

BILLING APP FOR INVENTORY

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science degree in Computer Science by

UDHAYANIDHI.V (REG.NO.39290115)

SRINIVASAN.A (REG.NO.39290103)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **UDHAYANIDHI.V(REG.NO.3929010115 and SRINIVASAN.A(REG.NO.39290103)** who carried out the project entitled "**BILLING APP FOR INVENTORY**" under my supervision from NOVEMBER 2021 to MAY 2022.

InternalGuide

DR.ASHA

Head of the Department

DR.L.LAKSHAMANAN.,M.E.,Ph.D,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **UDHAYANIDHI.V (REG.NO.39290115) and SRINIVASAN.A (REG.NO.39290103)** hereby declare that the Project Report entitled_“**BILLING APP FOR INVENTORY**” done by me under the guidance of **Dr.ASHA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer science.

DATE :

PLACE : CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.ASHA** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT OF THE WORK

The proposal put forward a unique solution for the stock management in the inventory. The application allows the user to be on up to date of the products remaining in the inventory.

Maintaining a balance between too much and too little stocks for the future use in the shop. This application also has tracking feature as the goods are transported between many locations.

To maintain a stock for each and every product they are dealing with. Paperless billing is a process that businesses can use to get paid quickly and securely without dealing with the hassles of paper bills and check payments

Paperless bills allow bill delivery and payment to take place completely online, which improves efficiency while reducing cost.

TABLE OF CONTENTS

CHAPTER NO NO	TITLE	PAGE
	ABSTRACT	
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	METHODOLOGY	
	3.1 HARDWARE REQUIRMENTS	4
	3.2 SOFTWARE REQUIREMENTS	4
	3.3 OUTLOOK OF THE PROJECT	20
	3.4 OVERVIEW	21
	3.4.1 BILL GENERATOR	21
	3.4.2 PRINT BILL	21
	3.4.3.VIEW BILL	21
	3.4.4 INVENTORY	22
	3.4.5 PRINCIPLE OF INVENTORY	22
	3.4.6 APPLICATIONS	22

	3.5 FEATURES OF BILLING SYSTEM	23
	3.5.1 PRODUCTS OR ITEMS	23
	3.5.2 PURCHASES	24
	3.5.3 CUSTOMERS	24
	3.5.4.SUPPLIERS	24
	3.5.5 REPORTS	25
	3.6 MODULES	26
	3.6.1 MODULES EXPLANATION	27
4	RESULT AND PERFORMANCE ANALYSIS	
	4.1 LOGIN PAGE	28
	4.2 SUPPLIER DETAILS PAGE	29
	4.3 CUSTOMER DETAILS PAGE	30
	4.4 COMPANY DETAILS	31
	4.5 USER DETAILS	32
	4.6 STAFF DETAILS	32
	4.7 SUPPLIER REPORT	33
	4.8 STAFF REPORT	33
	4.9 SALES DETAILS	34
	4.10 ORDER TOTAL REPORT	34
	4.11 ORDER LIST REPORT	35
	4.12 EDIT ORDER REPORT	35
	4.13 CUSTOMER REPORT	36
	4.14 COMPANY REPORT	36

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION	38
5.2 FUTURE WORK	38
REFERENCES	39
APPENDIX	40
A.SOURCE CODE	40

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Login Page	28
4.2	Supplier Details Page	29
4.3	Customer Details Page	30
4.4	Company Details	31
4.5	User Details	32
4.6	Staff Details	32
4.7	Supplier report	33
4.8	Staff report	33
4.9	sales details	34
4.10	order total report	34
4.11	order list report	35
4.12	Edit order report	35
4.13	Customer report	36
4.14	Company report	36

CHAPTER 1

INTRODUCTION

Billing software refers to programs that handle the tracking of billable products and services delivered to a customer or set of customers. A good inventory tracking system will tell you what merchandise in stock, what is on order, when it will arrive and what you've sold. With such a system, you can plan purchases intelligently and quickly recognize the fast-moving items you need to reorder and the slow-moving items you should mark down or specially promote. It is also used for to reduce overhead cost and Time saving application. It will give's an Maximum profit within less cost, update Log will be maintained. A bill is a record of statement, and transaction of the money. Bill is used to keep record of business when a person buys some goods or any valuable things. To maintain smooth usage of the bill from creation of the bill, analyzing it and the printing the bill to manage all the process we need a billing system. Billing system was started in the medieval period when people started buying things. The shopkeeper used to keep track of the all the products sold to the customer and how much products was there in their reserve. They need to write down all the records in a record book manually. . The bill is not just for keeping the record of transaction, the bill is used for so many other purposes. They use the data from the bills to control the extra expenditure and spend on what is needed and also use it to improve their businesses. computers to generate and print the bill.

CHAPTER 2

LITERATURE SURVEY

1. Automated Shopping and Billing System Using Radio-Frequency Identification (<https://ieeexplore.ieee.org/document/8776944>).

Authors: Raghav Chadha, Srishti Kakkar, Garima Aggarwal

Automated Shopping and Billing Trolley system that utilizes Radio Frequency Identification (RFID) with the billing side using Mobile Application. RFID technology offers the capacity to give numerous new administrations and comforts in the retail condition. It utilizes radio frequency to accomplish this. The framework comprises of a Radiofrequency Identification based trolley that communicates with the billing system that is the Mobile Application. The RFID reader examines every one of the product when product is placed in the trolley.

2. SMART PAYMENT AND BILLING MANAGEMENT SYSTEM (<https://ieeexplore.ieee.org/document/936964>).

Authors : Susila Sakthy.S, Ragupathy.T, Kishore.G

This paper proposes a concurrent equalizer, in which a Least Mean Square (LMS) equalizer operates cooperatively with a ChfA equalizer, controlled through a non-linear link that depends on the system a priory state. Simulation results using M-QAM signaling have shown that the Concurrent Equalizer presents a much lower sensitivity to local minima than the CMA approa

3. Vulnerabilities and Security of Web Applications

(<https://ieeexplore.ieee.org/document/8777558>).

Author: Divyani Yadav, Deeksha Gupta, Dhananjay Singh, Devendra Kumar, Upasana Sharma.

Web applications are active websites which are composition of server based programs serving user interaction and various other functionalities. Web Server security is thus an important aspect for any organisation having web server connectivity with the internet and also to ensure customers using their websites, for a secure online portal. In this age of digital revolution, there has been a rise in demand of web developers who can produce user friendly web platforms such as mobile applications, web applications. The user base for online web applications is on a rise too. We have seen a huge emphasis on creating visual and catchy web applications but with large amount of sensitive user data at stake there should be more focus on providing web security to the applications developed.

CHAPTER 3

METHODOLOGY

3.1 HARDWARE REQUIREMENTS

Processor: Intel i3 (minimum), i5 (recommended).

RAM: 2 GB

Hard Disk Space: 2 GB

3.2 SOFTWARE REQUIREMENTS

C# (FRONT END) :

C# is a programming language that translates an abstract idea into a program design we can see on screens. C# presents a three step approach for creating programs which are:

- a) To design the appearance of the application.
- b) Assign property settings to the objects of your program.
- c) Write the code to direct specific tasks at runtime.

DESIGN GOALS :

- The language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.
- The language is intended for use in developing software components suitable for deployment in distributed environments.
- Portability is very important for source code and programmers, especially those already familiar with C and C++.
- Support for internationalization is very important.
- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.
- Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language.

DISTINGUISHING FEATURES :

Some notable features of C# that distinguish it from C, C++, and Java where noted, are:

Portability

By design, C# is the programming language that most directly reflects the underlying Common Language Infrastructure (CLI). Most of its intrinsic types correspond to value-types implemented by the CLI framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate Common Intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or Fortran.

Typing

C# supports strongly, implicitly typed variable declarations with the keyword `var`, and implicitly typed arrays with the keyword `new[]` followed by a collection initializer.

C# supports a strict Boolean data type, `bool`. Statements that take conditions, such as `while` and `if`, require an expression of a type that implements the `true` operator, such as the Boolean type. While C++ also has a Boolean type, it can be freely converted to and from integers, and expressions such as `if (a)` require only that `a` is convertible to `bool`, allowing `a` to be an `int`, or a pointer. C# disallows this "integer meaning true or false" approach, on the grounds that forcing programmers to use expressions that return exactly `bool` can prevent certain types of programming mistakes such as `if (a = b)` (use of assignment `=` instead of equality `==`).

C# is more type safe than C++. The only implicit conversions by default are those that are considered safe, such as widening of integers. This is enforced at compile-time, during JIT, and, in some cases, at runtime. No implicit conversions occur between Booleans and integers, nor between enumeration members and integers (except for literal `0`, which can be implicitly converted to any enumerated type). Any user-defined conversion must be explicitly marked as `explicit` or `implicit`, unlike C++ copy constructors and conversion operators, which are both implicit by default.

C# has explicit support for covariance and contravariance in generic types, unlike C++ which has some degree of support for contravariance simply through the semantics of return types on virtual methods.

Enumeration members are placed in their own scope.

The C# language does not allow for global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variables and functions.

Local variables cannot shadow variables of the enclosing block, unlike C and C++.

Metaprogramming

Metaprogramming can be achieved in several ways:

Reflection is supported through .NET APIs, which enable scenarios such as type metadata inspection and dynamic method invocation.

Expression tree represent code as an abstract syntax tree, where each node is an expression that can be inspected or executed. This enables dynamic modification of executable code at runtime. Expression tree introduce some homoiconicity to the language.

Attributes are metadata that can be attached to types, members, or entire assemblies, equivalent to annotations in Java. Attributes are accessible both to the compiler and to code through reflection. Many of these attributes duplicate the functionality of GCC's and VisualC++'s platform-dependent preprocessor directives.[citation needed]

The .NET Compiler Platform (Roslyn) provides API access to language compilation services, allowing for the compilation of C# code from within .NET applications. It

exposes APIs for syntactic (lexical) analysis of code, semantic analysis, dynamic compilation to CIL, and code emission.

Source generators, a feature of the Roslyn C# compiler, enable compile time metaprogramming. During the compilation process, developers can inspect the code being compiled with the compiler's API and pass additional generated C# source code to be compiled.

Methods and functions

A method in C# is a member of a class that can be invoked as a function (a sequence of instructions), rather than the mere value-holding capability of a class property. As in other syntactically similar languages, such as C++ and ANSI C, the signature of a method is a declaration comprising in order: any optional accessibility keywords (such as `private`), the explicit specification of its return type (such as `int`, or the keyword `void` if no value is returned), the name of the method, and finally, a parenthesized sequence of comma-separated parameter specifications, each consisting of a parameter's type, its formal name and optionally, a default value to be used whenever none is provided. Certain specific kinds of methods, such as those that simply get or set a class property by return value or assignment, do not require a full signature, but in the general case, the definition of a class includes the full signature declaration of its methods.

Like C++, and unlike Java, C# programmers must use the scope modifier keyword `virtual` to allow methods to be overridden by subclasses.

Extension methods in C# allow programmers to use static methods as if they were methods from a class's method table, allowing programmers to add methods to an object that they feel should exist on that object and its derivatives.

The type dynamic allows for run-time method binding, allowing for JavaScript-like method calls and run-time object composition.

C# has support for strongly-typed function pointers via the keyword delegate. Like the Qt framework's pseudo-C++ signal and slot, C# has semantics specifically surrounding publish-subscribe style events, though C# uses delegates to do so.

C# offers Java-like synchronized method calls, via the attribute `[MethodImpl(MethodImplOptions.Synchronized)]`, and has support for mutually-exclusive locks via the keyword `lock`.

Property

C# supports classes with properties. The properties can be simple accessor functions with a backing field, or implement getter and setter functions.

Since C# 3.0 the syntactic sugar of auto-implemented properties is available, where the accessor (getter) and mutator (setter) encapsulate operations on a single attribute of a class.

Namespace

A C# namespace provides the same level of code isolation as a Java package or a C++ namespace, with very similar rules and features to a package. Namespaces can be imported with the "using" syntax.

Memory access

In C#, memory address pointers can only be used within blocks specifically marked as unsafe,[68] and programs with unsafe code need appropriate permissions to run. Most object access is done through safe object references, which always either point to a "live" object or have the well-defined null value; it is impossible to obtain a reference to a "dead" object (one that has been garbage collected), or to a random block of memory. An unsafe pointer can point to an instance of an 'unmanaged' value type that does not contain any references to garbage-collected objects, array, string, or a block of stack-allocated memory. Code that is not marked as unsafe can still store and manipulate pointers through the System.IntPtr type, but it cannot dereference them.

Managed memory cannot be explicitly freed; instead, it is automatically garbage collected. Garbage collection addresses the problem of memory leaks by freeing the

programmer of responsibility for releasing memory that is no longer needed in most cases. Code that retains references to objects longer than is required can still experience higher memory usage than necessary, however once the final reference to an object is released the memory is available for garbage collection.

Exception

A range of standard exceptions are available to programmers. Methods in standard libraries regularly throw system exceptions in some circumstances and the range of exceptions thrown is normally documented. Custom exception classes can be defined for classes allowing specific handling to be put in place for particular circumstances as needed.

Checked exceptions are not present in C# (in contrast to Java). This has been a conscious decision based on the issues of scalability and versionability.

Polymorphism

Unlike C++, C# does not support multiple inheritance, although a class can implement any number of "interfaces" (fully abstract classes). This was a design decision by the language's lead architect to avoid complications and to simplify architectural requirements throughout CLI.

When implementing multiple interfaces that contain a method with the same name and taking parameters of the same type in the same order (i.e. the same signature), similar to Java, C# allows both a single method to cover all interfaces and if necessary specific methods for each interface.

However, unlike Java, C# supports operator overloading.

Visual Studio .NET is a Microsoft-integrated development environment (IDE) that can be used for developing consoles, graphical user interfaces (GUIs), Windows Forms, Web services and Web applications.

Visual Studio is used to write native code and managed code supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Studio .NET's code editor supports IntelliSense and code refactoring, while the Visual Studio .NET integrated debugger supports both source and machine-level debugging. Visual Studio .NET includes other built-in tools, like a form designer, which is useful when building GUI applications; a Web designer that creates dynamic Web pages; a class designer that is used to create custom libraries, and a schema designer for database support..

There are Visual Studio versions that support individual languages, which are a cost-effective solution for beginners who only need Visual Basic, or advanced users who work on fast processing applications and continually need visual C# or visual C++, but are not interested in low-speed Visual Basic.

MICROSOFT SQL (BACKEND) :

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

Microsoft makes SQL Server available in multiple editions, with different feature sets and targeting different users. These editions are:

Mainstream editions

Enterprise

SQL Server Enterprise Edition includes both the core database engine and add-on services, with a range of tools for creating and managing a SQL Server cluster. It can manage databases as large as 524 petabytes and address 12 terabytes of memory and supports 640 logical processors (CPU cores).

Standard

SQL Server Standard edition includes the core database engine, along with the stand-alone services. It differs from Enterprise edition in that it supports fewer active instances (number of nodes in a cluster) and does not include some high-availability functions such as hot-add memory (allowing memory to be added while the server is still running), and parallel indexes.

Web

SQL Server Web Edition is a low-TCO option for Web hosting.

Business intelligence

Introduced in SQL Server 2012 and focusing on Self Service and Corporate Business Intelligence. It includes the Standard Edition capabilities and Business Intelligence tools: Power Pivot, Power View, the BI Semantic Model, Master Data Services, Data Quality Services and xVelocity in-memory analytics.

Workgroup

SQL Server Workgroup Edition includes the core database functionality but does not include the additional services. Note that this edition has been retired in SQL Server 2012.

Express

SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 10 GB database files (4 GB database files prior to SQL Server Express 2008 R2). It is intended as a replacement for MSDE. Two additional editions provide a superset of features not in the original Express Edition. The first is SQL Server Express with Tools, which includes SQL Server Management Studio Basic. SQL Server Express with Advanced Services adds full-text search capability and reporting services.

Specialized editions

Azure

Microsoft Azure SQL Database is the cloud-based version of Microsoft SQL Server, presented as a platform as a service offering on Microsoft Azure.

Azure MPP

Azure SQL Data Warehouse is the cloud-based version of Microsoft SQL Server in a MPP (massively parallel processing) architecture for analytics workloads, presented as a platform as a service offering on Microsoft Azure.

Compact (SQL CE)

The compact edition is an embedded database engine. Unlike the other editions of SQL Server, the SQL CE engine is based on SQL Mobile (initially designed for use with hand-held devices) and does not share the same binaries. Due to its small size (1 MB DLL footprint), it has a markedly reduced feature set compared to the other editions. For example, it supports a subset of the standard data types, does not support stored procedures or Views or multiple-statement batches (among other limitations). It is limited to 4 GB maximum database size and cannot be run as a Windows service, Compact Edition must be hosted by the application using it. The 3.5 version includes support for ADO.NET Synchronization Services. SQL CE does not support ODBC connectivity, unlike SQL Server proper.

Developer

SQL Server Developer Edition includes the same features as SQL Server Enterprise Edition, but is limited by the license to be only used as a development and test system, and not as production server. Starting early 2016, Microsoft made this edition free of charge to the public.

Embedded (SSEE)

SQL Server 2005 Embedded Edition is a specially configured named instance of the SQL Server Express database engine which can be accessed only by certain Windows Services.

Evaluation

SQL Server Evaluation Edition, also known as the Trial Edition, has all the features of the Enterprise Edition, but is limited to 180 days, after which the tools will continue to run, but the server services will stop.

Fast Track

SQL Server Fast Track is specifically for enterprise-scale data warehousing storage and business intelligence processing, and runs on reference-architecture hardware that is optimized for Fast Track.

LocalDB

Introduced in SQL Server Express 2012, LocalDB is a minimal, on-demand, version of SQL Server that is designed for application developers.[18] It can also be used as an embedded database

Analytics Platform System (APS)

Formerly Parallel Data Warehouse (PDW) A massively parallel processing (MPP) SQL Server appliance optimized for large-scale data warehousing such as hundreds of terabytes.

Datawarehouse Appliance Edition

Pre-installed and configured as part of an appliance in partnership with Dell & HP base on the Fast Track architecture. This edition does not include SQL Server Integration Services, Analysis Services, or Reporting Services.sqlcmd

ARCHITECTURE ;

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their Sybase SQL Server relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, named pipes, and shared memory. Consequently, access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over web services.

3.3 OUTLOOK OF THE PROJECT

- Aims to Create or Develop a system that is capable and reliable in the whole transaction flow such as tracking, retrieving and storing data in appropriate way.
- Stock proportionality is the objective of interest driven stock administration.
- The essential ideal result is to have a similar number of days' worth of stock available over all items so that the season of run out of all items would be concurrent.
- Thus, there is no "overabundance stock".
- Provides a platform to store information in database, summary report of annual,monthly,weekly reports.
- To illustrate and analyze the basic billing system and the main functionalities that surround the billing system from a business prospective and explains how each interacts to complete the billing cycle.
- To develop a billing system emulator that can bill more quickly and accurately.
- To update customer record

3.4 OVERVIEW

3.4.1 BILL GENERATOR

This is the main module of our venture. In which we can keep up the subtle elements of bill administration, with the goal that we can organize information in one table arrangement. The proprietor of this organization can discover every one of the subtle elements in one view page design. All the stock section ought to be keep up in that view, so it can be refreshed in each time as saucy the necessity.

3.4.2 PRINT BILL

Printing module can print out the design of the bill. It can be changed over into an pdf design, and produced the bill for the client.

3.4.3.VIEW BILL

In this the proprietor of this organization can see the one tab detail of all the client, which they purchase over a day, week or a month for the count the item, so they can assume the which item are required for the customer and book keeping to that they can settle the markdown and different offers.

3.4.4 INVENTORY

Inventory control is also about knowing where all your stock is and ensuring everything is accounted for at any given time. Inventory control involves keeping track of the stock that is already in the warehouse, such as what products are being stocked and how much of a particular item is available. It also involves aspects of warehousing designs, such as knowing where everything is and ensuring that the products are stored well.

3.4.5 PRINCIPLE OF INVENTORY PROPORTIONALITY

Stock proportionality is the objective of interest driven stock administration. The essential ideal result is to have a similar number of days' worth of stock available over all items so that the season of run out of all items would be concurrent. In such a case, there is no "overabundance stock," that is, stock that would be left over of another item when the primary item runs out. Overabundance stock is imperfect on the grounds that the cash spent to acquire it could have been used better somewhere else, i.e. to the item that just ran out.

3.4.6 APPLICATIONS

The procedure of stock proportionality is most fitting for inventories that stay inconspicuous by the purchaser, rather than "keep full" frameworks where a retail customer might want to see full retires of the item they are purchasing so as not to think they are purchasing something old, undesirable or stale; and separated from the "trigger point" frameworks where item is reordered when it hits a specific level; stock proportionality is utilized viably by without a moment to spare assembling procedures and retail applications where the item is escaped see.

3.5 FEATURES OF BILLING SYSTEM

3.4.1 PRODUCTS OR ITEMS:

Items are the products that you sell to various clients and can be purchased from various vendors in a business. There are two type of item. One is to track inventory and other one for customer items which do not include to inventory.

- General settings(name, code, descriptions, category)
- Sales pricing (Set Sales pricing for Inventory Item.)
- Purchase pricing(Set Purchase pricing for Inventory Item)
- Transaction(Shows each transaction history of an item with date and quantity)
- Status (When user create an item, it will shows stock status for all location for this company with quantity of item available.)

3.4.2 PURCHASES

A purchase order is an official document that you issue to your vendors, indicating relevant information about what you want to buy, the quantity, the price agreed for that particular product or service.

- Select a Supplier
- Select a store location
- Reference number will generate automatically.
- select items, supply quantity and click Add Item
- Write note of purchase order.
- Submit your order.

3.4.3 CUSTOMERS

Admin has permission of add, edit and delete customers. This is to manage all customers who are purchasing items/products from this selected company.

3.4.4.SUPPLIERS:

This is almost similar to customers. Admin can create suppliers for his/her company by inputting suppliers name, email, address and phone number.

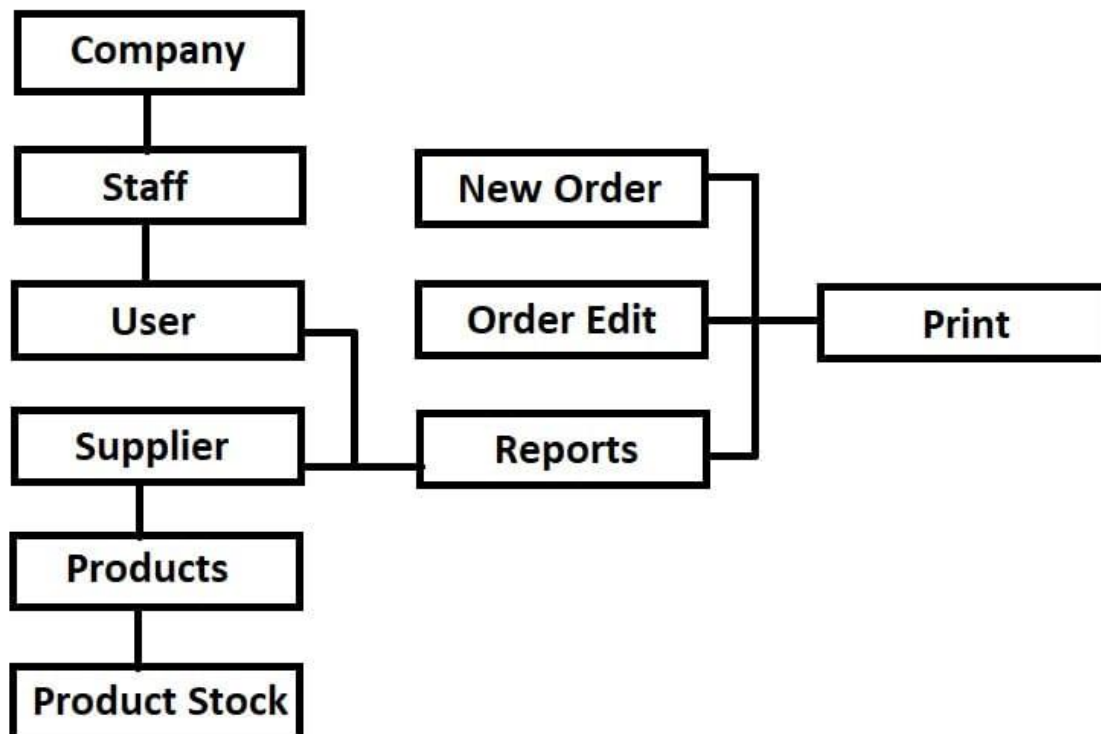
3.4.5 REPORTS:

Various reports are generated as they are given below.

- Inventory stock on Hand
- Sales Report
- Purchase Report
- Expense Report
- Team Member Report

3.6 MODULES

Module of Billing App for inventory



3.6.1 MODULES EXPLANATION

- **COMPANY:** Storing the details of the company such as Name of the Organization, address, phone no., etc...
- **STAFF:** Storing the details of the employee such as Name, address, phone no., etc...
- **USER:** Details of the user using the application.
- **SUPPLIER:** Buying the stock(items or products) from the supplier.
- **PRODUCTS:** Storing the Details of the Products(name, manufacture date, exp).
- **PRODUCT STOCK:** Details of the product stock available in the inventory.
- **NEW ORDERS:** Selling the products when a order is placed.
- **ORDER EDIT:** Customizing and Editing the Order.
- **REPORT:** Report of the Staff, company, stock, product such as ANNUAL, MONTHLY, WEEKLY and on DAILY basis.
- **PRINT:** Prints All of the details required.

4.RESULT AND PERFORMANCE ANALYSIS:

4.1 Login Page :

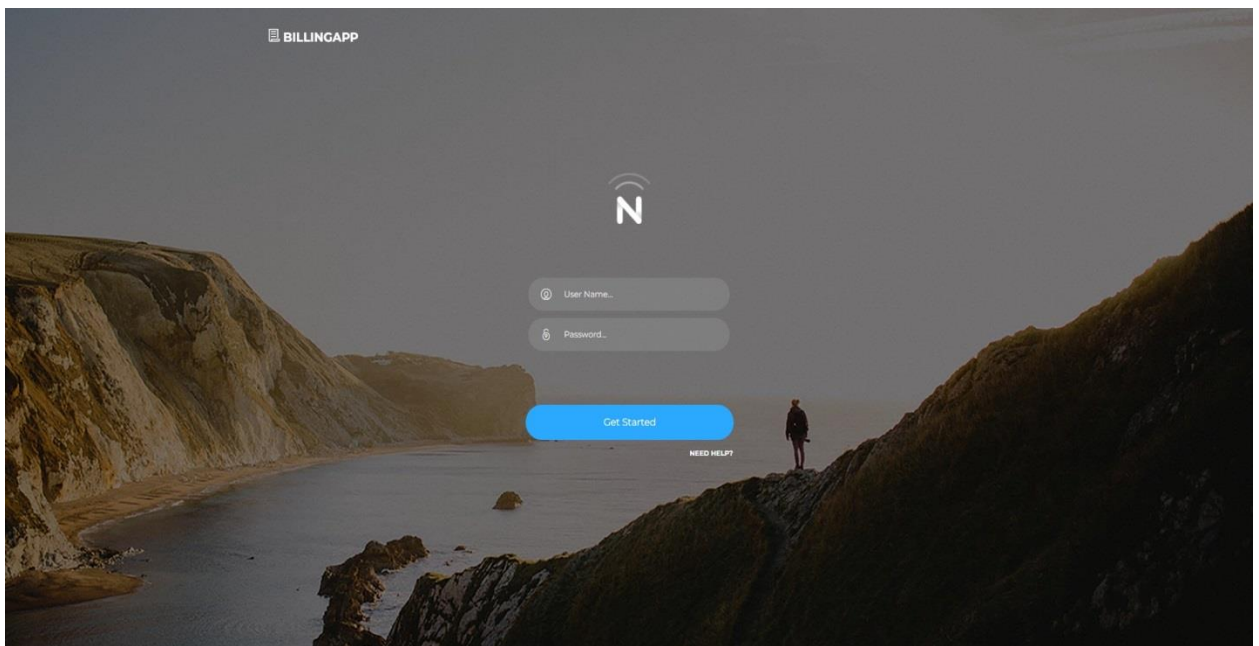


Fig 4.1 : Login Page for users

This is the home page or the main page of our project where the end users can able login with their credentials .The credentials entered in the textfields are verified with backend database and then the user can be logged in to the portal.

4.2 FEATURES OF APPLICATIONS:

BILLINGAPP HOME MASTER ORDER REPORTS ADMINISTRATOR LOGOUT

Supplier Details

Supplier Name...	Contactno...
Address Line 1...	Fanno...
Address Line 2...	Email...
City...	Enter Website Here...
State...	Enter Description Here...
Postcode...	
Country...	

Save

Fig 4.2 supplier details page

The admin once logged in can able to create a new supplier details in his database.

BILLINGAPP [HOME](#) [MASTER](#) [ORDER](#) [REPORTS](#) [ADMINISTRATOR](#) [LOGOUT](#)

Customer Details

Customer Name...

Email...

Contact Number...

Enter Address Here...

Aathar Number...

[Save](#)

Fig 4.3 Customer Details page

In this page the admin can able to create a new customer records for his business.

 **BILLINGAPP**

 HOME

 MASTER ▾

 ORDER ▾

 REPORTS ▾

 ADMINISTRATOR

 LOGOUT

Company Details

Company Name...

Address Line 1...

Address Line 2...

City...

State...

Postcode...

Country...

Contactno...

GST No...

Email...

Enter Website Here...

Enter Description Here...

 Save

Fig 4.4 Company Details

The admin can able to create a new company details here in this field so that in future he/she can able to see the details of the company to whom they are tied up.

 **BILLINGAPP**

 HOME

 MASTER ▾

 ORDER ▾

 REPORTS ▾

 ADMINISTRATOR

 LOGOUT

User Details

▾

Select Category : ☐ Admin ☐ Staff

Save

Fig 4.5 User Details

 **BILLINGAPP**

 HOME

 MASTER ▾

 ORDER ▾

 REPORTS ▾

 ADMINISTRATOR

 LOGOUT

Staff Details

▾

Save

Fig 4.6 Staff details

SUPPLIER LIST									
SN	NAME	CONTACT	ADDRESS	CITY	PIN CODE	STATE	EMAIL	WEBSITE	REMARKS
1	udhaya	5	no 7 south canal bank road	chennai	600028	tamilnadu	jasa@gmail.com		
2	hallow	9361615431	no.7 mylapore	chennai	600004	tamilnadu	hallow@gmail.com		dress

Fig 4.7 supplier report page

STAFF LIST							
SN	STAFF NO	NAME	DESIGNATION	CONTACT	AATHAR	EMAIL	ADDRESS
1	ILHC002	administrator	staff	9361615421	5	jafra@gmail.com	
2	NKSD001	hazard	staff	9841640887	6	hazd@gmail.com	

Fig 4.8 staff report page

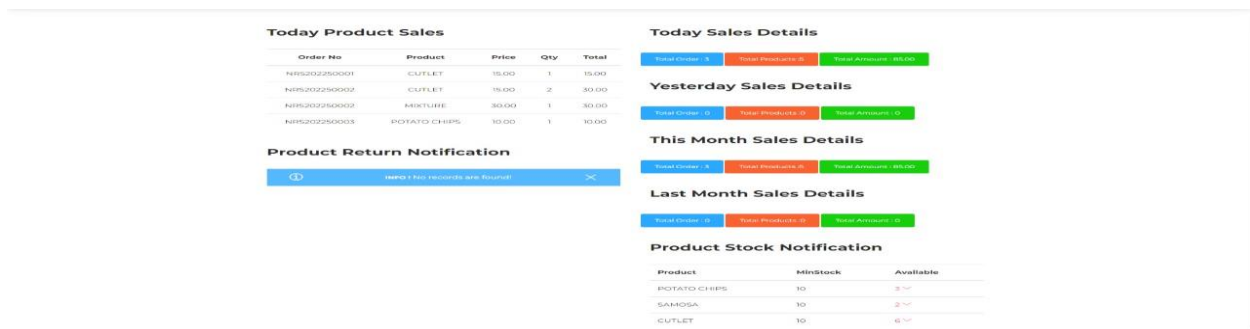


Fig 4.9 sales report page

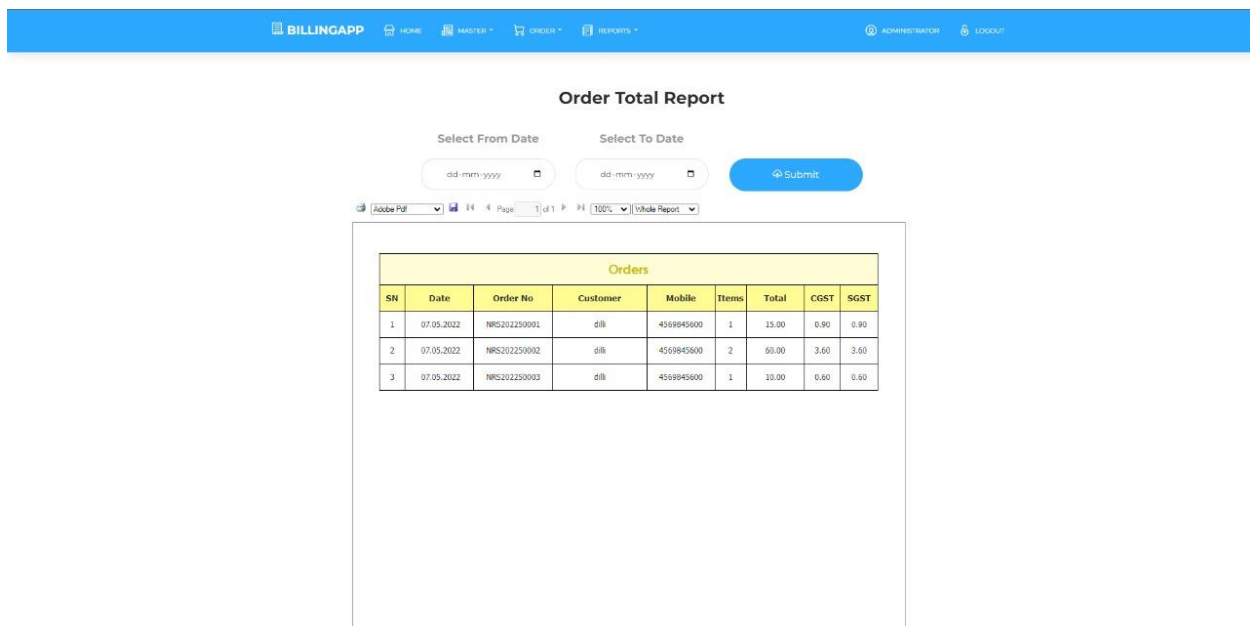


Fig 4.10 order total Report page

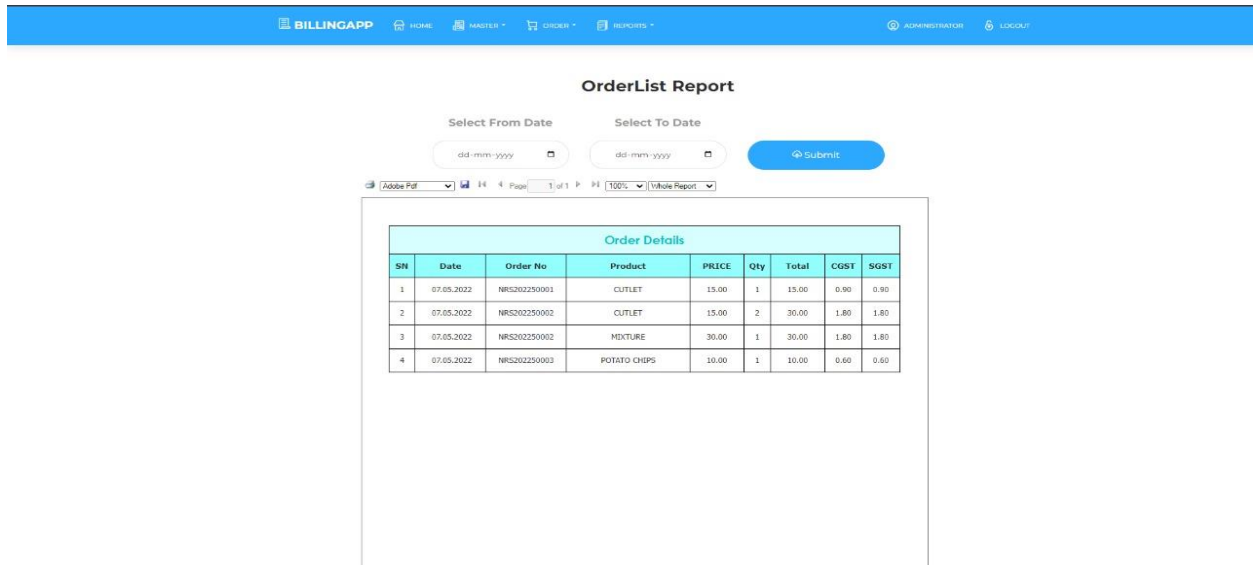


Fig 4.11 order total list report page

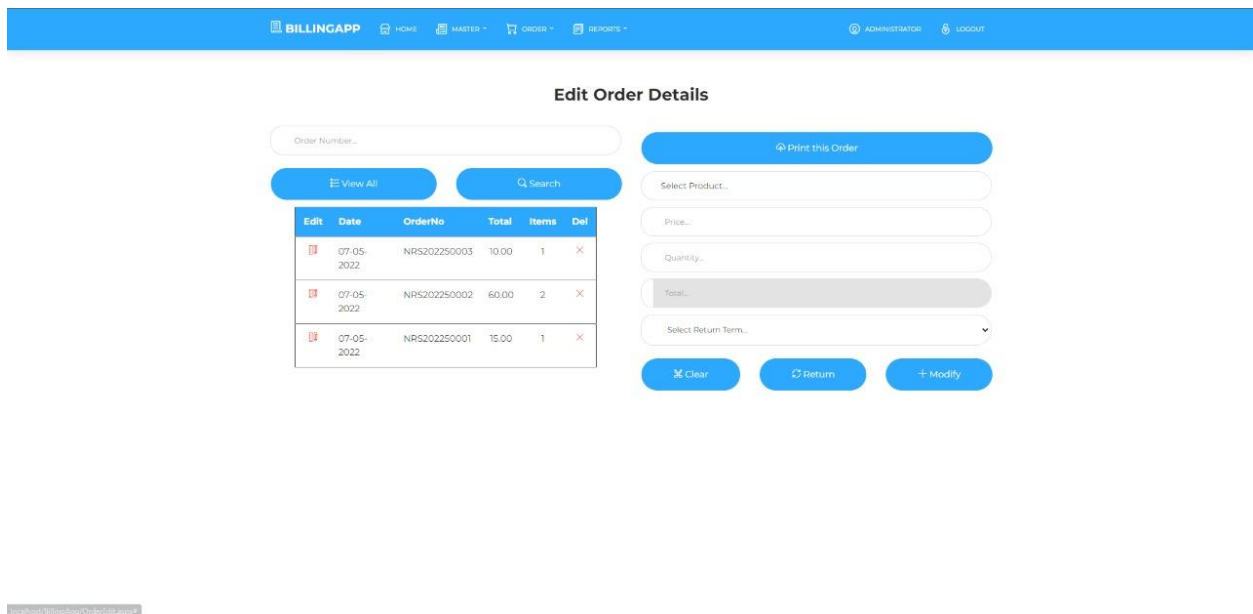


Fig 4.12 edit order details

Customer Report					
Adobe Pdf 14 Page 1 of 1 100% Whole Report					
CUSTOMER DETAILS					
SN	NAME	MOBILE	AATHAR	EMAIL	ADDRESS
1	ana	9841542897	5	hello@gmail.com	no 15 trust road
2	dilli	4565845600	3	dilli@gmail.com	3 farookh st chennai 35

Fig 4.13 Customer report page

Company Report								
Adobe Pdf 14 Page 1 of 1 100% Whole Report								
COMPANY LIST								
SN	NAME	CONTACT	ADDRESS	CITY	PIN CODE	STATE	EMAIL	REMARKS
1	hotchips	93616119421	no 7 south canal bank road	chennai	600028	tamilnadu	jafra@gmail.com	
2	TRINEDS	8736368421	no.13 mandaveli lane mandaveli	CHENNAI	600004	tamilnadu	check@gmail.com	

Fig 4.14 Company report page

BILLINGAPP

HOMEMASTER ~ORDERS ~REPORTS ~

ADMINISTRATORLOGOUT

Add Products

Select Product...

Price...

Quantity...

Total...

X Cancel

+ Add

Customer Details

Customer Mobile...

Customer Name...

Save

Save & Print

ORDER INFO - 1102302200000

DATE: 07-08-2022

INFO ! No records are Added!

Fig 4.15 Adding products to cart page

CHAPTER 5

5.1 CONCLUSION

We would like to address our billing app for inventory for the end-users who need the billing software that would be easy to handle in all the conditions. We had developed this project from the end-users perspective those who are not that familiar with handling computer software. So that they can able to take the daily and custom records for their business development. All the navigations in our software are more attractive for the users who are using our billing software.

5.2 FUTURE WORK

In the future, we hope to implement the face detection login page to save time instead of simply typing the credentials in the text fields. And also we hope to implement biometric verification for all the users to make an identity verification of the staff of their company, customers, etc.

REFERENCES :

1. Lesonsky, Rieva(1998). "Tracking Inventory". Entrepreneur Magazine.
2. Dolinsky, Anton. "Inventory Management History Part Four". Almyta Systems. Retrieved August 17, 2010
3. "Integrations and Apps for Online Inventory Management. SoftwareTradeGecko". www.tradegecko.com. Retrieved 2015-11-24.
4. Lockard, Robert (29 November 2010). "3 Advantages of Using Inventory Management Software". Inventory System Software Blog. Retrieved 23 November 2012.
5. H. Kaushik, B. Mounica, B. Swathi, "A survey on monitoring systems," International Journal of Computer Science and Mobile Computing, January 2015.
6. S. Angel Raphella, S. Gomathi Nathan and G. Chitra, Inventory Management- A Case Study, International Journal of Emerging Research in Management & Technology.
7. .P.G. Matsebatlela and K. Mpofu, Inventory Management Framework to Minimize Supply and Demand Mismatch on a Manufacturing Organization, International Federation of Automatic Control,

APPENDIX :

A.SOURCE CODE

HOME PAGE

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Home.aspx.cs"
Inherits="BillingApp.Home" %>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8" />
```

```
  <link rel="apple-touch-icon" sizes="76x76" href="/assets/img/apple-icon.png">
```

```
  <link rel="icon" type="image/png" href="/assets/img/favicon.png">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
```

```
  <title>BillingApp</title>
```

```
  <meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0, shrink-to-fit=no' name='viewport' />
```

```
  <!--   Fonts and icons   -->
```

```
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200"
rel="stylesheet" />
```

```
  <link href="https://use.fontawesome.com/releases/v5.0.6/css/all.css"
rel="stylesheet">
```

```

<!-- CSS Files -->
<link href="./assets/css/bootstrap.min.css" rel="stylesheet" />
<link href="./assets/css/now-ui-kit.css?v=1.2.0" rel="stylesheet" />
<!-- CSS Just for demo purpose, don't include it in your project -->
<link href="./assets/demo/demo.css" rel="stylesheet" />
</head>

<body class="index-page sidebar-collapse">
  <form id="form1" runat="server">
    <!-- Navbar -->
    <MenuCtrl:MenuControls id="MenuUserControls" runat="server" />
    <!-- End Navbar-->

    <!-- Carousel-->
    <!-- #Include file="MenuControl/Carousel.html" -->
    <!-- End Carousel-->

  </form>

  <!-- Core JS Files -->
  <script src="./assets/js/core/jquery.min.js" type="text/javascript"></script>
  <script src="./assets/js/core/popper.min.js" type="text/javascript"></script>
  <script src="./assets/js/core/bootstrap.min.js" type="text/javascript"></script>
  <!-- Plugin for Switches, full documentation here:
  http://www.jque.re/plugins/version3/bootstrap.switch/ -->
  <script src="./assets/js/plugins/bootstrap-switch.js"></script>

```

```

<!-- Plugin for the Sliders, full documentation here: http://refreshless.com/nouislider/ -
->
<script src="/assets/js/plugins/nouislider.min.js" type="text/javascript"></script>
<!-- Plugin for the DatePicker, full documentation here:
https://github.com/uxsolutions/bootstrap-datepicker -->
<script src="/assets/js/plugins/bootstrap-datepicker.js"
type="text/javascript"></script>
<!-- Google Maps Plugin -->
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_KEY_HERE"></script>
<!-- Control Center for Now Ui Kit: parallax effects, scripts for the example pages etc -
->
<script src="/assets/js/now-ui-kit.js?v=1.2.0" type="text/javascript"></script>
<script>
$(document).ready(function() {
    // the body of this function is in assets/js/now-ui-kit.js
    nowuiKit.initSliders();
});

function scrollToDownload() {

    if ($('#section-download').length != 0) {
        $("html, body").animate({
            scrollTop: $('#section-download').offset().top
        }, 1000);
    }
}

```

```
}  
</script>  
</body>  
  
</html>
```

ORDER EDIT PAGE MODULE :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="OrderEdit.aspx.cs"  
Inherits="OrderEdit" %>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <meta charset="utf-8" />  
  <link rel="apple-touch-icon" sizes="76x76" href="./assets/img/apple-icon.png" />  
  <link rel="icon" type="image/png" href="./assets/img/favicon.png" />  
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />  
  <title>Order Edit</title>  
  <meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-  
scalable=0, shrink-to-fit=no' name='viewport' />  
  <!--    Fonts and icons    -->  
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200"  
rel="stylesheet" />
```

```
<link href="https://use.fontawesome.com/releases/v5.0.6/css/all.css" rel="stylesheet" />
```

```
<!-- CSS Files -->
```

```
<link href="/assets/css/bootstrap.min.css" rel="stylesheet" />
```

```
<link href="/assets/css/now-ui-kit.css?v=1.2.0" rel="stylesheet" />
```

```
<link href="assets/css/GridDesign.css" rel="stylesheet" />
```

```
<link href="assets/css/sweetalert.css" rel="stylesheet" />
```

```
<script src="assets/js/core/sweetalert.min.js"></script>
```

```
<link href="assets/css/chosen.min.css" rel="stylesheet" />
```

```
<!-- CSS Just for demo purpose, don't include it in your project -->
```

```
<link href="/assets/demo/demo.css" rel="stylesheet" />
```

```
<%-- <link href="assets/css/chosen.min.css" rel="stylesheet" />--%>
```

```
<%--<script src="assets/js/core/JScript.js"></script>--%>
```

```
<script>
```

```
    //***** This Message for Cancel button onclick event
```

```
    *****/
```

```
    function deletealert(ctl, event) {
```

```
        // STORE HREF ATTRIBUTE OF LINK CTL (THIS) BUTTON
```

```
        var defaultAction = $(ctl).prop("href");
```

```
        // CANCEL DEFAULT LINK BEHAVIOUR
```

```
        event.preventDefault();
```

```
        swal({
```

```
            title: "Are you sure?",
```

```
            text: "Are you sure cancel this order details?",
```

```
            type: "warning",
```

```

        showCancelButton: true,
        confirmButtonColor: "#DD6B55",
        confirmButtonText: "Yes, cancel it!",
        cancelButtonText: "No, don't!",
        closeOnConfirm: false,
        closeOnCancel: false
    },
    function (isConfirm) {
        if (isConfirm) {
            swal({ title: "Canceled!", text: "Successfully canceled this order", type:
"success", confirmButtonText: "OK!", closeOnConfirm: false },
            function () {
                // RESUME THE DEFAULT LINK ACTION
                window.location.href = defaultAction;
                return true;
            });
        } else {
            swal("Safe", "Order detail is safe!", "error");
            return false;
        }
    });
}

```

// ***** This Message for Grid remove button *****//

```
function confirmOrderDel(ctl, event) {
```

```

// STORE HREF ATTRIBUTE OF LINK CTL (THIS) BUTTON
var defaultAction = $(ctl).prop("href");
// CANCEL DEFAULT LINK BEHAVIOUR
event.preventDefault();
swal({
    title: "Remove Order?",
    text: "Are you sure you want to remove this data?",
    type: "warning",
    showCancelButton: true,
    confirmButtonColor: "#DD6B55",
    confirmButtonText: "Yes, remove it!",
    cancelButtonText: "No, don't!",
    closeOnConfirm: false,
    closeOnCancel: false
},
function (isConfirm) {
    if (isConfirm) {
        swal({ title: "Removed!", text: "Successfully removed this order", type:
"success", confirmButtonText: "OK!", closeOnConfirm: false },
        function () {
            // RESUME THE DEFAULT LINK ACTION
            window.location.href = defaultAction;
            return true;
        });
    } else {
        swal("Safe", "Order data is safe!", "error");
    }
}

```

```

        return false;
    }
});
}

// ***** This Message for Grid remove button *****//
function calc() {
    var price = document.getElementById("txtPrice").value;
    var qnty = document.getElementById("txtqnty").value;
    var total = parseFloat(price) * qnty
    if (!isNaN(total))
        document.getElementById("txtTotal").value = parseFloat(total, 10).toFixed(2);
}
function calc2() {
    var price = document.getElementById("txtPrice").value;
    document.getElementById("txtPrice").value = parseFloat(price, 10).toFixed(2);
    var qnty = document.getElementById("txtqnty").value;
    var total = parseFloat(price) * qnty
    if (!isNaN(total))
        document.getElementById("txtTotal").value = parseFloat(total, 10).toFixed(2);
}

function CheckNumeric(event) {
    var _key = (window.Event) ? event.which : event.keyCode;

```

```

        if ((_key > 95 && _key < 106) || (_key > 47 && _key < 58) || _key == 8 || _key ==
9 || _key == 37 || _key == 39 || _key == 190 || _key == 13 || _key == 110) {
            return true;
        }
        else {
            return false;
        }
    }
}
$(function () {
    $("#txtqty").change(function () {
        var max = parseInt($(this).attr('max'));
        var min = parseInt($(this).attr('min'));
        if ($(this).val() > max) {
            $(this).val(max);
        }
        else if ($(this).val() < min) {
            $(this).val(min);
        }
    });
});
//function Showpop() {
//    $("#myModal").modal("show");
//}
</script>
</head>
<body class="index-page sidebar-collapse">

```

```

<form id="form1" runat="server">
    <!-- Navbar -->
    <MenuCtrl:MenuControls ID="MenuUserControls" runat="server" />
    <!-- End Navbar-->
    <!-- Form-->
    <div class="container">
        <div class="col-lg-6 text-center col-md-8 ml-auto mr-auto">
            <h3 class="title">Edit Order Details</h3></div>
        <div class="row justify-content-center">
            <div class="col-lg-6 col-md-8 ml-auto mr-auto">
                <div class="row">
                    <div class="col">
                        <div class="input-group input-lg">
                            <div class="input-group-prepend">
                                <span class="input-group-text"></span>
                            </div>
                            <asp:TextBox class="form-control" ID="txtOrderNo"
placeholder="Order Number..." runat="server"></asp:TextBox>
                        </div>
                    <div class="row">
                        <div class="col">
                            <div class="send-button">
                                <a class="btn btn-info btn-round btn-block btn-lg"
onserverclick="btnViewAll_OnClick" runat="server" id="A3"><i class="now-ui-icons
design_bullet-list-67"></i>&nbsp;View All</a>
                            </div>

```

```

        </div>
        <div class="col">
            <div class="send-button">
                <a class="btn btn-info btn-round btn-block btn-lg"
onserverclick="btnSearch_OnClick" runat="server" id="A1"><i class="now-ui-icons ui-
1_zoom-bold"></i>&nbsp;Search</a>
            </div>
        </div>
    </div>
    <asp:HiddenField ID="hdOrderID" runat="server" />
    <asp:HiddenField ID="hdOrderNo" runat="server" />
    <asp:HiddenField ID="hdTotAmt" runat="server" />
    <asp:HiddenField ID="hdgridRowcnt" runat="server" />
</div>
</div>
<div class="row justify-content-center">
    <asp:GridView ID="gdOrder" runat="server"
AutoGenerateColumns="False" CellPadding="4" CssClass="table"
        PageSize="10" ShowFooter="false" ForeColor="#333333"
FooterStyle-BackColor="#202020" FooterStyle-ForeColor="White"
        GridLines="Horizontal"
OnPageIndexChanging="OnPageIndexChanging" AllowPaging="true"
        PagerStyle-CssClass="mydatagrid" HeaderStyle-CssClass="bg-info"
HeaderStyle-ForeColor="White"
        Width="100%">
    <AlternatingRowStyle BackColor="White" />

```

```

<Columns>
    <asp:TemplateField HeaderText="Edit" ItemStyle-
HorizontalAlign="Center">
        <ItemTemplate>
            <asp:LinkButton ID="lnkEdit" runat="server"
CommandArgument='<%# Eval("OrderID")%>' OnCommand="EditOrder"><i
class="now-ui-icons design-2_ruler-pencil"></i></asp:LinkButton>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:BoundField DataField="CreatedDate" HeaderText="Date"
ItemStyle-Width="100px" DataFormatString = "{0:dd/MM/yyyy}" />
    <asp:BoundField DataField="OrderNo" HeaderText="OrderNo"
ItemStyle-HorizontalAlign="Left" />
    <asp:BoundField DataField="TotalAmount" HeaderText="Total"
ItemStyle-HorizontalAlign="Center" />
    <asp:BoundField DataField="ItemCount" HeaderText="Items"
ItemStyle-HorizontalAlign="Center" />
    <%--<asp:BoundField DataField="Name" HeaderText="Customer"
ItemStyle-HorizontalAlign="Center" HeaderStyle-HorizontalAlign="Center" />--%>
    <%-- <asp:BoundField DataField="StaffName"
HeaderText="Cashier" ItemStyle-HorizontalAlign="Left" HeaderStyle-
HorizontalAlign="Center" />--%>
    <asp:TemplateField HeaderText="Del" ItemStyle-
HorizontalAlign="Center">
        <ItemTemplate>

```

```

        <asp:LinkButton ID="btnDelete" runat="server"
OnClientClick="return confirmOrderDel(this, event);" CommandArgument='<%#
Eval("OrderID") %>' OnCommand="OnGridRowRemove"><i class="now-ui-icons ui-
1_simple-remove"></i></asp:LinkButton>
    </ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>
</div>
</div>
<div class="col-lg-6 text-center col-md-8 ml-auto mr-auto">
    <div class="row">
        <div class="col">
            <div class="send-button">
                <a class="btn btn-info btn-round btn-block btn-lg"
onserverclick="Print" runat="server" id="A2"><i class="now-ui-icons arrows-1_cloud-
upload-94"></i>&nbsp;Print this Order</a>
            </div>
        </div>
    </div>
</div>
<div class="row">
    <div class="col">
        <span id="sporderno" runat="server" style="text-align: left; float: left"
class="badge badge-primary text-lg-left"></span><span id="spdate" runat="server"
style="text-align: right; float: right" class="badge badge-default"></span>
    </div>

```

```

</div>
<div class="row justify-content-center">
    <asp:GridView ID="gvOrderList" runat="server" CellPadding="4"
OnRowDataBound="gvOrderList_RowDataBound" CssClass="table"
        PageSize="15" ShowFooter="true" ForeColor="#333333" FooterStyle-
ForeColor="White"
        GridLines="Horizontal"
OnPageIndexChanging="OnPageIndexChanging" AllowPaging="true"
AutoGenerateColumns="false"
        PagerStyle-CssClass="mydatagrid" HeaderStyle-CssClass="bg-info"
HeaderStyle-ForeColor="White" FooterStyle-CssClass="bg-info"
        Width="100%">
        <AlternatingRowStyle BackColor="White" />
        <Columns>
            <asp:TemplateField HeaderText="Edit" ItemStyle-
HorizontalAlign="Center">
                <ItemTemplate>
                    <asp:LinkButton ID="lnkEdit" runat="server"
CommandArgument='<%# Eval("OrderListID")%>' OnCommand="OnRowEdit"><i
class="now-ui-icons design-2_ruler-pencil"></i></asp:LinkButton>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:BoundField DataField="ProductName" HeaderText="Product"
ItemStyle-HorizontalAlign="Left" />
            <asp:BoundField DataField="Price" HeaderText="Price" ItemStyle-
HorizontalAlign="Center" />

```

```

        <asp:BoundField DataField="Qty" HeaderText="Qty" ItemStyle-
HorizontalAlign="Center" />
        <asp:BoundField DataField="Total" HeaderText="Total" ItemStyle-
HorizontalAlign="Center" />
        <asp:TemplateField HeaderText="Del" ItemStyle-
HorizontalAlign="Center">
            <ItemTemplate>
                <asp:LinkButton ID="btnDelete" runat="server"
OnClientClick="return confirmOrderDel(this, event);" CommandArgument='<%#
Eval("OrderListID") %>' OnCommand="OnRowDeleted"><i class="now-ui-icons ui-
1_simple-remove"></i></asp:LinkButton>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
</div>

<div class="row">
    <div class="col">
        <div class="input-group input-lg text-left">
            <%-- <div class="input-group-prepend">
                <span class="input-group-text"></span>
            </div>--%>
            <asp:DropDownList ID="ddlProducts" runat="server" Width="374px"
Height="48px" data-style="btn-info" CssClass="dropdown-info form-control custom-
combobox-input">

```

```

        </asp:DropDownList>
    </div>
    <div class="input-group input-lg">
        <div class="input-group-prepend">
            <span class="input-group-text"></span>
        </div>
        <asp:TextBox class="form-control" ID="txtPrice"
placeholder="Price..." runat="server" oninput="calc()" onkeydown="return
CheckNumeric(event);" MaxLength="15"></asp:TextBox>
    </div>
    <div class="input-group input-lg">
        <div class="input-group-prepend">
            <span class="input-group-text"></span>
        </div>
        <input type="number" class="form-control" id="txtqty"
name="txtqty" placeholder="Quantity..." runat="server" min="0" max="20"
oninput="calc2()" />
    </div>
    <div class="input-group input-lg">
        <div class="input-group-prepend">
            <span class="input-group-text"></span>
        </div>
        <input type="text" class="form-control" id="txtTotal" name="txtTotal"
placeholder="Total..." runat="server" readonly="true" />
    </div>
    <div class="input-group input-lg">

```

```

<div class="input-group-prepend">
  <span class="input-group-text">
    <!-- <i class="now-ui-icons ui-1_email-85"></i>-->
  </span>
</div>

<asp:DropDownList ID="ddlReturn" runat="server" Width="374px"
Height="48px" data-style="btn-info" CssClass="dropdown-info form-control custom-
combobox-input ">
  <asp:ListItem Text="Select Return Term..." Value="0"></asp:ListItem>
  <asp:ListItem Text="Return" Value="1"></asp:ListItem>
  <asp:ListItem Text="Replace" Value="2"></asp:ListItem>
</asp:DropDownList>
</div>

<div class="row">
  <div class="col">
    <div class="send-button">
      <asp:LinkButton ID="btncancel" runat="server" class="btn btn-
info btn-round btn-block btn-lg" OnClick="btnCancel_OnClick" OnClientClick="return
deletealert(this, event);"><i class="now-ui-icons
design_scissors"></i>&nbsp;Clear</asp:LinkButton>
    </div>
  </div>
  <div class="col">
    <div class="send-button">

```

```

        <asp:LinkButton ID="LinkButton1" runat="server" class="btn
btn-info btn-round btn-block btn-lg" OnClick="btnReturn_OnClick"><i class="now-ui-
icons arrows-1_refresh-69"></i>&nbsp;Return</asp:LinkButton>

    </div>

</div>

<div class="col">

    <div class="send-button">

        <a class="btn btn-info btn-round btn-block btn-lg"
onserverclick="btnModify_OnClick" runat="server" id="btnsave"><i class="now-ui-icons
ui-1_simple-add"></i>&nbsp;Modify</a>

        <asp:HiddenField ID="hdOrderListID" runat="server" />
        <asp:HiddenField ID="hdCompid" runat="server" />
        <asp:HiddenField ID="hdQty" runat="server" />

    </div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

<!-- End Form-->

<!-- Core JS Files -->

<script src="./assets/js/core/jquery.min.js" type="text/javascript"></script>
<%--<script src="assets/js/core/chosen.jquery.js"></script>--%>
<script src="./assets/js/core/popper.min.js" type="text/javascript"></script>

```

```

<script src="./assets/js/core/bootstrap.min.js" type="text/javascript"></script>
<!-- Plugin for Switches, full documentation here:
http://www.jque.re/plugins/version3/bootstrap.switch/ -->
<script src="./assets/js/plugins/bootstrap-switch.js"></script>
<!-- Plugin for the Sliders, full documentation here: http://refreshless.com/nouislider/ -
->
<script src="./assets/js/plugins/nouislider.min.js" type="text/javascript"></script>
<!-- Plugin for the DatePicker, full documentation here:
https://github.com/uxsolutions/bootstrap-datepicker -->
<script src="./assets/js/plugins/bootstrap-datepicker.js"
type="text/javascript"></script>
<!-- Google Maps Plugin -->
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_KEY_HERE"></script>
<!-- Control Center for Now Ui Kit: parallax effects, scripts for the example pages etc -
->
<script src="./assets/js/now-ui-kit.js?v=1.2.0" type="text/javascript"></script>

<%-- Chosen Script For ASP Dropdownlist With Search --%>

<script>
function scrollToDownload() {

    if ($('#section-download').length != 0) {
        $("html, body").animate({
            scrollTop: $('#section-download').offset().top

```

```
        }, 1000);  
    }  
}  
</script>
```

```
<script src="assets/js/core/chosen.jquery.js"></script>  
<script>  
$('#<%=ddlProducts.ClientID%>').chosen();  
</script>
```

```
</body>  
</html>
```