MAHATHMA GANDHI MEMORIAL COLLEGE
UDUPI - 576102

PROJECT REPORT ON

"Automation of Cake Shop Management System"

BY

Rachana	Reg.No:163532573
Amrutha Acharya	Reg.No:163532535
Anusha T.A	Reg.No:163532587

Under the guidance of

Mrs. Jayanthi R. Prabhu

Dept of Computer Science

Submitted to Mangalore University in partial fulfillment of the award of Bachelor in Computer Application

MANGALORE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MAHATHMA GANDHI MEMORIAL COLLEGE

UDUPI - 576102

2018-19

MAHATMA GANDHI MEMORIAL COLLEGE
UDUPI - 567102

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

*This is to certify that the Project entitled **"Cake Shop Management System"** has been carried out by:*

Rachana	**Reg. No: 163532573**
Amrutha Acharya	**Reg. No: 163532535**
Anusha T.A	**Reg. No: 163532587**

*Students of **B.C.A.** under the supervision and guidance of **Mrs. Jayanthi R. Prabhu** , Department of Computer Science, Mahatma Gandhi Memorial College, Udupi. This dissertation is submitted in partial fulfillment of the requirement for the award of **Bachelor in Computer Application** by **Mangalore University** during the academic year 2018-2019*

It is certified to the best of our knowledge that the matter embodied in this work has not been submitted for the award of any other degree.

Mrs.Jayanthi R. Prabhu	Prof. Dr. M.G Vijaya	Mr. Vishwanath Pai
(Project Guide)	(Principal)	(Head of the Department)

Submitted to the Mangalore University Practical Examination held on_____at Mahatma Gandhi Memorial College, Udupi.

Examiners:

1)_____

2)_____

To whomever it may concern

This is to certify that project entitled "Cake Shop Management System" for "Crumbz " has been carried out by:

1. Rachana	Reg no: 163532573
2. Amrutha Acharya	Reg no: 163532535
3.Anusha T.A	Reg no: 163532514

Students of BCA, Mahatma Gandhi Memorial College, Udupi, under one guidance and supervision submitted the partial fulfillment of requirements for award for Bachelor of Computer Application of Mangalore University during the period of 2018-2019. It is certified to the best of our knowledge.

Project Guide **Project Manager**

Date:

Place:

Acknowledgement

Behind every achievement, there is a sea of gratitude to those who have activated this project. The magnitude of this project demanded the cooperation, guidance and help of many people. By the grace of God, we have being fortunate enough to have this in the entire task of completion of our project on “Automation Cake Shop Management System”.

We would like to thank our principal Professor. Dr. M. G. Vijaya for giving us an opportunity to carry out project.

We thank Mr. Vishwanath Pai, a source of inspiration and encouragement, Head of the Department Computer Science, Mahatma Gandhi Memorial College Udupi for having permitted us to carry out our project work.

We are extremely great full to express our overwhelming gratitude to our guide Mrs.Jayanthi R. Prabhu, Lecturer of Computer Science Department, Mahatma Gandhi Memorial College Udupi for giving us valuable guidance to undertake this project .

First and foremost we are always thankful to Mr. Fredrick Mathias the manager and all the members of “Crumbz” for the kind co-operation and providing all the necessary information for developing a good system by supporting us till the successful completion of our project.

Last but not the least; we are indebted to the teaching and nonteaching staff members, Mahatma Gandhi Memorial College Udupi for making this project successful.
Thank you,

Rachana

Amrutha Acharya

Anusha T.A

INDEX

Automation of Cake Shop Management System

Introduction

PURPOSE:

• This specific tool is for managing the Automation of Cake Shop Management System.
• This application will keep records of the purchase and sales details of the company.
• It will manage the details of Stock.

SCOPE:

This system uses SQL server to store information, so many users can access the database from the server if they have a proper access right. This is user friendly. User can save, retrieve and update records on a single click using the mouse and keyboard. Any modification can be done easily. The application can be used to maintain the stock record, purchase and sales details, along with helping in managing the employee wages.

TECHNOLOGIES TO BE USED:

The different phase of development and support applications are as follows:

• Database Design(SQL Server)
• Form Design(ASP.NET)
• Coding(ASP.NET)
• Testing(ASP.NET)

PROJECT OVERVIEW:

Crumbz is situated in Manipal, Udupi District, which produces varieties of cakes, pastries and snacks where raw materials are received from various suppliers.

It offers purchase orders to different suppliers and imports the goods. These details are recorded for future reference. The raw materials undergo manufacturing process by the employees of the industry. After processing, the produced cakes are updated to the stocks. The product is then exported to customers based upon the orders.

All these recordings are done manually on paper (register). But this not only takes time and manpower, but also it is not so safe, secure and accurate. So an application is required to be developed which store all these data under norms and also necessary calculations. This application is supported to keep updating the database with the purchase orders, manufacturing details and Sales order. This also maintains the stock details, after the day's transactions are over.

The main intention of introducing the database to the inventory is to make the work of the organization mere flexible and easy recovery at any moment of time, whenever necessary.

SYNOPSIS

TITLE OF THE PROJECT:

Automation of Crumbz Cake Shop Management System

Introduction:

The cake shop system is application which is based on ordering and selling the cake and other items and generating bill. It is user friendly and modular approach. Through this project we provide software to automate the processes of Crumbs. This software provides the major processes such as purchase, sales and bill details.

Objective:

- Make all the systems computerized, means no paper work.
- Reduce time consumption.
- To maintain information about products, purchases, sales etc. systematically.
- Simple databases maintained.
- To generate reports on stock and transaction.

Project category:

- This is client server.

Tools/Platform:

- **Hardware requirements:-**
  - CPU
  - Ram

- Hard Disk
- Other hardware

- **Software requirements:-**
  - Operating system
  - Front end : Microsoft visual studio 2005
  - Back end : Microsoft SQL server 2005

Structure of the project:

- **Modules:-**
  - Login
  - Customer module
  - Employee module
  - Supplier module
  - Inventory
  - Inter branch transfer

- **Module description:-**

1) **Login module**
   This module provides security by the use of user name and password.
   - **Administrator :**
     Admin has power to add user and remove user. He can also change the password of the user.

- **User :**
  User has the user id and password assigned by administrator.

2) Customer Module

- **Customer information:**
  This module contains information about customers such as bill no, name, address, DOB etc.
- **Sales order:**
  This module contains details of the order placed by the customer.
- **Sales bill:**
  This module maintains the billing information of customer order.

3) Employee Module

- **Employee details:**
  This module contains information about staff such as name, address, phone no, DOB,DOJ etc.
- **Attendance:**
  This module contains the information about number of working days of employee.
- **Salary:**
  This module calculates the salary of employee.

4) Supplier Module

- **Supplier information:**
  This module stores supplier information such as name, address, email, contact info etc.
- **Purchase order:**
  This module contains the information order placed to the different supplier.

- **Purchase bill:**
  This module contains the bill information given by the supplier.

5) **Inventory**
   This module contains information about raw materials and finished products.
   - **Raw materials**(items):
     This module contains information about the raw materials
   - **Finished products**(products):
     This module contains information about the finished product

6) **Inter branch transfer**
   This module contains information about the product which is transferred to other branches.
   - **Branch information:**
     This module contains information about branch such as branch name, address, contact no etc.
   - **Branch order:**
     This module contains order placed by the other branch.
   - **Branch bill:**
     This module contains bill info

Testing process:

- Manual testing with real time data

Software Type:

- Client server software

Name and address:

- Name : Crumbz
- Address :Pratham Pride Building,

  End point Road, Manipal

  Udupi-576 104

LANGUAGES USED

FRONT-END:

ASP.NET:

ASP.NET is an object oriented programming language with many features such as inheritance, interface and overloading. It is a common language specification complaint that is any other language that is CLS complaint that creates in Visual Basic.Net.

Data Access Application: All corporate data is stored in one database or the other. Therefore, one of the most common requirements in application development is to access that data that is already present in the table. The data is not only to be displayed but the user must be able to retrieve specific data and update the data easily and quickly. To do so, the application requires some form of data access, if a new application is being created, and then there are 3 access choices.

1. ADO.NET

2. ADO

3. OLEDB

For existing application in the long run the newer data access technologies may have to re-engineered thus technologies reduce development the simply code and provide excellent performance.

OLEDB:

OLEDB is the strategic system level-programming interface for accessing data and is the underlying technology for ado as well as source of data for ADO.NET.

OLEDB is an open standard for accessing all kinds of data relational data including: main frame ISAM/ISAM and hierarchical databases, email and file

system stores; text, graphical, and graphical data; and custom business objects and OLEDB provides consistent, high performance access to data and supports a variety of development needs, including the creation of form and database clients and middle-tier business objects using live connection to data in relational databases and other stores.

CRYSTAL REPORT:

It is a reporting tool with the ability to create interactive, presentation quality report very easily and quickly. It has been in use for many years. A crystal report for ASP.Net has been in use for the .Net platform to provide .Net developers with the richest API possible. Now crystal reports is a standard reporting tool, which is part .Net platform with crystal report for ASP.Net, you can host reports on web and windows platform and publish crystal reports web services on a web server.

BACK-END:

SQL SERVER 2005:

About SQL Server 2005:

Microsoft SQL Server 2000 introduced Service Broker, a new Technology for building database intensive Distributed application that is server, reliable, scalable.

Service Broker-Description:

Service broker is a part of the DB engine service broker provide,

 Facilities for storing message queues in SQL server databases. New transact-SQL Statements that application can use to send and receive message is a part of a dialog: a reliable, persistent communication channel between 2 participants. Service broker provides. Unique capabilities for both applications that distribute work across multiple SQL server instances.

SQL SERVER 2005 TECHNOLOGIES:

SQL server 2005 contains these technologies.

1. SQL Server Database Engine overview: The database engine is the core service for storing, processing and security data. The DB engine provides controlled access rapid transaction. Processing to meet the requirements of the most demanding data consuming application within you enterprise the database engine also provides rich support for sustaining high availability.

1. SQL Server Analysis Services Overview: Analysis service delivers online analytical processing (OLAP) and data mining functionality for business intelligence application, Analysis services supports OLAP by allowing you to design, create and, manage multidimensional structures that contain data aggregated from other.

Data sources, such as relational databases for data applications, Analysis service allow you to design, create and visual data mining modules, constructed from other. Data source by using variety of industry standard data mining algorithms.

2. **SQL SERVER Integrating service (SSIS) overview:** An integration service is an enterprise data transformation and data integration solution that you can use to extract, transforms, and consolidates data form. Disparate sources and move it to single or multiple destinations.

3. **SQL server Replication overview:** Replication is a set of technologies for coping and distributing data database objects from one database to another and then synchronizing between database to maintain consistency. Using replication, you can distribute data to different locations and remote or mobile users over local and wide area network, dial-up connections, and the intent.

1. SQL reporting device overview: Reporting services is a new service based reporting platform that you can use to create and manage tubular, matrix, graphical and free form reports that contain data from relational and

multidimensional data sources. The reports that you create can be viewed and managed over a web based connection.

2. SQL server notification services overview: Notification service is the platform for developing a deploying application that generate and send notification services can generate and send timely personalized messages to thousands or millions of subscribers and deliver them to a wide variety of devices.

3. SQL server broker overview: Service broker is a technology for building reliable, scalable and service database application service broker is a technology within the DB engine that provides native support for queue. Service broker also provides a message that can be used to link disparate application components into a functioning whole service broker provides much of the infra structure necessary to build a distributed application significantly reducing the application development time. Service broker also makes it easy to scale you application up or down to accommodate the amount of traffic the application is receiving.

1. Full-text search overview: SQL server contains the functionality you need to issue full text queries against plain character based data in SQL server. Tables-full text queries could include words and phrases or multiple forms of a word or phrase.

2. SQL server tools and utilities overview: SQL server provides the tools you need to design, develop, deploy and administer relational databases analysis.

SOFTWARE REQUIREMENT AND SPECIFICATION

External interface requirements: External interface requirement is divided into 3 types, they are:-

- User Interface
- Hardware interface requirements
- Software interface requirements

User Interface: The user interface of material purchase and scheduling system is visual basic forms. It is a graphical user interface consisting of buttons and menus. Functional requirements:

It specifies which output should be produced from the given inputs. They describe the relationship between the input and output of the system.

All the operation to be performed on the input data to obtain output should be specified. An important part of the specification is the system behavior in the abnormal situation like invalid input or error during the computation. The functional requirements must clearly state what the system should do if such situation occur. This includes specifying the validity checks on the input.

Performance requirement: This part of an SRS specifies the performance constraints on the software system. All the requirements relating to the system must be clearly specified. There are 2 types of performance requirements:

- **Static requirements:** Static requirements are those that do not impose constraint on the execution characteristics of the system. These include requirements like the number of terminals to be supported the number of simultaneous users to be supported and the number of files that the system has to process and their sizes.
- **Dynamic requirements**: Dynamic requirements specify constraints on the execution behavior of the system. These typically include response time and throughput constraints on the system.

Design constraints: There are number of factors in client's environment that may restrict the choice of a designer. Such factors include standards that must be followed, resource limits, operating environment, and reliability and security requirements.

Standard compliances: This specifies the requirements for the standard the system must follow. The standard may include the report format and accounting procedure.

Hardware limitations: It can include the type of machines to be used, operating system available on the system language supported and limits on primary and secondary storage.

Reliability and fault tolerance: Reliability requirements are very important for critical applications. Fault tolerance requirements can place a major constraint on how the system is to be designed. It often makes system more complex and expensive.

Security: Security requirements are particularly significant in much database system. Security requirements place restrictions on the use of certain commands, control access to data, provide different kinds of access requirements for different people require the use of password. In this software only an authorized user can have access to our system. They provide with password and id, so that security can be provided.

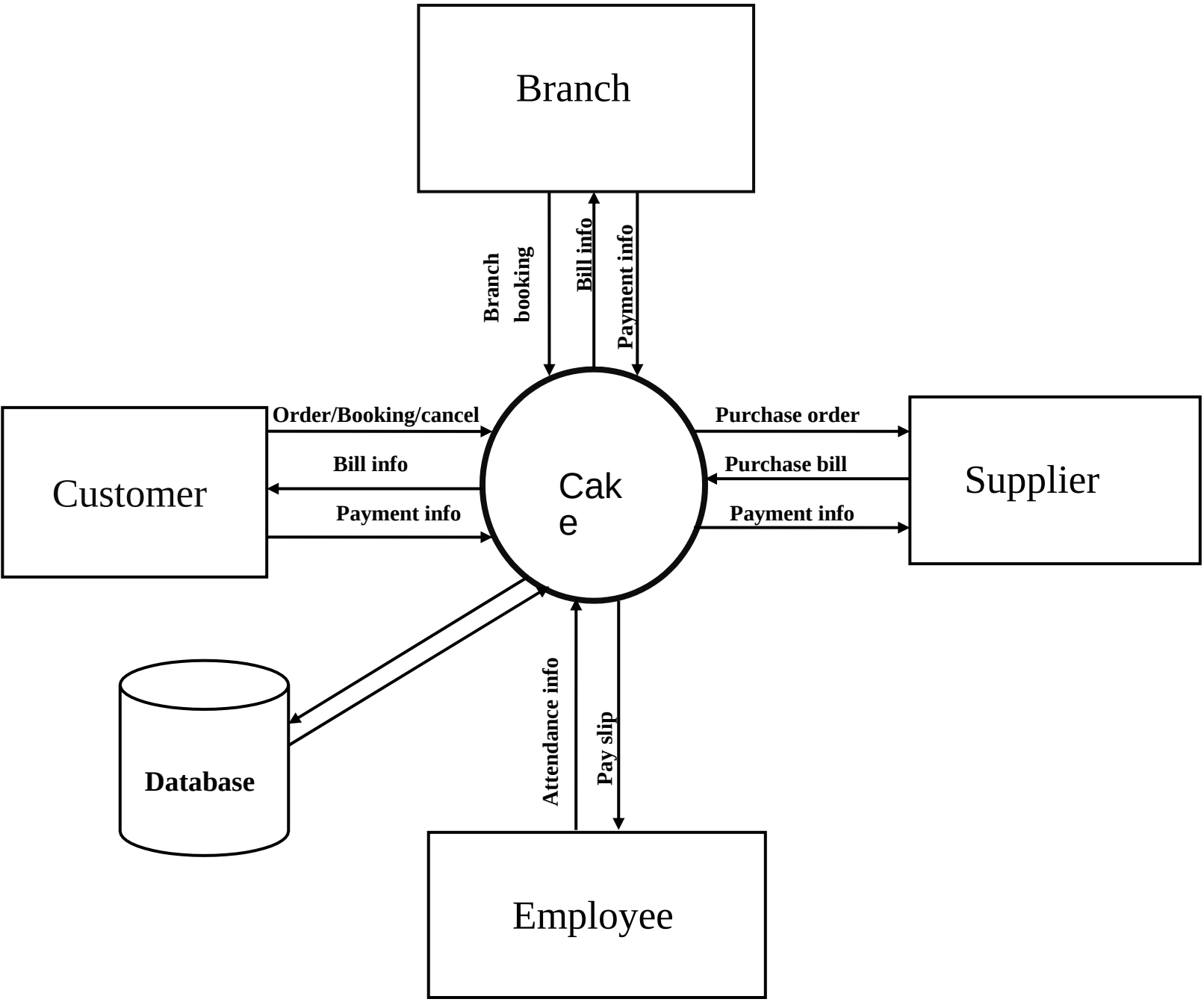
External internal requirements: All the interaction of the software with the people, hardware and other software should be satisfied. If the software is to execute on existing hardware; all the characteristic of the hardware and memory restrictions should be specified.

ANALYSIS

CONTEXT FLOW DIAGRAM (CFD):

The context switch diagram for construction software is as shown in the below figure. The input of this section is shown in this diagram. However, number of details about the function of the construction software is given here. Using this as starting point, login DFD is shown in the below figure.

The environment in which (the context) the software id is used is depicted in this picture. The CFD shows the external entity acting on the software.


Branch
Branch booking
Bill info
Payment info
Customer
Order/Booking/cancel
Bill info
Payment info
Cake
Purchase order
Purchase bill
Payment info
Supplier
Database
Attendance info
Pay slip
Employee

Data Flow Diagram (DFD)

The data flow diagram describes the flow of data with the help of various levels of crystal clear way.

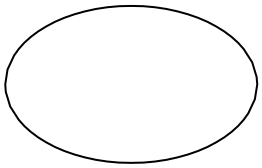
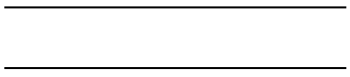
The DFD serves two purposes:

- To provide an indication of flow, data are transferred as they move through the system
- To depict the function that performs the data flow

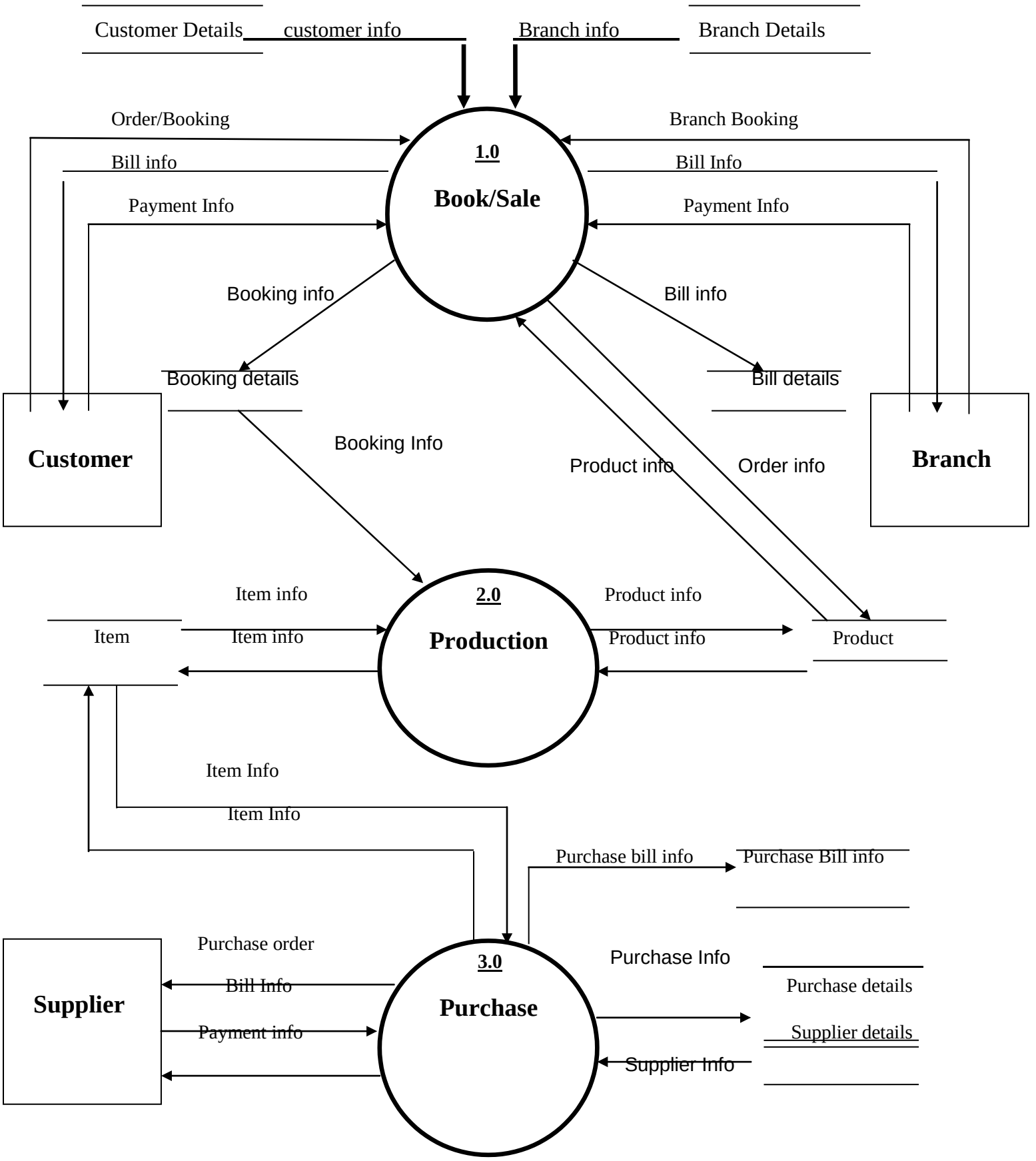
It provides additional information that is used during the analysis of the information domain and severs as the basic for modeling functions. The DFD is also knows as data flow diagram or bubble chart.

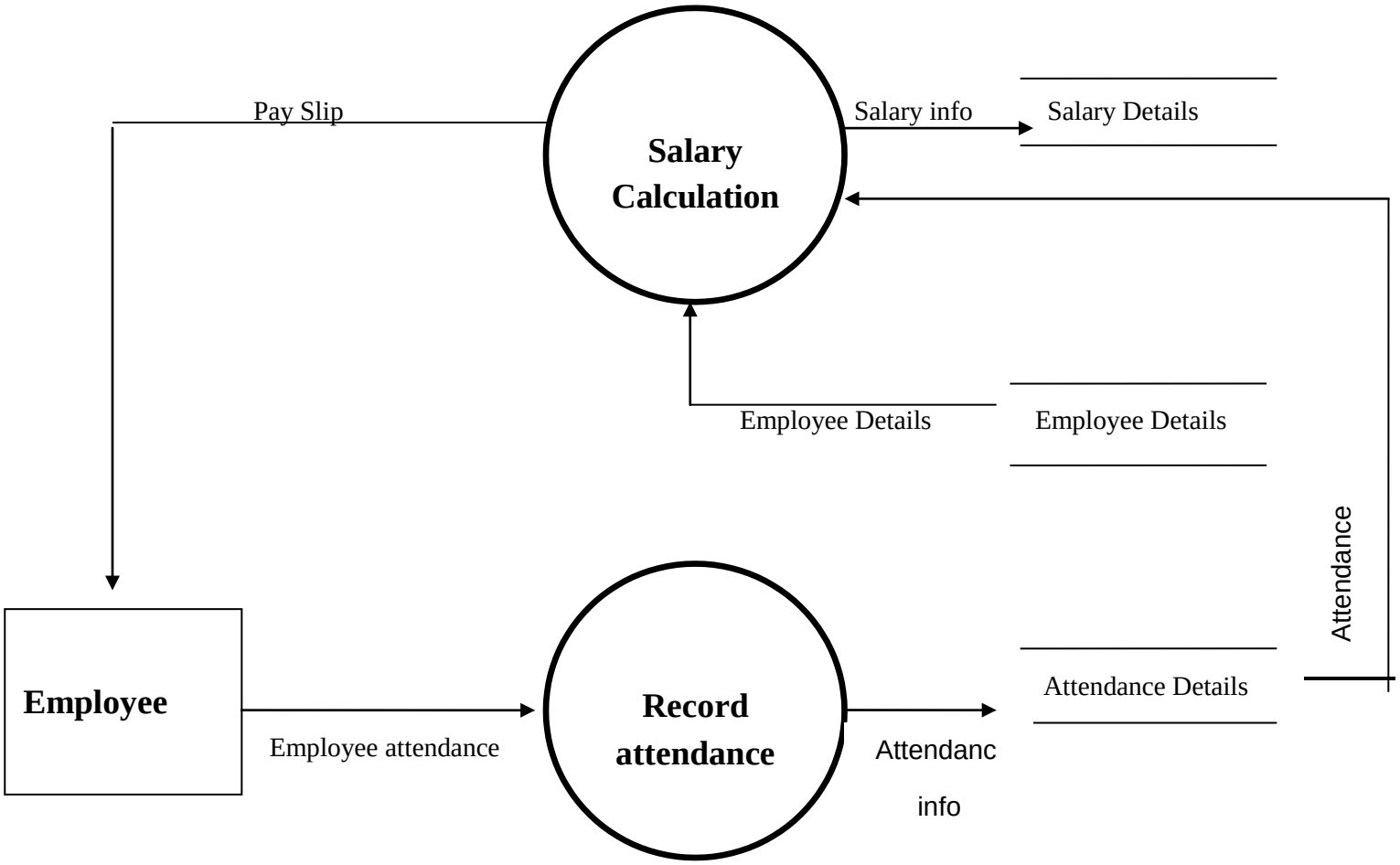
The DFD may be used to represent a system of software of any level of abstraction. In fact, DFD may be partitioned into level that represents increasing information flow and the function detail.

DFD Notation:

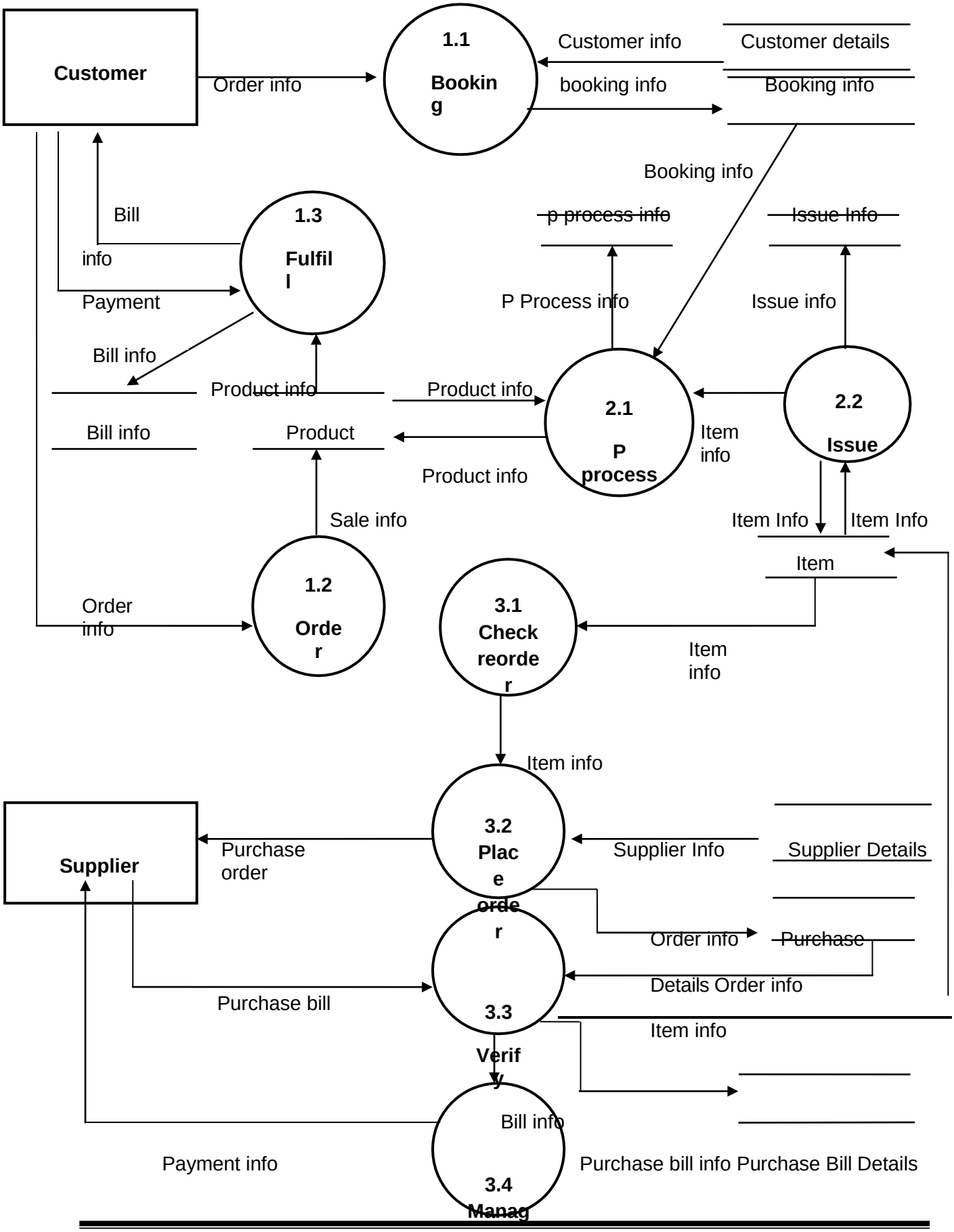
Notation	Description
	**Process** A circle represents process of transform that applied to data or control and changes it in same way.
	**Source\sink** A rectangle is used to represent an external entity that is system that produces information.
	Data flow An arrow represents one or more data items or data object.
	**File or Database** The open box represents data share, shared information.

LEVEL 1 DFD


Customer Details
customer info
Branch info
Branch Details
Order/Booking
Branch Booking
1.0
Book/Sale
Bill info
Bill Info
Payment Info
Payment Info
Booking info
Bill info
Booking details
Bill details
Customer
Booking Info
Product info
Order info
Branch
Item info
2.0
Product info
Item
Item info
Production
Product info
Product
Item Info
Item Info
Purchase bill info
Purchase Bill info
Purchase order
3.0
Purchase Info
Supplier
Bill Info
Purchase
Purchase details
Payment info
Supplier details
Supplier Info


Pay Slip
Salary Calculation
Salary info
Salary Details
Employee Details
Employee Details
Attendance
Employee
Employee attendance
Record attendance
Attendanc info
Attendance Details

Level 2 DFD


Customer
Order info
1.1
Booking
Customer info
booking info
Customer details
Booking info
Booking info
Bill
info
1.3
Fulfill
Payment
Bill info
p process info
P Process info
Issue Info
Issue info
Product info
Product info
Bill info
Product
2.1
P
process
Item
info
Product info
2.2
Issue
Sale info
Item Info
Item Info
Item
Order
info
1.2
Order
3.1
Check
reorder
Item
info
Item info
Supplier
Purchase
order
3.2
Place
order
Supplier Info
Supplier Details
Order info
Purchase
Details Order info
Purchase bill
3.3
Verify
Item info
Bill info
Payment info
Purchase bill info Purchase Bill Details
3.4
Manag

ER-Diagram

An entity relationship model is popular high-level conceptual model.

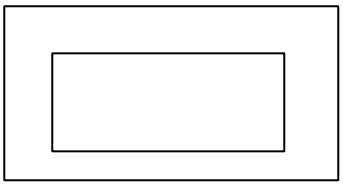
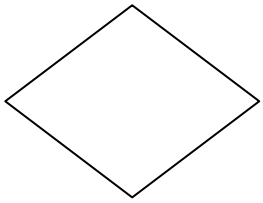
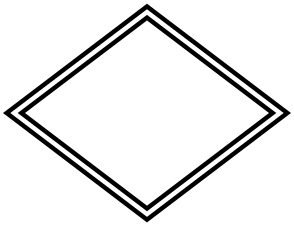
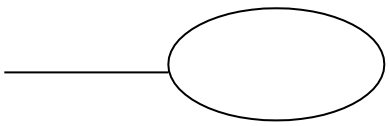
This model and variation are frequently used for the conceptual design of database application and many database design tools employee its concepts. We describe the basic structuring concepts and constraints of ER model and discuss their use in the design of conceptual scheme for database application.We also present the diagrammatic notation known as ER diagram.

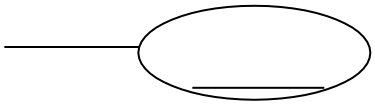
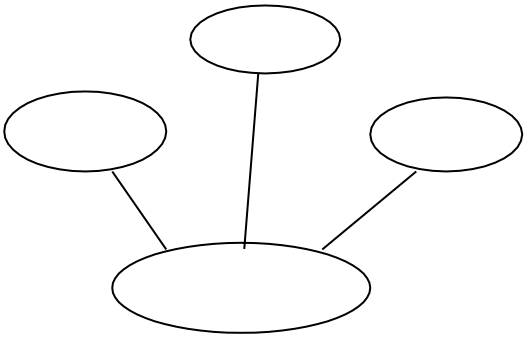
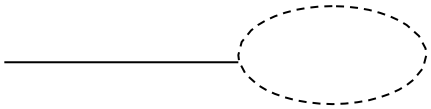
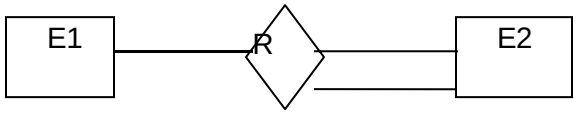
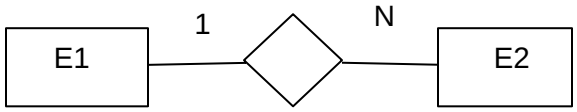
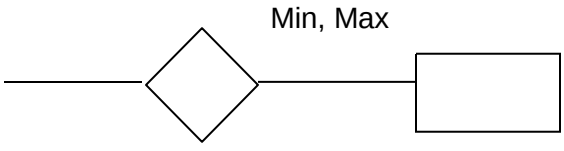
The main focus of ER modeling is data items in the system and relationship between them. Its main aim is to create an ER model for the data and user perspective. The sentence can be used during the development of the database and there are methods that use ER model to design the database for different database modules are frequently representing as ER diagram through the model.

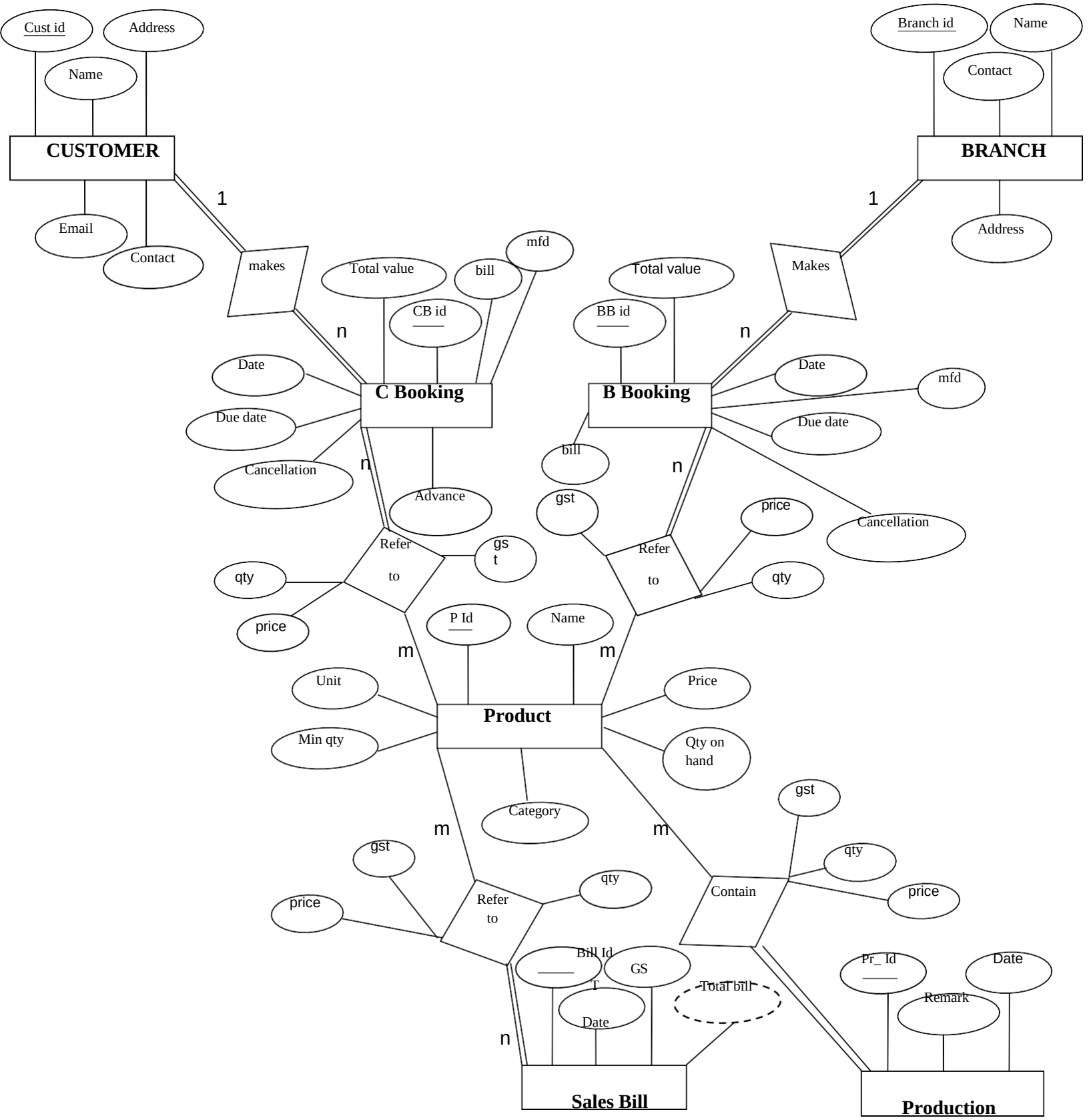
Can also be represented in mathematical form an entity types defines the collection of entities that have same attributes. Each entity types in database are described by its name and attributes. The collection of all entities of a particular entity types in the database at any point in the type is called as entity set. An entity describes the scheme or intention for the set of entities that share same structure.

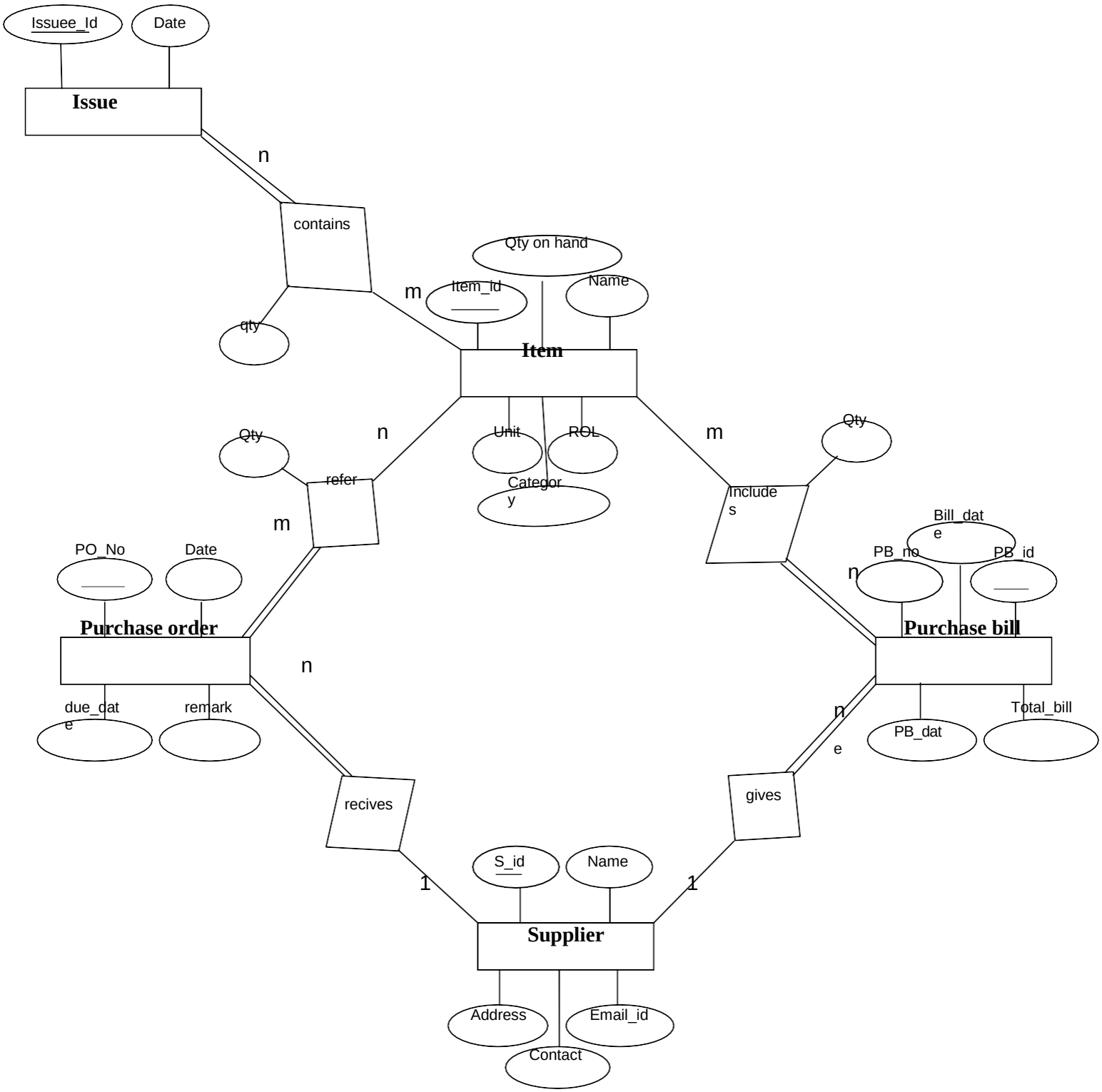
The collection of entities of particular entity type is grouped into an entity type. An important constraint on the entities of an entity type is the key constraints on the attributes. An entity is usually has an attribute is called key attribute and its value can be used to identify each entity uniquely.

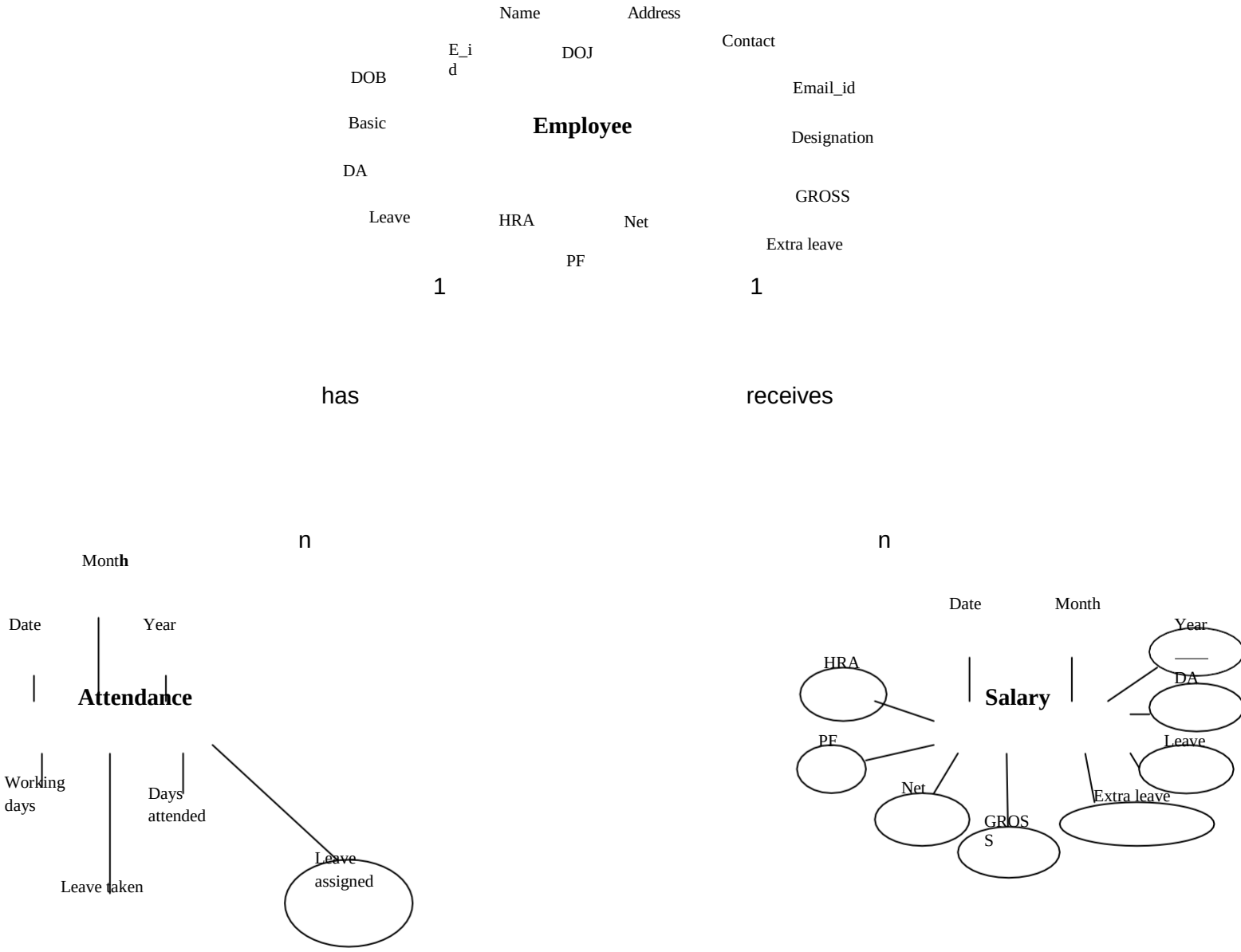
Basic ER Notation (Symbols):

Notations	Description
	Entity
	Weak Entity
	Relationship
	Identifying Attribute
	Attribute

	Key Attribute
	Composite
	Derived Attribute
	Total Participation of E2 in R
	Co-ordinality ratio 1:N for E2:E2 in R
	Structural constraint (min,max) on participation of E in R


Cust id
Address
Name
CUSTOMER
Email
Contact
1
makes
n
Total value
bill
mfd
CB id
C Booking
Date
Due date
Cancellation
n
Advance
Refer
to
gst
qty
price
m
P Id
Name
Unit
Min qty
Product
Price
Qty on hand
Category
m
gst
qty
price
Refer
to
Bill Id
GS
T
Date
n
Sales Bill
Total value
BB id
B Booking
n
Date
mfd
Due date
bill
n
gst
price
Refer
to
qty
Cancellation
m
gst
qty
price
Contain
Pr_ Id
Date
Remark
Production
Branch id
Name
Contact
BRANCH
Address
1
Makes
n


Issuee_Id
Date
Issue
n
contains
qty
m
Item_Id
Qty on hand
Name
Item
Qty
n
refer
m
Unit
ROL
Categor
y
m
Include
s
Qty
PO_No
Date
Purchase order
n
due_dat
e
remark
recives
1
S_id
Name
Supplier
Address
Email_id
Contact
1
gives
n
e
Bill_dat
e
PB_no
PB_id
n
Purchase bill
PB_dat
Total_bill


Name
Address
Contact
E_id
DOJ
DOB
Email_id
Basic
Employee
Designation
DA
GROSS
Leave
HRA
Net
Extra leave
PF
1
1
has
receives
n
n
Month
Date
Year
Attendance
Working days
Days attended
Leave taken
Leave assigned
Date
Month
Year
HRA
DA
Salary
PF
Leave
Net
Extra leave
GROSS

DATABASE TABLES

Customer details		
Fields	Type	constraints
Cust-id	Nvarchar(50)	Primary key
Name	Nvarchar(50)	Not null
Address	Nvarchar(50)	Not null
Email	Nvarchar(50)	Allow null
Contact	Bigint	Not null

Branch details		
bb-id	Nvarchar(50)	Primary key
Name	Nvarchar(50)	Not null
Contact	Bigint	Not null
Address	Nvarchar(50)	Not null

Customer Booking		
Fields	Types	constraints
Cb-id	Nvarchar(50)	Primary key
Date	Nvarchar(50)	Not null
Duedate	Nvarchsr(50)	Not null
totalvalue	Bigint	Not null
Cust-id	Nvarchar(50)	Foreign Key

Product		
Fields	Types	Constraints
Pid	Nvarchar(50)	Primary key
Name	Nvarchar(50)	Not null
Minqty	Smallint	Not null
Qtyonhand	Smallint	Not null
Cb-id	Nvarchar(50)	Foreign Key

C –booking Details		
Fields	Types	constraints
Cb –id	Nvarchar(50)	Primary key
pid	Nvarchar(50)	Primary key
Qty	smallint	Not null

B-book details		
Fields	types	constraints
Bb-id	Nvarchar(50)	Primary key
pid	Nvarchar(50)	Primary key
Qty	smallint	Not null

Sales bill		
Fields	Types	Constraints
Billid	Nvarchar(50)	Primary key
Date	Nvarchar(50)	Not null
Gst	Decimal	Not null
Total bill	double	Not null

Bill details		
Fields	Type	Constraints
Billid	Nvarchar(50)	Primary key
Pid	Nvarchar(50)	Primary key
Qty	smallint	Not null

Production		
Fields	Type	Constraints
Prid	Nvarchar(50)	Primary key
Remark	Nvarchar(50)	Allow null
Date	Nvarchar(50)	Not null

Issue		
Fields	Type	Constraints
issueid	Nvarchar(50)	Primary key
Date	Nvarchar(50)	Not null

Item		
Fields	Type	Constraints
Item-id	Nvarchar(50)	Primary key
Name	Nvarchar(50)	Not null
qtyonhand	smallint	Not null

Issue details		
Fields	Type	constraints
Issueid	Nvarchar(50)	Primary key
Item- id	Nvarchar(50)	Primary key
qty	smallint	Not null

Purchase Order		
Field	Type	Constraints
Pono	Nvarchar(50)	Primary key
Date	Nvarchar(50)	Not null
duedate	Nvarchar(50)	Not null
Remark	Nvarchar(50)	Allow null
Sid	Nvarchar(50)	Foreign Key

Purchase order details		
Fields	Type	Constraints
itemid	Nvarchar(50)	Primary key
Pono	Nvarchar(50)	Primary key
Qty	Smllint	Not null

Supplier		
Fields	Type	Constraints
sid	Nvarchar(50)	Primary key
name	Nvarchar(50)	Not null
Address	Nvarchar(50)	Not null
Email-id	Nvarchar(50)	Not null
Contact	Double	Not null

Purchase bill		
Fields	Type	Constraints
Pb-id	Nvarchar(50)	Primary key
Pb-no	Nvarchar(50)	Not null
Pb-date	Nvarchar(50)	Not null
Billdate	Nvarchar(50)	Not null
total	Double	Not null
Sid	Nvarchar(50)	Foreign key

Purchase bill details		
Fields	Type	Constraints
Pb-id	Nvarchar(50)	Primary key
Item-id	Nvarchar(50)	Primary key
Qty	Smllint	Not null

Menu		
Fields	Type	Constraints
P-id	Nvarchar(50)	Primary key
name	Nvarchar(50)	Not null
category	Nvarchar(50)	Not null
price	double	Not null

Employee		
Fields	Type	constraints
empid	Nvarchar(50)	Primary key
empname	Nvarchar(50)	Not null
dob	Nvarchar(50)	Not null
contact	Bigint	Not null
address	Nvarchar(50)	Not null
Email	Nvarchar(50)	Not null
doj	Nvarchar(50)	Not null
Designation	Nvarchar(50)	Not null
Basic	Double	Not null

Da	float	Not null
hra	float	Not null
Pf	float	Not null
tax	float	Not null
status	Nvarchar(50)	Not null

Attendence		
Fields	Type	constraints
Date	Nvarchar(50)	Not null
month	Nvarchar(50)	Primary key
year	Nvarchar(50)	Primary key
Total days	smallint	Not null
Leave assigned	Small int	Not null
Leave taken	Small int	Not null
Working days	Small int	Not null
Emp-id	Nvarchar(50)	Foreign key

DATA DICTIONARY

The data dictionary is repository of various data flows defined in DFD. The hodated data dictionary states precisely states the structure of each data flow in the DFD**.**

Customer Information:

Customer=@custid@custname@custadd@contact@email
Custid = {A-Z|a-z|0-9|}
Custname = {A-Z | a-
z} contact={0-9}
Custadd ={legal characters}
Email={A-Z|a-z|0-9|+@+.}

Customer Booking:

Booking=@cbid,@duedate,@date,@total,@advance
Cbid= Custid = {A-Z|a-z|0-9|}
Duedate=date
Date=date
Total=no(0-9)
Advance=no(0-9)

Branch Information:

Branch =@bbid@bname@add@phno@email
bbid = {A-Z|a-z|0-9|}
bname= {A-Z | a-
z} Phono={0-9}
add ={legal characters}
Email={A-Z|a-z|0-9|+@+.}

Branch booking :

Booking=@bid,@duedate,@date,@total,@advance
bbid= Custid = {A-Z|a-z|0-9|}
Duedate=date
Date=date
Total=no(0-9)
Advance=no(0-9)

Purchase order:

Purchase order=@orderno,@supid,@orderdate,@duedate Order
No= no {0-9}
Supid= no{0-9}
Order Date= {date}
Due Date= {date}

Purchase Bill:

Purchase Bill=@pbid,@billno,@orderno,@pbdate,@billdate,@gtotal
Pbid= no (0-9)
Bill No= no (0-9)
Order No= no (0-9)
Pd Date= {date}
Bill Date= {date}
Total= no (0-9)

Supplier information:

Supplier=@supid,@supname,@supadd,@phno,@email
Supid= no {0-9}
Supname= {A-Z | a-z}
Supdd = {legal characters}
Phno= no {0-9}
Email={A-Z|a-z|0-9|+@+.}

Product Details:

product Details:=@category,@pno,@name,@price,
Category= {legal characters}
pno= no {0-9}
name= {A-Z | a-
z} Price= no {0-
9}

Item Details:

Item Details:=@itemno,@itemname,@price
Item No= no {0-9}
Item Name= {A-Z | a-z}
Price= no {0-9}

Employee Details:

EmployeeDetails: @empid,@empname,@empadd,@phno,@dob,@doj,
@dept,@salary,@status
Empid= no {0-9}
Empname= {A-Z | a-z}
Empadd= {A-Z | a-z}
Phno= no {0-9}
Date of Birth= {date}
Date of Join= {date}
Department= {A-Z | a-z}
Salary= no {0-9} Status=
{A-Z | a-z}

Attendence Details:

Attendence Details=@empid,@month,@year,@totdays,@workdays,
@leavetaken @leaveassigned
Empid= no (0-9)
Month= {A-Z | a-
z} Year= no (0-9)
Total Days= no (0-9)
Work Days= no (0-9)
Leave taken= no (0-9)
Leave assigned=no(0-9)

Employee Salary:

Employee Salary=@empid,@paydate,@month,@year,@da,@hra,
@gross,@pf,@totalpay
Empid= no (0-9)
Pay Date= {date}
Month= {A-Z | a-
z} Year= no (0-9)
DA= no (0-9)
HRA= no (0-9)
Gross= no (0-9)
PF= no (0-9)
Total Pay= no (0-9)

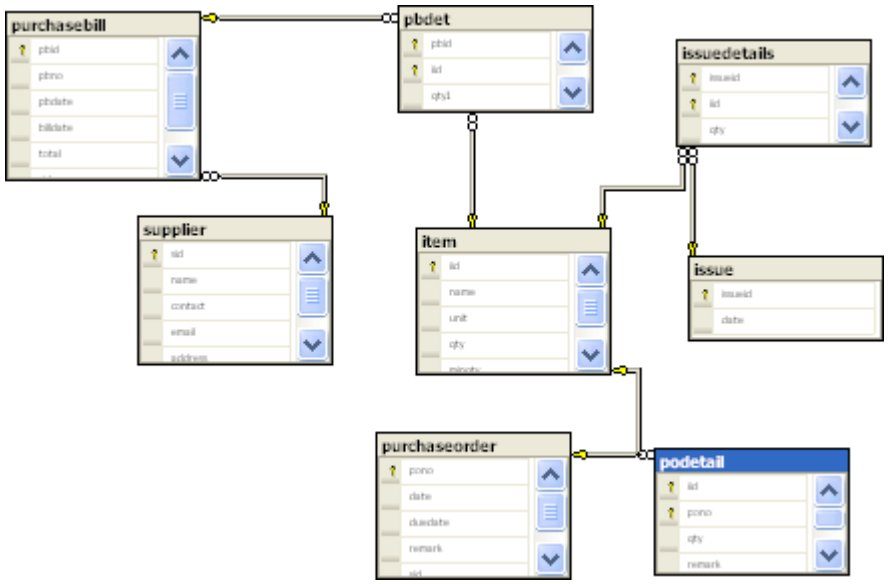
Log:

Log=@username,@password
User Name= {A-Z | a-z}
Password= {A-Z | a-z}

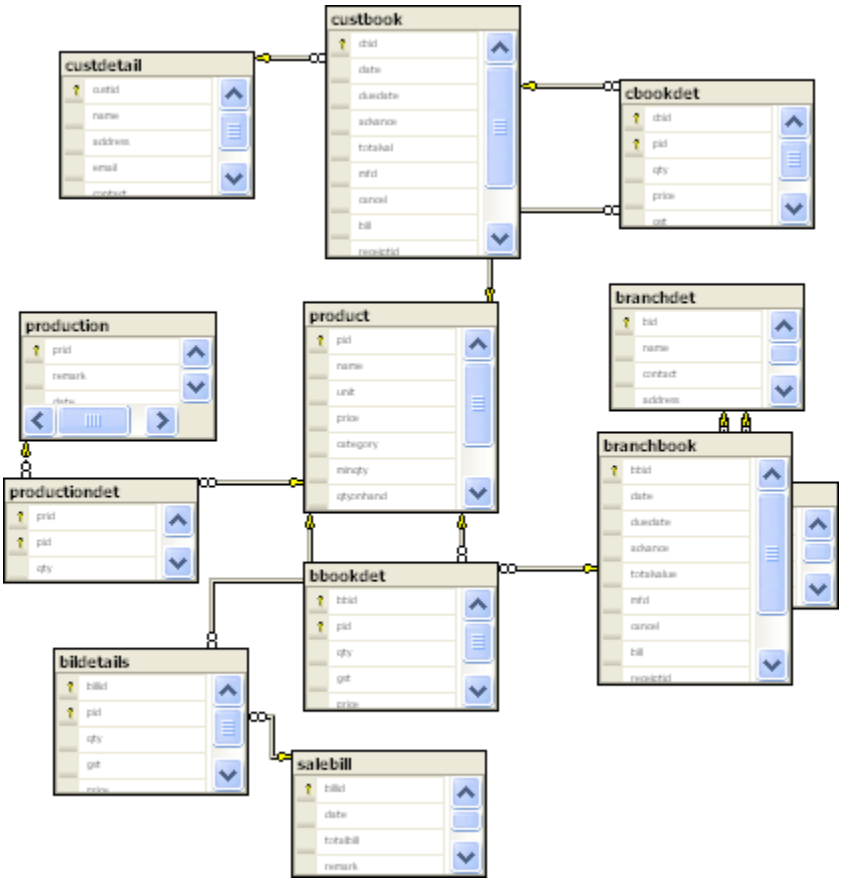
DATA BASE DESIGN

A database is inherent collection of data with some inherent meaning design, built populated with data for specific purpose the following guidelines have been followed during the database designs.

- Description names for the tables, columns and index.
- Distinct name for tables and columns.
- Proper data type for each column.




salary
emp
attendance


custbook
custdetail
cbookdet
production
product
branchdet
branchbook
productiondet
bbookdet
bildetails
salebill

CODe:

LOGIN PAGE:

Source Code:

```
<%@ Page Language="C#" AutoEventWireup="true"
MaintainScrollPositionOnPostback ="true" CodeFile="login2.aspx.cs"
Inherits="login2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
    <link href="style1.css" rel ="stylesheet" type ="text/css" /><link />
<script language="javascript" type="text/javascript">
// <!CDATA[

// ]]>
</script>
</head>
<body>
    <form id="form1" runat="server" defaultfocus="txtuser">
    <div class ="box" style="height: 262px; left: 30%; top: 25%;">
    <img alt ="not found"  class="userimg" src="download.jpeg" style="left:
185px; top: -66px" />
    <h2>
        <span style="color: #33cccc"><span style="font-size: 24pt; font-
family: Chiller; align:center">GET
            STARTED</span> </span></h2>

    <div class ="inputbox">
         <asp:TextBox ID="txtuser" placeholder="Enter the username"
runat="server" Width="335px"></asp:TextBox>
    <%--<input type ="text" name=" " placeholder="Enter username" id="Text1"
onclick="return Text1_onclick()" />--%>
    <label style="left: 2px; top: -38px" >
        <strong>User Name</strong></label>

        <br /> <br />
    </div>
    <div class ="inputbox">
    <label style="left: 0px; top: -17px" >
        <strong>Password</strong></label><br />
        <asp:TextBox ID="txtpass" placeholder="Enter the password"
runat="server" TextMode="Password" Width="343px"></asp:TextBox>
   <%-- <input type ="password" name=" " id="Password1" onclick="return
Password1_onclick()" />--%>


 <br />
```

```
        <asp:CheckBox ID="CheckBox1" runat="server" AutoPostBack="True"
OnCheckedChanged="CheckBox1_CheckedChanged"
            Style="left: 350px; position: relative; top: -33px" Width="56px"
/><br />
     </div>
        <asp:Label ID="Label1" runat="server" Font-Bold="True"
ForeColor="#C00000" Width="235px"></asp:Label><br />

   <asp:Button ID="btnlogin" runat="server" Text="LOGIN" style="left: -5px;
position: relative; top: 7px" OnClick="btnlogin_Click1" Font-Names="Chiller"
Font-Size="16pt" Width="349px" />
        <br />
        <br />
        <asp:LinkButton ID="lnkreg" runat="server" Font-Bold="True" Font-
Italic="False" Font-Size="1.3em" ForeColor="#33CC99"
OnClick="lnkreg_Click">ADMIN Registration</asp:LinkButton><br />
        <asp:LinkButton ID="lnkforgot" runat="server"
OnClick="lnkforgot_Click">forgot password</asp:LinkButton><br />



</div>
    </form>
</body>
</html>
```

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows.Forms;
using System.Data.SqlClient;

public partial class login2 : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        lnkreg.Visible = false;
        lnkforgot.Visible = false;
        c = new connect();
        ds = new DataSet();
```

```
        c.cmd.CommandText = "select * from login where username='" +
        "admin"   + "'";
        ds = new DataSet();
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "log");
        if (ds.Tables["log"].Rows.Count > 0)
        {
              if (IsPostBack)
              {
                    String password = txtpass.Text;
                    txtpass.Attributes.Add("value", password);
              }
        }
        else
        {
              lnkreg.Visible = true;
        }
        if (txtuser.Text == "admin")
        {
              lnkforgot.Visible = true;
        }
  }
  protected void btnlogin_Click1(object sender, EventArgs e)
  {
        if (txtuser.Text == "admin")
        {
              lnkforgot.Visible = true;
        }
        try
        {
              c = new connect();
              String user = txtuser.Text;
              String pass = txtpass.Text;
              Session["usr"] = user;
              Session["pass"] = pass;
              if (user != "")
              {
                    c.cmd.CommandText = "select * from login where
                    username='" + user + "'";
                    ds = new DataSet();
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "log");
                    if (ds.Tables["log"].Rows.Count > 0)
                    {
                    if(ds.Tables["log"].Rows[0].ItemArray[1].ToString()
                    == pass)
                    {
                          Response.Redirect("~/Default2.aspx");
                    }
                    else
                    {
                          Label1.Text = "INCORRECT PASSWORD";
                          txtpass.Attributes["value"] = "";
                    }
              }
              else
```

```
                    {
                        Label1.Text = "INCORRECT USERNAME";
                        txtpass.Attributes["value"] = "";
                    }
                }
                else
                {
                    Label1.Text = "ENTER THE FIELDS";
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
            }
    }
    protected void CheckBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (CheckBox1.Checked == true)
        {
            txtpass.TextMode = TextBoxMode.SingleLine;
        }
        else
        {
            txtpass.TextMode = TextBoxMode.Password;
        }
    }
    protected void lnkreg_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/register.aspx");
    }
    protected void lnkforgot_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/forgot.aspx");
      }
}
```

Change user password

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
public partial class chuser : System.Web.UI.Page
{
```

```
connect c;
DataSet ds = new DataSet();
SqlDataAdapter adp = new SqlDataAdapter();
protected void Page_Load(object sender, EventArgs e)
{
    txtuser.Text = "user";
    if (IsPostBack)
    {
        String password = txtpass.Text;
        txtpass.Attributes.Add("value", password);
        String cnpass = txtcnfrm.Text;
        txtcnfrm.Attributes.Add("value", cnpass);
    }
}
protected void cc_CheckedChanged(object sender, EventArgs e)
{

    if (cc.Checked == true)
    {
        txtpass.TextMode = TextBoxMode.SingleLine;
        txtcnfrm.TextMode = TextBoxMode.SingleLine;

    }
    else
    {
        txtpass.TextMode = TextBoxMode.Password;
        txtcnfrm.TextMode = TextBoxMode.Password;
    }
}
protected void btnsave_Click1(object sender, EventArgs e)
{
    c = new connect();

    if (txtpass.Text != "" && txtuser.Text != "")
    {
        c.cmd.CommandText = "Select * from login where username='" +
        txtuser.Text + "'";
        ds = new DataSet();
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "log");
        if (ds.Tables["log"].Rows.Count > 0)
        {
              if (MessageBox.Show("Are you sure?", "message",
              MessageBoxButtons.YesNo) == DialogResult.Yes)
              {
                    c.cmd.CommandText = "update login set password='" +
                    txtcnfrm.Text + "'where username='admin'";
                    c.cmd.ExecuteNonQuery();
                    MessageBox.Show("password changed");
                    txtpass.Attributes["value"] = "";
                    txtcnfrm.Attributes["value"] = "";
              }
              else
              {
                    txtuser.Text = "";
                    txtpass.Attributes["value"] = "";
```

```
                        txtcnfrm.Attributes["value"] = "";
                    }
                }
                else
                {
                Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "<script>alert('Check the user name')</script>");
                    txtpass.Attributes["value"] = "";
                    txtcnfrm.Attributes["value"] = "";
                }
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all the fields')</script>");
            }
       }
}
```

Change admin Password

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data .SqlClient ;
using System.Windows .Forms ;
public partial class chadmin : System.Web.UI.Page
{
    connect c;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack)
        {
            String password = txtpass.Text;
            txtpass.Attributes.Add("value", password);
            String cnpass = txtcnfrm.Text;
            txtcnfrm.Attributes.Add("value", cnpass);
        }
    }
    protected void btnsave_Click(object sender, EventArgs e)
    {
        c = new connect();
        if (txtpass.Text != "" && txtcnfrm.Text != "")
        {
            if (MessageBox.Show("Are you sure?", "message",
            MessageBoxButtons.YesNo) == DialogResult.Yes)
```

```
            {
                c.cmd.CommandText = "update login set password='" +
                txtcnfrm.Text + "'where username='admin'";
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("password changed");
                txtpass.Attributes["value"] = "";
                txtcnfrm.Attributes["value"] = "";
            }
            else
            {
                txtpass.Attributes["value"] = "";
                txtcnfrm.Attributes["value"] = "";
            }
        }
        else
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
            "<script>alert('Enter all fields')</script>");

        }
    }
    protected void cc_CheckedChanged(object sender, EventArgs e)
    {

        if (cc.Checked == true)
        {
            txtpass.TextMode = TextBoxMode.SingleLine;
            txtcnfrm.TextMode = TextBoxMode.SingleLine;
        }
        else
        {
            txtpass.TextMode = TextBoxMode.Password;
            txtcnfrm.TextMode = TextBoxMode.Password;
        }
    }

}
```

Branch Booking

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
public partial class branchadd : System.Web.UI.Page
{
    connect c;
```

```
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();
    int count;
    protected void Page_Load(object sender, EventArgs e)
    {
        DropDownList2.Items.Add("---Select---");
        lbldate1.Text = DateTime.Today.ToString("dd'/'MM'/'yyyy");
        Calendar1.Visible = false;
        if (!IsPostBack)
        {
            GenerateID();
        }
        c = new connect();
        ds = new DataSet();
        if (DropDownList1.Items.Count == 0)
        {
            c.cmd.CommandText = "SELECT DISTINCT [category] FROM [product]";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cat");
            if (ds.Tables["cat"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                  {
                  DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[
                  0].ToString());
                  }

            }
        }
        if (DropDownList3.Items.Count == 0)
        {
            c.cmd.CommandText = "SELECT [bid] FROM [branchdet]";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "bran");
            if (ds.Tables["bran"].Rows.Count > 0)
            {
                DropDownList3.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["bran"].Rows.Count; i++)
                {

DropDownList3.Items.Add(ds.Tables["bran"].Rows[i].ItemArray[0].ToString());
                }
            }
        }
        if (DropDownList2.Items.Count == 0)
        {
            c.cmd.CommandText = "SELECT [name] FROM [product] WHERE
            [category] = '"+DropDownList1 .SelectedItem .Text +"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "pro");
            if (ds.Tables["pro"].Rows.Count > 0)
            {
```

```
            DropDownList2.Items.Add("---Select---");
            int i;
            for (i = 0; i < ds.Tables["pro"].Rows.Count; i++)
              {
        DropDownList2.Items.Add(ds.Tables["pro"].Rows[i].ItemArray[0].ToS
        tring());
            }
        }
    }
}
private void GenerateID()
{
    String book = "BB";
    c = new connect();
    c.cmd.CommandText = "select count(bbid) from branchbook";
    int i = Convert.ToInt32(c.cmd.ExecuteScalar());
    i = i + 1001;
    lblid.Text = book + i.ToString();
}
protected void btnclear_Click(object sender, EventArgs e)
{
    Response.Redirect(Request.Url.AbsoluteUri);
    GenerateID();
}
protected void btnsave_Click(object sender, EventArgs e)
{
        if (DropDownList3.SelectedItem.Text == "" || txtduedate.Text ==
        "" || txtadvance.Text == "" || txttotal.Text == "")
        {
              Page.ClientScript.RegisterStartupScript(this.GetType(),
              "alert", "<script>alert('Enter the fields')</script>");
        }
        else
        {
              Double p = Convert.ToDouble(txttotal.Text);
              Double a = Convert.ToDouble(txtadvance.Text);
              if (a > p)
              {
        Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
        "<script>alert('Advance must be lesser then total
        value')</script>");

                    txtadvance.Focus ();
              }
              else
              {
                    try
                    {
                          c = new connect();
                          String mfd = "No", cancel = "No", bill = "No",
                          rid ="No";
                          c.cmd.CommandText = "insert into branchbook
                          values(@bbid,@date,@duedate,@advance,@totalvalu
                          e,@mfd,@cancel,@bill,@receiptid,@bid)";
                          c.cmd.Parameters.Add("@bbid",
                          SqlDbType.NVarChar).Value = lblid.Text;
```

```
                c.cmd.Parameters.Add("@date",SqlDbType.NVarChar
                ).Value = lbldate1.Text;
                c.cmd.Parameters.Add("@duedate",SqlDbType.NVarC
                har).Value = txtduedate.Text;
                c.cmd.Parameters.Add("@advance",SqlDbType.BigIn
                t).Value = Convert.ToInt64(txtadvance.Text);
                c.cmd.Parameters.Add("@totalvalue",
                SqlDbType.Decimal ).Value = Convert.ToDecimal
                (txttotal.Text);
                c.cmd.Parameters.Add("@mfd",SqlDbType.NVarChar)
                .Value = mfd;
                c.cmd.Parameters.Add("@cancel",
                SqlDbType.NVarChar).Value = cancel;
                c.cmd.Parameters.Add("@bill",
                SqlDbType.NVarChar).Value = bill;
                c.cmd.Parameters.Add("@receiptid",
                SqlDbType.NVarChar).Value = rid;
                c.cmd.Parameters.Add("@bid",
                SqlDbType.NVarChar).Value =
                DropDownList3.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
          }
          catch (Exception)
          {
                throw;
          }
          finally
          {
                c.cnn.Close();
          }
          try
          {
                Double gst = 3;
                c = new connect();
                for (int i = 0; i < GridView1.Rows.Count; i++)
                {
                      c.cmd.CommandText = "insert into
                      bbookdet(bbid,pid,qty,gst,price)
                      values('" + lblid.Text + "','" +
                      GridView1.Rows[i].Cells[0].Text + "','"
                      +Convert .ToInt64 (
                      GridView1.Rows[i].Cells[2].Text )+
                "','"+gst .ToString ()+"','" +Convert
                .ToDecimal ( GridView1.Rows[i].Cells[3].Text) +
                "')";
          c.cmd.ExecuteNonQuery();
    }
}
catch (Exception)
{
    throw;
}
finally
{
    c.cnn.Close();
}
```

```
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Order Submitted')</script>");
                Session["i1"] = lblid.Text;
                Response.Redirect("~/branchrecp.aspx");
                Response.Redirect(Request.Url.AbsoluteUri);

                GenerateID();
            }
        }
    }
protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
        try
        {
            c = new connect();
            ds = new DataSet();
            if (IsPostBack)
            {
                c.cmd.CommandText = "select * from product where name='" +
DropDownList2.SelectedItem.Text+ "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                                    txtpid.Text =
                  Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[0]);
                    c.cmd.ExecuteNonQuery();
                }
            }
        }
        catch (Exception)
        {
            throw
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void btnlist_Click(object sender, EventArgs e)
    {
        int n = txtqty.Text.Length;
        n--;
        Regex qty = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtpid .Text =="" || txtqty.Text == "")
        {
          Page .ClientScript .RegisterStartupScript (this.GetType
(),"alert","<script>alert('Enter the fields')</script>");

        }
        else if (qty.IsMatch(txtqty.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

        }
```

```
    else
    {
        Double t;
        DataTable dt = new DataTable();
        dt.Columns.Add("Product ID");
        dt.Columns.Add("Name");
        dt.Columns.Add("Quantity");
        dt.Columns.Add("Price");
        DataRow dr = null;
        if (ViewState["pro"] != null)
        {
            for (int i = 0; i < 1; i++)
            {
                dt = (DataTable)ViewState["pro"];
                if (dt.Rows.Count > 0)
                {
                    dr = dt.NewRow();
                    dr["Product ID"] = txtpid.Text;
                    dr["Name"] = DropDownList2.Text;
                    dr["Quantity"] = txtqty.Text;
                    dr["Price"] = txtprice.Text;
                    dt.Rows.Add(dr);
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                    t = Convert.ToDouble(txttotal.Text);
                    Double p = Convert.ToDouble(txtprice.Text);
                    Double s = t + p;
                    txttotal.Text = s.ToString();
                }
            }
        }
        else
        {
            dr = dt.NewRow();
            dr["Product ID"] = txtpid.Text;
            dr["Name"] = DropDownList2.Text;
            dr["Quantity"] = txtqty.Text;
            dr["Price"] = txtprice.Text;
            dt.Rows.Add(dr);
            GridView1.DataSource = dt;
            GridView1.DataBind();
            txttotal.Text = txtprice.Text;
        }
        ViewState["pro"] = dt;
        txtpid.Text = "";
        txtqty.Text = "";
        txtprice.Text = "";
        t = Convert.ToDouble(txttotal.Text);
        DropDownList2.SelectedIndex = 0;
        DropDownList1.SelectedIndex = 0;

    }
}
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
```

```
        if (lblid.Text == "")
        {
            Calendar1.Visible = false;
        }
        else
        {
            if (txtduedate.Text == "")
            {
                Calendar1.Visible = true;
            }
            else
            {
                Calendar1.Visible = false;
            }
        }
        DateTime dt1, dt2;
        dt1 = Convert.ToDateTime(DateTime.Today.ToShortDateString());
        dt2 = Convert.ToDateTime(Calendar1.SelectedDate.ToShortDateString());
        if (dt2 >= dt1)
        {

            txtduedate.TextCalendar1.SelectedDate.ToString("dd'/'MM'/'yyyy");
        }
        else
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Due date must greater or equal to date')</script>");
        }
    }
    protected void DropDownList3_SelectedIndexChanged(object sender,
EventArgs e)
    {
        try
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "select * from branchdet where bid='" +
DropDownList3.SelectedItem.Text + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "branch");
            if (ds.Tables["branch"].Rows.Count > 0)
            {
                txtname.Text =
Convert.ToString(ds.Tables["branch"].Rows[0].ItemArray[1]);
                txtaddress.Text =
Convert.ToString(ds.Tables["branch"].Rows[0].ItemArray[3]);
                txtcontact.Text =
Convert.ToString(ds.Tables["branch"].Rows[0].ItemArray[2]);
                c.cmd.ExecuteNonQuery();
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record Not Found ,Add new branch')</script>");

            }
```

```
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void txtqty_TextChanged(object sender, EventArgs e)
    {
        int n = txtqty.Text.Length;
        n--;
        Regex qty = new Regex("^[1-9][0-9]{" + n + "}");
        if (qty.IsMatch(txtqty.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from product where pid='" +
txtpid.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    Double p =
Convert.ToInt32(ds.Tables["prod"].Rows[0].ItemArray[3]);
                    Double g = 3;
                    Double gst = g * p / 100;
                    Double p1 = gst + p;
                    int q = Convert.ToInt32(txtqty.Text);
                    Double t = p1 * q;
                    txtprice.Text = t.ToString();
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
protected void  DropDownList1_SelectedIndexChanged(object sender, EventArgs
e)
{
```

```
    c = new connect();
        ds = new DataSet();
        DropDownList2.Items.Clear();
        c.cmd.CommandText = "select * from product where category ='" +
DropDownList1.SelectedItem.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "po");
        if (ds.Tables["po"].Rows.Count > 0)
        {
            DropDownList2.Items.Add("---Select---");
            int i;
            for (i = 0; i < ds.Tables["po"].Rows.Count; i++)
            {
            DropDownList2.Items.Add(ds.Tables["po"].Rows[i].ItemArray[1].ToSt
            ring());
            }
        }
      }
    protected void btndate_Click1(object sender, EventArgs e)
    {
        Calendar1.Visible = true;
    }
    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        dt =  (DataTable)ViewState["pro"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        Double pr = Convert.ToDouble(row.Cells[4].Text);
        Double tt = Convert.ToDouble(txttotal.Text);
        Double ttl = tt - pr;
        txttotal.Text = Convert.ToString(ttl);
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (GridView1.Rows.Count <= 0)
        {
            ViewState["pro"] = null;
        }
    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
    {
        count = 1;
        GridView1.EditIndex = e.NewEditIndex;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
```

```
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        dt =  (DataTable)ViewState["pro"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        dt.Rows[row.DataItemIndex]["Quantity"] =
((TextBox)(row.Cells[3].Controls[0])).Text;
        String pid =
Convert.ToString(((TextBox)(row.Cells[1].Controls[0])).Text);
        Double qty =
Convert.ToDouble(((TextBox)(row.Cells[3].Controls[0])).Text);
        Double pr =
Convert.ToDouble(((TextBox)(row.Cells[4].Controls[0])).Text);
        Double tt = Convert.ToDouble(txttotal.Text);
        if (qty < 1000)
        {
            double total = tt - pr;
            try
            {
                Double gst = 3;
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from product where pid='" + pid
+ "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    Double p =
Convert.ToInt32(ds.Tables["prod"].Rows[0].ItemArray[3]);
                    Double a = (p * gst) / 100;
                    //int q = Convert.ToInt32(txtqty.Text);
                    Double g = a + p;
                    Double t = g * qty;

                  dt.Rows[row.DataItemIndex]["Price"] = t.ToString();
                    txttotal.Text = Convert.ToString(t + total);
                }
            }

            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            GridView1.EditIndex = -1;
            GridView1.DataSource = dt;
            GridView1.DataBind();
            GridView1.Visible = true;
        }
        else
        {
```

```
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('We sale only 1000 items at a time')</script>");
        }
    }
    protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs
e)
    {
        if (count == 1)
        {
            e.Row.Cells[1].Enabled = false;
            e.Row.Cells[2].Enabled = false;
            e.Row.Cells[4].Enabled = false;

        }
    }
}
```

Branch Cancel

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data .SqlClient ;
public partial class branchcancel : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp=new SqlDataAdapter ();
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            c = new connect();
            ds = new DataSet();
            if (DropDownList1.Items.Count == 0)
            {
                c.cmd.CommandText = "SELECT distinct [bid] FROM [branchbook]
where cancel='"+"No"+"'and mfd='"+"No"+"' ";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cat");
                if (ds.Tables["cat"].Rows.Count > 0)
                {
                    DropDownList1.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                    {

DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
```

```
                    }

                }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }

    }

    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)




  {
        c = new connect();
        ds = new DataSet();
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList2 .SelectedItem .Text  + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        c = new connect();
        ds = new DataSet();
        Label ppid = GridView1.Rows[e.RowIndex].FindControl("Label2") as
Label;
        c.cmd.CommandText = "delete from bbookdet where pid= '" + ppid.Text +
"'";
        c.cmd.ExecuteNonQuery();
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList2.SelectedItem.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
```

```
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
        else
        {
            String yes="yes";
            GridView1.Visible = false;
            c.cmd.CommandText ="update branchbook set cancel=@cancel where
bbid='"+DropDownList2.SelectedItem .Text+"'" ;
            c.cmd.Parameters.Add("@cancel", SqlDbType.NVarChar).Value =
yes.ToString ();
            c.cmd.ExecuteNonQuery ();
        }
    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
 {
        c = new connect();
        ds = new DataSet();

        GridView1.EditIndex = e.NewEditIndex;
        c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList2.SelectedItem.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        c = new connect();
        ds = new DataSet();
        Label pid = GridView1.Rows[e.RowIndex].FindControl("lblproductid") as
Label;
        TextBox qty = GridView1.Rows[e.RowIndex].FindControl("txtqty") as
TextBox;
        c.cmd.CommandText = "Update bbookdet set qty='" + qty.Text + "'where
pid=  '" + pid.Text + "'";
        c.cmd.ExecuteNonQuery();
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList2.SelectedItem.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
```

```
    }
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            DropDownList2.Items.Clear();
            DropDownList2.Items.Add("---Select---");
 lblmsg.Text = "";
            lblname.Text = "";
            GridView1.Visible = false;
        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                DropDownList2.Items.Clear();
                c.cmd.CommandText = "SELECT * FROM [branchbook] WHERE [bid]
='" + DropDownList1.SelectedItem.Text + "' and mfd='" + "No" + "' and
cancel='"+"No"+"'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "bran");
                if (ds.Tables["bran"].Rows.Count > 0)
                {
                    DropDownList2.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["bran"].Rows.Count; i++)
                    {

DropDownList2.Items.Add(ds.Tables["bran"].Rows[i].ItemArray[0].ToString());
                    }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from branchdet where bid='" +
DropDownList1.SelectedItem.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "branch");
                if (ds.Tables["branch"].Rows.Count > 0)
                {
                    lblname.Text =
Convert.ToString(ds.Tables["branch"].Rows[0].ItemArray[1]);
                    c.cmd.ExecuteNonQuery();
```

```
}
                else
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('already manufactured')</script>");

                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void DropDownList2_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList2.SelectedItem.Text == "---Select---")
        {
            lblmsg.Text = "";
            GridView1.Visible = false;
        }
        else
        {
            try
            {
                c = new connect();
                c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList2.SelectedItem.Text + "'";
                ds = new DataSet();
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cu");
                if (ds.Tables["cu"].Rows.Count > 0)
                {
                    GridView1.DataSource = ds.Tables["cu"];
                    GridView1.DataBind();
                }
                else

{
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('No records')</script>");

                }

            }
            catch (Exception)
            {
                throw;
            }
            finally
```

```
            {
                c.cnn.Close();
            }
        }
    }
}
```

Branch Order Report:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
public partial class branchoder : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        //CrystalReportViewer1.Visible = false;
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [bbid] FROM [branchbook]";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cat");
            if (ds.Tables["cat"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");

                  int i;
                for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                }
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        //CrystalReportViewer1.Visible = true;
        CrystalReportViewer1.SelectionFormula = " {branchbook.bbid}='" +
DropDownList1.SelectedItem.Text + "'";
        CrystalReportViewer1.RefreshReport();
    }
```

```
    protected void Button2_Click(object sender, EventArgs e)
    {
        c = new connect();
        ds = new DataSet();
        c.cmd.CommandText = "select * from branchbook";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "order");
        if (ds.Tables["order"].Rows.Count > 0)
        {
            CrystalReportViewer1.SelectionFormula = "";
         }
    }
}
```

Branch receipt:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class branchrecp : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {




CrystalReportViewer1.SelectionFormula = " {branchbook.bbid}='" +
(String)Session["i1"] + "'";
        CrystalReportViewer1.RefreshReport();
    }
    protected void CrystalReportViewer1_Init(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/branchadd.aspx");
    }
}
```

Branch update

```
using System;
using System.Data;
using System.Configuration;
```

```
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data .SqlClient ;
using System.Windows .Forms ;
using System.Text.RegularExpressions;
public partial class branchupdate : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp=new SqlDataAdapter ();
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            c = new connect();
            c.cmd.CommandText = "select * from branchdet";
            ds = new DataSet();
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "branch");
            if (ds.Tables["branch"].Rows.Count > 0)
            {
                GridView1.DataSource = ds.Tables["branch"];
                GridView1.DataBind();
            }




   else
            {
                MessageBox.Show("No Records");
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void btnclear_Click(object sender, EventArgs e)
    {
        lblid.Text = "";
        txtname.Text = "";
        txtcontact.Text = "";
        txtaddress.Text = "";
    }
    protected void btnsearch_Click(object sender, EventArgs e)
```

```
    {
        if (txtsearch.Text == "")
        {
            lblsearch.Text = "Enter the ID first";
        }
        else
        {
            lblsearch.Text = "";
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from branchdet where bid='" +
txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "bd");
                if (ds.Tables["bd"].Rows.Count > 0)
                {
                    lblmsg.Text = "";
                    lblid.Text =
Convert.ToString(ds.Tables["bd"].Rows[0].ItemArray[0]);
                    txtname.Text =
Convert.ToString(ds.Tables["bd"].Rows[0].ItemArray[1]);
                    txtcontact.Text =
Convert.ToString(ds.Tables["bd"].Rows[0].ItemArray[2]);
                    txtaddress.Text =
Convert.ToString(ds.Tables["bd"].Rows[0].ItemArray[3]);
                    c.cmd.ExecuteNonQuery();

}
                else
                {
                    lblmsg.Text = "Record Not Found";
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnupdate_Click(object sender, EventArgs e)
    {
        Regex con = new Regex("^[6-9][0-9]{9}");
        Regex name = new Regex("^[a-zA-Z]");
        Regex email = new Regex(@"^[a-za-z0-9]+([-+.'][a-za-z0-9]+)*@[a-
z]+([-.][a-z]+)*\.[a-z]+([-.][a-z]+)*");
        if (txtsearch.Text == "")
        {
            lblmsg.Text = "Search the supplier";
        }
```

```
        else if (txtname.Text == "" || txtcontact.Text == "" ||
txtaddress.Text == "")
        {
            lblmsg.Text = "Enter all the fields";
        }
        else if (con.IsMatch(txtcontact.Text) == false)
        {
            lblmsg .Text  = "Enter valid contact number";
            txtcontact.Focus();
        }
        else if (name.IsMatch(txtname.Text) == false)
        {
            lblmsg.Text = "Only alphabets ";
            txtname.Focus();
        }
        else
        {
            lblmsg.Text = "";
            try
            {
                c = new connect();
                c.cmd.CommandText = "update branchdet set
name=@name,contact=@contact,address=@address where bid=@bid";
                c.cmd.Parameters.Add("@bid", SqlDbType.NVarChar).Value =
lblid.Text;
                c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
                c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value =
Convert.ToInt64(txtcontact.Text);
                c.cmd.Parameters.Add("@address", SqlDbType.NVarChar).Value =
txtaddress.Text;
                c.cmd.ExecuteNonQuery();
                //ds = new DataSet();
                //adp.SelectCommand = c.cmd;
                //adp.Fill(ds, "bd");
                lblmsg .Text ="Branch Updated";
                txtsearch.Text = "";
                lblid.Text = "";
                txtname.Text = "";
                txtcontact.Text = "";
                txtaddress.Text = "";
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnrefresh_Click(object sender, EventArgs e)
    {

    }
```

```
}
```

Source code:

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
//using System.Windows.Forms;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
public partial class custadd : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();
    int count;
    protected void Page_Load(object sender, EventArgs e)
    {
        //if (DropDownList2.Items.Count == 0)
        //{
        //    DropDownList2.Items.Add("---Select---");
        //}
        try
        {
            c = new connect();




   ds = new DataSet();
            if (DropDownList1.Items.Count == 0)
            {
                c.cmd.CommandText = "SELECT DISTINCT [category] FROM
[product]";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cat");
                if (ds.Tables["cat"].Rows.Count > 0)
                {
                    DropDownList1.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                    {
```

```
DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                    }
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
        lbldate1.Text = DateTime.Today.ToString("dd'/'MM'/'yyyy");

        txtcustid.ReadOnly = true;
        if (lblid.Text == "")
        {
            Calendar1.Visible = false;
        }
        if (!IsPostBack)
        {
            GenerateID();
        }
    }
    private void GenerateID()
    {
        String book = "CB";
        c = new connect();
        c.cmd.CommandText = "select count(cbid) from custbook";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;
        lblid.Text = book + i.ToString();
    }
    protected void btnsave_Click(object sender, EventArgs e)
```

```
{
         if (txtname.Text == "" || txtemail.Text == "" || txtaddress.Text ==
"" ||  txtduedate.Text == "" || txtadvance.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the fields')</script>");
        }
         else
         {
         Double p = Convert.ToDouble(txttotalvalue .Text );


        Regex name = new Regex("^[A-Z][a-zA-Z]");
```

```
        //Regex email = new Regex(@"^[a-za-z0-9]+([-+.'][a-za-z0-9]+)*@[a-
z]+([-.][a-z]+)*\.[a-z]+([-.][a-z]+)*");
        Double a = Convert.ToDouble(txtadvance.Text);
        Regex adv = new Regex("^[1-9][0-9]*");


         if (name.IsMatch(txtname.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('only alphabet and first letter should be
capital')</script>");

        }
        else if (adv.IsMatch(txtadvance.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Only valid numbers')</script>");
        }
        else if (a > p)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Advance must be lesser than a total value')</script>");
            txtadvance.Focus();
        }
        else
        {
            try
            {

              c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from custdetail where
contact='" + txtcontact.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cust");
                if (ds.Tables["cust"].Rows.Count > 0)
                {

                }
                else
                {
                    c.cmd.CommandText = "Insert into custdetail
values(@custid,@name,@address,@email,@contact)";
                    c.cmd.Parameters.Add("@custid", SqlDbType.VarChar).Value
= txtcustid.Text;
                    c.cmd.Parameters.Add("@name", SqlDbType.VarChar).Value =
txtname.Text;
                    c.cmd.Parameters.Add("@address", SqlDbType.VarChar).Value
= txtaddress.Text;
                    c.cmd.Parameters.Add("@email", SqlDbType.VarChar).Value =
txtemail.Text;
                    c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value
= Convert.ToInt64(txtcontact.Text);
                    c.cmd.ExecuteNonQuery();
```

```
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }

            //saving customer booking details
            try
            {
                c = new connect();
                String mfd = "No", cancel = "No", bill = "No", rid = "No";
                c.cmd.CommandText = "insert into custbook
values(@cbid,@date,@duedate,@advance,@totalval,@mfd,@cancel,@bill,@receiptid,
@custid)";
                c.cmd.Parameters.Add("@cbid", SqlDbType.NVarChar).Value =
lblid.Text;
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate1.Text;
                c.cmd.Parameters.Add("@duedate", SqlDbType.NVarChar).Value =
txtduedate.Text;




 c.cmd.Parameters.Add("@advance", SqlDbType.Int).Value =
Convert.ToInt32(txtadvance.Text);
                c.cmd.Parameters.Add("@totalval", SqlDbType.Decimal ).Value =
Convert.ToDecimal (txttotalvalue.Text);
                c.cmd.Parameters.Add("@mfd", SqlDbType.NVarChar).Value = mfd;
                c.cmd.Parameters.Add("@cancel", SqlDbType.NVarChar).Value =
cancel;
                c.cmd.Parameters.Add("@bill", SqlDbType.NVarChar).Value =
bill;
                c.cmd.Parameters.Add("@receiptid", SqlDbType.NVarChar).Value
= rid;
                c.cmd.Parameters.Add("@custid", SqlDbType.NVarChar).Value =
txtcustid.Text;
                c.cmd.ExecuteNonQuery();
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            // saving to cbookdet table m:n
            try
            {
```

```
                String gst = "3";
                c = new connect();
                for (int i = 0; i < GridView1.Rows.Count; i++)
                {
                    c.cmd.CommandText = "insert into
cbookdet(cbid,pid,qty,price,gst) values('" + lblid.Text + "','" +
GridView1.Rows[i].Cells[1].Text + "','" +Convert .ToInt16 (
GridView1.Rows[i].Cells[3].Text) + "','" +Convert .ToDecimal (
GridView1.Rows[i].Cells[4].Text) + "','" + gst.ToString() + "')";
                    c.cmd.ExecuteNonQuery();
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Bokking is done successfully')</script>");
        Session["i"] = lblid.Text;
        Response.Redirect("~/custrecp.aspx");



            Response.Redirect(Request.Url.AbsoluteUri);
            txtcustid.Text = "";
            txtname.Text = "";
            txtemail.Text = "";
            txtaddress.Text = "";
            txtpid.Text = "";
            txtqty.Text = "";
            txtprice.Text = "";
            txtduedate.Text = "";
            txttotalvalue.Text = "";
            txtadvance.Text = "";

            GenerateID();
        }
    }
    }
    protected void btnclear_Click(object sender, EventArgs e)
    {
        Response.Redirect(Request.Url.AbsoluteUri);
        dt.Rows.Clear();
        dt.Clear();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        txtcontact.Text = "";
        txtcustid.Text = "";
        txtname.Text = "";
        txtemail.Text = "";
        txtaddress.Text = "";
```

```
        txtpid.Text = "";
        txtqty.Text = "";
        txtprice.Text = "";
        txtduedate.Text = "";
        txttotalvalue.Text = "";
        txtadvance.Text = "";
    }
    protected void btngetinfo_Click1(object sender, EventArgs e)
    {
        Regex con = new Regex("^[6-9][0-9]{9}");
        if (con.IsMatch(txtcontact.Text) == false || txtcontact .Text =="")
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter a valid contact no')</script>");
            txtcontact.Text = "";
        }
        else
        {
            try
            {




    c = new connect();
                ds = new DataSet();

                c.cmd.CommandText = "select * from custdetail where
contact='" + txtcontact.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cust");
                if (ds.Tables["cust"].Rows.Count > 0)
                {
                    txtcontact.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[4]);
                    txtcustid.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[0]);
                    txtname.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[1]);
                    txtemail.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[3]);
                    txtaddress.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[2]);
                    c.cmd.ExecuteNonQuery();
                    txtname.ReadOnly = true;
                    txtcustid.ReadOnly = true;
                    txtemail.ReadOnly = true;
                    txtaddress.ReadOnly = true;
                }
                else
                {
                    txtaddress.Text = "";
                    txtemail.Text = "";
```

```
                    txtname.Text = "";
                    String cus = "C";
                    if (IsPostBack)
                    {
                        c.cmd.CommandText = "Select count(custid) from
custdetail";
                        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
                        i = i + 1001;
                        txtcustid.Text = cus + i.ToString();
                        txtname.ReadOnly = false ;
                        txtcustid.ReadOnly = false;
                        txtemail.ReadOnly = false;
                        txtaddress.ReadOnly = false;
                    }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();




 }
        }
    }
    protected void btnadd2_Click(object sender, EventArgs e)
    {

        int n = txtqty.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (DropDownList1.SelectedItem.Text == "" ||
DropDownList2.SelectedItem.Text == "" || txtqty.Text == "")
        {
           // MessageBox.Show("Enter the fields");
        }
        else if (con.IsMatch(txtqty.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter valid qyantity')</script>");
            txtqty.Focus();
        }

        else
        {
            try
            {

                c = new connect();
                Double t;
                DataTable dt = new DataTable();
```

```
        dt.Columns.Add("Product ID");
        dt.Columns.Add("Name");
        dt.Columns.Add("Quantity");
        dt.Columns.Add("Price");
        DataRow dr = null;
        if (ViewState["pro"] != null)
        {
            for (int i = 0; i < 1; i++)
            {
                dt = (DataTable)ViewState["pro"];
                if (dt.Rows.Count > 0)
                {
                    dr = dt.NewRow();
                    dr["Product ID"] = txtpid.Text;
                    dr["Name"] = DropDownList2.Text;
                    dr["Quantity"] = txtqty.Text;
                    dr["Price"] = txtprice.Text;
                    dt.Rows.Add(dr);
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                    t = Convert.ToDouble(txttotalvalue.Text);




Double p = Convert.ToDouble(txtprice.Text);
                    Double s = t + p;
                    txttotalvalue.Text = s.ToString();
                }
            }
        }
        else
        {
            dr = dt.NewRow();
            dr["Product ID"] = txtpid.Text;
            dr["Name"] = DropDownList2.Text;
            dr["Quantity"] = txtqty.Text;
            dr["Price"] = txtprice.Text;
            dt.Rows.Add(dr);
            GridView1.DataSource = dt;
            GridView1.DataBind();
            txttotalvalue.Text = txtprice.Text;
        }
        ViewState["pro"] = dt;
        t = Convert.ToDouble(txttotalvalue.Text);
        txtpid.Text = "";
        txtqty.Text = "";
        txtprice.Text = "";
        DropDownList1.SelectedIndex = 0;
        DropDownList1.SelectedIndex = 0;
    }
    catch (Exception)
    {
        throw;
```

```
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btndate2_Click(object sender, EventArgs e)
    {
        Calendar1.Visible = true;
    }
    protected void Calendar1_SelectionChanged1(object sender, EventArgs e)
    {
        DateTime dt1, dt2;
        dt1 =Convert .ToDateTime (DateTime.Today.ToShortDateString());
        dt2 = Calendar1.SelectedDate;
        if (dt2 < dt1)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Due date must be greter then date')</script>");
        }




else
        {
            String date = Calendar1.SelectedDate.ToString("dd'.'MM'.'yyyy");
            txtduedate.Text = date;

        }
    }
    protected void DropDownList2_SelectedIndexChanged(object sender,
EventArgs e)
    {

        try
        {
            c = new connect();
            ds = new DataSet();
            if (IsPostBack)
            {
                c.cmd.CommandText = "select * from product where name ='" +
DropDownList2.SelectedItem.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    txtpid.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[0]);
                    c.cmd.ExecuteNonQuery();
                }
            }
        }
```

```
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void txtcontact_TextChanged(object sender, EventArgs e)
    {

    }
    protected void Calendar1_SelectionChanged2(object sender, EventArgs e)
    {
        if (lblid.Text == "")
        {
            Calendar1.Visible = false;
        }
        else
        {



  if (txtduedate.Text == "")
            {
                Calendar1.Visible = true;
            }
            else
            {
                Calendar1.Visible = false;
            }
        }
        DateTime dt1, dt2;
        dt1 = Convert .ToDateTime ( DateTime.Today.ToShortDateString());
        dt2 =Convert .ToDateTime (
Calendar1.SelectedDate.ToShortDateString());
        if (dt2 > dt1)
        {
            //lbldate.Visible = false;
            txtduedate.Text =
Calendar1.SelectedDate.ToString("dd'/'MM'/'yyyy");
            Calendar1.Visible = false;
        }
        else
        {
            //lbldate.Visible = true;
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('due date must be greter then date')</script>");
        }
    }

    protected void txtprice_TextChanged(object sender, EventArgs e)
    {
```

```
    }
    protected void txtqty_TextChanged(object sender, EventArgs e)
    {

        int n = txtqty.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (con.IsMatch(txtqty.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Only valid no')</script>");
        }
        else
        {

            try
            {




 Double gst = 3;
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from product where pid='" +
txtpid.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    Double p =
Convert.ToInt32(ds.Tables["prod"].Rows[0].ItemArray[3]);
                    Double a = (p * gst) / 100;

                    int q = Convert.ToInt32(txtqty.Text);



                    Double g = a + p;
                    Double t = g * q;
                    txtprice.Text = t.ToString();

                    //txttotalvalue.Text = t.ToString();
                }
            }

            catch (Exception)
            {
                throw;
            }
            finally
            {
```

```
                c.cnn.Close();
            }
        }

    }


    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList1 .SelectedItem .Text =="---Select---")
        {
            DropDownList2 .Items .Clear ();
            DropDownList2 .Items .Add ("---Select---");
        }
        else
        {

c = new connect();
        ds = new DataSet();
        DropDownList2.Items.Clear();
        c.cmd.CommandText = "select * from product where category ='" +
DropDownList1.SelectedItem.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "po");
        if (ds.Tables["po"].Rows.Count > 0)
        {
            DropDownList2.Items.Add("---Select---");
            int i;
            for (i = 0; i < ds.Tables["po"].Rows.Count; i++)
            {

DropDownList2.Items.Add(ds.Tables["po"].Rows[i].ItemArray[1].ToString());
            }

        }

    }
    }
    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        dt =  (DataTable)ViewState["pro"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        Double pr = Convert.ToDouble(row.Cells[4].Text);
        Double tt = Convert.ToDouble(txttotalvalue.Text);
        Double ttl = tt - pr;
        txttotalvalue.Text = Convert.ToString(ttl);
```

```
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (GridView1.Rows.Count <= 0)
        {
            ViewState["pro"] = null;
        }

    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
    {

 count = 1;
        GridView1.EditIndex = e.NewEditIndex;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        dt =  (DataTable)ViewState["pro"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        dt.Rows[row.DataItemIndex]["Quantity"] =
((TextBox)(row.Cells[3].Controls[0])).Text;
        String pid =
Convert.ToString(((TextBox)(row.Cells[1].Controls[0])).Text);
        Double qty =
Convert.ToDouble(((TextBox)(row.Cells[3].Controls[0])).Text);
        Double pr =
Convert.ToDouble(((TextBox)(row.Cells[4].Controls[0])).Text);
        Double tt = Convert.ToDouble(txttotalvalue.Text);
        if (qty < 1000)
        {
            double total = tt - pr;
            try
            {
                Double gst = 3;
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from product where pid='" + pid
+ "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    Double p =
Convert.ToInt32(ds.Tables["prod"].Rows[0].ItemArray[3]);
                    Double a = (p * gst) / 100;
                    //int q = Convert.ToInt32(txtqty.Text);
                    Double g = a + p;
                    Double t = g * qty;
                    //txtprice.Text = t.ToString();
                    dt.Rows[row.DataItemIndex]["Price"] = t.ToString();
```

```
                    txttotalvalue.Text = Convert.ToString(t + total);
                }
            }

            catch (Exception)
            {
                throw;
            }




   finally
            {
                c.cnn.Close();
            }
            GridView1.EditIndex = -1;
            GridView1.DataSource = dt;
            GridView1.DataBind();
            GridView1.Visible = true;
        }
        else
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('We sale only 1000 items at a time')</script>");
        }

    }
    protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs
e)
    {
        if (count == 1)
        {
            e.Row.Cells[1].Enabled = false;
            e.Row.Cells[2].Enabled = false;
            e.Row.Cells[4].Enabled = false;
          // e.Row.Cells[5].Enabled = false;

        }
    }
}
```

Customer order cancel

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
//using System.Windows .Forms ;
using System.Data .SqlClient ;
```

```
using System.Text.RegularExpressions;

public partial class custcancel : System.Web.UI.Page
{
    connect c;
    DataSet ds;




 SqlDataAdapter adp=new SqlDataAdapter ();
    protected void btnsearch_Click(object sender, EventArgs e)
    {
        GridView1.Visible = true;
        lblid.Visible = true;
        lblname.Visible = true;
        lblcbid.Visible = true;
        Label4.Text = "Customer ID";
        Label5.Text = "Customer Name";
        Label6.Text = "Customer Booking ID";
        c = new connect();
        ds = new DataSet();
        Regex con = new Regex("^[6-9][0-9]{9}");

        if (txtsearch.Text == "")
        {
            lblcontact.Text = "Enter contact number";
            GridView1.Visible = false;
            Label7.Text = "";
            Label3.Text = "";
            lblname.Visible = false ;
        }
        else if (con.IsMatch(txtsearch.Text) == false ||
txtsearch.Text.Length < 10)
        {
            Label7.Text = "";
            lblcontact.Text = "Enter valid contact number";
            txtsearch.Focus();
            GridView1.Visible = false;
            Label3.Text = "";
        }
        else
        {
            lblcontact.Text = "";
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from custdetail where
contact='" + txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cust");
                if (ds.Tables["cust"].Rows.Count > 0)
                {
```

```
                    Label4.Visible = true;
                    Label5.Visible = true;
                    Label5.Visible = true;
                    lblid.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[0]);
                    lblname.Text =




Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[1]);
                    try
                    {
                        c = new connect();
                        ds = new DataSet();
                        c.cmd.CommandText = "select * from custbook where
custid='" + lblid.Text + "' and mfd='" + "No" + "'";
                        adp.SelectCommand = c.cmd;
                        adp.Fill(ds, "cust");
                        if (ds.Tables["cust"].Rows.Count > 0)
                        {
                            Label6.Visible = true;
                            lblcbid.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[0]);
                            c.cmd.ExecuteNonQuery();
                            Label7.Text = null;
                            try
                            {
                                c = new connect();
                                ds = new DataSet();
                                c.cmd.CommandText = "select
cbid,product.pid,name,qty from product,cbookdet where
product.pid=cbookdet.pid and cbookdet.cbid='" + lblcbid.Text + "'";
                                adp.SelectCommand = c.cmd;
                                adp.Fill(ds, "cus");
                                if (ds.Tables["cus"].Rows.Count > 0)
                                {
                                    GridView1.DataSource = ds.Tables["cus"];
                                    GridView1.DataBind();
                                }
                                else
                                {
                                    Label3.Text = "NO PRODUCT";
                                }
                            }

                            catch (Exception)
                            {
                                throw;
                            }
                            finally
                            {
                                c.cnn.Close();
                            }
                        }
```

```
                        else
                        {
                            lblid.Text  = "";
                            lblcbid.Text = "";
                            lblname.Text = "";



      Label4.Visible = false;
                            Label5.Visible = false;
                            Label6.Visible = false;
                            Label7.Text = "Product already
manufactured.Cancellation is not possible";
                            GridView1.Visible = false;
                        }
                    }
                    catch (Exception)
                    {
                        throw;
                    }
                    finally
                    {
                        c.cnn.Close();
                    }
                }
                else
                {

                    Label7.Text = "Record Not found";
                    lblname.Text = "";
                    GridView1.Visible = false;
                }


            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        Label4.Visible = false;
        Label5.Visible = false;
        Label6.Visible = false;
        txtsearch.MaxLength = 10;
    }
```

```
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
    {
        Label4.Visible = true;



  Label5.Visible = true;
        Label6.Visible = true;
        c = new connect();
        ds = new DataSet();
        GridView1.EditIndex = e.NewEditIndex;
        c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
lblcbid.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        Label4.Visible = true;
        Label5.Visible = true;
        Label6.Visible = true;
        c = new connect();
        ds = new DataSet();
        Label pid = GridView1.Rows[e.RowIndex].FindControl("lblproductid") as
Label;
        TextBox qty = GridView1.Rows[e.RowIndex].FindControl("txtqty") as
TextBox;
        c.cmd.CommandText = "Update cbookdet set qty='" + qty.Text + "'where
pid=  '" + pid.Text + "'";
        c.cmd.ExecuteNonQuery();
        Label3.Text = "updated";
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
lblcbid.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }
    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        Label4.Visible = true;
```

```
        Label5.Visible = true;
        Label6.Visible = true;



  c = new connect();
        ds = new DataSet();
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
lblcbid.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }



    protected void GridView1_RowDeleting1(object sender,
GridViewDeleteEventArgs e)
    {
        c = new connect();
        ds = new DataSet();
        Label ppid = GridView1.Rows[e.RowIndex].FindControl("lblpid") as
Label;
        c.cmd.CommandText = "delete from cbookdet where pid= '" + ppid.Text +
"'";
        c.cmd.ExecuteNonQuery();
        Label3.Text = "deleted";
        GridView1.EditIndex = -1;
        c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
lblcbid.Text + "'";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "cus");
        if (ds.Tables["cus"].Rows.Count > 0)
        {
            GridView1.DataSource = ds.Tables["cus"];
            GridView1.DataBind();
        }
    }
    protected void txtsearch_TextChanged(object sender, EventArgs e)
    {
        Label3.Text = "";
        Label4.Text = "";
        Label5.Text = "";
        Label6.Text = "";
        lblcbid.Text = "";
        lblcontact.Text = "";
        lblid.Text = "";
        GridView1.Visible = false;
```

```
  }
}
```

Customer_update

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Text.RegularExpressions;

public partial class custdisplay : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblmsg.Visible = false;
        txtsearch.MaxLength = 10;
        try
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "select * from custdetail";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cust");
            if (ds.Tables["cust"].Rows.Count > 0)
            {
                GridView1.DataSource = ds.Tables["cust"];
                GridView1.DataBind();
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('No records')</script>");
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
```

```
   {
            c.cnn.Close();
        }
    }
    protected void btnsearch_Click(object sender, EventArgs e)
    {
        Regex search = new Regex("^[6-9][0-9]{9}");
        if (search.IsMatch(txtsearch.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid contact no')</script>");
        }
        else
        {
            lblsearch.Text = "";
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from custdetail where
contact='" + txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cust");
                if (ds.Tables["cust"].Rows.Count > 0)
                {
                    lblmsg.Text = "";
                    lblid.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[0]);
                    txtname.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[1]);
                    txtcontact.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[4]);
                    txtemail.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[3]);
                    txtaddress.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[2]);
                }
                else
                {
                    //lblmsg.Visible = true;
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('No records')</script>");
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
```

```
  {
                c.cnn.Close();
            }
        }
    }
    protected void btnupdate_Click(object sender, EventArgs e)
    {
        Regex name = new Regex("^[A-Z]([a-zA-Z])*");
        Regex con = new Regex("^[1-9][0-9]{9}");
       // Regex email = new Regex();
        if (lblid.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Search the customer first')</script>");
            txtsearch.Focus();
        }
        else if (txtname.Text == "" || txtcontact.Text == "" ||
txtaddress.Text == "" || txtemail.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter al the fields')</script>");

        }
        else if (name.IsMatch(txtname.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only alphabets  and first letter should be
capital')</script>");

        }
        else if (con.IsMatch(txtcontact.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid contact no')</script>");

        }
        else
        {
            lbllmsg.Text = "";
            lblmsg.Text = "";
            lblname.Text = "";
            lblsearch.Text = "";
            try
            {
                c = new connect();
                c.cmd.CommandText = "update custdetail set
name=@name,address=@address,email=@email,contact=@contact where
custid=@custid";



     c.cmd.Parameters.Add("@custid", SqlDbType.NVarChar).Value =lblid.Text;
                c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
```

```
                c.cmd.Parameters.Add("@address", SqlDbType.NVarChar).Value =
txtaddress.Text;
                c.cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value =
txtemail.Text;
                c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value =
Convert.ToInt64(txtcontact.Text);
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Records  Updated')</script>");

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnclear_Click(object sender, EventArgs e)
    {
        txtaddress.Text = "";
        txtcontact.Text = "";
        txtemail.Text = "";
        lblid.Text = "";
        txtname.Text = "";
        txtsearch.Text = "";

    }
    protected void btnrefresh_Click(object sender, EventArgs e)
    {

    }
}
```

Customer oreder report

code :

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;




using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Text.RegularExpressions;
```

```
public partial class custdisplay : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblmsg.Visible = false;
        txtsearch.MaxLength = 10;
        try
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "select * from custdetail";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cust");
            if (ds.Tables["cust"].Rows.Count > 0)
            {
                GridView1.DataSource = ds.Tables["cust"];
                GridView1.DataBind();
            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('No records')</script>");
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void btnsearch_Click(object sender, EventArgs e)
    {
        Regex search = new Regex("^[6-9][0-9]{9}");
        if (search.IsMatch(txtsearch.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid contact no')</script>");
        }
        else




  {
            lblsearch.Text = "";
            try
            {
                c = new connect();
                ds = new DataSet();
```

```
                c.cmd.CommandText = "select * from custdetail where
contact='" + txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cust");
                if (ds.Tables["cust"].Rows.Count > 0)
                {
                    lblmsg.Text = "";
                    lblid.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[0]);
                    txtname.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[1]);
                    txtcontact.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[4]);
                    txtemail.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[3]);
                    txtaddress.Text =
Convert.ToString(ds.Tables["cust"].Rows[0].ItemArray[2]);
                }
                else
                {
                    //lblmsg.Visible = true;
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('No records')</script>");
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnupdate_Click(object sender, EventArgs e)
    {
        Regex name = new Regex("^[A-Z]([a-zA-Z])*");
        Regex con = new Regex("^[1-9][0-9]{9}");
       // Regex email = new Regex();
        if (lblid.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Search the customer first')</script>");
            txtsearch.Focus();
        }



 else if (txtname.Text == "" || txtcontact.Text == "" || txtaddress.Text ==
"" || txtemail.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter al the fields')</script>");

        }
```

```
        else if (name.IsMatch(txtname.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only alphabets  and first letter should be
capital')</script>");

        }
        else if (con.IsMatch(txtcontact.Text) == false)
        {

            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid contact no')</script>");

        }
        else
        {
            lbllmsg.Text = "";
            lblmsg.Text = "";
            lblname.Text = "";
            lblsearch.Text = "";
            try
            {
                c = new connect();
                c.cmd.CommandText = "update custdetail set
name=@name,address=@address,email=@email,contact=@contact where
custid=@custid";
                c.cmd.Parameters.Add("@custid", SqlDbType.NVarChar).Value
=lblid.Text;
                c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
                c.cmd.Parameters.Add("@address", SqlDbType.NVarChar).Value =
txtaddress.Text;
                c.cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value =
txtemail.Text;
                c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value =
Convert.ToInt64(txtcontact.Text);
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Records  Updated')</script>");

            }
            catch (Exception)



 {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
```

```
    protected void btnclear_Click(object sender, EventArgs e)
    {
        txtaddress.Text = "";
        txtcontact.Text = "";
        txtemail.Text = "";
        lblid.Text = "";
        txtname.Text = "";
        txtsearch.Text = "";

    }
    protected void btnrefresh_Click(object sender, EventArgs e)
    {

    }
}
```

Customer order Report

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
public partial class custorder : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();

    protected void Page_Load(object sender, EventArgs e)
    {
        CrystalReportViewer1.Visible = false;

        if (DropDownList1.Items.Count == 0)
        {



     c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [cbid] FROM [custbook]";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cat");
            if (ds.Tables["cat"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
```

```
                {

DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                }
            }
        }

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        CrystalReportViewer1.Visible = true;
        CrystalReportViewer1.SelectionFormula = " {custbook.cbid}='" +
DropDownList1.SelectedItem.Text + "'";
        CrystalReportViewer1.RefreshReport();
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        CrystalReportViewer1.RefreshReport();
        //CrystalReportViewer1.SelectionFormula = "";
        c = new connect();
        ds = new DataSet();
        c.cmd.CommandText = "select * from custbook";
        adp.SelectCommand = c.cmd;
        adp.Fill(ds, "order");
        if (ds.Tables["order"].Rows.Count > 0)
        {
            CrystalReportViewer1.RefreshReport();
            CrystalReportViewer1.Visible = true;

            CrystalReportViewer1.ReportSource = CrystalReportSource1;
            CrystalReportViewer1.RefreshReport();
        }
    }
}
```

Order Receipt

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;




using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class custrecp : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
```

```
        CrystalReportViewer1.SelectionFormula = " {custbook.cbid}='" +
(String)Session["i"] + "'";
        CrystalReportViewer1.RefreshReport();
    }
    protected void CrystalReportViewer1_Init(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/custadd.aspx");
    }
}
```

Sale bill:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows .Forms ;
using System.Data.SqlClient ;
using System.Text.RegularExpressions;
public partial class empattendance : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (txtname.Text == "")
        {
            Panel1.Visible = false;
        }
        lblyr.Visible = false;
 lblmn.Visible = false;
        lbldate.Text = Convert.ToString(DateTime.Now.Day);
        lblmonth.Text = Convert.ToString(DateTime.Now.Month);
        lblyear.Text = Convert.ToString(DateTime.Now.Year);
        switch (Convert.ToInt32(lblmonth.Text))
        {
            case 1: lblmonth.Text = "Jan";
                break;
            case 2: lblmonth.Text = "Feb";
                break;
            case 3: lblmonth.Text = "March";
                break;
            case 4: lblmonth.Text = "April";
                break;
```

```
            case 5: lblmonth.Text = "May";
                break;
            case 6: lblmonth.Text = "June";
                break;
            case 7: lblmonth.Text = "July";
                break;
            case 8: lblmonth.Text = "Aug";
                break;
            case 9: lblmonth.Text = "Sep";
                break;
            case 10: lblmonth.Text = "Oct";
                break;
            case 11: lblmonth.Text = "Nov";
                break;
            case 12: lblmonth.Text = "Dec";
                break;
        }
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [empid] FROM [emp] where
status='"+"Active"+"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["em"].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables["em"].Rows[i].ItemArray[0].ToString());
                }

            }
        }

    }
protected void btncal_Click(object sender, EventArgs e)
    {
        lblmn.Visible = true;
        lblyr.Visible = true;
        Double p = Convert.ToDouble (txttotalday.Text);
        Double k = Convert.ToDouble (txtleaveassigned.Text);
        int n = txtleavetaken.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtleavetaken.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the leave taken')</script>");

            txtleavetaken.Focus();
        }
        else if (con.IsMatch(txtleavetaken.Text) == false)
```

```
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

            txtleavetaken.Text = "";
            txtleavetaken.Focus();
        }
        else if (Convert .ToDouble  (txtleavetaken .Text )> p)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Check the leave')</script>");

            txtleavetaken.Focus();
        }

        else
        {
            Double w = Convert.ToDouble (txtleavetaken.Text);
            Double s = k - w;
            Double t = p - w;
            if (w > k)
            {
                Double r = w - k;
                txtextra.Text = r.ToString();
            }
            else
            {
                txtextra.Text = "0";
            }
            txtworking.Text = t.ToString();
        }

    }

   protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (txtname.Text == "" || txtdesig.Text == "" || txttotalday.Text ==
"" || txtleaveassigned.Text == "" || txtleavetaken.Text == "" ||
txtworking.Text == "" || txtextra.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all fields')</script>");

        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "insert into attendance
values(@date,@month,@year,@totaldays,@leaveassigned,@leavetaken,@workingdays,
@empid)";
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
```

```
                c.cmd.Parameters.Add("@month", SqlDbType.NVarChar).Value =
lblmonth.Text;
                c.cmd.Parameters.Add("@year", SqlDbType.NVarChar).Value =
lblyear.Text;
                c.cmd.Parameters.Add("@totaldays", SqlDbType.NVarChar).Value
= txttotalday.Text;
                c.cmd.Parameters.Add("@leaveassigned",
SqlDbType.NVarChar).Value = txtleaveassigned.Text;
                c.cmd.Parameters.Add("@leavetaken", SqlDbType.NVarChar).Value
= txtleavetaken.Text;
                c.cmd.Parameters.Add("@workingdays",
SqlDbType.NVarChar).Value = txtworking.Text;
                c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value =
DropDownList1.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record inserted')</script>");

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
   protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs
e)
    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtname.Text = "";
            txtdesig.Text = "";
            Panel1.Visible = false;
            DropDownList1.Focus();
        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from emp where empid='" +
DropDownList1.SelectedItem + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "emp");
                if (ds.Tables["emp"].Rows.Count > 0)
                {
                    c.cmd.CommandText = "select * from attendance where
empid='" + DropDownList1.SelectedItem + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "att");
                    if (ds.Tables["att"].Rows.Count > 0)
```

```
                    {

Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('You already given the attendence')</script>");

                        txtname.Text = "";
                        txtdesig.Text = "";
                        Panel1.Visible = false;
                    }
                    else
                    {
                        Panel1.Visible = true;
                        txtname.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[1]);
                        txtdesig.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[7]);
                    }
                }
                else
                {
     Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Employee details dpes not exists')</script>");

                long m, y;
                txtleaveassigned.Text = "4";
                m = Convert.ToInt32(DateTime.Now.Month);
                y = Convert.ToInt32(DateTime.Now.Year);
                if (m == 1)
                {
                    m = 12;
                    y = y - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                else
                {
                    m = m - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                switch (m)
                {
                    case 1: lblmn.Text = "Jan";
                        txttotalday.Text = "31";
                        break;
                    case 2: lblmn.Text = "Feb";
                        txttotalday.Text = "28";
                        break;
                    case 3: lblmn.Text = "March";
                        txttotalday.Text = "31";
                        break;
                    case 4: lblmn.Text = "April";
                        txttotalday.Text = "30";
                        break;
                    case 5: lblmn.Text = "May";
                        txttotalday.Text = "31";
                        break;
                    case 6: lblmn.Text = "June";
```

```
                        txttotalday.Text = "30";
                        break;
                    case 7: lblmn.Text = "July";
                        txttotalday.Text = "31";
                        break;
                    case 8: lblmn.Text = "Aug";
                        txttotalday.Text = "31";
                        break;
                    case 9: lblmn.Text = "Sep";
                        txttotalday.Text = "30";
                        break;
                    case 10: lblmn.Text = "Oct";
                        txttotalday.Text = "31";
                        break;
                    case 11: lblmn.Text = "Nov";
                        txttotalday.Text = "30";
                        break;
                    case 12: lblmn.Text = "Dec";
                        txttotalday.Text = "31";
                        break;
                }
                lblmn.Visible = true;
                lblyr.Visible = true;
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
}
```

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows .Forms ;
using System.Data.SqlClient ;
using System.Text.RegularExpressions;
public partial class empattendance : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
```

```
    {
        if (txtname.Text == "")
        {
            Panel1.Visible = false;
        }



   lblyr.Visible = false;
        lblmn.Visible = false;
        lbldate.Text = Convert.ToString(DateTime.Now.Day);
        lblmonth.Text = Convert.ToString(DateTime.Now.Month);
        lblyear.Text = Convert.ToString(DateTime.Now.Year);
        switch (Convert.ToInt32(lblmonth.Text))
        {
            case 1: lblmonth.Text = "Jan";
                break;
            case 2: lblmonth.Text = "Feb";
                break;
            case 3: lblmonth.Text = "March";
                break;
            case 4: lblmonth.Text = "April";
                break;
            case 5: lblmonth.Text = "May";
                break;
            case 6: lblmonth.Text = "June";
                break;
            case 7: lblmonth.Text = "July";
                break;
            case 8: lblmonth.Text = "Aug";
                break;
            case 9: lblmonth.Text = "Sep";
                break;
            case 10: lblmonth.Text = "Oct";
                break;
            case 11: lblmonth.Text = "Nov";
                break;
            case 12: lblmonth.Text = "Dec";
                break;
        }
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [empid] FROM [emp] where
status='"+"Active"+"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["em"].Rows.Count; i++)
                {
```

```
DropDownList1.Items.Add(ds.Tables["em"].Rows[i].ItemArray[0].ToString());
                }

            }
        }


 }

    protected void btncal_Click(object sender, EventArgs e)
    {
        lblmn.Visible = true;
        lblyr.Visible = true;
        Double p = Convert.ToDouble (txttotalday.Text);
        Double k = Convert.ToDouble (txtleaveassigned.Text);
        int n = txtleavetaken.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtleavetaken.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the leave taken')</script>");

            txtleavetaken.Focus();
        }
        else if (con.IsMatch(txtleavetaken.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

            txtleavetaken.Text = "";
            txtleavetaken.Focus();
        }
        else if (Convert .ToDouble  (txtleavetaken .Text )> p)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Check the leave')</script>");

            txtleavetaken.Focus();
        }

        else
        {
            Double w = Convert.ToDouble (txtleavetaken.Text);
            Double s = k - w;
            Double t = p - w;
            if (w > k)
            {
                Double r = w - k;
                txtextra.Text = r.ToString();
            }
            else
            {
                txtextra.Text = "0";
            }
```

```
            txtworking.Text = t.ToString();
        }

    }
    protected void btnsubmit_Click(object sender, EventArgs e)




 {
        if (txtname.Text == "" || txtdesig.Text == "" || txttotalday.Text ==
"" || txtleaveassigned.Text == "" || txtleavetaken.Text == "" ||
txtworking.Text == "" || txtextra.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all fields')</script>");

        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "insert into attendance
values(@date,@month,@year,@totaldays,@leaveassigned,@leavetaken,@workingdays,
@empid)";
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
                c.cmd.Parameters.Add("@month", SqlDbType.NVarChar).Value =
lblmonth.Text;
                c.cmd.Parameters.Add("@year", SqlDbType.NVarChar).Value =
lblyear.Text;
                c.cmd.Parameters.Add("@totaldays", SqlDbType.NVarChar).Value
= txttotalday.Text;
                c.cmd.Parameters.Add("@leaveassigned",
SqlDbType.NVarChar).Value = txtleaveassigned.Text;
                c.cmd.Parameters.Add("@leavetaken", SqlDbType.NVarChar).Value
= txtleavetaken.Text;
                c.cmd.Parameters.Add("@workingdays",
SqlDbType.NVarChar).Value = txtworking.Text;
                c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value =
DropDownList1.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record inserted')</script>");

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
```

```
            }
        }
}
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtname.Text = "";
            txtdesig.Text = "";
            Panel1.Visible = false;
            DropDownList1.Focus();
        }
        else
        {
            try
            {
                  c = new connect();
                  ds = new DataSet();
                  c.cmd.CommandText = "select * from emp where empid='" +
                  DropDownList1.SelectedItem + "'";
                  adp.SelectCommand = c.cmd;
                  adp.Fill(ds, "emp");
                  if (ds.Tables["emp"].Rows.Count > 0)
                  {
                    c.cmd.CommandText = "select * from attendance where
                    empid='" + DropDownList1.SelectedItem + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "att");
                    if (ds.Tables["att"].Rows.Count > 0)
                    {
                        Page.ClientScript.RegisterStartupScript(this.GetType(
                        ), "alert", "<script>alert('You already given the
                        attendence')</script>");
                        txtname.Text = "";
                        txtdesig.Text = "";
                        Panel1.Visible = false;
                    }
                    else
                    {
                        Panel1.Visible = true;
                        txtname.Text =
                        Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[1
                        ]);
                        txtdesig.Text =
                        Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[7
                        ]);
                    }
                }
                else
                  {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
                    "alert", "<script>alert('Employee details dpes not
                    exists')</script>");
                }
                long m, y;
```

```
        txtleaveassigned.Text = "4";
        m = Convert.ToInt32(DateTime.Now.Month);
        y = Convert.ToInt32(DateTime.Now.Year);
        if (m == 1)
        {
            m = 12;
            y = y - 1;
            lblyr.Text = Convert.ToString(y);
        }
        else
        {
            m = m - 1;
            lblyr.Text = Convert.ToString(y);
        }
        switch (m)
        {
            case 1: lblmn.Text = "Jan";
                txttotalday.Text = "31";
                break;
            case 2: lblmn.Text = "Feb";
                txttotalday.Text = "28";
                break;
            case 3: lblmn.Text = "March";
                txttotalday.Text = "31";
                break;
            case 4: lblmn.Text = "April";
                txttotalday.Text = "30";
                break;
            case 5: lblmn.Text = "May";
                txttotalday.Text = "31";
                break;
            case 6: lblmn.Text = "June";
                txttotalday.Text = "30";
                break;
            case 7: lblmn.Text = "July";
                txttotalday.Text = "31";
                break;
            case 8: lblmn.Text = "Aug";
                txttotalday.Text = "31";
                break;
            case 9: lblmn.Text = "Sep";
                txttotalday.Text = "30";
                break;
            case 10: lblmn.Text = "Oct";
                txttotalday.Text = "31";
                break;
            case 11: lblmn.Text = "Nov";
                txttotalday.Text = "30";
                break;
            case 12: lblmn.Text = "Dec";
                txttotalday.Text = "31";
                break;
        }
        lblmn.Visible = true;
        lblyr.Visible = true;
    }
```

```
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
}
```

Purchase bill

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
public partial class purchorder : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();
    DataRow dr = null;
    int count;
    protected void Page_Load(object sender, EventArgs e)
    {

        Label1.Visible = false;
        btnsubmit.Visible = false;
        if (lblsid.Text == "")
        {
            Calendar1.Visible = false;
        }
        lbldate.Text = DateTime.Today.ToShortDateString();
        if (!IsPostBack)
        {
            GenerateID();
        }
        try
        {
            c = new connect();
            ds = new DataSet();
            if (DropDownList3.Items.Count == 0)
            {
                c.cmd.CommandText = "SELECT [name] FROM [supplier]";
```

```
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cat");
                if (ds.Tables["cat"].Rows.Count > 0)
                {
                    DropDownList3.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                    {

DropDownList3.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                    }
                  }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
        try
        {

            if (DropDownList2.Items.Count == 0)

            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "SELECT [name] FROM [item]";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cat");
                if (ds.Tables["cat"].Rows.Count > 0)
                {
                    DropDownList2.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                    {
   DropDownList2.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                    }

                }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
        if (GridView1.Rows.Count > 0)
        {
            btnsubmit.Visible = true;
```

```
        }
    }
    private void GenerateID()
    {
        String po = "PO";



c = new connect();
        c.cmd.CommandText = "select count(pono) from purchaseorder";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;
        lblpurchaseorder.Text = po + i.ToString();
    }
    protected void DropDownList3_SelectedIndexChanged1(object sender,
EventArgs e)
    {
        if (DropDownList3.SelectedItem.Text == "---Select---")
        {
            Response.Redirect(Request.Url.AbsoluteUri);
        }
        else
        {
            lblsid.Visible = true;
            txtduedate.Visible = true;
            txtqty.Visible = true;
            txtunit.Visible = true;
            DropDownList2.Visible = true;
            btndae.Visible = true ;
            try
            {
                c = new connect();
                ds = new DataSet();
                if (IsPostBack)
                {
                        c.cmd.CommandText = "select * from supplier where
                        name='" + DropDownList3.SelectedItem.Text + "'";
                        adp.SelectCommand = c.cmd;
                        adp.Fill(ds, "sup");
                        if (ds.Tables["sup"].Rows.Count > 0)
                        {
                              lblsid.Text =
                              Convert.ToString(ds.Tables["sup"].Rows[0].ItemA
                              rray[0]);
                              c.cmd.ExecuteNonQuery();
                        }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
```

```
        }
    }

  protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        lblsid.Visible = true;
        String c = Calendar1.SelectedDate.ToString("dd'/'MM'/'yyyy");
        if (lblsid.Text == "")
        {
            Calendar1.Visible = false;
        }
        else
        {
            if (c == "")
            {
                Calendar1.Visible = true;
            }
            else
            {
                Calendar1.Visible = false;
            }
        }
        DateTime dt1, dt2;
        dt1 = Convert.ToDateTime(DateTime.Today.ToShortDateString());
        dt2 = Calendar1.SelectedDate;
        if (dt2 <= dt1)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
            "<script>alert('duedate must be greter then date')</script>");
        }
        else
        {
            lblduedate.Text = "";
            txtduedate.Text =
            Calendar1.SelectedDate.ToString("dd'/'MM'/'yyyy");
        }
    }
    protected void btndae_Click(object sender, EventArgs e)
    {
        lblsid.Visible = true;
        Calendar1.Visible = true;
    }
protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (DropDownList2.SelectedItem.Text == "---Select---")
        {
            txtqty.Text ="";
            txtunit.Text  = "";

            dt.Rows.Clear();
            dt.Clear();
        }
        else
        {
            lblsid.Visible = true;
            try
```

```
            {
                c = new connect();
                ds = new DataSet();
                if (IsPostBack)
                {
                                    c.cmd.CommandText = "select * from item
                        where name='" +   DropDownList2.SelectedItem.Text +
                        "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "it");
                    if (ds.Tables["it"].Rows.Count > 0)
                    {
                        Label1.Text =
Convert.ToString(ds.Tables["it"].Rows[0].ItemArray[0]);

                        txtunit.Text =
Convert.ToString(ds.Tables["it"].Rows[0].ItemArray[2]);

                        c.cmd.ExecuteNonQuery();
                    }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        DropDownList2.Enabled = true;
        int n = txtqty.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");

        if (DropDownList2.SelectedItem.Text == "" || txtqty.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the fields')</script>");

        }
 else if (con.IsMatch(txtqty.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

        }
        else
        {
            btnsubmit.Visible = true;
            dt.Columns.Add("item id");
            dt.Columns.Add("Name");
```

```
            dt.Columns.Add("Qty");
            dt.Columns.Add("Unit");
            if (ViewState["it"] != null)
            {
                for (int i = 0; i < 1; i++)
                {
                    dt = (DataTable)ViewState["it"];
                    if (dt.Rows.Count > 0)
                    {
                        dr = dt.NewRow();
                        dr["item id"] = Label1.Text;
                        dr["Name"] = DropDownList2.SelectedItem.Text;
                        dr["Qty"] = txtqty.Text;
                        dr["Unit"] = txtunit.Text;
                        dt.Rows.Add(dr);
                        GridView1.DataSource = dt;
                        GridView1.DataBind();
                    }
                }
            }
            else
            {
                dr = dt.NewRow();
                dr["item id"] = Label1.Text;
                dr["Name"] = DropDownList2.SelectedItem.Text;
                dr["Qty"] = txtqty.Text;
                dr["Unit"] = txtunit.Text;
                dt.Rows.Add(dr);
                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
            ViewState["it"] = dt;
            DropDownList2.SelectedItem.Enabled = false;
            txtqty.Text = "";
            txtunit.Text = "";
            lblsid.Visible = true;
        }
    }
    protected void btnclear_Click(object sender, EventArgs e)
    {
        Response.Redirect(Request.Url.AbsoluteUri);

    }
    protected void btnsubmit_Click(object sender, System.EventArgs e)
    {
        if (txtduedate.Text == "" || DropDownList3.SelectedItem.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all fields')</script>");

        }
        else
        {
            lblqty.Text = "";
            try
            {
```

```
                string remark = "not billed";
                c = new connect();
                c.cmd.CommandText = "insert into purchaseorder
values(@pono,@date,@duedate,@remark,@sid)";
                c.cmd.Parameters.Add("@pono", SqlDbType.NVarChar).Value =
lblpurchaseorder.Text;
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
                c.cmd.Parameters.Add("@duedate", SqlDbType.NVarChar).Value =
txtduedate.Text;
                c.cmd.Parameters.Add("@remark", SqlDbType.NVarChar).Value =
remark.ToString();
                c.cmd.Parameters.Add("@sid", SqlDbType.NVarChar).Value =
lblsid.Text;
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Order Submitted')</script>");
                try
                {
                    c = new connect();
                    for (int i = 0; i < GridView1.Rows.Count; i++)
                    {
                        string remark1 = "not billed";
                        c.cmd.CommandText = "insert into
podetail(iid,pono,qty,remark) values('" + GridView1.Rows[i].Cells[0].Text +
"','" + lblpurchaseorder.Text + "','" + GridView1.Rows[i].Cells[1].Text +
"','" + remark1.ToString() + "')";
                        c.cmd.ExecuteNonQuery();
                    }
                }
                catch (Exception)
                {
                    throw;
                }
                finally
                {
                    c.cnn.Close();
                }
                Response.Redirect(Request.Url.AbsoluteUri);

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }

            GenerateID();
        }

    }
protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs
e)
```

```
    {
        dt = (DataTable)ViewState["it"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        dt.Rows[row.DataItemIndex]["Qty"] =
((TextBox)(row.Cells[3].Controls[0])).Text;
        GridView1.EditIndex = -1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
    {
        count = 1;
        GridView1.EditIndex = e.NewEditIndex;
        dt = (DataTable)ViewState["it"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs
e)
    {
        dt = (DataTable)ViewState["it"];
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (GridView1.Rows.Count <= 0)
        {
            ViewState["it"] = null;
        }
    }
protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (count == 1)
        {
            e.Row.Cells[1].Enabled = false;
            e.Row.Cells[2].Enabled = false;
            e.Row.Cells[4].Enabled = false;
            // e.Row.Cells[5].Enabled = false;

        }
    }
}
```

Purchase bill:

```
using System;
```

```
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
public partial class purchasebill : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();
    int count;
    protected void Page_Load(object sender, EventArgs e)
    {
        Calendar1.Visible = false;
        txtpbdate.ReadOnly = true;
        lbldate.Text = DateTime.Today.ToShortDateString();
        if (!IsPostBack)
        {
            GenerateId();
        }
        if (DropDownList3.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [pono] FROM [podetail] where
[remark] = '" + "not billed" + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cat");
            if (ds.Tables["cat"].Rows.Count > 0)
            {
                DropDownList3.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                  {
            DropDownList3.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToS
            tring());
                }

            }

        }
    }
private void GenerateId()
    {
        String bill = "PB";
        c = new connect();
        c.cmd.CommandText = "select count(pbid) from purchasebill";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;
```

```
        lblid.Text = bill + i.ToString();
    }
protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (txtsno.Text == "" || txtno.Text == "" || txtpbdate.Text == "" ||
txttotalbill.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all the fields')</script>");
        }
        else
        {
            try
            {
                c = new connect();
                  c.cmd.CommandText = "insert into purchasebill
                  values(@pbid,@pbno,@pbdate,@billdate,@total,@sid)";
                  c.cmd.Parameters.Add("@pbid", SqlDbType.NVarChar).Value =
                  lblid.Text;
                  c.cmd.Parameters.Add("@pbno", SqlDbType.NVarChar).Value =
                  txtno.Text;
                  c.cmd.Parameters.Add("@pbdate", SqlDbType.NVarChar).Value =
                  txtpbdate.Text;
                  c.cmd.Parameters.Add("@billdate", SqlDbType.NVarChar).Value
                  = lbldate.Text;
                  c.cmd.Parameters.Add("@total", SqlDbType.NVarChar).Value =
                  txttotalbill.Text;
                  c.cmd.Parameters.Add("@sid", SqlDbType.NVarChar).Value =
                  txtsno.Text;
                  c.cmd.ExecuteNonQuery();
                  Page.ClientScript.RegisterStartupScript(this.GetType(),
                  "alert", "<script>alert('inserted')</script>");
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            try
            {
                  String remark = "bill";
                  c = new connect();
                  c.cmd.CommandText = "update podetail set remark=@remark
                  where iid='" + DropDownList1.SelectedItem.Text + "' and
                  pono='" + DropDownList3.SelectedItem.Text + "'";
                  c.cmd.Parameters.Add("@remark", SqlDbType.NVarChar).Value =
                  remark.ToString();
                  c.cmd.ExecuteNonQuery();
            }
            catch (Exception)
            {
                throw;
            }
```

```
finally
{
    c.cnn.Close();
}
try
{
    c = new connect();
    for (int i = 0; i < GridView1.Rows.Count; i++)
    {
      c.cmd.CommandText = "insert into pbdet(pbid,iid,qty1)
      values('" + lblid.Text + "','" +
      GridView1.Rows[i].Cells[0].Text + "','" +
      GridView1.Rows[i].Cells[2].Text + "')";
      c.cmd.ExecuteNonQuery();
    }
}
catch (Exception)
{
    throw;
}
finally
{
    c.cnn.Close();
}
try
{
    c = new connect();
    ds = new DataSet();
    for (int j = 0; j < GridView1.Rows.Count; j++)
    {
      c.cmd.CommandText = "select * from item where iid='" +
      GridView1.Rows[j].Cells[0].Text + "'";
      adp.SelectCommand = c.cmd;
      adp.Fill(ds, "it");
      if (ds.Tables["it"].Rows.Count > 0)
      {
            for (int i = 0; i < ds.Tables["it"].Rows.Count; i++)
            {
                  Double q =
                  Convert.ToInt16(ds.Tables["it"].Rows[i].ItemArr
                  ay[3]);
                  Double qt =
                  Convert.ToDouble(GridView1.Rows[j].Cells[2].Tex
                  t);
                  Double qty = q + qt;
                  c.cmd.CommandText = "update item  set qty='" +
                  Convert.ToInt16(qty) + "'where iid='" +
                  GridView1.Rows[j].Cells[0].Text + "'";
                c.cmd.ExecuteNonQuery();
            }
        }
        ds.Tables["it"].Clear();
    }
}
catch (Exception)
{
```

```
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
}
protected void btnadd_Click(object sender, EventArgs e)
{
    btnsubmit.Visible = true;
    int n = txtqty.Text.Length;
    n--;
    Regex con = new Regex("^[1-9][0-9]{" + n + "}");
    Regex digit = new Regex("^[1-9][0-9]*");
    if (txtiqty.Text == "" || txtprice.Text == "" || txtname.Text == "")
    {
        Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
        "<script>alert('Enter all the fields')</script>");
    }
    else if (con.IsMatch(txtqty.Text) == false)
    {
        lblqtymsg.Text = "check the quantity";
    }
    else if (digit.IsMatch(txtprice.Text) == false)
    {
        lblqtymsg.Text = "";
        lblpricemsg.Text = "Enter valid price";
    }
    else
    {
        int m = Convert.ToInt32(txtqty.Text);
        int n1 = Convert.ToInt32(txtiqty .Text );
        if (m > n1)
        {
              lblqtymsg.Text = "qty must be less than or equal to qty on
              hand";
        }
        else
        {
            lblqtymsg.Text = "";
            lblpricemsg.Text = "";

            try
            {
                c = new connect();

                dt.Columns.Add("Item ID");
                dt.Columns.Add("Name");
                dt.Columns.Add("Quantity");
                dt.Columns.Add("Price");
                DataRow dr = null;
                if (ViewState["it"] != null)
                {
                    for (int i = 0; i < 1; i++)
                    {
```

```
                            dt = (DataTable)ViewState["it"];
                            if (dt.Rows.Count > 0)
                            {
                                dr = dt.NewRow();
                                dr["Item ID"] = DropDownList1.SelectedValue;
                                dr["Name"] = txtiname.Text;
                                dr["Quantity"] = txtqty.Text;
                              dr["Price"] = txtprice.Text;
                              dt.Rows.Add(dr);
                              GridView1.DataSource = dt;
                              GridView1.DataBind();
                              Double t = Convert.ToDouble(txttotalbill.Text);
                              Double p = Convert.ToDouble(txtprice.Text);
                              Double q = Convert.ToDouble(txtqty.Text);
                              Double r = p * q;
                              Double z = r + t;
                              txttotalbill.Text = z.ToString();
                            }
                        }
                    }
                    else
                    {
                        dr = dt.NewRow();
                        dr["Item ID"] = DropDownList1.SelectedValue;
                        dr["Name"] = txtiname.Text;
                        dr["Quantity"] = txtqty.Text;
                        dr["Price"] = txtprice.Text;
                        dt.Rows.Add(dr);
                        GridView1.DataSource = dt;
                        GridView1.DataBind();
                        Double p = Convert.ToDouble(txtprice.Text);
                        Double q = Convert.ToDouble(txtqty.Text);
                        Double r = p * q;
                        txttotalbill.Text = r.ToString();
                    }
                    ViewState["it"] = dt;
                    DropDownList1.SelectedItem.Enabled = false;
                }
                catch (Exception)
                {
                    throw;
                }
                finally
                {
                    c.cnn.Close();
                }
                txtprice.Text = "";
                txtqty.Text = "";
                txtiname.Text = "";
                txtiqty.Text = "";
            }
        }
    }
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (DropDownList1.SelectedItem.Text == "---SELECT---")
```

```
        {
            txtiname.Text = "";
            txtiqty.Text = "";
            txtqty.Text = "";
            txtprice.Text = "";
        }

  else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                if (IsPostBack)
                {
                        c.cmd.CommandText = "select * from item where iid ='"
                        + DropDownList1.SelectedItem.Text + "'";
                        adp.SelectCommand = c.cmd;
                        adp.Fill(ds, "item");
                        if (ds.Tables["item"].Rows.Count > 0)
                        {
                              txtiname .Text  =
                              Convert.ToString(ds.Tables["item"].Rows[0].Item
                              Array[1]);
                        }
                        c.cmd.CommandText = "Select * from podetail where
                        pono='" + DropDownList3.SelectedItem.Text + "'and
                        iid='" + DropDownList1.SelectedItem.Text + "'";
                        adp.SelectCommand = c.cmd;
                        adp.Fill(ds, "po");
                        if (ds.Tables["po"].Rows.Count > 0)
                        {
                              txtiqty .Text  =
                              Convert.ToString(ds.Tables["po"].Rows[0].ItemAr
                              ray[2]);
                        }
                   }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        txtpbdate.Text = Calendar1.SelectedDate.ToString("dd'.'MM'.'yyyy");

    }
protected void btncalender_Click(object sender, EventArgs e)
    {
        btnsubmit.Visible = true;
```

```
        Calendar1.Visible = true;
    }
protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
    {
        txtsno.Text = "";
        txtpbdate.Text = "";
        txttotalbill.Text = "";
        ViewState["it"] = null;
        dt.Clear();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (DropDownList3.SelectedItem.Text == "---Select---")
        {
            Response.Redirect(Request.Url.AbsoluteUri);

        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();

                c.cmd.CommandText = "select * from purchaseorder where
                pono='" + DropDownList3.SelectedItem + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "pur");
                if (ds.Tables["pur"].Rows.Count > 0)
                {
                  txtsno.Text =
                  Convert.ToString(ds.Tables["pur"].Rows[0].ItemArray[4]);
                }
                  c.cmd.CommandText = "select * from supplier where sid='" +
                  txtsno.Text + "'";
                  adp.SelectCommand = c.cmd;
                  adp.Fill(ds, "sup");
                  if (ds.Tables["sup"].Rows.Count > 0)
                  {
                    txtname.Text =
Convert.ToString(ds.Tables["sup"].Rows[0].ItemArray[1]);
                }

                c.cmd.CommandText = "select * from podetail where pono='" +
DropDownList3.SelectedItem.Text + "' and remark='"+"not billed"+"'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "pro");
                DropDownList1.Items.Clear();
                if (ds.Tables["pro"].Rows.Count > 0)
                {
                    DropDownList1.Items.Add("---SELECT---");
```

```
    int i;
                    for (i = 0; i < ds.Tables["pro"].Rows.Count; i++)
                    {
DropDownList1.Items.Add(ds.Tables["pro"].Rows[i].ItemArray[0].ToString ());
                    }
                }

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnclear_Click(object sender, EventArgs e)
    {
        Response.Redirect(Request.Url.AbsoluteUri);
        //DropDownList1.Items.Add("---SELECT---");

    }

    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        dt = (DataTable)ViewState["it"];

        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        dt = (DataTable)ViewState["it"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        Double pr = Convert.ToDouble(row.Cells[4].Text);
        Double tt = Convert.ToDouble(txttotalbill.Text);
        Double ttl = tt - pr;
        txttotalbill.Text = Convert.ToString(ttl);
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (GridView1.Rows.Count <= 0)
        {
            ViewState["it"] = null;




  }
```

```
    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
    {
        count = 1;
        //GridViewRow row = GridView1.Rows[e.RowIndex];
        Double qt = Convert.ToDouble(GridView1.Rows[e.NewEditIndex ].Cells
[3].Text );
        Session["qq"] = qt;
        GridView1.EditIndex = e.NewEditIndex;
        dt = (DataTable)ViewState["it"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        dt = (DataTable)ViewState["it"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
         // Double qt =
Convert.ToDouble(GridView1.Rows[e.RowIndex].Cells[3].Text);

        dt.Rows[row.DataItemIndex]["Quantity"] =
((TextBox)(row.Cells[3].Controls[0])).Text;
        String pid =
Convert.ToString(((TextBox)(row.Cells[1].Controls[0])).Text);
        Double qty =
Convert.ToDouble(((TextBox)(row.Cells[3].Controls[0])).Text);
        Double pr =
Convert.ToDouble(((TextBox)(row.Cells[4].Controls[0])).Text);
        Double tt = Convert.ToDouble(txttotalbill.Text);
        if (qty != 0)
        {
            if (qty < 1000)
            {
                int qt = Convert.ToInt32(Session["qq"]);
                if (qty <= qt)
                {
                    double total = tt - pr;
                    try
                    {
                        Double gst = 3;
                        c = new connect();
                        ds = new DataSet();
                        c.cmd.CommandText = "select * from product where
pid='" + pid + "'";
                        adp.SelectCommand = c.cmd;




adp.Fill(ds, "prod");
                        if (ds.Tables["prod"].Rows.Count > 0)
```

```
                        {
                            Double p =
Convert.ToInt32(ds.Tables["prod"].Rows[0].ItemArray[3]);
                            Double a = (p * gst) / 100;
                            //int q = Convert.ToInt32(txtqty.Text);
                            Double g = a + p;
                            Double t = g * qty;
                            //txtprice.Text = t.ToString();
                            dt.Rows[row.DataItemIndex]["Price"] =
t.ToString();
                            txttotalbill.Text = Convert.ToString(t + total);
                        }
                    }

                    catch (Exception)
                    {
                        throw;
                    }
                    finally
                    {
                        c.cnn.Close();
                    }

                    GridView1.EditIndex = -1;
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                    GridView1.Visible = true;
                }
                else
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('qty must be less than or equal to qty
ordered')</script>");
                }

            }
            else
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('We sale only 1000 items at a time')</script>");
            }
        }
        else
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('check the qty')</script>");
        }
    }

protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (count == 1)
        {
            e.Row.Cells[1].Enabled = false;
            e.Row.Cells[2].Enabled = false;
            e.Row.Cells[4].Enabled = false;
```

```
        }
    }
}
```

Employee_Add

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Text.RegularExpressions;
public partial class empadd : System.Web.UI.Page
{
    connect c;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {

        ddldate.Items.Add("Date");
        if (ddlyear.Items.Count == 0)
        {
            int year = Convert.ToInt32(DateTime.Now.Year);
            int end = year - 18;
            int start = year - 50;
            ddlyear.Items.Add("Year");
            for (int i = start; i <= end ; i++)
            {
                ddlyear.Items.Add("" + i + "");
            }
        }
        txtid.ReadOnly = true;
        if (txtid.Text == "")
        {
           // Calendar1.Visible = false;
            Calendar2.Visible = false;
        }
        if (!IsPostBack)
        {
            GenerateID();
        }
    }
    private void GenerateID()
    {
        String emp = "E";
        c = new connect();
```

```
        c.cmd.CommandText = "select count(empid) from emp";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;
        txtid .Text  = emp  + i.ToString();
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        Regex con = new Regex ("^[6-9][0-9]{9}");



  Regex name= new Regex ("^[A-Z][a-zA-Z]");
        Regex basic = new Regex ("^[1-9][0-9]*");
        //int yr = Convert.ToInt32(Session["year"]);

        c = new connect();
        Regex email = new Regex (@"^[a-za-z0-9]+([-+.'][a-za-z0-9]+)*@[a-
z]+([-.][a-z]+)*\.[a-z]+([-.][a-z]+)*");
        try
        {
            if (txtaddress.Text == "" || DropDownList1 .SelectedItem .Text
==""  || txtbasic.Text == "" || txtcontact.Text == "" || txtdoj.Text == "" ||
txtid.Text == "" || txtname.Text == "")
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Enter  all fields')</script>");

            }

            else if (con.IsMatch(txtcontact.Text.Trim()) == false ||
txtcontact.Text.Length < 10)
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Enter only valid contact no')</script>");

                txtcontact.Text = "";
                txtcontact.Focus();
            }
            else if (name.IsMatch(txtname.Text) == false)
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('enter only alphabets and first letter should be
capital')</script>");

                txtname.Focus();
            }
            else if (basic.IsMatch(txtbasic.Text) == false)
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Basic should be anumber and cannot start with
zero')</script>");

            }
            else if (ddlyear.SelectedIndex ==0|| ddlmonth.SelectedIndex ==0
|| ddldate.SelectedIndex ==0)
```

```
            {
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Select the date')</script>");

            }




            else
            {
                if (txtemail.Text  !="")
                {
                    if (email.IsMatch(txtemail.Text) == false)

Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Check the email')</script>");

                }
                else
                {
                String s = "Active";
                int da = Convert.ToInt32(ddlyear.SelectedItem.Text);
                int y = Convert.ToInt32(Calendar2.SelectedDate.Year);
                int final = y - da;
                if (final < 18)
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Employee is not 18 years')</script>");

                }
                else
                {
                    c.cmd.CommandText = "insert into emp
values(@empid,@empname,@dob,@contact,@address,@email,@doj,@desig,@basic,@da,@
hra,@pf,@tax,@status)";
                    c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value
= txtid.Text;
                    c.cmd.Parameters.Add("@empname",
SqlDbType.NVarChar).Value = txtname.Text;
                    c.cmd.Parameters.Add("@dob", SqlDbType.NVarChar).Value =
ddldate .SelectedItem .Text +"-"+ddlmonth.SelectedValue +"-"+ddlyear
.SelectedItem .Text ;
                    c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value
= Convert.ToInt64(txtcontact.Text);
                    c.cmd.Parameters.Add("@address",
SqlDbType.NVarChar).Value = txtaddress.Text;
                    c.cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value
= txtemail.Text;
                    c.cmd.Parameters.Add("@doj", SqlDbType.NVarChar).Value =
txtdoj.Text;
                    c.cmd.Parameters.Add("@desig", SqlDbType.NVarChar).Value
= DropDownList1.SelectedItem.Text;
                    c.cmd.Parameters.Add("@basic", SqlDbType.BigInt).Value =
Convert.ToInt64(DropDownList1.SelectedValue);
```

```
                    c.cmd.Parameters.Add("@da", SqlDbType.Decimal).Value =
Convert.ToDecimal(txtda.Text);
                    c.cmd.Parameters.Add("@hra", SqlDbType.Decimal).Value =



Convert.ToDecimal(txthra.Text);
                    c.cmd.Parameters.Add("@pf", SqlDbType.Decimal).Value =
Convert.ToDecimal(txtpf.Text);
                    c.cmd.Parameters.Add("@tax", SqlDbType.Decimal).Value =
Convert.ToDecimal(txttax.Text);
                    c.cmd.Parameters.Add("@status", SqlDbType.NVarChar).Value
= s.ToString();

                    c.cmd.ExecuteNonQuery();

                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record inserted')</script>");

                }
            }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }

    protected void btndob_Click(object sender, EventArgs e)
    {
      //  Calendar1.Visible = true;
    }
    protected void btndoj_Click(object sender, EventArgs e)
    {
        Calendar2.Visible = true;
    }


    protected void btnclear_Click1(object sender, EventArgs e)
    {
        txtid.Text = "";
        txtname.Text = "";
        txtaddress.Text = "";
        txtcontact.Text = "";
        txtemail.Text = "";
        txtdoj.Text = "";
        txtbasic.Text = "";

    }
```

```
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)




    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtbasic.Text = "";
        }
        //else
        //txtbasic.Text = DropDownList1.SelectedValue.ToString();
    }

    protected void DropDownList3_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (ddlyear.SelectedIndex ==0)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Select the year first')</script>");

            ddlmonth.SelectedIndex = 0;
            ddldate.Items.Clear();
            ddldate.Items.Add("date");
        }
        else if (ddlmonth.SelectedIndex == 0)
        {
            ddldate.Items.Clear();
            ddldate.Items.Add("date");
        }
        else
        {
            ddldate.Items.Clear();
            ddldate.Items.Add("date");
            if (ddlmonth.SelectedItem.Text == "Date")
            {

            }
            else if (ddlmonth.SelectedValue == "2")
            {
                int y = Convert.ToInt32(ddlyear.Text);
                if (y / 4 == 0)
                {
                    for (int i = 1; i <= 28; i++)
                    {
                        ddldate.Items.Add("" + i + "");
                    }
                }
                else
                {
                    for (int i = 1; i <= 29; i++)
                    {
                        ddldate.Items.Add("" + i + "");
```

```
                    }
                }



     }
            else if (ddlmonth.SelectedValue == "4" || ddlmonth.SelectedValue
== "6" || ddlmonth.SelectedValue == "9" || ddlmonth.SelectedValue == "11")
            {
                for (int i = 1; i <= 30; i++)
                {
                    ddldate.Items.Add("" + i + ""); ;
                }
            }
            else
            {
                for (int i = 1; i <= 31; i++)
                {
                    ddldate.Items.Add("" + i + "");
                }
            }
        }
    }
    protected void Calendar2_SelectionChanged(object sender, EventArgs e)
    {

        txtdoj.Text = Calendar2.SelectedDate.ToString("dd'.'MM'.'yyyy");
            if (txtid.Text == "")
            {
                Calendar2.Visible = false;
            }
            else
            {
                if (txtdoj.Text == "")
                {
                    Calendar2.Visible = true;
                }
                else
                {
                    Calendar2.Visible = false;
                }
            }



    }
    protected void ddlyear_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void ddlyear_SelectedIndexChanged1(object sender, EventArgs e)
    {
        if (ddlyear.SelectedIndex ==0)
        {
            ddlmonth.SelectedIndex = 0;
```

```
            ddldate.Items.Clear();




      ddldate.Items.Add("date");
        }



    }
    protected void ddldate_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (ddlyear.SelectedItem.Text == "Year")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Select the year of DOB')</script>");

        }
        if (ddlmonth.SelectedIndex == 0)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Select the month ')</script>");

            ddldate.SelectedIndex = 0;
        }
    }
}
```

Employee update

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Text.RegularExpressions;

public partial class empupdate : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {

        txtdob.Enabled = false ;
        txtdoj.Enabled = false ;
        txtda.Enabled = false ;
```

```
        txthra.Enabled = false ;



    txtpf.Enabled = false ;
        txttax.Enabled = false ;
        try
        {
            c = new connect();
            c.cmd.CommandText = "select
empid,empname,dob,contact,address,email,doj,desig,basic,da,hra,pf,tax from
emp where status='" + "Active" + "'";
            ds = new DataSet();
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "emp");
            if (ds.Tables["emp"].Rows.Count > 0)
            {
                GridView2.DataSource = ds.Tables["emp"];
                GridView2.DataBind();
            }
            else
            {
               lblmsg .Text ="No Records";
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
        }
        if (DropDownList5.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [empid] FROM [emp] where
status='" + "Active" + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                DropDownList5.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["em"].Rows.Count; i++)
                {

DropDownList5.Items.Add(ds.Tables["em"].Rows[i].ItemArray[0].ToString());
                }

            }
        }
        if (DropDownList5.SelectedItem.Text == "" ||
DropDownList5.SelectedItem.Text == "---Select---")
        {
```

```
            txtname.ReadOnly = true;



  txtaddress.ReadOnly = true;
            txtcontact.ReadOnly = true;
            txtemail.ReadOnly = true;
            txtdoj.ReadOnly = true;
            txtda.ReadOnly = true;
            txthra.ReadOnly = true;
            txtpf.ReadOnly = true;
            txtstatus.ReadOnly = true;
            txtbasic.ReadOnly = true;
            txttax.ReadOnly = true;
        }

    }

    protected void btnclear_Click(object sender, EventArgs e)
    {

        txtname.Text = "";
      //  txtdob.Text = "";
        txtcontact.Text = "";
        txtaddress.Text = "";
        txtemail.Text = "";
        txtdoj.Text = "";
        DropDownList1.SelectedValue = "";
        txtbasic.Text = "";
        txtda.Text = "";
        txthra.Text = "";
        txtpf.Text = "";
        txttax.Text = "";
    }
    protected void btnupdate_Click(object sender, EventArgs e)
    {
        Regex con = new Regex ("^[6-9][0-9]{9}");
        Regex name= new Regex ("^[A-Z][a-zA-Z]");
        Regex basic = new Regex ("^[1-9][0-9]*");
        c = new connect();
            Regex email = new Regex (@"^[a-za-z0-9]+([-+.'][a-za-z0-9]+)*@[a-
z]+([-.][a-z]+)*\.[a-z]+([-.][a-z]+)*");
            if (txtaddress.Text == "" || DropDownList1 .SelectedItem .Text
==""  || txtbasic.Text == "" || txtcontact.Text == "" ||txtdoj.Text == "" ||
txtemail.Text == "" ||  txtname.Text == "")
            {
                MessageBox.Show("Enter all the fields");
            }
            else  if (con.IsMatch (txtcontact.Text.Trim ())==false ||
txtcontact.Text.Length <10)
            {
                    MessageBox.Show("Enter valid contact number");
                    txtcontact.Text = "";
                    txtcontact .Focus ();
```

```
   }
            else if (name.IsMatch(txtname.Text) == false)
            {
                MessageBox.Show("Only alphabet and First Letter should be
capital");
                txtname.Focus();
            }
            else if (basic.IsMatch(txtbasic.Text) == false)
            {
                MessageBox.Show("Basic should be a number and it cannot start
with zero");
            }

            else
            {
                try
                {
                    c = new connect();
                    c.cmd.CommandText = "update emp set
empname=@empname,contact=@contact,address=@address,email=@email,desig=@desig,
basic=@basic where empid=@empid";
                    c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value
= DropDownList5.SelectedItem.Text;
                    c.cmd.Parameters.Add("@empname",
SqlDbType.NVarChar).Value = txtname.Text;
                    c.cmd.Parameters.Add("@contact", SqlDbType.BigInt).Value
= Convert.ToInt64(txtcontact.Text);
                    c.cmd.Parameters.Add("@address",
SqlDbType.NVarChar).Value = txtaddress.Text;
                    c.cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value
= txtaddress.Text;
                    c.cmd.Parameters.Add("@desig", SqlDbType.NVarChar).Value
= txtdesig.Text;
                    c.cmd.Parameters.Add("@basic", SqlDbType.BigInt).Value =
Convert.ToInt64(txtbasic.Text);
                    c.cmd.ExecuteNonQuery();
                    c.cmd.ExecuteNonQuery();
                    ds = new DataSet();
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "emp");
                    MessageBox.Show("Employee Updated");
                }
                catch (Exception)
                {
                    throw;
                }
                finally
                {
                    c.cnn.Close();
                }
```

```
}
    }
    protected void DropDownList3_SelectedIndexChanged(object sender,

    protected void DropDownList5_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList5.SelectedItem.Text == "---Select---")
        {
            txtname.Text = "";
            txtcontact.Text = "";
            txtaddress.Text = "";
            txtemail.Text = "";
            txtdoj.Text = "";
            txtbasic.Text = "";
            txtda.Text = "";
            txthra.Text = "";
            txtpf.Text = "";
            txttax.Text = "";
            txtname.ReadOnly = true;
            txtaddress.ReadOnly = true;
            txtcontact.ReadOnly = true;
            txtemail.ReadOnly = true;
            txtdoj.ReadOnly = true;
            txtda.ReadOnly = true;
            txthra.ReadOnly = true;
            txtpf.ReadOnly = true;
            txtstatus.ReadOnly = true;
            txtbasic.ReadOnly = true;
            txttax.ReadOnly = true;
        }
        else
        {
            txtname.ReadOnly = false;
            txtaddress.ReadOnly = false ;
            txtcontact.ReadOnly = false;
            txtemail.ReadOnly = false;
            txtdoj.ReadOnly = false;
            txtda.ReadOnly = false;
            txthra.ReadOnly = false;
            txtpf.ReadOnly = false;
            txtstatus.ReadOnly = false;
            txtbasic.ReadOnly = false;
            txttax.ReadOnly = false;
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from emp where empid='" +
DropDownList5 .SelectedItem .Text  + "'";
                adp.SelectCommand = c.cmd;
```

```
 adp.Fill(ds, "emp");
                if (ds.Tables["emp"].Rows.Count > 0)
                {
                    //txtid.Text =                    txtname.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[1]);
                    txtdob .Text =Convert .ToString (ds.Tables ["emp"].Rows
[0].ItemArray [2]);
                    txtcontact.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[3]);
                    txtaddress.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[4]);
                    txtemail.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[5]);
                    txtdoj.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[6]);
                   txtdesig.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[7]);
                    txtbasic.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[8]);
                    txtda.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[9]);
                    txthra.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[10]);
                    txtpf.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[11]);
                    txttax.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[12]);
                    txtstatus.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[13]);
                    c.cmd.ExecuteNonQuery();
                }
                else
                {
                    MessageBox.Show("Record Not Found");
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
```

```
if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtbasic.Text = "";
        }
        else
        {
            txtbasic.Text = DropDownList1.SelectedValue;
            txtdesig.Text = DropDownList1.SelectedItem.Text;
        }

    }
}
```

Employee_Display

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
public partial class empdisplay : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            c = new connect();
            c.cmd.CommandText = "select
empid,empname,dob,contact,address,email,doj,desig,basic,da,hra,pf,tax from
emp where status='"+"Active"+"'";
            ds = new DataSet();
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                GridView1.DataSource = ds.Tables["em"];
                GridView1.DataBind();
            }
            else
```

```
      {
                MessageBox.Show("No Records");
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
}
```

Employee Delete

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows.Forms;
using System.Data.SqlClient;
public partial class empdelete : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            c = new connect();
            c.cmd.CommandText = "select
empid,empname,dob,contact,address,email,doj,desig,basic,da,hra,pf,tax from
emp where status='" + "Active" + "'";
            ds = new DataSet();
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "emp");
            if (ds.Tables["emp"].Rows.Count > 0)
            {
                GridView1.DataSource = ds.Tables["emp"];
                GridView1.DataBind();
            }
            else



 {
```

```
                MessageBox.Show("No Records");
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void btnsearch_Click(object sender, EventArgs e)
    {
        if (txtsearch.Text == "")
        {
            MessageBox.Show("Enter Emplooyee ID");
        }
        else
        {
            try
            {
                string r = "Inactive";
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from emp where empid ='" +
txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "emp");
                if (ds.Tables["emp"].Rows.Count > 0)
                {
                    c.cmd.CommandText = "delete from emp where empid='" +
txtsearch.Text + "'";
                    MessageBox.Show("Record Deleted");
                    c.cmd.CommandText = "update emp set status=@status where
empid='" + txtsearch.Text + "'";
                    c.cmd.Parameters.Add("@status", SqlDbType.NVarChar).Value
= Convert.ToString(r);
                    c.cmd.ExecuteNonQuery();
                }
                else
                {
                    MessageBox.Show("Record does not exist");
                    txtsearch.Text = "";
                }
            }
            catch
            {
                throw;
            }



 finally
            {
```

```
                c.cnn.Close();
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        txtsearch.Text = "";
    }
}
```

Employee_Atendence

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows .Forms ;
using System.Data.SqlClient ;
using System.Text.RegularExpressions;
public partial class empattendance : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (txtname.Text == "")
        {
            Panel1.Visible = false;
        }
        lblyr.Visible = false;
        lblmn.Visible = false;
        lbldate.Text = Convert.ToString(DateTime.Now.Day);
        lblmonth.Text = Convert.ToString(DateTime.Now.Month);
        lblyear.Text = Convert.ToString(DateTime.Now.Year);
        switch (Convert.ToInt32(lblmonth.Text))
        {
            case 1: lblmonth.Text = "Jan";
                break;
            case 2: lblmonth.Text = "Feb";
                break;
            case 3: lblmonth.Text = "March";
                break;
```

```
  case 4: lblmonth.Text = "April";
                break;
            case 5: lblmonth.Text = "May";
```

```
                break;
            case 6: lblmonth.Text = "June";
                break;
            case 7: lblmonth.Text = "July";
                break;
            case 8: lblmonth.Text = "Aug";
                break;
            case 9: lblmonth.Text = "Sep";
                break;
            case 10: lblmonth.Text = "Oct";
                break;
            case 11: lblmonth.Text = "Nov";
                break;
            case 12: lblmonth.Text = "Dec";
                break;
        }
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [empid] FROM [emp] where
status='"+"Active"+"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["em"].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables["em"].Rows[i].ItemArray[0].ToString());
                }

            }
        }

    }

    protected void btncal_Click(object sender, EventArgs e)
    {
        lblmn.Visible = true;
        lblyr.Visible = true;
        Double p = Convert.ToDouble (txttotalday.Text);
        Double k = Convert.ToDouble (txtleaveassigned.Text);
        int n = txtleavetaken.Text.Length;
        n--;




    Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtleavetaken.Text == "")
        {
```

```
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the leave taken')</script>");

            txtleavetaken.Focus();
        }
        else if (con.IsMatch(txtleavetaken.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");

            txtleavetaken.Text = "";
            txtleavetaken.Focus();
        }
        else if (Convert .ToDouble  (txtleavetaken .Text )> p)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Check the leave')</script>");

            txtleavetaken.Focus();
        }

        else
        {
            Double w = Convert.ToDouble (txtleavetaken.Text);
            Double s = k - w;
            Double t = p - w;
            if (w > k)
            {
                Double r = w - k;
                txtextra.Text = r.ToString();
            }
            else
            {
                txtextra.Text = "0";
            }
            txtworking.Text = t.ToString();
        }

    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (txtname.Text == "" || txtdesig.Text == "" || txttotalday.Text ==
"" || txtleaveassigned.Text == "" || txtleavetaken.Text == "" ||
txtworking.Text == "" || txtextra.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all fields')</script>");

        }



 else
        {
            try
            {
```

```
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "insert into attendance
values(@date,@month,@year,@totaldays,@leaveassigned,@leavetaken,@workingdays,
@empid)";
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
                c.cmd.Parameters.Add("@month", SqlDbType.NVarChar).Value =
lblmonth.Text;
                c.cmd.Parameters.Add("@year", SqlDbType.NVarChar).Value =
lblyear.Text;
                c.cmd.Parameters.Add("@totaldays", SqlDbType.NVarChar).Value
= txttotalday.Text;
                c.cmd.Parameters.Add("@leaveassigned",
SqlDbType.NVarChar).Value = txtleaveassigned.Text;
                c.cmd.Parameters.Add("@leavetaken", SqlDbType.NVarChar).Value
= txtleavetaken.Text;
                c.cmd.Parameters.Add("@workingdays",
SqlDbType.NVarChar).Value = txtworking.Text;
                c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value =
DropDownList1.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record inserted')</script>");

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtname.Text = "";
            txtdesig.Text = "";
            Panel1.Visible = false;
            DropDownList1.Focus();




    }
        else
        {
            try
            {

                c = new connect();
```

```
                ds = new DataSet();
                c.cmd.CommandText = "select * from emp where empid='" +
DropDownList1.SelectedItem + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "emp");
                if (ds.Tables["emp"].Rows.Count > 0)
                {
                    c.cmd.CommandText = "select * from attendance where
empid='" + DropDownList1.SelectedItem + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "att");
                    if (ds.Tables["att"].Rows.Count > 0)
                    {

Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('You already given the attendence')</script>");

                        txtname.Text = "";
                        txtdesig.Text = "";
                        Panel1.Visible = false;
                    }
                    else
                    {
                        Panel1.Visible = true;
                        // txtid.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[0]);
                        txtname.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[1]);
                        txtdesig.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[7]);
                    }
                }
                else
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Employee details dpes not exists')</script>");

                }
                long m, y;
                txtleaveassigned.Text = "4";
                m = Convert.ToInt32(DateTime.Now.Month);
            y = Convert.ToInt32(DateTime.Now.Year);
                if (m == 1)
                {
                    m = 12;
                    y = y - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                else
                {
                    m = m - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                switch (m)
                {
                    case 1: lblmn.Text = "Jan";
```

```
                    txttotalday.Text = "31";
                    break;
                case 2: lblmn.Text = "Feb";
                    txttotalday.Text = "28";
                    break;
                case 3: lblmn.Text = "March";
                    txttotalday.Text = "31";
                    break;
                case 4: lblmn.Text = "April";
                    txttotalday.Text = "30";
                    break;
                case 5: lblmn.Text = "May";
                    txttotalday.Text = "31";
                    break;
                case 6: lblmn.Text = "June";
                    txttotalday.Text = "30";
                    break;
                case 7: lblmn.Text = "July";
                    txttotalday.Text = "31";
                    break;
                case 8: lblmn.Text = "Aug";
                    txttotalday.Text = "31";
                    break;
                case 9: lblmn.Text = "Sep";
                    txttotalday.Text = "30";
                    break;
                case 10: lblmn.Text = "Oct";
                    txttotalday.Text = "31";
                    break;
                case 11: lblmn.Text = "Nov";
                    txttotalday.Text = "30";
                    break;
                case 12: lblmn.Text = "Dec";
                    txttotalday.Text = "31";
                    break;
            }




        lblmn.Visible = true;
                lblyr.Visible = true;

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
         }
    }
}
```

Employee_Salary

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Windows .Forms ;
using System.Data.SqlClient ;
using System.Text.RegularExpressions;
public partial class empattendance : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (txtname.Text == "")
        {
            Panel1.Visible = false;
        }
        lblyr.Visible = false;
        lblmn.Visible = false;
        lbldate.Text = Convert.ToString(DateTime.Now.Day);
        lblmonth.Text = Convert.ToString(DateTime.Now.Month);
        lblyear.Text = Convert.ToString(DateTime.Now.Year);
        switch (Convert.ToInt32(lblmonth.Text))
        {
            case 1: lblmonth.Text = "Jan";
                break;



       case 2: lblmonth.Text = "Feb";
                break;
            case 3: lblmonth.Text = "March";
                break;
            case 4: lblmonth.Text = "April";
                break;
            case 5: lblmonth.Text = "May";
                break;
            case 6: lblmonth.Text = "June";
                break;
            case 7: lblmonth.Text = "July";
                break;
            case 8: lblmonth.Text = "Aug";
                break;
            case 9: lblmonth.Text = "Sep";
                break;
```

```
            case 10: lblmonth.Text = "Oct";
                break;
            case 11: lblmonth.Text = "Nov";
                break;
            case 12: lblmonth.Text = "Dec";
                break;
        }
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT DISTINCT [empid] FROM [emp] where
status='"+"Active"+"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "em");
            if (ds.Tables["em"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["em"].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables["em"].Rows[i].ItemArray[0].ToString());
                }

            }
        }

    }

    protected void btncal_Click(object sender, EventArgs e)
    {
        lblmn.Visible = true;
        lblyr.Visible = true;
        Double p = Convert.ToDouble (txttotalday.Text);




Double k = Convert.ToDouble (txtleaveassigned.Text);
        int n = txtleavetaken.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtleavetaken.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the leave taken')</script>");

            txtleavetaken.Focus();
        }
        else if (con.IsMatch(txtleavetaken.Text) == false)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter only valid number')</script>");
```

```
            txtleavetaken.Text = "";
            txtleavetaken.Focus();
        }
        else if (Convert .ToDouble  (txtleavetaken .Text )> p)
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Check the leave')</script>");

            txtleavetaken.Focus();
        }

        else
        {
            Double w = Convert.ToDouble (txtleavetaken.Text);
            Double s = k - w;
            Double t = p - w;
            if (w > k)
            {
                Double r = w - k;
                txtextra.Text = r.ToString();
            }
            else
            {
                txtextra.Text = "0";
            }
            txtworking.Text = t.ToString();
        }

    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (txtname.Text == "" || txtdesig.Text == "" || txttotalday.Text ==
"" || txtleaveassigned.Text == "" || txtleavetaken.Text == "" ||
txtworking.Text == "" || txtextra.Text == "")




 {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all fields')</script>");

        }
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "insert into attendance
values(@date,@month,@year,@totaldays,@leaveassigned,@leavetaken,@workingdays,
@empid)";
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
                c.cmd.Parameters.Add("@month", SqlDbType.NVarChar).Value =
lblmonth.Text;
```

```
                c.cmd.Parameters.Add("@year", SqlDbType.NVarChar).Value =
lblyear.Text;
                c.cmd.Parameters.Add("@totaldays", SqlDbType.NVarChar).Value
= txttotalday.Text;
                c.cmd.Parameters.Add("@leaveassigned",
SqlDbType.NVarChar).Value = txtleaveassigned.Text;
                c.cmd.Parameters.Add("@leavetaken", SqlDbType.NVarChar).Value
= txtleavetaken.Text;
                c.cmd.Parameters.Add("@workingdays",
SqlDbType.NVarChar).Value = txtworking.Text;
                c.cmd.Parameters.Add("@empid", SqlDbType.NVarChar).Value =
DropDownList1.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Record inserted')</script>");

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)

{
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtname.Text = "";
            txtdesig.Text = "";
            Panel1.Visible = false;
            DropDownList1.Focus();
        }
        else
        {
            try
            {

                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from emp where empid='" +
DropDownList1.SelectedItem + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "emp");
                if (ds.Tables["emp"].Rows.Count > 0)
                {
                    c.cmd.CommandText = "select * from attendance where
empid='" + DropDownList1.SelectedItem + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "att");
                    if (ds.Tables["att"].Rows.Count > 0)
```

```
                    {

Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('You already given the attendence')</script>");

                        txtname.Text = "";
                        txtdesig.Text = "";
                        Panel1.Visible = false;
                    }
                    else
                    {
                        Panel1.Visible = true;
                        // txtid.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[0]);
                        txtname.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[1]);
                        txtdesig.Text =
Convert.ToString(ds.Tables["emp"].Rows[0].ItemArray[7]);
                    }
                }
                else
                {
                    Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Employee details dpes not exists')</script>");

                }



                //lblyear.Text = Convert.ToString(DateTime.Now.Year);
                long m, y;
                txtleaveassigned.Text = "4";
                m = Convert.ToInt32(DateTime.Now.Month);
                y = Convert.ToInt32(DateTime.Now.Year);
                if (m == 1)
                {
                    m = 12;
                    y = y - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                else
                {
                    m = m - 1;
                    lblyr.Text = Convert.ToString(y);
                }
                switch (m)
                {
                    case 1: lblmn.Text = "Jan";
                        txttotalday.Text = "31";
                        break;
                    case 2: lblmn.Text = "Feb";
                        txttotalday.Text = "28";
                        break;
                    case 3: lblmn.Text = "March";
                        txttotalday.Text = "31";
```

```
                    break;
                case 4: lblmn.Text = "April";
                    txttotalday.Text = "30";
                    break;
                case 5: lblmn.Text = "May";
                    txttotalday.Text = "31";
                    break;
                case 6: lblmn.Text = "June";
                    txttotalday.Text = "30";
                    break;
                case 7: lblmn.Text = "July";
                    txttotalday.Text = "31";
                    break;
                case 8: lblmn.Text = "Aug";
                    txttotalday.Text = "31";
                    break;
                case 9: lblmn.Text = "Sep";
                    txttotalday.Text = "30";
                    break;
                case 10: lblmn.Text = "Oct";
                    txttotalday.Text = "31";
                    break;
                case 11: lblmn.Text = "Nov";
                    txttotalday.Text = "30";




    break;
                case 12: lblmn.Text = "Dec";
                    txttotalday.Text = "31";
                    break;
            }
            lblmn.Visible = true;
            lblyr.Visible = true;

        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
}
```

Product

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data .SqlClient ;
using System.Windows .Forms ;
using System.Text.RegularExpressions;
public partial class menu : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp= new SqlDataAdapter ();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblid.Visible = false;
        try
        {
            c=new connect();
            c.cmd .CommandText ="select pid,name,unit,price,category,otc from
product";
            ds=new DataSet ();
            adp.SelectCommand =c.cmd ;
            adp.Fill (ds,"prod");
            if(ds.Tables ["prod"].Rows .Count >0)
            {
                GridView1 .DataSource =ds.Tables ["prod"];
                GridView1 .DataBind();
            }
            else
            {
                MessageBox .Show ("No Records");
            }
        }
        catch (Exception )
        {
            throw ;
        }
        finally
        {
            c.cnn .Close ();
        }
    }
    protected void btnclear_Click(object sender, EventArgs e)




 {
```

```
        lblid .Text = "";
        txtname.Text = "";
        txtunit.Text = "";
        txtprice.Text = "";
        DropDownList1.Text = "";
       // txtminqty.Text = "";
       // txtqtyonhand.Text = "";

    }
    protected void btnedit_Click(object sender, EventArgs e)
    {
        Regex con = new Regex("^[1-9][0-9]*");
        if (lblid.Text == "")
        {
            lblmsg.Text = "Search";
        }
        else if (txtname.Text == "" || txtunit.Text == "" || txtprice.Text ==
"" || DropDownList1.SelectedItem.Text == "" || txtotc.Text == "")
        {
            lblmsg.Text = "Enter all the fields";
        }
        else if (con.IsMatch(txtprice .Text ) == false)
        {
            lblprice.Visible = true;
            lblprice.Text  = "Only valid number";
        }
        else
        {
        try
        {
            lblmsg.Text = "";
            lblprice.Text = "";
            c = new connect();
            c.cmd.CommandText = "update product set
name=@name,unit=@unit,price=@price,category=@category where pid=@pid";
            c.cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value =
lblid.Text;
            c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
            c.cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value =
txtunit.Text;
            c.cmd.Parameters.Add("@price", SqlDbType.BigInt).Value =
Convert.ToInt16(txtprice.Text);
            c.cmd.Parameters.Add("@category", SqlDbType.NVarChar).Value =
DropDownList1.Text;
            c.cmd .Parameters .Add ("@otc",SqlDbType .NVarChar ).Value
=txtotc .Text ;
            c.cmd.ExecuteNonQuery();

           lblmsg .Text ="Product Updated";



 }
        catch (Exception)
        {
```

```
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
        }
    }
    protected void btnsearch_Click(object sender, EventArgs e)
    {
        if (txtcake.Text == "" || txtsearch.Text == "")
        {
            lblsearch.Text = "Enter the fields";
        }
        else
        {
            lblsearch.Text = "";
            lblid.Visible = true;
            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from product where name='" +
txtcake.Text + "' and category='" + txtsearch.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    lblid.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[0]);
                    txtname.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[1]);
                    txtunit.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[2]);
                    txtprice.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[3]);
                    DropDownList1.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[4]);
                    txtotc.Text =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[7]);
                    c.cmd.ExecuteNonQuery();
                }
                else
                {
                    MessageBox.Show("Record Not Found");
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
```

```
    }

    protected void txtprice_TextChanged(object sender, EventArgs e)
    {

    }
}
```

New product

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Text.RegularExpressions;

public partial class newproduct : System.Web.UI.Page
{
    connect c;
    SqlDataAdapter adp = new SqlDataAdapter();

    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Visible = false;
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        Regex con = new Regex("^[1-9][0-9]*");
        if (txtname.Text == "" || txtprice.Text == "" || DropDownList1
.SelectedItem .Text ==""|| DropDownList2 .SelectedItem .Text ==""||
DropDownList3 .SelectedItem .Text ==""  )
        {
            MessageBox.Show("Enter all the fields");




  }
        else if (con.IsMatch(txtprice.Text ) == false)
        {
            Label1.Visible = true;
            Label1 .Text = "Only valid Price";
            txtprice.Focus();
        }
        else
        {
            try
            {
                Label1.Text = "";
```

```
                c = new connect();
                c.cmd.CommandText = "insert into product
values(@pid,@name,@unit,@price,@category,@minqty,@qtyonhand,@otc)";
                c.cmd.Parameters.Add("@pid", SqlDbType.NVarChar).Value =
txtid.Text;
                c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
                c.cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value
=DropDownList2 .SelectedItem .Text ;
                c.cmd.Parameters.Add("@price", SqlDbType.BigInt).Value =
Convert.ToInt64(txtprice.Text);
                c.cmd.Parameters.Add("@category", SqlDbType.NVarChar).Value =
DropDownList1.Text;
                c.cmd.Parameters.Add("@minqty", SqlDbType.SmallInt).Value =
2;
                c.cmd.Parameters.Add("@qtyonhand", SqlDbType.SmallInt).Value
= 0;
                c.cmd.Parameters.Add("@otc", SqlDbType.NVarChar).Value =
DropDownList3.SelectedItem.Text;
                c.cmd.ExecuteNonQuery();
                Label1.Visible = true;
                Label1.Text = "Product Inserted";
                txtid.Text = "";
                txtname.Text= "";
                txtprice.Text = "";
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnclear_Click1(object sender, EventArgs e)
    {




        txtid.Text = "";
        txtname.Text = "";
        txtprice.Text = "";
        Label1.Visible = false;
    }

    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        try
        {
            c = new connect();
            String egg = "EG", eggless = "EL", pastries = "P", snacks = "S";
```

```
            if (IsPostBack)
            {
                if (DropDownList1.SelectedItem.Text == "Egg Cake")
                {
                    c.cmd.CommandText = "select count(pid) from product where
pid like 'EG%'";
                    int i = Convert.ToInt32(c.cmd.ExecuteScalar());
                    i = i + 1001;
                    txtid.Text = egg + i.ToString();
                }
                else if (DropDownList1.SelectedItem.Text == "Eggless Cake")
                {
                    c.cmd.CommandText = "select count(pid) from product where
pid like 'EL%'";
                    int j = Convert.ToInt32(c.cmd.ExecuteScalar());
                    j = j + 1001;
                    txtid.Text = eggless + j.ToString();
                }
                else if (DropDownList1.SelectedItem.Text == "Pastries")
                {
                    c.cmd.CommandText = "select count(pid) from product where
pid like 'P%'";
                    int k = Convert.ToInt32(c.cmd.ExecuteScalar());
                    k  = k + 1001;
                    txtid.Text = pastries + k.ToString();
                }
                else
                {
                    c.cmd.CommandText = "select count(pid) from product where
pid like 'S%'";
                    int l = Convert.ToInt32(c.cmd.ExecuteScalar());
                    l= l  + 1001;
                    txtid.Text = snacks + l.ToString();
                }
            }
        }
        catch (Exception)




    {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }

    }
    protected void txtprice_TextChanged(object sender, EventArgs e)
    {

    }
}
```

Production details

Code :

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Text.RegularExpressions;
public partial class productiondetails : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();




 protected void Page_Load(object sender, EventArgs e)
    {
        c = new connect();
        btnsdone.Visible = false;
        btnadd.Visible = false;
        btndone.Visible = false;
        Label3.Visible = false;
        Label2.Visible = false;
        Label1.Visible = false;
        Label5.Visible = false;
        CheckBox1.Visible = false;
        Label6.Visible = false;
        CheckBox3.Visible = false;

        Panel1.Visible = false;
        Panel2.Visible = false;
        lbldate.Text = DateTime.Today.ToString("dd'/'MM'/'yyyy");
        String dy = Convert.ToString(DateTime.Now.Day);
        int d = Convert.ToInt32(dy);
        d++;
        Label2.Text = Convert.ToString(DateTime.Now.Month);
        Label3.Text = Convert.ToString(DateTime.Now.Year);
        Label1.Text = d.ToString();
        Label3.Text = Label1.Text + "/0" + Label2.Text + "/" + Label3.Text;

        ds = new DataSet();
        if (DropDownList3.Items.Count == 0)
        {
```

```
            c.cmd.CommandText = "SELECT DISTINCT [category] FROM [product]";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cat");
            if (ds.Tables["cat"].Rows.Count > 0)
            {
                DropDownList3.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                {

DropDownList3.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                }
            }
        }
        if (DropDownList2.Items.Count == 0)
        {
            c.cmd.CommandText = "SELECT [cbid] FROM [custbook] WHERE
[duedate] = '"+ Label3 .Text +"' AND [mfd] = '"+"No"+"'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "ct");
            if (ds.Tables["ct"].Rows.Count > 0)
            {
                DropDownList2.Items.Add("---Select---");
                int i;




    for (i = 0; i < ds.Tables["ct"].Rows.Count; i++)
                {

DropDownList2.Items.Add(ds.Tables["ct"].Rows[i].ItemArray[0].ToString());
                }
            }
            else
            {
                DropDownList2.Items.Add("---Select---");
                lblcbook.Text = "No Booking";
            }
        }
        if (DropDownList4.Items.Count == 0)
        {
            c.cmd.CommandText = "SELECT [bbid] FROM [branchbook] WHERE
[duedate] ='" + Label3.Text + "' AND [mfd] = '" + "No" + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "ca");
            if (ds.Tables["ca"].Rows.Count > 0)
            {
                DropDownList4.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["ca"].Rows.Count; i++)
                {

DropDownList4.Items.Add(ds.Tables["ca"].Rows[i].ItemArray[0].ToString());
                }
            }
```

```
            else
            {
                DropDownList4.Items.Add("---Select---");
                lblbbook.Text = "No Booking";
            }
        }

        if (!IsPostBack)
        {
            GenerateId();
        }
    }
    private void GenerateId()
    {
        String bill = "PD";
        c = new connect();
        c.cmd.CommandText = "select count(pbid) from purchasebill";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;
        lblid.Text = bill + i.ToString();
    }
```

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {

        try
        {
            c = new connect();
            ds = new DataSet();
            if (IsPostBack)
            {
                c.cmd.CommandText = "select * from product where name ='" +
DropDownList1.SelectedItem.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "prod");
                if (ds.Tables["prod"].Rows.Count > 0)
                {
                    txtid .Text  =
Convert.ToString(ds.Tables["prod"].Rows[0].ItemArray[0]);
                    c.cmd.ExecuteNonQuery();
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
```

```
    }
    protected void DropDownList3_SelectedIndexChanged(object sender,
EventArgs e)
    {

        Panel1.Visible = true;

        if (GridView1.Rows.Count > 0)
        {
            btnsdone.Visible = true;
        }
        try
        {
            if (DropDownList3.SelectedItem.Text == "---Select---")
            {
                txtid.Text = "";
                txtqty.Text = "";
                btnadd.Visible = false;
                DropDownList1.Items.Clear();
                DropDownList1.Items.Add("---Select---");

            }
            else



 {
                DropDownList1.Items.Clear();
                c.cmd.CommandText = "SELECT [name], [pid] FROM [product]
WHERE [category] = '" + DropDownList3.SelectedItem.Text + "' AND [otc] = '" +
"True" + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cat");
                if (ds.Tables["cat"].Rows.Count > 0)
                {
                    DropDownList1.Items.Add("---Select---");
                    int i;
                    for (i = 0; i < ds.Tables["cat"].Rows.Count; i++)
                    {

DropDownList1.Items.Add(ds.Tables["cat"].Rows[i].ItemArray[0].ToString());
                    }
                }
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn .Close ();
        }
    }
    protected void DropDownList1_SelectedIndexChanged1(object sender,
EventArgs e)
```

```
    {
        txtqty.Text = "";
        Panel1.Visible = true;
        btnadd.Visible = true;
        if (GridView1.Rows.Count > 0)
        {
            btnsdone.Visible = true;
        }
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtid.Text = "";
            txtqty.Text = "";
            btnadd.Visible = false;

        }
        else
        {









  try
            {
                c = new connect();
                c.cmd.CommandText = "select * from product where name='" +
DropDownList1.SelectedItem.Text + "' and category='" +
DropDownList3.SelectedItem.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "pro");
                if (ds.Tables["pro"].Rows.Count > 0)
                {
                    txtid.Text =
Convert.ToString(ds.Tables["pro"].Rows[0].ItemArray[0]);

                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        //Label4.Visible = true;
        //CheckBox2.Visible = true;
        Panel1.Visible = true;
```

```
        int n = txtqty.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{" + n + "}");
        if (txtqty.Text == "" || txtid.Text == "")
        {
            MessageBox.Show("Enter all the fields");
            btnadd.Visible = true;
        }
        else if (con.IsMatch(txtqty.Text) == false)
        {
            //lbladvance.Visible = true;
            MessageBox.Show("Enter valid number");
            txtqty.Text = "";
            btnadd.Visible = true;
        }
        else
        {
            btnsdone.Visible = true;
            try
            {


c = new connect();


                dt.Columns.Add("Product ID");
                dt.Columns.Add("Name");
                dt.Columns.Add("Quantity");
                DataRow dr = null;
                if (ViewState["pro"] != null)
                {
                    for (int i = 0; i < 1; i++)
                    {
                        dt = (DataTable)ViewState["pro"];
                        if (dt.Rows.Count > 0)
                        {
                            dr = dt.NewRow();
                            dr["Product ID"] = txtid.Text;
                            dr["Name"] = DropDownList1.SelectedItem.Text; ;
                            dr["Quantity"] = txtqty.Text;
                            dt.Rows.Add(dr);
                            GridView1.DataSource = dt;
                            GridView1.DataBind();
                        }
                    }
                }
                else
                {
                    dr = dt.NewRow();
                    dr["Product ID"] = txtid.Text;
                    dr["Name"] = DropDownList1.SelectedItem.Text;
                    dr["Quantity"] = txtqty.Text;
                    dt.Rows.Add(dr);
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                }
```

```
                ViewState["pro"] = dt;
                txtid.Text = "";
                txtqty.Text = "";
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }

    protected void DropDownList4_SelectedIndexChanged(object sender,
EventArgs e)
    {
        Panel2.Visible = true;
        btndone.Visible = true;
        Label6.Visible = true ;
        CheckBox3.Visible = true ;
        GridView2.Visible = true;
        btnadd.Visible = true;
        if (DropDownList4.SelectedItem.Text == "---Select---")
        {
           btndone .Visible =false;
           Label6.Visible = false  ;
        CheckBox3.Visible = false ;
            dt.Rows.Clear();
            dt.Clear();
            GridView2.DataSource = dt;
            GridView2.DataBind();


        }
        else
        {

        try
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "select bbid,product.pid,name,qty from
product,bbookdet where product.pid=bbookdet.pid and bbookdet.bbid='" +
DropDownList4 .SelectedItem .Text  + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cu");
            if (ds.Tables["cu"].Rows.Count > 0)
            {
                GridView2.DataSource = ds.Tables["cu"];
                GridView2.DataBind();
            }

        }
        catch (Exception)
```

```
        {

        throw;
        }
        finally
        {
        }
    }
    }
    protected void DropDownList2_SelectedIndexChanged(object sender,
EventArgs e)
    {
        btndone.Visible = true;
        Label5.Visible = true;
        CheckBox1.Visible = true;
        Panel2.Visible = true;
        btnadd.Visible = true;
        CheckBox1.Checked = false;
        if (DropDownList2.SelectedItem.Text == "---Select---")
        {
            btndone.Visible = false;
            Label5.Visible = false;
            CheckBox1.Visible = false;
            CheckBox3.Checked = false;
            dt.Rows.Clear();
            dt.Clear();
            GridView2.DataSource = dt;
            GridView2.DataBind();


        }
        else
        {

            try
            {
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
DropDownList2.SelectedItem.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "cus");
                if (ds.Tables["cus"].Rows.Count > 0)
                {
                    GridView2.DataSource = ds.Tables["cus"];
                    GridView2.DataBind();
                }
            }

catch (Exception)
            {
                throw;
            }
            finally
            {
```

```
                c.cnn.Close();
            }
        }
    }
    protected void btndone_Click(object sender, EventArgs e)
    {
        if (CheckBox1.Checked == true   || CheckBox3.Checked == true )
        {


            lnkdaily.Visible = true;
            lnkbookinh.Visible = true;
            if (CheckBox1.Checked == true)
            {
                String y = "Yes";
                c = new connect();
                c.cmd.CommandText = "update custbook set mfd=@mfd where
cbid='" + DropDownList2.SelectedItem.Text + "'";
                c.cmd.Parameters.Add("@mfd", SqlDbType.NVarChar).Value =
 y.ToString();
                c.cmd.ExecuteNonQuery();
            }
            else if (CheckBox3.Checked == true)
            {
                String x = "Yes";
                c = new connect();
                c.cmd.CommandText = "update branchbook set mfd=@mfd where
bbid='" + DropDownList4.SelectedItem.Text + "'";
                c.cmd.Parameters.Add("@mfd", SqlDbType.NVarChar).Value =
 x.ToString();
                c.cmd.ExecuteNonQuery();
            }
            MessageBox.Show("Booking production submitted");
            Response.Redirect(Request.Url.AbsoluteUri);
        }
    else
        {
            MessageBox.Show("think");
        }

    }
    protected void btnsdone_Click(object sender, EventArgs e)
    {
        Panel1.Visible = false ;

  lnkbookinh.Visible = true;
        lnkdaily.Visible = true;
        try
        {
            c = new connect();
            c.cmd.CommandText = "insert into production
values(@prid,@remark,@date)";
            c.cmd.Parameters.Add("@prid", SqlDbType.NVarChar).Value =
lblid.Text;
            c.cmd.Parameters.Add("@remark", SqlDbType.NVarChar).Value =
txtid.Text;
```

```
            c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
lbldate.Text;
            c.cmd.ExecuteNonQuery();
            for (int i = 0; i < GridView1.Rows.Count; i++)
            {
                c.cmd.CommandText = "insert into productiondet(prid,pid,qty)
values('" + lblid.Text + "','" + GridView1.Rows[i].Cells[0].Text + "','" +
GridView1.Rows[i].Cells[2].Text + "')";

                c.cmd.ExecuteNonQuery();
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
        try
        {
            c = new connect();
            ds = new DataSet();
            for (int j = 0; j < GridView1.Rows.Count; j++)
            {
                c.cmd.CommandText = "select * from product where pid='" +
GridView1.Rows[j].Cells[0].Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "it");
                if (ds.Tables["it"].Rows.Count > 0)
                {
                    for (int i = 0; i < ds.Tables["it"].Rows.Count; i++)
                    {
                        Double q =
Convert.ToInt16(ds.Tables["it"].Rows[i].ItemArray[6]);




                        Double qt =
Convert.ToDouble(GridView1.Rows[j].Cells[2].Text);
                        Double qty = q + qt;
                        c.cmd.CommandText = "update product  set qtyonhand='"
+ Convert.ToInt16(qty) + "'where pid='" + GridView1.Rows[j].Cells[0].Text +
"'";
                        // c.cmd.Parameters.Add("@qtyonhand",
SqlDbType.SmallInt).Value = Convert.ToInt16(qty);
                        c.cmd.ExecuteNonQuery();
                        MessageBox.Show("Daily production submitted");
                    }
                }
                ds.Tables["it"].Clear();
            }
             Response.Redirect(Request.Url.AbsoluteUri);
```

```
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }
    protected void lnkdaily_Click(object sender, EventArgs e)
    {
        Panel1.Visible = true;
        //lnkbookinh.Visible = false;


    }
    protected void lnkbookinh_Click(object sender, EventArgs e)
    {
        Panel2.Visible = true;
        //lnkdaily.Visible = false;
    }
    protected void txtqty_TextChanged(object sender, EventArgs e)
    {
        Panel1.Visible = true;

    }
}
```

Production request

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;




using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
```

```
using System.Windows.Forms;
public partial class productionrequest : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblbb.Visible = false;
        lblcb.Visible = false;
        lbldate1.Visible = false;
        lblmonth.Visible = false;
        lblyear.Visible = false;
        Label1.Visible = false;
        lbldate.Text = DateTime.Today.ToString("dd'/'MM'/'yyyy");
        String dy = Convert.ToString(DateTime.Now.Day);
        int d = Convert.ToInt32(dy);
        d++;
        lblmonth .Text= Convert.ToString(DateTime.Now.Month);
        lblyear.Text = Convert.ToString(DateTime.Now.Year);
        lbldate1.Text = d.ToString();
        Label1.Text = lbldate1.Text + "/0" + lblmonth.Text + "/" +
lblyear.Text;
        try
        {
            c = new connect();
            ds = new DataSet();

            c.cmd.CommandText = "select
cbookdet.pid,product.name,cbookdet.qty,custbook.duedate from
product,cbookdet,custbook where product.pid=cbookdet.pid and
cbookdet.cbid=custbook.cbid and custbook.duedate='" + Label1.Text + "'";
            //c.cmd.CommandText = "select cbid,product.pid,name,qty from
product,cbookdet where product.pid=cbookdet.pid and cbookdet.cbid='" +
lblcbid.Text + "'";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "cus");
            if (ds.Tables["cus"].Rows.Count > 0)
            {
                lblcb.Visible = true;
                GridView1.DataSource = ds.Tables["cus"];
                GridView1.DataBind();
            }
        }




  catch (Exception)
        {
            throw ;
        }
        finally
        {
            c.cnn .Close ();
```

```
        }
        try
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "select
bbookdet.pid,product.name,bbookdet.qty,branchbook.duedate from
product,bbookdet,branchbook where product.pid=bbookdet.pid and
bbookdet.bbid=branchbook.bbid and branchbook.duedate='" + Label1.Text + "'";

            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "bran");
            if (ds.Tables["bran"].Rows.Count > 0)
            {
                lblbb.Visible = true;
                GridView2.DataSource = ds.Tables["bran"];
                GridView2.DataBind();
            }
        }
        catch (Exception)
        {
            throw;
        }
        finally
        {
            c.cnn.Close();
        }
    }


}
```

Item

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data .SqlClient ;
using System.Windows .Forms ;
public partial class item : System.Web.UI.Page
{
    connect c;
    //DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
```

```
        if (!IsPostBack)
        {
            Generateid();
        }
    }
    private void Generateid()
    {
        String item = "I";
        c = new connect();
        c.cmd.CommandText = "select count(iid) from item";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());
        i = i + 1001;









        lblid.Text = item + i.ToString();
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        if (txtname.Text == "" || DropDownList1.SelectedItem.Text == "")
        {
            MessageBox.Show("Enter all the fields");
        }
        else if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            MessageBox.Show("Select the UNIT");
        }
        else
        {
            try
            {
                c = new connect();
                int q = 0, t = 0;
                c.cmd.CommandText = "insert into item
values(@iid,@name,@unit,@qty,@minqty)";
                c.cmd.Parameters.Add("@iid", SqlDbType.NVarChar).Value =
lblid.Text;
                c.cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value =
txtname.Text;
                c.cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value =
DropDownList1.SelectedItem.Text;
                c.cmd.Parameters.Add("@qty", SqlDbType.SmallInt).Value = q;
                c.cmd.Parameters.Add("@minqty", SqlDbType.SmallInt).Value =
t;
                c.cmd.ExecuteNonQuery();
                MessageBox.Show("Item Inserted");
                Generateid();
                txtname.Text = "";
                //DropDownList1.SelectedItem.Text = "";
            }
            catch (Exception)
            {
                throw;
```

```
            }
            finally
            {
                c.cnn.Close();
            }
        }


    }
    protected void DropDownList1_SelectedIndexChanged(object sender,




EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {

        //c=new connect ();
        //c.cmd .CommandText ="delete from item";
        //c.cmd.ExecuteNonQuery();
    }
}
```

Issue

Code:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;




using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
//using System.Windows.Forms;
using System.Text.RegularExpressions;
public partial class issue : System.Web.UI.Page
{
    connect c;
    DataSet ds;
    SqlDataAdapter adp = new SqlDataAdapter();
    DataTable dt = new DataTable();
    int count;
```

```
    protected void Page_Load(object sender, EventArgs e)
    {
        btnsubmit.Visible = false;
        lbldate.Text = DateTime.Today.ToShortDateString();
        if (!IsPostBack)
        {
            GenerateID();
        }
        if (DropDownList1.Items.Count == 0)
        {
            c = new connect();
            ds = new DataSet();
            c.cmd.CommandText = "SELECT [name] FROM [item] ";
            adp.SelectCommand = c.cmd;
            adp.Fill(ds, "it");
            if (ds.Tables["it"].Rows.Count > 0)
            {
                DropDownList1.Items.Add("---Select---");
                int i;
                for (i = 0; i < ds.Tables["it"].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables["it"].Rows[i].ItemArray[0].ToString());
                }

            }
        }
    }
    private void GenerateID()
    {
        String book = "IT";
        c = new connect();
        c.cmd.CommandText = "select count(issueid) from issue";
        int i = Convert.ToInt32(c.cmd.ExecuteScalar());







    i = i + 1001;
        lblissueid.Text = book + i.ToString();
    }
    protected void DropDownList1_SelectedIndexChanged1(object sender,
EventArgs e)
    {
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            txtitemid.Text = "";
            txtqty.Text = "";
            txtunit.Text = "";
            if (GridView1.Rows.Count > 0)
            {
                btnsubmit.Visible = true;
            }
        }
```

```
        else
        {
            try
            {
                c = new connect();
                ds = new DataSet();
                if (IsPostBack)
                {
                    c.cmd.CommandText = "select * from item where name='" +
DropDownList1.SelectedItem.Text + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "it");
                    if (ds.Tables["it"].Rows.Count > 0)
                    {
                        txtitemid.Text =
Convert.ToString(ds.Tables["it"].Rows[0].ItemArray[0]);
                        txtunit.Text =
Convert.ToString(ds.Tables["it"].Rows[0].ItemArray[2]);
                        c.cmd.ExecuteNonQuery();
                    }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        btnsubmit.Visible = true;



  int n = txtqty.Text.Length;
        n--;
        Regex con = new Regex("^[1-9][0-9]{"+n+"}");
        if (DropDownList1.SelectedItem.Text == "---Select---")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Select the item')</script>");
            btnsubmit.Visible = false;
        }
        else if (txtqty.Text == "")
        {
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter the quantity')</script>");
            btnsubmit.Visible = false;
        }
        else if (con.IsMatch(txtqty.Text) == false)
        {
```

```
            lblqty.Visible = true;
            Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Only valid number')</script>");
            btnsubmit.Visible = false;
        }
        else
        {
            try
            {

                lblqty.Visible = false;
                c = new connect();
                ds = new DataSet();
                c.cmd.CommandText = "select * from item where iid='" +
txtitemid.Text + "'";
                adp.SelectCommand = c.cmd;
                adp.Fill(ds, "it");
                if (ds.Tables["it"].Rows.Count > 0)
                {
                    Double s =
Convert.ToDouble(ds.Tables["it"].Rows[0].ItemArray[3]);
                    Double k = Convert.ToDouble(txtqty.Text);
                    if (k > s)
                    {

Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('qty on hand is less')</script>");
                        txtqty.Text = "";
                    }

                    else
                    {
                        c = new connect();



                        dt.Columns.Add("Item ID");
                        dt.Columns.Add("Name");
                        dt.Columns.Add("Quantity");
                        dt.Columns.Add("unit");
                        DataRow dr = null;
                        if (ViewState["it"] != null)
                        {
                            for (int i = 0; i < 1; i++)
                            {
                                dt = (DataTable)ViewState["it"];
                                if (dt.Rows.Count > 0)
                                {
                                    dr = dt.NewRow();
                                    dr["Item ID"] = txtitemid.Text;
                                    dr["Name"] =
DropDownList1.SelectedItem.Text;
                                    dr["Quantity"] = txtqty.Text;
                                    dr["unit"] = txtunit.Text;
                                    dt.Rows.Add(dr);
```

```
                                    GridView1.DataSource = dt;
                                    GridView1.DataBind();

                                }
                            }
                        }
                        else
                        {
                            dr = dt.NewRow();
                            dr["Item ID"] = txtitemid.Text;
                            dr["Name"] = DropDownList1.SelectedItem.Text;
                            dr["Quantity"] = txtqty.Text;
                            dr["unit"] = txtunit.Text;
                            dt.Rows.Add(dr);
                            GridView1.DataSource = dt;
                            GridView1.DataBind();



                        DropDownList1.SelectedItem.Enabled = false;
                         ViewState["it"] = dt;
                         txtqty.Text = "";
                         txtitemid.Text = "";
                         txtunit.Text = "";
                    }
                }
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();



 }
        }
    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (GridView1.Rows.Count < 0)
        {
            if (txtqty.Text == "" || txtunit.Text == "")
            {
             Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
"<script>alert('Enter all the fields')</script>");
            }
        }
        else
        {
            try
            {
```

```
                c = new connect();
                c.cmd.CommandText = "insert into issue
values(@issueid,@date)";
                c.cmd.Parameters.Add("@issueid", SqlDbType.NVarChar).Value =
lblissueid.Text;
                c.cmd.Parameters.Add("@date", SqlDbType.NVarChar).Value =
 lbldate.Text;
                c.cmd.ExecuteNonQuery();
            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();

            }
            try
            {
                c = new connect();
                for (int i = 0; i < GridView1.Rows.Count; i++)
                {
                    c.cmd.CommandText = "insert into
issuedetails(issueid,iid,qty) values('" + lblissueid.Text + "','" +
GridView1.Rows[i].Cells[0].Text + "','" + GridView1.Rows[i].Cells[2].Text +
"')";
                    c.cmd.ExecuteNonQuery();
                }
                Page.ClientScript.RegisterStartupScript(this.GetType(),
"alert", "<script>alert('Item/s issued')</script>");
                DropDownList1.SelectedItem.Text = "---Select---";
            }




   catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
            try
            {
                c = new connect();
                ds = new DataSet();
                for (int j = 0; j < GridView1.Rows.Count; j++)
                {
                    c.cmd.CommandText = "select * from item where iid='" +
GridView1.Rows[j].Cells[0].Text + "'";
                    adp.SelectCommand = c.cmd;
                    adp.Fill(ds, "it");
```

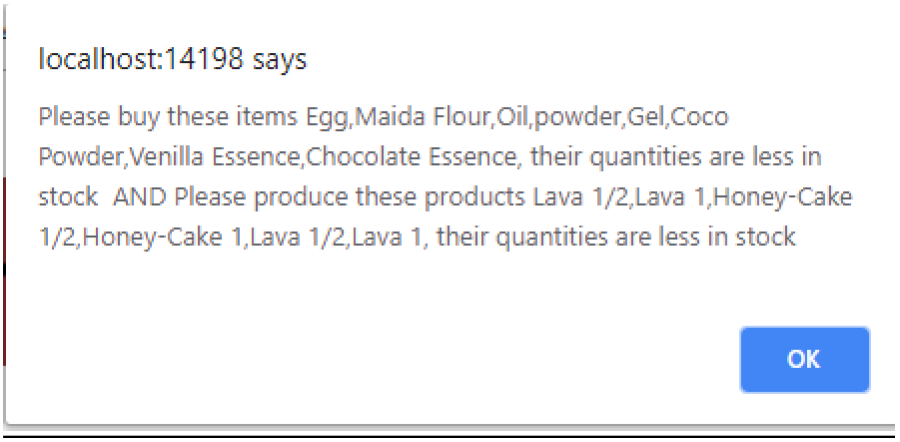
```
                    if (ds.Tables["it"].Rows.Count > 0)
                    {
                        for (int i = 0; i < ds.Tables["it"].Rows.Count; i++)
                        {
                            Double q =
Convert.ToInt16(ds.Tables["it"].Rows[i].ItemArray[3]);
                            Double qt =
Convert.ToDouble(GridView1.Rows[j].Cells[2].Text);
                            Double qty = q - qt;
                            c.cmd.CommandText = "update item  set qty='" +
Convert.ToInt16(qty) + "'where iid='" + GridView1.Rows[j].Cells[0].Text +
"'";
                            // c.cmd.Parameters.Add("@qtyonhand",
SqlDbType.SmallInt).Value = Convert.ToInt16(qty);
                            c.cmd.ExecuteNonQuery();
                        }
                    }
                    ds.Tables["it"].Clear();
                }

            }
            catch (Exception)
            {
                throw;
            }
            finally
            {
                c.cnn.Close();
            }
        }
```

```
  Response.Redirect(Request.Url.AbsoluteUri);

    }
    protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        dt = (DataTable)ViewState["pro"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
    protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs
e)
    {
        if (count == 1)
        {
            e.Row.Cells[1].Enabled = false;
            e.Row.Cells[2].Enabled = false;
            e.Row.Cells[4].Enabled = false;
```

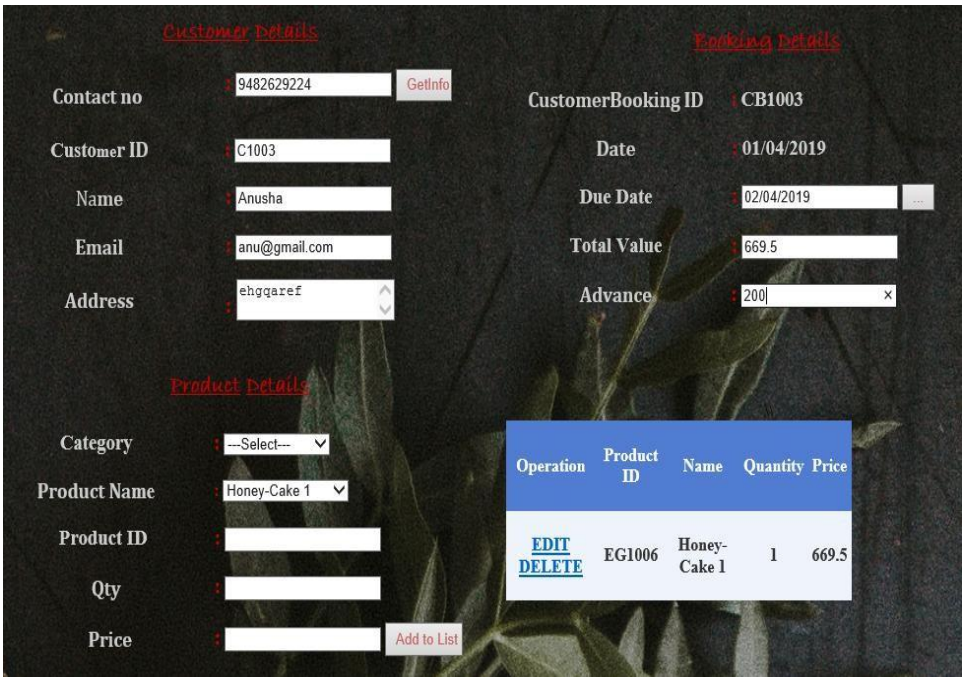
```
        }
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        dt = (DataTable)ViewState["it"];
        dt.Rows[e.RowIndex].Delete();
        GridView1.DataSource = dt;
        GridView1.DataBind();
        if (GridView1.Rows.Count <= 0)
        {
            ViewState["it"] = null;
        }
    }
    protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
    {
        dt = (DataTable)ViewState["it"];
        GridViewRow row = GridView1.Rows[e.RowIndex];
        dt.Rows[row.DataItemIndex]["Quantity"] =
((TextBox)(row.Cells[3].Controls[0])).Text;
        GridView1.EditIndex = -1;
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs
e)
    {

  count = 1;
        GridView1.EditIndex = e.NewEditIndex;
        dt = (DataTable)ViewState["it"];
        GridView1.DataSource = dt;
        GridView1.DataBind();
        GridView1.Visible = true;
    }
}
```

USER MANUAL

Login:



Customer Booking:



Crumbz

A Unit of gyp-gyp-gy
Manipal
Pratham Pride Building,End Point
Road,Manipal Udupi-576104
Ph : 0820 - 2370960

Customer booking ID CB1006

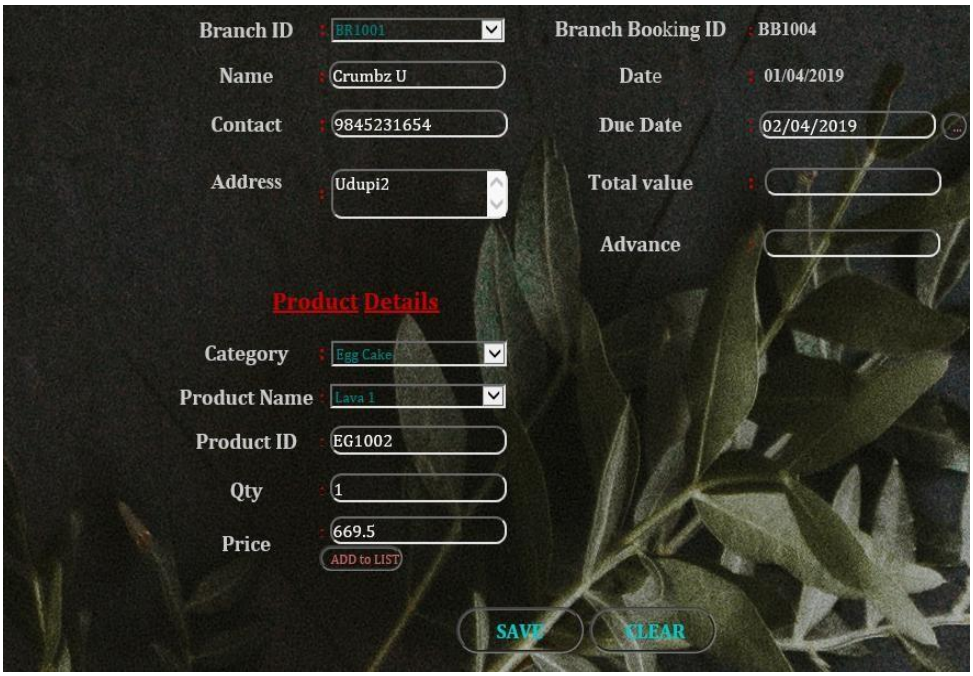
Date 04-04-2019

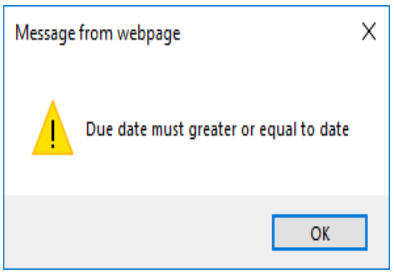
Advance Receipt

Product Name	Quantity	Rate	Amount
Dark-Fantasy 1/2	3	400.00	1236.00
		Net Amount	1236.00
		Advance	1000.00

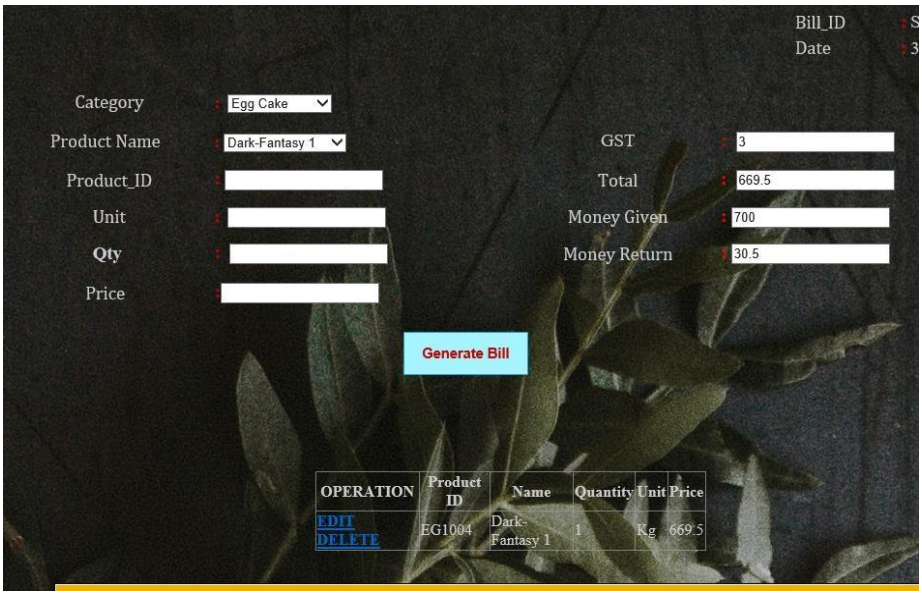
*** Have A Nice Day ***
*** VISIT AGAIN ***

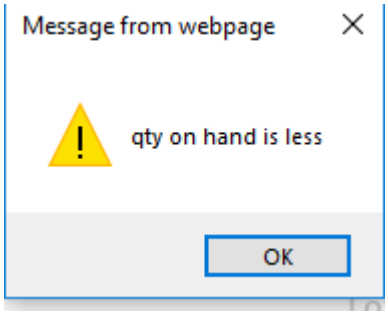
Branch Booking:





Sales Bill:





Crumbz

A Unit of gyp-gyp-gy
Manipal

Pratham Pride Building,End Point
Road,Manipal Udupi-576104
Ph : 0820 - 2370960

Tax Invoice

Bill ID: SB1001 **Date :** 03/31/2019

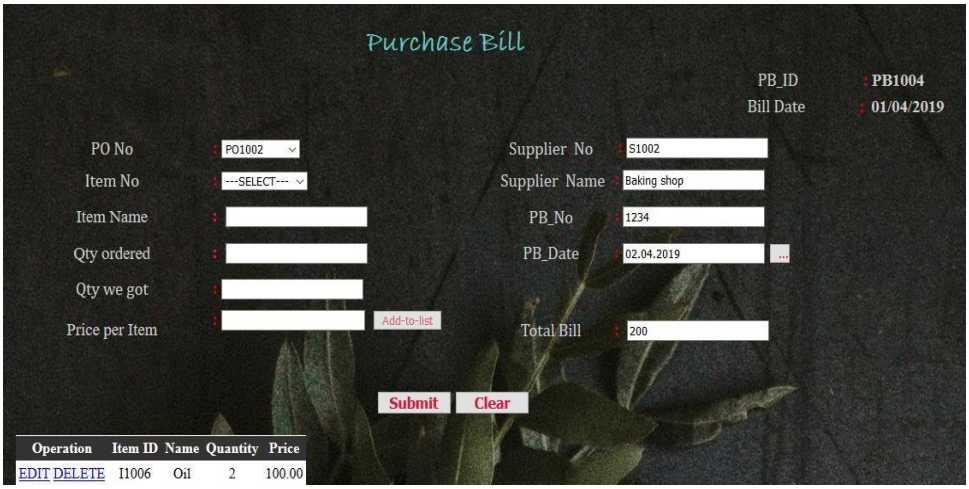
Name	Quantity	Per Item Price	GST	Price
Dark-Fantasy 1	1	650.00	3.00	670.00

Total Bill **670.00**

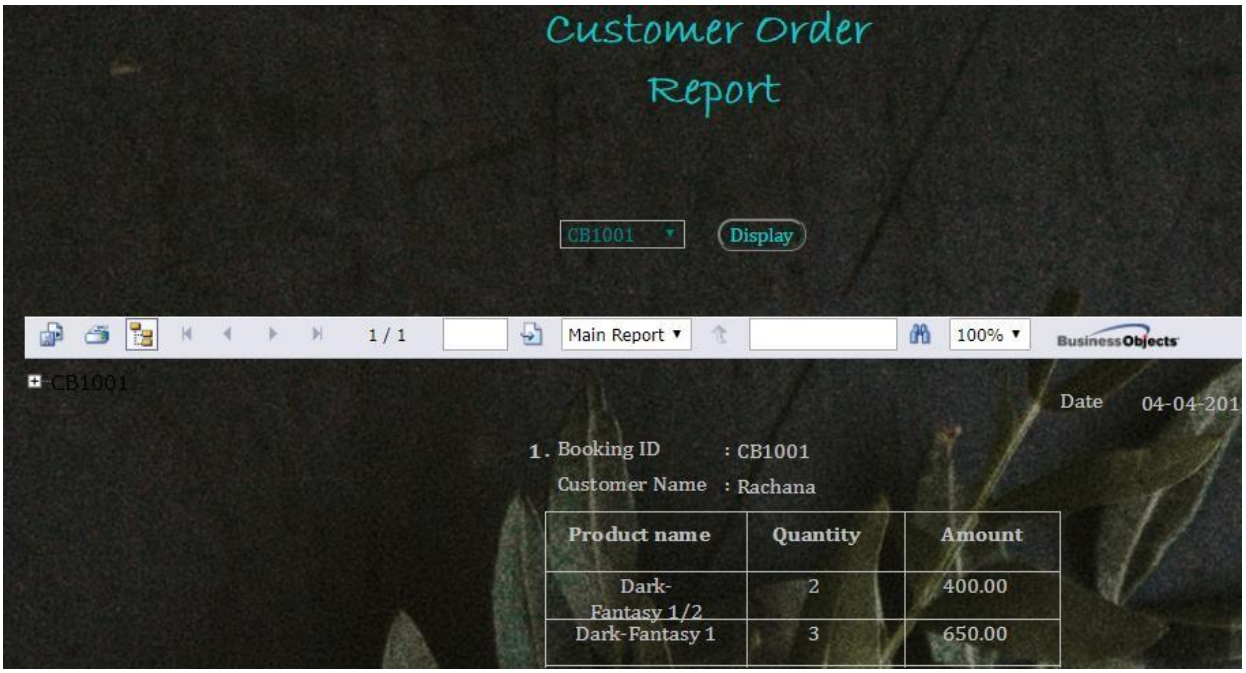
* Have A Nice Day *
* VISIT AGAIN *

Purchase order:

Purchase Bill:



Item Issue:

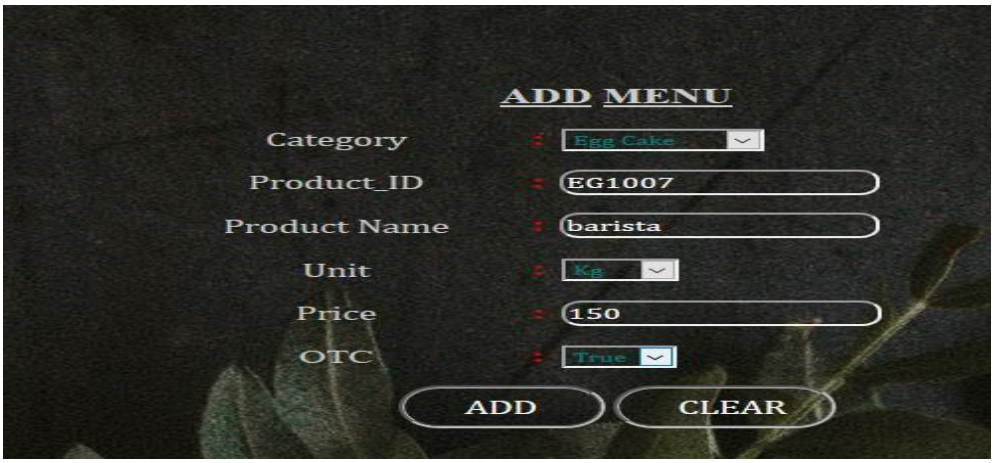

Customer Order
Report
CB1001
Display
1 / 1
Main Report
100%
BusinessObjects
CB1001
Date 04-04-201
1. Booking ID : CB1001
Customer Name : Rachana
Product name | Quantity | Amount
Dark-Fantasy 1/2 | 2 | 400.00
Dark-Fantasy 1 | 3 | 650.00


Lava 1 | 1 | 650.00
Dark-Fantasy 1 | 2 | 650.00
3. Booking ID : CB1003
Customer Name : Anusha a
Product name | Quantity | Amount
Dark-Fantasy 1 | 9 | 650.00
4. Booking ID : CB1004
Customer Name : Amrutha
Product name | Quantity | Amount
Lava 1 | 2 | 650.00
5. Booking ID : CB1005
Customer Name : Amrutha
Product name | Quantity | Amount
Honey-Cake 1/2 | 4 | 400.00
6. Booking ID : CB1006
Customer Name : Amrutha
Product name | Quantity | Amount
Dark-Fantasy 1/2 | 3 | 400.00

Customer update and display

Supplier Registration

Add new Product:



Testing Phases

Testing is the process of detecting errors. Testing performs a very special for quality I assurance and for ensuring the reliability of the software. The results of testing are I used later on maintenance also.

Philosophy of testing:

The aim of testing is often to demonstrate that a program works by showing that it I has no errors. The Basic purpose of testing phase is to detect the errors that may be I present in the program. Hence one should not start testing with the intent to show I that a program doesn't work. Testing is the process of executing a program with a I intent of finding errors.

Testing Objectives;

The main objective of testing is to uncover a host of errors, systematically and I with minimum effort and time, starting formally, we can say, testing is a process of I executing a program with the intent of finding an error. A successful test is one that I uncovers an as yet undiscovered error.

A good test case is one that has a high probability of finding error, is it exist

- The tests are inadequate to detect possibly present errors.
- The software more or less confirm to the quality and reliable standards

System Testing:

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design and coding. The testing phase involves the testing of system using various test data; preparation test data plays a vital role in the system testing. After preparation test data plays a vital role in the system testing. After preparation test data, the system under study is tested.

Those test data, errors were found and corrected by following testing steps and corrections are recorded for future references. Thus, a series testing is performed on the system before it is ready for implementation.

The various types of testing on the system are:

- Unit testing
- Integrated testing
- Validation testing
- Output testing

Unit testing:

Unit focuses verification effort on the smallest unit of software i.e the module. Using detailed design and the process specification testing is done to uncover efforts with in the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of as a module. There are three basic modules. Giving different sets of input has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated, to form a specific function. Unit testing is first done on modules, independent of one another to locate error. This enables to detect errors. Through these errors resulting form the interaction between modules are initially avoided.

Integrated Testing:

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interaction.

In this project integration all the modules, I have checked whether the integration effects working on any of the services by giving different combination of inputs.

Validation Testing;

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation testing began. Here we test the system in a manner that can be reasonably accepted by the customer, the system was tested against system requirement specification.

Output Testing;

After performing validation test the next phase is output test of the system, since no system could be useful if it does not produce the desired output in the desired format. By considering the format of the report/output, Output/report is generated or displayed and is tested. Here output format is considered in two ways: one is the screen and other is an printed form.

User Acceptance Testing:

Acceptance test is performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here is focused on external behavior of the system; the internal logic is not emphasized.
Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of the software developed. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirement

Program Testing:

Testing plan:

Test Case	Test Objectives
1	Test for username and password entry
2	Test for changing admin password
3	Test for adding an employee
4	Test for deleting employee
5	Testing for branch booking and customer booking
6	Test for branch and customer cancellation
7	Test for adding product information
8	Test for modifying product information
9	Test for adding sales billinformation
10	Test for adding purchase order information
11	Test for PurchaseBill information
12	Test for suppliers information
13	Test for entering the name
14	Test for entering mobile number
15	Test for entering email id

Test case: 1

Objectives: Test for username and password entry

Test Data: Valid: One of the valid login name and password to enter the program

Invalid: Invalid login name and password to enter the program

Output: Valid: Enter into the System normally

Invalid: Show the error message

Result: Valid: The user was allowed to enter the program

Invalid: The user is prompt with an error message and restricted to enter the program

Conclusion: Both the valid and invalid results are tested. Output tally with the required result hence the test is successful.

Test case:2

Objectives: Test for changing password

Test Data: Valid: valid admin password

Invalid: Password is blank and invalid password

Output: Valid: Allows changing the password

Invalid: The user is prompt with an error message

Result: Valid: Password will be changed

Invalid: The record is not updated to password

Conclusion: Both the valid and invalid results are tested. Output tally with the required result hence the test is successful
User password is change by admin .

Test case:4

Objectives: Test for deleting Employee information

Test Data: Valid: All required fields are selected

Invalid: Some required fields are not selected or incorrect

Output: Valid: Allows record to be deleted from the database

Invalid: The user is prompt with an error message

Result: Valid: Record status will become inactive

Invalid :The record will remain as active

Conclusion: Both the valid and invalid results are tested. And then the Record will be inactivated from the database

Test case:5

Objectives: Test for branch booking and customer booking

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid:The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be saved into the database

Test case:6

Objectives: Test for branch cancellation and customer cancellation

Test Data: Valid: Product should not manufactured and not billed.

Invalid:product already manufactured or billed

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record cancelation will be saved

Invalid:The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be saved into the database

Test case:7

Objectives: Test for adding product information

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid:The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be saved into the database

Test case:8

Objectives: Test for adding sales bill information

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be updated into the database

Test case: 9

Objectives: Test for adding purchase order information

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be updated into the database

Test case: 10

Objectives: Test for adding purchase bill information .

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be updated to the database

Test case: 11

Objectives: Test for adding supplier information

Test Data: Valid: All required fields are entered

Invalid: Some required fields are blank or incorrect

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. And then the Record will be updated to the database

Test case: 12

Objectives: Test for modifying supplier information

Test Data: Valid: Only numbers to be entered

Invalid: If character, space, and phone number less than 11 digits

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. Output tally with the require results hence the test is successful

Test case: 13

Objectives: Test for entering the name

Test Data: Valid: Only letters, numbers, underscore, dot and

Invalid: Characters other than letters, underscore, dot and '@'

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. Output tally with the require results hence the test is successful

Test case: 14

Objectives: Test for entering mobile number

Test Data: Valid: All required fields are entered and the user logged in as Administrator

Invalid: Test for mandatory fields, if all the fields are not filled, User logged in is other than administrator

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. Output tally with the require results hence the test is success*9sful

Test case: 15

Objectives: Test for entering email id

Test Data: Valid: All required fields are entered and the user logged in as Administrator

Invalid: Test for mandatory fields, if all the fields are not filled, User logged in is other than administrator

Output: Valid: Allows record to be added to the database

Invalid: The user is prompt with an error message

Result: Valid: Record will be saved

Invalid: The record will not saved

Conclusion: Both the valid and invalid results are tested. Output tally with the require results hence the test is successful

Conclusion,Limitations and Future Enhancement

Conclusion and Future Enhancement:

Conclusion:

- This program is written to provide easy to access each form by clicking on the icon in the main screen. It provides the flexibity to work on the system very easily. The user doesn't need to remember the entire menu system. Each column of each form provides clear description so that we can say it as simple program and user-friendly program.
- This program will verify and authenticate user id and password before allowing user to access. According to the user id and user name access rights are given so that everyone doesn't allow the form to access.
- The more important of this program is it allows to search. This allows the user to search for a particular item.

Limitations:

- No online service
- Cannot be connected to be any network on LAN's.
- The software can be installed only in system which has backend of SQL Server & front-end as Visual Basic

Future Enhancement:

- ❖ Fast generation of reports
- ❖ Eliminating redundancy and accuracy in result
- ❖ Reduce the cost and accuracy in result
- ❖ Reduce the cost and time
- ❖ Helps to improve efficiency
- ❖ Sharing limited resources

DEFINITION, ACRONYMS AND ABBREVIATIONS

Connection String:

We use connection string for connecting to the sql server. Connection string defines three things data source, initial catalog and integrated security.

- Data Source: This defines the source from which data needs to be extracted.
- Initial Catalog: It is the name of the database requires.
- Integrated Security: This asks if to use the application the login name given in the beginning is enough or before starting the application if new login and password is required.
  rdf

  - DFD: Data Flow Diagram
  - CFD: Context Flow Diagram
  - SQL: Structured Query Language

BIBLIOGRAPHY

1. An Integrated approach to Software Engineering: Pankaj Jalote.
2. www.google.com