price information, etc. These objectives have been met by the current application. We completely adhered to the requirements but, when necessary, we extended some of the features. The foundation of the application and this project has been accomplished with the targets met. Building this web application has been tough and rewarding because I've learned a lot about PHP, jQuery, React, and what it takes to create a completely functional website while working on the project. There have been difficulties, particularly with the backend and ensuring that application replies are predictable. The task was made simpler by thorough planning because it required thoughtful thought.

For this project, React was used because of its strong performance and cross platform development. We can use React by downloading Node Js and we are good to go.

5.3 Future Recommendation

In order to enhance this website's mobility, usability, and user experience, the following can be added in the future. Although completing all of these will require more time, finances, research, and study, it is still highly achievable and doable.

- Adding FAQ section
- Adding custom feedback and reviews section

- Adding comments section
- Allowing users to rate products.

References

[1] WordPress, "WordPress," WordPress, [Online]. Available: https://wordpress.org/.

[2] Shopify, "Shopify," Shopify, [Online]. Available: https://www.shopify.com/.

[3] Joomla, "Joomla," Joomla, [Online]. Available: https://www.joomla.org/.

[4] B. Hari, "Article cms bala hari : pulse : linkedin," 11 July 2023. [Online]. Available https://www.linkedin.com/pulse/article-cms-bala-hari.

[5] S. A. F. C. Scott Robinson, "Content Management: TechTarget," April 2024. [Online] Available: https://www.techtarget.com/searchcontentmanagement/definition/content management-system-CMS#:~:text=A%20CMS%20provides%20a%20graphical,content%20delivery%20application%20(CDA)..

[6] A. Volle, "Computers : Technology : Britannica," 05 June 2024. [Online]. Available https://www.britannica.com/technology/content-management-system.

[7] R. Themes, "Scholastica," RoseaThemes Pvt Ltd, [Online]. Available https://roseathemes.com/scholastica/1/my/.

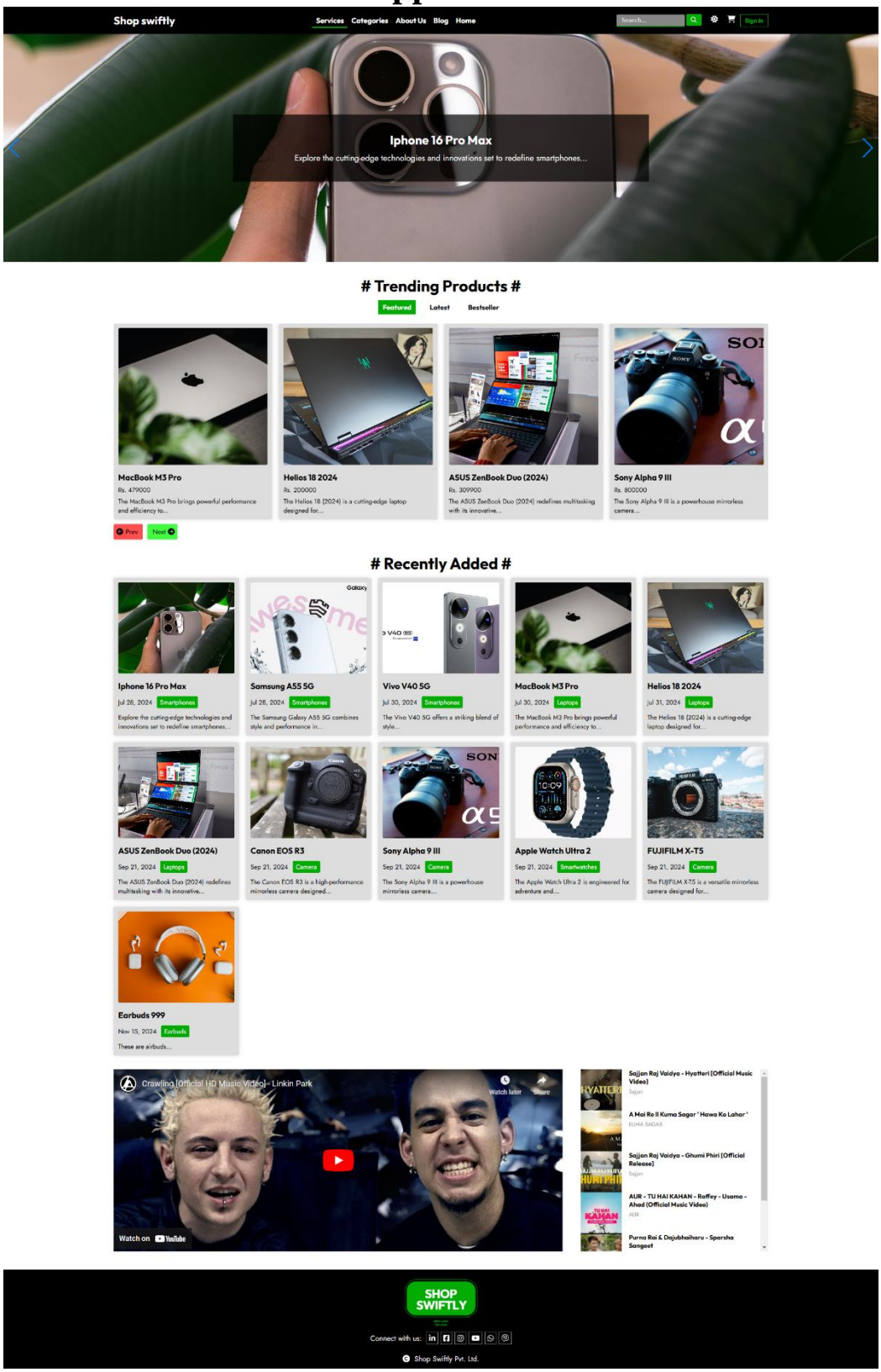
[8] Keenthemes, "Metronic," KeenThemes Pvt. Ltd., [Online]. Available https://keenthemes.com/metronic/tailwind/demo1/.

[9] Pixinvent, "Veuxy," Pixinvent Pvt. Ltd., [Online]. Available https://demos.pixinvent.com/vuexy-html-admin-template/html/vertical-menu-template/.

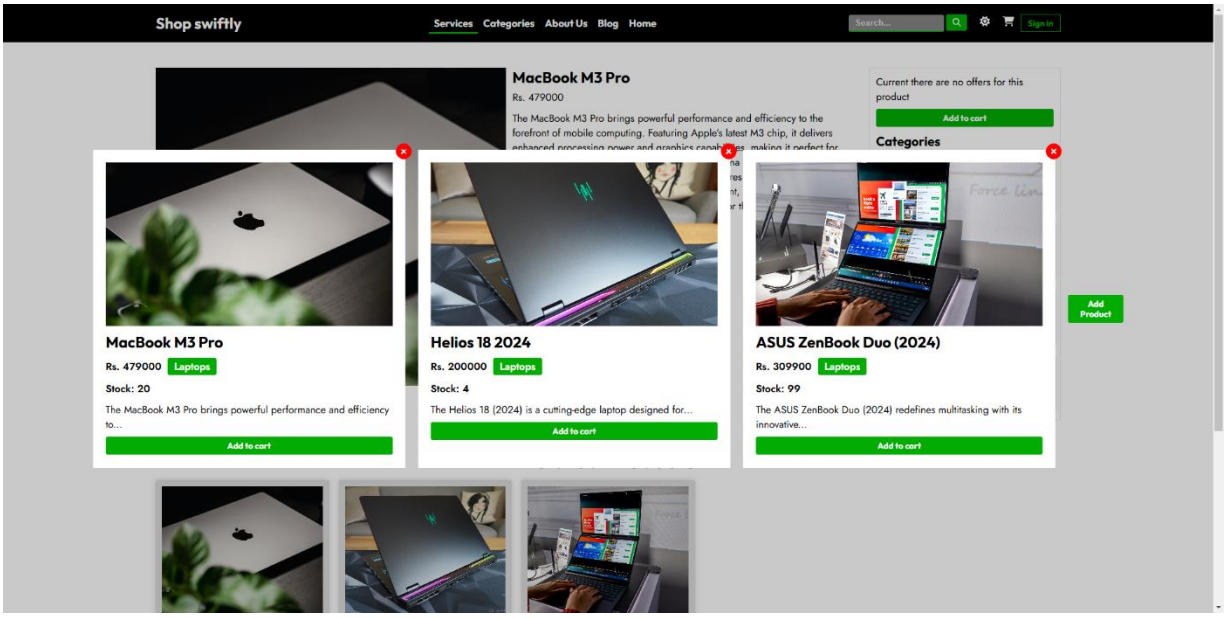
[10] T. Brand, "Velzon," Themesbrand Pvt. Ltd., [Online]. Available https://themesbrand.com/velzon/html/corporate/index.html.

[11] C. Morris, 03 November 2022. [Online]. Available https://www.searchenginejournal.com/what-is-a-content-management-system-cms/468136/.

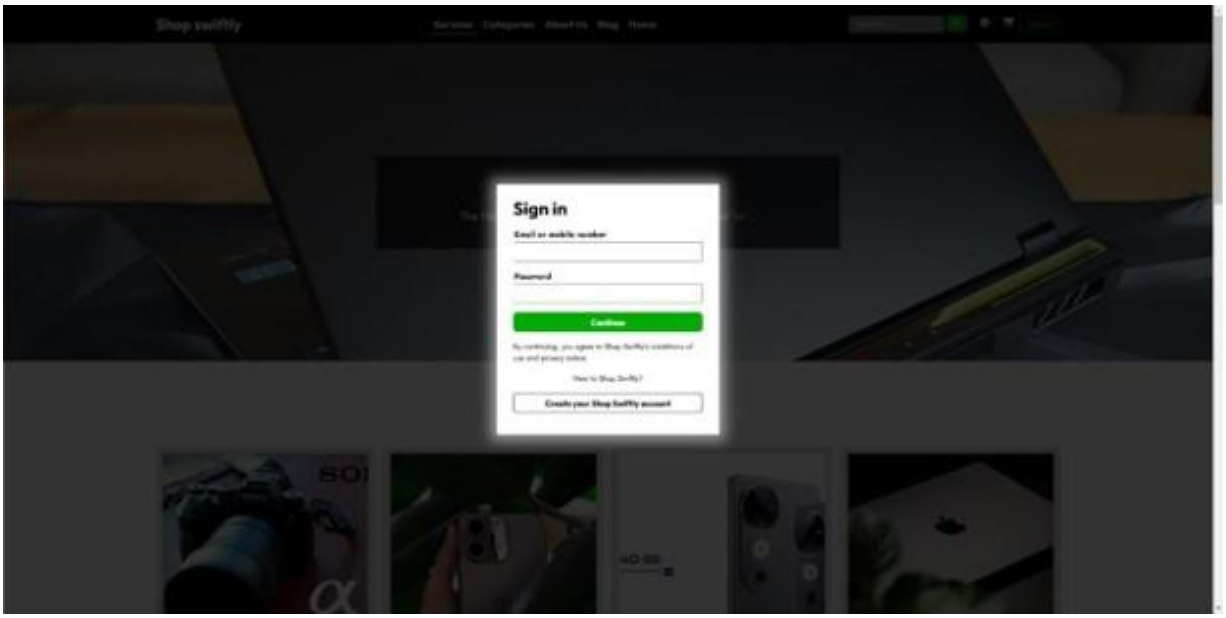
Appendix



Appendix 1: Homepage

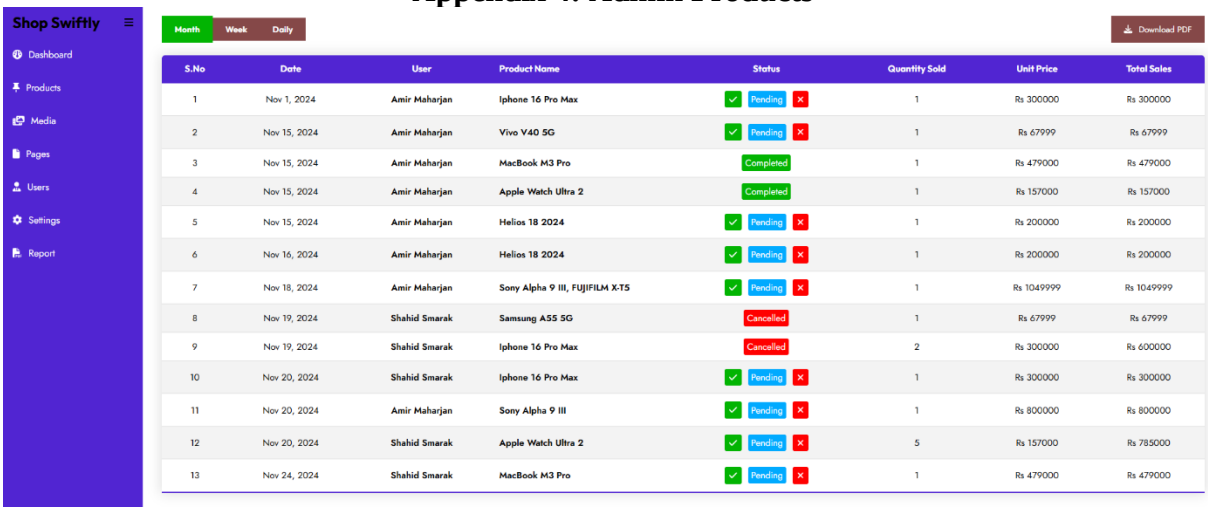


Appendix 2: Compare Products



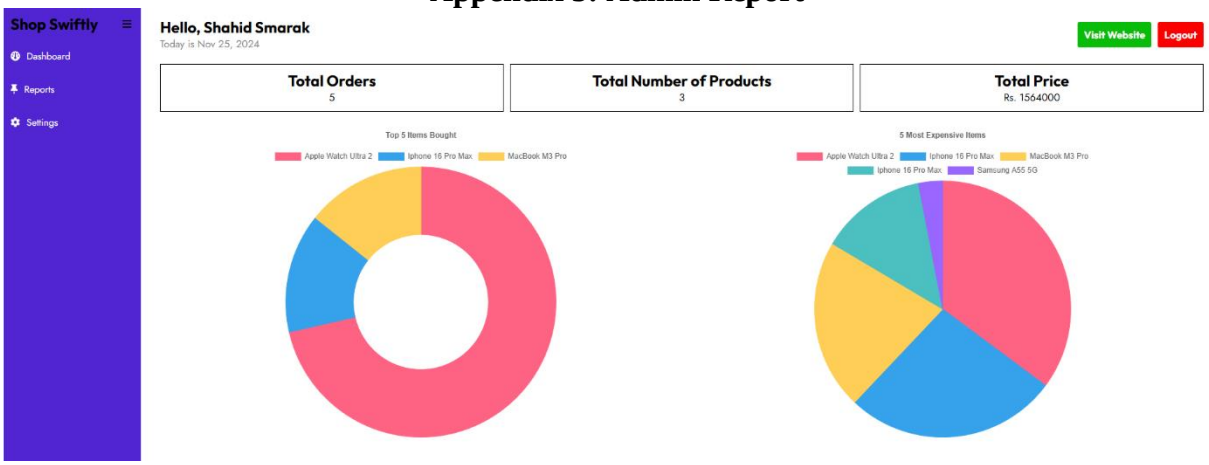
Appendix 3: User Login

Appendix 4: Admin Products

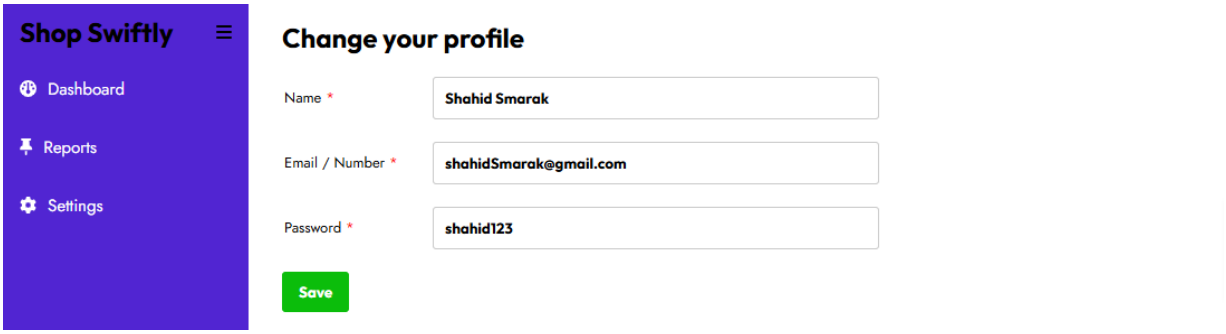


S.No	Date	User	Product Name	Status	Quantity Sold	Unit Price	Total Sales
1	Nov 1, 2024	Amir Maharjan	Iphone 16 Pro Max	Pending	1	Rs 300000	Rs 300000
2	Nov 15, 2024	Amir Maharjan	Vivo V40 5G	Pending	1	Rs 67999	Rs 67999
3	Nov 15, 2024	Amir Maharjan	MacBook M3 Pro	Completed	1	Rs 479000	Rs 479000
4	Nov 15, 2024	Amir Maharjan	Apple Watch Ultra 2	Completed	1	Rs 157000	Rs 157000
5	Nov 15, 2024	Amir Maharjan	Helios 18 2024	Pending	1	Rs 200000	Rs 200000
6	Nov 16, 2024	Amir Maharjan	Helios 18 2024	Pending	1	Rs 200000	Rs 200000
7	Nov 18, 2024	Amir Maharjan	Sony Alpha 9 III, FUJIFILM X-T5	Pending	1	Rs 1049999	Rs 1049999
8	Nov 19, 2024	Shahid Smarak	Samsung A55 5G	Cancelled	1	Rs 67999	Rs 67999
9	Nov 19, 2024	Shahid Smarak	Iphone 16 Pro Max	Cancelled	2	Rs 300000	Rs 600000
10	Nov 20, 2024	Shahid Smarak	Iphone 16 Pro Max	Pending	1	Rs 300000	Rs 300000
11	Nov 20, 2024	Amir Maharjan	Sony Alpha 9 III	Pending	1	Rs 800000	Rs 800000
12	Nov 20, 2024	Shahid Smarak	Apple Watch Ultra 2	Pending	5	Rs 157000	Rs 785000
13	Nov 24, 2024	Shahid Smarak	MacBook M3 Pro	Pending	1	Rs 479000	Rs 479000

Appendix 5: Admin Report



Appendix 6: User Dashboard



Appendix 7: User Settings