**1**.**INTRODUCTION**

Welcome to **Computer Emporium**, an innovative online platform crafted to transform the shopping experience for computer hardware and accessories. In an era where technology evolves at an unprecedented rate, the demand for reliable and high-performance computing solutions is greater than ever. Whether you’re a tech enthusiast, a professional seeking top-tier equipment, or a casual user planning to upgrade, Computer Emporium is your one-stop destination for all computing needs.

This project highlights the development of a user-friendly website that operates as a comprehensive marketplace, offering an extensive selection of computer products. From state-of-the-art processors to the latest gaming peripherals, **Computer Emporium** is designed to cater to a wide range of customer needs, ensuring an effortless and engaging shopping experience. This documentation explores each stage of development, from initial planning to final implementation, showcasing the functionality and features tailored for an optimal user experience.

**1.1OBJECTIVE OF THE PROJECT**

The Computer Emporium project aims to create an intuitive, user-friendly platform that caters to both tech enthusiasts and casual users, focusing on delivering a seamless shopping experience. The website will allow users to easily browse, search, and purchase a wide range of computer hardware and accessories, ensuring accessibility and convenience in every step. By implementing a clean and organized interface, the platform will enable users to find products quickly, enhancing overall usability and customer satisfaction.

In addition to core shopping functionalities, the project will incorporate features designed to enrich the user experience, such as product comparison tools and accessible customer support. These features will help users make informed decisions and receive prompt assistance when needed. Built with scalability in mind, the platform will readily accommodate the addition of new products, categories, and advanced features, evolving to meet future demands. Through these objectives, Computer Emporium aims to establish itself as a reliable and efficient marketplace for computer hardware and accessories.

**2.SYSTEM ANALYSIS**

The system analysis for "Computer Emporium" involves a comprehensive evaluation to ensure a successful e-commerce platform. It begins with identifying user requirements, focusing on ease of navigation, effective product search, and a secure checkout process. Functional requirements include features such as product catalogue management, shopping cart integration, user authentication, and order processing. The analysis also covers non-functional requirements, emphasizing scalability, security, and performance to handle varying traffic loads. Technical aspects involve selecting appropriate technologies for both front-end and back-end development, as well as database management. Integration with payment gateways and other third-party services is also planned to provide a seamless and reliable online shopping experience.

**2.1 EXISTING SYSTEM**

In the current landscape, small and medium-sized businesses dealing with computer- related products face several challenges due to the lack of a centralized online platform. Customers who wish to purchase computer hardware or accessories often struggle with limited availability, as they are required to physically visit multiple stores to compare prices and check stock, making the process time-consuming and inefficient. Furthermore, many local computer shops have limited inventories, reducing the variety of products available to customers, especially those searching for specific components.

For shop owners, managing inventory and updating product listings is a difficult task, as they often rely on physical storefronts or fragmented online platforms. This limits their ability to reach a broader customer base. Additionally, customers may face trust issues, as the scattered nature of product listings makes it harder to ensure product quality or verify the seller's reputation. Transactions in the existing system are mostly manual, complicating the process of order tracking for customers and making payment processing and invoicing more cumbersome for sellers.

**DISADVANTAGES (DRAWBACKS)**

* Inconvenience for Customers: Customers must physically visit multiple stores to find and compare computer products, leading to a time-consuming and inefficient shopping experience.
* Limited Product Availability: Local computer shops often have restricted inventories, resulting in a lack of variety and forcing customers to compromise on their desired products.
* Inventory Management Challenges: Shop owners struggle to manage inventory effectively without a centralized platform, leading to discrepancies between listed products and actual stock availability.
* Low Visibility for Sellers: Individual shop faces reduced visibility in the market, limiting their ability to attract new customers and diminishing their competitive edge.
* Manual Transaction Processes: The reliance on manual transactions increases the likelihood of errors and delays in order processing, negatively impacting customer satisfaction and complicating payment management for sellers.

***2.2* PROPOSED SYSTEM**

The proposed system, Computer Emporium, is an online platform that centralizes the buying and selling of computer products. It connects users with various shops, offering a wide range of products while streamlining inventory management, transactions, and customer service for both sellers and buyers.

**Advantages of the Proposed System**

1. Convenient Shopping: Customers can easily browse and compare products from multiple sellers in one location, saving time and effort.
2. Diverse Product Range: A centralized platform allows access to a wider variety of computer products and accessories.
3. Efficient Inventory Management: Sellers can manage their inventories in real-time, reducing discrepancies and improving stock accuracy.
4. Increased Visibility: Shops benefit from enhanced online visibility, attracting more customers and increasing sales potential.
5. Streamlined Transactions: Automated order processing and payment systems minimize errors and improve the overall customer experience.
6. Improved Customer Trust: A unified platform enhances buyer confidence through user reviews, ratings, and verified seller profiles.

**2.3 SYSTEM REQUIREMENT SPECIFICATION**

A software requirement specification (SRS) is a comprehensive description of the intended purpose and environment under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-worked situation

**Customer Requirements**

▪ The system should be fast.

▪ User friendly

▪ Maintaining security of data

▪ Efficiency in data retrieval and management

**What the developer needs to know?**

▪ Must know the existing system and its drawbacks

▪ Must know what will be needed in the proposed system

**User Requirements**

The user requirement(s) specification is a document usually that specifies the requirements the user expects from software to be constructed in a software project.

Administrator has overall control in the system

Admin can view customer details

Faster processing

Functional Requirement

User Registration and Authentication:

* Users must be able to create accounts with a secure registration process, including email verification.
* The system should allow users to log in and log out securely.

Product Browsing and Search:

* Users should be able to browse products by categories, such as hardware, software, and accessories.
* A search functionality must be implemented to enable users to find products by name, category, or other criteria.

Product Listings and Details:

* Sellers must be able to add, edit, and delete their product listings, including descriptions, prices, and images.
* Users should view detailed product information, including specifications, availability, and seller information.

Shopping Cart and Checkout:

* Users must have the ability to add products to a shopping cart and view the cart contents.
* The system should support a secure checkout process, including payment options and order confirmation.

Order Management:

* Users should be able to track their orders, view order history, and receive updates on order status.
* Sellers must be able to manage their orders, update order statuses, and process returns.

User Reviews and Ratings:

* Customers should be able to leave reviews and ratings for products they have purchased, providing feedback to other users and sellers.

Admin Panel:

* Admin users must have access to a dashboard for managing users, products, and orders.
* The admin should be able to monitor and resolve disputes, manage reported reviews, and analyse sales data.

**2.3.1 Hardware Specification**

Processor: Intel i3 or higher

Processor Speed :1.5GHz or higher

System Bus :64bits

Memory:4GB RAM or Higher

Hard Disk :40GB or Higher

Monitor: 14’ LED Monitor

Keyboard: 104 Keys

Pointing Device: Two or Three Button Mouse

**2.3.2 Software Requirements**

Operating System: Windows 11

Front End: HTML, PHP, CSS Scripting

Language: JavaScript

Back End: MySQL Server Web Server Wamp 2.0

Browser: Chrome

**2.3.3 Front End**

**HTML**

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document**.**

**CSS**

Cascading Style Sheets (CSs) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML. The language can be applied to any XML document, including plain XML. SVG and XUL and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification or presentation characteristics, enable multiple HTML pages to share formatting by specifying on all platforms except Windows. MySQL ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many third party GUI tools are also available.

**JAVASCRIPT**

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser’s display accordingly. Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client- side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags,

<SCRIPTS>…</SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

**2.3.4 Back End**

**PHP**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994.the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page. but it now stands for the recursive backronym PHP: Hypertext Preprocessors.

PHP code may be embedded into HTML code. or it can be used in combination with various web template systems. Web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images. with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement. standalone graphical applications

**MySQL**

MySQL server is powerful database and it requires limited programs and used has back end. It supports GUI and more application is developed by help this server. Collection of tables which holds the data is called database. A beginner can create their own database by click home page ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or install MySQL Workbench via a separate download. Many thirdparty GUI tools are also available.

**2.4 FEASIBILITY ANALYSIS**

The main objective of the feasibility study is to treat the technical, operational, logical and economic feasibility of developing the computerized system. A feasibility study involves taking a judgment call on whether a project is doable. The two criteria to judge feasibility are cost required and value to be delivered. All systems are feasible, given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at System study phase itself. A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions*.*

The feasibility study to be conducted for this project involves.

• Technical feasibility

• Economic feasibility

• Operational feasibility

• Behavioural feasibility

Technical Feasibility

The technical feasibility centres on the existing system and what extend it can support the proposed addition. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. The minimum requirements of the system are met by average user. The developer system has a modest technical requirement as only minimal or null changes are required for implementing system.

Normally associated with the technical feasibility includes:

Development risk

Resource availability

Technology

The proposed system can work without any additional hardware or software support other than the computer system and networks. So, I analysed that the proposed system is much more technically feasible than other systems when comparing with the benefits of the new system.

Feasibility

In the economic feasibility the development cost of the system is evaluated weighing it against the ultimate benefit derived from the new system. It helps organizations assess the viability, cost, and benefits associated with projects before financial resources are allocated. It is found that the benefit, from the new system would be more than the cost and time involved in its development.

Operational Feasibility

In the existing system it is very difficult to maintain and update huge amount of information. The development of the system was started because of the requirement put forward by the management. This system, will handles the request in a better way and make the process easier thus, it is sure that the system developed is operationally feasible.

Behavioural feasibility

People are inherently resistant to changes and computer is known for facilitating the chances. An estimate should be made to how strongly the users react towards the e development of the system. The proposed system consumes less time. Thus, the people are made to engage in some other important work.

**2.5 DATA FLOW DIAGRAM (DFD)**

2.5.1 Introduction of Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design). Data Flow Diagrams were invented by Larry Constantine, the original developer of structured design based on Martin and Estrin’s “Data Flow Graph” model of computation. Data Flow Diagrams (DFD) are one of the three essential perspectives of Structured System Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system’s evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system’s data flow diagram can be drawn up and compared with the new system’s data flow diagram to implement a more efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately affects the structure of the whole system, from order to dispatch to report. How a system is developed can be determined through a data flow diagram. Developing a data flow diagram helps in identifying the transaction data in the data model. There are different notations to draw data flow diagrams, defining different visual representations for processes, data stores, data flow, and external entities. The first step is to draw a Data Flow Diagram (DFD), also known as a “bubble chart,” with the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design phase, functionally decomposing the requirements specification down to the lowest level of detail. DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation, and the lines represent data flow in the system

**DFD Symbols**

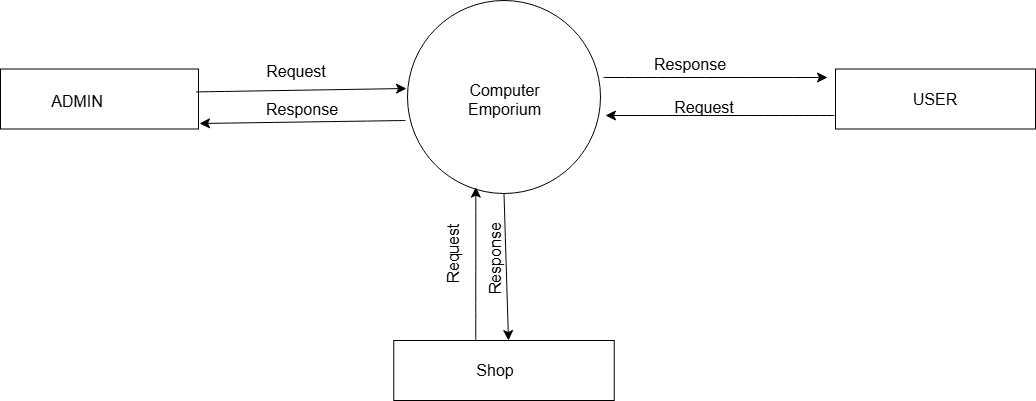
* Square – Define source or Destination of system
* Data Flow – Identifies data flow circle
* Open Rectangle – Data store



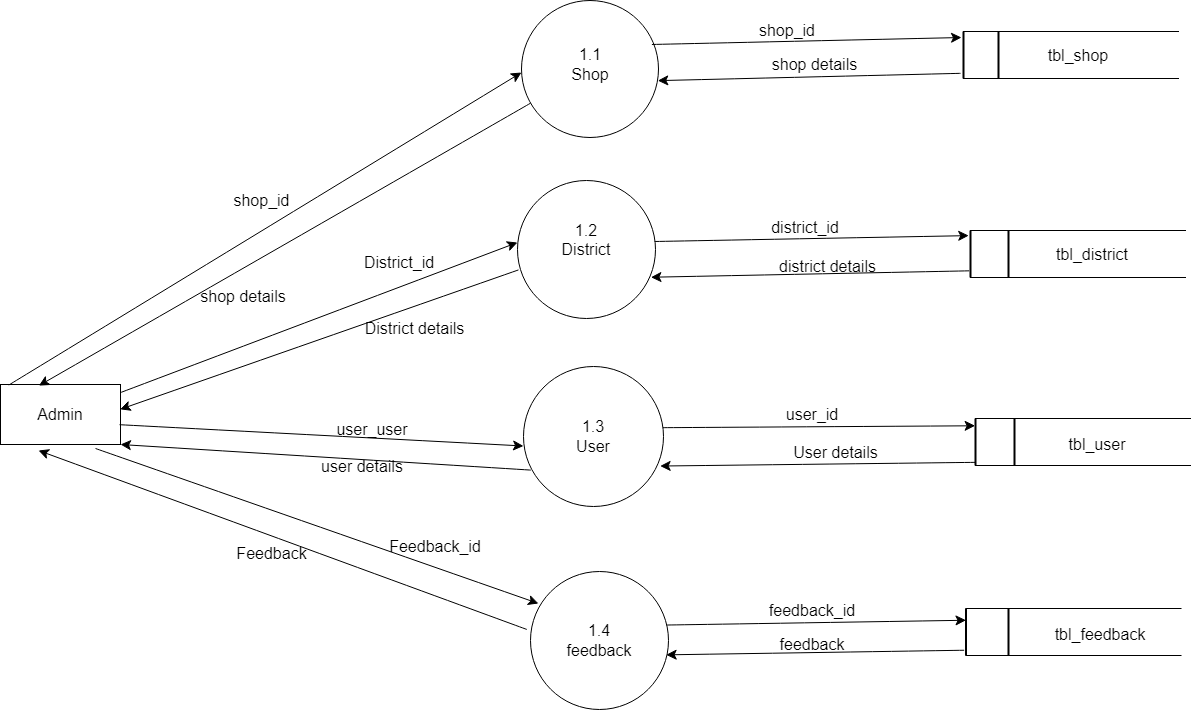
* Data flow - Identifies data flow Circle

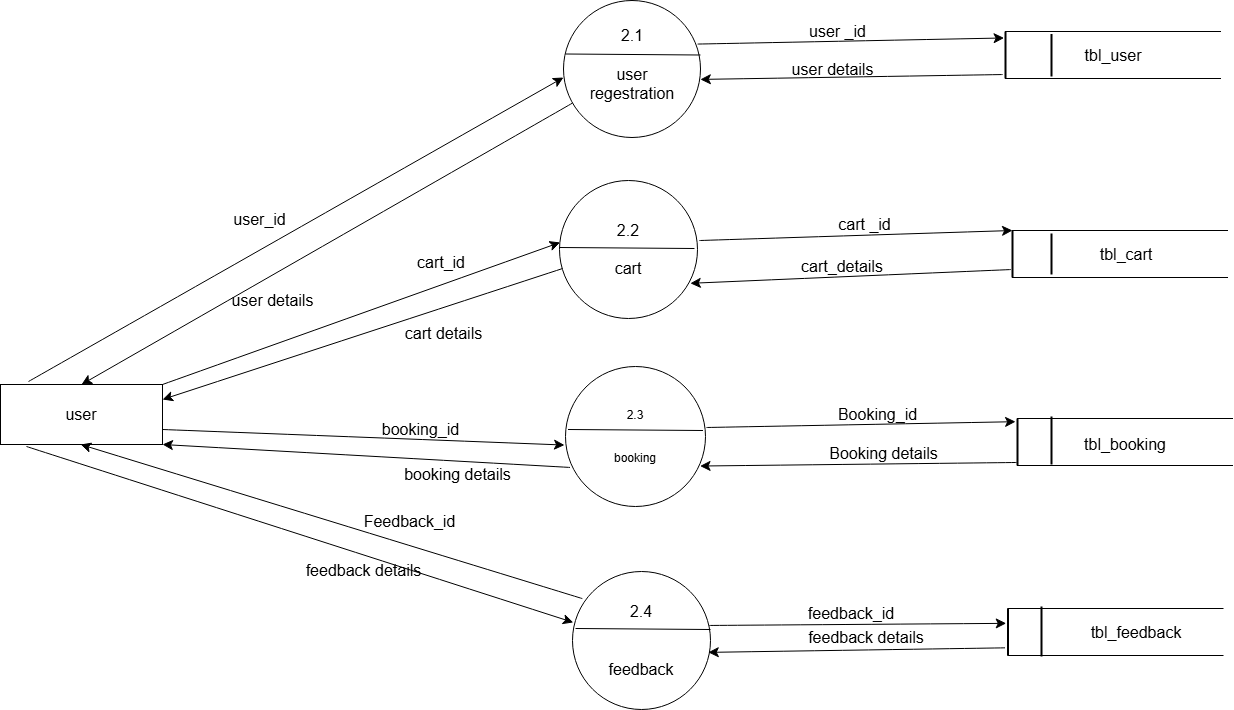
DFD

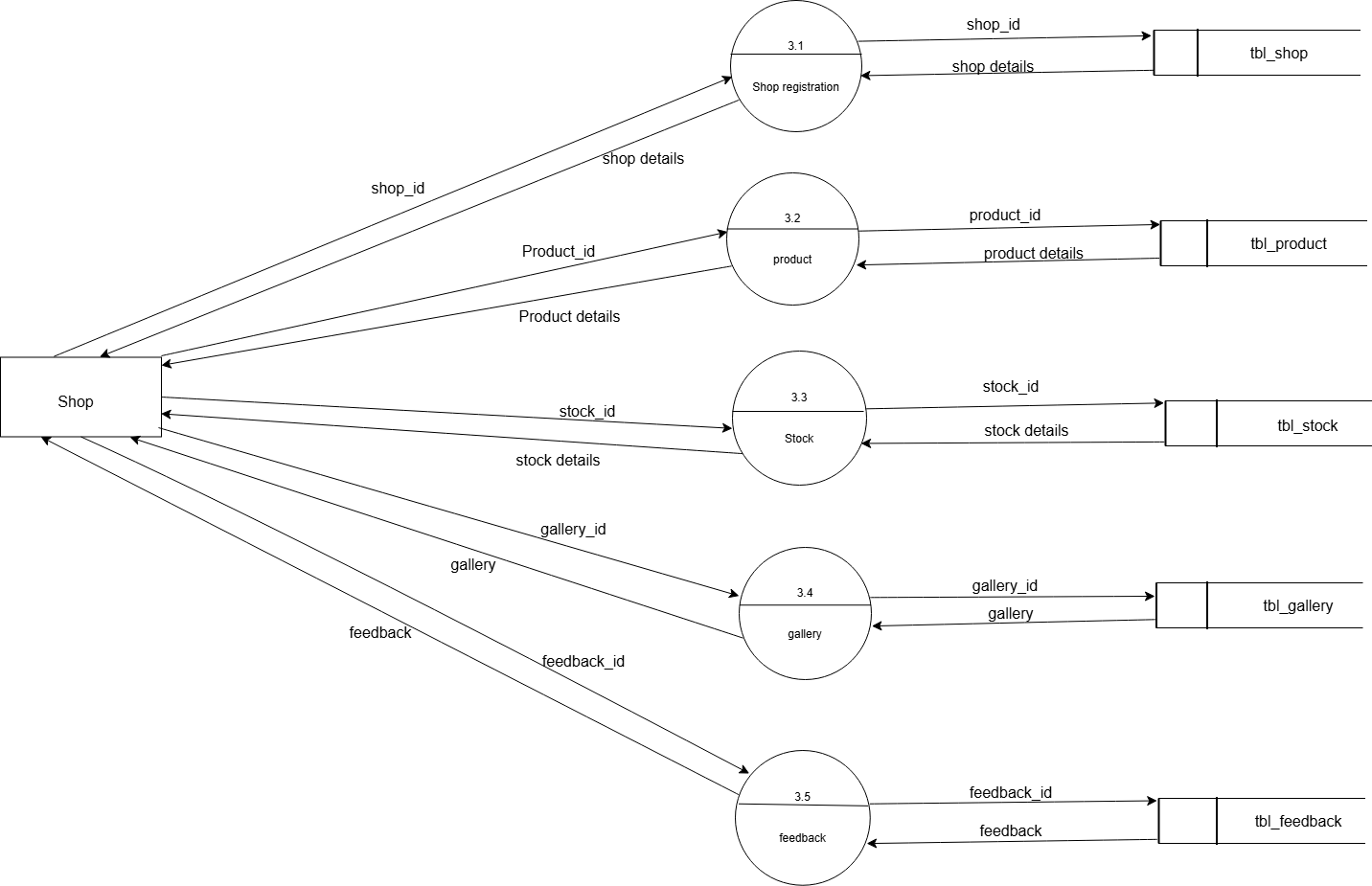
Level 0



Level 1







**3. *SYSTEM DESIGN***

**3.1 *INPUT DESIGN***

The quality of the system input determines the quality of the system output. Input specification describes the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and procedure result from accurate data, or they can result in the production or erroneous information. The input design also determines whether the user can interact efficiently with the system. In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

**Admin**

Admin is the one who controls the system. The admin can access the page using username and password. Admin can view and register branches, add districts and places, accept/reject user requests and give complaint reply to users.

**Shop**

The shop input design includes registration fields for shop name, owner details, contact information, and address. It features product management inputs for adding, editing, and deleting products, including name, description, price, quantity, and images. The design ensures user-friendly interfaces and validation for accurate data entry.

**User**

The user input design consists of registration fields for username, email, password, and personal details. It includes a login form for authentication and a product search bar for browsing. Additionally, users can submit reviews and ratings, ensuring a seamless experience while capturing essential data for interactions.

**3.2 OUTPUT DESIGN**

One of the important features of an information system for users is the output produces. Output is the information delivered to users through the information system. Without quality of the output, the entire system appears to be unnecessary that users will avoid using it. Users generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user though an interactive process. Until the result is considered to be satisfactory

* **User Outputs**: For regular users (customers), list product information, search results, purchase history, etc.
* **Shop Outputs**: For shop owners, describe outputs like inventory status, sales reports, and order lists.
* **Admin Outputs**: For admins, include control panels, user management screens, reports, and system logs.

**3.3 *TABLE DESIGN***

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data directory provide the basis for the data design activity*.*

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database Management System allows data to be protected and organized separately from other resources. Database is an integrated collection of data. This is the difference between logical and physical data.

The organization of data in the database aims to achieve three major objectives:

* Data integration
* Data integrity
* Data independence

The databases are implemented using a DBMS package. Each particular DBMS has unique characteristics and general techniques for database design. There are 6 major steps in design process. The first 5 steps are usually done on paper and finally the design is implemented.

* Identify the table and relationships
* Identify the data that is needed for each table and relationship
* Resolve the relationship
* Verify the design
* Implement the design

The database uses tables for storage. A table also contain records, which is a set of fields. All records in a table have the same set of fields with different information.

**TABLE DESIGN**

**1. tbl\_district**

Primary Key: district\_id

Foreign Key: NIL

Description: The table stores information regarding districts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| **1** | **district\_id** | **INT** |  | **Unique ID of District** |
| **2** | **district\_name** | **VARCHAR** | **25** | **Name of District** |

**2. tbl\_place**

Primary Key: place\_id

Foreign Key: district\_id

Description: The table stores information regarding places within the district.

| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| --- | --- | --- | --- | --- |
| 1 | place\_id | INT |  | Unique ID of Place |
| 2 | place\_name | VARCHAR | 50 | Name of Place |
| 3 | place\_pincode | INT |  | Pin code of Place |

**3. tbl\_category**

Primary Key: cat\_id

Foreign Key: NIL

Description: The table contains details about the categories of products to be sold in the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | category\_id | INT |  | Unique ID of the Category |
| 2 | category\_name | VARCHAR | 50 | Name of the Category |

**4. tbl\_subCategory**

Primary Key: sub\_id

Foreign Key: category\_id

Description: The table contains details about product subcategories of the products to be sold in the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | category\_id | INT |  | Unique ID of Category |
| 2 | subCategory\_id | INT |  | Unique ID of Sub-Category |
| 3 | subCategory\_name | VARCHAR | 50 | Name of Sub-Category |

**5. tbl\_shop**

Primary Key: shop\_id  
Foreign Key: place\_id, admin\_id

Description: The table contains information regarding the shops integrated into the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | shop\_id | INT |  | Unique ID of the Shop |
| 2 | shop\_name | VARCHAR | 50 | Name of the Shop |
| 3 | shop\_email | VARCHAR | 50 | Email of the Shop |
| 4 | shop\_password | VARCHAR | 50 | Password of the Shop |
| 5 | shop\_address | VARCHAR | 50 | Address of the Shop |
| 6 | place \_id | INT |  | Unique ID of the Place |
| 7 | shop\_status | VARCHAR | 50 | Status of the Shop |
| 8 | shop\_logo | VARCHAR | 50 | Logo of the Shop |
| 9 | shop\_proof | VARCHAR | 50 | Proof of the Shop |
| 10 | shop\_contact | VARCHAR | 50 | Contact of the Shop |

**6. tbl\_user**

Primary Key: user\_id

Foreign Key: place\_id, admin\_id

Description: The table contains information about the Users of the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | user\_id | INT |  | Unique ID of the User |
| 2 | user\_name | VARCHAR | 50 | Name of the User |
| 3 | user\_email | VARCHAR | 50 | Email of the User |
| 4 | user\_password | VARCHAR | 50 | Password of the User |
| 5 | user\_address | VARCHAR | 50 | Address of the User |
| 6 | place\_id | INT |  | Unique ID of the Place |
| 7 | user\_number | INT |  | Number of the User |

**7. tbl\_product**

Primary Key: product\_id

Foreign Key: sub\_id, shop\_id

Description: The table contains information regarding the products to be sold in the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | product\_id | INT |  | Unique ID of the Product |
| 2 | product\_name | VARCHAR | 50 | Name of the Product |
| 3 | product\_desc | VARCHAR | 50 | Description of the Product |
| 4 | product\_photo | VARCHAR | 50 | Photo of the Product |
| 5 | product\_price | INT |  | Price of the Product |
| 6 | sub\_id | INT |  | Unique ID of the Sub\_Cat |
| 7 | shop\_id | INT |  | Unique ID of the Shop |

**8. tbl\_stock**

Primary Key: stock\_id

Foreign Key: product\_id, shop\_id

Description: The table contains information regarding the stock details of the products to be sold in the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | stock\_id | INT |  | Unique ID of the Stock |
| 2 | stock\_qty | INT |  | Quantity of the Stock |
| 3 | stock\_date | DATE |  | Date of stock updation |
| 4 | product\_id | INT |  | Unique ID of the product |
| 5 | shop\_id | INT |  | Unique ID of the Shop |

**9. tbl\_gallery**

Primary Key: gallery\_id

Foreign Key: product\_id

Description: The table contains the photos to be uploaded to the products page of the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | gallery\_id | INT |  | Unique ID of the Gallery |
| 2 | gallery\_file | VARCHAR | 50 | File name of the Gallery |
| 3 | product\_id | INT |  | Unique ID of the product |

**10. tbl\_booking**

Primary Key: booking\_id

Foreign Key: user\_id

Description: The table contains the booking details of the orders placed by the users of the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | booking\_id | INT |  | Unique ID of the Booking |
| 2 | booking\_date | DATE |  | Date of booking |
| 3 | booking\_amount | INT |  | Amount of booking |
| 4 | booking\_status | VARCHAR | 50 | Booking status |
| 5 | user\_id | INT |  | Unique ID of the User |

**11. tbl\_cart**

Primary Key: cart\_id

Foreign Key: product\_id

Description: The table contains information regarding the cart details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | cart\_id | INT |  | Unique ID of Cart |
| 2 | cart\_qty | INT |  | Quantity of the Cart |
| 3 | cart\_status | VARCHAR | 50 | Status of the Cart |
| 4 | product\_id | INT |  | Unique ID of the Product |
| 5 | tracking\_id | INT |  | Unique ID for Tracking |
| 6 | booking\_id | INT |  | Booking ID |

**12. tbl\_review**

Primary Key: review\_id

Foreign Key: user\_id, product\_id, shop\_id, admin\_id

Description: The table contains information regarding customers sending reviews of the products sold in the project system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **NAME** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| 1 | complaint\_id | INT |  | Unique ID of the Complaint |
| 2 | complaint\_title | VARCHAR | 50 | Title of Complaint |
| 3 | complaint\_description | VARCHAR | 50 | Complaint Description |
| 4 | complaint\_file | VARCHAR | 50 | Complaint File |
| 5 | complaint\_date | DATE |  | Date of Complaint |
| 6 | user\_id | INT |  | Unique ID of the User |
| 7 | product\_id | INT |  | Unique ID of the Product |
| 8 | shop\_id | INT |  | Unique ID of the Shop |

**4. SYSTEM TESTING AND IMPLEMENTATION**

**4.1 SYSTEM TESTING**

Testing is the process of examining the software to compare the actual behaviour with that of the expected behaviour. The major goal of software testing is to demonstrate that fault are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is defined as the process by which one detects the defects in the software. Any software development organization or team has to performs several processes. Software testing is one among them. It is the final opportunity of any programmer to detect and rectify any defects that may have appeared during the software development stage. Testing is a process of testing a program with the explicit intention of finding errors that makes the program fail to short system testing and quality assurance is a review in software products and related documentation for completion, correctness, reliability and maintainability.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

The testing steps are:

* Unit Testing
* Integration Testing
* Validation
* Output Testing
* Acceptance Testing

System Testing provides the file assurance that software once validated must combined with all other system elements. System testing verifies whether all elements have been combined properly and that overall system function and performance is achieved. FA the integration of modules, the validation test was carried out over the system. It was that all the modules work well together and meet the overall system function and performance

1. **Unit Testing**

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Various test cases are prepared. For each module these test cases are implemented and it is checked whether the module is executed as per the requirements and outputs the desired result. In this test each service input and output parameters are checked. In unit testing

* Module interface was tested to ensure that information properly flows into and out of the program under test.
* Boundary condition was tested to ensure that module operates properly at boundaries established to limit or restrict processing.
* All independent paths through the control structures were executed to ensure that all statements in the modules have been executed at least once.
* Error handling paths were also tested

B**. Integration Testing**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

Unit tested module were taken and a single program structure was built that has been dictated by the design. Incremental integration has been adopted here.

The modules are tested separately for accuracy and modules are integrated tooth in using bottom-up integration i.e., by integrating from moving from bottom to the top the system is checked and errors found during integration are rectified. In this testing individual modules were combined and he module wise Shifting was verified to be alright.

The entire software was developed and tested in small segments, where errors were easy to locate and rectify. Program builds (group of modules) were constructed corresponding to the successful testing of user interaction, data manipulation analysis, and display processing and database management.

C. **Validation Testing**

Validation testing is done to ensure complete assembly of the error-free software. Validation can be termed successful only if it functions in manner. Reasonably expected by the student under validation is alpha and beta testing. The student-side validation is done in this testing phase. It is checked whether the data passed to each student is valid or not. Entering incorrect values does the validation testing and it is checked whether the errors are being considered.

Incorrect values are to be discarded. The errors are rectified.

In COMPUTER EMPORIUM verification are done correctly. So, there is no chance for users to enter incorrect values. It will give error messages by using different validations. The validation testing is done very clearly and found it is error free.

*D.* **Output Testing**

After performing the validation testing the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format

The output format on the screen was found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as specified requirement by the user. Hence output testing does not result in any Correction in the system output. This project is developed based on the user choice. It is user friendly. The output format is very clear to user. Output testing is done on Smart builders correctly.

*E.***Acceptance Testing**

Acceptance involves running a suite of tests on the completed system. Each individual test, known as a Case, exercise particular operating condition of the operating condition of the user's environment or feature of the system, and will result in a pass fail, or Boolean outcome.

***4.2* SYSTEM IMPLEMENTATION**

The implementation is the final state and it is an important phase. It involves the invalid programming system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. A major task of preparing for implementation is education of users, which should really have been taken place much carrier in the project when they were belong involved in the investigation and design work. During the implementation phase system actually take physical shape. In order to develop a system implemented planning is very essential

The implementation phase of the software development is concerned with translating design specification into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented.

Before implementation several tests have been conducted to ensure that no errors are encountered during the operation. The implementation phase ends with an evaluation of the system after placing into the operation for a period of time. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from old system to new system. The system can be implemented only after testing is done and is found to be working to specifications. The implementation stage is a systems project in its own right

The implementation stage involves following tasks:

* Careful planning
* Investigation of system and constraints.
* Design of method to achieve change over
* Evaluation of the changeover method

In the case of this project all the screens are designed first. For making it to be executable, codes are written on each screen and performs the implementation by creating the database and connecting to the server. After that the system, is Checked, whether it performs all the transactions Correctly. Then databases are cleared and made it to be usable to the technicians.

***Implementation Plans***

The following are the step involved in the implementation plan of "Computer emporium":

❖ Test system with sample data

❖ Detection and correction of errors

❖ Make the necessary changes in the system

❖ Check the existing system

❖ Installation of hardware and software utilities

❖ Training and involvement of user personals

***5.* SECURITY TECHNOLOGIES & POLICIES**

The protection of computer-based resources that includes hardware, software, data procedures and people against unauthorized use or natural.

Disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality
* SYSTEM SECURITY*:* refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.
* DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.
* SYSTEM INTEGRITY refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as caves dropping and wiretapping
* PRIVACY*:* defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.
* CONFIDENTIALITY: is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.
* SECURITY IN SOFTWARE System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system.
* CLIENT-SIDE VALIDATIONVarious client-side validations are used to ensure on the client side that only valid data is entered. Client-side validation saves server time and load to handle invalid data. Some checks imposed are:
* Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
* Tab-indexes are set according to the need and taking into account the ease of user while working with the system
* *SERVER-SIDE VALIDATION* Some checks cannot be applied at client side. Server-side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server-side checks imposed is:
* Server-side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
* User is intimating through appropriate messages about the successful operations or exceptions occurring at server side
* Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category
* User name, passwords and permissions are controlled over the server side. Using server-side validation, constraints on several restricted operations are imposed.

6.**MAINTENANCE**

Software maintenance is the modification of a software product delivery to correct faults, to improve performance or other attributes. Maintenance is the case with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer desires a change in requirement. Maintenance follows conversation to extend that changes are necessary to maintain satisfactory operations relative to changes in the user's environment

Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

**CATEGORIES OF MAINTENANCE**

***A.* Corrective Maintenance**

Corrective maintenance is the most commonly used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometimes damages other parts. In most cases, this is expensive. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is unknown and cannot be measured. Corrective maintenance is possible since the consequences of failure or wearing out are not significant and the cost of this maintenance is not great.

***B.* Perfective Maintenance**

Modification of a software product alter delivery to improve performance or maintainability. This term is used to describe changes undertaken to expand the existing requirements of the system. A successful piece or software lends to be subjected to the Succession of changes resulting in an increase in user requirements. This is based the premise that as the software becomes useful, the user experiment with new cases beyond the of Scope for which it was initially developed. Expansion requirements can take the form enhancement of existing system functionality and improvement in computational efficiency.

***C.* Adaptive Maintenance**

Modification of a software product performed after delivery to keep a are product usable a changed or changing environment. Adaptive maintenance includes any work initiated as a consequence of moving the software to a different hardware or software platform. It is a change driven by the need to accommodate modifications in the environment of software system. The environment in this context refers to the totality of all conditions and influences which act from outside upon the system. A change to the whole or part of this environment will Warrant a corresponding modification of the software

***D.* Preventive Maintenance**

Preventive maintenance is a schedule of planned maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it actually occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they actually fail. Preventive maintenance activities include equipment checks, partial or complete overhauls at specified periods*.*

Long-term benefits of preventive maintenance include:

* Improved system reliability.
* Decreased cost of replacement.
* Decreased system downtime.

**7.SCOPE FOR FUTURE ENHANCEMENT**

The Computer Emporium project offers ample opportunities for future development and improvements. One of the potential enhancements is the integration of AI-based productrecommendations to provide personalized suggestions to users based on their browsing and purchasing behavior. Expanding the platform with advanced filtering options and search functionality would help users find products more efficiently.

To improve user engagement, live chat support could be added, enabling real-time assistance for users and shops. Expanding payment options, including support for digital wallets and cryptocurrencies, would cater to a broader audience. Additionally, developing a mobile app version would increase accessibility, allowing users to shop on the go, while multi-languagesupport would broaden the platform's global appeal.

On the seller side, introducing analytics tools to track sales trends and customer behaviour would offer valuable insights, helping businesses optimize their product listings and marketing strategies. These enhancements will make Computer Emporium more competitive and user-friendly in the long term.

**8*.*CONCLUSION**

The "Computer Emporium" project showcases the development of a comprehensive e-commerce platform designed to cater to users, shop owners, and administrators. It effectively streamlines the process of browsing and purchasing computer-related products, with shop owners given the ability to manage their inventories through a simple yet powerful interface. Administrators can oversee the platform, manage user roles, and ensure smooth operations. The project not only provides a real-world solution for online retail but also emphasizes secure data handling, user-friendly interactions, and efficient back-end management.

Through this project, I have applied my knowledge of web development technologies, including front-end and back-end integration. The use of structured databases and proper validation methods ensures data consistency and security. Furthermore, the documentation process has highlighted the importance of clear communication of the system's functionality, making it easier for future developers to expand or maintain the platform. In conclusion, "Computer Emporium" is a scalable and efficient solution, with potential for further enhancements such as integrating payment gateways, user feedback systems for personalized shopping experience.

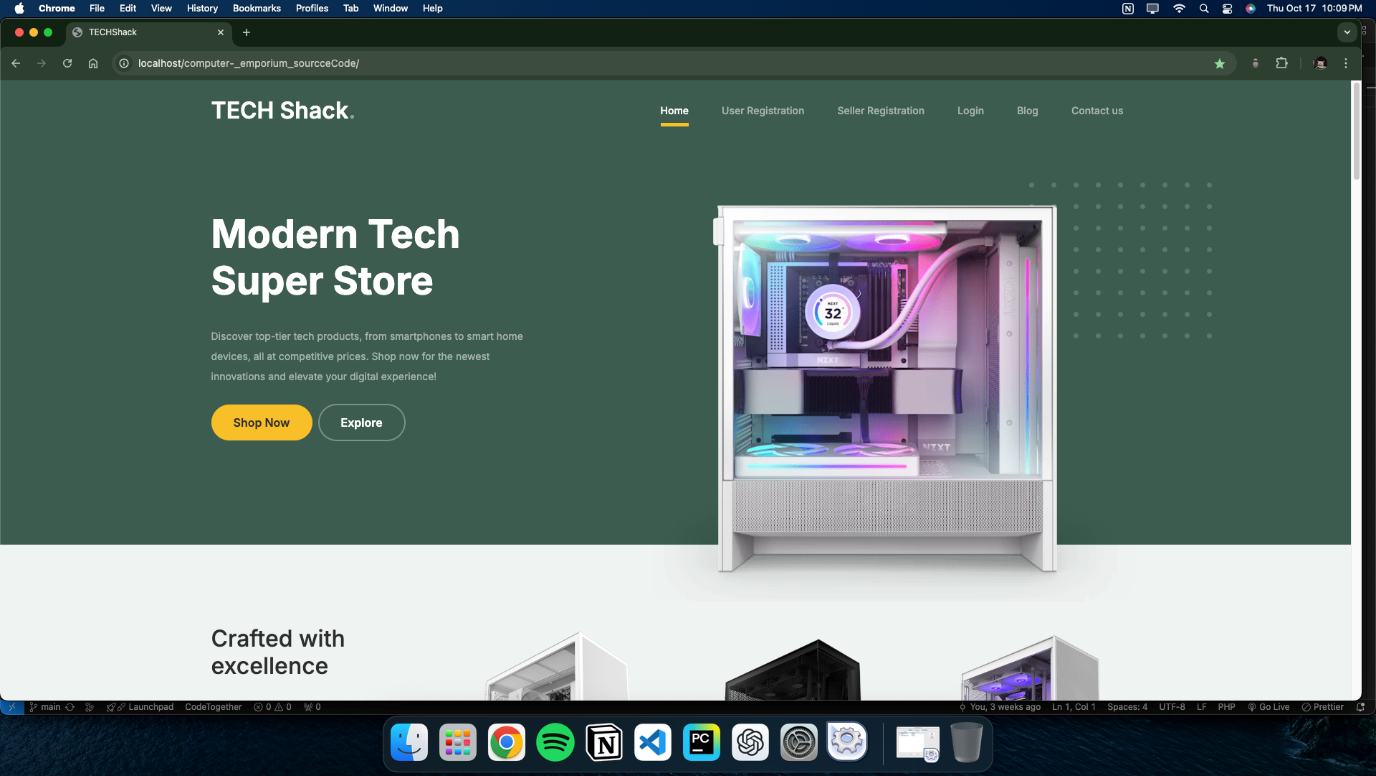
**9*.*BIBLIOGRAPHY**

* 1. Jalotte, Pankaj. " SOFTWARE ENGINEERING", Third Edition, Spring Publishers, Narosa Publishing House.
  2. Ramakrishnan, Raghu. and\_ Gehrke, Johannes. DATABASE MANAGEMENT SYSTEM", Third Edition, McGraw-Hill Professional Publication.
  3. Awad, Elias M. "SYSTEM ANALYSIS AND DESIGN", Second Edition, McGraw-Hill Professional publication.
  4. Rob, Peter. and Coronel, Carlos. "DATABASE SYSTEM DESIGN AND MANAGEMENT", Fifth Edition, Thomson Learning Publication.
  5. Pressman, Roger S. "SOFTWARE ENGINEERING", Fifth Edition, McGraw-Hill Publication.
  6. Bayross, Ivan. "SQL/MySQL" Second Revised Edition, BPB Publications
  7. Ramus Leodorf and Kevin Tatore. “PROGRAMMING PHP”
* W3Schools Online Web Tutorials. [www.w3schools.com](http://www.w3schools.com)
* WebPage Theme Selection. <https://themeselection.com>
* Template. https://themes.getbootstrap.com

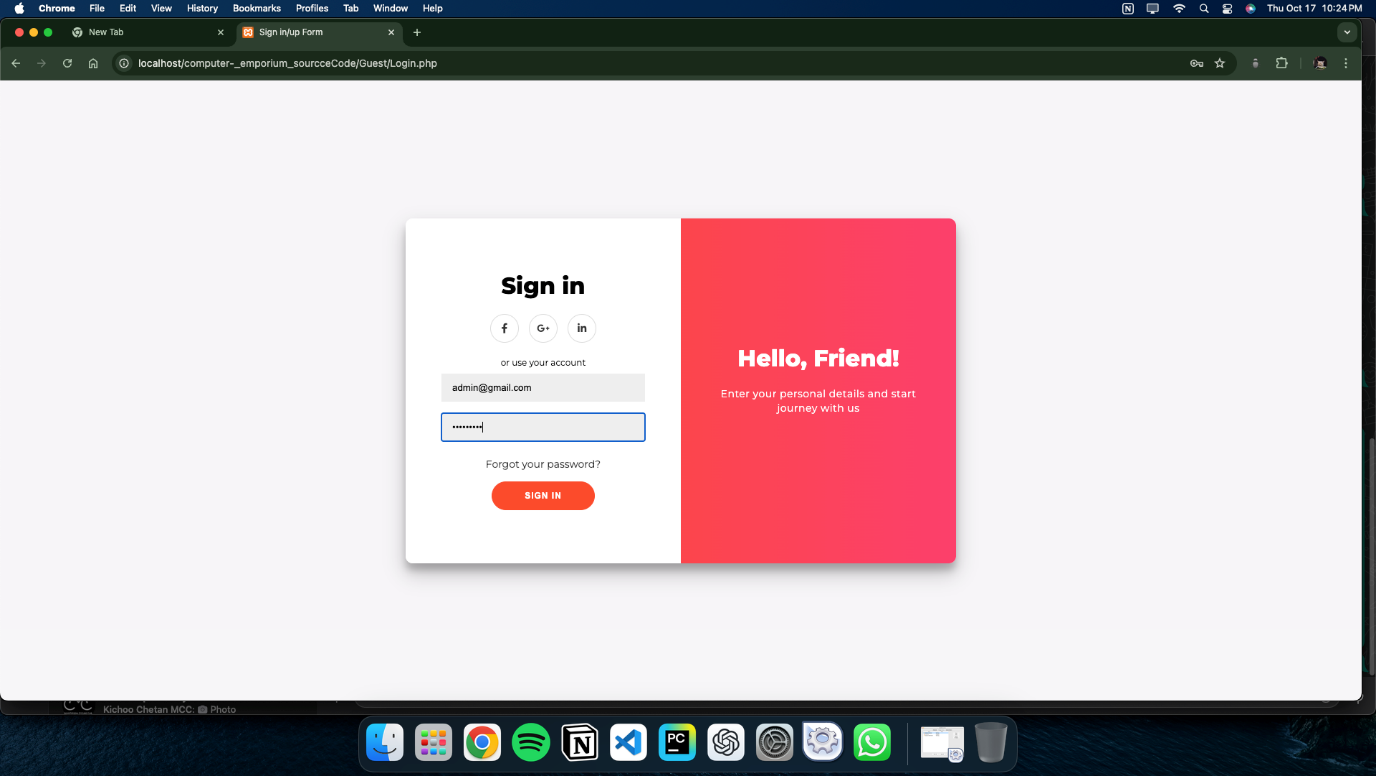
***10****.***APPENDIX**

**10.1 SCREENSHORT**

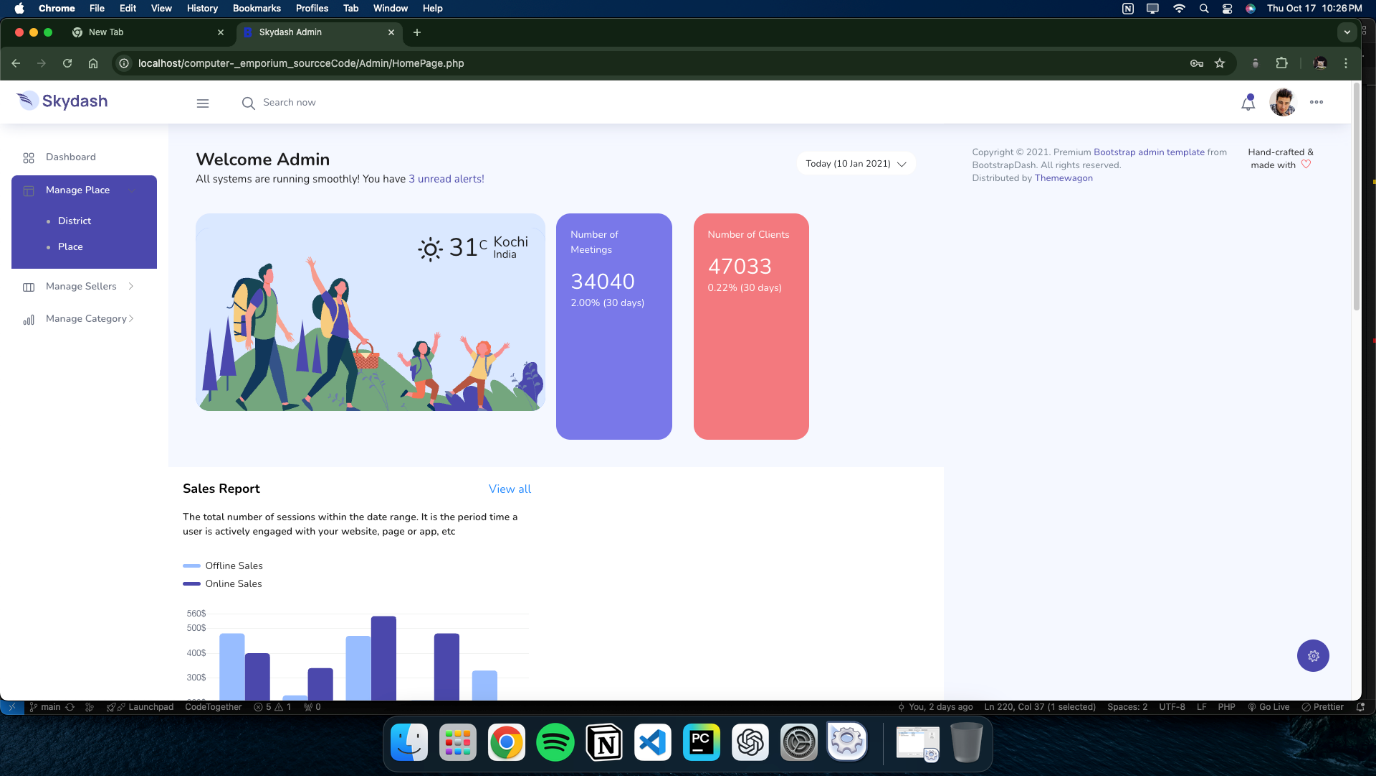
Home page:



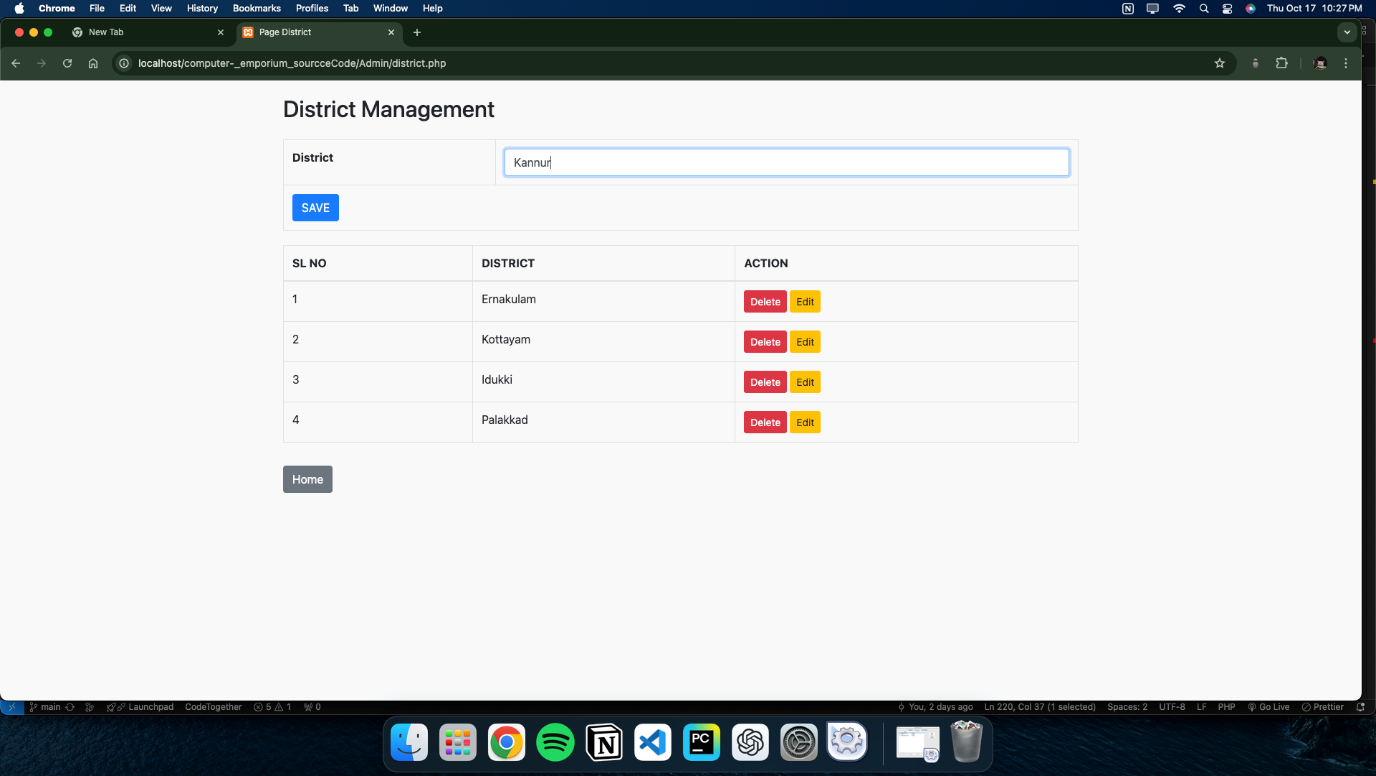
Sign in:

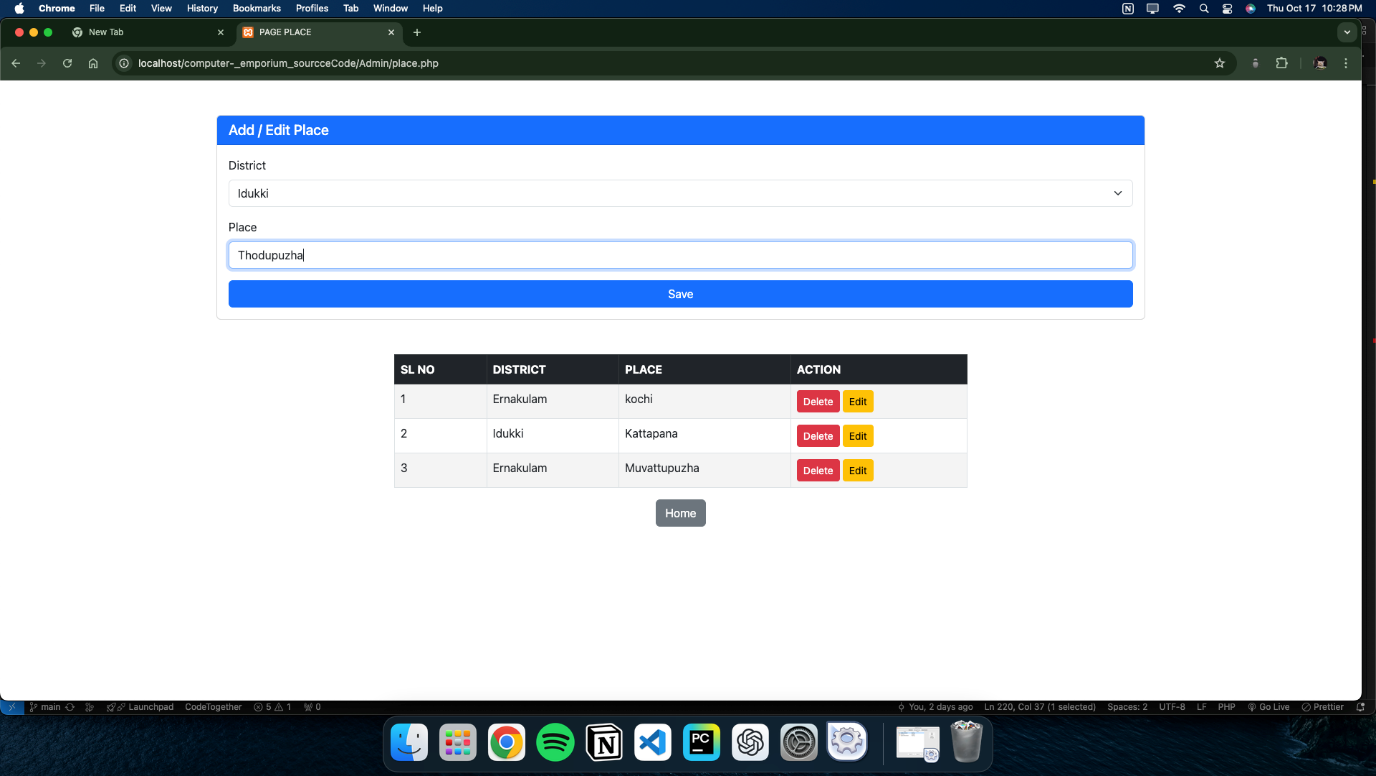


Admin dashboard:

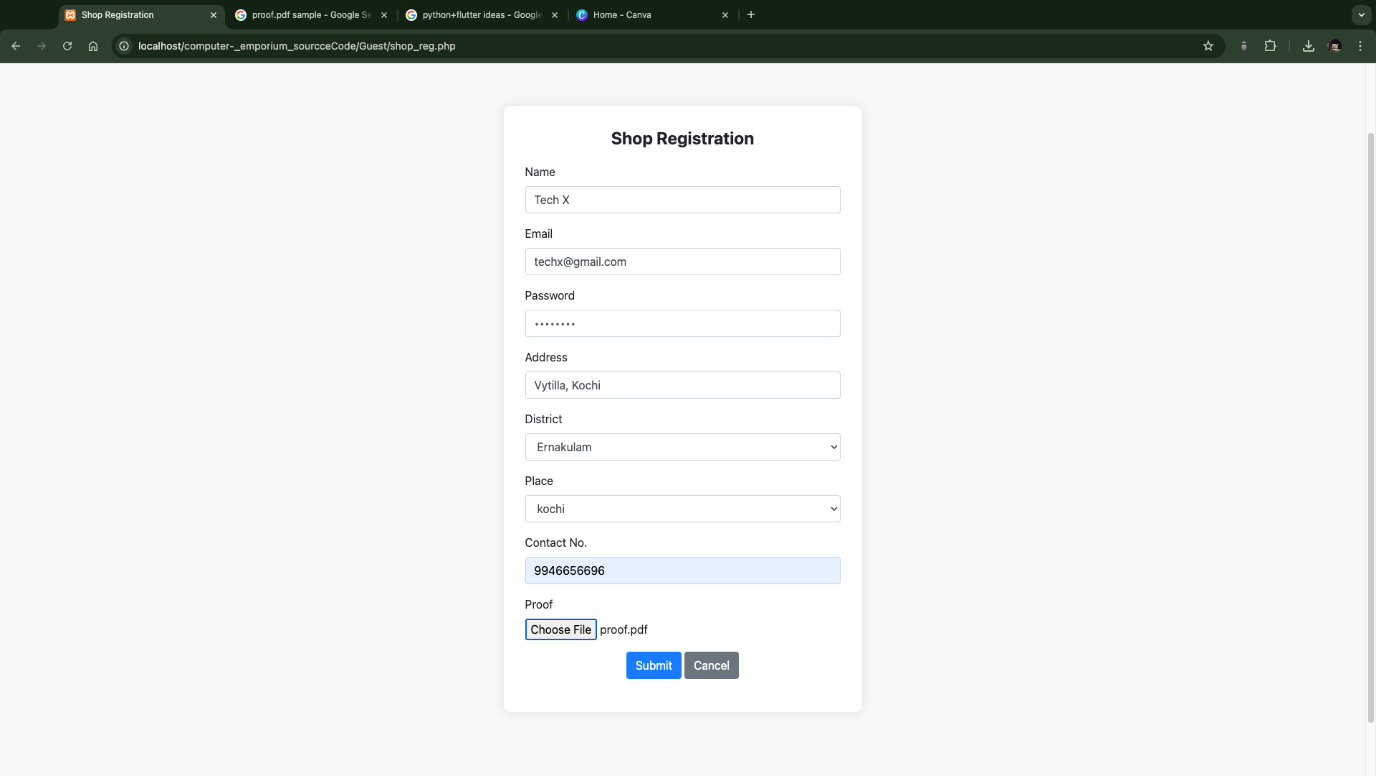


District:



Place

Shop Registration:



User Registration:

