**e-Energy**

**INTRODUCTION**

**2.INTRODUCTION**

The e- Energy manages a network of related companies that are active in the offering the product and service packages that an e- Energy's end user demand. e-Energy includes all raw material transportation and storage, as well as work-in-progress inventory and finished goods. A simple E-Energy is composed of numerous components that are connected by the flow of goods along it, from point of origin to point of consumption. The customer is where the e-Energy begins and ends. e-Energy is a management of a network of interconnected businesses involved in the provision of product and service packages required by the end customers in a e-energy. E-Energy spans all movement and storage of raw materials work-in-process inventory, and finished goods from point of origin to point of consumption. An integral part of the global energy sector is the buying and selling of energy equipment. The demand for energy equipment has significantly increased as the globe focuses more on switching to cleaner and more sustainable energy sources. This introduction gives a general overview of the purchasing and selling of energy equipment while highlighting the process' significance, significant players, and important factors.Themarket for energy equipment includes a wide variety of goods, including renewable energy technology like solar panels, wind turbines, energy storage systems, and infrastructure for electric vehicle charging. It also comprises standard energy machinery like transformers, generators, and transmission lines.

**Customer:** When a customer decides to buy a product that a business has made available for purchase, the sequence of actions is initiated by the customer. A sales order for a certain amount to be delivered on a specific date is entered once the customer contacts the company's sales department. If the product needs to be made, the sales order will have a condition that the manufacturing facility must meet.

**Planning:** The need brought on by the client's sales order will be merged with other orders. A production schedule will be developed by the planning division in order to produce the goods and fulfil client requests.

**Purchasing:** The manufacturing department sends a list of the raw materials and services needed to complete the customers' orders to the purchasing department. To ensure that the required raw materials are delivered to the manufacturing site on time, the buying department issues purchase orders to a list of approved suppliers.

**Inventory:** After receiving the raw materials from the suppliers, they are inspected for accuracy and quality before being placed in the warehouse. After then, the supplier will invoice the business for the goods they delivered. The raw materials are kept in storage until the production department needs them.

**Production:** Using the raw materials obtained from suppliers, the final goods ordered by the client are produced in accordance with a production plan. Prior to delivery to the customer, the completed and tested items are put back in the warehouse.

**Shipping:** After the finished product is received in the warehouse, the shipping department chooses the best shipping strategy to ensure that the products are delivered on time or earlier than the date the client has selected. The business will send an invoice for the delivered goods once the customer has received the goods.

**Area of Work:**

* Information System

**Type of Approach:**

* Software Based

**Existing System:** There are numerous online marketplaces that make it easy to acquire and sell energy equipment. These platforms offer a digital market place where equipment vendors may post their goods and customers can browse and buy them. Industry-specific websites, e-commerce platforms, and online classified ads are a few examples of such platforms.

**Proposed System:** Create a separate online marketplace that is intended just for the purchase and sale of e-energy equipment. To enable simple transactions, this platform should have a user-friendly interface, sophisticated search and filtering capabilities, and thorough equipment listings.

**AIMS AND OBJECTIVE**

**3.AIMS AND OBJECTIVE**

Depending on the viewpoints of the buyers and sellers involved, the goals of buying and selling e-energy equipment can change. The following objectives are shared by both parties:

**Buyer Objectives:**

* **Purchasing Reliable Equipment:** The main goal of purchasers is to purchase dependable, high-quality, and customized e-energy equipment. They want to invest in machinery that can produce, store, or distribute energy in an efficient and sustainable way.
* **Cost Effectiveness:** To address their energy needs, buyers frequently look for affordable alternatives. They search for energy equipment that offers value for money, taking into account things like up-front costs, ongoing expenditures, maintenance costs, and the possibility of long-term savings due to increased energy efficiency.
* **Technical Innovation:** Purchasers may seek out cutting-edge energy equipment that incorporates the newest technological innovations. This goal could be motivated by a desire to raise energy production, boost operational effectiveness, or implement cutting-edge strategies that support sustainability objectives.

**Seller Objectives:**

* **Revenue Generation:** By successfully marketing their products, manufacturers of e-energy equipment hope to make money. By contacting potential customers and concluding deals at advantageous prices, their goal is to increase sales and profits.
* **Market Expansion:** The goal of certain vendors may be to broaden their customer base and market penetration. They want to take on new markets or industries and establish themselves as trustworthy suppliers of energy machinery.
* **Collaborations and Partnerships:** Sellers may look to collaborate or form strategic alliances with other companies in the energy sector. By collaborating with like-minded companies or organizations, they want to take advantage of synergies, increase the range of their product offers, and strengthen their market position.

**ROJECT CATEGORY**

**4.PROJECT CATEGORY**

**Front End Technology:**

**HTML:** Hypertext Markup Language (HTML) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) an [web applications.](https://en.wikipedia.org/wiki/Web_application) With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript,](https://en.wikipedia.org/wiki/JavaScript) it forms a triad of cornerstone technologies for the [World Wide Web.](https://en.wikipedia.org/wiki/World_Wide_Web)

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links,](https://en.wikipedia.org/wiki/Hyperlink) quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img/> and <input /> directly introduce content into the page.

Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**HTML Versions**

**HTML 2.0**

HTML 2.0 is released in 1995 and it was the standard for website design until January 1997 and defined many core HTML features for the first time. HTML 2.0 starts supports core HTML elements and features such as tables and forms with limited set of form elements such as text boxes, and option buttons.

###### HTML 3.2

HTML 3.2 finalized by the W3C in early 1997. This version included support for creating tables and expanded options for form elements. It also allowed web pages to include complex mathematical equations. This version of HTML supports many presentation-focused elements such as font, as well as early support for some scripting features.

###### HTML 4.01

HTML 4.01 is the current official standard. This version is very stable, having been released in December 1999. This version added support for style sheets and scripting ability for multimedia elements.

###### HTML 5

The HTML5 specification that we see today has been published as a working draft and it is not yet final. HTML5 is still a work in progress, and all major browsers support many of the new HTML5 elements and APIs. Its core aim have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.

#### CSS (Cascading Style Sheets)

**CSS-** Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

###### Advantages of CSS

* + **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
  + **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
  + **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
  + **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
  + **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
  + **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

###### Who Creates and Maintains CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

**NOTE**− The World Wide Web Consortium, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

###### CSS Versions

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

CSS3 is the latest standard of CSS earlier versions (CSS2). Cascading Style Sheets, level 1 (CSS1) was came out of W3C as a recommendation in December 1996.

CSS3 became a W3C recommendation in June 1999 and builds on older versions CSS. It has divided into documentation is called as Modules and here each module having new extension features defined in CSS2.

###### TYPE OF CSS-

There are three types of CSS which are given below:

1. Inline CSS
2. Internal or Embedded CSS
3. External CSS

**Inline CSS:** Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

**Internal or Embedded CSS:** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file

**External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

Below is the HTML file that is making use of the created external style sheet

* link tag is used to link the external style sheet with the html webpage.
* href attribute is used to specify the location of the external style sheet file.

###### Properties of CSS

Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. As

Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.

###### SYNTAX OF CSS-

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts

−

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table>etc.
* **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
* **Value** − Values are assigned to properties. For example, color property can have value either red or #F1F1F1etc.

You can put CSS Style Rule Syntax as follows − selector {property: value} **Example** − You can define a table border as follows − table {border: 1px solid #C00; }

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these Selectors one by one.

###### 

###### The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings −

h1 {

color: #36CFFF;

}

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type −

\* {

color: #000000;

}

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

ul em {

color: #000000;

}

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

.black {

color: #000000;

}

This rule renders the content in black for every element with class attribute set to black in our document. You can make it a bit more particular. For example −

h1.black {

color: #000000; }

This rule renders the content in black for only <h1> elements with class attribute set to black.

You can apply more than one class selectors to given element. Consider the following example −

<p class = "center bold">

This para will be styled by the classes center and bold.

</p>

The ID Selectors

You can define style rules based on the id attribute of the elements. All the elements having that id will be formatted according to the defined rule.

#black {

Color: #000000;

}

This rule renders the content in black for every element with id attribute set to black in our document. You can make it a bit more particular. For example −

h1#black { color: #000000;

}

This rule renders the content in black for only <h1> elements with id attribute set to black.

The true power of id selectors is when they are used as the foundation for descendant selectors, For example −

#black h2

{

color:#000000; }

In this example all level 2 headings will be displayed in black color when those headings will lie within tags having id attribute set to black.

The Child Selectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example −

body > p { color: #000000;

}

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text −

input[type = "text"] { color: #000000; }

The advantage to this method is that the <input type = "submit" /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

* p[lang] − Selects all paragraph elements with a langattribute.
* p[lang="fr"] − Selects all paragraph elements whose lang attribute has a value of exactly"fr".
* p[lang~="fr"] − Selects all paragraph elements whose lang attribute contains the word"fr".
* p[lang|="en"] − Selects all paragraph elements whose lang attribute contains values that are exactly "en", or begin with"en-".

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example –

h1 {

Color: #36C;

Font-weight: normal; letter-spacing: .4em; margin-bottom: 1em;

Text-transform: lowercase;

}

Here all the property and value pairs are separated by a **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find complete detail about properties in CSS References

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example −

h1, h2, h3 {

color: #36C;

font-weight: normal; letter-spacing: .4em; margin-bottom: 1em;

text-transform: lowercase;

}

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *id* selectors together as shown below –

#content, #footer, #supplement {

position: absolute; left: 510px;

width: 200px;

}

#### Bootstrap

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for creating common user interface components like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and soon.

Bootstrap gives your ability to create flexible and responsive web layouts with much less efforts.

Bootstrap was originally created by a designer and a developer at Twitter in mid-2010.Beforebeing an open-sourced framework, Bootstrap was known as Twitter Blueprint.

You can save a lot of time and effort with Bootstrap. So bookmark this website and continue on.

**What You Can Do with Bootstrap**

There are lot more things you can do with Bootstrap.

* You can easily create responsive websites.
* You can quickly create multi-column layout with pre-defined classes.
* You can quickly create different types of form layouts.
* You can quickly create different variation of navigation bar.
* You can easily create components like accordions, modals, etc. without writing any JS code.
* You can easily create dynamic tabs to manage large amount of content.
* You can easily create tool tips and popover’s to show hint text.
* You can easily create carousel or image slider to show case your content.
* You can quickly create different types of alert boxes.

**Advantages of Using Bootstrap**

If you have had some experience with any front-end framework, you might be wondering what makes Bootstrap so special. Here are some advantages why one should opt for Bootstrap framework:

* **Save lots of time**—you can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.
* **Responsive features** — Using Bootstrap you can easily create responsive web sites that appear more appropriately on different devices and screen resolutions without any change in mark-up.
* **Consistent design** — all Bootstrap components share the same design templates and styles through a central library, so the design and layout of your web pages will be consistent.
* **Easy to use** —Bootstrap is very easy to use. Anybody with the basic working knowledge of HTML, CSS and JavaScript can start development with Bootstrap.
* **Compatible with browsers** — Bootstrap is created with modern web browsers in mind and it is compatible with all modern browsers such as Chrome, Firefox, Safari, Internet Explorer, etc.
* **Open Source**—and the best part is, it is completely free to download and use.

**Creating Fluid Layout with Bootstrap**

In Bootstrap you can use the class .container-fluid to create fluid layouts to utilize the100% width of the view port across all device sizes (extra small, small, medium, large, and extra-large).

The class .container-fluid simply applies the width: 100% instead of different width for different view port sizes. However, the layout will still responsive and you can use the

grid classes as usual. See the tutorial on Bootstrap grid system to learn more about grid classes.

#### JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMA Script specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamictyping,prototype-basedobject-orientation,andfirst-classfunctions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior, often in comporting third-party libraries. All major web browsers shave a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). JavaScript engines were originally used only in web browsers, but they are now core components of other software systems, most notably servers and avarietyofapplications.AlthoughtherearesimilaritiesbetweenJavaScriptand Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

**Features Of JavaScript** JavaScript is divided into two main features, they are as follows- **General JavaScript Features-**

JavaScript language consists of several different features. Some of the general JavaScript features are as follows–

1. **Validating User’s Input-**

JavaScript is very useful while using forms. It has the capability to validate user input for errors and also saves time. If the user leaves a required field empty or the information is incorrect , JavaScript checks for them before sending the data over to the server.

1. **Simple Client-side Calculations-**

Since JavaScript is a client-side technology, it can perform basic calculations on the browser. The browser does not need to ask server time for every task. This is especially helpful when a user needs topper form these calculations repeatedly. In these cases, connecting to the server would take a lot more time than performing the actual calculations.

1. **Greater Control-**

JavaScript provides greater control to the browser rather than being completely dependent on the web servers. JavaScript provides various browsers with additional functionalities that help reduce server load and net work traffic.

1. **Platform Independent-**

Since browsers interpret JavaScript, it solves the problem of compilation and compatibility. Thus it can run on Windows, Macintosh, and other Netscape-supported systems. Also, it is possible to embed them in any other script like HTML that keeps JavaScript into use.

1. **Handling Dates and Time-**

Unlike other programming languages, JavaScript has built-in functions to determine the date and time. Thus it is very easy to code only by using methods like .getDate ().

1. **Generating HTML Content-**

JavaScript has very handy features to dynamically generate HTML content for the web. It allows us to add text, links, images, tables, etc after an event occurrence (example – mouse click).

1. **Detecting the User’s Browser and OS-**

JavaScript is very capable in the detection of the user’s browser and OS information. Though JavaScript runs on every platform, there may occur a situation where we need the user’s browser before processing. This can be helpful for writing code that results in different outputs in different browsers.

**Modern JavaScript Features**

If we dive into some more recently added features of JavaScript that makes it unique from other programming languages. There are a lot more modern features of JavaScript in vented after some general features. Some of them are as follows–

1. **Let/Const-**

JavaScript has introduced the keywords ‘let’ and ‘const’ that are available to replace ‘var’. Unlike ‘var’, they are important due to their blocked scope i.e we can only access them in the block we defined them in. Whereas ‘var’, even if we initialize it inside a function, we can access it outside of the function .

1. **Arrow Functions-**

These functions are very useful in simplifying the syntax and tamp down the lines of codes for the web page or web application. Since these are light-weight in syntax, they can be very easily used in anonymous functions in JavaScript.

1. **Template Literal-**

This is a common feature in other programming languages that allows you to save variables directly in to strings. This proves to be an important tool for developers as it

Permits them to focus more on the development of the application rather than spending the time on syntax .

1. **New Array Functions-**

Though array functions are not necessary for any programming language, they do simplify things for the developer. This also compacts the code and makes it much easier to understand. A regular array and an associative array, JavaScript supports them both. While a regular array contains integer values for its index, indexes can be strings for an associative array.

1. **Default Parameters-**

This JavaScript feature helps to avoid collapsing the whole code for a simple mistake. It is very useful when the developer needs to check the working of a function without any parameters.

1. **Property Shorthand-**

Built-in methods like .get () are available for the developer’s use. These methods help avoid writing the same code every time and cut back on various lines of code. These in born methods are really supportive of cutting back the developing time and cost.

**Back End Technology:**

###### PHP

**PHP: P**HP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by RasmusLerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team.

PHP originally stood for Personal Home Page, but it now stands for the recursive acronym

PHP Hypertext Preprocessor.

PHP is an acronym for "PHP: Hypertext Preprocessor"

PHP is a widely-used, open source scripting language.

PHP scripts are executed on the server.

PHP is free to download and use.

PHP is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

###### Features of PHP

**Simple, Familiar and ease of use:** Its popularly known for its simplicity, familiarity and easy to learn the language as the syntax is similar to that of ‘C’ or Pascal language.

So the language is very logical and well organized general-purpose programming language. Even people with a normal programming background can easily understand and capture the

use of language. PHP is very advantageous for new users as its a very reliable, fluent, organized, clean, demandable and efficient.

The main strength of PHP is the availability of rich pre-defined functions. The core distribution helps the developers implement dynamic websites very easily with secured data. PHP applications are very easy to optimize.

**Loosely typed language:** PHP encourages the use of variables without declaring its data types. So this is taken care at the execution time depending on the value assigned to the variable. Even the variable name can be changed dynamically.

**Flexibility:** PHP is known for its flexibility and embedded nature as it can be well integrated with HTML, XML, Javascript and many more. PHP can run on multiple operating systems like Windows, Unix, Mac OS, Linux, etc. The PHP scripts can easily run on any device like laptops, mobiles, tablets, and computer. It is very comfortably integrated with various Databases. Desktop applications are created using advanced PHP features. The executable PHP can also be run on command-line as well as directly on the machine. Heavyweight applications can be created without a server or browser.

It also acts as an excellent interface with relational databases.

**Open Source**: All PHP frameworks are open sources, No payment is required for the users and its completely free. User can just download PHP and start using for their applications or projects. Even in companies, the total cost is reduced for software development providing more reliability and flexibility..

PHP provides libraries to access these databases to interact with web servers. Developers are free to post errors, inspect codes and can contribute to code as well as bug fixing. Many frameworks like Codeignitor, Zend Framework, CakePHP make use of PHP

It supports a popular range of databases like MySQL, SQLite, Oracle, Sybase, Informix, and PostgreSQL Even many popular content management systems like WordPress, Joomla and Drupal use PHP as prime language. Because of the above reasons many web hosting companies and Internet Service providers prefers PHP.

**Cross-platform compatibility:** PHP is multi-platform and known for its portability as it can run on any operating System and windows environments. The most common are XAMPP (Windows, Apache Server, MySQL, Perl, and PHP) and LAMP (Linux, Apache, MySQL, PHP). As PHP is platform-independent, it’s very easy to integrate with various databases and other technologies without re-implementation. It effectively saves a lot of energy, time and money.

**Active community support:** PHP is very rich with many diverse online community developers to help beginners for web-based applications. These worldwide volunteers contribute many features as well as new versions for PHP libraries. Even they contribute a translation in different languages to help out programmers. There is a bundle of third-party open-source libraries which provide basic functionalities. Even the documentation given by the official site helps in implementing new features providing access to a variety of creative imagination.

**Fast and efficient performance:** Users generally prefer fast loading websites. For any web development, speed becomes an important aspect which is taken care of by PHP.

PHP scripts are faster than other scripting languages like ASP.NET, PERL, and JSP. The memory manager of PHP 7 is very optimized and fast as compared to older versions of PHP. Even connecting to the database and loading of required data from tables, are faster than other programming languages. It provides a built-in module for easy and efficient database management system. The high speed of PHP is advantageous for users for its server administration and mail functionality. Also, it supports session management and removing of unwanted memory allocation..

**Third-party application support and security:** Many PHP’s predefined functions support data encryption options keeping it more secure. Even the users can use third-party applications to secure data. Real time access monitoring: PHP also provides a summary of user’s recent logging accesses.

**Memory and CPU usage information:** PHP can provide memory usage information from functions like memory\_get\_usage() or memory\_get\_peak\_usage(), which can help the developers optimize their code. In the similar way, the CPU power consumed by any script can be retrieved for further optimization.

**DATABASE**

#### Introduction to MySQL:

It is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Wideners](https://en.wikipedia.org/wiki/Michael_Widenius) daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language.](https://en.wikipedia.org/wiki/Structured_Query_Language) The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License,](https://en.wikipedia.org/wiki/GNU_General_Public_License) as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-](https://en.wikipedia.org/wiki/Business) [profit](https://en.wikipedia.org/wiki/Business) firm, the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB,](https://en.wikipedia.org/wiki/MySQL_AB) now owned by [Oracle Corporation.](https://en.wikipedia.org/wiki/Oracle_Corporation) For proprietary use, several paid editions are available, and offer additional functionality MySQL is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++.](https://en.wikipedia.org/wiki/C%2B%2B)

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets

**REQUIREMENT AND ANALYSIS**

**5. Requirement and Analysis**

In systems engineering and software engineering, requirements analysis refers to the processes used to identify the requirements that must be met for a new or modified product, taking into account any potential conflicts between the various requirements identified during the process of analysing, documenting, validating, and managing software or system requirements.

The identification of the need for change to enhance or improve an existing system is the first step in the system development life cycle. A preliminary analysis of the current system was done. The hospital's current system is entirely manual. Several issues were discovered during the preliminary analysis of the current system. The process of determining the requirements or conditions that must be met for a new or modified product, while taking into account the various analyzing , documenting, validating, and managing software or system requirements, is known as requirements analysis in systems engineering and software engineering.

The identification of the need for change to enhance or improve an existing system is the first step in the system development life cycle. A preliminary analysis of the current system was done. The hospital's current system is entirely manual. many issues were discovered during the preliminary analysis of the current system. By speaking with people and learning about their wants, the analyst must determine the requirements. It should go without saying that the success of any software development endeavor depends on having a thorough understanding of the software requirements. all future research, including system analysis.

Poorly analyzed and specified software will let down the user and present a brief to the developer. System design and coding will depend on how precise and well understood the requirements are. No matter how carefully thought out and executed are frequently false of misunderstanding are very high, and ambiguity is likely because a communication gap between the developer and the client is certain to cause confusion.

The task that needs to be completed is stated in a clear and simple heading that serves as the foundation for requirements understanding. The requirements are then stated precisely and in a technical manner.

**5.1 Requirement Specification**

**1. A planned approach to work:** The organization's work will be carefully organised and planned. In data stores, the information will be appropriately stored, facilitating both its storage and retrieval.

**2. precision:** The proposed system will have a better level of precision. Every action would be conducted correctly, and it would guarantee the accuracy of any information coming from the facility.

**3. Reliability:** As a result of the aforementioned factors, the suggested system will be highly reliable. The system will be more reliable because there will be better information storage going forward.

**4. No Redundancy:** In the proposed system, every effort would be made to ensure that no information—in storage or elsewhere—is repeated. This would guarantee efficient storage space use and data consistency.

**5. Instantaneous information retrieval:** The major goal of the suggested system is to enable an effective and speedy information retrieval. Any kind of information would always be accessible to users.

**6. Immediately storing information:** In a manual system, it is difficult to record the most information possible.

**7. Simple to Operate:** The system should be simple to use, able to be constructed quickly, and able to match the user's limited budget.

* 1. **Identification of Needs**

Identification of need mainly concern with requirements of the project how much time it would be taken and how much cost it would require to complete. The hardware requirement of project also concerns. The member of the project team is also decided in this phase.

Here we focus our points towards organizing needs i.e. to organize the things which are scattered here and there. Need to gather information about various aspects like customer information, bill calculation and reports. Each & every level of entity that is involving in the management system needs to communicate together with the help of management system.

#### Preliminary Investigation

A preliminary investigation is carried out by the system analyst working under the direction of the steering committee to evaluate project request. Preliminary investigation is the collecting of information that permits committee members to evaluate the merits of the project request and make informed judgment about feasibility of the proposed system.

In preliminary investigation we should accomplish the following objectives: -

**1: -** Clarify and understanding of the project request.

**2: -** Determine the size of the Project.

**3: -** Assess costs and benefits of alternative approaches.

**4: -** Determine the technical and operational feasibility of alternative approaches.

**5:-**Reports the findings to management, with the recommendation outlining the acceptance or rejection of the project.

**FEASIBILITY STUDY:**

Given infinite time and resources, all endeavours are possible. regrettably, the Lack of resources and delivery deadlines are frequently more problematic for computer-based system development. As a result, we have utilized the idea of reusability, which is central to object-oriented programming (OOPS).

The benefits and flexibility of the project are detailed in the feasibility report. There are three parts to this:

1. Economical Feasibility

2. Technical Feasibility

3. Behavioral Feasibility

**Economic Feasibility:** The most popular technique for assessing a potential system's efficacy is economic analysis. The process, more often known as cost/benefit analysis, is calculating the predicted savings and advantages from a candidate while contrasting them with the expenses. The choice to develop and deploy the system is made if advantages outweigh expenses.

A system's financial gain must outweigh its development costs. i.e., the organisation should benefit from the investment of a new system that is being built. Economic viability takes into account the following:

i. The price of performing a thorough system investigation.

ii. The price of the necessary hardware and software for the application class.

iii. The advantages in terms of lower costs or fewer expensive mistakes.

iv. The price if nothing is done (that is, if the suggested system is not created).

The system needs relatively little time, hence the proposed E- Energy is practically possible.

ii. The system will offer a quick and effective automated environment rather than a cumbersome and error-prone manual approach, hence requiring less time and labour to operate.

iii. The system will have a GUI interface, which will require relatively little user training.

iv. The system will let users access a variety of data in order to make wise administrative decisions.

**Technical Feasibility:** Technical viability focuses on the current computer system's capabilities to accommodate the suggested addition (hardware, software, etc.).

This project has taken all the essential precautions to make it technically feasible.

Using a key, text or an object is displayed very quickly. Additionally, the programming language, operating system, and tools utilised in this localization process are compatible with the current ones. Analyse the technological setup needed for the system for buying and selling. Take into account the architecture's ability to handle numerous users, vast amounts of data, and concurrent transactions. Examine the compatibility and integration potential with current platforms or systems. Analyse the technological and infrastructural requirements for the system. Take into account the system's hardware (servers, storage, and networking), software (databases, frameworks, and APIs), and other technological requirements.

**Behavioral Feasibility:** Computers have a history of facilitating change, but people are by nature resistive to change. The likelihood of the user personnel reacting negatively to the creation of a computerised system should be estimated. Therefore, it makes sense that extra efforts will need to be made to inform and teach the employees before the installation of a candidate system. The programmer currently in development is user-friendly and simple to understand. In this manner, the generated programmer is actually effective and adaptable to any conditions, customs, or geographies. The goal of behavioral research is to ensure that the organization's equilibrium and status quo are not upset and that users are open to change.

**METHODOLOGY ADOPTED**

**Prototype Model:**

Before full-scale production or development, a prototype model is an early version or representation of a product, system, or design that is made to test and validate concepts, features, and functioning. It is a physical or visual representation that enables users to engage with the model and offer suggestions for improvement.

**Prototype models have a variety of uses, such as:**

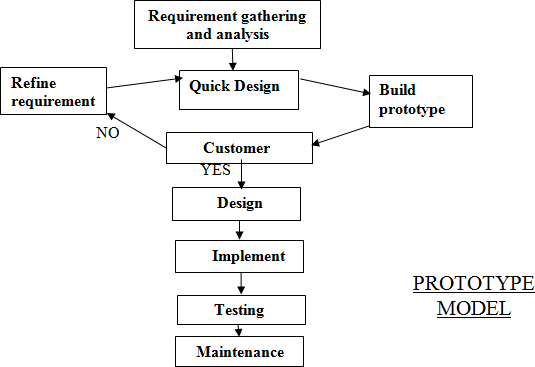
**Proof of Concept:** A prototype enables the validation of an idea's or concept's viability. It enables designers, engineers, and other interested parties to assess and decide whether the suggested solution is workable and capable of fulfilling the required specifications.

**User Input:** Early in the design phase, developers can collect user input by making a prototype. Users can interact with the prototype and offer insightful feedback and improvement recommendations, ensuring that the final product satisfies user requirements and expectations.

**Iterative Design:** Using prototypes makes it possible to iterate and enhance designs over time in response to user input and testing. By identifying design faults, usability problems, and prospective problems early on, this iterative technique helps produce products more quickly and efficiently.

**Communication and Collaboration:** Prototypes serve as a communication and collaboration tool for a variety of stakeholders, including designers, developers, project managers, and clients. They aid in facilitating talks, delineating specifications, and coordinating everyone's comprehension of the system or product.

Depending on the project's needs, prototype models might take on a variety of shapes. Simple paper sketches or wireframes can be used, as well as more complex physical models, interactive simulations, or working software prototypes. The precise objectives, available resources, and project-specific limitations all influence the prototype model selection.



A prototype is the sample implementation of the system that shows limited and main functional capabilities of the proposed system. After a prototype is built, it is delivered to the customer for the evaluation. The prototype helps the customer determine how the feature will function in the final software. The customer provides suggestion and improvements on the prototype. The development team implements the suggestion in the new prototype, which is again evaluated by the customer.

**Hardware & Software Requirements**

##### The Following are the Minimum Hardware and Software requirement to run the proposed system.

**Hardware Requirement:**

|  |  |
| --- | --- |
| **COMPONENT** | **MINIMUM SPECIFICATION** |
|  |  |
| 1. Processor | Pentium IV |
| 1. Memory | 1 GB |
| 1. HDD | 2 GB |

**Software Requirement :**

|  |  |
| --- | --- |
| **Use of Software** | **Name of Software** |
|  |  |
| 1. Operating System | Window 7, XP-SP3 |
| 1. Front End | PHP, HTML |
| 1. Back End | MySql 5.5 |
| 1. For Documentation | MS-Office 2007 |

**6.PROBLEM DEFINITION**

Challenges associated with the purchase, sale, and use of equipment related to energy are included in the problem definition for buying and selling energy equipment. Among the problem's crucial facets are:

**Market Complexity:** Energy equipment markets can be complicated and diversified, with a large variety of goods, technologies, and providers. It might be difficult for purchasers to find the appropriate equipment that satisfies particular needs, such as capacity, efficiency, and compatibility. Similar to buyers, sellers struggle to comprehend market demand, rivalry, and pricing dynamics.

**Technological Developments:** The energy industry is always changing due to technological developments, such as energy storage, smart grids, renewable energy systems, and energy management tools. Both consumers and sellers may find it difficult to keep up with these technology developments and comprehend their applications and advantages.

**Quality and Reliability:** It's important for buyers to ensure the quality and dependability of energy equipment. Failure or poor performance of equipment can result in operational hiccups, monetary losses, and safety issues. Sellers must build confidence by offering dependable, certified items that adhere to industry standards and have a track record of success.

**Cost Factors:** Buyers may need to make a sizable investment in energy equipment, therefore cost factors are quite important when making a choice. Buyers must weigh the equipment's initial expenses against its long-term advantages, such as energy savings, maintenance needs, and operational effectiveness. Offering competitive price while preserving profitability is a problem for sellers.

**Regulatory Compliance:** Energy equipment must adhere to a number of laws and standards to assure its performance, safety, and environmental sustainability. To make sure that the equipment complies with the required standards, buyers and sellers must traverse various regulatory regulations, certifications, and compliance procedures.

**Installation and Integration:** Integrating new energy equipment into an already-built infrastructure might be difficult. To maximise the functionality and advantages of the equipment, buyers must ensure appropriate installation, interoperability with current systems, and effective integration. To ensure a seamless integration, sellers might need to offer advice, assistance, and technical knowledge.

**Lifecycle Management and Sustainability**: Energy equipment has a lifecycle that involves installation, operation, maintenance, and eventually replacement. This is known as lifecycle management. For both buyers and sellers, managing the equipment's lifecycle, optimising maintenance schedules, and guaranteeing correct disposal or recycling at the end of its useful life are issues. Additionally important is supporting ecologically friendly disposal methods and energy-efficient equipment.

In order to successfully overcome these obstacles, buyers and sellers must cooperate effectively, be aware of market trends and technological developments, and have a thorough understanding of the requirements for each type of energy equipment. In order to guarantee quality, dependability, and compliance, industry standards, certifications, and regulatory frameworks are essential. Furthermore, promoting trust between buyers and sellers, giving clear information, and fostering transparency can all help to lessen the difficulties associated with buying and selling energy equipment.

**7.PLANNING AND SCHEDULING**

#### Project Planning:

The team decides what to construct, how to build it, and who will build it during the planning phase. The team also creates the functional specification, goes through the design process, and creates work plans, cost estimates, and timelines for the deliverables during this phase.

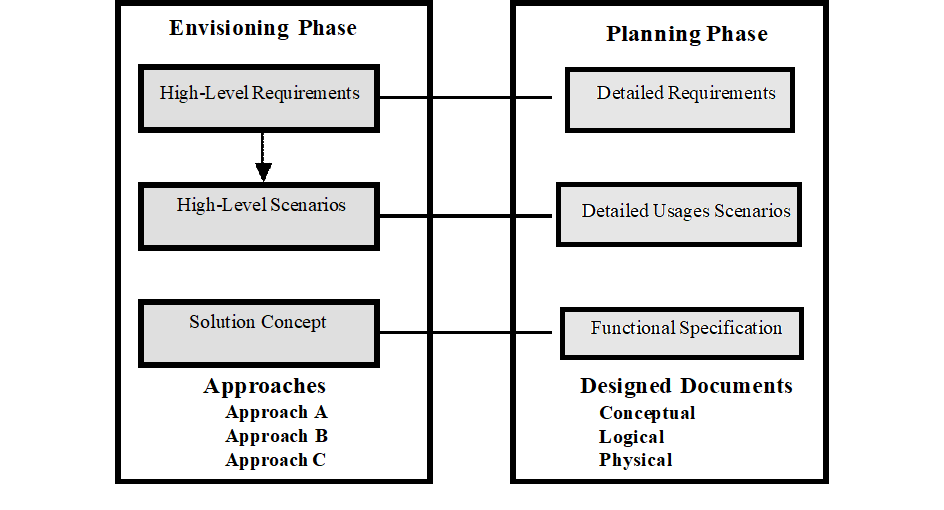


Fig.7

There are three design processes in the planning phase: conceptual, logical and physical design. These three processes are not parallel. There starting and ending points are staggered. There processes are dependent on each other.

Logical design is dependent on conceptual design, and physical design is dependent on logical design any changes to the conceptual design affect logical design leading to changes in the physical design.

|  |  |  |
| --- | --- | --- |
| **Type of design** | **Perspective** | **Purpose** |
| **Conceptual** | **View the problem from the** | **Define the problem and solution** |
| **Design** | **perspective of the user and** | **in term of usage scenarios** |
|  | **businesses** |  |
| **Logical Design** | **View the solution from the perspective of project team** | **Define the solution as a logical set of cooperating services** |
| **Physical Design** | **View the solution from the**  **perspective of the developer** | **Define the solution’s services and**  **technologies** |

#### Reliable System:

There are two levels of reliability. The first is meeting the right requirements. A careful and

through systems study is needed to satisfy this aspect of reliability. The second level of

systems reliability involves the actual working delivered to the user. At this level, the systems

reliability is interwoven with software engineering and development. There are three approaches to reliability:

1. **Error Avoidance:** Prevents errors from occurring in software.
2. **Error Detection and Correction:** In this approach errors are recognized whenever they are encountered and correcting the error by effect of error, of the system does not fail.
3. **Error Tolerance:** In this approach errors are recognized whenever they occur,

but

enables the system to keep running through degraded perform or by applying values that instruct the system to continue process.

## Maintenance:

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documentation.
3. Using more effective methods for designing, processing, login and communicating information with project team members.
4. Making better use of existing tools and techniques.
5. Managing system engineering process effectively.

#### Output Design:

One of the most important factors of an information system for the user is the output the system produces. Without the quality of the output, the entire system may appear unnecessary that will make us avoid using it possibly causing it to fail. Designing the output should process the in an organized well throughout the manner. The right output must be developed while ensuring that each output element is designed so that people will find the system easy to use effectively.

The term output applying to information produced by an information system whether printed or displayed while designing the output we should identify the specific output that is needed to information requirements select a method to present the formation and create a document report or other formats that contains produced by the system.

#### Types of output:

Whether the output is formatted report or a simple listing of the contents of a file, a computer process will produce the output.

* + A Document
  + A Message
  + Retrieval from a data store
  + Transmission from a process or system activity
  + Directly from an output source

#### Layout Design:

It is an arrangement of items on the output medium. The layouts are building a mockup of the actual reports or document, as it will appear after the system is in operation. The output layout has been designated to cover information. The outputs are presented in the appendix.

#### Input Design and Control:

Input specifications describe the manner in which data enter the system for processing. Input design features will ensure the reliability of the systems and produce results from accurate data, or thus can be

result in the production of erroneous information. The input design also determines whenever the user can interact efficiently with this system.

#### Objectives of Input Design:

Input design consists of developing specifications and procedures for data preparation, the steps necessary to put transaction data into a usable from for processing and data entry, the activity of data into the computer processing. The five objectives of input design are:

* Controlling the amount of input
* Avoiding delay
* Avoiding error in data
* Avoiding extra steps
* Keeping the process simple

#### Controlling the Amount of Input:

Data preparation and data entry operation depend on people, because labour costs are high, the cost of preparing and entering data is also high. Reducing data requirement expense. By reducing input requirement, the speed of entire process from data capturing to processing to provide results to users.

#### Avoiding Delay:

The processing delay resulting from data preparation or data entry operations is called bottlenecks. Avoiding bottlenecks should be one objective of input.

#### Avoiding Errors:

Through input validation we control the errors in the input data.

#### Avoiding Extra Steps:

The designer should avoid the input design that cause extra steps in processing saving or adding a single step in large number of transactions saves a lot of processing time or takes more time to process.

#### Keeping Process Simple:

If controls are more people may feel difficult in using the systems. The best-designed system fits the people who use it in a way that is comfortable for them.

**MODULE DISCRIPTION**

**8. MODULE DISCRIPTION**

The project contains several important modules, such as:

* **ADMIN OR DISTRIBUTOR:**

The DISTRIBUTOR module is the most crucial component of the system since it configures the entire workflow by determining whether the retailer is legitimate. Its primary duty is to retrieve information from the database and give it to the sales agent and the retailer. Additionally, it assesses the quantity of orders and the number of products and gets store feedback.

* **SALES AGENT:**

The system's next module, the sales agent module, is responsible for supplying the goods to each retailer. Sales agents are organisations that purchase non-competing products or product lines from distributors and resell them to retailers. It provides the distributor with the information it receives from the retailer. It serves the retailer as a go-between. It operates on the basis of the retailer-generated demand for the products.

* **RETAILER:**

The system's next module is the retailer. It provides the distributor all the information they need to register. It gives the sales representative an order for the popular goods. The retailer returns the receipt after giving the payment to the salesperson.

* **PAYMENT:**

All of the purchases completed by the retailer, sales agent, and distributor are included in the payment module.

* **STOCK:**

This module is maintained by the distributor as it consists of the product details i.e. model no. , color, product id, etc. which are the details of the product as well as carries the information of stock level and reorder level.

**ANALYSIS DIAGRAMS**

**9.Analysis Diagrams**

**9.1 DATA FLOW DIAGRAM**

A data flow diagram (DFD) shows how data moves through a system or process graphically. The transit of information between different system components is frequently visualised in software engineering and systems analysis. Without delving into the more specific implementation details, DFDs offer a high-level picture of how data is entered, processed, stored, and outputted inside a system. Data flow diagrams are process-oriented models that show how data moves from one system process to another. A data flow diagram is a visual aid for describing and analysing how data moves through a system. These serve as the main resource and serve as the foundation for the creation of the other components. The logical transition of data from input to output through processing can be explained without reference to the system's physical components. The logical data flow diagrams are what these are called. The real tools and methods used to transfer data between individuals, offices, and workstations are depicted in the physical data flow diagrams. A collection of data flow diagrams serve as the complete description of a system. The data flow diagrams are created using two well-known notations, Yourdon, Gane, and Sarson notation. In a DFD, each component is identified by a name that is illustrative. A number that will be used for identifying purposes is also utilised to better identify the process. DFD development takes place on a number of levels. Each process in the lower level diagrams can be further delineated in the higher level DFD. The context diagram is another name for the lop-level diagram. It just has one process bit, but that process bit is crucial to understanding how the current system works.

In a data flow diagram, there are four main components:

**Processes:** These stand in for the numerous tasks or actions used to manipulate data. Processes convert incoming data into departing data and are frequently represented as circles or rectangles.

**Data Flows:** These show how data is transferred between various processes, outside entities, or data repositories. Arrows used to represent data flows show the direction of the data flow.

**External Entities:** These are external entities that interact with the system under study. They might be data sources or destinations. They may be users, different systems, or businesses.

**Data Stores:** These are locations in the system where data is kept in repositories. They could be files, databases, or some other kind of data storage.

**DFD SYMBOLS:**

**Process (Circle):** Depicts a data-transforming function, task, or activity. Processes are given names that indicate their purpose and are identified by special IDs.

**Data Flow (Arrow):** Represents the flow of data between various system components. Data flows are labelled to identify the data being transported and highlight the direction of data transfer.

**External Entity (Square or Rectangle):** An external source or destination of data that interacts with the system under study is represented by an external entity (Square or Rectangle). Users, other systems, and organisations are examples of external entities.

**Data Store (Open-ended Rectangles):** Represents a location in the system where data is kept. Databases, files, and other types of data storage can all be considered data stores. To identify the kind of data stored, they are labelled.

Process

Source or Destination of Data

Data flow

Data Store

#### CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD’s:

1. Processes should be given names and numbers for quick identification. Each name need to reflect the procedure it refers to.

2. The flow is in a top-to-bottom and left-to-right orientation. Data They typically flow from the source to the destination, though they can also travel the other way. Drawing a long flow line back to a source is one approach to show this. Repetition of the source sign as a destination is an alternate method. A short diagonal is used to indicate that it appears more than once in the DFD.

3. Lower level details of a process that have been exploded are given numbers.

4. Capital letters are used to identify data repositories and destinations. The first letter of each word in the names of processes and dataflows is capitalised.

A DFD typically displays the bare minimum of the data

store's contents. All the data components that flow into and out of any data store should be present.

All the data pieces that come in and go out should be included in the questionnaires. Then, interviews are frequently used to account for missing interfaces, redundancies, and the like.

**SAILENT FEATURES OF DFD’s:**

1. Data flow diagrams (DFDs) only depict data flow; they do not depict control loops or decision-controlling factors.

2. Whether data flows occur daily, weekly, monthly, or yearly, the DFD does not identify the time factor involved in any activity.

3. The DFD does not outline the order of occurrences.

#### TYPES OF DATA FLOW DIAGRAMS:

* + 1. Current Physical
    2. Current Logical
    3. New Logical
    4. New Physical

1. **CURRENT PHYSICAL:**

A description of the technology used to process the data is included in the Current Physical DFD process label, along with the names of any individuals or their positions or computer systems that may contribute to portion of the overall system processing. Similar to how data flows and stores are frequently identified by the names of the actual physical media—such as file folders, computer files, business forms, or computer tapes—on which data are kept.

**2.) CURRENT LOGICAL:**

To the greatest extent possible, the system's physical components are eliminated, leaving only the data and the processors that change them, regardless of their actual physical shape, as the remaining components of the system.

1. **NEW LOGICAL:**

The new logical model will differ from the current logical model while having additional functions, absolute function removal, and recognition of inefficient flows. This is exactly like a current logical model if the user was completely satisfied with the functionality of the current system but had issues with how it was implemented.

**4.) NEW PHYSICAL:**

The new physical represents only the physical implementation of the new system.

#### RULES GOVERNING THE DFD’S:

#### PROCESS:

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase label.

#### DATA STORE:

1. Data cannot be transferred directly from one data store to another; a process is required.

2. Data cannot be transferred directly from an external source to a data store; instead, a process that receives the data must transfer the data from the source and store it in the data store.

3. A data storage has a label for a noun phrase.

#### SOURCE OR SINK:

The origin and /or destination of data.

**1.)** Data cannot move direly from a source to sink it must be moved by a process

**2.)** A source and /or sink has a noun phrase land.

#### DATA FLOW:

1) There is only one direction of flow between symbols in a data flow. A read may occur before an update in both directions between a process and a data storage. The latter is typically represented by two distinct arrows, though, because these occur at different types.

2) In DFD, a join denotes the exact same data coming from any two or more distinct processes' data storage or sink to a shared place.

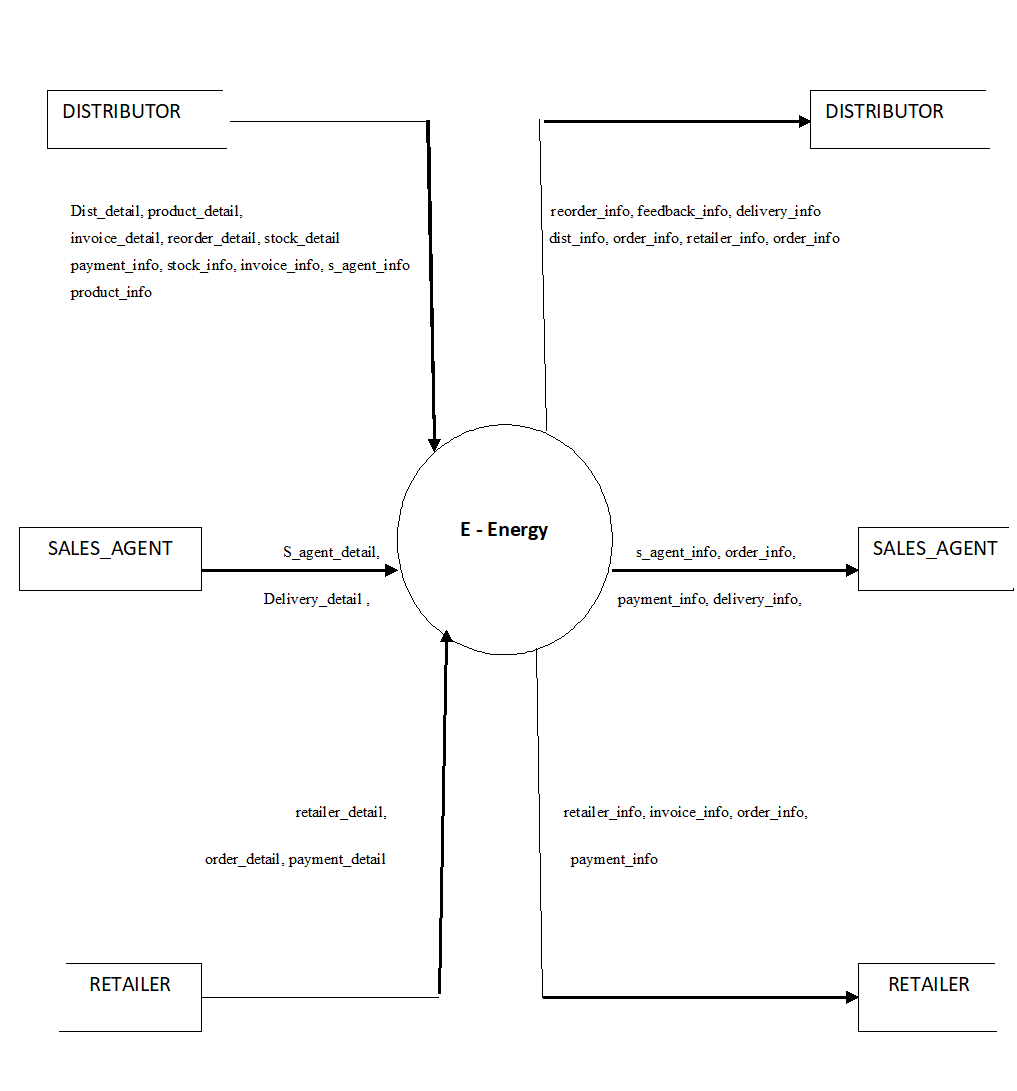
3. A data flow cannot immediately return to the process it originated from. There must be at least one additional process that manages the data flow, produces another data flow, and then feeds the starting process with the original data.

4) An update (delete or change) is indicated by a data flow to a data store.

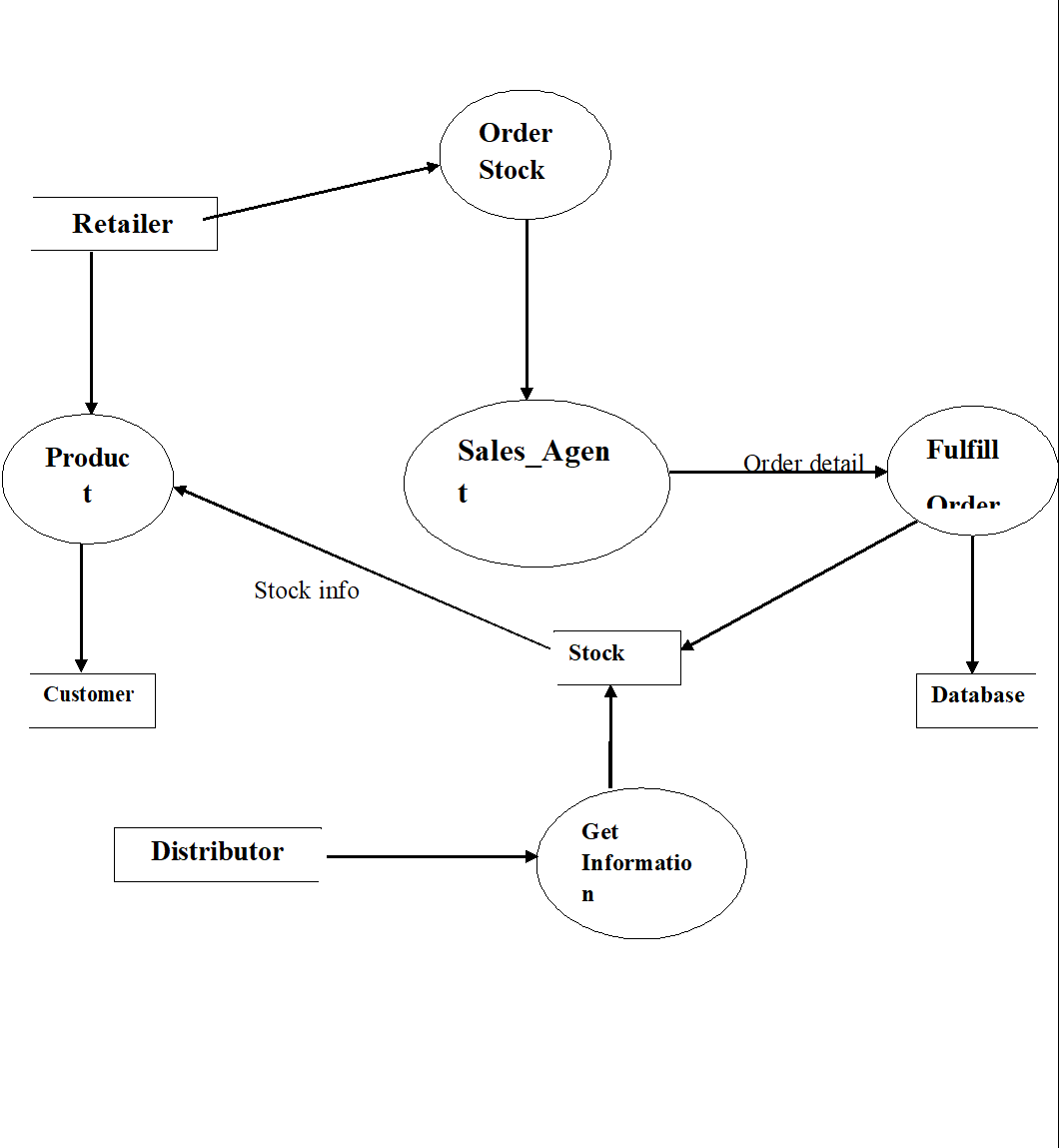
5) To retrieve or use data from a data repository.

More than one data flow noun phrase may appear on a single arrow as long as all of the flows travel as a single package. A data flow has a noun phrase label.

**CONTEXT DIAGRAM(0 LEVEL DFD)**

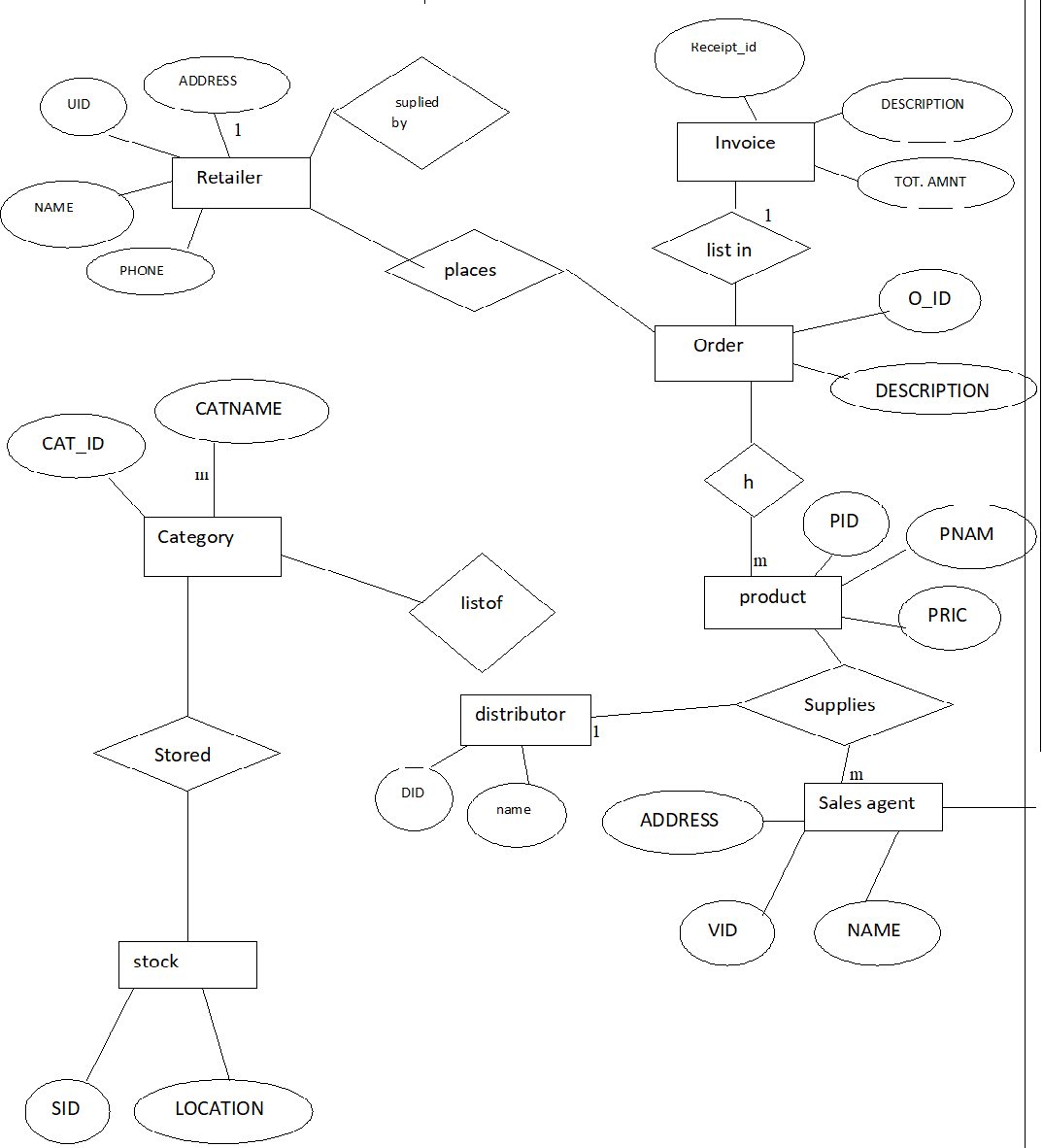
**0 Level DFD**

**CONTEXT DIAGRAM (1 LEVEL DFD)**



**1 Level DFD**

**Entity Relationship Diagram:**



**ER DIAGRAM**

**SYSTEM DESIGN**

# SYSTEM DESIGN

#### THE SYSTEM DESIGN PROCESS:

System design develops the architectural detail required to build a system or product.

The system design process encompasses the following activities:

* Partition the analysis model into subsystems.
* Identify concurrency that is dictated by the problem.
* Allocate subsystems to processors and tasks.
* Develop a design for the user interface.
* Choose a basic strategy for implementing data management.
* Identify global resources and the control mechanisms required to access them.
* Design an appropriate control mechanism for the system, including task management.
* Consider how boundary conditions should be handled.
* Review and consider trade-offs.

OUTPUT DESIGN:

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization.
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly with system.

#### Output Definition:

The outputs should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output
* It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

#### For Example:

Will decimal points need to be inserted? Should leading zeros be suppressed.

#### Output Media:

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

* The suitability for the device to the particular application.
* The need for a hard copy.
* The response time required.
* The location of the users
* The software and hardware available.
* The cost.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hot copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

#### INPUT DESIGN:

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

#### INPUT STAGES:

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

#### INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

#### INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability
* Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

#### ERROR AVOIDANCE:

* At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

#### ERROR DETECTION:

* Even though every effort is making to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

#### DATA VALIDATION:

* Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.
* The system is designed to be a user friendly one. In other words, the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

## 10.1 USER INTERFACE DESIGN

It is essential to consult the system users and discuss their needs while designing the user interface:

#### USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:

User initiated interface the user is in charge, controlling the progress of the user/computer dialogue.In the computer-initiated interface, the computer selects the next stage in the interaction.

#### COMPUTER INITIATED INTERFACES:

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer acts or displays further information.

#### USER INITIATED INTERFACES:

User initiated interfaces fall into two approximate classes:

* Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
* Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

#### COMPUTER-INITIATED INTERFACES:

The following computer – initiated interfaces were used:

The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.

Questions – answer type dialog system where the computer asks question and acts based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

#### ERROR MESSAGE DESIGN:

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

#### Performance Requirements:

Performance is measured in terms of the output provided by the application.

Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

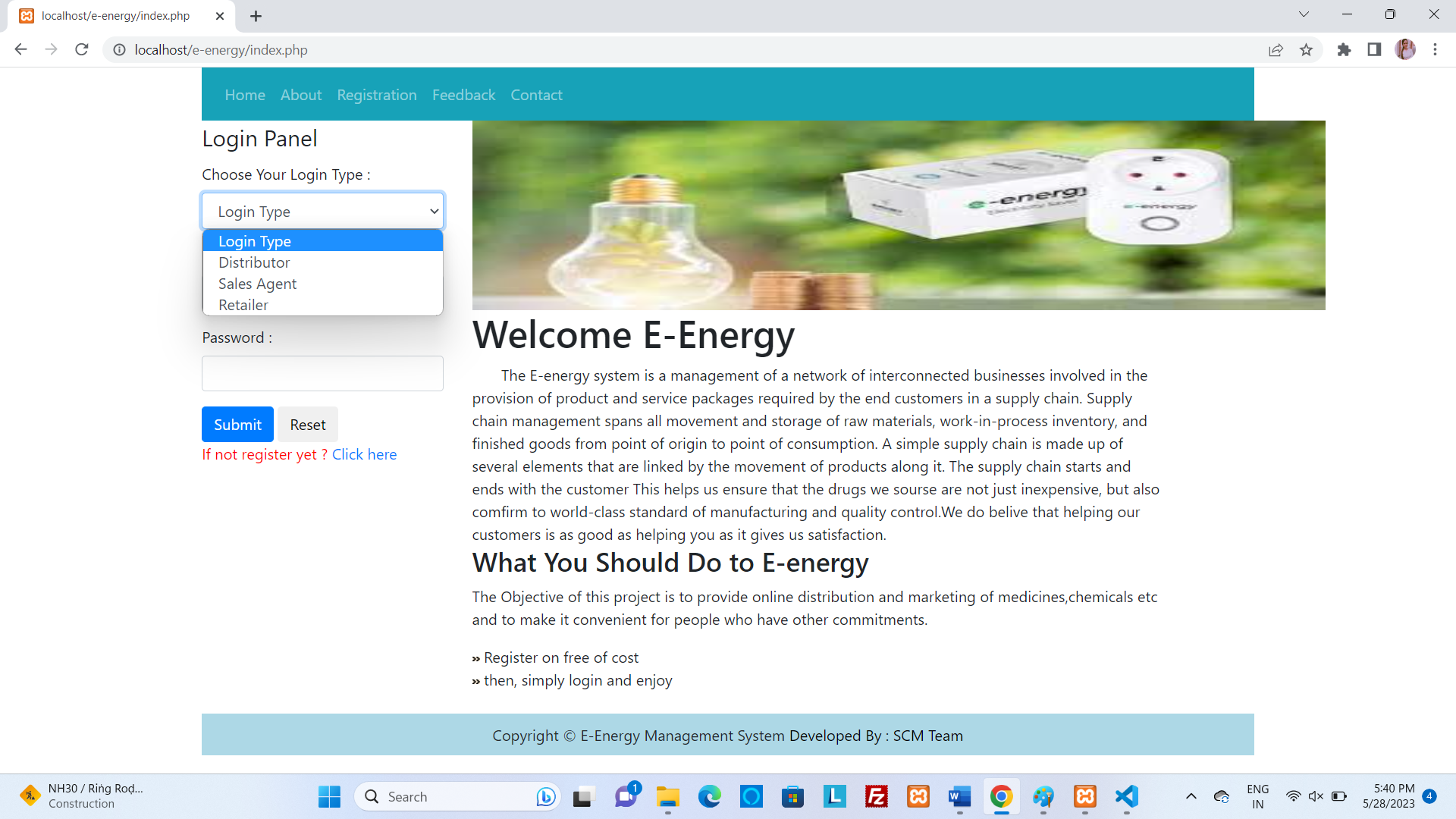
* The requirement specification for any system can be broadly stated as given below:
* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system
* The existing system is completely dependent on the staff to perform all the duties.

**INPUT/OUTPUT LAYOUT INTERFACES**

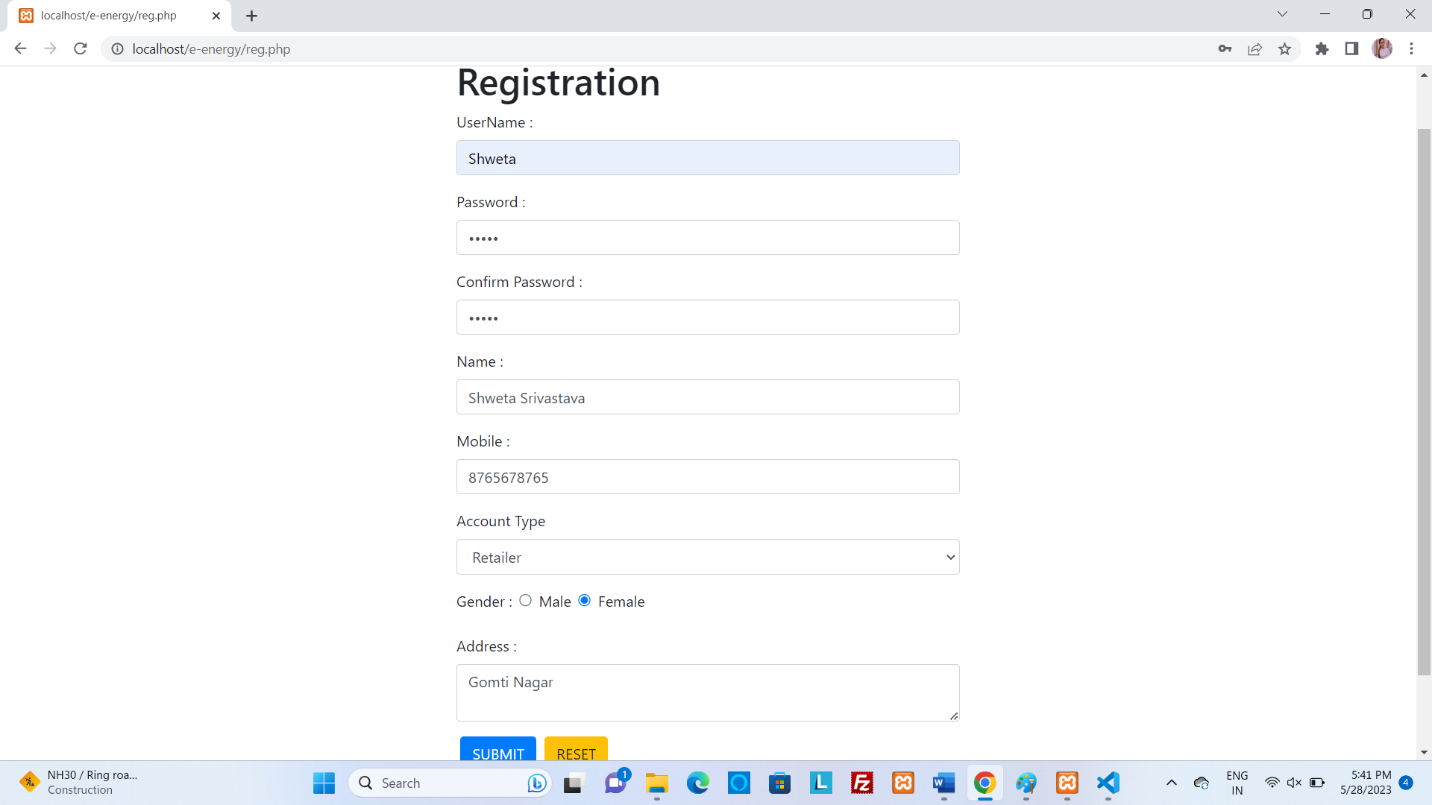
# INPUT/OUTPUT LAYOUT INTERFACES

# INPUT/OUTPUT LAYOUT INTERFACES

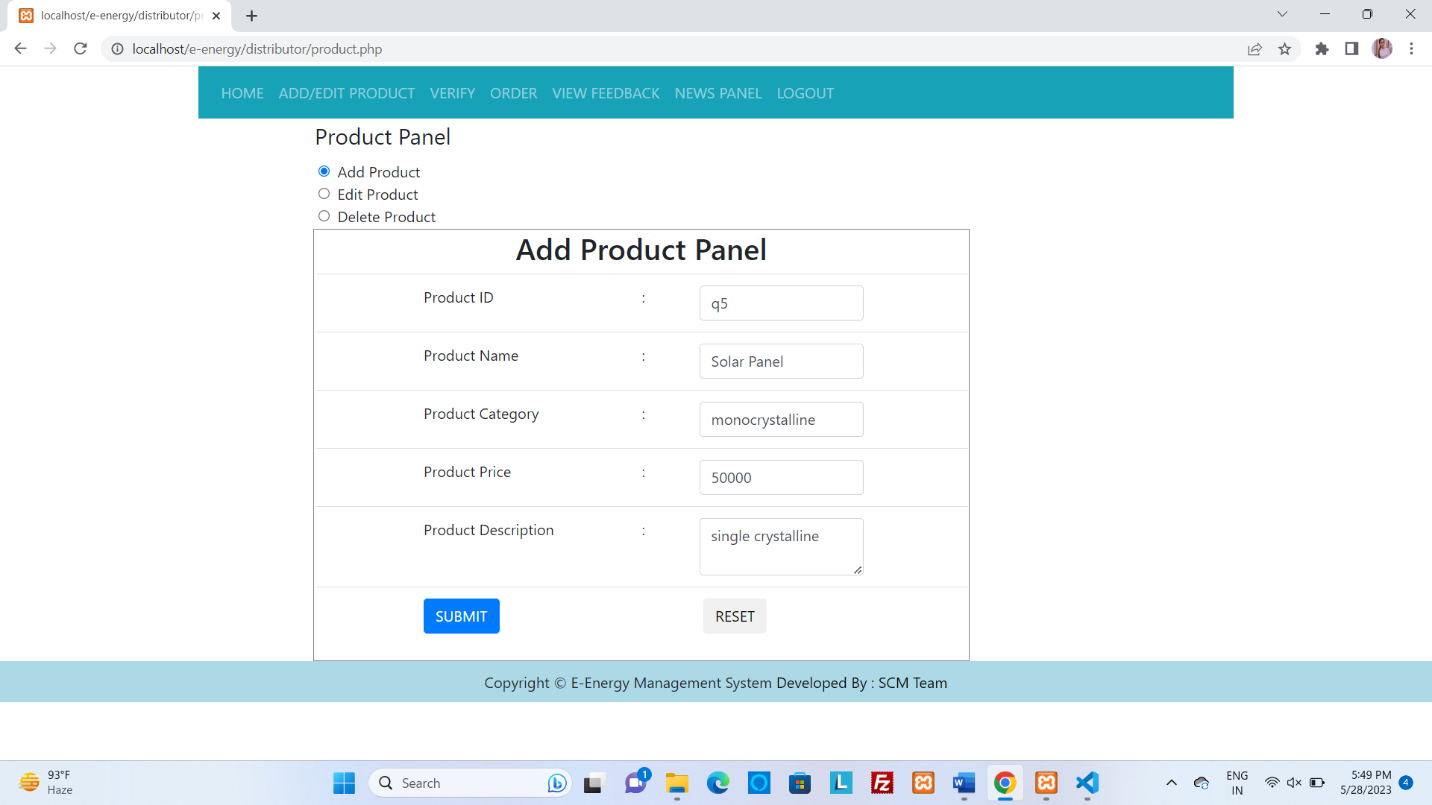
**INTERFACE 1: Distributor Login Panel:**

****

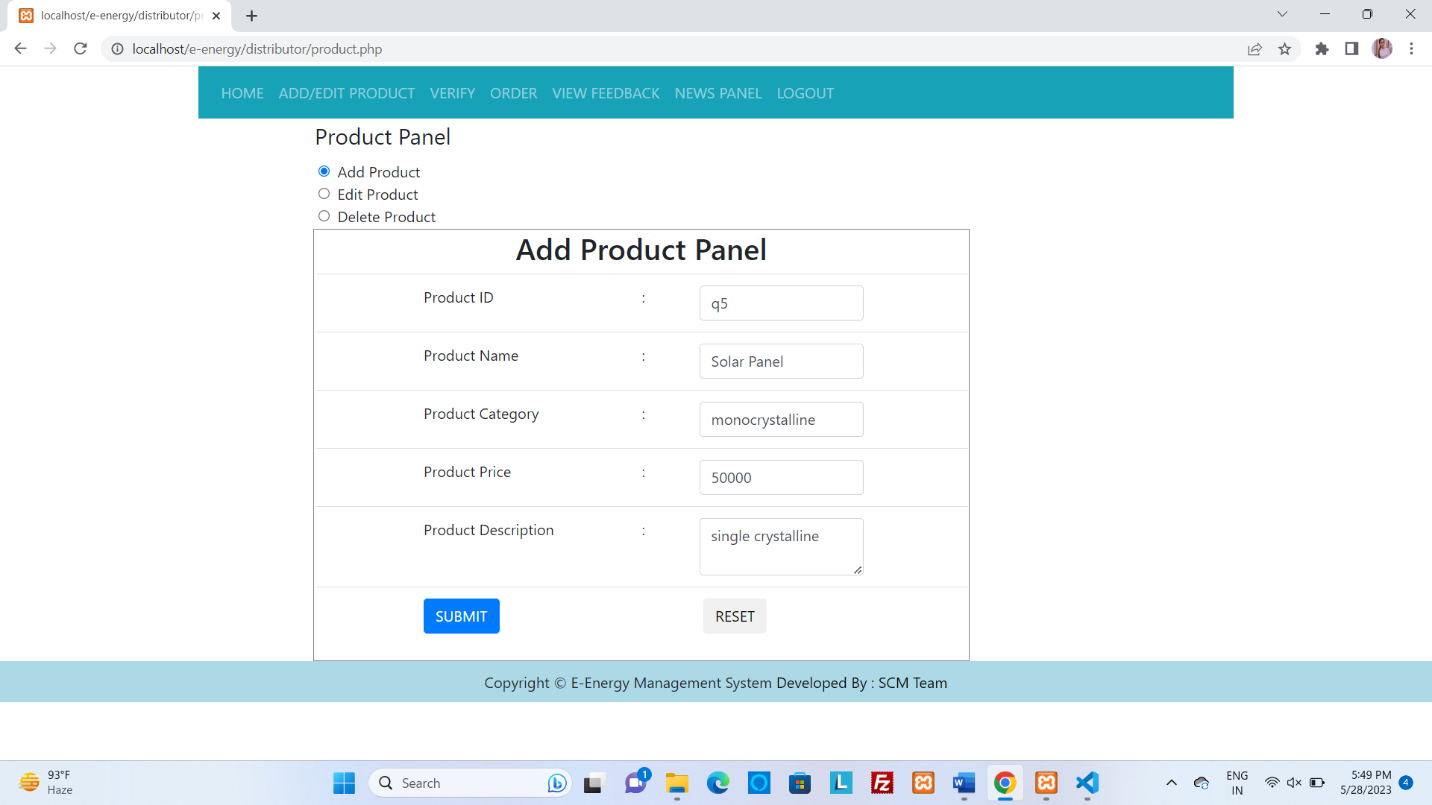
**INTERFACE 2: Registration:**

****

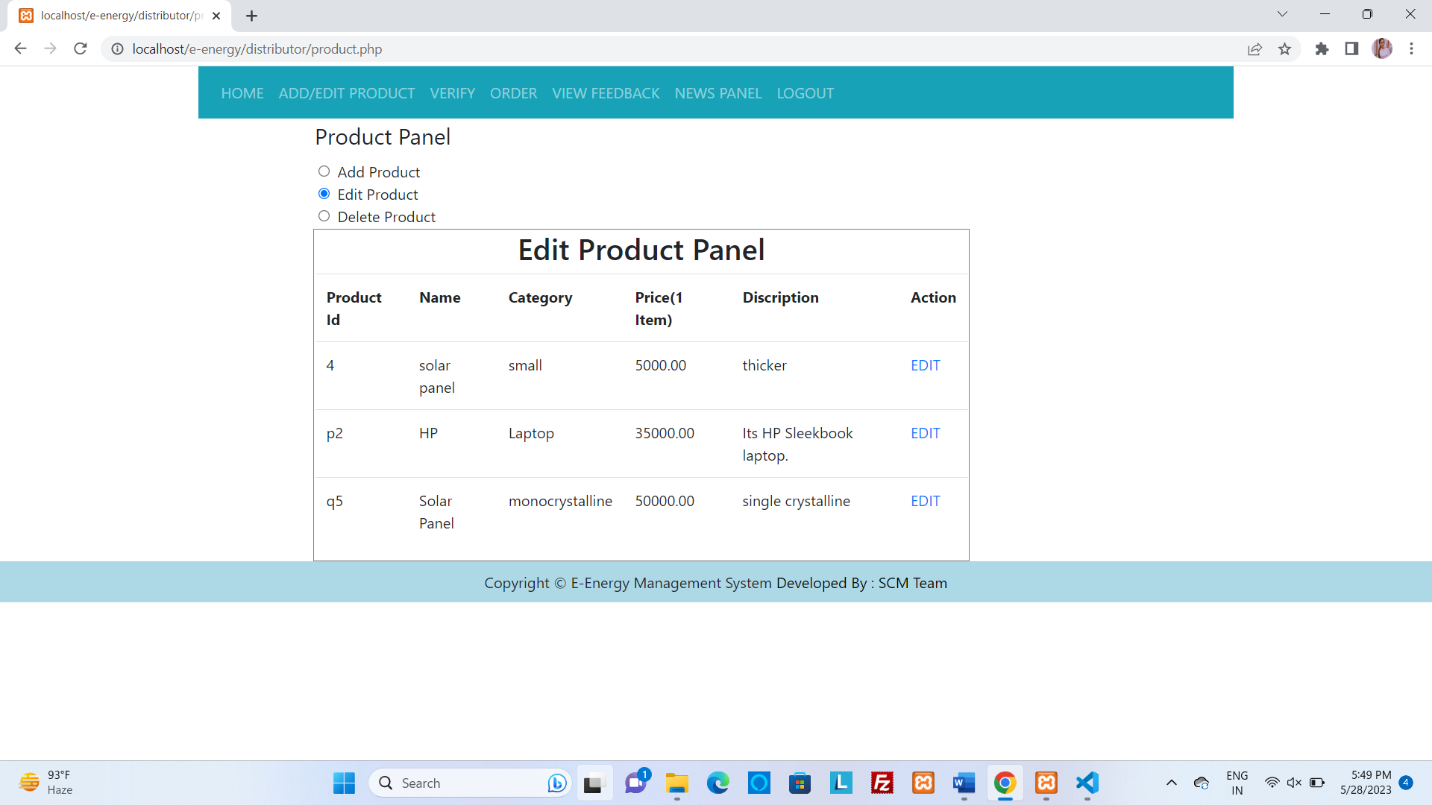
**INTERFACE 3: Distributor Login Panel:**

****

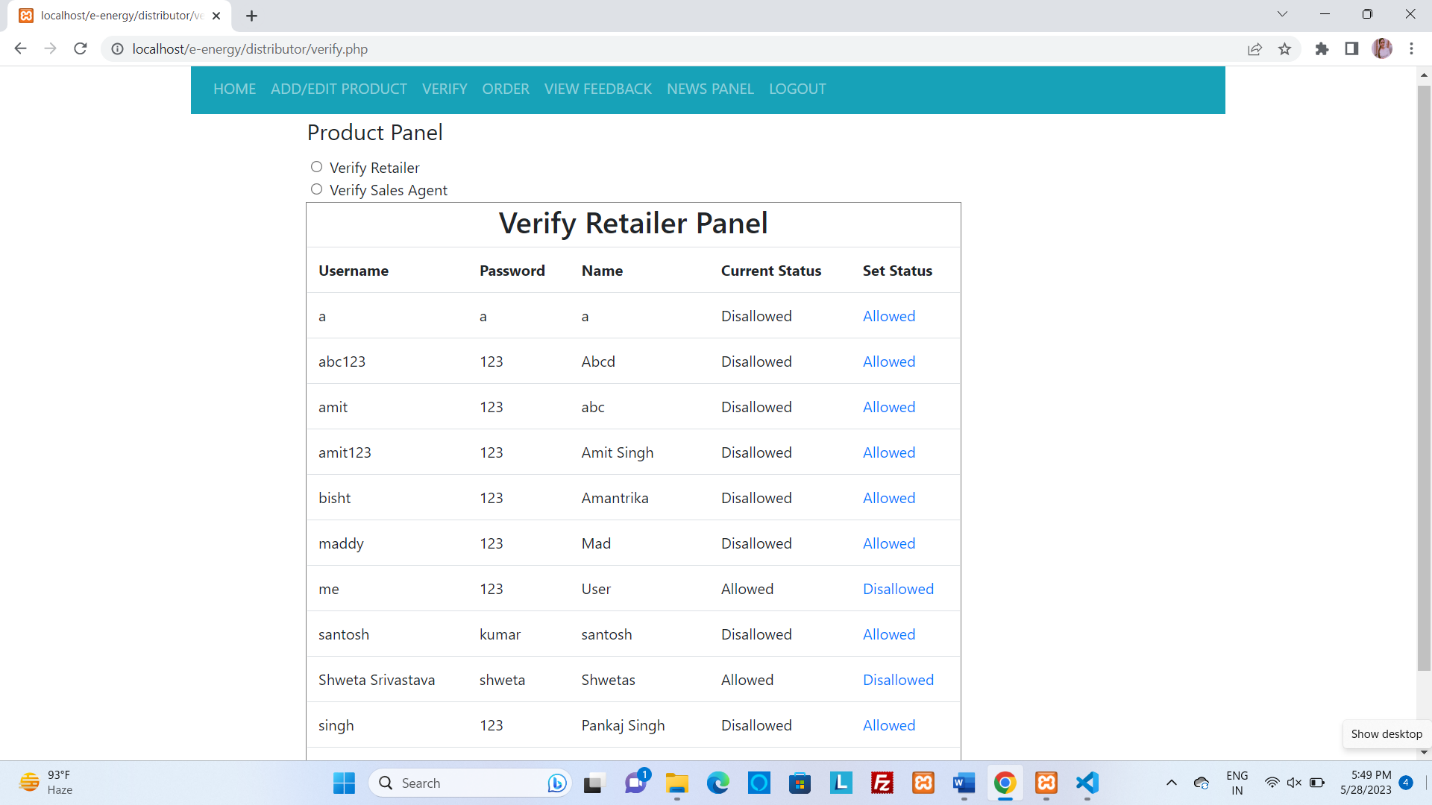
**INTERFACE 4: Add Product:**

****

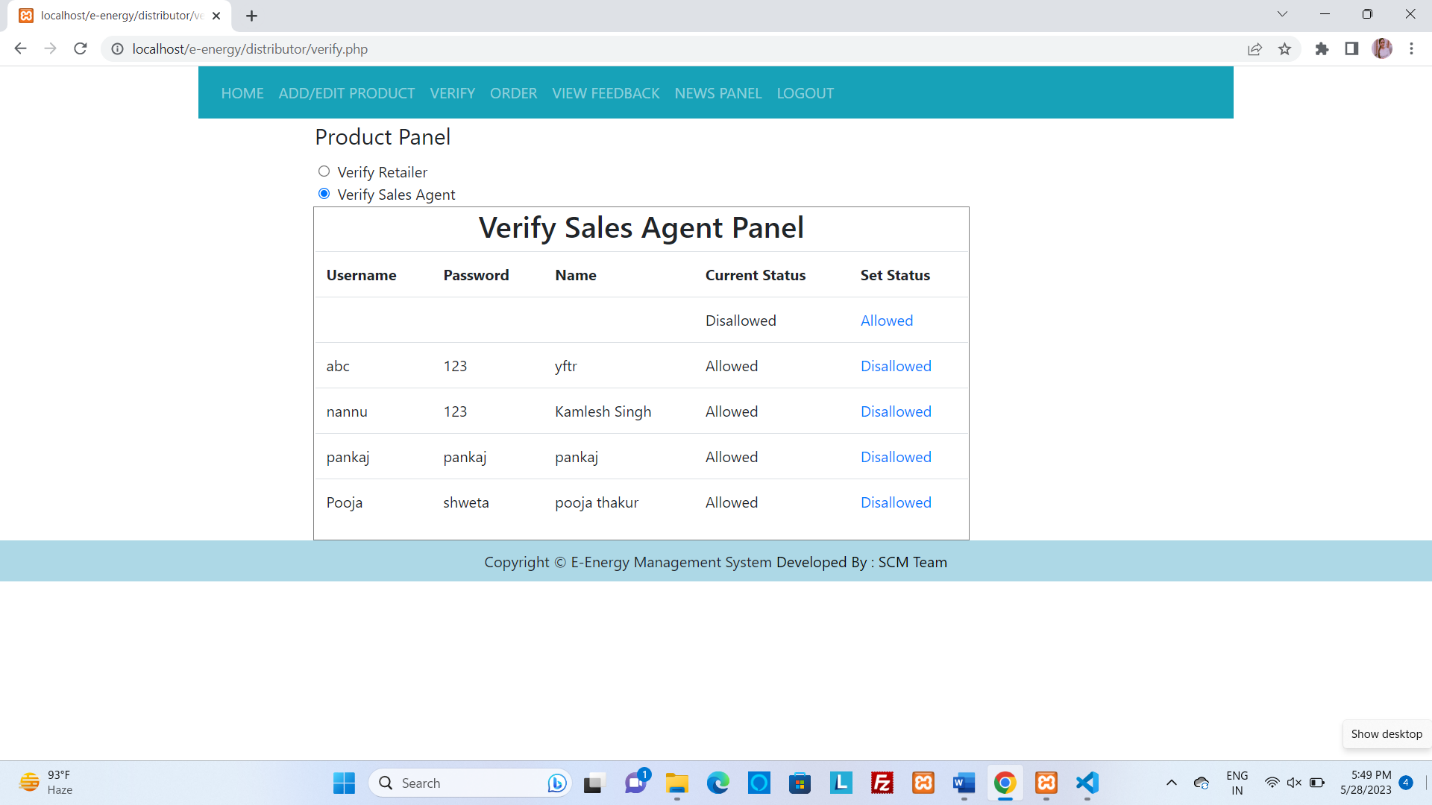
**INTERFACE 5: Product List:**

****

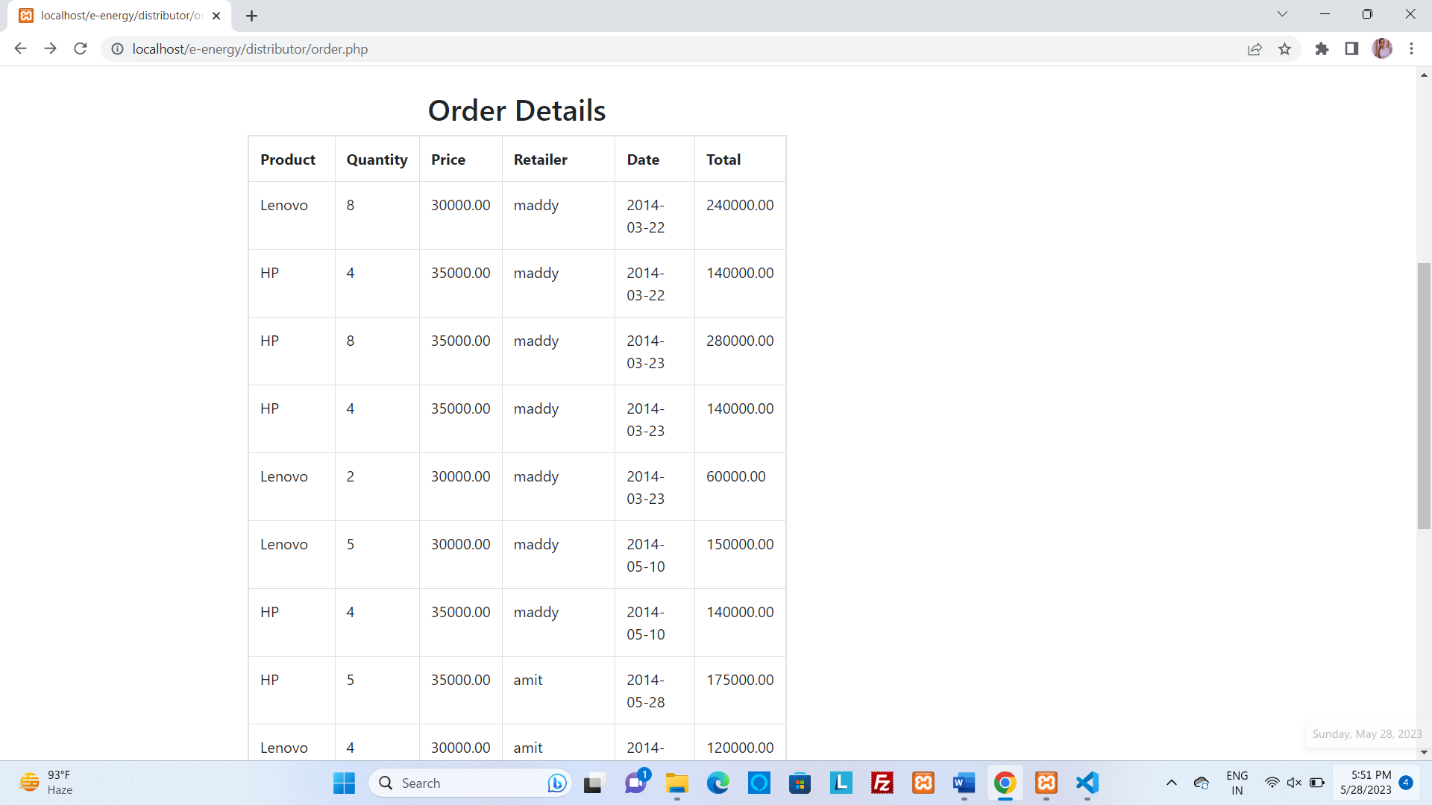
**INTERFACE 6: Verify Retailer Panel:**

****

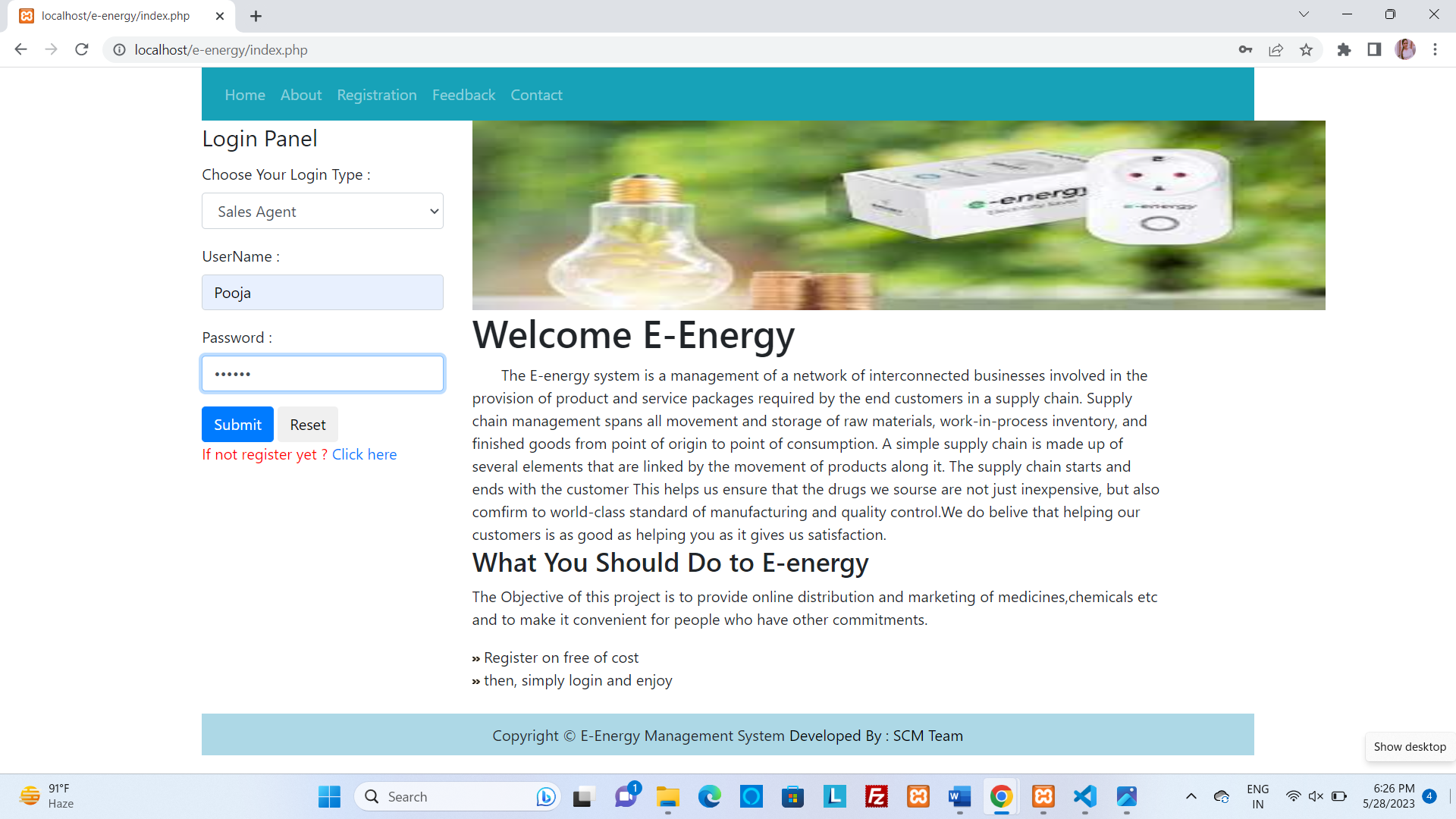
**INTERFACE 7: Verify Sales Panel:**

****

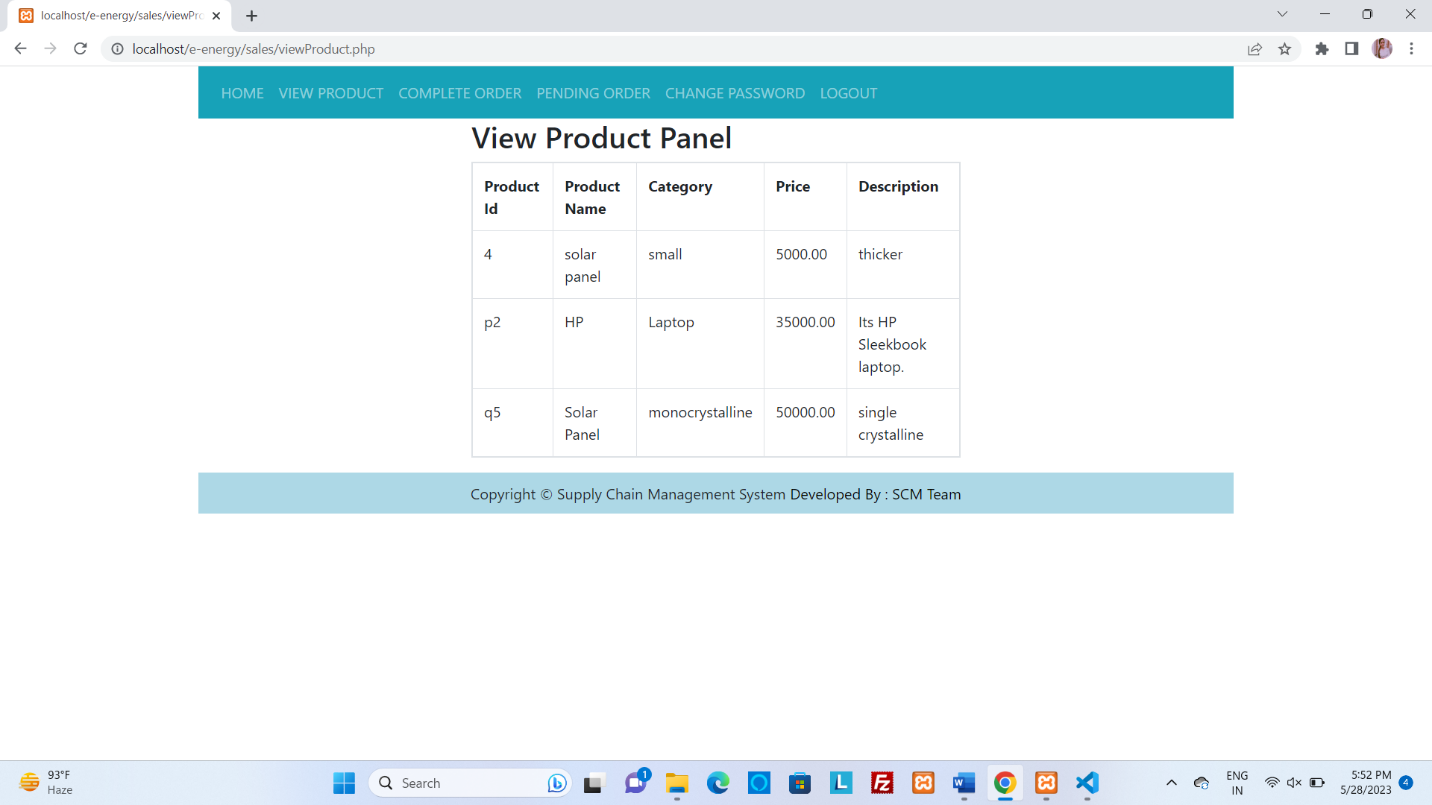
**INTERFACE 8: Order Detail:**

****

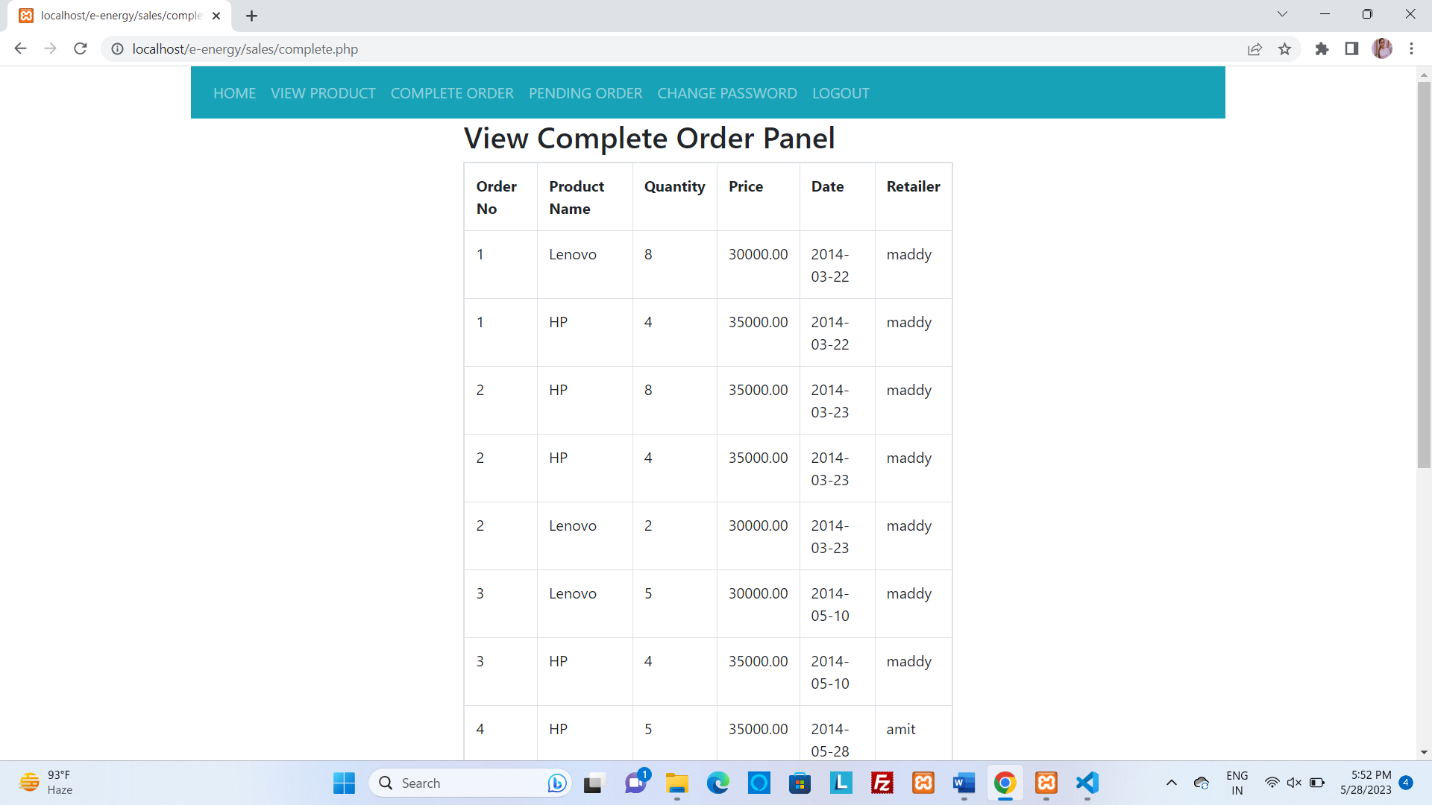
**INTERFACE 9: Salest Agent Login Panel:**

****

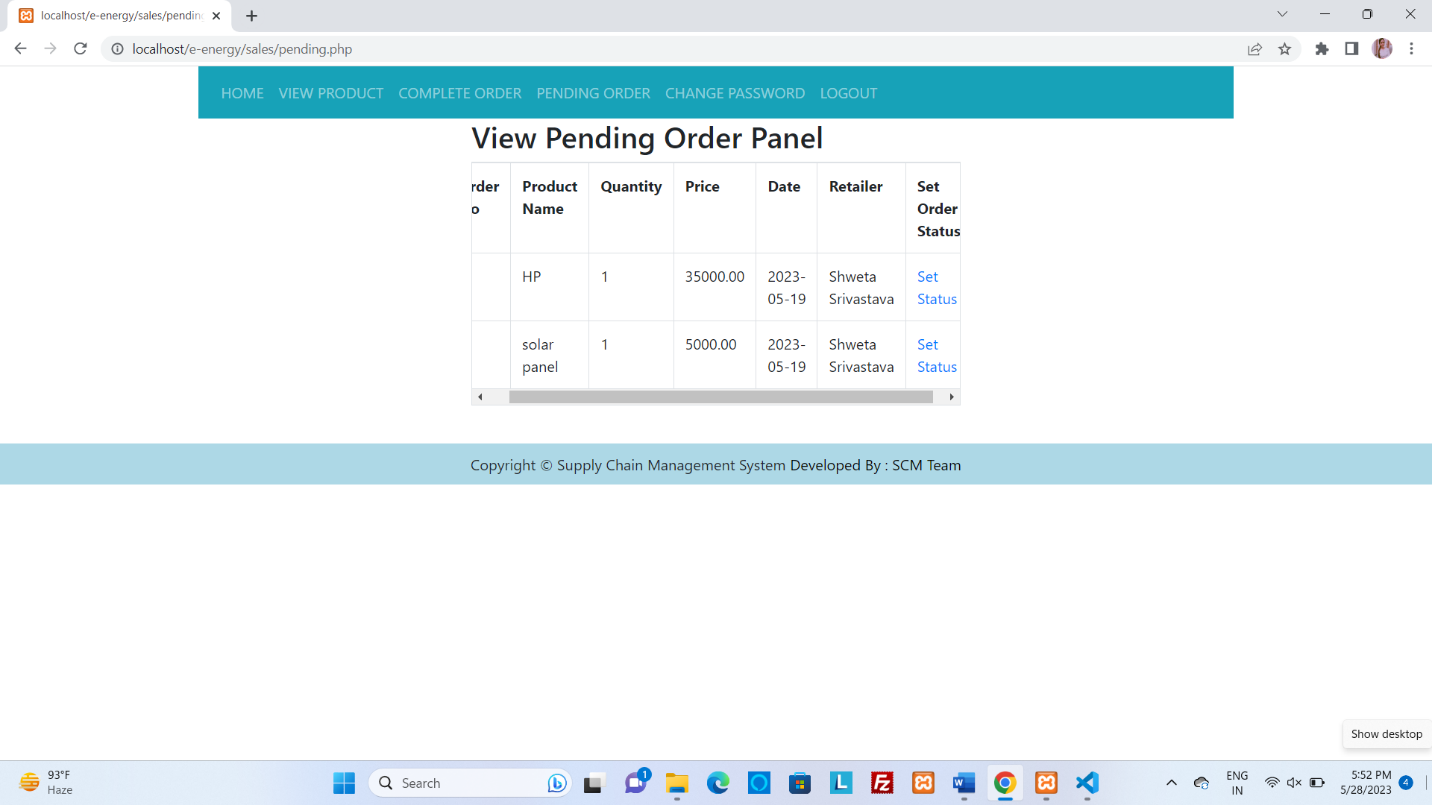
**INTERFACE 10: Product Panel:**

****

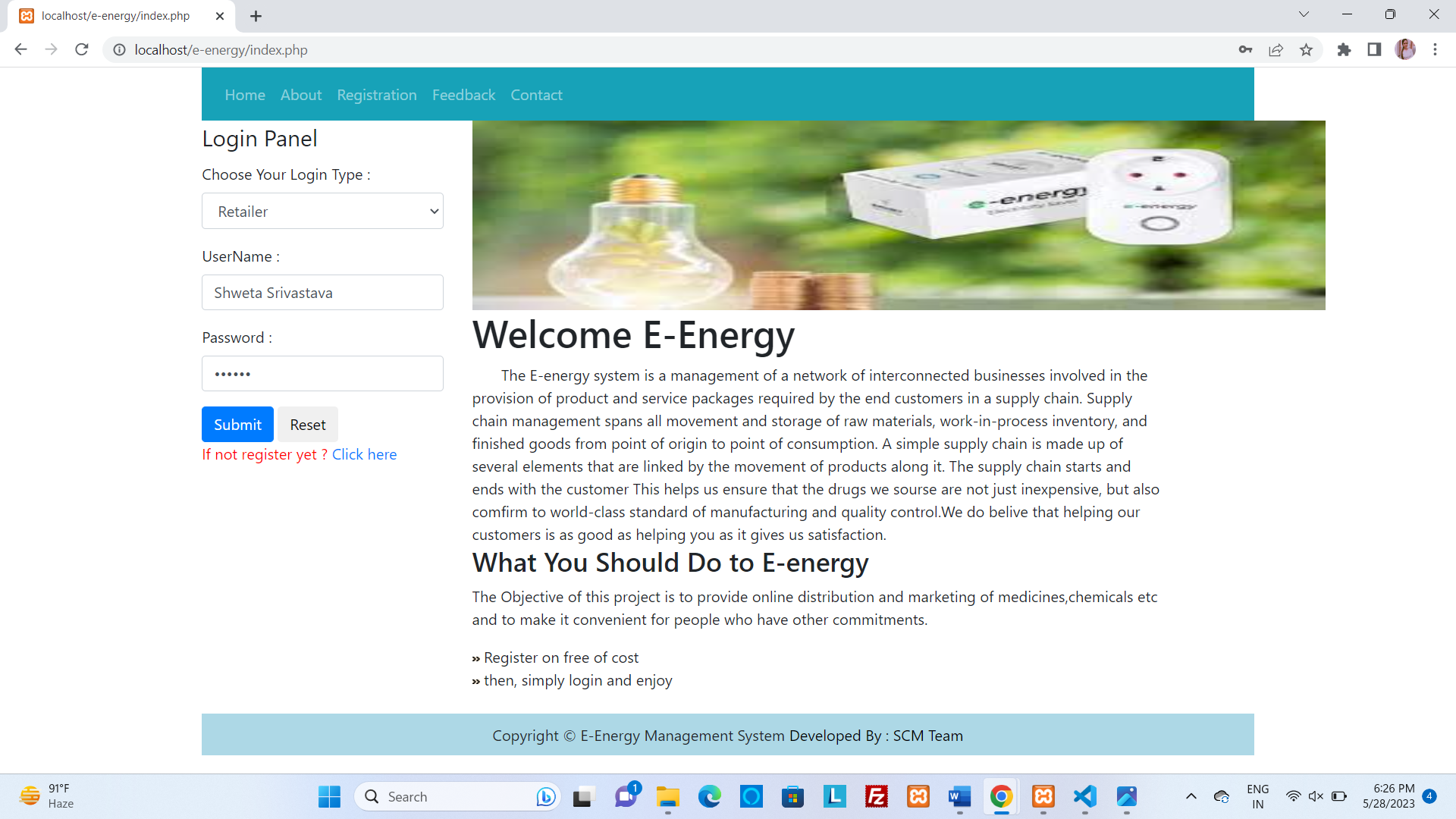
**INTERFACE 11: Complete Order Panel:**

****

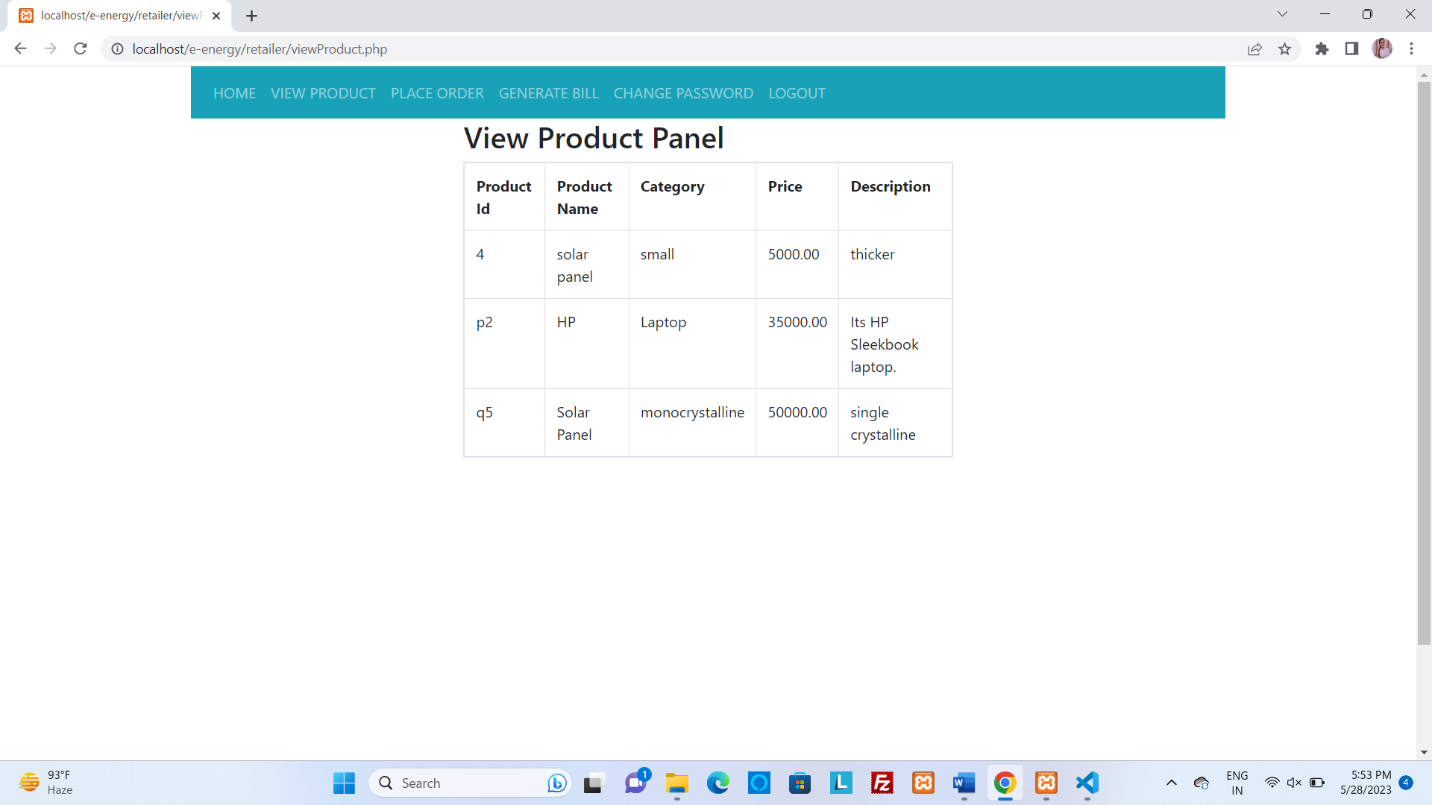
**INTERFACE 12: Pending Order Panel:**

****

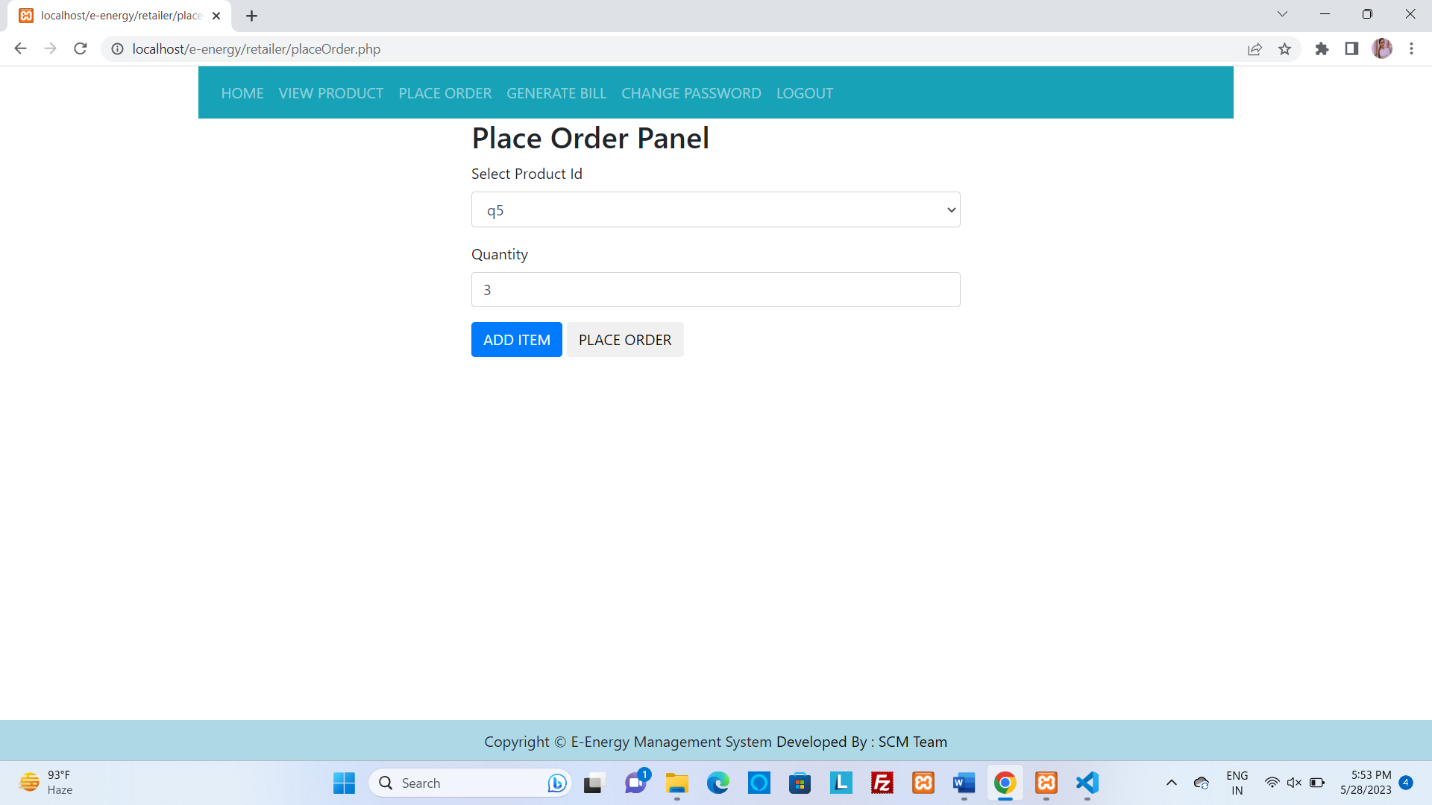
**INTERFACE 13: Retailer Login Panel:**

****

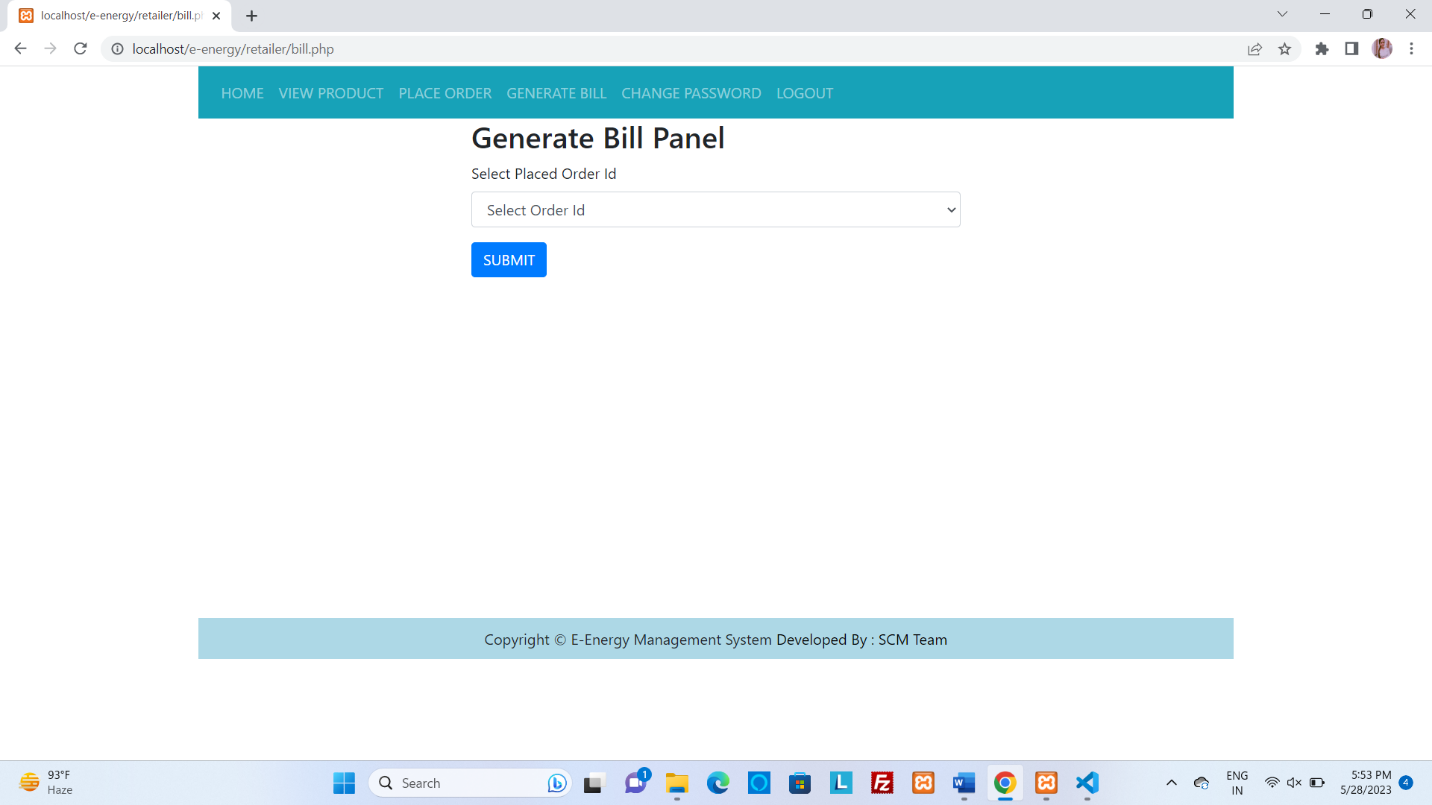
**INTERFACE 14: View Product:**

****

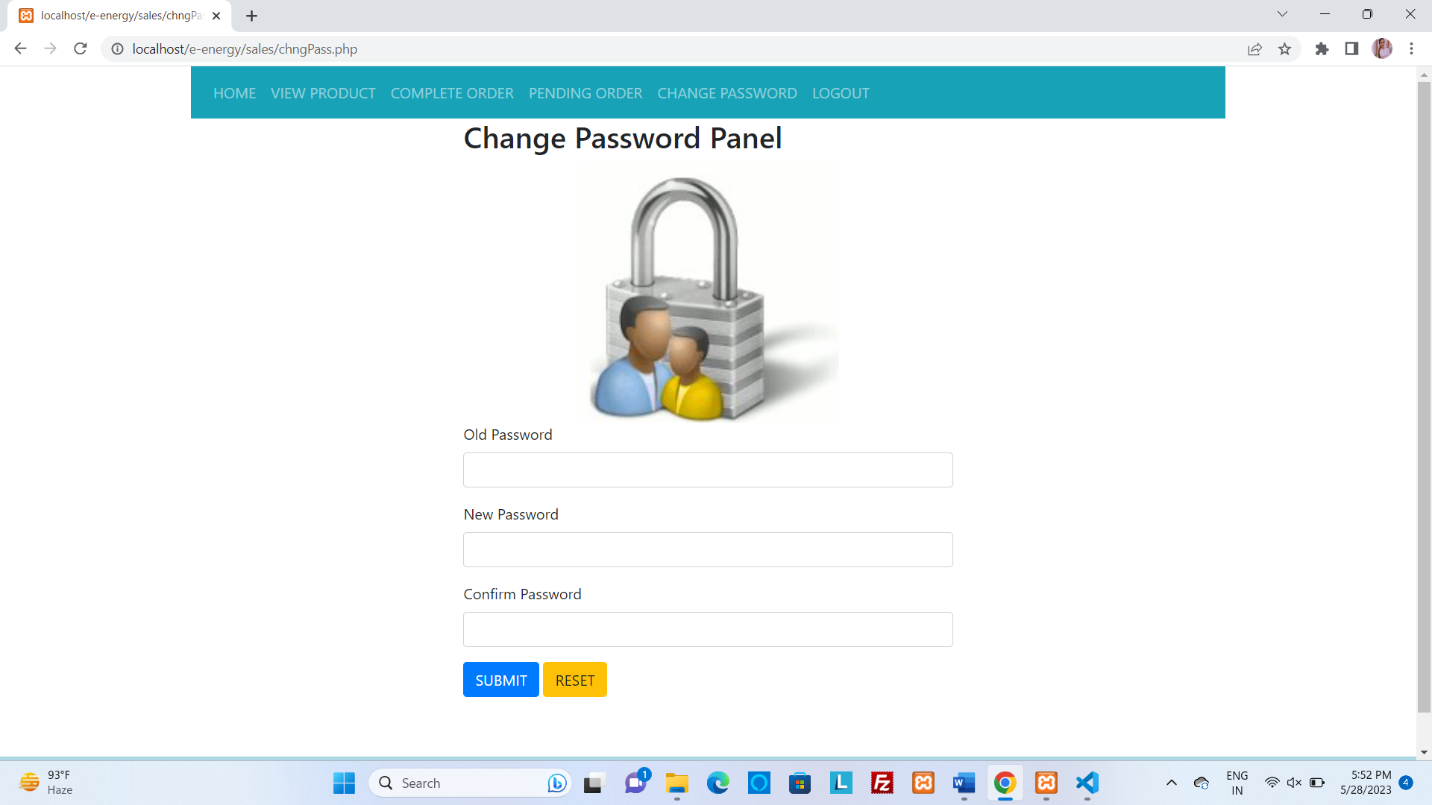
**INTERFACE 15: Place Order Panel:**

****

**INTERFACE 16: Generate Bill:**

****

**INTERFACE 17: Change Password:**

****

**DATABASE SCHEMA**

# 

# 12.DATABASE SCHEMA

**TABLE 1: REGISTRATION:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Length** |
| **UId** | VARCHAR | 15 |
| **Name** | VARCHAR | 20 |
| **Email-id** | VARCHAR | 20 |
| **Contact** | INT | 11 |
| **Address** | VARCHAR | 100 |
| **Gender** | VARCHAR | 10 |
| **Pin Code** | INT | 6 |
| **Password** | VARCHAR | 20 |
| **UserType** | CHAR | 20 |

**TABLE 2: LOGIN:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Length** |
| **Email-id** | VARCHAR | 20 |
| **Name** | VARCHAR | 20 |
| **Password** | VARCHAR | 20 |
| **Usertype** | VARCHAR | 10 |

|  |  |  |
| --- | --- | --- |
| Field Name | **Data Type** | **Length** |
| UId | VARCHAR | 15 |
| Name | VARCHAR | 20 |
| Email-id | VARCHAR | 20 |
| Contact | INT | 11 |
| Address | VARCHAR | 100 |
| Gender | VARCHAR | 10 |
| Pin Code | INT | 6 |
| Password | VARCHAR | 20 |

**TABLE 3: DISTRIBUTOR:**

**TABLE 4: SALES AGENT:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Length** |
| **UId** | **VARCHAR** | **15** |
| **Name** | **VARCHAR** | **20** |
| **Email-id** | **VARCHAR** | **20** |
| **Contact** | **INT** | **11** |
| **Address** | **VARCHAR** | **100** |
| **Gender** | **VARCHAR** | **10** |
| **Pin Code** | **INT** | **6** |
| **Password** | **VARCHAR** | **20** |

**TABLE 5: RETAILER:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Length** |
| **UId** | VARCHAR | 15 |
| **Name** | VARCHAR | 20 |
| **Email-id** | VARCHAR | 20 |
| **Contact** | INT | 11 |
| **Address** | VARCHAR | 100 |
| **Gender** | VARCHAR | 10 |
| **Pin Code** | INT | 6 |
| **Password** | VARCHAR | 20 |

**TABLE 6: PRODUCT DETAILS:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Length** |
| **Product\_id** | VARCHAR | 50 |
| **Product \_name** | VARCHAR | 50 |
| **Product\_category** | VARCHAR | 50 |
| **Product\_price** | INT | 10 |
| **Product\_color** | CHAR | 10 |

**TABLE 7: PAYEMENT DETAILS:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Length** |
| **Payment\_id** | VARCHAR | 20 |
| **Payment\_day** | VARCHAR | 10 |
| **Payment\_date** | DATE | 10 |
| **Payment\_price** | INT | 10 |
| **Payment\_type** | VARCHAR | 50 |

**TABLE 8: STOCK TABLE:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key constraint** |
| **Stock\_id** | VARCHAR | 20 |
| **Product\_id** | VARCHAR | 20 |
| **Product\_available** | INT(10) | 20 |
| **Product\_required** | INT(10) | 20 |
| **Reorder\_level** | INT | 20 |

**TABLE 9: ORDER TABLE:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key constraint** |
| **Order\_no** | VARCHAR | 20 |
| **Order\_date** | DATE | 15 |
| **Product\_id** | VARCHAR | 20 |
| **Salesagent\_id** | VARCHAR | 20 |
| **Location** | CHAR | 30 |

**TABLE 10: FEEDBACK TABLE:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key constraint** |
| **Feedback\_id** | VARCHAR | 20 |
| **Retailer\_id** | VARCHAR | 20 |
| **Distributor\_id** | VARCHAR | 20 |
| **Feedback** | CHAR | 100 |

**TABLE 11: INVOICE TABLE:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key constraint** |
| **Receipt\_id** | VARCHAR | 20 |
| **Amount** | INT | 20 |
| **Date** | DATE | 10 |
| **Salesagent\_id** | VARCHAR | 20 |
| **Retailer\_id** | VARCHAR | 20 |

**SYSTEM SECURITY**

# 13.SYSTEM SECURITY

The following securities can be made available in the system. These security measures must be highly strong to limit unauthorised access, shield software & data from virus attacks, and guard against file damage. We can group it into the following types:

• **Client Server Security:** Makes use of a variety of authorization techniques to ensure that only authorised people and software can access information resources like databases. Access control mechanisms can be configured to ensure that only users who have been properly authenticated are given access to the resources they are authorised to use. Password security features are part of this method.

**• Data Transaction Security:** This protects user authentication in networks as well as the privacy and confidentiality of electronic messages and data packets.

• **Database Security:** By granting such privileges, this technique restricts access to only the authorised user. There will be some restrictions that are strictly enforced at the common or table level. Numerous validation tests were performed to ensure that the data was accurate when it was input.

• **Password security:** In the majority of businesses or computer systems, entering the right password is the sole way to authorise access to data. However, this is merely the first step in a longer process that might include:

1. Identification
2. Authentication
3. Authorization

• Every user had access to what they could.

• Any user who uses this software and makes any changes to the database will have their entries [id, name, price, category] maintained in log files managed by sql transaction log files, which can be tracked at any time by the DBA of the phpMyAdmin.

In the event of data loss, log files for information transactions are kept and can be utilised.

**SYSTEM TESTING**

**14.System Testing**

A crucial phase of every system development life cycle is system testing. The process of testing involves running a programmer with the goal of identifying errors. It is impossible to overstate the value of software testing and its implications for software quality. Software testing serves as the final assessment of the specification, design, and coding and is a crucial component of software quality assurance. A excellent test case is one that has a decent chance of spotting an error that hasn't been found yet.

A collection of tasks that can be planned ahead of time and carried out methodically is testing. The various test circumstances should be carefully examined, and any faults found should be addressed.

#### UNIT TESTING:

The programmer that makes up the system is tested by the analyst during unit testing. The modules and processes that are put together and integrated to carry out a certain function make up the software components in a system. A huge system requires a lot of modules at various levels.

By starting with the smallest and lowest level modules and working your way up, unit testing can be done. A little programmer executes each module in a bottom-up test and delivers the necessary data.

#### INTEGRATION TESTING:

The goal of the software testing technique known as "integration testing" is to examine how various system modules or components interact with one another. It seeks to confirm that the integrated parts perform properly as a whole and cooperate as planned.

During integration testing, previously tested discrete components are merged and put to the test as a whole. The goal is to find any flaws or problems that might develop as a result of how these components work together.

#### VALIDATION TESTING:

User acceptability testing (UAT), commonly referred to as validation testing, is a method of software testing that focuses on determining if a system or software application satisfies the stated objectives and caters to the demands of the users or stakeholders. Validation testing's primary goal is to ascertain whether the system is suitable for its intended use and whether it lives up to user expectations.

Since they have the necessary domain knowledge and are familiar with the business requirements, stakeholders or end users of the system are often the ones who carry out validation testing. The goal is to confirm that the system is functionally adequate, meets user needs, and operates as intended.

#### WHITE BOX TESTING:

Software testing methodology known as white box testing, often referred to as clear box testing or structural testing, focuses on analysing a system's internal structure, design, and implementation specifics. White box testing, as contrast to black box testing, calls for knowledge of the system's fundamental logic, algorithms, and source code.

White box testing's primary goal is to assess the system's internal parts, such as distinct functions, modules, or classes, and guarantee their accuracy, reliability, and effectiveness. It seeks to check the code's integrity, spot any potential flaws, and make sure that all of the pathways, branches, and conditions are tested.

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that are Guarantee that all independent paths with in a module have been exercised at least once.

#### BLACK BOX TESTING:

Black box testing is a software testing technique that concentrates on assessing a system's functionality without taking into account its internal organisation or implementation specifics. In "black box" testing, the system is treated as a "black box," and the tester tests it based on the inputs and anticipated outputs without being aware of the system's internal workings.

Black box testing's primary goal is to evaluate the system's usability and behaviour from the viewpoint of the end user. The system's specifications and requirements are the only ones that testers have access to while creating and carrying out test cases. They are not given access to the source code or internal design specifications.

#### CONTENT TESTING:

The practise of assessing the correctness, completeness, and quality of the material on a website, application, or digital platform is referred to as content testing. It emphasises ensuring that the material achieves the desired goals, is free of errors, and offers a satisfying user experience.

Examining and confirming the material's many components, including text, photographs, videos, links, documents, and multimedia features, is known as content testing. Finding and fixing any problems with content display, consistency, relevance, and functionality are the main objectives.

Content testing attempts to uncover errors in content of the desktop application. In addition to examining static content for errors, this testing step also considers dynamic content derived from data maintained as a part of database system that has been integrated with the application.

Content testing of all modules is evaluated for syntactic and semantic errors.

At syntactic level I have verified the content for spelling, punctuation and any grammar mistakes of all interfaces which contain the content of the software.

#### DATABASE TESTING:

Database testing is a type of software testing that focuses on assessing the reliability, accuracy, and integrity of a database and the other components that are connected to it. It comprises checking the accuracy of the data kept in the database and testing the functionality, speed, and security of data manipulation and retrieval.

The quality, consistency, and dependability of the data recorded in the database are checked by database testers. To make sure the data complies with the requirements, they look for data integrity constraints such as unique key constraints, referential integrity, and data validation rules.

Data manipulation operations such as insert, update, delete, and select statements are tested as part of database testing. The accuracy of data entry, updating, and deletion is checked, as well as the retrieval of the desired information from the database using the specified queries.

#### USER INTERFACE TESTING:

User interface (UI) testing is a method of software testing that focuses on analysing the graphical user interface (GUI) of a programme or system. It attempts to make sure the user interface is aesthetically pleasing, useful, and offers a simple and intuitive user experience.

The user interface's organisation, style, and general appeal are evaluated by testers. They make sure that the design follows the stated style guide or branding rules, that the elements are correctly aligned, that the fonts and colours are consistent, etc.

The navigation flow and user interactions inside the interface are validated as part of UI testing. Button, link, menu, and other interactive element functionality is tested to ensure that users are being directed to the relevant destinations or actions.

#### INTERFACE MECHANISM TESTING:

Testing of interface mechanisms involves examining how various software systems or components interact with one another. It focuses on ensuring that the protocols, interfaces, and data transfers between various components operate properly and according to plan.

Through testing, it is made sure that inputs given to a system or component through its interface are correctly processed and result in the desired outputs. Testers make sure the interface takes legitimate inputs, responds appropriately to illegitimate inputs, and produces the desired results.

Testing interface mechanisms also involves analysing the protocols and data exchange techniques applied between systems or components. Testing ensures that communication protocols (such as HTTP, TCP/IP, SOAP, REST, etc.) are appropriately implemented and that data is accurately transferred and received. This testing is done in following areas.

**BUTTONS:**

To check whether the proper module is linked or not, each button is tested. To check whether each module is connecting to the right button or not, I have listed them all in the form.

**FORMS:**

At two separate levels, a more general level and a more focused one, testing forms have been conducted. I have examined, at the very least, whether fields' labels have been accurately defined.

• Whether the server is receiving all the data on the form and that there were no data losses during the client-to-server communication.

• Whether suitable default values are available if the user leaves the selection box empty.

I've conducted tests at a more focused level for:

• Whether text entry fields are the right width.

• Whether text fields permit strings to be longer than the allowed length.

• Whether or not the tab order for the various controls is in the correct order.

**USABILITY TEST:**

I have confirmed through this testing that users can properly interact with the system up to the degree that I tested. Tests are created to identify how user-friendly a desktop application's interface is for its users. I created the test case so that several degrees of usability testing could be verified:

• Usability tests have been run on every single interface, including forms.

• A client-side function-related usability test was conducted on all frames.

• A usability test of the entire desktop programme has been conducted.

**COMPATIBILITY TEST:**

This desktop application should function in a variety of settings, including those with various computer architectures, operating systems, and network connection rates.

Operating system variation may result in problems with the processing of desktop applications, much as varied computing configurations can lead to differences in client-side scripting speeds and display resolution.

As we anticipated, different machines will yield slightly different results. In some circumstances, these results won't cause any issues, but in other cases, there will be significant mistakes.

We have compiled a list of all client-side functions that have compatibility issues before beginning these testing strategies. In essence, those that we tested can be summarised by naming various computing platforms, typical display devices, operating systems supported by the platform, and the browsers that I have access to.

**NAVIGATION TESTING:**

In order to find any navigation issues, the navigation syntax and semantics are checked for navigability. The purpose of navigation testing is to confirm that the navigational mechanisms are reliable and that the relevant user category may successfully complete each Navigation Semantic Unit.

In the sections below, we have tested navigation.

• Navigation frames have undergone rigorous testing.

• Correct checks are made for redirects.

• Is a navigation button's target page accurate or not?

• Has the button's caption any real meaning?

**DEPLOYMET:**

The error-free project, which passed all the tests, is now deployed at the client environment in this phase.

**SYSTEM IMPLEMENTATION:**

The process of implementation entails training users on how to install new applications, providing new equipment into use, and creating any data files required to use them. Implementation comes in three different forms.

• The installation of a computer system to take the place of a manual one. Covering files, educating users, making accurate files, and checking the integrity of printouts are issues that arise.

• The installation of a new computer system to swap out an older one. This conversion is typically challenging. If well planned, there could be a lot of issues. Therefore, converting huge computer systems could take up to a year.

**IMPLEMENTATION APPROACH**

Project implementation is based on the developers' abilities and the accessibility of the necessary tools and technology. Any project can be implemented more easily if the developer is familiar with the tools and technologies; nevertheless, if the developer is not, project implementation will be more difficult.

The project anticipates that since we selected NetBeans IDE 8.2, HTML, CSS, Bootstrap, and JavaScript for the front end and PHP, MySQL, and phpMyAdmin for the back end, we should be familiar with all of the tools. In our fifth semester, I took a PHP course. I didn't know much about databases before I started this project, but I worked really hard to understand DBMS and MySQL.

I began using these tools to implement our project after learning all the necessary ones. First, I used NetBeans 8.2 to create the full user interface. Having designed interfaces for usage, For our project, I started building a database and tables. I build a database called dbms2 and a table called home in our database. I established the limitations and connections for these tables.

Then I began writing code for our project. Because our thoughts are transformed into codes during this step, the coding phase is an essential component of any web development process. Hard work and patience are needed while coding since coding cannot be implemented if patience is not maintained.

System modules all function properly. I've done my best to provide each button in the application dynamic features. Each module is user-friendly and has an intuitive design. Here is some of the core application code for our desktop programmer.

**CODING DETAILS**

# 15.CODING DETAILS

**CODE EFFICIENCY:**

It is well known that a solid coding style can make up for many of a simple programming language's shortcomings whereas a poor style can undermine the purpose of a powerful language. The objective of good coding is to provide clear, elegant code that is simple to understand. Use a few common control constants is one of the coding best practises.

• The disciplined use of unconditional branching.

• Adding user-defined data types to represent things in the problem area.

• Data structures being concealed behind access functions.

• Including standard documentation prologues for each compilation unit and subprogram.

• To improve readability, use indentation, parenthesis, blank spaces, blank lines, borders, and comments.

**CONCLUSION**

# 16.CONCLUSION

The goal of this project was to create a programmer for recording all Supply Order information. The system created can satisfy all fundamental needs. The user will have the ability to keep track of all the supplied equipment thanks to this function. The suggested solution will also assist inventory management because it will automate the entire supply process, which will lighten the workload. One of the main considerations is also the system's security.

Any software may always be made better, regardless of how effective the system is already. It's crucial that the system be adaptable enough to accommodate future changes.

It's crucial that the system be adaptable enough to accommodate future changes.

To enable the system to adapt to future modifications, it has been divided into various parts. All user requirements have been addressed, and it has been designed to be user-friendly.

#### ADVANTAGES:

I made an effort to provide these facilities with the new computer system.

1. Manual conversion of a system to a computeried system.

2. A user-friendly interface.

3. Saving time.

4. Keep paper records.

5. Establishing a database connection to enable the usage of various types of queries and data reports.

**FUTURE SCOPE OF THE PROJECT**

**17.FUTURE SCOPE OF THE PROJECT**

• The manufacturer and supplier modules, which can produce the finished product and supply the raw materials, respectively, may also be included.

• the project's payment gateway addition. the project's payment gateway addition.

• Online retailers typically operate around-the-clock, and many customers have access to the Internet both at home and at work.

• The E- Energy was created using PHP and MySql, and it perfectly satisfies the goals of the system for which it was created.

• The business world may use this.

• All users involved in the system are aware of its benefits, and the system is run at a high degree of efficiency.

• As stated in the requirement specification, the system resolves the issue.

**BIBLIOGRAPHY**

# 18.BIBLIOGRAPHY

**References:**

**BOOKS:**

* Software Engineering by Dr.H. S. Behera, Asst. Prof. K. K. Sahu, Asst. Prof. Gargi Bhattacharjee.
* Rajeev Mall, Fundamental of Software Engineering, Published by PHI,4th Edition, 2014
* Rasmus Lerdorf and Levin Tatroe, "Programming PHP", 2nd Edition, O'Reilly
* Ivan Bayross, “HTML, DHTML, Java Script, Perl & CGI” , 2nd Edition, BPB Publication.
* Money Management for Indians by Partha P. Debnath, Published By Notion Press, 1st Edition, 2021

**WEBSITES:**

* [*https://www.geeksforgeeks.org/software-engineering-prototyping-model/https://developer.mozilla.org*](https://www.geeksforgeeks.org/software-engineering-prototyping-model/https:/developer.mozilla.org)
* [*https://www.javatpoint.com/software-engineering-software-development-life-cycle*](https://www.javatpoint.com/software-engineering-software-development-life-cycle)
* [*https://www.w3schools.com/html/html\_css.asp*](https://www.w3schools.com/html/html_css.asp)
* [*https://www.tutorialspoint.com/mysql/index.html*](https://www.tutorialspoint.com/mysql/index.html)