

**STEP UP WHOLESALE
PROJECT REPORT**

Submitted by

L. KARTHI

(Reg. No: 22-IT-10)

Under the Guidance of

Mrs. V. Prabavathi M.Sc., B.Ed., M. Phil., NET., (Ph. D)

Nallamuthu Gounder Mahalingam College (Autonomous), Pollachi

Affiliated to Bharathiar University, Coimbatore

in partial fulfilment of the requirements for the award of

BACHELOR OF SCIENCE [INFORMATION TECHNOLOGY]



**DEPARTMENT OF INFORMATION TECHNOLOGY
NALLAMUTHU GOUNDER MAHALINGAM COLLEGE
(AUTONOMOUS)**

(An ISO 9001:2015 Certified Institution)

(Re-Accredited by NAAC with A++ grade)

Pollachi-642 001

APRIL-2025

DECLARATION

DECLARATION

L. KARTHI (Reg. No: 22-IT-10) do here by declare that this project entitled “**STEP UP WHOLESALE**” submitted to the Department of Information Technology, NGM College, Affiliated to Bharathiar University, Coimbatore in partial fulfilment of the requirement for the award of **Bachelor Degree in Information Technology** is a record of original work done by me during the period of study at NGM College, Pollachi, Under the guidance of **Mrs. V. Prabavathi M.Sc., B.Ed., M.Phil., NET., (Ph. D) Assistant Professor, Department of Information Technology.**

PLACE: Pollachi

DATE:

Signature of the Candidate

(L. KARTHI)

CERTIFICATES

CERTIFICATE – I

I hereby declare that the original project work entitle
“**STEP UP WHOLESALE**” is a record work done by **L. KARTHI** under the supervision
and guidance of me.

PLACE: Pollachi

DATE:

Signature of the Guide

Mrs. V. Prabavathi M.Sc.,B.Ed.,M.Phil.,NET., (Ph.D)

CERTIFICATE -II

This is to certify that the project entitled “STEP UP WHOLESALE” is a bonafied record of work done by **L. KARTHI (Reg.No.: 22-IT-10)** submitted in partial fulfilment of the Requirements for the award of the degree Bachelor of Science (Information Technology) under Bharathiar University, Coimbatore, during the academic year 2024-2025 under my supervision.

Signature of the Guide

Signature of the HOD

Counter Signed by

For external viva – voce examinations held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I thank God for giving me good health and wisdom to do my project and complete it successfully. It is my immense pleasure to acknowledge the contributors towards the successful completion of this project.

It is a great privilege and pleasure of mine to express my deep sense of gratitude to **Dr. R. MANICKACHEZIAN M.Sc., M.S., Ph.D.**, Principal, Nallamuthu Gounder Mahalingam College, Pollachi for allowing me to take on this project.

I express my heart full thanks to **Mr. K. VIJAYAKUMAR MCA., M.Phil. SET** Head, Department of Information Technology for his kind help and guidance in completing this project.

I would like to express my deep sense of gratefulness and sincere thanks to my guide **Mrs. V. PRABAVATHI M.Sc., B.Ed., M.Phil., NET.,(Ph. D)** Assistant Professor, Department of Information Technology, NGM College, Pollachi, for her valuable suggestions and timely guidance, which paved way to streamline and who is behind the success of this project and my career.

I also extend my sincere thanks to all faculty members and staff of the **Department of Information Technology, NGM College, Pollachi**, for their valuable assistance and encouragement.

Finally, I am profoundly grateful to my family, friends, and well-wishers for their unwavering support, motivation, and encouragement throughout this project.

CONTENTS

CONTENT

	Page. No.
1. INTRODUCTION	10
1.1. OVERVIEW OF THE PROJECT	
1.2. SYSTEM SPECIFICATION	
1.2.1. HARDWARE SPECIFICATION	
1.2.2. SOFTWARE SPECIFICATION	
1.3. FRONT END-VB.NET	
1.4. BACK END-SQL SERVER	
2. SYSTEM ANALYSIS	15
2.1. EXISTING SYSTEM	
2.1.1. DISADVANTAGES OF EXISTING SYSTEM	
2.2. PROPOSED SYSTEM	
2.2.1. ADVANTAGES OF PROPOSED SYSTEM	
3. SYSTEM DESIGN	16
3.1. DESIGN NOTATION	
3.1.1. DATA FLOW DIAGRAM	16
3.1.2. E R DIAGRAM	18
3.2. DESIGN PROCESS	
3.2.1. INPUT DESIGN	
3.2.2. DATABASE DESIGN	
3.2.3. TABLE DESIGN	
3.2.4. OUTPUT DESIGN	
4 .SYSTEM TESTING AND IMPLEMENATION	26
4.1. SYSTEM TESTING	
4.2. IMPLEMENTATION	
5. CONCLUSION AND FUTURE ENHANCEMENT	30
6. BIBLIOGRAPHY	32
APPENDIX	33
SAMPLE SCREENS	

INTRODUCTION

1.INTRODUCTION

1.1. OVERVIEW OF THE SYSTEM

This project is entitled as “**STEP UP WHOLESALE**”. This project focuses on developing a comprehensive website for a wholesale footwear shop, where the admin can manage users and footwear stock efficiently. The platform enables the admin to create user accounts for retail footwear shop owners. Each retailer is assigned a unique ID and password to log in, view available footwear stock, place orders, and track their purchases.

The admin has complete control over user management, stock inventory, and order tracking, making the platform a centralized solution for managing wholesale transactions. This website simplifies the wholesale process, enhances stock management efficiency, and delivers a seamless experience for both the admin and retail users.

MODULES

ADMIN MODULE

- ▣ User Management
- ▣ Footwear Inventory Management
- ▣ Order Management
- ▣ Report and Analysis
- ▣ Data Access Module

USER MODULE

- ▣ User Authentication
- ▣ Browser and Search Footwear
- ▣ Order Placement
- ▣ Ordre Tracking
- ▣ Feedback and Support

MODULE DESCRIPTION

- **Authentication Module**

This module handles login and registration for users. Retailers use their credentials to securely access the platform and perform operations like browsing and ordering footwear.

- **User Management Module**

Allows the admin to create and manage user accounts. Retailers are registered with unique IDs and credentials, which can be updated or removed by the admin.

- **Footwear Inventory Module**

Enables the admin to add, update, or delete footwear models and stock details. It helps in maintaining real-time inventory visibility.

- **Order Management Module**

Users can place orders through this module. Admin can view all orders and update their status such as confirmed, packed, or delivered.

- **Order Tracking Module**

Lets users track the status of their orders. Admin updates the status, and users can view the progress from their account.

- **Report and Analysis Module**

Generates simple reports for the admin based on sales, users, and stock levels to help in decision-making.

- **Feedback and Support Module**

Retail users can send feedback or support queries. Admin can view and respond to them to improve service quality.

1.2 SYSTEM SPECIFICATION

1.2.1 HARDWARE SPECIFICATION

Processor	: Intel Core i3
RAM	: 4 GB DDR4
Hard Disk	: 500 GB HD / SSD
Clock Speed	: 2.1 GHz

1.2.2 SOFTWARE SPECIFICATION

Technology	: XAMPP
Front End	: HTML, CSS(BOOTSTRAP), JS, AJAX
Back End	: PHP, MySQL
Server	: Apache

1.3 FRONT END- HTML, CSS(BOOTSTRAP), JS, AJAX

The front-end of the application is developed using a combination of standard web technologies such as HTML, CSS, and JavaScript. These technologies together provide a structured, styled, and interactive user interface.

- HTML is used to create the structure and content of web pages. It defines the elements such as forms, buttons, tables, and sections that users interact with.
- CSS is responsible for designing and styling the user interface, ensuring the layout is visually appealing and consistent across pages.
- JavaScript adds interactivity to the web pages. It enables dynamic content updates, form validations, and smooth navigation for a better user experience.
- Bootstrap, a powerful front-end framework, is integrated to ensure mobile responsiveness and a modern UI using ready-to-use components such as modals, navbars, and cards. It simplifies layout design and speeds up development time.

- AJAX (Asynchronous JavaScript and XML) is used to communicate with the server without reloading the entire page. This enhances user experience by allowing actions like placing orders, fetching data, or updating status dynamically in real-time.

1.4 BACK END- PHP, MySQL

The back-end logic and server-side functionalities of the application are developed using PHP, a widely-used scripting language for web development. PHP is responsible for processing user inputs, interacting with the database, handling sessions, and managing server responses.

- PHP handles the core business logic of the application, including user authentication, order placement, inventory updates, and report generation. It receives data from the front end, processes it, and returns appropriate responses.
- The application uses XAMPP as the local development environment, which includes Apache as the web server and MySQL as the database server.
- MySQL is used to store and manage all the data related to the footwear inventory, user credentials, orders, and admin operations. Tables are structured to efficiently handle relational data and ensure data integrity.
- Through structured SQL queries and prepared statements, PHP securely performs Create, Read, Update, and Delete (CRUD) operations.
- The integration of AJAX with PHP allows data to be sent and received asynchronously, reducing page reloads and improving speed and performance.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The current system for managing wholesale footwear transactions is largely manual, relying on traditional methods such as, Stock is managed through physical records or spreadsheets, which are prone to human error. Retailers place orders via phone or email, leading to potential miscommunication and delays. Admins manually manage retailer details, which makes it difficult to handle large numbers of users efficiently. Retailers do not have a platform to view stock in real-time, check availability, or track their orders.

2.1.1 DISADVANTAGES OF EXISTING SYSTEM

- Inefficiency
- Human Errors
- No Real-Time Updates
- Miscommunication
- Scalability Issues

2.2 PROPOSED SYSTEM

The proposed system introduces a web-based platform for the wholesale footwear business. It aims to streamline operations through automation and real-time data management. Key features include, Admin functionalities such as, Create and manage retailer accounts with unique credentials. Add, update, and delete footwear stock with real-time availability updates. Manage and track orders placed by retailers. Retailer functions such as, Secure login with admin-provided credentials. Browse footwear stock, view detailed product information, and check availability. Place orders online and track order status in real-time. Access order history and manage account details.

2.2.1 ADVANTAGE OF PROPOSED SYSTEM

- Efficiency
- Error Reduction
- Real-Time Updates
- Improved Communication
- Scalability

3. SYSTEM DESIGN

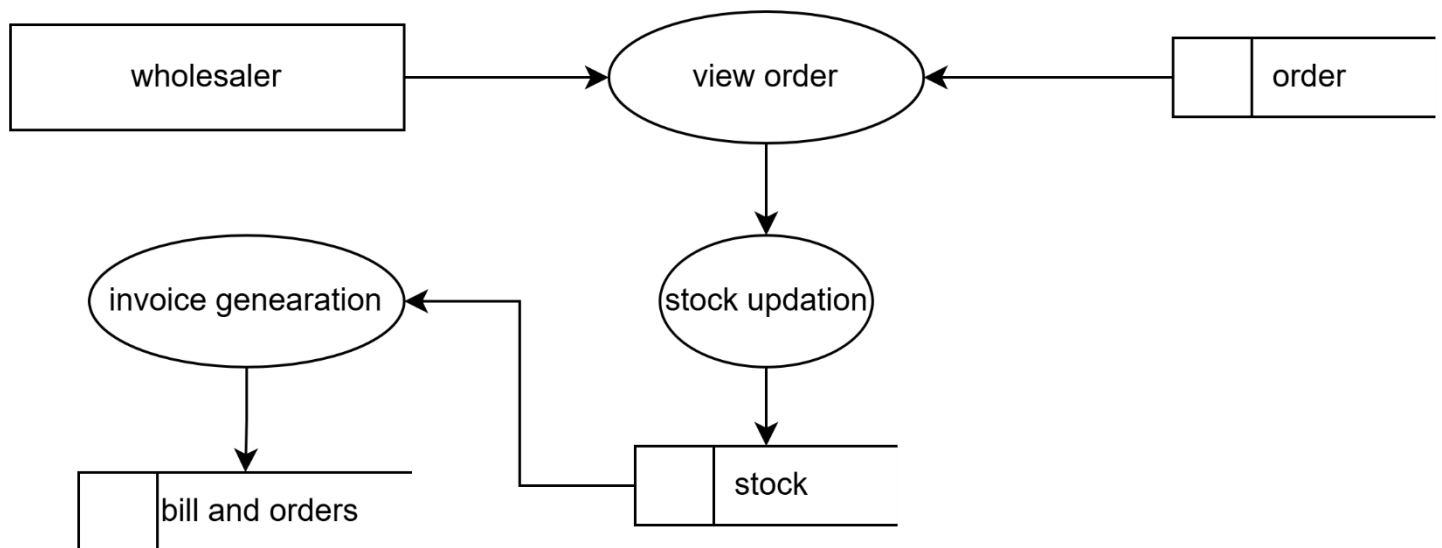
3.1 DESIGN NOTATION

3.1.1 DATAFLOW DIAGRAM

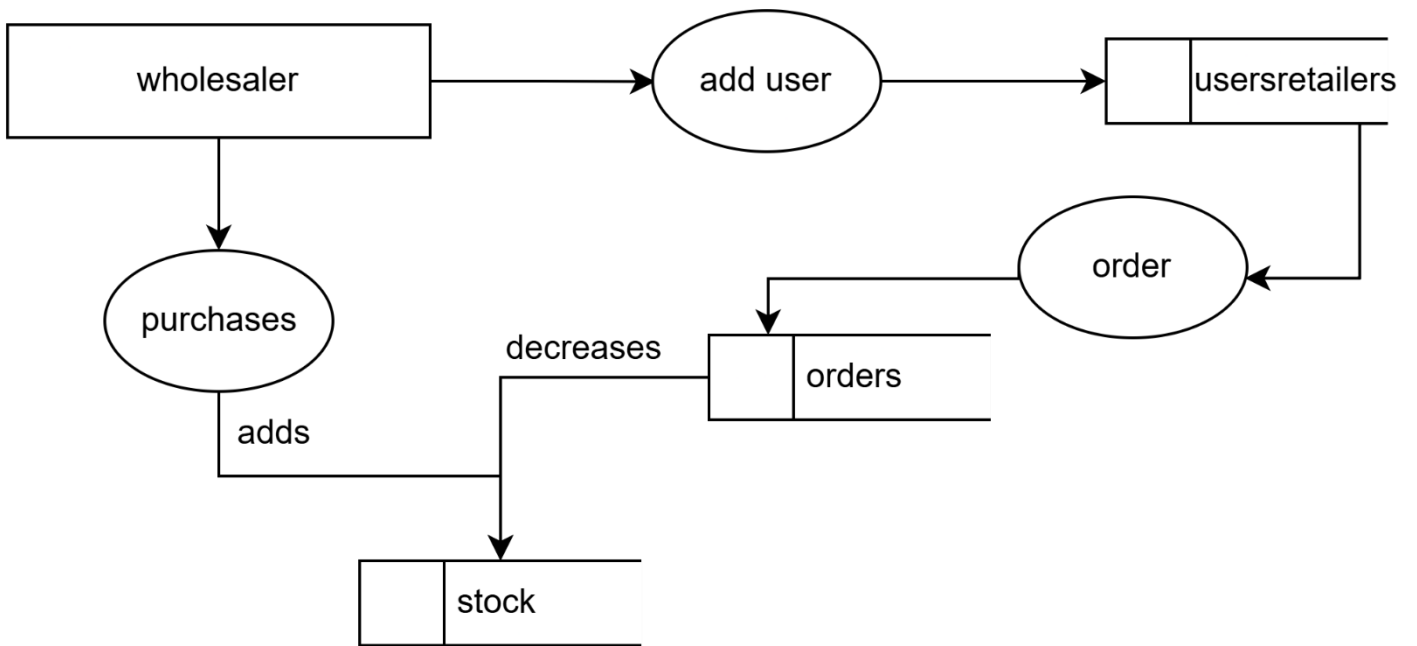
Level 0



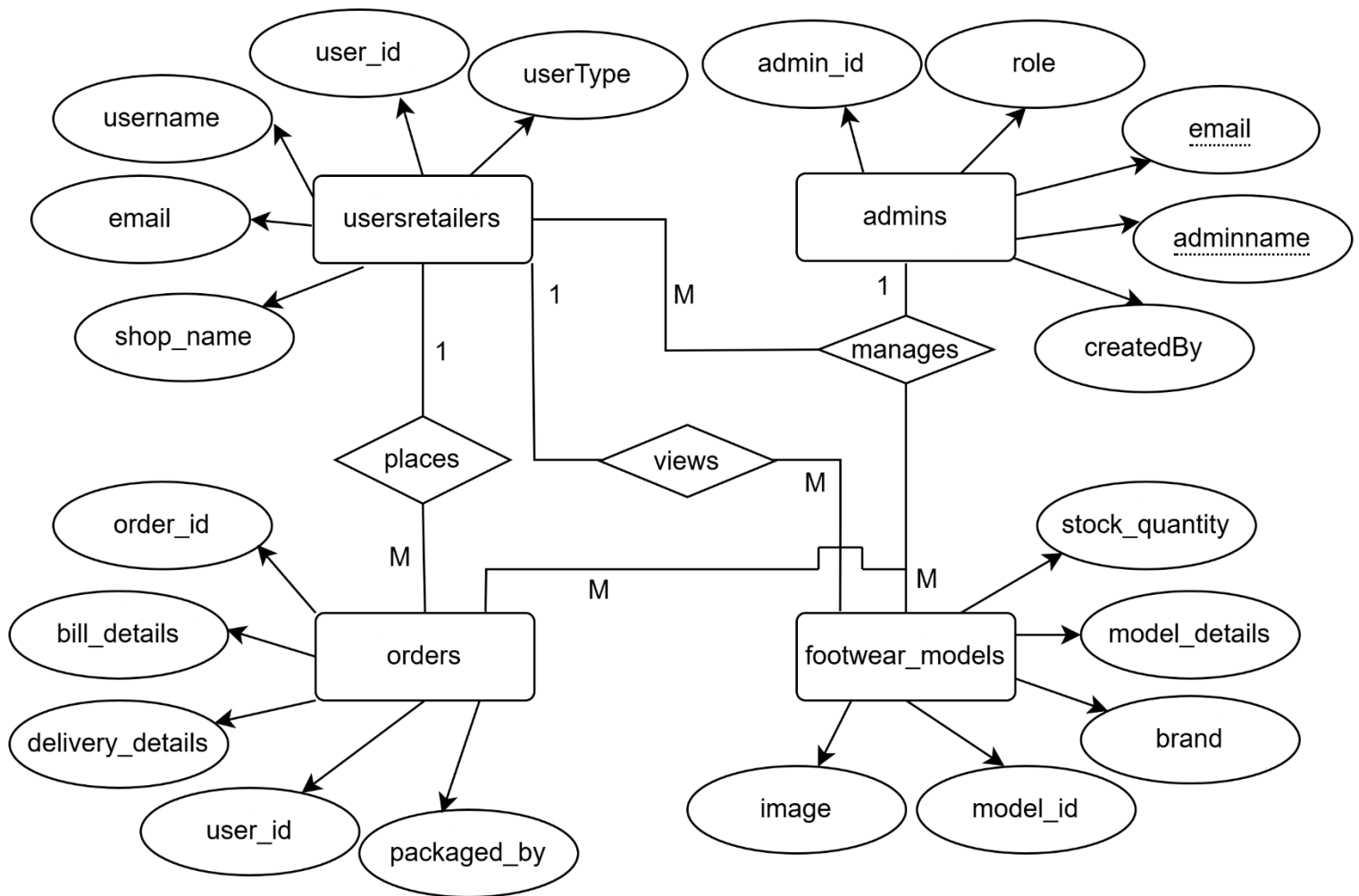
Level 1



Level 2



3.1.2 ENTITY RELATIONSHIP DIAGRAM



3.2 DESIGN PROCESS

System design refers to a structured and methodical approach to creating a system. It involves analyzing all the components required, including the system architecture, the necessary hardware and software, user interfaces, and data flow. Whether following a top-down or bottom-up methodology, the focus is to ensure that each element works cohesively to meet the functional and non-functional requirements of the system. In this project, design considerations included user interaction, order management, stock control, and database efficiency. The design process overlaps with systems analysis and systems architecture to produce a scalable and maintainable system.

3.2.1 INPUT DESIGN

Input design is the process of converting user inputs into a structured format that the system can interpret. Good input design ensures data accuracy, efficiency, and usability. For the

Step-up Wholesale application, input forms are designed to be user-friendly, responsive, and secure.

Key input forms used in the project include:

- **User Registration Form** – Collects essential details from retail shop owners such as name, shop name, email, phone number, and password to create user accounts.
- **Login Form** – Allows registered users (both admin and retail shop owners) to securely log in using their email and password.
- **Order Placement Form** – Enables users to select footwear models, specify quantity, and place orders from the available stock.
- **Product Management Form (Admin only)** – Allows the admin to add new footwear models, update stock quantities, and manage product information.
- **User Management Form (Admin only)** – Lets the admin view, update, or delete retail user details as necessary.
- **Order Status Update Form (Admin only)** – Enables the admin to update order and

delivery status (e.g., Processing, Shipped, Delivered)

3.2.2 DATABASE DESIGN

Database design plays a critical role in efficiently storing and retrieving data in the application. A well-structured database ensures data consistency, integrity, and scalability. The Step-up Wholesale system uses a relational database (MySQL) to manage information related to users, products, orders, and admin operations.

The key tables designed in the system include:

- **Users Table** – Stores information about registered users, including user ID, name, email, phone number, password, and user role (admin or retailer).
- **Products Table** – Contains details about footwear models such as product ID, name, size, price, available quantity, and image path.
- **Orders Table** – Records user orders with fields for order ID, user ID, product ID, quantity, order date, and status.
- **User cart Table** – Stores models in Users Cart
- **Admin Table** – Stores admin credentials and logs admin actions for audit purposes.

3.2.3 TABLE DESIGN

1) **Table Name:** usersretailers

Primary Key: user_id

Field Name	Data Type	Size	Description
User_id	Int	11	Registered user ID
Username	Varchar	100	Registered Username
Password	Varchar	255	Hashed password of User
Shop_name	Varchar	255	Shop name of the user
Shop_address	Varchar	255	Shop address of the user
Email	Varchar	255	Email ID of the user
Mobile_number	Varchar	15	Mobile number of the user
Otp	Varchar	6	Otp if the user tried to log in
Otp_expiry	Datetime	--	Otp expiring date and time
Is_email_varified	Tinyint	1	Is the user loggedIn and verified
Owner_name	Varchar	255	Owner of the Shop
UserCreationDate	Datetime	--	User's created date and time
LastLogin	Datetime	--	User's recent login date and time
UserType	Enum	--	Type of the user
Status	Enum	--	User's Current status

2) Table Name: admins

Primary Key: admin_id

Field Name	Data Type	Size	Description
Admin_id	Int	11	ID of the admin
Adminname	Varchar	100	Name of the admin
Email	Varchar	255	Email of the admin
Password	Varchar	255	Hashed password of the admin
Mobile_number	Varchar	15	Mobile number of the admin
Role	Enum	--	Role of the admin
CreatedDate	Datetime	--	Admin created date and time
Created_by	Int	11	Denotes the id of admin who created
Lastlogin	Datetime	--	Admin's recent login date and time
Status	Enum	--	Status of Admin
Otp	Varchar	6	Otp if admin tried and time
Otp_Expiry	Datetime	--	Otp expiring date and time
Is_email_verified	Tinyint	1	Is the admin loggedin and verified

3) Table Name: footwear_models

Primary Key: model_id

Field Name	Data Type	Size	Description
Model_id	Int	11	Id of the model
Main_brand	Varchar	255	Main Brand of the model
Sub_brand	Varchar	255	Sub brand of the model
Commodity	Varchar	50	Commodity of the model
Article	Varchar	50	Article of the model
Color	Varchar	50	Color of the model
Price	Decimal	10,2	Price of the model
Type	Varchar	50	Type of the model
Material	Varchar	50	Material of the model
Stock_available	Varchar	50	Availability of the model
Image_name	Varchar	255	Image name of the model
Image_type	Varchar	100	Type of the image
Image_data	LongBlob	--	Image data

4) Table Name: Orders

Primary Key: Order_id

Foreign Key: User_id, Packaged_by

Field Name	Data Type	Size	Description
Order_id	Int	11	Id of the order
User_id	Int	11	User's id who placed the order
Order_place_time	Datetime	--	Order placed time
Order_updated_time	Datetime	--	Order updated time
Order_packaged_time	Datetime	--	Order Packaged time
Bill_amount	Decimal	10,2	Amount of the bill
Bill_file	Varchar	255	Path to the Bill invoice's PDF file
Delivery_status	Enum	--	Status of the delivery
Delivered_time	Datetime	--	Order delivered time
Estimate_delivery_date	Datetime	--	Estimated delivery date

5) Table Name: user_cart

Primary Key: Cart_id

Foreign Key: user_id, model_id

Field Name	Data Type	Size	Description
Cart_id	Int	11	The ID of the Cart
Model_id	Int	11	ID of the model in the cart
User_id	Int	11	ID of the cart holder user
Quantity	Int	11	Quantity of the model in cart

3.2.4 OUTPUT DESIGN

Output design focuses on presenting the results and responses generated by the system in a clear, accessible, and meaningful way to the end users. In any application, outputs are the primary means through which users interact with the system's results, making this one of the most important aspects of system development. Properly designed outputs help users make informed decisions and improve the system's usability and effectiveness.

In the **Step-up Wholesale** application, the output design ensures that both the admin and retail shop users receive timely, visually organized, and relevant information. Outputs are presented via dynamic web pages using PHP, Bootstrap, and AJAX, ensuring responsiveness and clarity.

The major output components in this system include:

- **Order Confirmation Page** – Displays order details after successful placement, including product names, quantities, total cost, and expected delivery timeline.
- **Invoice Generation (User Side)** – Generates a downloadable PDF invoice for each order placed, with detailed item breakdown and user/shop details.
- **Stock Overview (Admin Side)** – Shows a summarized view of available products, stock levels, and alerts for low-stock items.
- **Order Status Display** – Provides real-time status updates to users on their orders (e.g., Pending, Approved, Shipped, Delivered).
- **User List & Activity Logs (Admin Side)** – Outputs detailed lists of registered users, their order history, and actions taken for transparency and management.
- **Dashboard (Admin and User)** – A summarized graphical interface that displays key data such as total orders, total users, current stock, and order fulfillment rates.

4. SYSTEM TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

System testing is a crucial process in ensuring that the application performs according to its specifications and meets the end users' needs. It involves validating and verifying that every module, feature, and functionality is working as intended. Testing is an ongoing activity that should be performed alongside development, covering analysis, design, implementation, and maintenance.

For **Step-up Wholesale**, multiple types of tests were conducted to ensure the reliability, security, and performance of the system. Test cases were created with detailed steps, expected outcomes, and pass criteria

Types of Testing Performed:

1. **Functionality Testing** – Ensuring each module (e.g., registration, login, order placement, admin dashboard) works correctly.
2. **Security Testing** – Ensuring secure login, data access control, and protection against unauthorized actions.
3. **Performance Testing** – Checking response times and system behavior under concurrent usage.
4. **Usability Testing** – Verifying user-friendly interface for retail shop owners and admins.
5. **Database Testing** – Verifying correct data insertions, updates, deletions, and validations.
6. **Backup & Recovery Testing** – Ensuring that order data and invoices can be recovered in case of system failure.

Test Case ID	Test Case Description	Objective	Test Steps	Expected Result	Pass Criteria
SEC-01	Secure User Login	Ensure that only registered users can access their accounts.	1. Attempt login with correct credentials. 2. Try with invalid details.	Only valid users can log in; others are denied.	Unauthorized users blocked, valid users logged in.
SEC-02	Admin Access Restriction	Verify that only admin can access the admin dashboard.	1. Log in as a user. 2. Attempt to access the admin panel.	Regular users are denied admin access.	Admin-only routes are protected.
FUNC-01	Order Placement	Ensure that a user can place an order with valid cart data.	1. Add items to cart. 2. Click on "Place Order".	Order is placed and saved in the database with a unique order ID.	Confirmation shown and order appears in admin dashboard.
FUNC-02	Invoice Generation	Verify invoice is generated after order is placed.	1. Place an order. 2. Check if invoice is downloadable.	Invoice contains order details, user info, and pricing.	Invoice contains order details, user info, and pricing.
FUNC-03	Stock Update (Admin)	Ensure stock is correctly updated when admin adds/edits items.	1. Log in as admin. 2. Add a new product or update existing stock.	Changes reflect immediately in product list.	Changes reflect immediately in product list.
PERF-01	Page Load Speed (Cart, Products)	Measure the speed of critical pages.	1. Load user dashboard, cart, and product listing.	Pages should load under 2 seconds on average.	Load time \leq 2 seconds.

PERF-02	Multiple User Access	Verify system performance with multiple concurrent users.	1. Simulate 100+ users browsing and placing orders.	No major lag or crash occurs; server handles requests.	System stable with concurrent use.
DB-01	Database Consistency	Ensure data integrity in user, order, and product tables.	1. Insert and update data via UI. 2. Validate in phpMyAdmin.	Data matches with UI operations.	Data is stored, updated, and retrieved accurately.
USAB-01	Responsive Design (Mobile/Desktop)	Verify that UI is user-friendly on all devices.	1. Access the site on various screen sizes.	Bootstrap layout adjusts properly.	Mobile and desktop users can navigate easily.
BKP-01	Backup and Restore of Order Data	Verify that data is backed up and recoverable.	1. Export database. 2. Simulate data loss. 3. Import backup.	Order data is restored from backup.	Orders recovered with no loss.

4.2 SYSTEM IMPLEMENTATION

System implementation refers to the process of putting the developed software system into actual use after development and testing phases. It is a post-development activity that ensures the transition of the system from development to production. The process involves requirement analysis, system configuration, customizations, data migration, user training, and final deployment.

In the case of **Step-up Wholesale**, the implementation process was carefully planned and executed to ensure the smooth onboarding of retail users and administrators. The web-based platform, built using PHP and MySQL with XAMPP as the local server stack, was designed to be simple, responsive, and secure.

Before implementation, the project team reviewed all functional and non-functional requirements. This included ensuring that the platform supports user registration, order management, product stock updates, secure login, and invoice generation.

The project was deployed using XAMPP, and the codebase was placed within the Apache htdocs directory. Database schemas were created using phpMyAdmin, and proper relationships were established between tables to maintain data consistency.

The interface was customized for ease of use by retail shop owners. Bootstrap and AJAX were used to enhance responsiveness and interactivity. The admin dashboard was implemented to support real-time order and user management.

Role-based access was configured:

- **Admins** can add/update footwear models, manage stock, view orders, and change order statuses.
- **Retail Users** can browse products, add items to cart, and place orders.

A brief walkthrough was provided to both admin and user roles on how to use the system. This included training on order placement, dashboard navigation, and invoice handling.

Continuous interaction was maintained with potential users to ensure the system addressed their needs. Feedback during testing was used to tweak the UI and backend logic before full-scale deployment. Care was taken to minimize the communication gap between developers and users. Functionalities were explained in simple terms, and system behavior was continuously aligned with user expectations. Active involvement of the shop owner during the implementation ensured better understanding of business flow. Frequent interactions helped in refining the inventory and order features.

After successful testing, the system was made live on a local server. Sample data was used to validate all core modules, including product browsing, ordering, stock updates, and invoice generation

5. CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

The goal of the **Step-up Wholesale** project was to design and implement a user-friendly, secure, and efficient online platform that allows retail shop owners to place orders from a wholesale footwear store. The system was developed using PHP, MySQL, Bootstrap, and AJAX, and thoroughly tested with sample data to ensure accuracy and functionality.

This application streamlines the wholesale process by allowing users to browse available footwear models, place orders, and track their status — all in a simple and intuitive interface. Admins have access to powerful tools for managing stock, updating delivery status, and maintaining user accounts.

Security measures were taken to ensure that only authorized users can access sensitive features, making the system reliable and safe. With minimal typing required, users can interact with the system using easy-to-navigate menus and options. The modular structure of the system also ensures that individual components can be updated or replaced without affecting the rest of the application.

We believe the system meets the needs of both administrators and retail users effectively, and we are confident that this platform can greatly enhance operational efficiency for wholesale footwear businesses.

5.2 FUTURE ENHANCEMENT

The **Step-up Wholesale** platform has been designed to support scalability and flexibility, making it easy to enhance or modify the system as requirements evolve. Potential future enhancements include:

- **Invoice PDF Generation:** Automatically generate and download invoices in PDF format for completed orders.
- **Email Notifications:** Notify users of order confirmations and status updates via email.
- **Analytics Dashboard:** Add visual insights for sales, stock levels, and order trends.
- **Multi-user Admin Support:** Allow multiple admins with varying levels of access control.
- **Mobile App Integration:** Create an Android/iOS version for easier access on mobile devices.
- **Online Payment Integration:** Enable secure payment gateways for pre-order payment collection.

6. BIBLIOGRAPHY

BOOK REFERENCES

1. **Kevin Yank**, *Build Your Own Database Driven Website Using PHP & MySQL*, SitePoint, 5th Edition, 2011.
2. **Luke Welling & Laura Thomson**, *PHP and MySQL Web Development*, Addison-Wesley, 5th Edition, 2016.
3. **Robin Nixon**, *Learning PHP, MySQL & JavaScript*, O'Reilly Media, 6th Edition, 2018.
4. **David Powers**, *PHP Solutions: Dynamic Web Design Made Easy*, Apress, 2nd Edition, 2010.
5. **Craig Grannell**, *The Essential Guide to CSS and HTML Web Design*, Apress, 2008.

WEB REFERENCES

1. **www.php.net**– Official PHP documentation and best practices.
2. **www.mysql.com**– MySQL database reference and documentation.
3. **www.w3schools.com** – PHP, MySQL, AJAX, and Bootstrap tutorials.
4. **www.getbootstrap.com** – Official Bootstrap documentation and components.
5. **www.tutorialspoint.com/php** – PHP & MySQL tutorials, examples, and explanations.

APPENDIX

SAMPLE SCREENS