

**ACCOUNTS MANAGAEMENT SYSTEM FOR CHOTTA BAGHIYA PAZHAMUTHIR
CHOLAI**

**PROJECT WORK SUBMITTED TO
THE DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

ARUL ANANDAR COLLEGE (AUTONOMOUS)

IN PARTIAL FULFILLMENT OF THE DEGREE OF

BACHELOR OF COMPUTER SCIENCE

By

ASHIK AMIR. J

Reg.No : 20CSC108

Under the Guidance of

Mr.S.CHARLES ILAYARAJA,M.Sc.,M.Phil.,NET

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

ARUL ANANDAR COLLEGE (AUTONOMOUS)

(Affiliated to Madurai Kamaraj University)

Nationally Re-Accredited by NACC at “A” Grade

KARUMATHUR-625 514, MADURAI

APRIL-2023

CERTIFICATE

This is to certify that the project Titled “**ACCOUNTS MANAGAEMENT SYSTEM FOR CHOTTA BAGHIYA PAZHAMUTHIR CHOLAI** ” is a bonafide work done by **J.ASHIK AMIR**, Reg. No. **20CSC108** in partial fulfillment of the requirements for the award of the degree of Bachelor of Computer science and Applications of Madurai Kamaraj University.

Ms.I.JULIET SHANTHI MCA.,M.Phil.,

Head of the Department

Department of Computer Science
and Applications

Arul Anandar College (Autonomous)
Karumathur.

Mr.S.CHARLES ILAYARAJA,M.Sc.,M.Phil.,NET

Project Guide

Assistant Professor, Department of
Computer Science & Applications

Arul Anandar College (Autonomous)
Karumathur.

The viva voice examination of this Mini Project work was held on

Internal Examiner

External Examiner

ACKNOWLEDGMENT

First of all I record my gratefulness to the god almighty for his blessing to make this project a great success. I express my sincere thanks to my parents for their blessings and encouragement for the success of the project.

A word of thanks to our respected principal, **Rev. Dr.Godwin Rufus S.J** for this encouragement and moral support.

I wish to express my profound gratitude to the deputy principal of self-financed courses **Dr.A.Sundararaj MSc., M.Phil., PhD.,PGDCA** for this encouraging word to take up any challenging task.

I would like to express my heartfelt thanks to **Ms.I.Juliet Shanthi MCA.,M.Phil.,**Head, Department of Computer Science and Applications for given me permission to carry out the project work and also for her valuable suggestion, guidance and support.

I extent my sincere thanks **Mr. S.CHARLES ILAYARAJA, M.Sc.,M.Phil.,NET** Assistant Professor,,Department of Computer Science and Applications for his motivation, guidance and moral support.

I extent my sincere thanks to **Mr.M.Vinayaga Moorthy CEO at Techvolt Software Pvt.Ltd** for giving me the opportunity to do an internship on Web Development within the company

Finally, I express my thanks to all my friends for their encouragement and help for completing this project and those who have directly and indirectly contributed towards the success of this project.

Table of Content

S.NO	CONTENT	PAGE NO
1	Introduction 1.1 About the project 1.2 About company profile	6
2	Problem Definition And Description 2.1 Problem Definition 2.2 Problem Description	8
3	System Analysis 3.1 Existing System 3.2 Problem in existing system 3.3 Proposed System 3.4 Advantages of proposed system 3.5 Feasibility study 3.6 Data Flow Diagram	10
4	System Design 4.1 Architectural Design 4.2 Module design 4.3 Input Design 4.4 Output Design 4.5 Normalization 4.6 Software Description	15
5	Implementation	26
6	Testing 6.1 Unit Testing 6.2 Integrated Test	28
7	Installation 7.1 Hardware Installation 7.2 Software Installation	31
8	Maintenance	33
9	Applications	36
10	Conclusion	38
11	Bibliography	40
12	Appendix 12.1. Screen Shots 12.2. Source Code	42

CHAPTER 01

INTRODUCTION

1. INTRODUCTION

1.1. About the project:

An accounting management system software helps businesses to keep track of all incomes, expenses, and every other financial transaction. It also helps business owners to get the data that they need for making business decisions

1.2 About the company:

Chotta baghiya pazhamudhir cholai is famous stop point in nagamalai pudukottai, madurai.

It contains all kinds of fruits and vegetables at reasonable price. Now, there is an option of getting fruits, vegetables and dryfruits at home delivery.

CHAPTER - 2

PROBLEM DEFINITION & DESCRIPTION

2. PROBLEM DEFINITION & DESCRIPTION

2.1 Problem definition

The problem at hand is an inefficient manual process that leads to inaccurate reporting and potential data security risks. The current process requires a significant amount of manual effort, which is prone to errors and inconsistencies. As a result, the reporting generated from this process is not entirely reliable, and it is difficult to draw accurate conclusions from the data. Furthermore, the manual process lacks appropriate security measures, leaving the data vulnerable to unauthorized access or theft. Therefore, there is a pressing need to find a more efficient and secure solution that can eliminate the risk of errors and ensure the confidentiality, integrity, and availability of the data.

2.2 Problem description

Inefficient manual process - The current manual process for managing accounts is time-consuming, error-prone, and does not provide real-time access to data.

Inaccurate reporting - The current manual process is prone to errors and inconsistencies, making it difficult to produce accurate financial reports.

Data security - The manual process does not provide adequate protection against unauthorized access to sensitive financial data.

CHAPTER – 3

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The existing system of writing notes for accounts management is a manual process where transactions, payments, and other financial data are recorded on paper or in a notebook. This process can be time-consuming and prone to errors.

3.2. PROPOSED SYSTEM:

A proposed system in software-based accounts management would automate the process, providing many benefits such as faster and more accurate data entry, improved organization and access to information, and the ability to generate reports and analyze data more easily. Additionally, a software-based system provides better security and backup options, making it a more reliable way to manage financial information.

3.3. FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization.

It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

3.1 ECONOMIC FEASIBILITY:

A feasibility study is an important step in determining whether a photo studio management system is viable and worth pursuing. The feasibility study involves evaluating the technical, economic, operational, and scheduling aspects of implementing such a system.

3.2 TECHNICAL FEASIBILITY:

The first step in determining the technical feasibility of a photo studio management system is to assess whether the technology needed to create and operate the system is available and whether it is feasible to implement. This includes evaluating the software and hardware requirements, including the necessary programming languages, operating systems, and databases, and ensuring that the system can be integrated with existing systems, if any.

3.3 OPERATIONAL FEASIBILITY:

The operational feasibility of a photo studio management system is determined by evaluating whether the system can be implemented and used effectively by the employees of the photo studio. This includes analyzing the training and support requirements for the system, evaluating how the system will integrate with existing processes, and determining whether the system can be easily used by employees with varying levels of technical expertise.

3.5. Data Flow Diagram

A data flow diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is also known as Data Flow Graphs or Bubble Chart. The DFD is used to represent increasing information flow and functional details. Also DFD can be states as the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail.

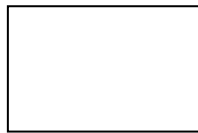
A level zero also called the fundamental system model or a context level DFD that represent the entire software elements as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Additional process and information flow parts are represented in the next level, i.e., level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model. Any processes that are complex in level 1 will be further represented into sub functions in the next level, i.e., level 2.

Data Flow Diagram is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes and data sources. The purpose of data flow diagram is to provide a semantic bridge between users and system developers. The diagram is the basis of structured system analysis. A DFD describes what data flows rather than how they are processed, so it does not depend on hardware, software, data structure or file organization.

Components of Data Flow Diagram

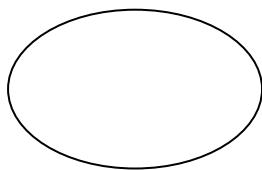
There are four symbols that are used in the drawing of data flow diagrams

- Entities



External entities represent the sources of data that enter the system or the recipients of data that leave the system

- Process



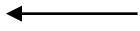
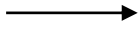
Processes represent activities in which data is manipulated by being stored or retrieved or transformed in some way. A circle represents it. The process will show the data transformation or change.

- Database



Databases represent storage of data within the system

- Data Flow



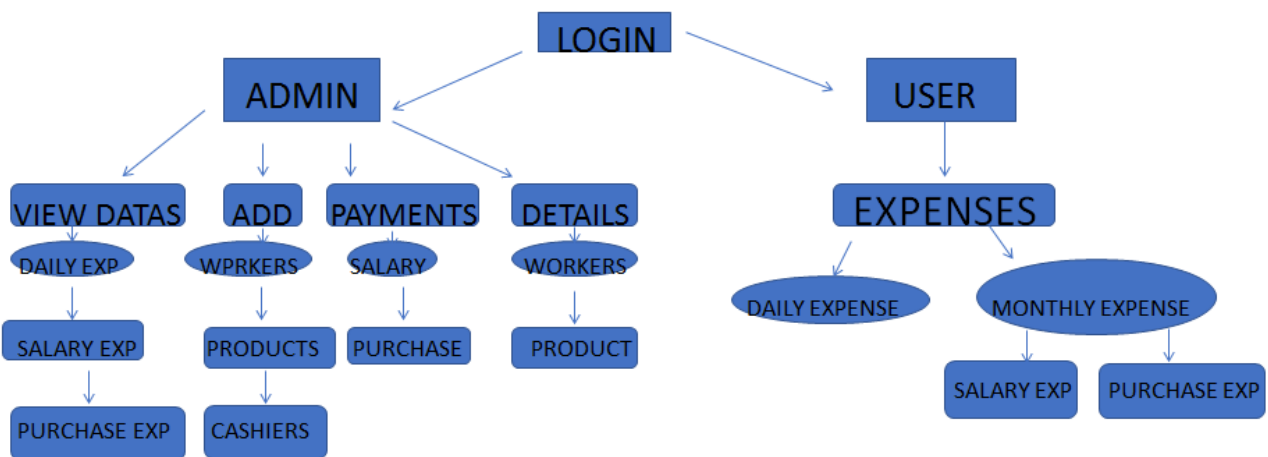
A data flow shows a flow of information from its source to its destination. A line represents a data flow, with arrow heads showing the direction of flow.

CHAPTER – 4
SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 Architectural Design

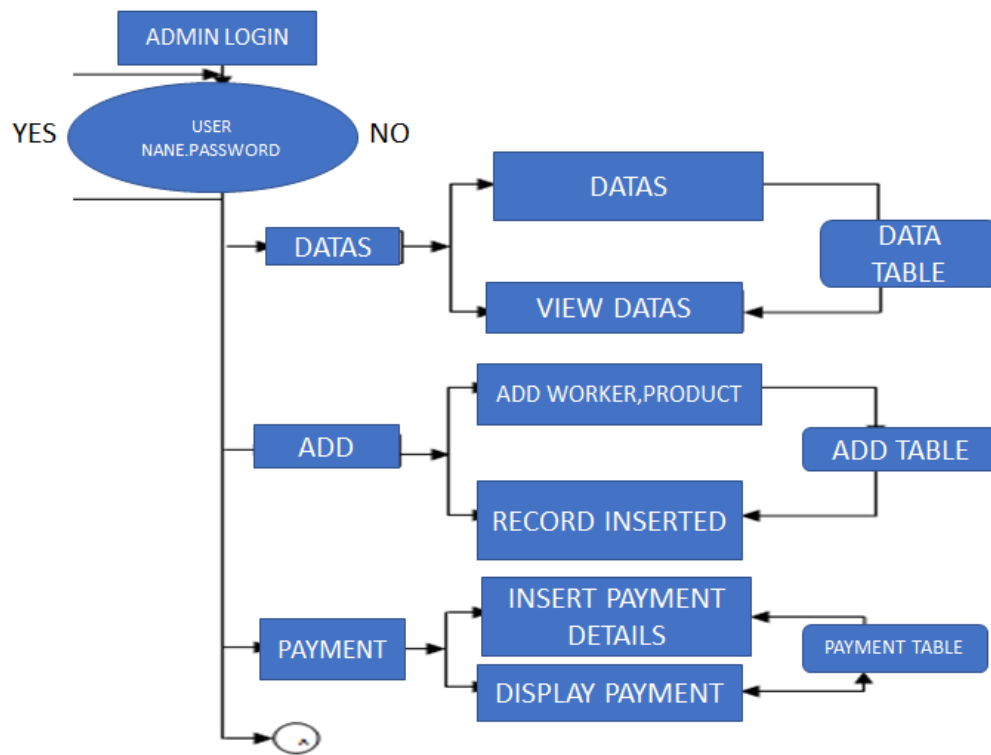
A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).



4.2 Module Description

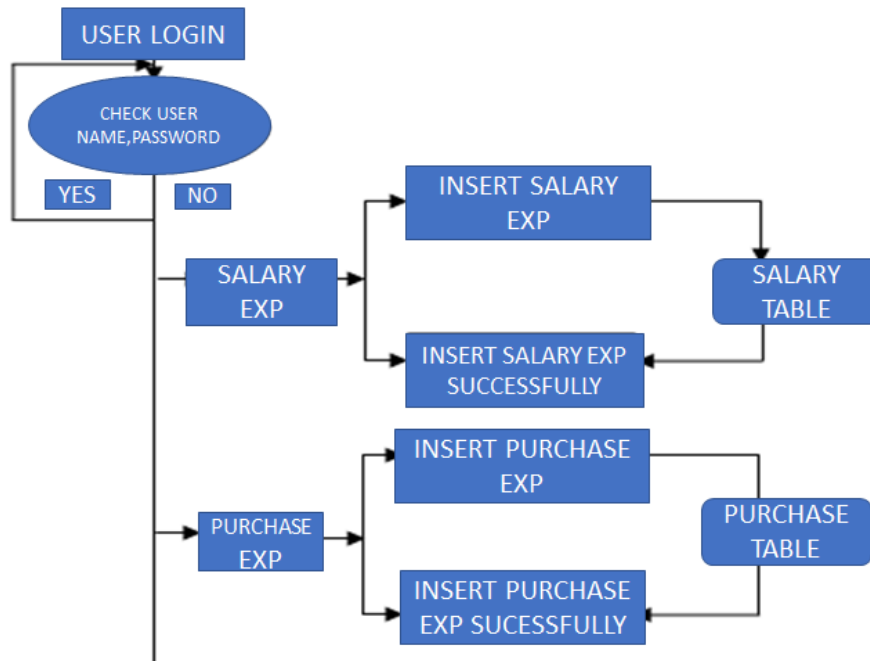
ADMIN MODULE:

The Admin module is used for view the data's such as expense, purchase and add worker details etc,



USER MODULE:

The user module used to upload the expense details of the wages and debt etc.,.



4.3 INPUT DESIGN:

Input design is part of overall system design that requires special attention. Designing input data is to make the data entered easy and free from **errors**. Validation is made for each and every data that is entered. Help information is provided for the users during when the customer feels difficult. Input design is the process of converting the user originated inputs to a computer based format. A system user interacting through a workstation must be able to tell the system whether to accept the input to produce reports. The collection of input data is considered to be most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information.

4.4 OUTPUT DESIGN

The output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from a

hard copy or screen display. Computers output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making. As the outputs are the most important source of information to the user, better design should improve the systems relation and also should help in decision-making. The output devices capability, print capability, print capability, response time requirements etc... should also be considered form design elaborates the way output is presented and layout available for capturing information. Its very helpful to produce the clear, accurate and speedy information for end users.

4.5 Normalization

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the samedata in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

The Normal Forms

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF. Fifth normal form is very rarely seen and won't be discussed in this article. Before we begin our discussion of the normal forms, it's important to point out that they are guidelines and guidelines only. Occasionally, it becomes necessary to stray from them to meet practical business requirements.

However, when variations take place, it's extremely important to evaluate any possibleramifications they could have on your system and account for possible inconsistencies. That said, let's explore the normal forms.

First Normal Form (1NF)

First normal form (1NF) sets the very basic rules for an organized database:

- Eliminate duplicative columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second Normal Form (2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data:

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Third Normal Form (3NF)

Third normal form (3NF) goes one large step further:

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

Fourth Normal Form (4NF)

Finally, fourth normal form (4NF) has one additional requirement:

- Meet all the requirements of the third normal form.
- A relation is in 4NF if it has no multi-valued dependencies.

Remember, these normalization guidelines are cumulative. For a database to be in 2NF, it must first fulfill all the criteria of a 1NF database

Table Structure

ADD WORKERS

FieldName	Datatype	Key/Null
NAME	varchar(50)	Not Null
ADDRESS	Text(50)	Not Null
PH NO	varchar(50)	Not Null
SALARY	Varchar(10)	Not Null

MANAGER TABLE

FieldName	Datatype	Key/Null
ID	varchar(50)	Not Null
USER NAME	varchar(50)	Not Null
PASSWORD	varchar(50)	Not Null

SALARY TABLE

FieldName	Datatype	Key/Null
Name	varchar(50)	Not Null
DATE	Varchar(20)	Not Null
IN TIME	varchar(50)	Not Null
OUT TIME	varchar(50)	Not Null
Working Hour	varchar(50)	Not Null
Salary	varchar(50)	Not Null

PURCHASE TABLE

FieldName	Datatype	Key/Null
Date	varchar(50)	Not Null
Product Name	Varchar(20)	Not Null
Purchase Amount	varchar(50)	Not Null
Supplier Name	varchar(50)	Not Null
Supplier Ph.Num	varchar(50)	Not Null

DATABASE DESIGN:

Add Workers:

PERSONAL\SQLEX... - dbo.ADDWORK		Object Explorer Details		
	NAME	ADDRESS	PH_NO	SALARY_1HR
▶	ASHIK	MADURAI	7639906657	48
	BANU	MADURAI	9843509354	50
	GOKUL	CHECKANAM	5678902345	40
*	NULL	NULL	NULL	NULL

Manager Table:

PERSONAL\SQLEX...bo.ManagerLogin		PERSONAL\SQLEX...	
	id	UserName	Pwd
▶	1	manager	manager
	2	ashik	786
	3	banu	984
	4	ARSHU	123
	5	SAM	123
*	NULL	NULL	NULL

Salary Table:

PERSONAL\SQLEXP...t - dbo.SALARY1						
PERSONAL\SQLEXP...ort - dbo.SALARY						
PERSONAL\SQLEXP...t - dbo.S						
	NAME	DATE	IN_TIME	OUT_TIME	WORKING_HO...	SALARY
▶	ASHIK	03/10/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-576
	BANU	03/10/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-600
	ASHIK	03/13/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-576
	BANU	03/13/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-600
	GOKUL	03/13/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-480
	ASHIK	03/14/2023	09:00:00:AM	09:00:00:PM	-12:00:00	-576

Orders:

PERSONAL\SQLEX... dbo.PURCHASE2					
PERSONAL\SQLE...RCHASE_DETAILS					
PERSONAL\SQLE					
	DATE	PRODUCT_NAME	PURCHASE_AM...	SUPPLIER_NAME	SUPPLIER_PH_NC
▶	03/10/2023	dairy milk	1000	MANI	8907654321
	03/10/2023	guna shacks	2000	HARI	4567890321
	03/10/2023	halwa	3000	GUNA	2345654321
	03/10/2023	HONEY BALLS	3500	RAJA	5675432345
	03/15/2023	dairy milk	3000	MANI	7890765432

4.6 SOFTWARE DESCRIPTION:

Introduction to C# .NET

C# .NET is a popular object-oriented programming language developed by Microsoft for building various types of applications. In this article, we will provide an overview of its characteristics, types, uses, and applications.

Characteristics of C# .NET

Here are some of the key characteristics of C# .NET:

- Object-oriented: C# .NET supports the principles of object-oriented programming, such as encapsulation, inheritance, and polymorphism.
- Type safety: C# .NET provides strong type checking at compile time, which helps in detecting type-related errors early in the development process.
- Garbage collection: C# .NET includes a garbage collector that automatically frees up unused memory and resources
- Cross-language interoperability: C# .NET can be used with other .NET languages, such as Visual Basic .NET and F#.

Types of C# .NET

C# .NET has two main types:

- Console applications: These are applications that run in a console window and can take input and display output on the console.
- Windows applications: These are graphical applications that run in a windowed environment and provide a more interactive user interface

Uses of C# .NET

Here are some of the common uses of C# .NET:

- Windows desktop applications: C# .NET can be used to build Windows desktop applications that provide various functionalities such as multimedia, gaming, and productivity tools.
- Web applications: C# .NET can be used to build web applications using ASP.NET, Razor, and other web frameworks.
- Mobile applications: C# .NET can be used to build cross-platform mobile applications using Xamarin.

- Game development: C# .NET can be used to build games using popular game engines such as Unity and Mono Game.

Applications of C# .NET

Here are some of the popular applications that are built using C# .NET:

- Microsoft Office: C# .NET is used in building various functionalities of Microsoft Office, such as Excel and Word.
- Windows operating system: C# .NET is used in building various components of Windows operating system, such as Windows Forms and WPF.
- Games: Popular games such as Rust, Stardew Valley, and Rimworld are built using C# .NET.
- Web applications: Popular web applications such as Stack Overflow and Groupon are built using C# .NET.

Conclusion

C# .NET is a versatile and popular programming language used for building various types of applications. Its object-oriented nature, strong type checking, and garbage collection make it a powerful tool for developers. With its wide range of uses and applications, learning C# .NET can open up many opportunities in the software development industry.

CHAPTER – 5

IMPLEMENTATION

5. IMPLEMENTATION

Introduction

An accounts management system for a fruits shop is a software application that helps the shop owners to manage their accounts efficiently. It enables them to keep track of their customers, record sales transactions, generate invoices and receipts, view customer accounts and transaction history, and calculate total revenue and profit.

Features:

Customer Management

The accounts management system allows the fruits shop to manage their customers by adding their details such as name, address, phone number, and email address. This information can be used to keep track of the customer's purchases and generate invoices.

Sales Transactions

The system allows the fruits shop to record their sales transactions by adding details such as the date of the transaction, the customer's ID, and the amount. The system also allows the shop owners to specify whether the transaction is a sale or a purchase.

Invoices and Receipts

The accounts management system generates invoices and receipts for the customers based on their transactions. The invoices and receipts include details such as the customer's name and address, the date of the transaction, the item purchased, the quantity, the price, and the total amount due.

Customer Accounts and Transaction History

The system allows the fruits shop to view their customer's accounts and transaction history. This information can be used to keep track of the customer's purchases and generate reports for analysis.

Revenue and Profit Calculation

The accounts management system allows the fruits shop to calculate their total revenue and profit. The revenue is calculated based on the total amount of sales, while the profit is calculated by subtracting the total cost of goods sold from the total revenue.

CHAPTER – 6

TESTING

6. TESTING

System testing is the stage of implementation, which aimed at ensuring that the system works accurately and efficiently before the live operation commences. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

6.1 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

6.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

- i) Top-down integration testing.
- ii) Bottom-up integration testing.

6.3 White box testing

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

6.4 Black box testing

- Black box testing is done to find incorrect or missing function
- Interface error
- Errors in external database access
- Performance errors
- Initialization and termination errors

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

CHAPTER – 7

INSTALLATION

7. INSTALLATION

7.1. HARDWARE REQUIREMENTS

The Below Hardware Configuration were used in both Server and Client machines when developing.

Laptop	: Lenovo ideapad
Processor	: AMD PRO A4-4350B
Installed memory (RAM)	: 4.00 GB
System type	: 64-bit Operating System, x64-based processor
Hard Disk Drive	: 415 GB

7.2. SOFTWARE REQUIREMENT

The Below Software Configuration was used in machines when developing.

SERVER

Operating System	: Windows 8/10/11
Technology Used	: C#.NET, SQL SERVER
Database	: SQL SERVER Database
Browser	: Chrome / Microsoft Edge

CLIENT

Operating System	: Windows 8/10/11
Browser	: Chrome (or) Microsoft Edge

CHAPTER – 8
MAINTENANCE

8.MAINTENANCE

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

Corrective Maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

Adaptive Maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

Perceptive Maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

Preventive Maintenance

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques

CHAPTER – 9
APPLICATION

9. APPLICATION

Improved Customer Management

An accounts management system can help a fruits shop improve their customer management by keeping track of their customer's details and sending promotions and updates.

Streamlined Sales Transactions

An accounts management system can help a fruits shop automate generating invoices and receipts to save time and reduce errors.

Accurate Revenue and Profit Calculation

An accounts management system can help a fruits shop calculate their revenue and profit accurately to provide insights into the shop's financial performance.

Data Analysis

An accounts management system can help a fruits shop analyze their transaction data to identify trends and patterns and optimize their operations.

Improved Financial Management

An accounts management system can help a fruits shop keep track of their expenses, revenue, and profit and identify areas of the business that are not profitable.

CHAPTER – 10

CONCLUSION

10.CONCLUSION

Conclusion:

In conclusion, an accounts management system for a fruits shop can have numerous applications, including improved customer management, streamlined sales transactions, accurate revenue and profit calculation, data analysis, and improved financial management. Such a system can help a fruits shop optimize its operations and enhance its financial performance.

Future Works:

In the future, developing a mobile app for an accounts management system can make it even more convenient and accessible for the shop owners and managers to manage their finances and operations. A mobile app can allow them to access the system from anywhere and at any time, providing real-time updates on transactions and other crucial information. Additionally, with the increasing demand for mobile apps, a well-designed and user-friendly app can help a fruits shop attract more customers and boost their revenue.

CHAPTER – 11
BIBLIOGRAPHY

11. BIBLIOGRAPHY

Books for References:

- "Database Systems: Design, Implementation, and Management" by Carlos Coronel and Steven Morris: This book provides an in-depth guide to database systems design and management, including database connectivity and SQL queries.
- "Beginning Database Design Solutions" by Rod Stephens: This book covers the basics of database design, normalization, and creating a database schema.
- "Professional ASP.NET 4.5 in C# and VB" by Jason N. Gaylord, Christian Wenz, Pranav Rastogi, Todd Miranda, and Scott Hanselman: This book provides a comprehensive guide to developing web applications using ASP.NET, including database connectivity and SQL queries.
- "Learning C# by Developing Games with Unity" by Harrison Ferrone: While this book is focused on game development, it provides a practical introduction to C# programming language, including database connectivity.
- "C# 9.0 in a Nutshell: The Definitive Reference" by Joseph Albahari and Eric Johanssen: This book is an extensive reference guide to C# programming language, including database connectivity and SQL queries.

Web references:

C#.NET

- <https://msdn.microsoft.com/> www.w3schools.com/php/default.php
- <https://stackoverflow.com/> www.cristiandarie.ro/php-tutorial/
- <https://www.codeproject.com/>

SQL SERVER

- <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>
- <https://blog.sqlauthority.com/>
- <https://www.brentozar.com/sql/>

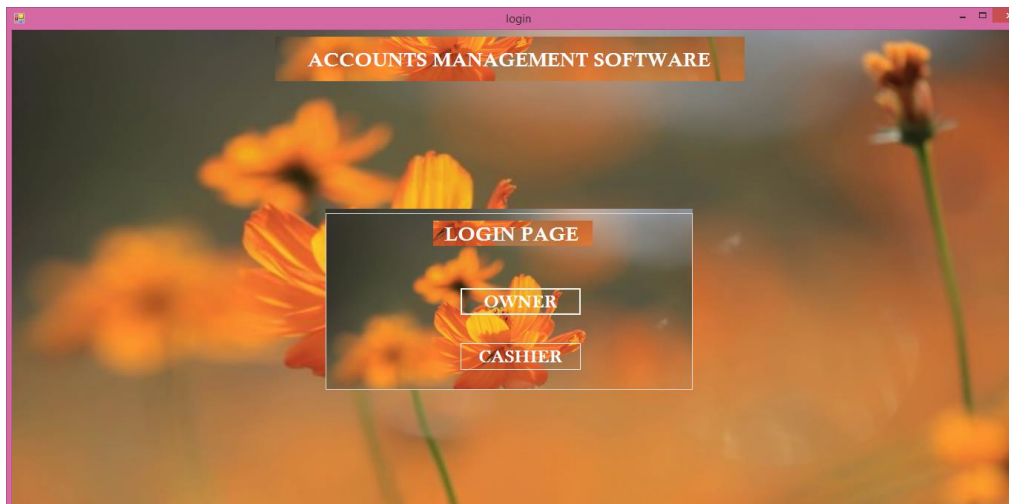
CHAPTER – 12

APPENDICES

12. APPENDICES

12.1. SCREENSHOT:

Login Page:



Admin Login:



Data Entry:



Cashier Login:



Expenses:

BACK

DATAS
[DAILY EXPENSE DATA](#)
[PURCHASE EXPENSE DATA](#)
[SALARY EXPENSE DATA](#)

ACCOUNTS MANAGEMENT SOFTWARE
EXPENSES
DAILY EXPENSES
MONTHLY EXPENSES

Daily Expense:

BACK

DATE : 04/10/2023

ACCOUNTS MANAGEMENT SOFTWARE
EXPENSE

SUPPLIERS

S.NO	DAILY SUPPLIERS	AMOUNT
1	MILK	
2	COFFEE	
3	KOYYA	
4	LEMON	
5	MANGAI	
CLICK		TOTAL

SHOP

S.NO	ITEM NAME	AMOUNT
1		
2		
3		
4		
5		
CLICK		TOTAL

OWNER

S.NO	OWNER EXPENSE	AMOUNT
1		
2		
3		
4		
CLICK		TOTAL

DEBT

S.NO	NAME	AMOUNT
1		
2		
3		
4		
5		
CLICK		TOTAL

FINAL CALCULATION

TOTAL SALES	
TOTAL EXPENSE	VIEW
G.PAY AMOUNT	
SHOP CASH	
CALCULATE	

AMOUNT CORRECT	
EXCESS	
SHORTAGE	
INSERT	

Monthly Expense:

BACK

ACCOUNTS MANAGEMENT SOFTWARE

MONTHLY EXPENSES

SUPPLIERS EXPENSE

SALARY EXPENSE

This screenshot shows a mobile application interface for 'ACCOUNTS MANAGEMENT SOFTWARE'. On the left, there is a vertical orange bar with a 'BACK' button. The main screen has a background of orange flowers. A central white box contains the title 'MONTHLY EXPENSES' and two options: 'SUPPLIERS EXPENSE' and 'SALARY EXPENSE'.

Supplier Expense:

BACK

DATE: 04/10/2023

ACCOUNTS MANAGEMENT SOFTWARE

PURCHASE DETAILS

PRODUCT NAME

PRODUCT AMOUNT

SUPPLIER NAME

SUPPLIER.PH.NO

SAVE

This screenshot shows a mobile application interface for 'ACCOUNTS MANAGEMENT SOFTWARE'. On the left, there is a vertical orange bar with a 'BACK' button. The main screen has a background of orange flowers. At the top left, there is a 'DATE' field showing '04/10/2023'. A central white box contains the title 'PURCHASE DETAILS' and four input fields: 'PRODUCT NAME' (with a dropdown arrow), 'PRODUCT AMOUNT', 'SUPPLIER NAME', and 'SUPPLIER.PH.NO'. Below these fields is a green 'SAVE' button. On the right side of the screen, there is a vertical orange bar with several icons: a magnifying glass, a circular arrow, a Windows logo, a document icon, and a gear icon.

Salary Expense:

[BACK](#)

ACCOUNTS MANAGEMENT SOFTWARE

SALARY REGISTRATION

NAME

IN -TIME

09:13:47:PM

OUT -TIME

09:13:47:PM

SALARY-1HR

DATE

04/10/2023

CALCULATE

WORKING HOURS

SALARY

SAVE

Source Code:

Login Page:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace ImportExport1
{
    public partial class login : Form
    {
        public login()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            adminlogin adm = new adminlogin();
            adm.Show();
            this.Hide();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            CASHIER_LOGIN adm = new CASHIER_LOGIN();
            adm.Show();
            this.Hide();
        }
    }
}
```

Datas:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace ImportExport1
{
    public partial class Form2 : Form
    {
        public Form2()
        {

```



```

        InitializeComponent();
    }
    private void linkLabel1_LinkClicked_1(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        DAILYEXPDATA adm = new DAILYEXPDATA();
        adm.Show();
        this.Hide();
    }

    private void linkLabel2_LinkClicked_1(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        PURCHASE_DATA adm = new PURCHASE_DATA();
        adm.Show();
        this.Hide();
    }

    private void linkLabel3_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        SALARY_DATAS adm = new SALARY_DATAS();
        adm.Show();
        this.Hide();
    }

    private void linkLabel6_LinkClicked_1(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        ADD_WORKERS adm = new ADD_WORKERS();
        adm.Show();
        this.Hide();
    }

    private void linkLabel8_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        WEEKLY_SUPPLIERS adm = new WEEKLY_SUPPLIERS();
        adm.Show();
        this.Hide();
    }

    private void linkLabel9_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        WORKERS_DETAILS adm = new WORKERS_DETAILS();
        adm.Show();
        this.Hide();
    }

    private void linkLabel10_LinkClicked(object sender,
    LinkLabelLinkClickedEventArgs e)
    {
        DEALERS_DETAILS adm = new DEALERS_DETAILS();
        adm.Show();
    }

```



```

        this.Hide();
    }

    private void linkLabel11_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        WORKERS_SALARY adm = new WORKERS_SALARY();
        adm.Show();
        this.Hide();
    }

    private void linkLabel12_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        PURCHASE_PAYMENTS adm = new PURCHASE_PAYMENTS();
        adm.Show();
        this.Hide();
    }

    private void linkLabel4_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        admin adm = new admin();
        adm.Show();
        this.Hide();
    }

    private void linkLabel15_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        login adm = new login();
        adm.Show();
        this.Hide();
    }

    private void groupBox1_Enter_2(object sender, EventArgs e)
    {
    }

    private void button3_Click(object sender, EventArgs e)
    {
    }

    private void linkLabel15_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        login adm = new login();
        adm.Show();
        this.Hide();
    }
}

```

Daily Expense:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace ImportExport1
{
    public partial class exp : Form
    {
        SqlConnection con;
        SqlCommand cmd;
        SqlDataReader dr;

        public exp()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int num1 = int.Parse(label35.Text);
            int num2 = int.Parse(label41.Text);
            int num3 = int.Parse(label44.Text);
            int num4 = int.Parse(label45.Text);
            int Res = num1 + num2 + num3 + num4;
            label28.Text = Res.ToString();
        }

        private void button2_Click(object sender, EventArgs e)
        {
        }

        private void exp_Load(object sender, EventArgs e)
        {
            con = new SqlConnection("Data
Source=PERSONAL\\SQLEXPRESS;Initial
Catalog=ImportExport;Integrated Security=True");
            con.Open();

            textBox1.Text = DateTime.Now.ToShortDateString();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
```

```

cmd.CommandText = "insert into report values('" + textBox1.Text +
"', '" + label15.Text + "', '" + label9.Text + "', '" + textBox2.Text + "', '" +
label10.Text + "', '" + textBox3.Text + "', '" + label11.Text + "', '" +
textBox4.Text + "', '" + label12.Text + "', '" + textBox5.Text + "', '" +
label13.Text + "', '" + textBox6.Text + "', '" + label14.Text + "', '" +
label15.Text + "', '" + label16.Text + "', '" + textBox50.Text + "', '" +
textBox43.Text + "', '" + textBox51.Text + "', '" + textBox42.Text + "', '" +
textBox52.Text + "', '" + textBox41.Text + "', '" + textBox53.Text + "', '" +
textBox40.Text + "', '" + textBox49.Text + "', '" + textBox39.Text + "', '" +
label29.Text + "', '" + label41.Text + "', '" + label17.Text + "', '" +
textBox45.Text + "', '" + textBox19.Text + "', '" + textBox44.Text + "', '" +
textBox18.Text + "', '" + textBox38.Text + "', '" + textBox17.Text + "', '" +
textBox15.Text + "', '" + textBox16.Text + "', '" + label14.Text + "', '" +
label44.Text + "', '" + label33.Text + "', '" + textBox56.Text + "', '" +
textBox25.Text + "', '" + textBox55.Text + "', '" + textBox24.Text + "', '" +
textBox54.Text + "', '" + textBox23.Text + "', '" + textBox48.Text + "', '" +
textBox22.Text + "', '" + textBox47.Text + "', '" + textBox21.Text + "', '" +
label43.Text + "', '" + label45.Text + "', '" + linkLabel6.Text + "', '" +
label25.Text + "', '" + textBox28.Text + "', '" + label24.Text + "', '" +
label28.Text + "', '" + label23.Text + "', '" + textBox12.Text + "', '" +
label22.Text + "', '" + textBox11.Text + "', '" + label21.Text + "', '" +
label27.Text + "', '" + label20.Text + "', '" + label31.Text + "', '" +
label19.Text + "', '" + label32.Text + "')";

```

```

cmd.ExecuteNonQuery();

```

```

con.Close();

```

```

con.Open();

```

```

MessageBox.Show("record insert sucessfully");

```

```

}

```

```

private void linkLabel3_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)

```

```

{

```

```

WEEKLY_SUPPLIERS adm = new WEEKLY_SUPPLIERS();

```

```

adm.Show();

```

```

this.Hide();

```

```

}

```

```

private void linkLabel11_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)

```

```

{

```

```

Form2 adm = new Form2();

```

```

adm.Show();

```

```

this.Hide();

```

```

}

```

```

private void linkLabel4_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)

```

```

{

```

```

        sal adm = new sal();
        adm.Show();
        this.Hide();

    }

    private void linkLabel5_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        totalexp1 adm = new totalexp1();
        adm.Show();
        this.Hide();

    }

private void textBox7_Enter(object sender, EventArgs e)
{
    int num1 = int.Parse(textBox2.Text);
    int num2 = int.Parse(textBox3.Text);
    int num3 = int.Parse(textBox4.Text);
    int num4 = int.Parse(textBox5.Text);
    int num5 = int.Parse(textBox6.Text);
    int Res = num1 + num2 + num3 + num4 + num5;

}

private void textBox26_Enter(object sender, EventArgs e)
{
    int num61 = int.Parse(textBox43.Text);
    int num71 = int.Parse(textBox42.Text);
    int num81 = int.Parse(textBox41.Text);
    int num91 = int.Parse(textBox40.Text);
    int num101 = int.Parse(textBox39.Text);
    int Res = num61 + num71 + num81 + num91 + num101;

}

private void textBox14_Enter(object sender, EventArgs e)
{
    int num10 = int.Parse(textBox19.Text);
    int num11 = int.Parse(textBox18.Text);
    int num12 = int.Parse(textBox17.Text);
    int num13 = int.Parse(textBox16.Text);
    int Res = num10 + num11 + num12 + num13;

}

private void textBox20_Enter(object sender, EventArgs e)
{
    int num11 = int.Parse(textBox25.Text);
    int num12 = int.Parse(textBox24.Text);
    int num13 = int.Parse(textBox23.Text);
    int num14 = int.Parse(textBox22.Text);

```

```

        int num15 = int.Parse(textBox21.Text);
        int Res = num11 + num12 + num13 + num14 + num15;

    }

    private void button2_Click_1(object sender, EventArgs e)
    {
        SqlCommand cmd = con.CreateCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "Delete from baghya";
        cmd.ExecuteNonQuery();
        con.Close();

        MessageBox.Show("All Records Removed sucessfully");
    }

    private void label9_Click(object sender, EventArgs e)
    {

    }

    private void textBox26_Enter_1(object sender, EventArgs e)
    {
        int num16 = int.Parse(textBox43.Text);
        int num17 = int.Parse(textBox42.Text);
        int num18 = int.Parse(textBox41.Text);
        int num19 = int.Parse(textBox40.Text);
        int num20 = int.Parse(textBox39.Text);
        int Res = num16 + num17 + num18 + num19 + num20;

    }

    private void textBox7_Enter_1(object sender, EventArgs e)
    {
        int num11 = int.Parse(textBox2.Text);
        int num12 = int.Parse(textBox3.Text);
        int num13 = int.Parse(textBox4.Text);
        int num14 = int.Parse(textBox5.Text);
        int num15 = int.Parse(textBox6.Text);
        int Res = num11 + num12 + num13 + num14 + num15;

    }

    private void textBox14_Enter_1(object sender, EventArgs e)
    {
        int num21 = int.Parse(textBox19.Text);
        int num22 = int.Parse(textBox18.Text);
        int num23 = int.Parse(textBox17.Text);
        int num24 = int.Parse(textBox16.Text);
        int Res = num21 + num22 + num23 + num24;

    }

```

```

private void textBox20_Enter_1(object sender, EventArgs e)
{
    int num25 = int.Parse(textBox25.Text);
    int num26 = int.Parse(textBox24.Text);
    int num27 = int.Parse(textBox23.Text);
    int num28 = int.Parse(textBox22.Text);
    int num29 = int.Parse(textBox21.Text);
    int Res = num25 + num26 + num27 + num28 + num29;
    // textBox20.Text = Res.ToString();
}

private void textBox2_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox3.Focus();
    }
}

private void textBox2_KeyDown_1(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox3.Focus();
    }
}

private void textBox3_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox4.Focus();
    }
}

private void textBox4_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox5.Focus();
    }
}

private void textBox5_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox6.Focus();
    }
}

```

```

private void textBox6_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button2.Focus();
    }
}

private void textBox7_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox50.Focus();
    }
}

private void textBox50_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox43.Focus();
    }
}

private void textBox43_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox51.Focus();
    }
}

private void textBox51_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox42.Focus();
    }
}

private void textBox42_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox52.Focus();
    }
}

private void textBox52_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)

```

```

        {
            textBox41.Focus();
        }
    }

private void textBox41_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox53.Focus();
    }
}

private void textBox53_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox40.Focus();
    }
}

private void textBox40_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox49.Focus();
    }
}

private void textBox49_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox39.Focus();
    }
}

private void textBox39_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button5.Focus();
    }
}

private void textBox26_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox45.Focus();
    }
}

```



```

    }

    private void textBox45_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox19.Focus();
        }
    }

    private void textBox19_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox44.Focus();
        }
    }

    private void textBox44_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox18.Focus();
        }
    }

    private void textBox18_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox38.Focus();
        }
    }

    private void textBox38_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox17.Focus();
        }
    }

    private void textBox17_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox15.Focus();
        }
    }

```

```

private void textBox15_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox16.Focus();
    }
}

private void textBox16_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button6.Focus();
    }
}

private void textBox14_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox56.Focus();
    }
}

private void textBox56_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox25.Focus();
    }
}

private void textBox25_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox55.Focus();
    }
}

private void textBox55_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox24.Focus();
    }
}

private void textBox24_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)

```

```

        {
            textBox54.Focus();
        }
    }

private void textBox54_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox23.Focus();
    }
}

private void textBox23_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox48.Focus();
    }
}

private void textBox48_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox22.Focus();
    }
}

private void textBox22_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox47.Focus();
    }
}

private void textBox47_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox21.Focus();
    }
}

private void textBox21_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button7.Focus();
    }
}

```

```

private void textBox20_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button1.Focus();
    }
}

private void linkLabel1_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
{
    expensess adm = new expensess();
    adm.Show();
    this.Hide();
}

private void linkLabel5_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
{
    totalexp1 adm = new totalexp1();
    adm.Show();
    this.Hide();
}

private void linkLabel4_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
{
    sal adm = new sal();
    adm.Show();
    this.Hide();
}

private void linkLabel3_LinkClicked_1(object sender,
LinkLabelLinkClickedEventArgs e)
{
    Form2 adm = new Form2();
    adm.Show();
    this.Hide();
}

private void button2_Click_2(object sender, EventArgs e)
{
    MessageBox.Show("All Records Removed sucessfully");
}

private void button8_Click(object sender, EventArgs e)
{
    int a, b ,c;
    int sale = int.Parse(textBox28.Text);
    int exp = int.Parse(label28.Text);
    int gpay = int.Parse(textBox12.Text);
    int shopcash = int.Parse(textBox11.Text);
    a = sale ;
    b = exp+ gpay + shopcash;
    c = b-a;
}

```

```

        if (c == 0)
        {
            label27.Text = c.ToString();
        }
        else if (c > 0) { label31.Text = c.ToString(); }
        else if (c < 0) { label32.Text = c.ToString(); }
        else { }
    }

    private void linkLabel7_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        report adm = new report();
        adm.Show();
        this.Hide();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        report adm = new report();
        adm.Show();
        this.Hide();
    }

    private void button2_Enter(object sender, EventArgs e)
    {
        int num11 = int.Parse(textBox2.Text);
        int num12 = int.Parse(textBox3.Text);
        int num13 = int.Parse(textBox4.Text);
        int num14 = int.Parse(textBox5.Text);
        int num15 = int.Parse(textBox6.Text);
        int Res = num11 + num12 + num13 + num14 + num15;
        label35.Text = Res.ToString();
    }

    private void button5_Enter(object sender, EventArgs e)
    {
        int num16 = int.Parse(textBox43.Text);
        int num17 = int.Parse(textBox42.Text);
        int num18 = int.Parse(textBox41.Text);
        int num19 = int.Parse(textBox40.Text);
        int num20 = int.Parse(textBox39.Text);
        int Res = num16 + num17 + num18 + num19 + num20;
        label41.Text = Res.ToString();
    }

    private void button6_Enter(object sender, EventArgs e)
    {
        int num21 = int.Parse(textBox19.Text);
        int num22 = int.Parse(textBox18.Text);
        int num23 = int.Parse(textBox17.Text);
        int num24 = int.Parse(textBox16.Text);
        int Res = num21 + num22 + num23 + num24;
    }

```

```

        label44.Text = Res.ToString();
    }

    private void button7_Enter(object sender, EventArgs e)
    {
        int num25 = int.Parse(textBox25.Text);
        int num26 = int.Parse(textBox24.Text);
        int num27 = int.Parse(textBox23.Text);
        int num28 = int.Parse(textBox22.Text);
        int num29 = int.Parse(textBox21.Text);
        int Res = num25 + num26 + num27 + num28 + num29;
        label45.Text = Res.ToString();
    }

    private void button2_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox50.Focus();
        }
    }

    private void button5_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox45.Focus();
        }
    }

    private void button6_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox56.Focus();
        }
    }

```