****

**YELAHANKA**

**Bangalore – 560064**

**BANGALORE UNIVERSITY**

**Project Title**

**SPORTS INVENTORY**

Partially fulfillment of the Requirements For The Award Of Degree In

VI Semester Bachelor of Science 2019-2020

****

**YELAHANKA**

**Bangalore – 560064**

**BANGALORE UNIVERSITY**

**Certificate**

This project report entitled "**SPORTS INVENTORY**” is a bonafied record of the work done and submitted in a partial fulfillment of the requirement for the award of the degree bachelor of science,

VI semester [PMCs] by the University of Bangalore.

**HOD: Submitted By:**

**Department of Computer Science Name1 – 17P5S850\_\_**

**Name2 – 17P5S850\_\_**

**Name3 – 17P5S850\_\_**

**In charge:**

**Prof. ---------- Examiner 1:**

**Examiner 2:**

**ACKNOWLEDGEMENT**

A project is always a result of collaborative efforts. This one has been a special one so. We are included it to many people who helped this project happen. It is a right time to speak out the word of thanks for them. I sincerely admit that without their efforts.

We would like to express our simple words of respect and gratitude imprinted deep in our heart to **Prof. Sangayya Swamy** of Computer Science Department and **Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** who has been source of inspiration for us a guide through our project and for all the support and encouragement given to us to finish the project within a given time finally we thank each other for completion of our project with full co-operation

We would like to thank our esteemed principal **Dr. (princi name)** for giving us the best facilities. The satisfaction and publication that accompany successful completion of this project would be incomplete without our project guides **Prof. Sangayya Swamy** and **Prof. \_\_\_\_\_\_** who made it possible with constant guidance and encouragement and reward our effort with success we thank to Computer Science faculty.

**Project Group:**

**Name1**

**Name2**

**Name3**

**Contents:**

1. **Introduction**
2. **Synopsis**
3. **System Requirements**
4. **About Tools**
5. **Microsoft Visual Basic 6.0**
6. **Microsoft Access**
7. **System Design**
8. **System Requirements Specifications**
9. **ER Diagrams**
10. **Tables**
11. **Snapshots**
12. **Source Code**
13. **Testing**
14. **Conclusion**
15. **Future Enhancements**
16. **Bibliography**

**SYNOPSIS**

**PROJECT TITLE: SPORTS INVENTORY**

**Description of the project:**

* The Project Sports Inventory System includes stock sales details in the system, The software has the facility to stores the details of every product in the inventory shop automatically.
* The Sports Inventory System can be entered by using a valid username and password. It is accessible by an administrator or Other Staff Users. Only they can add data into the database.
* The interface is user-friendly.
* It deals with the collection of Stock information, and Sales.
* Earlier it used to done manually. The main function of the system is to add stock details and retrieve these details as and when required.

**Project perspectives:-**

* This project gives the procedural approach how a stock will come to shop and sell of that

**Project Functions:-**

Sports Inventory System application will perform the following major function:-

* Stock details
* Sales details

This software will help to keep the information in efficient and systematic way.

**Goals:**

* It is the user friendly application for Sports inventory shops  to reduces the routine work and manual works and helps to manages Shop..

**Purpose:**

* The Software is for the automation of Sports Inventory System .
* It maintains two levels of users:-
* Administrator Level
* Other User Lever

**Scope:-**

* It can be used in any Sports Shops for maintaining Shop daily transaction details.
* Every sales details automatically recorded on the database
* All this work is Entered by the Admin and other other staff.

**Proposed System:-**

In Proposed System, Admin Can Add and Modify the Stock Records as per requirements. The Shop details can be easily maintained in this System.

**Existing System:-**

Sport Inventory Systemcurrently use a manual system for the maintenance of Shop daily transaction information. The current system requires numerous paper forms, with data stores spread throughout the system.

**Advantages:-**

* It is fast, efficient and reliable
* Avoids data redundancy and inconsistency
* Application-based
* Very user-friendly
* Easy accessibility of data

**Disadvantages:-**

* The total software is manual

**SYSTEM REQUIREMENTS**

**Software Used:**

* Front end: Visual Basics 6.0
* Back end: MS Access
* Operating System: windows 10 professional

**Hardware used:**

* Harddisk : 1GB
* Ram : 1GB
* Processor : Intel Pentium IV 1.10GHz

**System Attributes**

**Robustness**

Product is robust. Because security has become more robust in vb 6.0 In addition to the role based security in vb 6, it comes with a new security model, codes access security. This security controls on what the code can access.

**Portability**

The Hospital Management System shall run in any Microsoft Windows

**Reusability**

Our system is reusability system since a segment of source code that can be used again to add new functionalities with slight modification.

**Testability**

Our system is testability system since it supports different types of testing methods

**Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an Information System. A data flow diagram can also be used for the visualization of Data Processing. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

A DFD represents flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that transforms the input into desired output. A DFD shows movement of data through the different transformations or processes in the system.

Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram. The appropriate register saved in database and maintained by appropriate authorities.

**Data Flow Diagram Notation:**

**Function**

**File/Database**

**Input/output**

**Flow**

**ABOUT TOOLS**

**About Visual Basic 6.0:**

In order to understand the application development process, it is helpful to understand some of the key concepts language: some familiarity with the windows environment is necessary.

A simplified version of the working of windows involves three key Concept windows, events and message.

Think of a window simply as a rectangular region with its own Boundaries. You are probably already aware of several different types of windows, an explorer window in windows; a document pops up to remind you of an appointment. While these are some of the windows a command button is a window: icons, text boxes and menu bars are all windows.

The Microsoft windows operating system manages all these windows. By assigning each one of unique id number. The system continually monitors each of these windows for signs of activity of events. Events can occur through user action such as a mouse click or a key press. Through programmatic control, or even as result of another windows actions. Each time an event occurs, it causes a message broadcast it to the other windows. Each window can then take the appropriate message. As you might imagine, dealing with all the of the possible combination of windows, events and messages could be mind-boggling. Fortunately, VB insulates you from having to deal with all of the low=level message handling. Many of the messages are handled automatically by VB. Others are exposed as event format. Data is formatted coming out of the source, and unformatted going back in you can also do custom formatting and perform additional checks using the format and also un-format events.

Data validation is also enhanced using the cause’s validation property with the validate event. By setting the cause’s validation property to true validate event for the previous control in the tab order will occur. Thus, by programming the validate event, you can prevent a control from losing focus until the information it contains has been validated.

**Data Access using ActiveX data objects (ADO):**

ActiveX Data Objects (ADO) is designed to be an easy-to-use application-level interface to any OLE DB data provider, including relational and non-relational databases, e-mail and file systems, text and graphics, and custom business objects, as well as existing ODBC data sources. Virtually all of the data available throughout the enterprise is available using the ADO data access technology.

ADO is easy to use, language-independent, implemented with a small footprint, uses minimal network traffic, and has few layers between the client application and the data source — all to provide lightweight, high-performance data access.

The general characteristics of ADO are:

* Ease of use.
* High performance.
* Programmatic control of cursors.
* Complex cursor types, including batch and server- and client-side cursors.
* Ability to return multiple results sets from a single query.

**ADO Object Model:** The ADO object model defines a collection of programmable objects that support the Component Object Model (COM) and OLE Automation to leverage the powerful partner technology called OLE DB. The ADO object model — when compared to other data access objects such as RDO or DAO — is flatter (has fewer objects) and simpler to use.

**Description of objects:**

**Command:** Maintains information about a command, such as a query string, parameter definitions, and so on. You can execute a command string on a Connection object of a query string as part of opening a Record set object, without defining a Command object. The Command object is useful where you want to define query parameters or execute a stored procedure that returns output parameters. The Command object supports a number of properties to describe the type and purpose of the query and help ADO optimize the operation.

**Connection:** Maintains connection information such as cursor type, connect string, query time-out, connection time-out, and default database.

**Error:** Contains extended error information about error conditions raised by the data provider. Because a single statement can generate two or more errors, the Errors collection can contain more than one Error object at a time.

**Field:** Contains information about a single column of data within a record set. The Record set object uses the Fields collection to contain all of its Field objects. This Field information includes data type, precision, and numeric scale.

**Parameter:** A single parameter associated with a Command. The Command object uses the Parameters collection to contain all of its Parameter objects. ADO Parameter objects can be created automatically by sending queries to the database. However, you can also build this collection programmatically to improve performance at run time.

**Property:** A provider-defined characteristic of an ADO object. ADO objects have two types of properties: built-in and dynamic. Built-in properties are those properties implemented in ADO and available to any new ADO object. Dynamic properties are defined by the underlying data provider and appear in the Properties collection for the appropriate ADO object. For example, a property may indicate if a Record set object supports transactions or updating. This is one of the greatest features of ADO, in that it lets the ADO service provider present special interfaces.

**Record set:** A set of rows returned from a query, including a cursor into those rows. You can open a Record set object (that is, execute a query) without explicitly opening a Connection object. However, if you do first create a Connection object, you can open multiple Record set objects on the same connection.

**Implementing the ADO Data Control:**

The easiest way to implement the ADO in your application is through the ADO data control. This active-X control is a simple data browser and editor in your applications.

Although the ADO data control is extremely simple to implement, you do give up some flexibility. Working with ADO model with code gives you much more control your data. However, for prototyping and interface the ADO data control can provide a useful interface for accomplishing the task quickly and easily.

**Microsoft Access Database**

**Introduction:** Microsoft Access is a Database Management System (DBMS) package. MS Access provide window-based interface and for creating new database, one has to select as usual “New Database”.

From the file menu, the database window lists all the objects:

• Tables store your data in your database

• Queries ask questions about information stored in your tables

• Forms allow you to view data stored in your tables

• Reports allow you to print data based on queries/tables that you have created

**Microsoft Access Features:**

* Ideal for individual users and smaller teams.
* Easier than client-server database to understand and use.
* Import and export to other Microsoft Office and other applications.
* Ready templates for regular users to create and publish data.
* Allows building and publishing Web databases effortlessly.

**Easy to Use:** Microsoft Access is a menu-driven database program that includes a graphical user interface enabling novice users to enter, manage and report on their data. Data can be entered using a familiar spreadsheet or table interface and the program will automatically generate forms based upon the data and labels that have been entered. Routine tasks such as searches, sorts and queries can be automated with macros using a point-and-click paradigm. Reports can be formatted by dragging and dropping fields on a graphical interface and the program includes many templates that can be easily modified by end users.

Customization: Microsoft Access users can easily create and modify database queries and reports using the graphical user interface, and a menu-driven macro recorder to automate repetitive steps. Advanced users can customize Access applications using the Visual Basic for Applications "VBA" programming language common to all Microsoft Office applications. Access also includes many templates that can be easily modified by end users.

**SYSTEM DESIGN**

**INPUT DESIGN**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

**OBJECTIVES FOR INPUT DESIGN**

### The objectives of input design are:

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.

To use validation checks and develop effective input controls.

**Data Input Methods**

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by −

* Clear form design by leaving enough space for writing legibly.
* Clear instructions to fill form.
* Clear form design.
* Reducing key strokes.
* Immediate error feedback.

Some of the popular data input methods are −

* Online data input method
* Computer readable forms

Interactive data input

**Input Integrity Controls**

Input integrity controls include a number of methods to eliminate common input errors by end-users. They also include checks on the value of individual fields; both for format and the completeness of all inputs.

Audit trails for data entry and other system operations are created using transaction logs which gives a record of all changes introduced in the database to provide security and means of recovery in case of any failure.

**Output Design**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

**Objectives for Output Design**

### The objectives of input design are −

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end users requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**Various Types of Outputs**

**External Outputs**

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

**Internal Outputs**

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

There are three types of reports produced by management information −

* **Detailed Reports** − they contain present information which has almost no filtering or restriction generated to assist management planning and control.
* **Summary Reports** − they contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details.
* **Exception Reports** − they contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.

**Output Integrity Controls**

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

**Entity Relationship Diagrams**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes

An entity-relationship model (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

**Symbols used in this E-R Diagram:**

**Entity:** Entity is a “thing” in the real world with an independent existence. An entity may be an object with a physical existence such as person, car or employee. Entity symbol is as follows

**Attribute:** Attribute is a particular property that describes the entity. Attribute symbol is

**Relationship**: Relationship will be several implicit relationships among various entity types whenever an attribute of one entity refers to another entity type some relationship exits**.**

**Key attributes:** An entity type usually has an attribute whose values are distinct for each individual entity in the collection. Such an attribute is called key attribute. Key attribute symbol is as follows

**Composite attributes:**

**Multivalve attributes:**

**System Requirements Specifications**

**SOFTWARE:**

Frontend: **Visual Basic 6.0**

Backend: **Microsoft Access 2007 (DBMS)**

Operating System: **Windows 7 (or Above)**

**HARDWARE:**

Harddisk: **1 GB**

Ram: **1 GB**

Processor: **Intel Pentium IV 1.1GHz** Or Above

.

**ER DIAGRAMS**

**Login**

**Login**

**Validate**

**HOME**

**New User Registration**

**User Registration**

**ER-Name (create ur self for all table as per ur curriculum )**

**TABLES**

**Login**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| Username | Text | Primary key |
| Password | Text | Not Null |

**Goods**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| Item\_code | Text | Foreign key referenced from Student Table Regno |
| Item\_name | Text | Foreign key referenced from Book Table ISBN |
| Stock\_qty | Text | Not Null |
| Stock\_date | Date/Time | Not Null |
| Net\_price | Number | Not Null |
| Supplier\_Code | Text | Not Null |

**Invoice**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| Invoice\_id | Text | Primary Key |
| Buyer\_name | Text | Not Null |
| Item\_code | Text | Foreign key referenced from Goods Table Item\_code |
| Item\_name | Date/Time | Not Null |
| Net\_price | Number | Not Null |
| CGST | Number | Not Null |
| SGST | Number | Not Null |
| MRP | Number | Not Null |
| Invoice\_date | Date/Time | Not Null |

**SNAPSHOTS**

**SOURCE CODE**

Private Sub Form\_Load()

Timer1.Enabled = True

Me.Left = (Screen.Width - Me.Width) / 2

Me.Top = (Screen.Height - Me.Height) / 2

End Sub

Private Sub Timer1\_Timer()

ProgressBar1.Value = ProgressBar1.Value + 5

If (ProgressBar1.Value = ProgressBar1.Max) Then

Timer1.Enabled = False

Form9.Show

Unload Me

End If

End Sub

Private Sub Combo1\_Click()

Adodc1.RecordSource = "select \* from Stock where Supplier\_code ='" & Combo1 & "'"

Adodc1.Refresh

End Sub

Private Sub Form\_Load()

Adodc1.Refresh

With Adodc1.Recordset

Do Until .EOF

Combo1.AddItem ![Supplier\_code]

.MoveNext

Loop

End With

End Sub

Private Sub homew\_Click()

Form3.Show

Unload Me

End Sub

Private Sub mnurefresh\_Click()

Adodc1.RecordSource = "select \* from Stock"

Adodc1.Refresh

End Sub

Private Sub exitmenu\_Click()

End

End Sub

Private Sub mnulog\_Click()

Form9.Show

Unload Me

End Sub

Private Sub mnurep1\_Click()

Form5.Show

Unload Me

End Sub

Private Sub mnusales\_Click()

Form8.Show

Unload Me

End Sub

Private Sub mnustknew\_Click()

Form6.Show

Unload Me

End Sub

Private Sub mnustkup\_Click()

Form7.Show

Unload Me

End Sub

Private Sub report1\_Click()

Form2.Show

Unload Me

End Sub

Private Sub usermenu\_Click()

Form4.Show

End Sub

Private Sub Form\_Load()

Me.Left = (Screen.Width - Me.Width) / 2

Me.Top = (Screen.Height - Me.Height) / 2

End Sub

Private Sub menuhome\_Click()

Unload Me

End Sub

Private Sub newuserbtn\_Click()

Adodc1.Recordset.AddNew

Adodc1.Recordset.Fields("Username").Value = Text1.Text

If Text2.Text = Text3.Text Then

Adodc1.Recordset.Fields("Password").Value = Text3.Text

Adodc1.Recordset.Update

MsgBox "New User Added Successfully"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Else

MsgBox "password not matched"

Text2.Text = ""

Text3.Text = ""

End If

End Sub

Private Sub homew\_Click()

Form3.Show

Unload Me

End Sub

Private Sub mnurefresh\_Click()

Adodc1.RecordSource = "select \* from Sale"

Adodc1.Refresh

End Sub

Dim prodcode As String

Private Sub cmdsave\_Click()

If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text4.Text = "" Then

MsgBox "Enter Correct Details"

Else

Adodc1.Recordset.Fields("Item\_code").Value = Text1.Text

Adodc1.Recordset.Fields("Item\_name").Value = Text2.Text

Adodc1.Recordset.Fields("Stock\_qty").Value = Text3.Text

Adodc1.Recordset.Fields("Net\_price").Value = Text4.Text

Adodc1.Recordset.Fields("Stock\_date").Value = Format(Now, "dd/mm/yyyy")

Adodc1.Recordset.Fields("Supplier\_code").Value = Text5.Text

Adodc1.Recordset.Update

MsgBox "New Item Added"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

cmdsave.Visible = False

Adodc1.RecordSource = "select \* from Goods"

Adodc1.Refresh

End If

End Sub

Private Sub Command1\_Click()

Call AutoReg

Adodc1.Recordset.AddNew

Text1 = Format(prodcode, "G000")

cmdsave.Visible = True

End Sub

Private Sub Form\_Load()

cmdsave.Visible = False

End Sub

Private Sub mnuhm\_Click()

Form3.Show

Unload Me

End Sub

Public Sub AutoReg()

On Error GoTo Err\_id

Adodc1.Refresh

If Adodc1.Recordset.RecordCount = 0 Then

prodcode = 1

Else

Adodc1.Recordset.MoveLast

prodcode = Mid(Adodc1.Recordset("Item\_code"), 2, 3) + 1

Adodc1.Refresh

End If

Exit Sub

Err\_id:

prodcode = 1

MsgBox "Error Found", vbCritical, "Sports Inventory"

End Sub

Private Sub cmdcheck\_Click()

Adodc2.RecordSource = "select \* from Goods where Item\_code = '" + Text1.Text + "'"

Adodc2.Refresh

If Adodc2.Recordset.RecordCount = 0 Then

MsgBox "Invalid Product code"

ElseIf Adodc2.Recordset.RecordCount = 1 Then

Text2 = Adodc2.Recordset("Item\_name")

Text3 = Adodc2.Recordset("Stock\_qty")

Text4 = Adodc2.Recordset("Net\_price")

Text5 = Adodc2.Recordset("Stock\_date")

cmdsave.Visible = True

End If

End Sub

Private Sub cmdsave\_Click()

If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Or Text4.Text = "" Then

MsgBox "Enter Correct Details"

Else

Adodc2.RecordSource = "select \* from Goods where Item\_code = '" + Text1.Text + "'"

Adodc2.Refresh

Adodc2.Recordset.Fields("Item\_code").Value = Text1.Text

Adodc2.Recordset.Fields("Item\_name").Value = Text2.Text

Adodc2.Recordset.Fields("Stock\_qty").Value = Text3.Text

Adodc2.Recordset.Fields("Net\_price").Value = Text4.Text

Adodc2.Recordset.Fields("Stock\_date").Value = Format(Now, "dd/mm/yyyy")

Adodc2.Recordset.Fields("Supplier\_code").Value = Text5.Text

Adodc2.Recordset.Update

MsgBox "Goods Updated"

Adodc2.RecordSource = "select \* from Goods"

Adodc2.Refresh

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text4.Text = ""

cmdsave.Visible = False

End If

End Sub

Private Sub Form\_Load()

cmdsave.Visible = False

End Sub

Private Sub mnuhm\_Click()

Form3.Show

Unload Me

End Sub

Dim invcode As String

Private Sub Command1\_Click()

Call AutoReg

Adodc1.Recordset.AddNew

Text1 = Format(invcode, "I000")

Command3.Visible = True

End Sub

Private Sub Command2\_Click()

If Text1.Text = "" Or Text2.Text = "" Or Text3.Text = "" Or Text4.Text = "" Then

MsgBox "Enter Correct Details"

Else

Adodc1.Recordset.Fields("Invoice\_id").Value = Text1.Text

Adodc1.Recordset.Fields("Buyer").Value = Text2.Text

Adodc1.Recordset.Fields("Item\_code").Value = Text3.Text

Adodc1.Recordset.Fields("Item\_name").Value = Text4.Text

Adodc1.Recordset.Fields("Net\_price").Value = Label11.Caption

Adodc1.Recordset.Fields("CGST").Value = Label8.Caption

Adodc1.Recordset.Fields("SGST").Value = Label9.Caption

Adodc1.Recordset.Fields("MRP").Value = Label10.Caption

Adodc1.Recordset.Fields("Invoice\_date").Value = Format(Now, "dd/mm/yyyy")

Adodc1.Recordset.Update

MsgBox "Done"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Label11.Caption = ""

Label8.Caption = ""

Label9.Caption = ""

Label10.Caption = ""

Command2.Visible = False

End If

End Sub

Private Sub Command3\_Click()

Adodc2.RecordSource = "select \* from Goods where Item\_code = '" + Text3.Text + "'"

Adodc2.Refresh

If Adodc2.Recordset.RecordCount = 0 Then

MsgBox "Invalid Product code"

ElseIf Adodc2.Recordset.RecordCount = 1 Then

Text4 = Adodc2.Recordset("Item\_name")

Label11 = Adodc2.Recordset("Net\_price")

Label8 = Val(Label11) \* 0.08

Label9 = Val(Label11) \* 0.08

Label10 = Val(Label11) + Val(Label8) + Val(Label9)

Command2.Visible = True

End If

End Sub

Private Sub Form\_Load()

Command2.Visible = False

Command3.Visible = False

End Sub

Private Sub mnuhmm\_Click()

Form3.Show

Unload Me

End Sub

Public Sub AutoReg()

On Error GoTo Err\_id

Adodc1.Refresh

If Adodc1.Recordset.RecordCount = 0 Then

invcode = 1

Else

Adodc1.Recordset.MoveLast

invcode = Mid(Adodc1.Recordset("Invoice\_id"), 2, 3) + 1

Adodc1.Refresh

End If

Exit Sub

Err\_id:

invcode = 1

MsgBox "Error Found", vbCritical, "Sports Inventory"

End Sub

Private Sub cmdCancel\_Click()

End

End Sub

Private Sub cmdOK\_Click()

If Text1 = "" Or Text2 = "" Then

MsgBox "Please Enter Valid Details"

Text1.SetFocus

Exit Sub

End If

Adodc1.RecordSource = "select \* from Admin where Username = '" + Text1.Text + "' and Password = '" + Text2.Text + "'"

Adodc1.Refresh

If Adodc1.Recordset.RecordCount = 0 Then

MsgBox " Login Failed"

Text1.Text = ""

Text2.Text = ""

Else

MsgBox "Login Successfull"

Form3.Show

Form2.Hide

Unload Me

Exit Sub

End If

End Sub

Private Sub Form\_Load()

Me.Left = (Screen.Width - Me.Width) / 2

Me.Top = (Screen.Height - Me.Height) / 2

End Sub

**TESTING**

**Introduction of Testing:**

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

**Objectives of Testing:**

Testing is a process of executing a program with the intent of finding errors. A Successful test case is one that uncovers an as- yet-undiscovered error. System testing is a stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences. As stated before, testing is vital to the success of a system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. A series of tests are performed before the system is ready for the user acceptance test.

**Testing Methods:**

System testing is the stage of implementation. This is to check whether the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: on line response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

**Testing Steps are:**

1. **Unit Testing:**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

1. **Integration Testing**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

1. **Validation**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

a) The function or performance characteristics confirm to specification and are accepted.

b) A deviation from specification is uncovered and a deficiency lists is created.

c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

1. **Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

1. **User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

a) Input Screen Design

b) Output Screen Design

c) Format of reports and other outputs.

Security mechanisms:

This system is provided with authentication, without this user can pass. So only the legitimate users are allowed to use the application. If the legitimate users share the authentication information then the system is open to outsiders.

**Limitations:**

* Since it is an online project, customers need internet connection to buy products.
* People who are not familiar with computers can’t use this software.
* Customer must have debit card or credit card to purchase products.

Future scope and further enhancement:

This web application involves almost all the features of the online shopping. The future implementation will be online help for the customers and chatting with website administrator

**CONCLUSION**

This software was developed specifically to answer to the needs of [specify]. We need to improve on and some new features to be added. Even then, this does not affect the working of this software. Some of the obvious Changes/Additions.

**BIBLIOGRAPHY**

* [www.youtube.com](http://www.youtube.com)
* [www.vbtutor.net](http://www.vbtutor.net)