IT FUNDAMENTALS AND APPLICATIONS

AF-101

PRACTICAL WORKBOOK
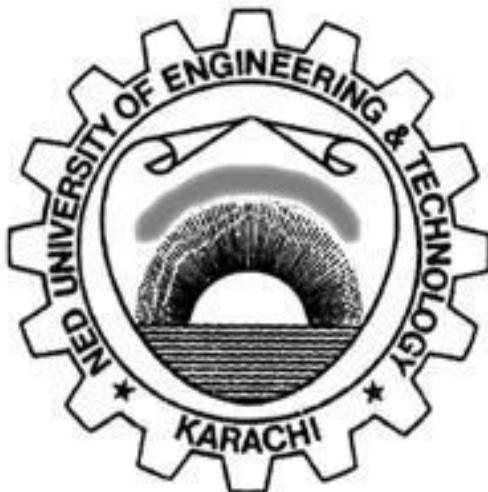
FOR

First Year

Batch: _____

NAME OF STUDENT: _____

CLASS ROLL NO.:_____SECTION: _____

DISCIPLINE: _____

SEMESTER: _____

DEPARTMENT OF PHYSICS

NED UNIVERSITY OF ENGINEERING &
TECHNOLOGY, KARACHI, PAKISTAN.

PRACTICAL WORK BOOK

For The Course

IT FUNDAMENTALS AND APPLICATIONS
(AF-101)

Prepared By:

Ms. Shazia Ahmed

Reviewed By:

Dr. Saba Javaid

Approved By:

The Board of Studies of the Department of Physics

<u>CERTIFICATE</u>

Certified that Mr./ Ms. _____ of class **First year**

Bearing Seat No: _____ Department _____

has completed the course **IT Fundamentals and Applications (AF-101)**

Practical as prescribed by the NED University of Engineering &

Technology, Karachi, for the Academic Session: _____

Date:_____   _____

Lab. Teacher

IT Fundamentals and Applications

INDEX

S. no.	Objects	Page no.	Remarks	Signature & Date
1.	Components and structure of a computer system.	5 - 8		
2.	Basic Document Formatting and Styles in MS Word	9 -13		
3.	Basic Formulas and Functions in MS Excel	14 - 15		
4.	Database Management and Querying in MS Access	16 - 26		
5.	Basic Programming Concepts in C Language	27 – 28		
6.	Control statements and loops in the C language	29 – 35		
7.	Website, Document, and Form Creation Using Google Tools	36 - 47		

Practical # 01

OBJECT:

To understand the basic components of a computer system, their structure, and functions.

THEORY:

INTRODUCTION TO COMPUTER HARDWARE

Computer hardware refers to the physical and tangible components of a computer system that can be seen and touched. These components work together to perform input, processing, storage, and output operations. Hardware includes devices such as the central processing unit (CPU), motherboard, memory, storage devices, input devices, and output devices.

Computer hardware forms the foundation of a computer system, as software cannot operate without hardware support. Understanding computer hardware helps users recognize different components, their functions, and how they contribute to the overall performance of a computer system. This knowledge is essential for basic system usage, maintenance, and troubleshooting.

DIFFERENCE BETWEEN HARDWARE AND SOFTWARE:

Hardware refers to the physical parts of a computer that can be seen and touched.
Software refers to programs and instructions that control and operate the hardware.

COMPONENTS OF A COMPUTER SYSTEM:

1. INPUT UNIT:
The input unit consists of devices connected to a computer that allow users to enter data and instructions. These devices convert the input into a form that the computer can understand and send it to the processor for further processing.

2. CENTRAL PROCESSING UNIT (CPU):
The CPU is the brain of the computer that controls all operations. It receives data from input devices, fetches and decodes instructions from memory, processes the data, and sends the results to memory or output devices.

MAIN PARTS OF CPU:

A. Arithmetic and Logic Unit (ALU):
The ALU performs arithmetic operations such as addition, subtraction, multiplication, and division, and logical operations like comparison and decision-making.

B. Control Unit (CU):
The Control Unit directs all activities of the computer. It interprets instructions and controls the flow of data between the CPU, memory, and input/output devices.

C. Memory Registers:
Registers are small, fast memory units inside the CPU that temporarily store data, instructions, and addresses being used during processing.

3. OUTPUT UNIT:
The output unit consists of devices that convert processed data from the CPU into a human-

readable form. Common devices include monitors, printers, and plotters, which display or print information for the user.

4. MOTHERBOARD:
The motherboard is the main circuit board that connects all computer components, including the CPU, memory, and storage. It distributes power, allows communication between parts, and connects external devices.

5. RANDOM ACCESS MEMORY (RAM):
RAM is the computer's short-term memory that temporarily stores data and programs while they are in use, allowing fast access for the CPU.

6. READ-ONLY MEMORY (ROM):
ROM is non-volatile memory that permanently stores essential instructions for booting the computer and basic hardware control. Unlike RAM, its contents are not erased when the computer is turned off.

7. POWER SUPPLY UNIT (PSU):
The PSU provides electrical power to all computer components, converting AC from a wall socket into the required DC voltages for the system.

8. PORTS
Ports are interfaces on a computer that allow external devices to connect. They enable communication between the computer and peripherals such as keyboards, mice, printers, and storage devices. Common types of ports include:

- **USB (Universal Serial Bus):** For connecting most peripherals
- **HDMI/VGA:** For monitors and projectors
- **Ethernet (RJ-45):** For wired network connections
- **Audio ports:** For speakers and microphones

Function: Ports transfer data, video, audio, and power between the computer and external devices.

9. OPERATING SYSTEM (OS)

The operating system is system software that manages computer hardware and software resources and provides services for programs. It acts as an intermediary between the user and the hardware.

Main Functions of an OS:

- Managing hardware components (CPU, memory, storage)
- Running and controlling application software
- Handling input/output operations
- Managing files and data
- Providing a user interface (GUI or command-line)

Examples: Windows, macOS, Linux, Android

ACTIVITIES:

1. Recognize the input and output devices connected to the computer.
   Steps:
   i.   Observe the computer setup in the lab.
   ii.   Identify the devices you can see.
   iii. Classify each device as **Input** or **Output**.

S.no	Device Name	Type(input/output)	Use
1			
2			
3			
4			

2. Observe and identify common computer ports.

Steps:
   i.   Look at the back and sides of the CPU casing.
   ii. Identify these ports:

   - USB
   - HDMI
   - VGA
   - Audio ports
   - Ethernet (RJ-45)

   iii. Write down what devices can be connected to each port.

S.no	Port	Location	Device(s) connected
1			
2			
3			
4			
5			

3. Identify and write the names of the ports below.

_____   _____

_____   _____

_____

4. Find system information from the operating system.

Steps:
1. On Windows: Right-click **This PC → Properties**
2. Record:
   o Installed RAM
   o Processor
   o Storage type and capacity
   o Operating System and version

S.no	Component	Details
1	RAM	
2	Processor	
3	Storage	
4	OS	

Result and Discussion:

_____

_____

_____

_____

_____

_____

Practical # 02

OBJECT:

Gain proficiency in Word formatting, covering styles, Table of Contents, and cross-references.

THEORY:

In this lab, you will learn to apply advanced formatting techniques in Microsoft Word. These skills are essential for creating professional-looking documents, especially for long reports, theses, and research papers. The lab will guide you through using custom styles, managing section breaks, creating a table of contents, and applying cross-references for easier navigation.

PROCEDURE

Follow the steps given below to complete each task.

Creating and Applying Custom Styles

1. Open a new blank document in Microsoft Word.
2. Create a custom style for the title of the document:
   - Go to the **Home** tab → open the **Styles Pane** → click **New Style**.
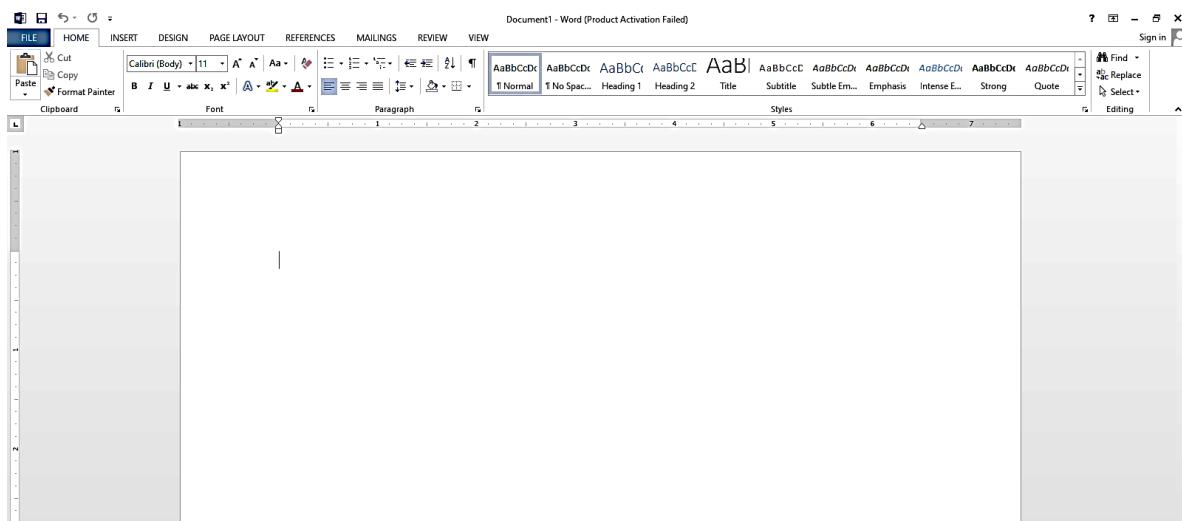
   - Set the style name as "Lab Title" and format it with **24pt Times New Roman, Blue color**.
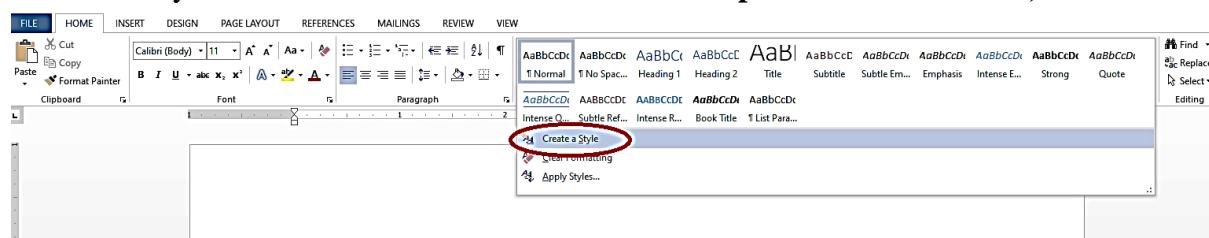
3. Apply this style to the title of your document.
4. Create another custom style for section headers (e.g., "Lab Section Header") with 16pt Bold, Underlined text, with Blue color, and apply it to each section heading.
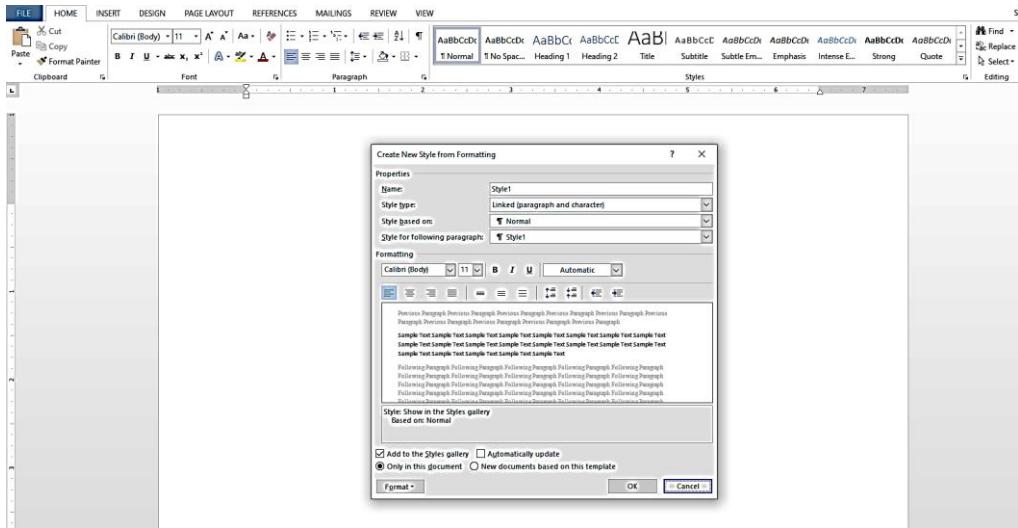
Using Section Breaks

1. Add at least three sections to your document: Introduction, Method, and Results.
   - Insert section breaks between these sections:
   - Go to the **Layout** tab → click on **Breaks** → choose **Next Page** under **Section Breaks**.
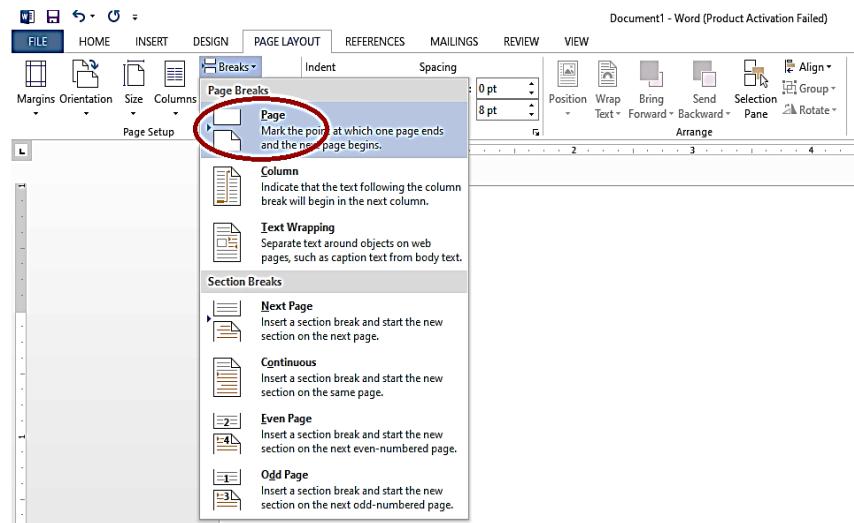
2. Verify that each section begins on a new page.

Managing Long Documents

1. Open the **Navigation Pane** by going to the **View** tab → check the box for **Navigation Pane**.
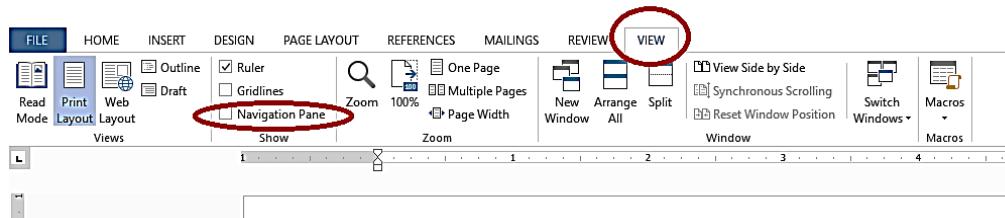
    i.   Apply **Heading 1** style to all section headers (Introduction, Method, Results).

   ii.   Use the **Navigation Pane** to scroll through the sections of your document.

  iii.   You can click on any of the section headers in the Navigation Pane to quickly jump to that section of your document. This is particularly useful in long documents, as it allows you to easily navigate between sections without scrolling manually.
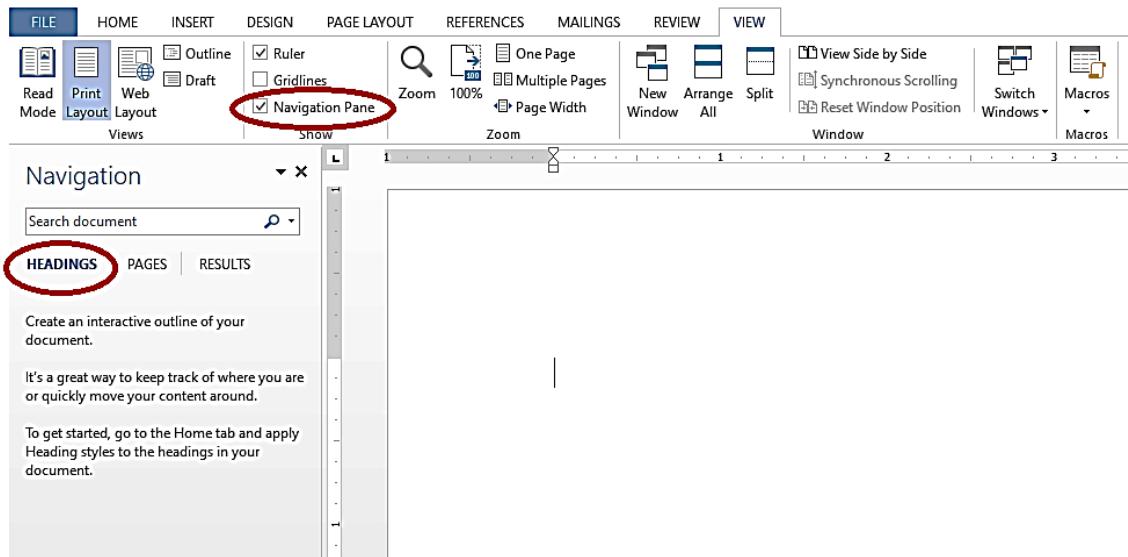
2. Inserting a Table of Contents (TOC)

- Place your cursor at the beginning of your document.
- Insert a Table of Contents:
- Go to the References tab → click on Table of Contents → select an automatic TOC format.
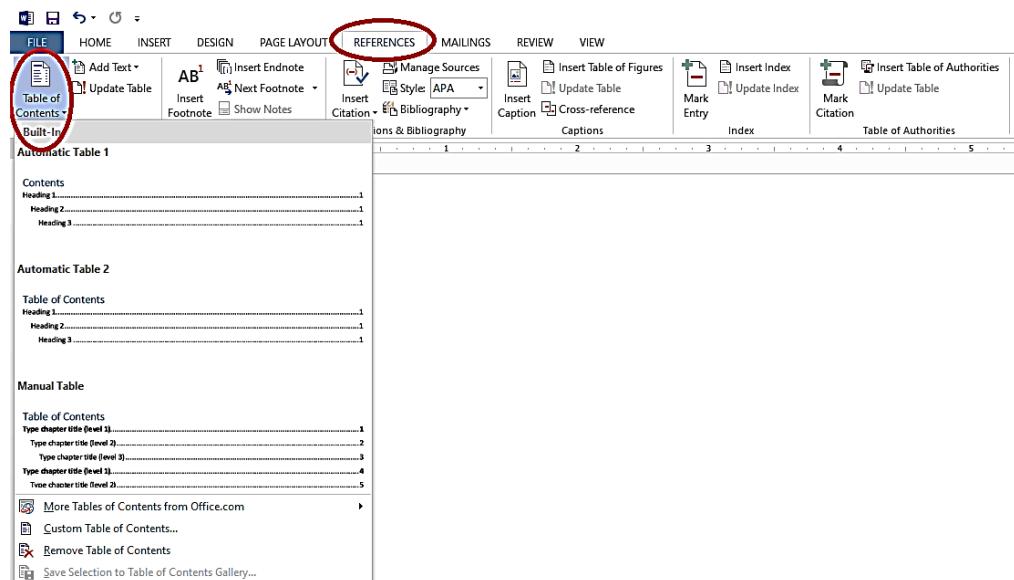
The TOC should reflect all your headings.

1. Add more sections and update the TOC by right-clicking on it → select **Update Field**.

Applying Cross-References
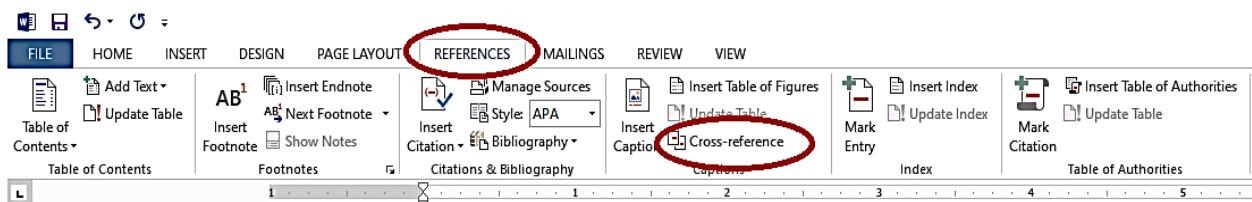
- Insert a reference to the "Results" section within the "Introduction" section.

- Go to the **References** tab → click on **Cross-reference**.
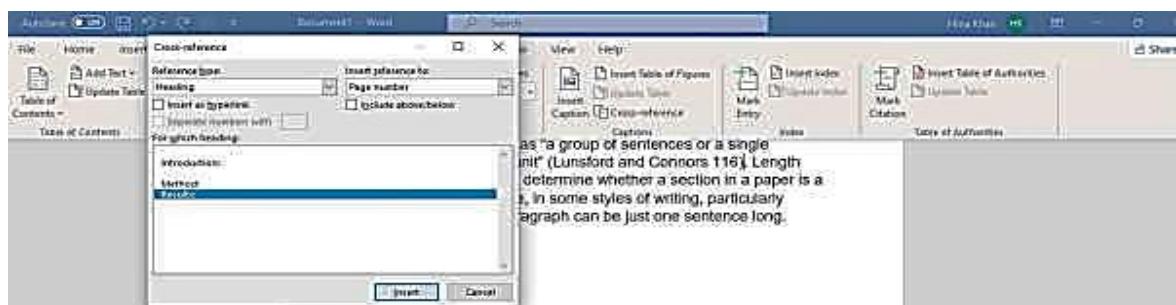
- Choose **Heading** under **Reference Type** and select **Page number** under **Insert reference to**.

- This will link the Introduction to the Results section, showing the page number dynamically.
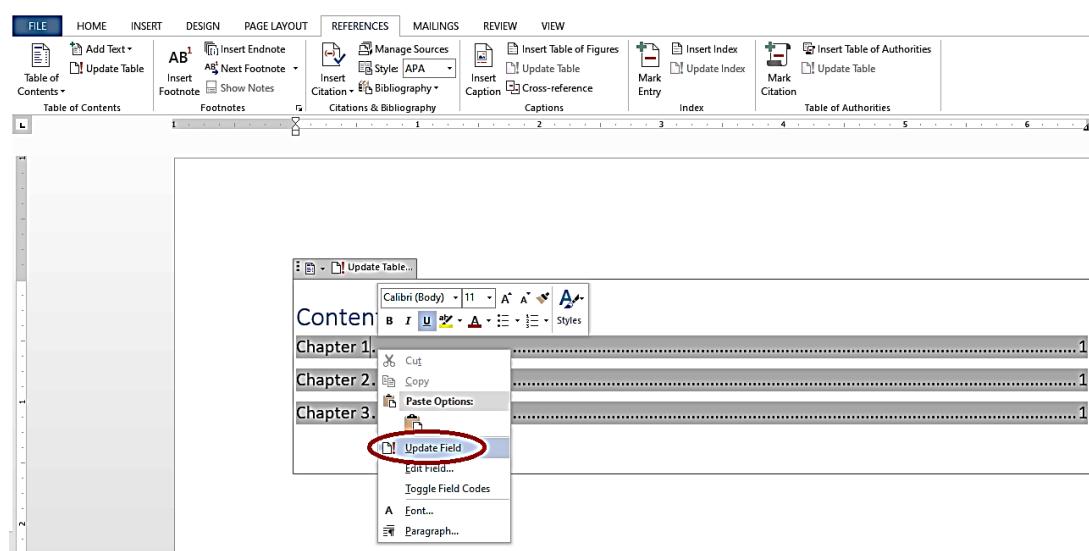
RESULT

Create an MS Word document on any topic using styles, section breaks, table of contents (TOC), and cross-references**,** and attach the final printout here.

Practical # 03

OBJECT:

To understand and apply basic formulas and commonly used functions in MS Excel.

THEORY:

1. Introduction to Excel Formulae

A formula is an expression that performs a calculation on values. All formulas in Excel begin with an equal sign (=).

For example,

- Addition: `=A1 + B1`
- Subtraction: `=A1 - B1`
- Multiplication: `=A1 * B1`
- Division: `=A1 / B1`
- Manual Average: `=(A1 + A2 + A3) / 3`

2. Understanding Excel Functions

Functions are built-into formulas that perform specific calculations quickly.

For example,

- **MEDIAN:** `=MEDIAN(A1:A5)` – Returns the middle value in a data set.
- **MODE:** `=MODE(A1:A5)` – Returns the most frequently occurring value.
- **IF Function:** `=IF(condition, value_if_true, value_if_false)`
  - Example: `=IF(B2>=50, "Pass", "Fail")`
- **SUM:** `= SUM(A1:A5)` – Adds values in a range.
- **AVERAGE:** `= AVERAGE(A1:A5)` – Calculates mean value.
- **MAX:** `= MAX(A1:A5)` – Finds the highest number.
- **MIN:** `= MIN(A1:A5)` – Finds the lowest number.
- **COUNT:** `= COUNT(A1:A5)` – Counts numeric cells.
- **COUNTA:** `= COUNTA(A1:A5)` – Counts non-empty cells.

3. Referencing in Excel (Relative vs Absolute)

Relative Reference (A1)
- Adjusts automatically when the formula is copied.

Absolute Reference (A1, A$1, $A1)
- `$A$1` locks both row and column.
- `A$1` locks only the row.
- `$A1` locks only the column.

4. Creating graphs

Graphs in Excel are used to visually represent data for easier understanding and comparison. They help identify trends, patterns, and differences quickly. Charts like column, bar, line, and pie make data interpretation simple and effective.

PROCEDURE:

Adding Formulae:

1. Open a new Excel sheet.
2. Enter any five numbers in cells A1 to A5.
3. Use formulae to add, subtract, multiply, and divide selected pairs of numbers.
4. Calculate the average of all five numbers manually.

Adding Functions:

1. Enter five numbers in cells C1 to C5.
2. Find the **median** and **mode**.
3. Use an **IF function** to check whether each value is above or below average.
4. Use SUM, AVERAGE, MAX, and MIN to analyze the marks.
5. Use COUNT and COUNTA to understand the number of filled cells.

Adding Graphs:

1. Select the data to be plotted.
2. Go to **Insert → Chart** and choose the chart type.
3. Add a title and label the axes as needed.

ACTIVITY:

Student Marks Sheet

Enter the following data in Excel:

Student	Math	Physics	Chemistry
Ali	78	85	90
Sara	88	92	84
Ahmed	67	70	75

Perform the following tasks:

1. **Calculate Total Marks:** Use `=SUM(B2:D2)`
2. **Calculate Average Marks:** Use `=AVERAGE(B2:D2)`
3. **Find Highest Math Mark:** Use `=MAX(B2:B4)`
4. **Count Number of Students:** Use `=COUNTA(A2:A4)`
5. **Find Lowest Physics Mark:** Use `=MIN(C2:C4)`
6. **Create a Chart/Graph:**
   - Insert a **column chart** or **bar chart** to visually compare the marks of all students in Math, Physics, and Chemistry.
   - Label the chart title as **"Student Marks Comparison"**.
   - Ensure axes are properly labeled (Students on X-axis, Marks on Y-axis).

RESULT:

- Submit the completed Excel file containing all data, formulae, and functions used.
- Ensure that all formulae (e.g., SUM, AVERAGE, IF) and functions are correctly applied.
- Attach charts/graphs along with the file.

Practical # 04

OBJECT:

To create, manage, and query a relational database using MS Access.

THEORY:

Microsoft Access is a database management system (DBMS) used to store, organize, and manage large amounts of data efficiently. It allows users to create databases consisting of tables, queries, forms, and reports. Tables are used to store data in rows and columns, queries help in retrieving and manipulating specific data, forms provide a user-friendly interface for data entry, and reports are used to present data in a structured and printable format. MS Access reduces data redundancy, ensures data consistency, and enables quick searching and updating of records, making it suitable for managing databases in schools, offices, and organizations.

PROCEDURE:

Creating a blank database

1. Select a Blank Database

Open Microsoft Access and click Blank desktop database from the Welcome screen.
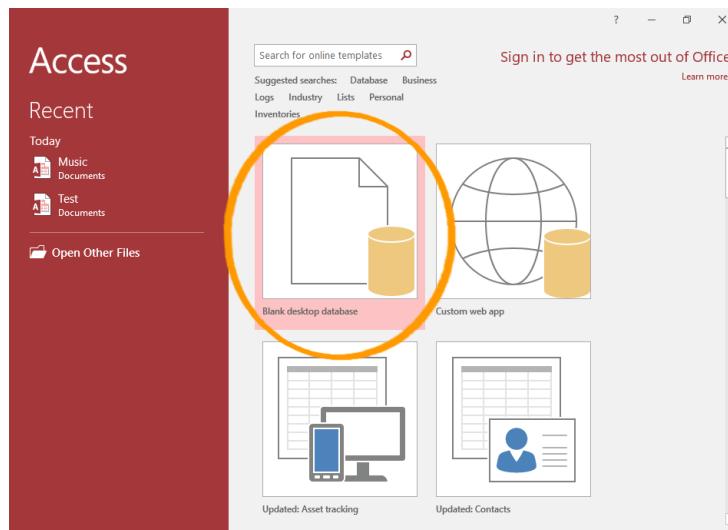
2. Name the Database

Name the database at the prompt, then click Create.
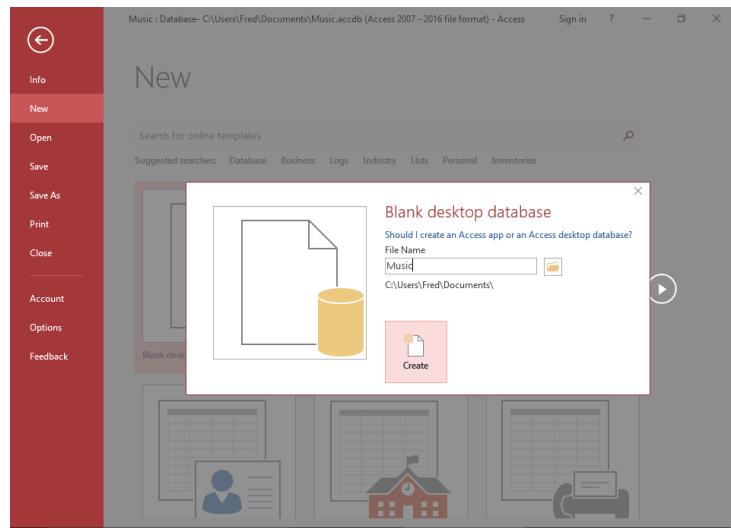
3. The Result

A blank database is created. However, it does contain one table that is ready to be configured.
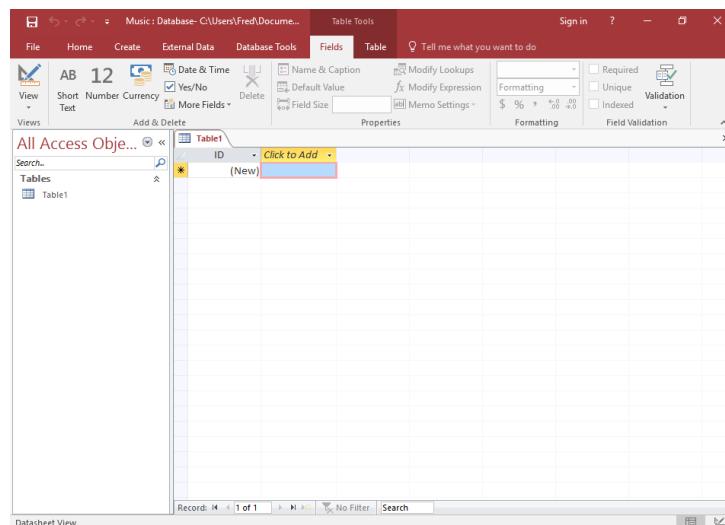
Adding a New Table

We now have all we need to create a new table. We are going to add the table Shippers. This table will hold data on who the shipping agent is for all our orders.

To create the table, go to the Create Menu (top ribbon menu) and click on Table.
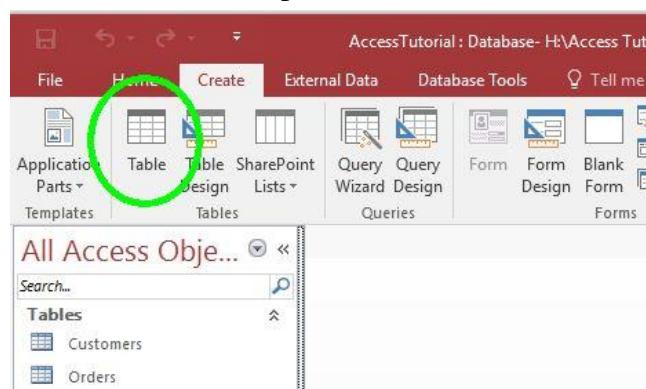
You will see a new table in datasheet view. By default, Access gives it the name Table1 until you save it. Do that now, by right-clicking on the Table tab, and clicking "save. Call the new table *Shippers*.
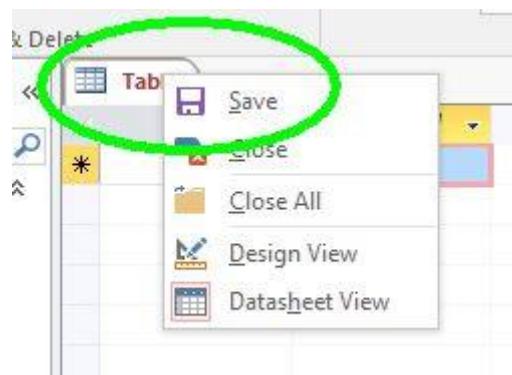
Now click to change to the table design view. You will see Access wants you to have a primary key field, so by default, it gives you the field "ID" with the type autonumber, and the little primary icon is present. Just to make things very clear, we are going to rename *ID* and call it ShipperID. Click on the field and edit the name. You can leave the datatype as autonumber.
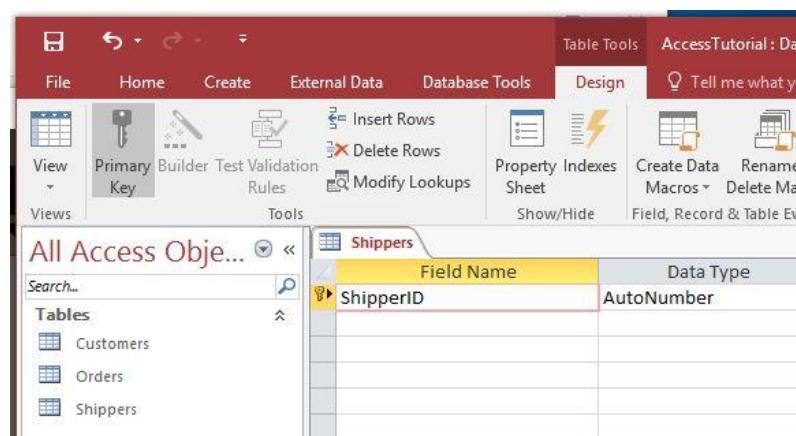
Now we want to add all the other fields to our table. If this were a database you were designing, you would need to think about which fields you need and what types they should have, but we are going to follow the Northwind example, so add fields until your table matches the image below.

Table Relationships

Now that we have a new table created, it's time to look at relationships.

Go back and look at the Order table. The third field in it is called CustomerID. If we look at the first record in the order table (order #30), we see it is for customer #27. It would be natural to assume this is Karen Toh - if we open the customer table, it has a field called CustomerID - *the exact same name.* We can also see there is a customer with an ID of 27 - Karen Toh. It is perfectly obvious this must be the same customer, correct?

Not so fast. The fact that the fields have the same name is irrelevant. **Access will not associate the CustomerID fields in the two tables with each other unless we link them together.** We do this in the relationships window. Go to the design menu ribbon and click on Relationships.
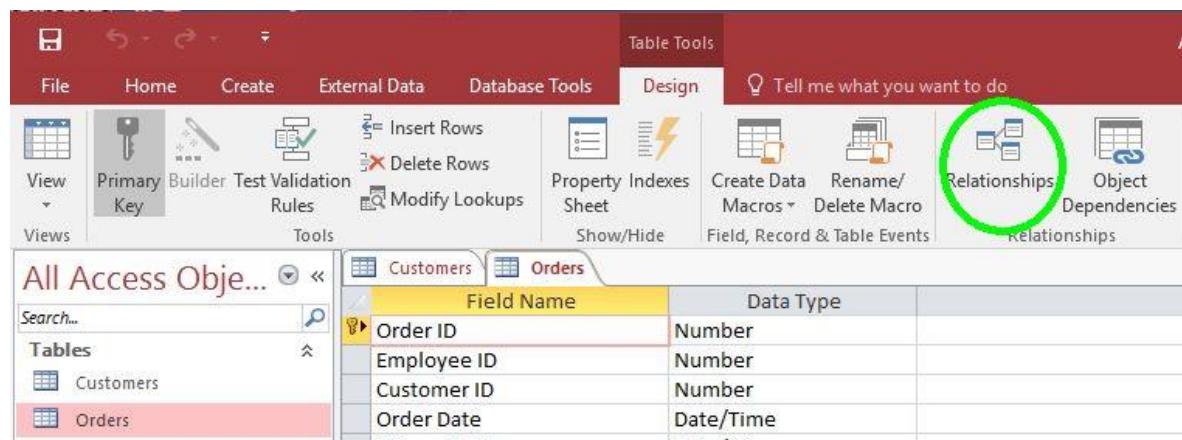
You should see that the Order and Customer tables are there, and there is a line connecting them. This means they *are* linked - that we can, in fact, be sure that order #30 is for customer #27, Karen Toh.
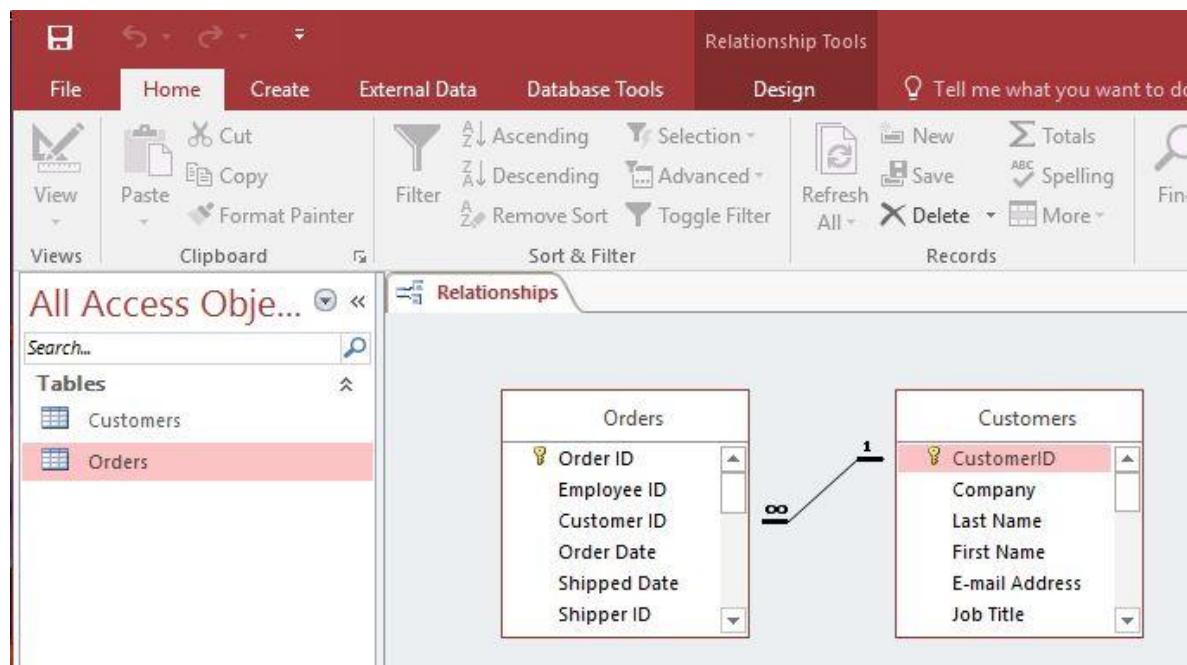
To link two tables together, we need a field in each table that holds the same data. Often, those two fields have the same name, for convenience, but they don't have to. **They must, however, have the same datatype.**

A table relationship works by matching the data in those fields — so in the Order table in our example, customerID is a number, as it is in Customer (note that you can match number/autonumber, but for every other datatype pairing they should be exactly the same, e.g., Date/Date or Short Text / Short Text. This makes sense because we are telling Access *that this is the same data.* The reason we want to make the link is so that we can find the customer record that goes with this order. We also want to be sure that there *is* a customer record for every Order, so we link to the **Primary Key** of the Customer table because that field has to be there, and has to have a value.

When we add a field to a table for the purpose of linking to another table, we call it a **Foreign Key**.

To link two tables together, make sure they are both displayed in the Relationships window. Then, drag the foreign key field onto the primary key field.

Try this now - click on the relationship line you see between Orders and Customers. You can tell when it is highlighted, it will be a little bit thicker. Now, **delete** it. Once it is gone, drag Orders.CustomerID onto the Customers.CustomerID field (you have to drop it on the right field, so be accurate. You should see the following dialog box.
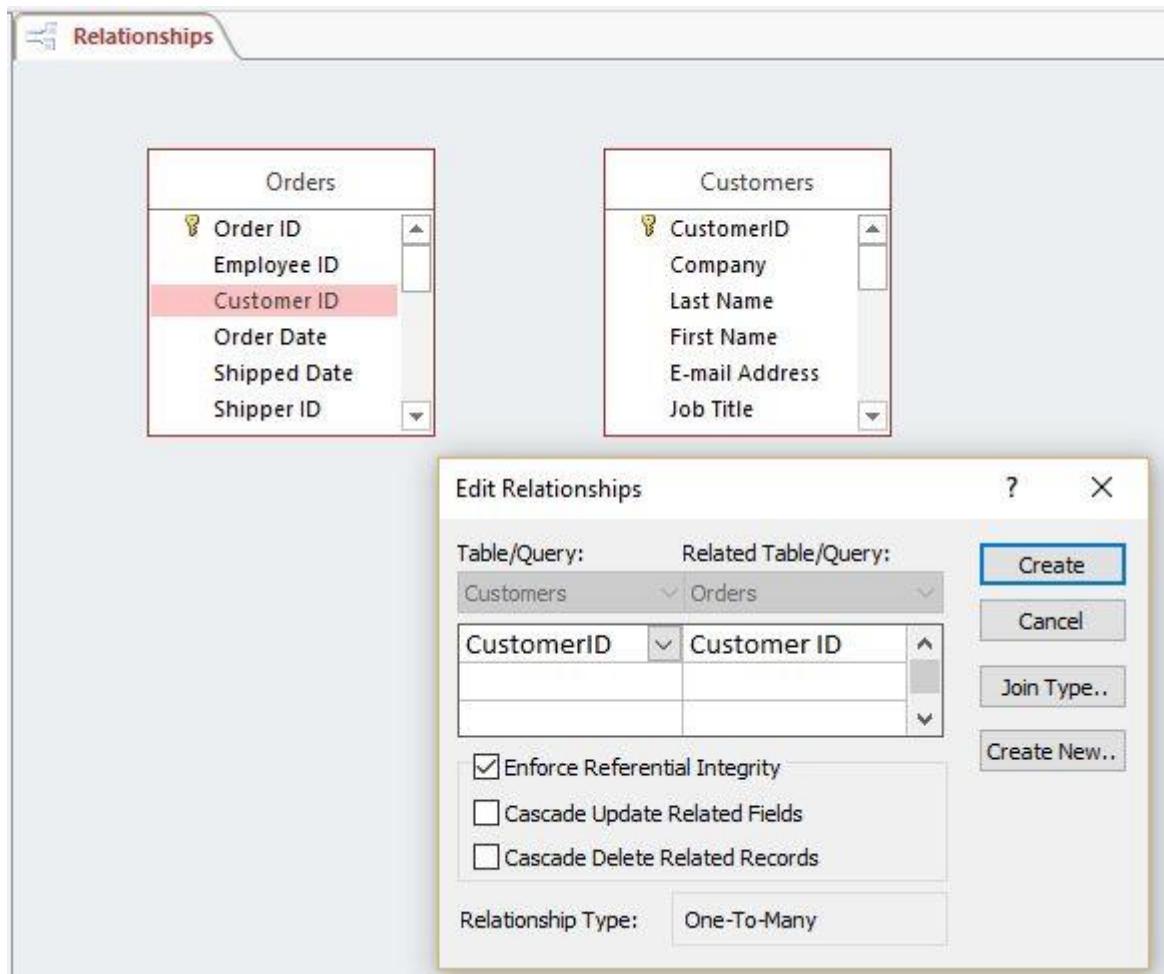

Adding our new table

Now we want to add our Shippers table to the mix, so that we will be able to find the shipper for each order. Remember, we need two fields to create a relationship - one in the Shipper table (the primary key of the Shipper table, in fact), and one in the Order table. Shippers.ShipperID already exists, but Order.ShipperId *doesn't*, so **step one is add the foreign key to Order**.

Do this by opening the design view of Order, and adding ShipperID to the end of the list (the order of fields doesn't matter to Access). Remember that the datatypes must match. Because ShipperID is an autonumber in Shippers, we use Number in the Orders table. You can't use autonumber here because it is the Shippers table that controls how numbers will be auto-assigned. The Orders field just has to match those. Also, you can't have Autonumber on anything but a primary key.


Now we can link them together.

Go to the Relationship window. "Drag" the Shippers table from the list of tables on the left onto the canvas.
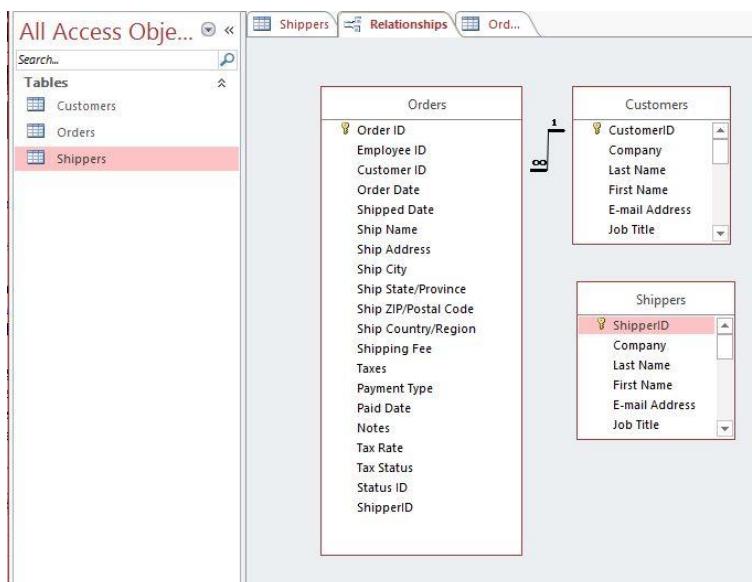
Now link the **Shippers** and **Orders** tables in the same way you linked **Orders** and **Customers**. Drag **Orders.ShipperID** onto the **Shippers.ShipperID** field. If you encounter any issues, the tables may be open—close them using the small **X** on the right side of each table window and then try again.
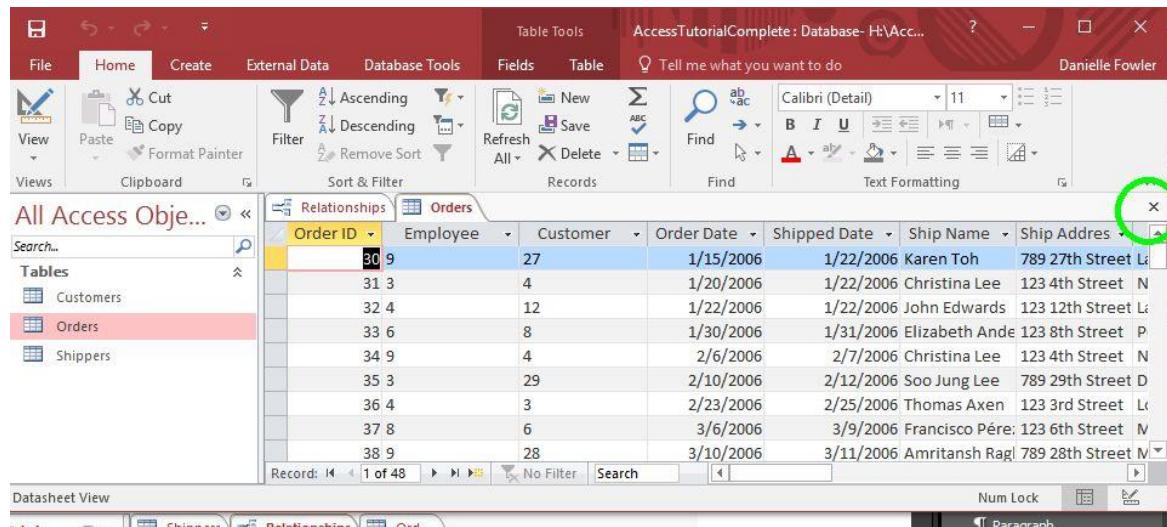
Make sure only Relationships is open.

One Last Step

We have one step left. So far we have:

1. created the Shippers table
2. filled it with data
3. linked it to Orders

The final step is to enter data into the ShipperID field in the Orders table. Currently, this field is empty for all records. We need to add values so that each order is linked to a shipper. In the Northwind database, the Orders datasheet is set up to display the shipper name instead of the ShipperID number, but we will not cover that here—simply enter a variety of values. Keep in mind that there are only three shippers with IDs 1, 2, and 3, so only these values are allowed. If you want to link an order to a new shipper (for example, shipper 4), you must first add that shipper to the Shippers table before entering 4 in the Orders table.

Create a Query
1. Click the **Create** tab on the ribbon.
2. Click the **Query Design** button.
   You can also use the Query Wizard to create a query. Click the **Create** tab on the ribbon and click the **Query Wizard** button in the Queries group.
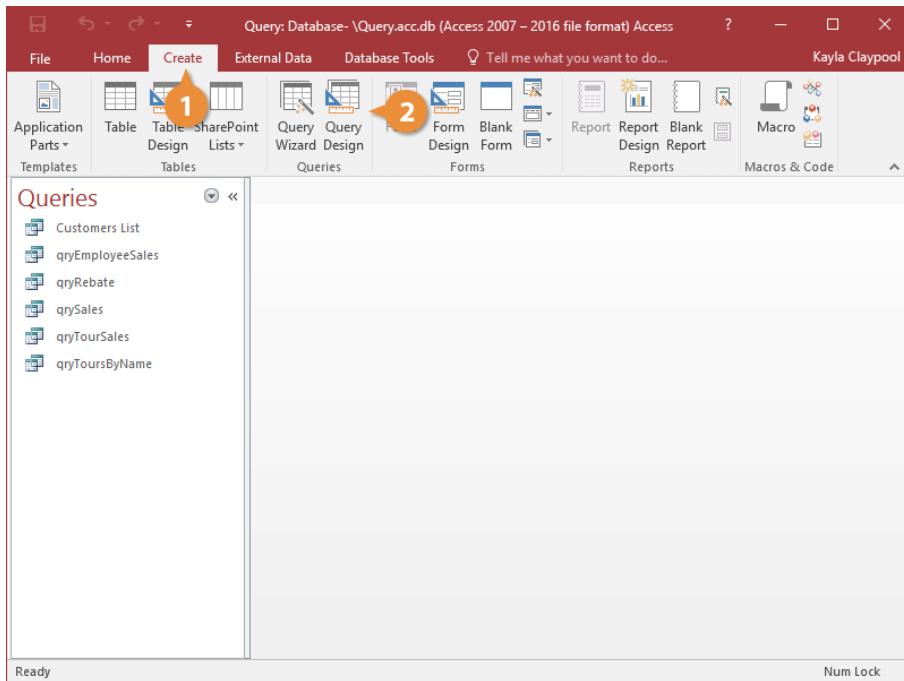
1. The Show Table dialog box appears.
   Follow the onscreen instructions to create the query.
2. Select the table(s) you want to add to the query and click **Add**.
3. Click **Close**.
   You can also add tables to a query by dragging them from the Navigation Pane to the top half of the Query window.
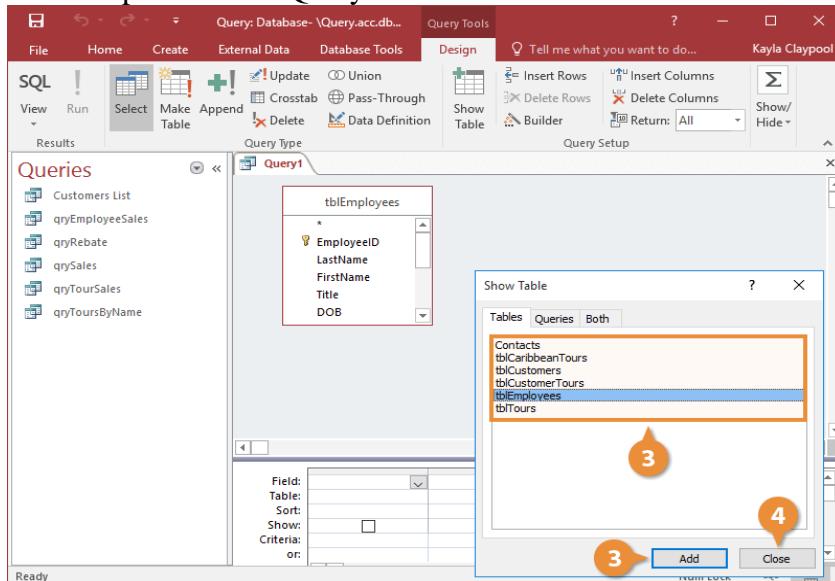
1. The **Query window** opens in **Design View** and is divided into two sections. The top section displays a box containing all the fields from the table added to the query, while the bottom section shows the **design grid**, where you select the fields to include in your query results.
2. Double-click each field you want to add to the query. Alternatively, you can drag fields from the field list into the design grid. You may need to use the scroll bar in the field list to locate specific fields.
3. Next, specify any **criteria** for the query by entering them in the **Criteria** row of the design grid. For example, you can filter records where the **City** field contains "London" or use **K*** to return records that begin with the letter K.

If you want to use a field for filtering but do not want it to appear in the results, clear the **Show** checkbox for that field.
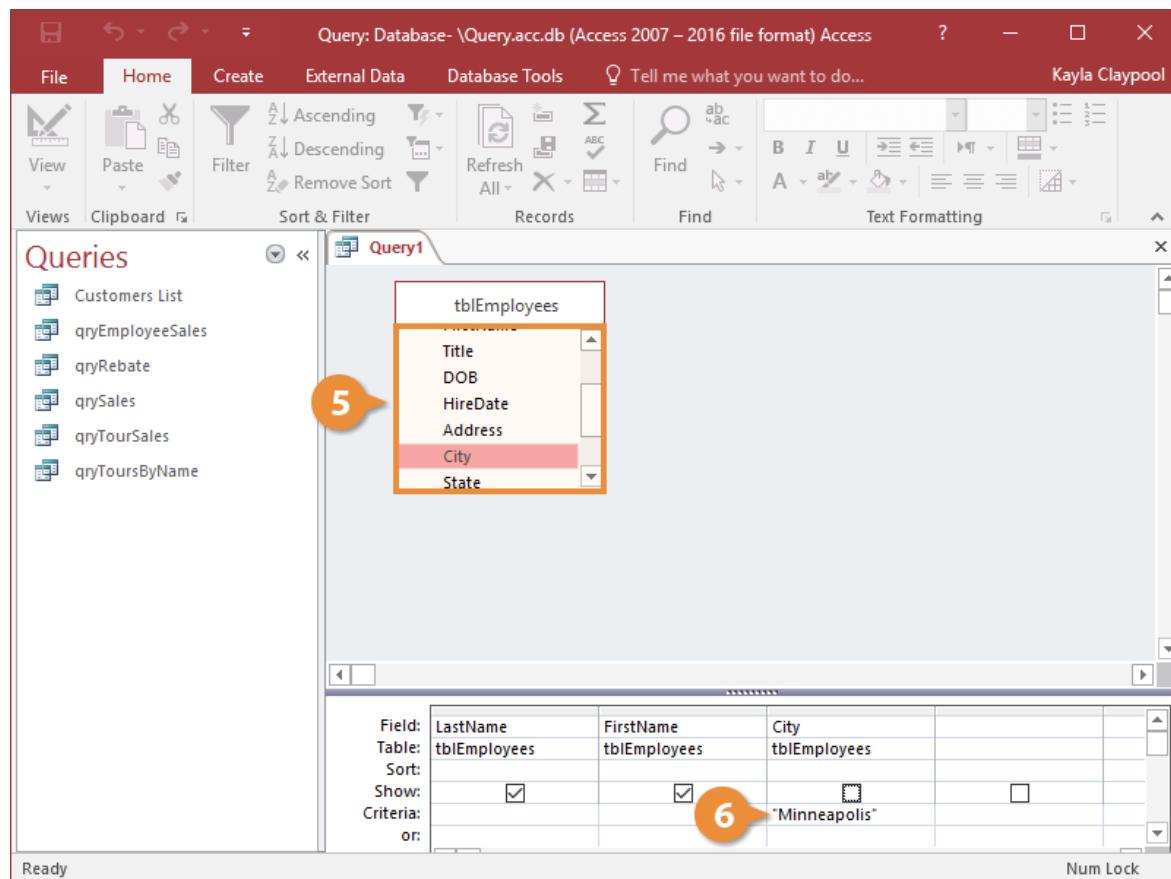
Save a Query

Once you have created a query, you'll want to save it to use later.
1. Click the **Save** button.
2. Enter a name for the query.
3. Click **OK**.
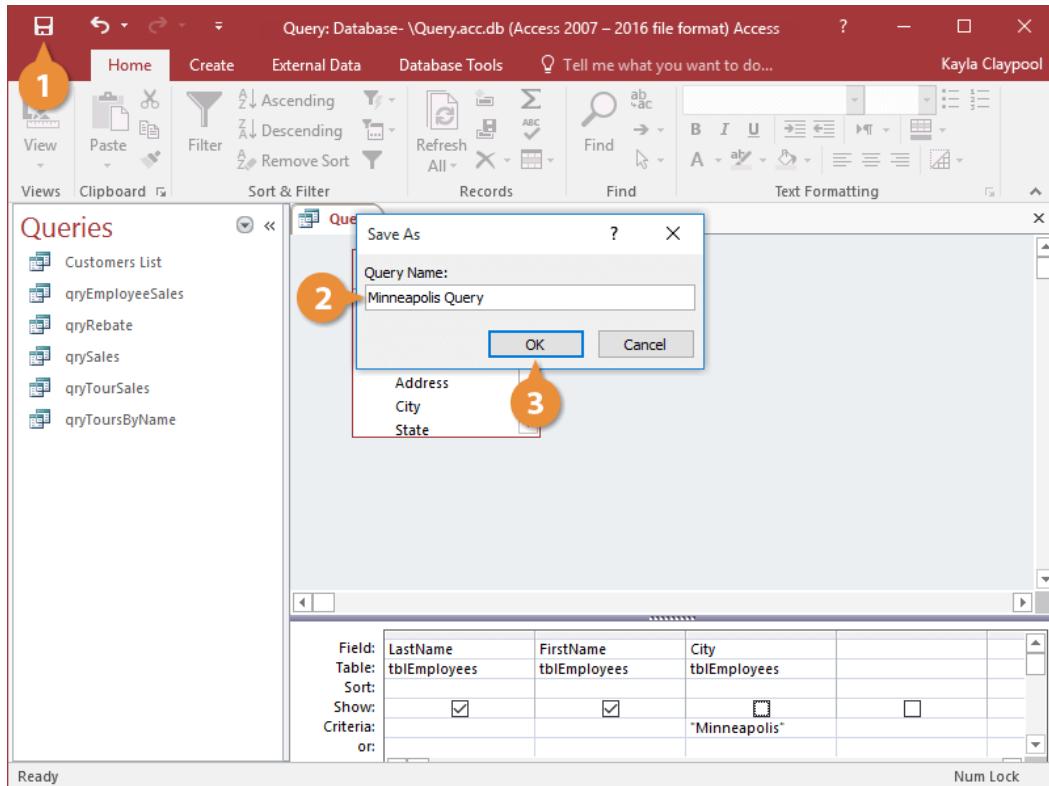
The query is saved and now appears in the Navigation Pane.

Run a Query

Your queries run when you open them through the Navigation Pane.

1. Open the query or click the **Run** button if you're in Design view.
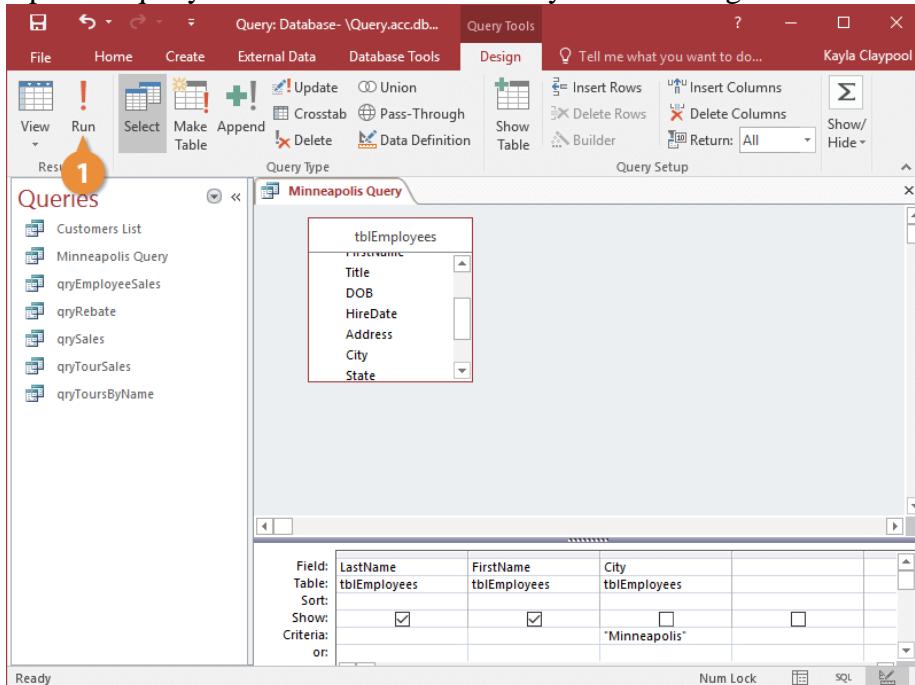
Access displays the results of the query.

ACTIVITY:

You are provided with the following sample data about students, courses, and their marks. Using this information, **create a database in MS Access** and then **generate the required queries**.

Student Data

StudentID	StudentName	Gender	Contact
1	Ali Khan	M	0301-1111111
2	Sara Ahmed	F	0302-2222222
3	Ahmed Raza	M	0303-3333333
4	Fatima Noor	F	0304-4444444

Course Data

CourseID	CourseName	CreditHours
101	Mathematics	3
102	Physics	4
103	Computer Science	3

Enrollment Data

Enroll ID	Student ID	Course ID	Marks	Semester
1	1	101	78	Fall 2024
2	1	102	85	Fall 2024
3	2	101	92	Spring 2024
4	3	103	67	Fall 2024
5	4	102	55	Spring 2024
6	4	103	88	Fall 2024

Using the above data:

Create an MS Access database with three tables (Students, Courses, Enrollments), enter the data, set relationships, and generate the following queries:

1. List all students enrolled in **Physics**.
2. Show students who scored **more than 80 marks**.
3. Display the **average marks** for each course.
4. Create a **parameter query** asking: *"Enter Semester"*.
5. Create a **Pass/Fail field** using the formula:

$$IIF([Marks] >= 50, "Pass", "Fail")$$

RESULT: Submit your file to the teacher after completion.

Practical # 05

OBJECT:

To understand and apply the basic programming concepts of the C language.

THEORY:

Computer Language:

A computer language is a formal system of communication that allows humans to give instructions to a computer so that it can perform specific tasks.

C Language Introduction:

C is a general-purpose procedural programming language initially developed by **Dennis Ritchie** in **1972** at Bell Laboratories of AT&T Labs. It was mainly created as a system programming language to write the **UNIX operating system**.

Basic Structure of a C Program:

	1	#include <stdio.h>	Header
	2	int main(void)	Main
BODY	3	{	
BODY	4	// This prints "Hello World"	Comment
BODY	5	printf("Hello World");	Statement
BODY	6	return 0;	Return
BODY	7	}	

1. #include <stdio.h> is a preprocessor directive**.** It tells the compiler to include the Standard Input Output library (stdio.h) in your program. stdio.h contains functions like printf() and scanf() for input and output.
   Some of the other libraries are

#include <math.h>	Mathematics: Provides mathematical functions such as `sqrt()`, `pow()`, `sin()`, `cos()`, and `log()`.
#include <string.h>	String Handling: Contains functions for manipulating strings, e.g., `strcpy()`, `strlen()`, `strcmp()`, `strcat()`.
#include <ctype.h>	Character Handling: Functions to check or convert characters, e.g., `toupper()`, `tolower()`, `isdigit()`.

2. Int main(void) is the **main function**, where the execution of the program starts. `Int` indicates that the function returns an **integer value** to the operating system. `(void)` means the function **does not take any parameters**.
3. // This prints "Hello World" is a single-line comment**.** Comments are ignored by the compiler and are used to explain the code for humans. In C, comments start with //.
4. printf("Hello World");
   • printf() is a function from stdio.h that prints text to the screen**.**
   • "Hello World" is a **s**tring literal and is displayed exactly as written.
   • The statement ends with a **s**emicolon, which marks the end of a C statement
5. return 0;
   • This statement **e**nds the main function and returns the value 0 to the operating system.
   • Returning 0 usually indicates that the program executed **s**uccessfully**.**
At the end, this closing curly brace } marks the end of the main function**.**

ACTIVITIES:

Type the following programs into a C language compiler, study and understand each line, and then run the code to observe and write the output just below the code.

```c
#include <stdio.h>

int main() {
    int a = 5, b = 3, sum;
    sum = a + b;
    printf("Sum = %d\n", sum);
    return 0;
}
```
Output:

```c
#include <stdio.h>

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    printf("Product = %d\n", num1 * num2);
    return 0;
}
```
Output:

```c
#include <stdio.h>

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if(num % 2 == 0)
        printf("%d is Even\n", num);
    else
        printf("%d is Odd\n", num);
    return 0;
}
```
Output:

```c
#include <stdio.h>

int main() {
    int i;
    for(i = 1; i <= 5; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```
Output:

Practical # 06

OBJECT:

To understand and use control statements and loops in the C language to control program flow and perform repetitive tasks.

THEORY:

Control statements (if, if-else, and switch-case)
Control statements in C help the computer execute a certain logical statement and decide whether to enable the control of the flow through a certain set of statements or not. Also, it is used to direct the execution of statements under certain conditions.

Iteration Statements(For loop, while loop and do-while loop)
Iteration statements are used to execute a particular set of instructions repeatedly until a particular condition is met or for a fixed number of iterations.

PROCEDURE and OBSERVATIONS:

Control statements

4. if-else Statement
The if-else statement, a vital part of control flow statements in C, executes blocks of code based on a true or false condition. Complex decisions often involve multiple if-else statements or an else-if statement, ensuring different outcomes for various conditions.
Example: Check Positive or Negative Number

```c
#include <stdio.h>
int main( ) {
    int a;
    printf("n Enter a number:");
    scanf("%d", &a);

    if(a>0)
    {
        printf( " n The number %d is positive.",a);
    }
    else
        printf("n The number %d is negative.",a);
    return 0;
}
```

Output:

5. Nested if and if-else Statements

It is also possible to embed or to nest if-else statements one within the other. Nesting is useful when multiple conditions need to be evaluated.

Example: Find the Greatest of Three Numbers

```c
#include <stdio.h>

int main() {
    int a = 6, b = 5, c = 10;

    if (a > b) {
        if (a > c) {
            printf("\nGreatest is: %d", a);
        } else {
            printf("\nGreatest is: %d", c);
        }
    } else if (b > c) {
        printf("\nGreatest is: %d", b);
    } else {
        printf("\nGreatest is: %d", c);
    }

    return 0;
}
```

Output:

6. Switch Statement

The switch-case statement is used for multi-way branching, where a variable or expression's value is matched against predefined cases. It's ideal for replacing long chains of if-else when testing for equality.

However, you have to be careful when using a switch statement since missing a break statement can cause fall-through behavior, executing subsequent cases, and leading to unexpected results.

Example: Print the Day of the Week

```c
#include <stdio.h>

int main() {
    int day;
    printf("Enter the number of the day: ");
    scanf("%d", &day);

    switch(day)
     {
        case 1:
            printf("Sunday");
            break;
        case 2:
            printf("Monday");
            break;
        case 3:
            printf("Tuesday");
            break;
        case 4:
            printf("Wednesday");
            break;
        case 5:
            printf("Thursday");
            break;
        case 6:
            printf("Friday");
            break;
        case 7:
            printf("Saturday");
            break;
        default:
            printf("Invalid choice");
    }

    return 0;
}
```

Output:

Iteration Statements

1. for Loop
The `for` loop is a pre-test loop, which evaluates the condition before the loop executes. It is best suited for situations where the number of iterations is known beforehand.

Example: Fibonacci Series

```c
#include <stdio.h>

int main() {
    int i, n, a = 0, b = 1, sum;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("%d %d", a, b);

    for(i = 2; i < n; i++) {
        sum = a + b;
        printf(" %d", sum);
        a = b;
        b = sum;
    }

    return 0;
}
```

e.g, If the input number is 7, the output will be:

```
0 1 1 2 3 5 8
```

Output:

2. While Loop

The while statement executes a block of statements repeatedly while a particular condition is true.

Example: Sum of Digits

```c
#include <stdio.h>

int main() {
    int n, a, sum = 0;
    printf("\nEnter a number: ");
    scanf("%d", &n);

    while(n > 0) {
        a = n % 10; // Extract the last digit
        sum = sum + a; // Add it to the sum
        n = n / 10; // Remove the last digit
    }

    printf("\nSum of the digits = %d", sum);
    return 0;
}
```

Output:

3. do-while loop

The do-while loop is exit-controlled, meaning the loop body executes at least once before the condition is checked.

The difference between while and do-while is that the while loop is entry-controlled; it tests the condition at the beginning of the loop and will not execute even once if the condition is false, whereas the do-while loop is exit-controlled; it tests the condition at the end of the loop after completing the first iteration.

For many applications, it is more natural to test for the continuation of a loop at the beginning rather than at the end of the loop. For this reason, the do-while statement is used less frequently than the while statement.

Example: Sum of Digits

```c
#include <stdio.h>

int main() {
    int n, a, sum = 0;
    printf("\nEnter a number: ");
    scanf("%d", &n);

    do {
        a = n % 10;
        sum = sum + a;
        n = n / 10;
    } while(n > 0);

    printf("\nSum of the digits = %d", sum);
    return 0;
}
```

Output:

ACTIVITIES:

1. Write a program that asks the user to enter a number and prints whether it is even or odd using if-else.
2. Write a program to find the sum of the first N natural numbers using a while loop.
3. Ask the user to enter positive numbers repeatedly until they enter 0. Use a do-while loop to calculate the sum of all entered numbers.
4. Write a program that asks the user to enter marks (0–100) and prints the grade based on the following: 90–100 → A, 80–89 → B, 70–79 → C, 60–69 → D, Below 60 → F.

RESULTS AND DISCUSSIONS

1. What are the steps C follows to run a program, from writing the code to displaying the output?

2. What is the name of the window where the program's output is displayed?

3. What conclusions did you draw from the programs you ran in this lab, and what do they show about how the C language works?

4. How did you debug errors or incorrect outputs during the lab?

5. How do input and output operations work in C programs?

6. Although C remained foundational for system-level programming, which newer languages have emerged as more versatile, safe, and application-oriented alternatives?

Practical # 07

OBJECT:
To understand and use basic IT applications, including Google Sites, Google Docs, and Google Forms, for creating websites, documents, and data collection forms.

THEROY:

1. Google Sites
Google Sites is a web-based platform that allows you to create websites for free. It's a simple website builder, but it offers enough features to host your blog, portfolio, or even classroom learning materials.

2. Google Docs
Google Docs is an online word processing application that allows users to create, edit, and format documents using the internet. It enables real-time collaboration, allowing multiple users to work on the same document simultaneously. Documents are automatically saved to Google Drive, reducing the risk of data loss. Google Docs also provides tools for inserting images, tables, and comments. It is widely used in education for assignments, reports, and group work.

3. Google Forms
Google Forms is an online tool used to create surveys, quizzes, and data collection forms. It allows users to design different types of questions such as multiple-choice, short answer, and checkboxes. Responses are collected automatically and can be viewed in summary or spreadsheet form. Google Forms supports automatic grading for quizzes and real-time response analysis. It is commonly used in education for assessments, feedback, and registrations.

PROCEDURE:

1.Google Sites

Steps to create a Google Site:
1. Go to sites.google.com.
2. Click Blank site to create a new site. Alternatively, you can choose a premade template to kickstart the process.
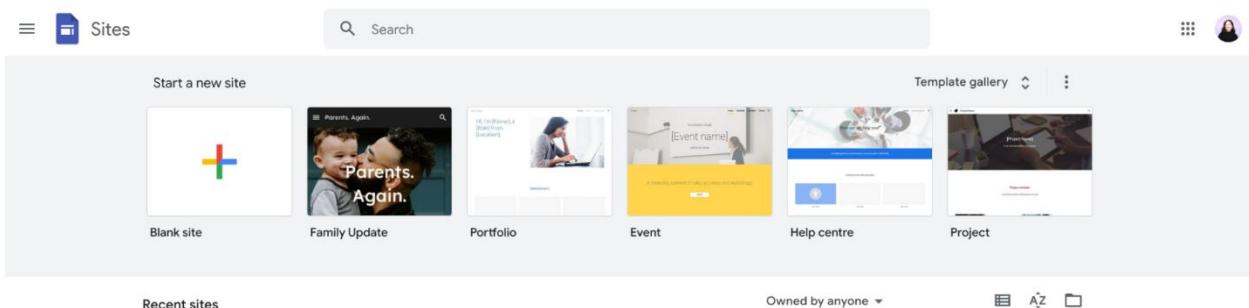
A blank page will automatically appear in the Google Sites editor.
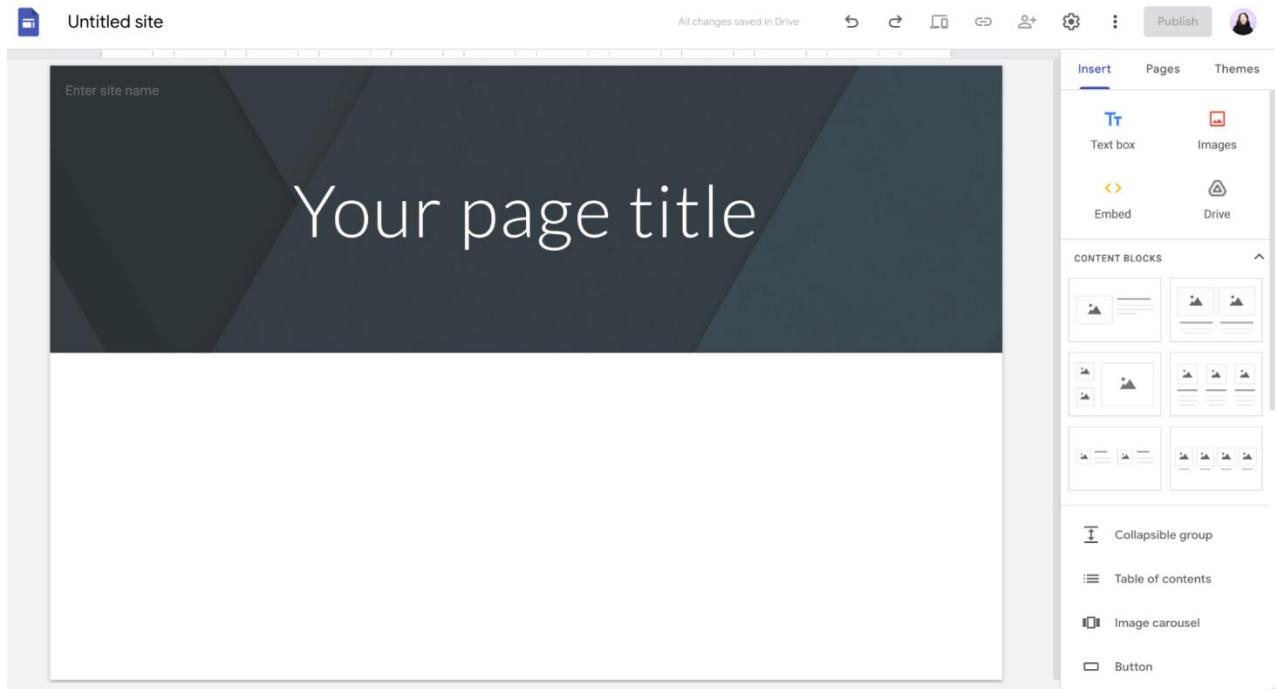
Now let's add and edit all the elements of a basic website.

- Create a header
- Choose a design theme
- Add text and media
- Incorporate Google apps
- Arrange and edit columns and sections
- Add and manage pages
- Publish and collaborate

Note: At any point, you can click the Preview icon above the editor to make sure your site is appearing the way you want. Google Sites offers previews for large screens, tablets, and phones.

Create a header

The header is the menu banner at the top of most web pages—the spot where you might see a company's logo and a series of navigation tabs like About, Services, and Contact us.
Note: If you created a blank site, the header block in the editor allows you to change the site name, add a page title, and customize how the header appears. For most of the premade templates, your changes are limited to only editing the text fields and adding a logo.

1. Enter your site name in the Enter site name field. If you're working from a premade template, this might already have a name, like Project Name, but you can change it. The name you enter here will also appear on your website.
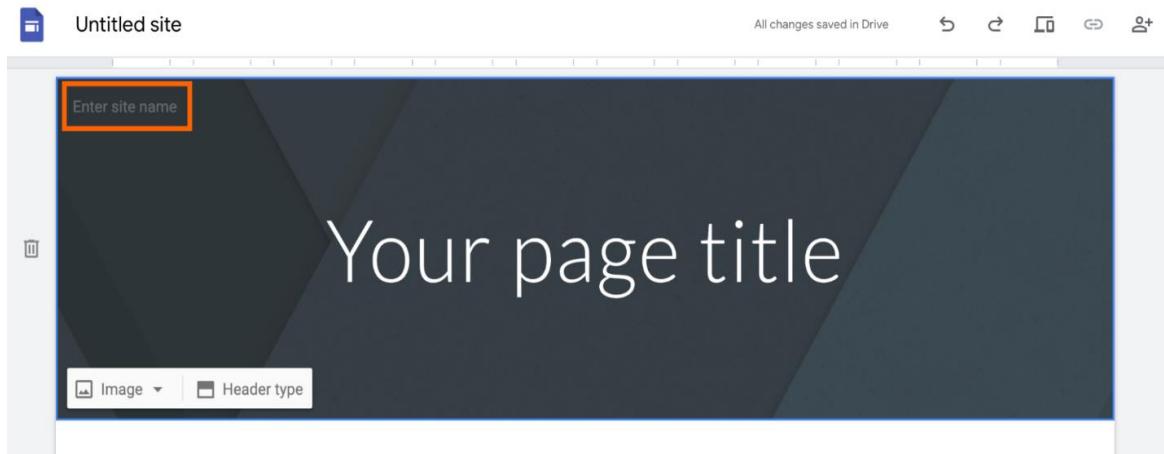
- Click Header type, and select how you want your header to appear: Cover, Large banner, Banner, or Title only. The first three options present your background image in various sizes, while the last displays only header text.
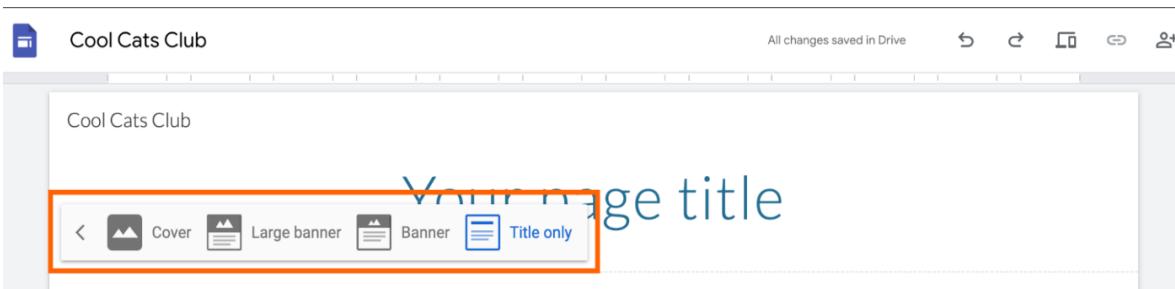
3. If your header type includes an image, click Image to upload a background image. Note: Your Google Sites banner image can be any size and aspect ratio, but in my experience, 1600 x 1068 pixels works well.
4. To reposition your banner image, click the Anchor image icon, which looks like an anchor, to adjust your image. The editor will also automatically adjust your header image for better readability. I recommend letting the editor do its thing, but if you want to restore the original image, click the Remove readability adjustment icon, which looks like a series of stars.
5. To add a logo, hover over the site name and click Add logo. Google recommends uploading an image that's at least 112 pixels tall.
6. Click the header text to enter your header title and edit its style, size, font, and alignment. Or delete the box if you don't want any text. To reposition the text box, hover over any corner of the text box until a four-way arrow appears. Then drag and drop the box to where you want.

Choose a design theme.

By default, the theme for a brand-new Google Sites web page is simple—literally. That's the design theme name. Here's how to modify the theme to add a little personality or match your branding.

1. In the side menu, click Themes.
2. Click one of the other theme options. Each theme comes with a few preset color schemes and fonts.

If the five premade themes aren't cutting it, click the Create theme icon, which looks like a plus sign (+), to make your own theme. Google Sites will save these designs for later, too.
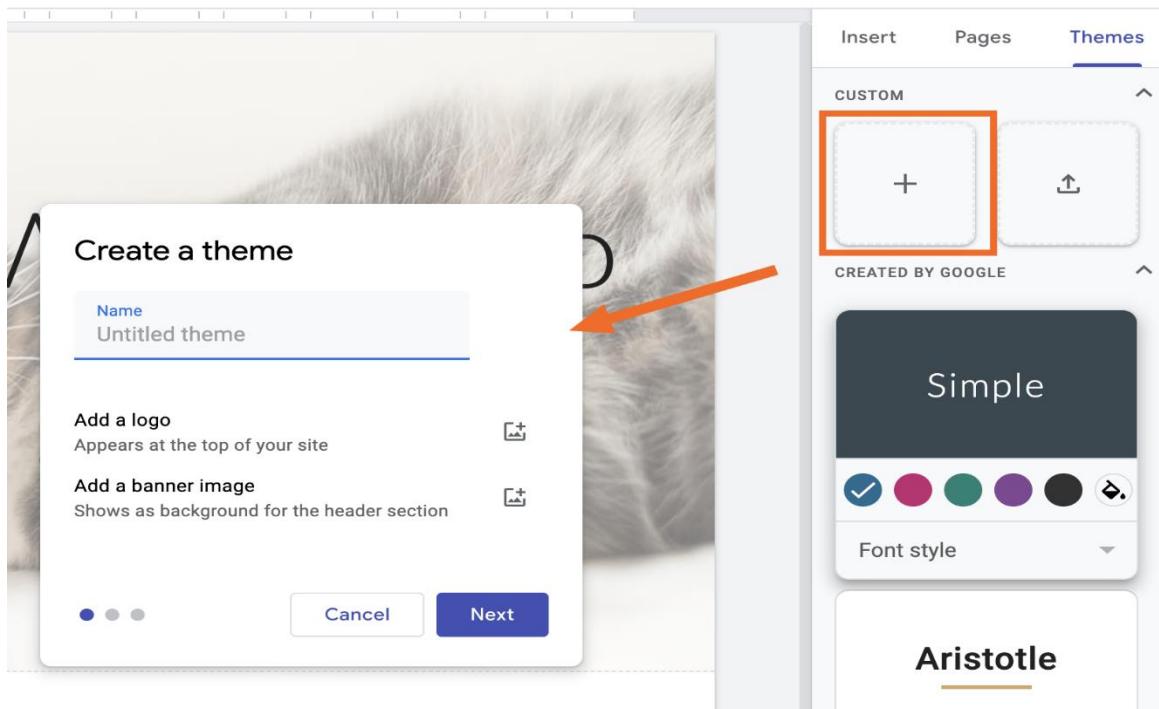
Add text and media

Time to dig into Google Sites' website-building elements. First, click **Insert** in the side menu to access all your options.

There are a lot of options, so let's review what each one lets you do.

Basic elements: The first set of elements lets you insert a text box and images, embed HTML code, or upload files from Google Drive. You can also access these options by double-clicking anywhere in the editor.

Content blocks: Drag and drop any of the prearranged blocks of text and images directly onto your page, which you can edit afterward.

Collapsible group: Add a header with collapsible text lines underneath. If you don't want the text group to collapse, click the toggle next to Collapsible in the editor.

Table of contents: Automatically turn your page's section headings into a table of contents with clickable links that jump to the appropriate section. You can't manually add other section headings to your table of contents, but you can hide them. Hover over any section header and click the Hide icon, which looks like an eye.

Image carousels: Upload two or more images to display in an image carousel

By default, users have to manually click through the photos. If you want the carousel to automatically cycle through, however, you can change this. Click the Settings icon in the Edit carousel window.

Click the checkbox next to Auto-start and select the Transition speed. Then click Update.

Buttons: Create a link button. By default, the button font and colors are automatically drawn from your design theme.

Dividers: This inserts a simple horizontal line onto your page. You can't edit the style in any way.

Spacer: This inserts a thick, rectangular block. To change the color, you have only three choices predetermined by your design theme—click the Section colors icon, which looks like a paint palette, next to the block. Then choose your style or upload an image.

Social links: Similar to button links, these allow you to add links to your social media pages. Let's say you add your Instagram profile. Google Sites will automatically use the Instagram logo as your button link. If you want to use a different design, click the Add image icon next to the Link field.
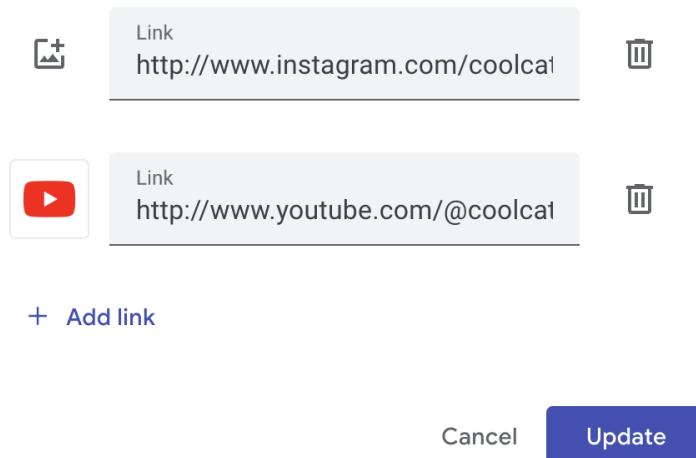
Incorporate Google Apps:

Use built-in Google tools such as Google Docs, Sheets, Slides, Forms, YouTube, and Google Drive to embed content directly into a Google Site. This helps in adding documents, presentations, videos, and forms for better interaction.

Arrange and Edit Columns and Sections:

Organize content by adding sections and adjusting column layouts. Resize columns, move sections, or change background styles to improve the visual structure and readability of the site.
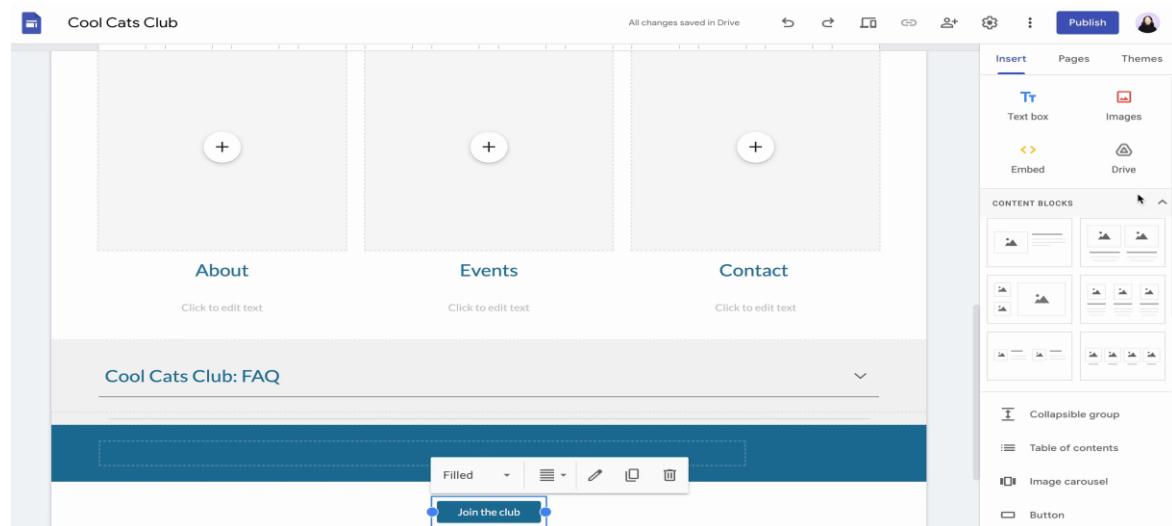

Add and Manage Pages:

Create new pages for different topics using the Pages tab. Rename, reorder, nest (sub-pages), or delete pages to maintain a clear navigation structure for the website.
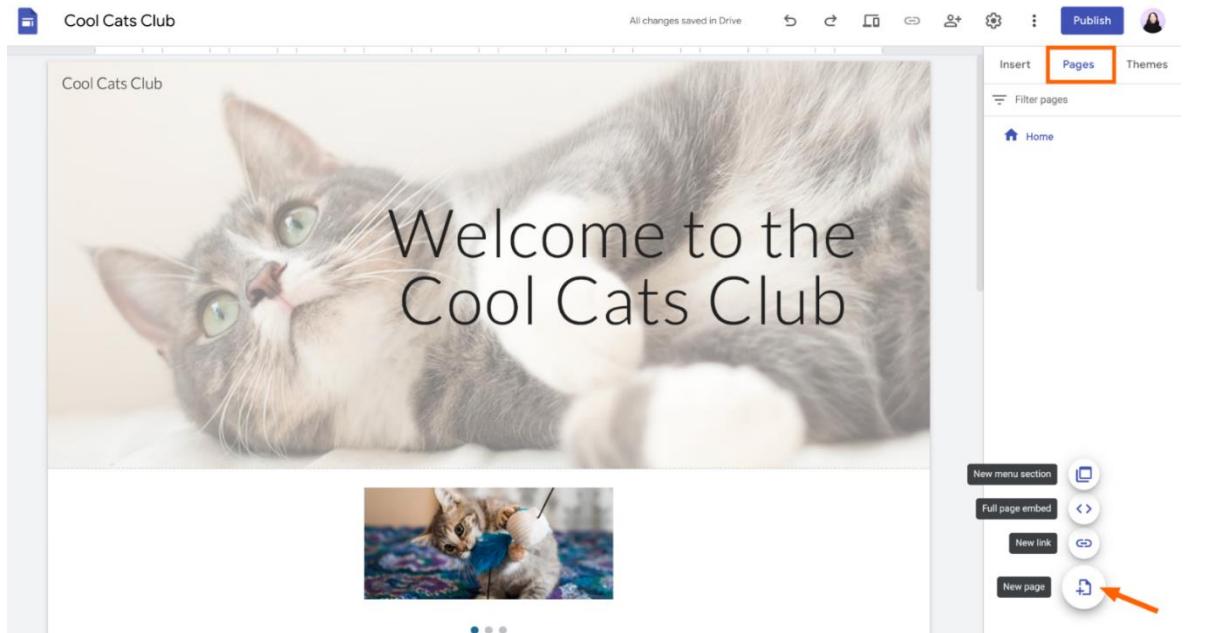
Publish and Collaborate:

Publish the site to make it accessible on the web. Share editing or viewing access with collaborators, allowing multiple users to work together while controlling permissions.
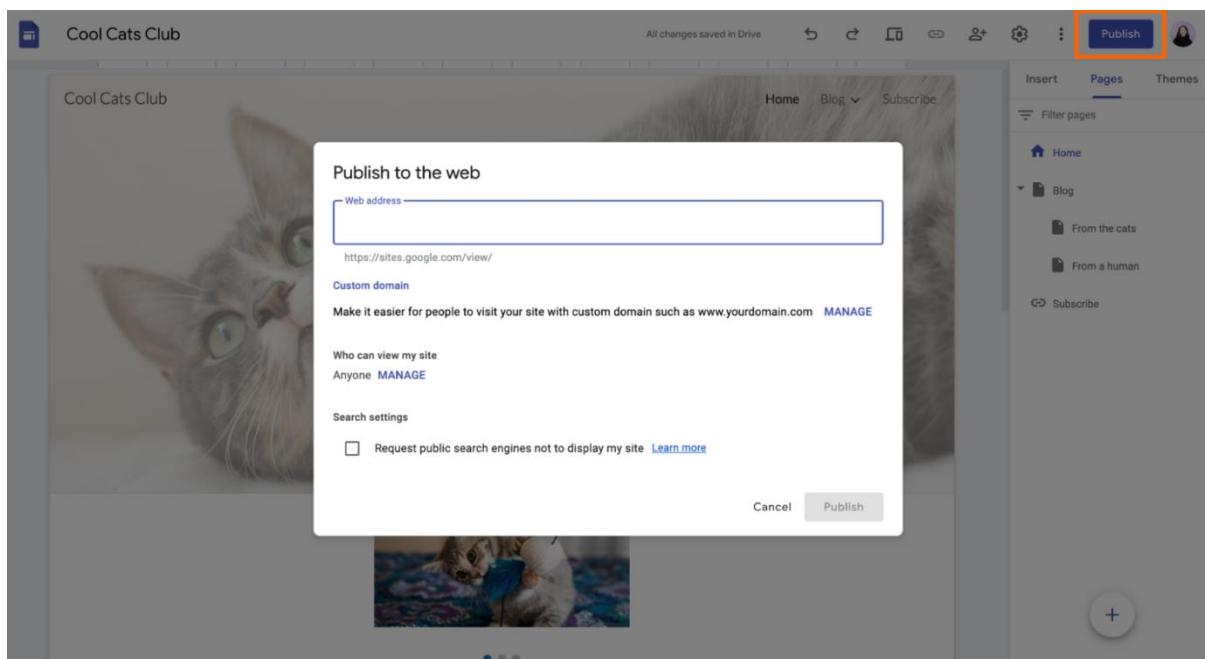
2. Google Docs

Steps to create a Google Doc with important features:

1. It can also be accessed from Gmail by clicking the **Google Apps (9-dot) menu** in the top-right corner and selecting **Docs**.
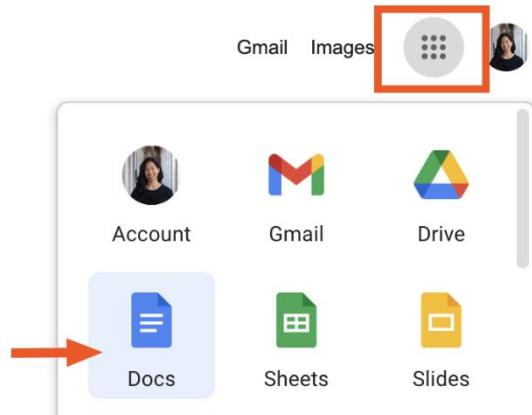
4. Select the Template gallery to view Google's assortment of premade templates.
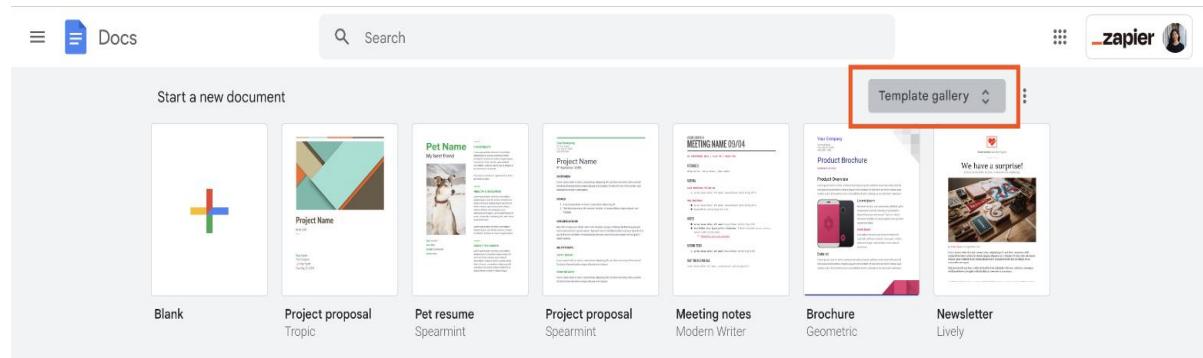
If you can't find the Template gallery, select the menu icon (≡) in the top-left corner of the Google Docs home screen > Settings > Display recent templates on home screens > OK.

5. Select a template from the gallery.
6. Edit the template as needed.
7. Without using any template, a Google Doc can also be created using a blank document.
8. Start typing your content; the document is saved automatically.
9. To share the document, click the **Share** button and add email addresses.
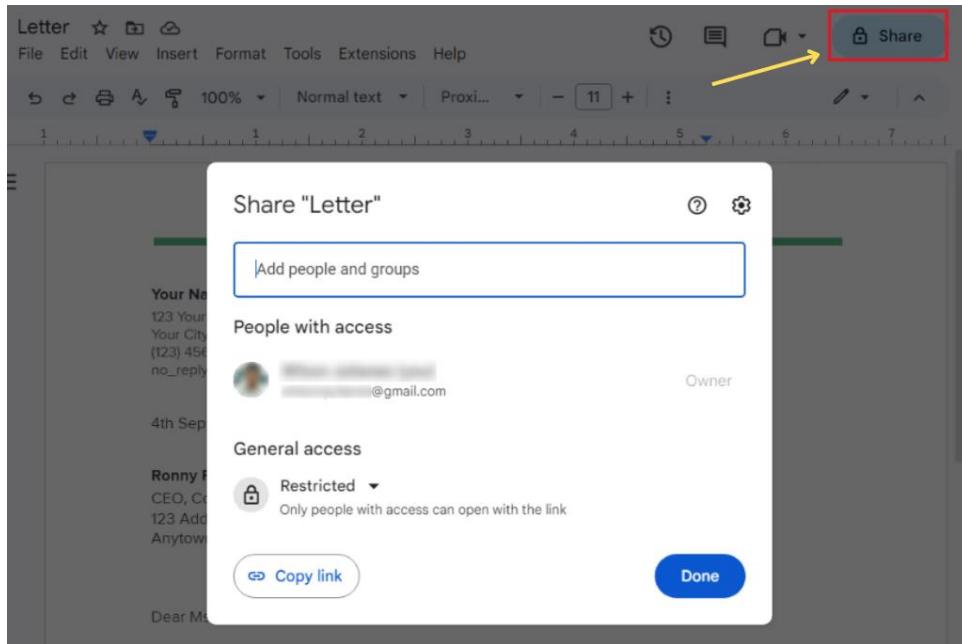
10. Click the **Share** button, add collaborators' emails, and assign **Viewer, Commenter, or Editor** access. Multiple users can **edit or comment in real time** while controlling who can modify the content.
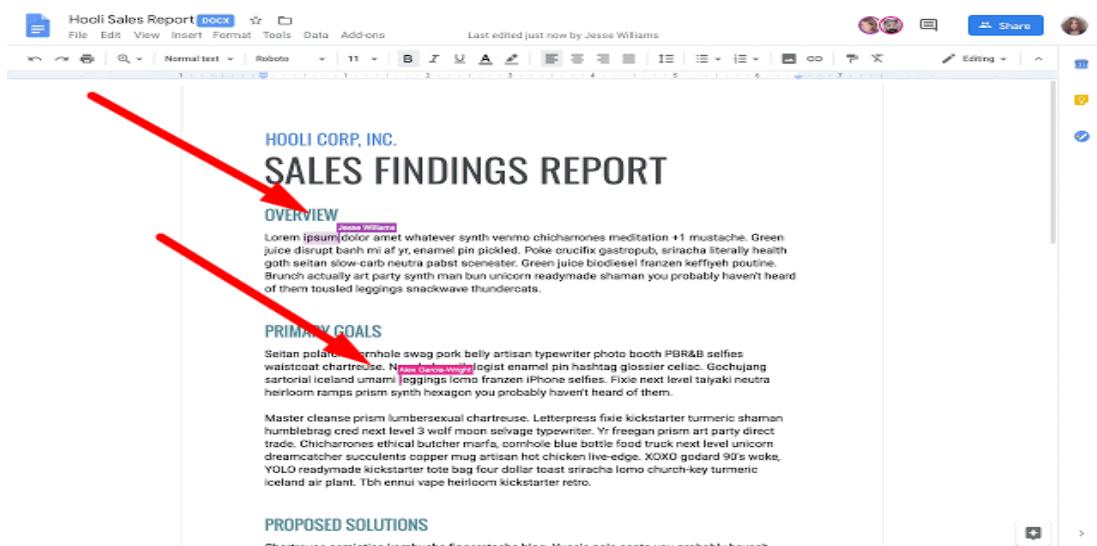
11. Use **Comments** or **Suggestion mode** for feedback and collaboration.
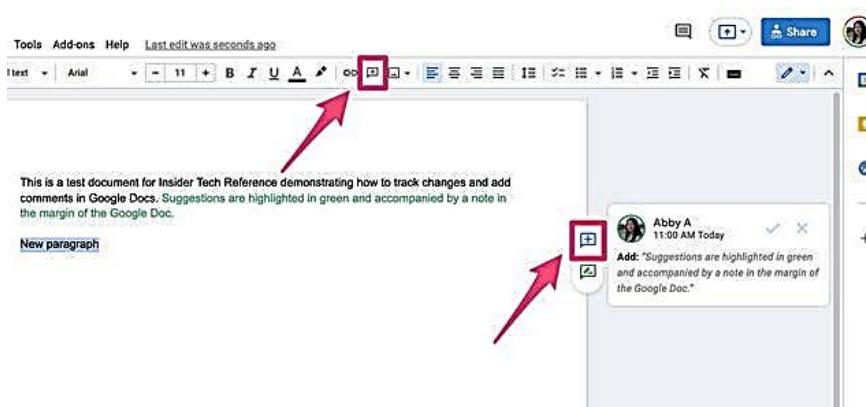
12. To check previous changes, go to **File → Version history**.
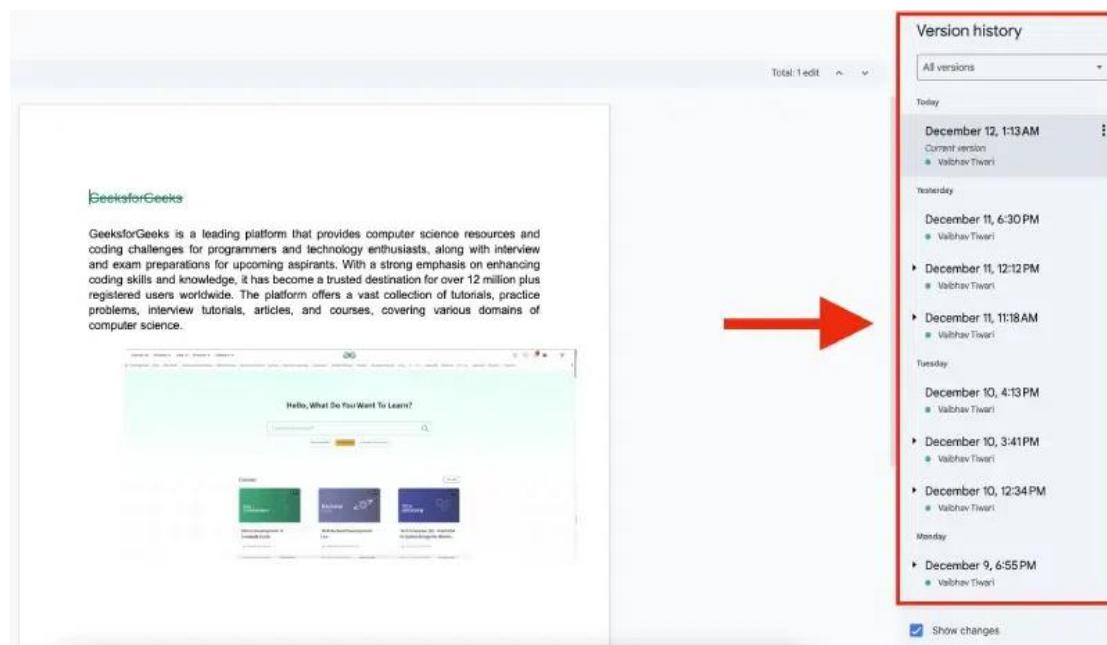
3. Google Forms

How to Access:

1. Go to forms.google.com or open Google Apps menu → Forms.
2. Click Blank to create a new form or choose a template.
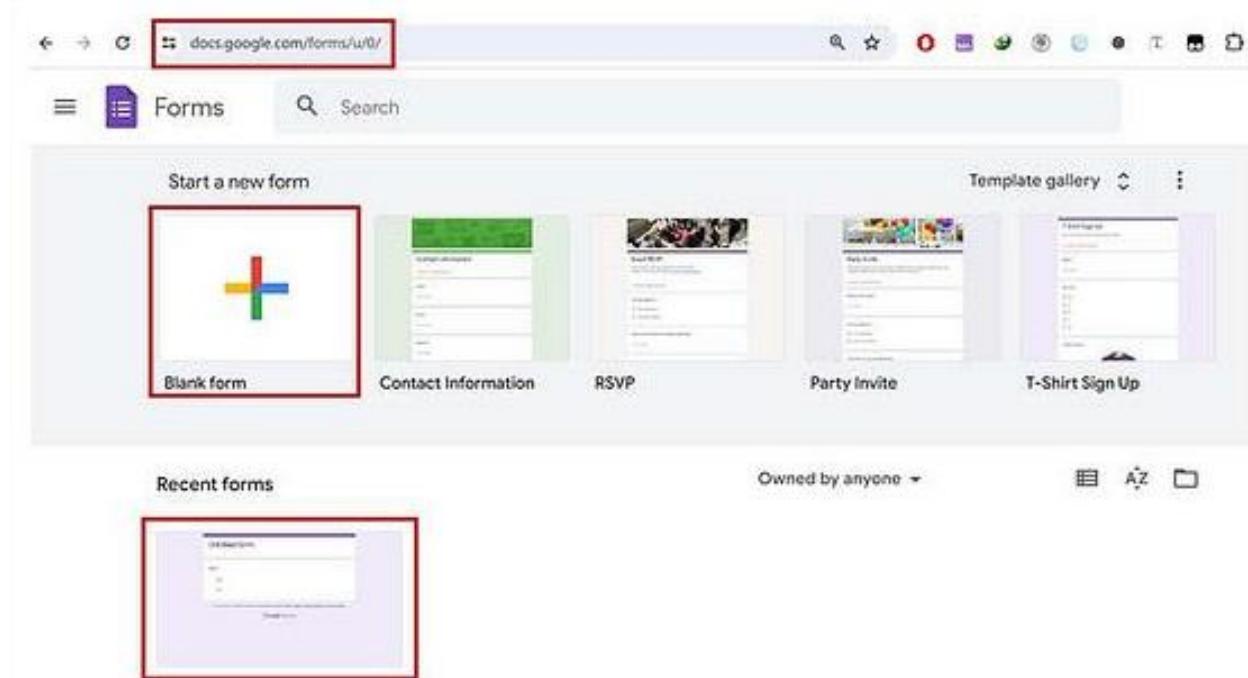
Important Features:

- Add **multiple question types**: short answer, multiple choice, checkboxes, dropdowns, etc.
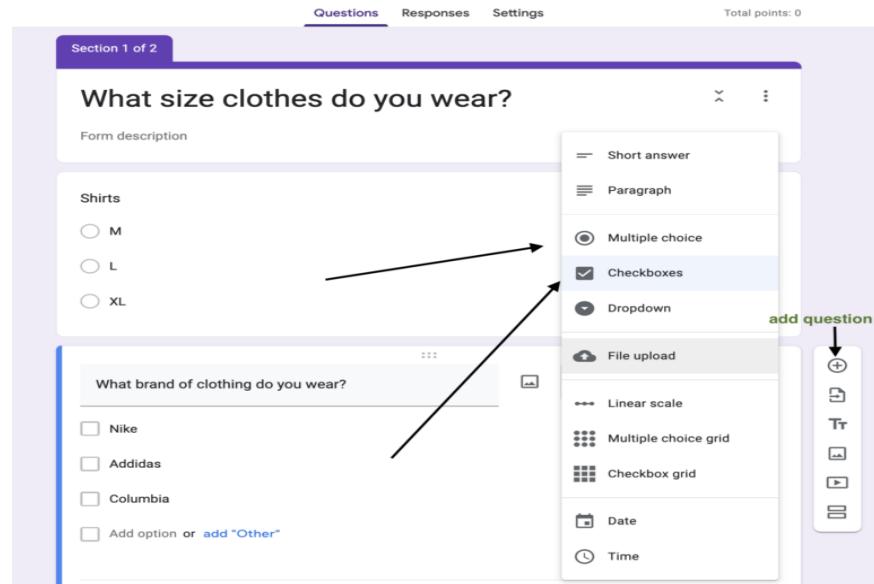
- To make a form a quiz, go to **Settings → Quiz → Make this a quiz**. To allow students to **see their marks immediately**, enable **"Immediately after submission"** under the quiz settings.
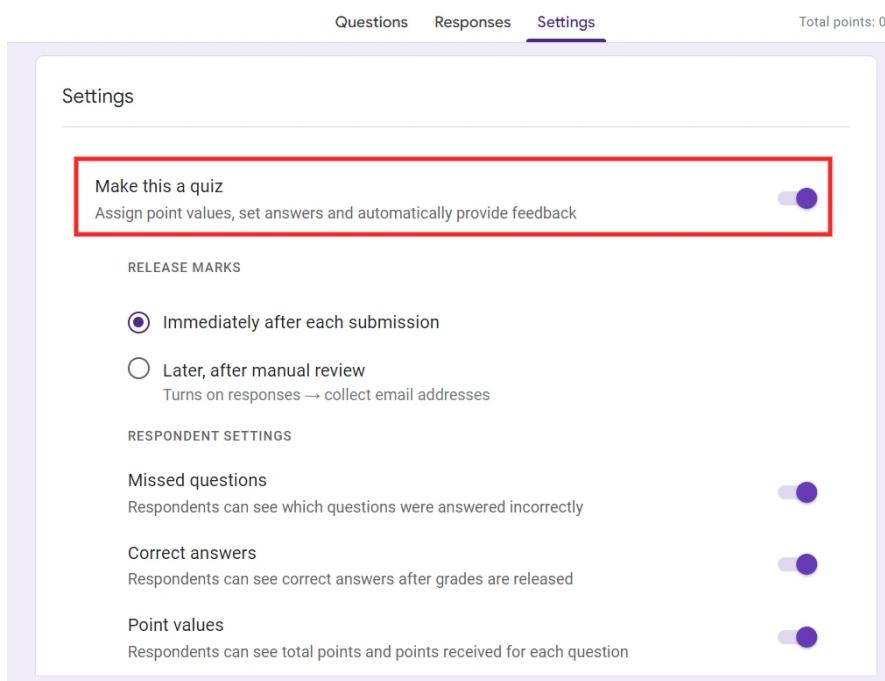

- Use the **"Add section"** option (two-rectangle icon) to divide a form into parts, helping organize questions by topic or page and making the form easier for respondents to complete.

- Enable **Quiz** in **Settings** to assign correct answers and points for automatic checking, and turn on **Required** to make the question compulsory.

- When sharing a Google Form, **turn on responses** to allow others to submit answers, and you can **automatically record all responses** in Google Sheets for easy tracking and analysis.

- Forms can be shared via link, email, or embedded in websites, with settings to make questions mandatory, limit responses, or allow collaborators to edit, and multiple users can create and edit the form together in real time**.**

ACTIVITIES:

1. Google Sites (Website Creation)

- Create a Google Site on **any topic of your interest** (e.g., Science, Sports, Travel, Technology, Education).
- Give an appropriate **title** to your website.
- Create at least **three pages** related to your topic.
- Add **text, images, and links** on different pages.
- Arrange content properly using layouts and sections.
- Publish the website,

2. Google Docs (Collaborative Work)

- Create a Google Doc and write a formal letter to the Principal requesting permission to organize a school event.
- Share the document with two or more classmates.
- Work on the same document collaboratively.
- Use Comments to suggest improvements or ask questions.
- Use Reply and Resolve options in comments.

3. Google Forms (Questionnaire Creation)

- Create a Google Form to design a questionnaire on any topic.
- Use different question types (short answer, multiple choice, checkboxes, and dropdown).
- Add a new section and organize questions properly.
- Insert images or videos in questions where suitable.
- Turn the form into a Quiz, set correct answers, and assign marks.
- Make all questions required.
- Apply a theme and header image to the form.
- Submit responses and view the Responses summary.

RESULTS:

1. Write the Google sites **website link** that you have created in the box given below, and attach the **QR code image** of the website.

QR CODE TO BE PASTED HERE

Website link:

2. Attach a printout of the letter created on google Docs. showing real-time collaborative editing with different contributors' names/cursors or text visible, including comments made by collaborators.

3. Write the sharing link and paste the QR code of the Google form you have created.

Form link:

QR CODE TO BE PASTED HERE