Sidhu Vidya Samsthe's

NALANDA FOUNDATIONS POLYTECHNIC

AICTE Approved | Govt. Recognized | DCTE Affiliated

IT SKILLS – 25CS011

I year common subject

Manual

Department of Computer Science and Engineering

Vidyaranya Campus, Gadag Road, Bhandiwad, Hubballi – 580 023.

SL. NO.	NAME	SIGN
1.	Identify parts of a computer system.	
2.	Identify OS and hardware specifications.	
3.	Perform basic file/folder operations (GUI).	
4.	Install application software (browser, Scratch).	
5.	List 3 real-world examples of LAN, MAN, WAN.	
6.	Find MAC and IP addresses.	
7.	Create email account and explore security settings.	
8.	Search effectively: "How does a Search Engine work?"	
9.	Design and host personal website (e.g., Wix, WordPress).	
10.	Test internet speed.	
11.	Identify different cyber threats (real-world).	
12.	Install and run antivirus scan.	
13.	Create strong passwords using a password manager.	
14.	Enable/test Multi-Factor Authentication (MFA).	
15.	To configure User Access Control (UAC) settings in Windows	
16.	Identify safe/unsafe websites.	
17.	Encrypt/decrypt files using OS tools.	
18.	Perform basic data backup.	
19.	Spot fake websites and phishing emails.	
20.	Analyze real vs. fake websites.	
21.	Identifying Phishing emails.	
22.	Check OS updates; remove unnecessary apps.	
23.	Recognize scam calls/messages.	
24.	Explore block coding tool interface.	
25.	Develop algorithms and flowcharts for: a. Arithmetic operations b. Metric conversions	
26.	Create animated sequence (Scratch).	
27.	Design flowchart for real-world task.	
28.	Score counter for simple game.	
29.	Interactive greeting app with user input.	
30.	Algorithm and flowchart with conditions (e.g., compare numbers).	
31.	Interactive story with yes/no choices.	
32.	Traffic light simulator using conditions.	
33.	Bouncing ball animation using loops.	
34.	Counting program (1 to 20 using loops).	
35.	Create a free cloud account (AWS, Azure, GCP).	
36.	Explore cloud console and services.	
37.	Upload/download files via cloud storage.	

38.	Create online forms/surveys using cloud services.	
39.	Blink LED using visual block code and Arduino.	
40.	Create 3-LED traffic signal controller (Arduino).	
41.	Explore AI tools (ChatGPT, Gemini, Grok, Copilot, Sora, etc.)	
42.	Test basic prompts and analyze responses.	
43.	Research and report on popular IT certifications.	

Experiment 1

Computer System

Aim:

To identify and understand the various internal and external components of a computer system and their functions.

Theory:

A computer system consists of both hardware and software components. In this experiment, we focus on identifying and understanding hardware parts.

External Components:

1. Monitor – Displays output from the computer.
2. Keyboard – Used to input text and commands.
3. Mouse – A pointing device used to interact with the GUI.
4. Cabinet (CPU Case) – Encloses all internal components.
5. Printer / Speakers / Webcam – Optional output/input devices.

Internal Components:

1. Motherboard – The main circuit board connecting all components.

2. CPU (Processor) – Executes instructions and processes data.
3. RAM (Memory) – Temporary memory for fast data access.
4. Power Supply (SMPS) – Converts AC to DC for component use.
5. Hard Disk Drive (HDD) / SSD – Stores data and software permanently.
6. Cooling Fan / Heat Sink – Keeps the processor from overheating.

Result:

The internal and external components of a computer system were successfully identified. Their functions were noted, aiding in understanding how a computer works as an integrated system.

Experiment 2

Operating System and Hardware Specifications of a Computer System

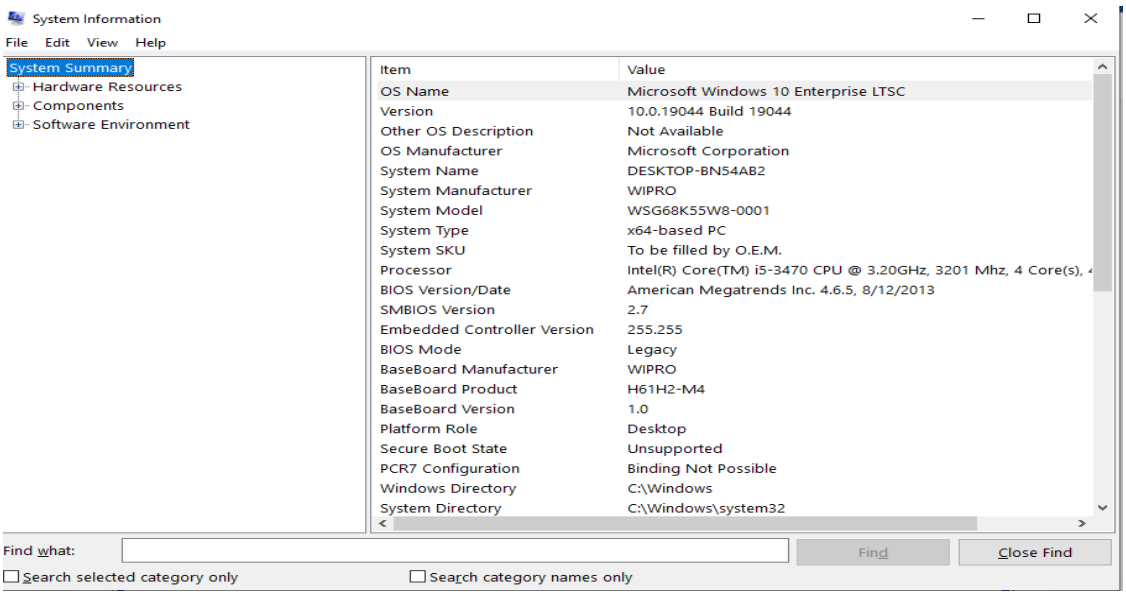
Aim:

To identify and document the operating system details and hardware configuration (processor, RAM, storage, system architecture) of a computer using built-in system tools.

Steps:

To understand the system configuration, we collect information from system settings or commands.

1. Press Windows + R, type msinfo32, and press Enter to open System Information.
2. Open Settings → System → About to view:
    a. OS edition and version
    b. Processor type
    c. Installed RAM
    d. System type (32-bit or 64-bit)



Result:

The operating system version, processor details, system architecture, RAM size, and storage specifications of the system were successfully identified using system tools. This information is crucial for software compatibility, upgrades, and troubleshooting.

Experiment 3

Basic File and Folder Operations Using Graphical User Interface (GUI)

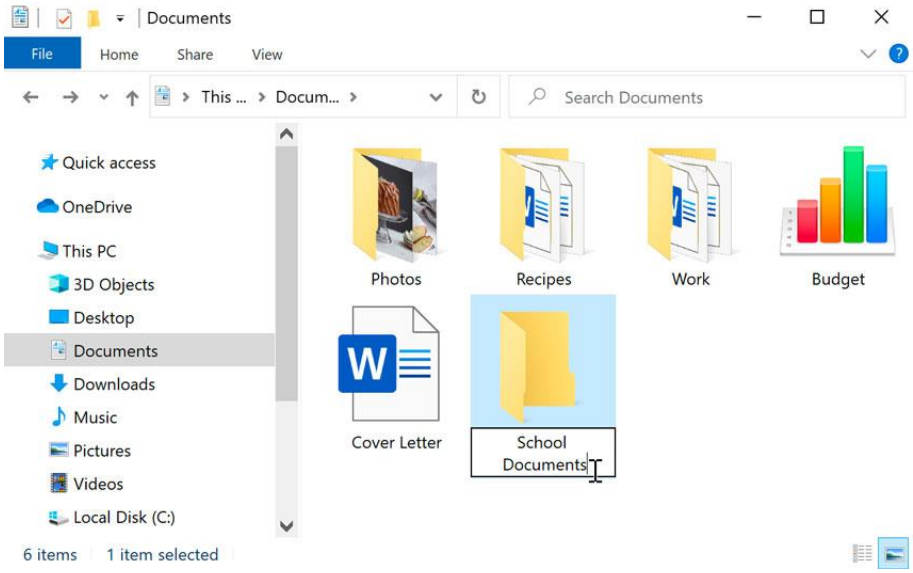
Aim:

To perform basic file and folder operations such as creating, renaming, copying, moving, and deleting files and folders using the GUI-based file explorer in Windows or Linux.

Steps:

Most operating systems offer a Graphical User Interface (GUI) file manager to manage files and folders easily using point-and-click operations.

1. Open File Explorer (Windows + E).
2. Create a Folder:
    a. Right-click → New → Folder → Name it "LabPractice".
3. Create a File:
    a. Inside the folder, right-click → New → Text Document → Name it "testfile.txt".
4. Rename:
    a. Right-click the folder/file → Rename → Enter new name.
5. Copy and Paste:
    a. Right-click the file → Copy → Paste it into another location.
6. Move:
    a. Drag and drop the file to another folder.
7. Delete:
    a. Right-click → Delete → File goes to Recycle Bin.



Result:

Basic file and folder operations such as create, rename, copy, move, and delete were successfully performed using the GUI. These operations are essential for effective file management in everyday computer use.

Experiment 4

Installation of Application Software

Aim:

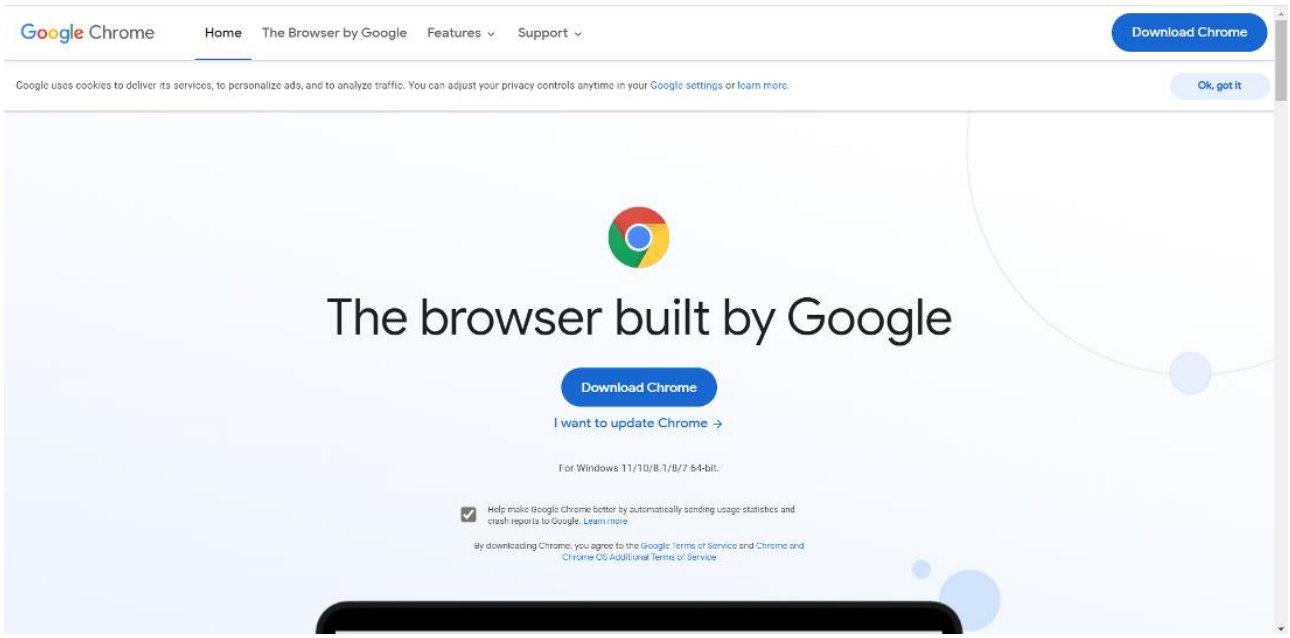
To install and verify the proper functioning of application software such as a web browser (e.g., Google Chrome) and Scratch (block-based coding tool) using GUI-based installation procedures.

Steps:

Application software performs specific user tasks like web browsing, coding. Installing them enhances system functionality.

Installing a Web Browser (Google Chrome)

1. Open an existing browser (Edge).
2. Visit: https://www.google.com/chrome
3. Download the appropriate installer (.exe for Windows / .deb for Ubuntu).
4. Run the installer and follow on-screen instructions.
5. After installation, launch Chrome and verify functionality.

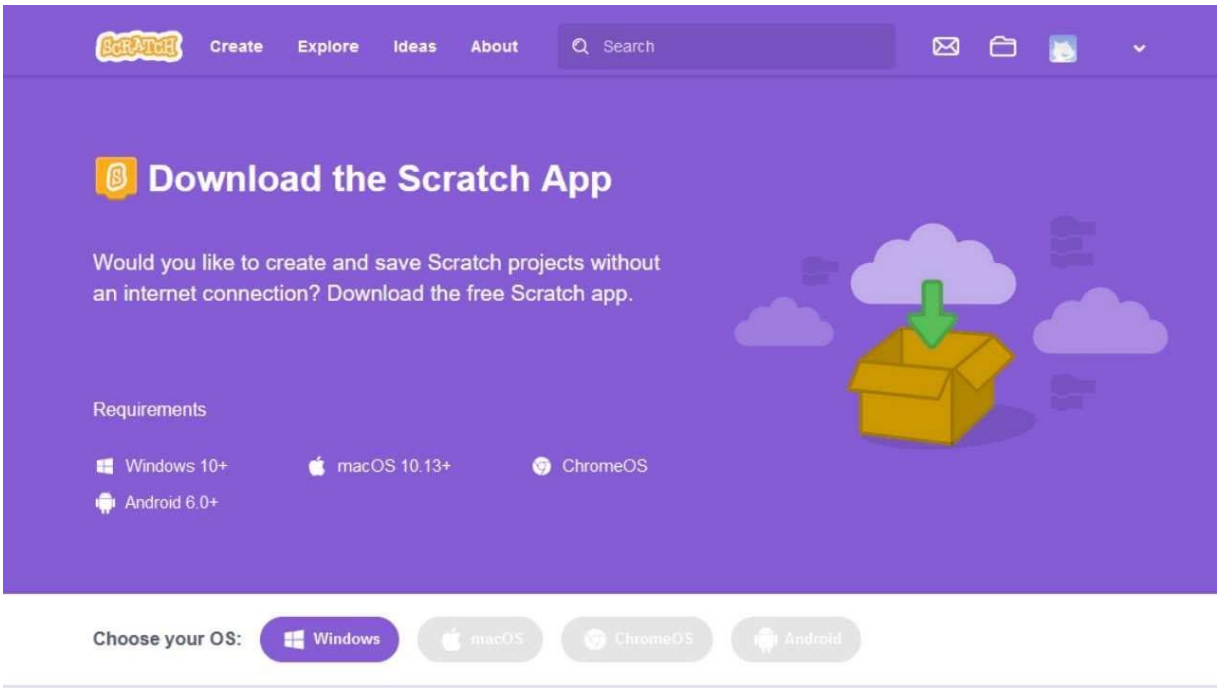


Installing Scratch

1. Visit https://scratch.mit.edu/download

2. Download the offline editor for Windows/Linux.
3. Run the setup file and install Scratch.
4. Launch Scratch and test by creating a simple project.

Note: Alternatively, you can use the online version at https://scratch.mit.edu



Result:

Google Chrome and Scratch were successfully downloaded, installed, and launched. This confirms the system's readiness to support basic productivity and programming tasks. Students can now explore online content and learn block-based coding.

Experiment 5

Three Real-World Examples of LAN, MAN, and WAN Networks

Aim:

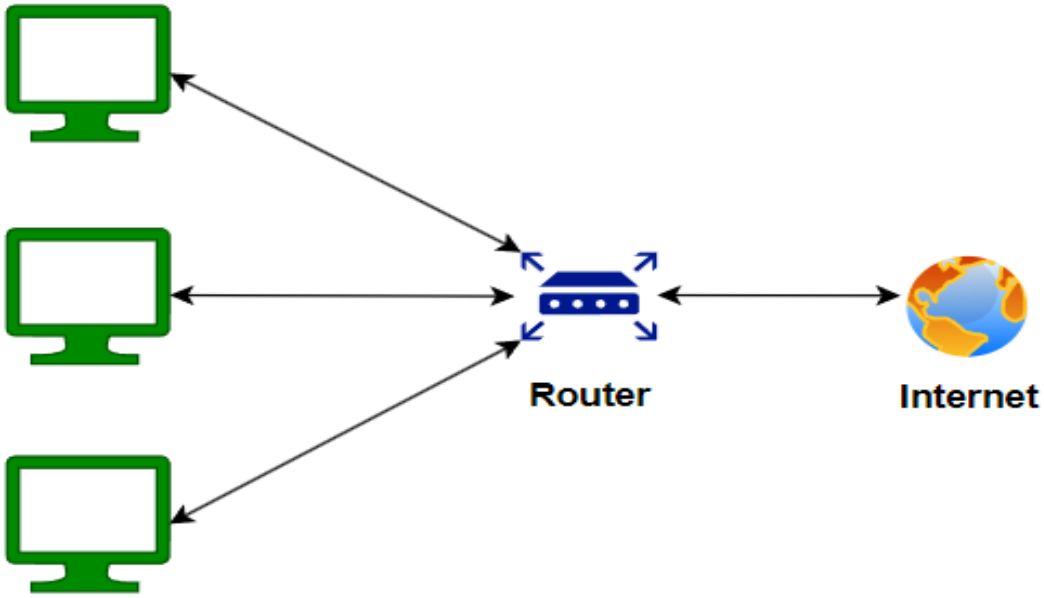
To understand the classification of networks based on geographical area and to list three real-world examples for each: LAN (Local Area Network), MAN (Metropolitan Area Network), and WAN (Wide Area Network).

Theory:

Computer networks are categorized by their coverage area. Each type has unique characteristics and

Use cases:

🔹 1. LAN – Local Area Network

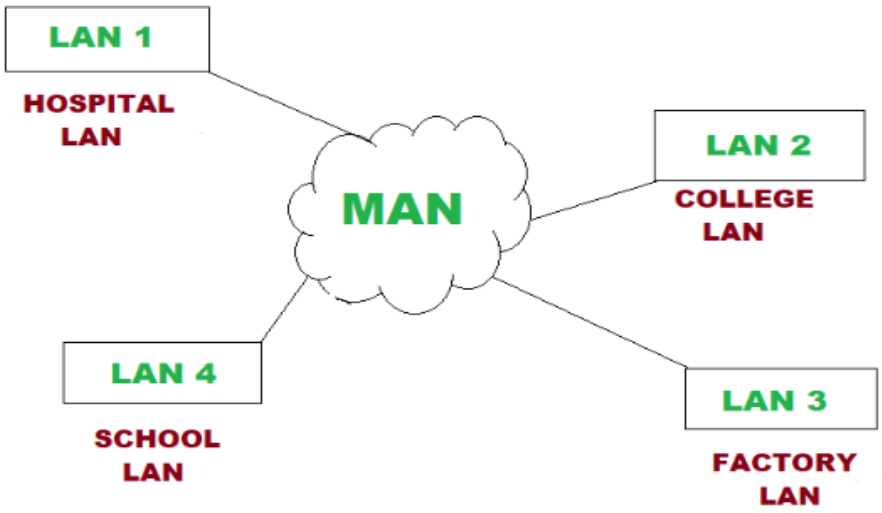


- Covers a small area like a room, home, or building
- High speed, low latency
- Privately owned

Real-World Examples:

Location	Description
College Computer Lab	All lab systems connected via Ethernet
Home Wi-Fi Network	Connects phones, TV, laptop, etc.
Office LAN	Connects employees' computers in a room

2. MAN – Metropolitan Area Network

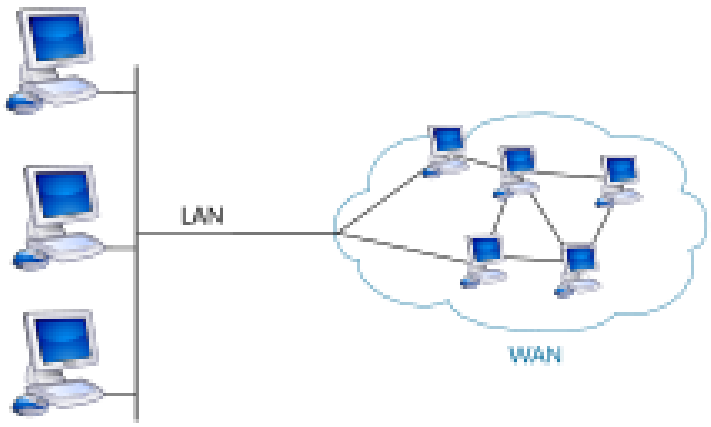


- Covers an entire city or campus
- Higher range than LAN, often owned by ISPs

Real-World Examples:

Location	Description
Campus Network (e.g., VTU)	University buildings connected across a city
Cable TV Networks	Broadband TV via fiber in urban regions
Smart Traffic Control Systems	IoT devices across a city's intersections

🔻 3. WAN – Wide Area Network



- Spans large geographical areas
- Connects LANs across cities, countries

Real-World Examples:

Network	Description
The Internet	Largest WAN; global communication network
Multinational Company VPN	Connects branch offices in different countries
Banking Network	ATM and branches across regions connected together

Result:
Three real-world examples were successfully identified for each network type (LAN, MAN, WAN), helping differentiate between them based on scale and application.

Experiment 6

MAC and IP Address of a Computer

Aim:

To find and record the MAC (Media Access Control) and IP (Internet Protocol) addresses of a computer using system commands or GUI tools.

Steps:

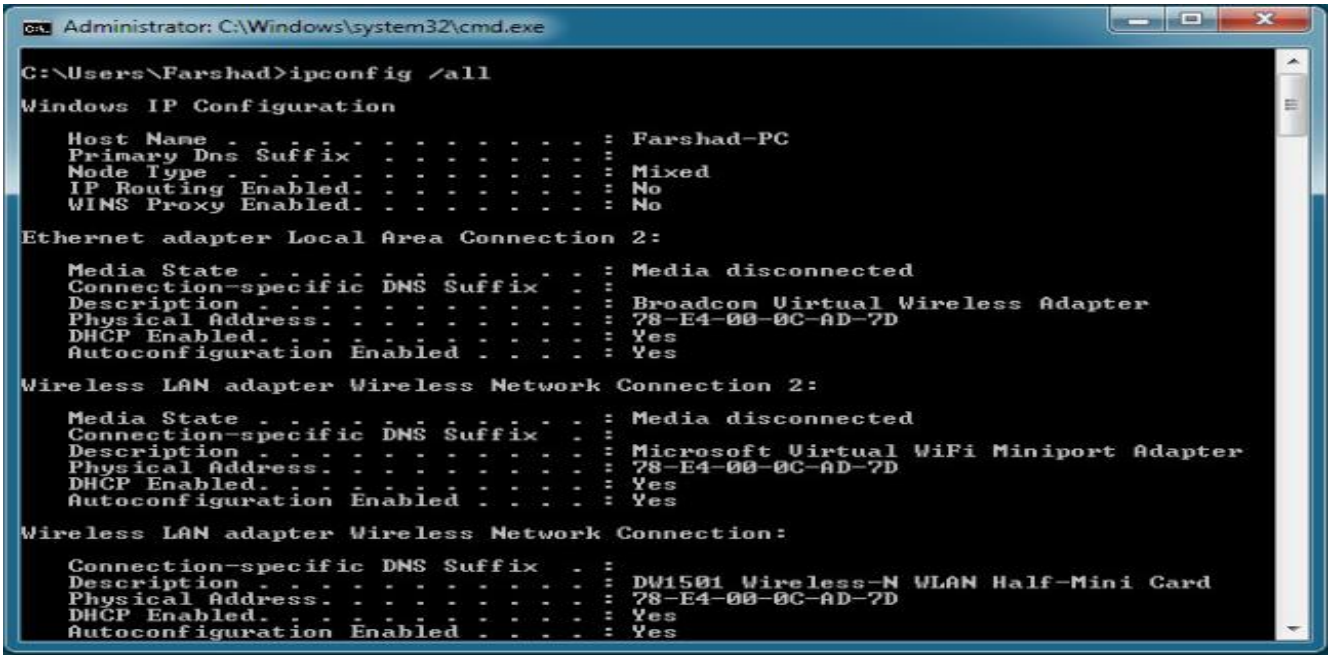
✓*MAC Address:*

- A unique hardware identifier for the network adapter.
- It's permanent and assigned by the manufacturer.
- Format: 00-1A-2B-3C-4D-5E

✓*IP Address:*

- Logical address assigned to a device on a network.
- Two types: IPv4 (e.g., 192.168.0.100) and IPv6

1. Press Windows + R → Type cmd → Press Enter.
2. In Command Prompt, type: ipconfig /all → Press Enter.
3. Locate:
    a. Physical Address = MAC Address
    b. IPv4 Address = IP Address



Result:

The system's MAC and IP addresses were successfully identified using command-line tools. MAC helps identify the device physically, while IP allows communication over networks like LAN and the internet.

Experiment 7

Email Account

Aim:

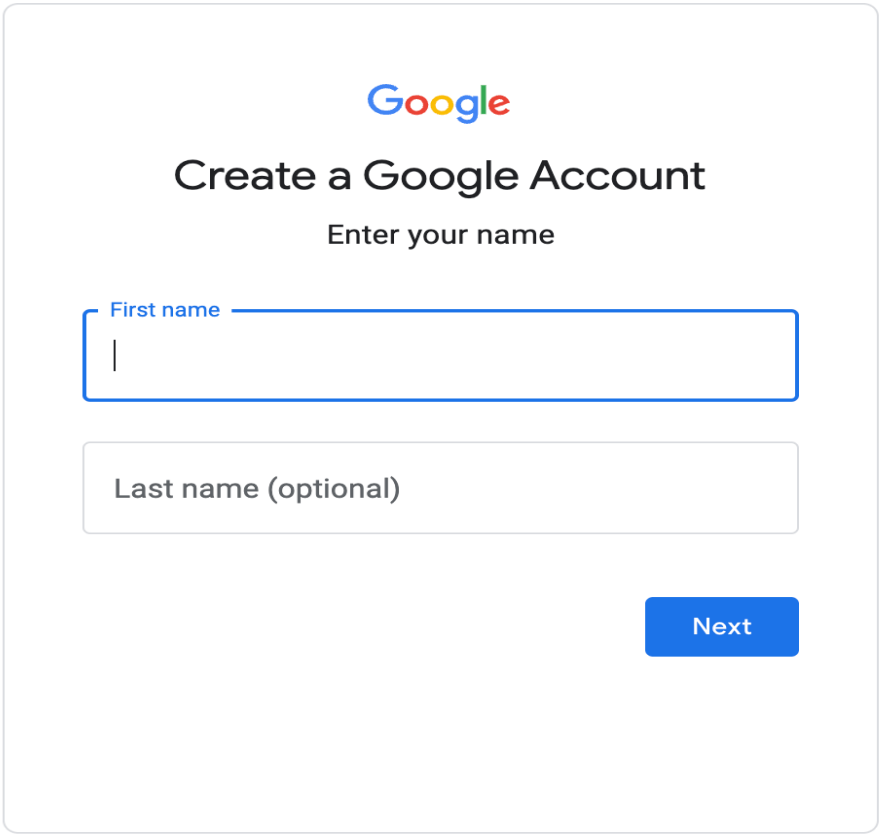
To create a new email account using a web-based service (e.g., Gmail or Outlook) and explore key security settings such as password strength, recovery options, and two-factor authentication.

Steps:

Email is an essential communication tool. For security, it's important to configure settings that protect the account from unauthorized access.

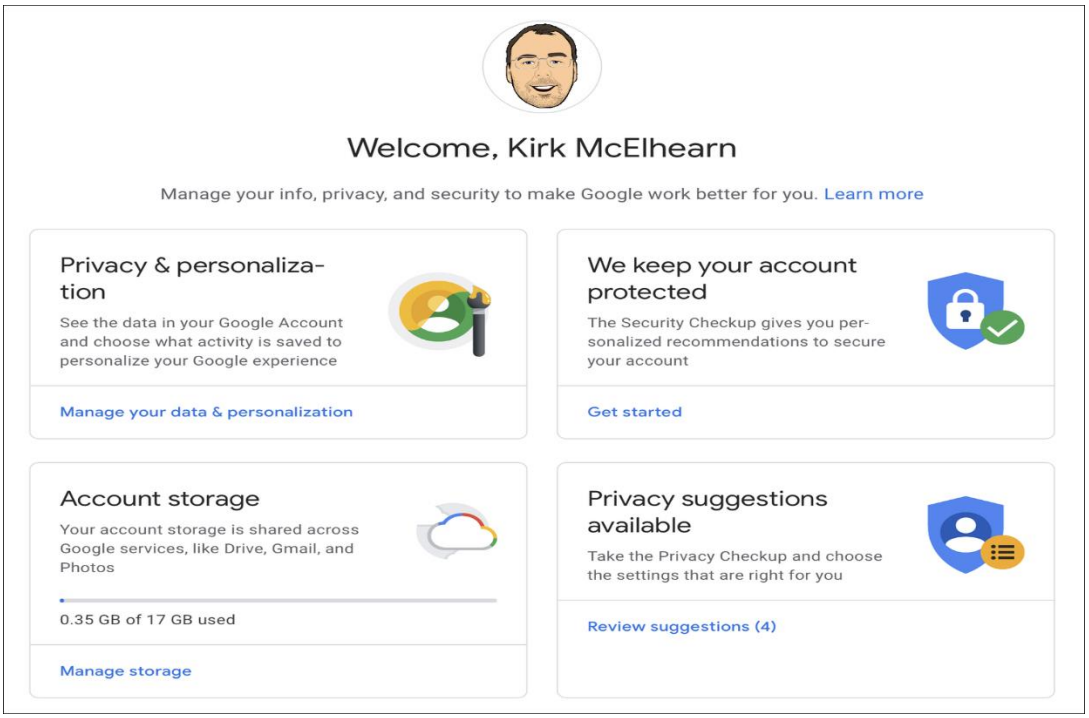
Create a New Email Account

1. Go to https://www.gmail.com or https://www.outlook.com
2. Click on "Create account".
3. Fill in:
    a. First and last name
    b. Desired email ID and password
    c. Phone number (for verification)
    d. Recovery email and date of birth
4. Verify using OTP (sent to phone number).
5. Once done, sign in to the email dashboard.



Explore Security Settings

1. Go to Google Account Settings → https://myaccount.google.com/security
2. Enable 2-Step Verification (MFA)
3. Set up Recovery phone/email
4. Review recent device activity
5. Ensure the password is strong (mix of letters, numbers, symbols)
6. Enable alerts for suspicious logins or device access



Result:

A new email account was successfully created. Security settings such as password recovery, multi-factor authentication, and recent activity alerts were explored and configured to enhance account protection.

Experiment 8

Search Engine

Aim:

To understand how a search engine works and to practice effective searching techniques using keywords, filters, and search operators.

Theory:

A search engine (like Google, Bing, DuckDuckGo) is a software system that helps users find information on the internet.

🔻 How a Search Engine Works (Basic Process):

1. Crawling:
    a. Automated programs (bots/spiders) visit web pages and gather content.
2. Indexing:
    a. Collected data is organized in large databases (index).
    b. Content is sorted by keywords, topics, and relevance.
3. Ranking:
    a. When a user searches, the engine uses algorithms to rank and display the most relevant results based on:
        i. Keywords
        ii. Page authority
        iii. Freshness
        iv. User location/preferences

Effective Search Techniques:

Search Technique	Example	Result
Quotation marks	"climate change report"	Finds exact phrase
Site filter	site:gov.in climate policy	Searches only government sites
File type search	internet safety filetype:pdf	Finds only PDF documents
Minus operator	apple -fruit	Removes fruit results, focuses on tech
Wildcard	how to * a computer	Fills in the blank (e.g., repair, build)

Result:
The working of a search engine (crawl, index, rank) was understood. Effective searching techniques using quotes, filters, and operators were practiced. This improves accuracy and efficiency while searching online.

Experiment 9

Personal Website Hosting

Aim:

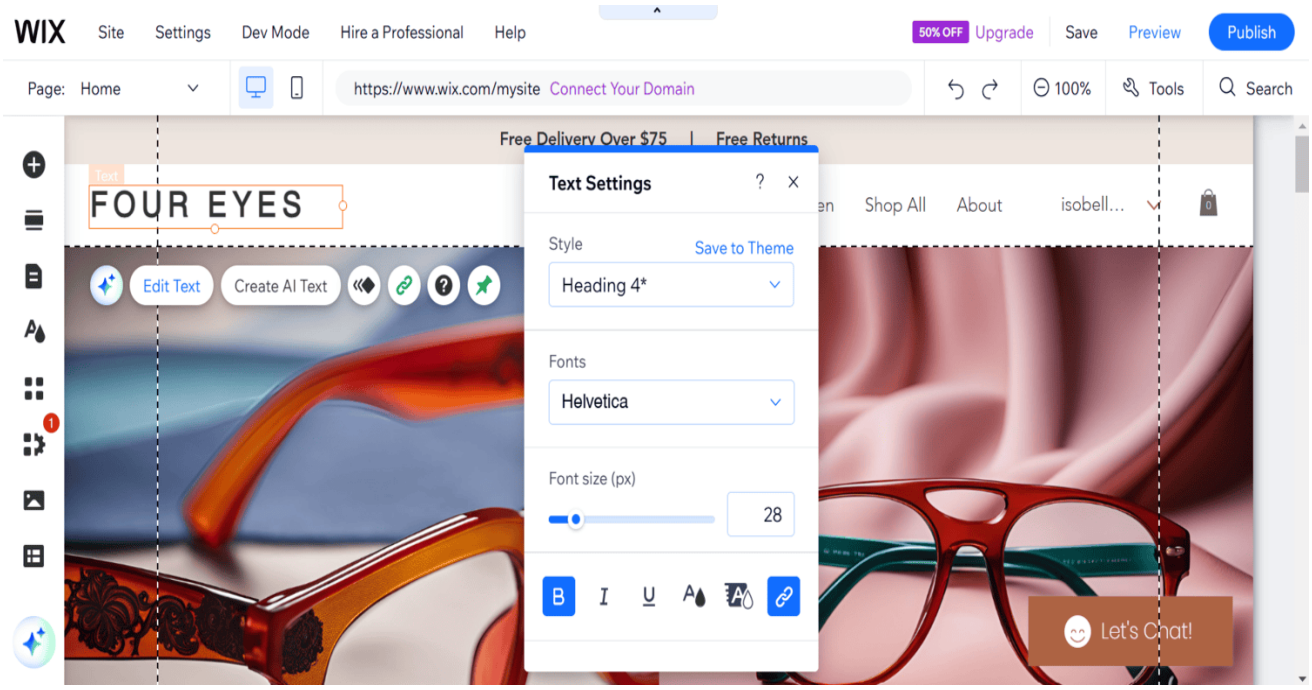
To create and publish a personal website using a free web-hosting platform such as Wix or WordPress, including basic elements like text, images, and navigation.

About Website Hosting:

Website hosting allows users to make their websites accessible via the internet. Free platforms like Wix.com and WordPress.com offer drag-and-drop tools to design websites without coding.

Steps to Create a Website (using Wix):

1. Go to https://www.wix.com
2. Create a free account using email or Google login.
3. Click on "Create New Site" → Choose category: e.g., "Personal", "Portfolio"
4. Select Wix Editor or Wix ADI (automated design).
5. Choose a template or start from scratch.
6. Add site elements:
    a. Header, About Me section
    b. Text, images, contact form
    c. Social media links
7. Click Publish and get a free Wix subdomain link (e.g., https://username.wixsite.com/mysite)



Result:

A personal website was successfully designed and hosted using a free platform (Wix/WordPress). The site includes a homepage, bio, images, and links, demonstrating an understanding of web design and hosting fundamentals.

Experiment 10

Internet Speed

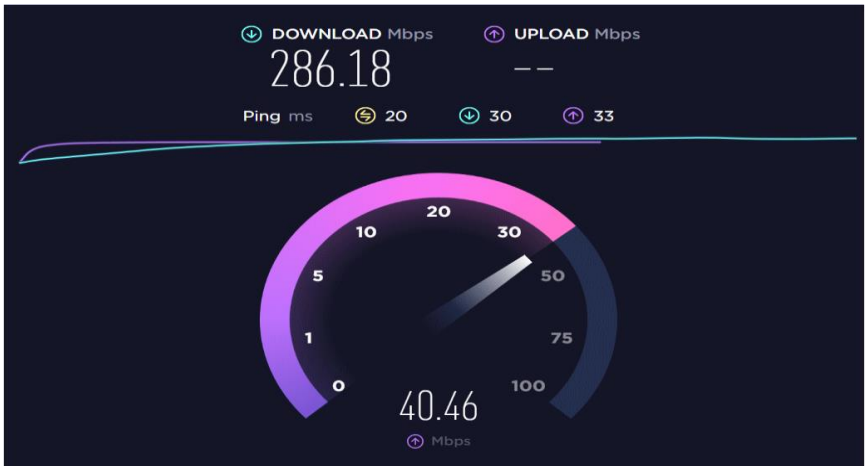
Aim:

To test the internet connection speed and understand the meaning of key terms such as download speed, upload speed, ping, and latency using an online speed test tool.

Steps to Perform the Test:

- Download Speed: Speed at which data is received (measured in Mbps).
- Upload Speed: Speed at which data is sent from your device.
- Ping/Latency: Time taken for a data packet to travel to the server and back (measured in milliseconds - ms).

1. Open a browser and visit:
    a. https://www.speedtest.net or
    b. https://fast.com (by Netflix)
2. Click the "Go" or "Start" button.
3. Wait for the test to complete. It will display:
    a. Download Speed
    b. Upload Speed
    c. Ping
4. Note the values and compare with your ISP plan.



Result Table:

Parameter	Value
Download Speed	
Upload Speed	
Ping	
ISP Plan	

Result:

The internet speed was successfully tested using Speedtest.net. The values obtained matched the expected ISP plan performance. Understanding these parameters helps troubleshoot network issues and optimize internet usage.

Experiment 11

Cyber Threats

Aim:

To study and identify various types of cyber threats using real-world examples, and understand how they affect individuals, organizations, and governments.

Theory:

Cyber threats are attacks launched through digital means to disrupt, damage, or steal data from systems and networks.

Here are five major types of cyber threats with real-world examples:

1. Phishing

Definition: Fraudulent emails or messages that trick users into giving personal information.

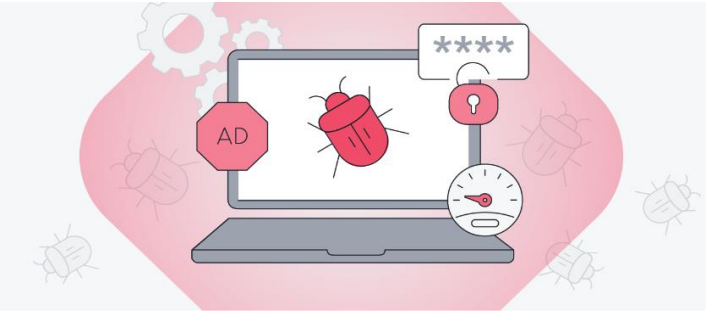
Real Example: In 2020, hackers posed as WHO officials during the COVID-19 outbreak, sending fake emails to steal login data.

2. Ransomware

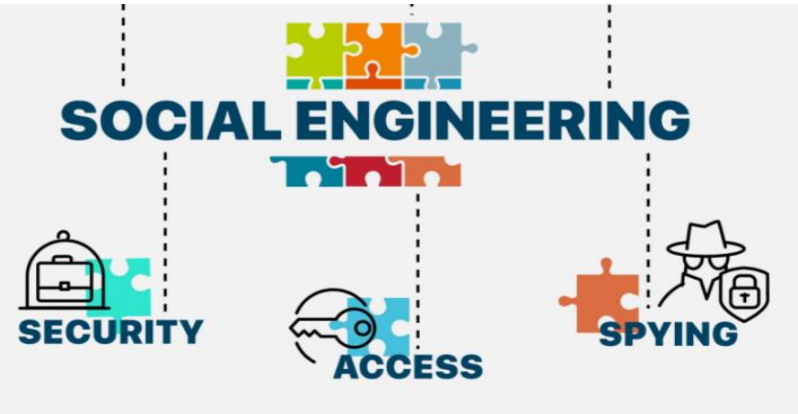
Definition: Malware that locks data and demands money for access.

Real Example: WannaCry Attack (2017) – Affected over 200,000 computers in 150 countries, including hospitals in the UK.

3. Malware (Viruses, Trojans)
Definition: Software that harms or spies on your computer.
Real Example: ILOVEYOU Virus (2000) – Spread via email attachment and infected millions of systems globally.



4. Social Engineering
Definition: Manipulating people into revealing confidential information.
Real Example: A fake IT support caller tricked an employee at Twitter in 2020, leading to a major account hack.



5. Keylogging
Definition: Records what users type to steal passwords and sensitive data.
Real Example: Hackers used keyloggers in cyber cafés to steal online banking details in multiple Indian cities.

Result:
Various cyber threats were identified and understood with real-world examples. This experiment emphasizes the need for cybersecurity awareness, strong passwords, and using antivirus/firewall tools to protect personal and organizational data.

Experiment 12

Antivirus Installation

Aim:

To install antivirus software, perform a system scan, detect any malware or viruses, and remove them to protect the system.

Steps:

Antivirus software helps protect a computer from malware such as viruses.

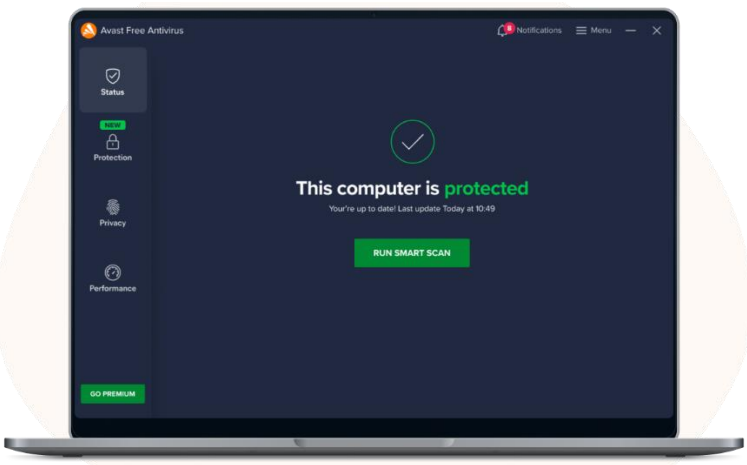
Steps to Perform the Experiment:

A. Installing Antivirus Software

1. Visit a trusted website (e.g., https://www.avast.com).
2. Download the free antivirus version (e.g., Avast Free Antivirus).
3. Run the installer and follow the on-screen instructions.
4. Restart the computer if prompted.

B. Running the Antivirus Scan

1. Launch the antivirus application.
2. Select Smart Scan or Full Scan option.
3. The antivirus will begin scanning all files and programs.
4. If threats are found:
    a. Choose “Resolve”, “Quarantine”, or “Delete” to remove them.
5. Review the scan report showing:
    a. Number of threats
    b. Actions taken
    c. Time taken for scan



Result:

Antivirus software was successfully installed and a full system scan was performed. The scan either removed or confirmed no presence of malware. The system is now protected against common cyber threats.

Experiment 13

Password Manager

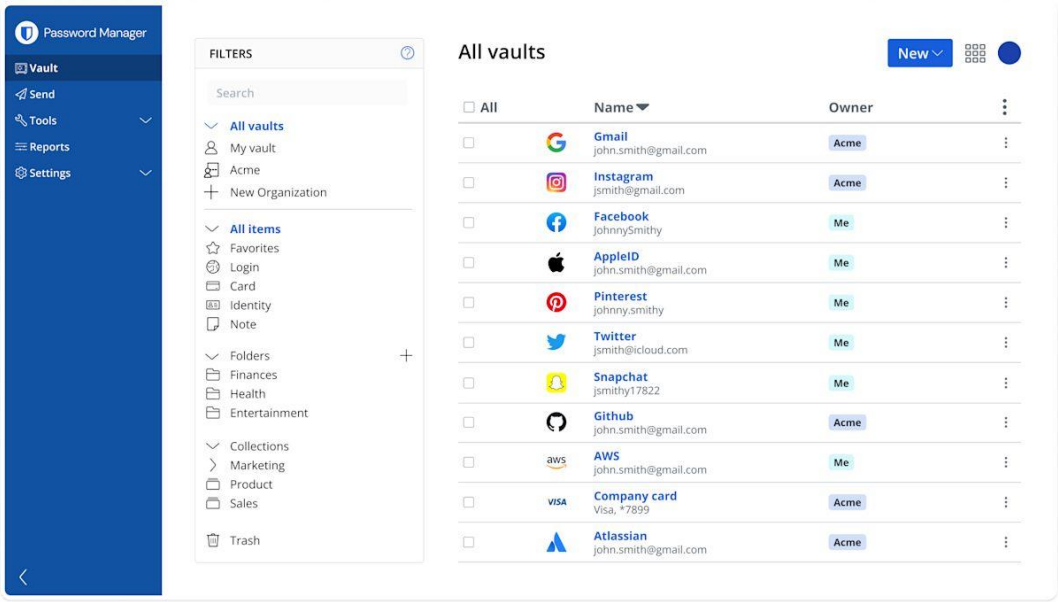
Aim:
To understand the importance of strong passwords and demonstrate the creation and storage of secure passwords using a password manager.

Theory:
About Password Managers
Password managers are secure tools used to generate, store, and manage passwords for various accounts. They prevent users from reusing weak passwords and protect credentials from phishing attacks.
Examples:

- Bitwarden (free, open-source)
- LastPass, Dashlane, 1Password, Google Password Manager

Steps to Perform the Activity (Using Bitwarden):

1. Go to https://bitwarden.com
2. Create a free Bitwarden account with a strong master password.
3. Log in and open the password vault.
4. Click on "Generator" to create a new password:
    a. Length: 16+ characters
    b. Include uppercase, lowercase, numbers, symbols
    c. Example: D$3kL#p9vQx!7eFb
5. Save the generated password for a sample website entry



Result:
Strong and secure passwords were generated and stored using a password manager. This method reduces the risk of password reuse, improves security, and supports safe browsing habits.

Experiment 14

Enable and Test Multi-Factor Authentication (MFA)

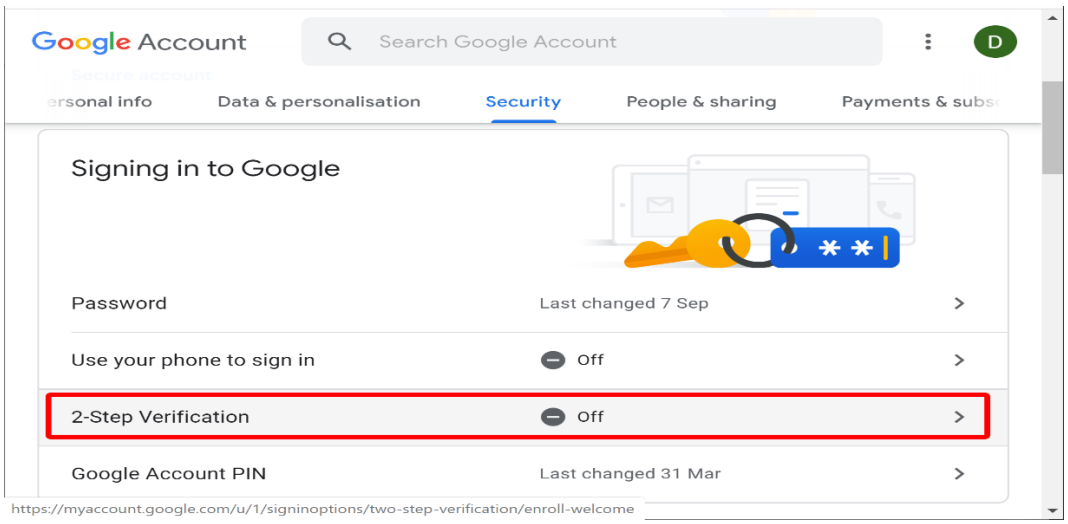
Aim:
To understand the concept of Multi-Factor Authentication (MFA) and enable it for an online account to enhance security.

Theory:
What is MFA?
Multi-Factor Authentication (MFA) adds an extra layer of security to the login process by requiring more than just a password. Common MFA methods include:

- Something you know (password)
- Something you have (OTP, phone)
- Something you are (fingerprint, face scan)

Steps to Enable MFA (Using Gmail as an Example):

1. Sign in to your Google account: https://myaccount.google.com
2. Go to Security tab → Find "2-Step Verification"
3. Click Get Started
4. Enter your password for verification
5. Add a phone number → Choose SMS or phone call
6. Enter the OTP sent to your phone
7. Click Turn On



Result:
Multi-Factor Authentication (MFA) was successfully enabled and tested. This ensures stronger protection against unauthorized access and phishing attacks. It is now recommended to enable MFA on all critical accounts.

Experiment 15

User Access Control (UAC)

Aim:

To configure User Access Control (UAC) settings in Windows to manage user privileges and enhance system security by preventing unauthorized changes.

Theory:

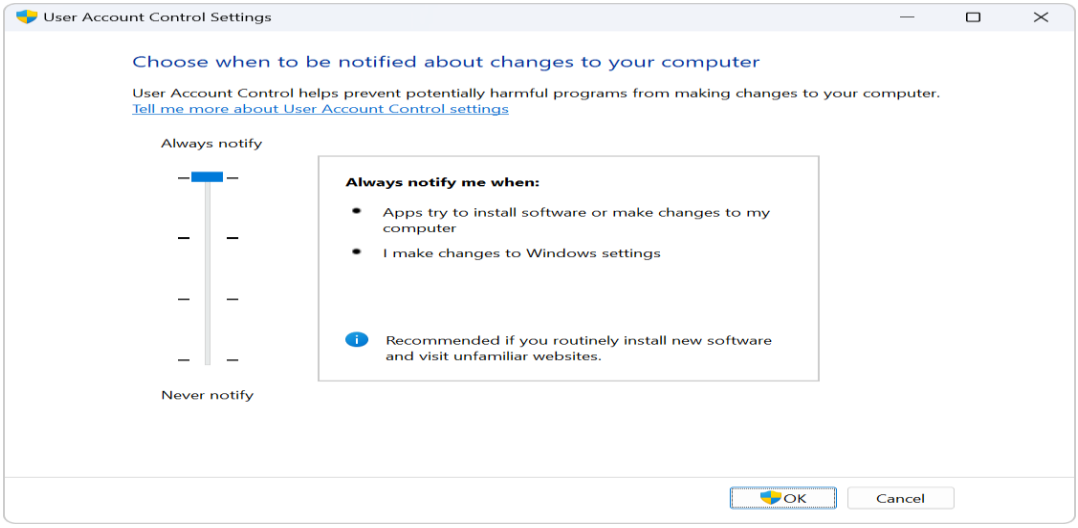
What is UAC?

User Access Control (UAC) is a security feature in Windows that helps prevent unauthorized changes to the operating system. It notifies or prompts the user when:

- A program tries to install software
- System settings are modified
- Administrative privileges are requested

Steps to Configure UAC Settings:

1. Press Windows + S → Search for "UAC"
2. Click on "Change User Account Control settings"
3. A slider will appear with 4 levels of notification:
    a. Always notify (most secure)
    b. Notify only when apps try to make changes (default)
    c. Notify without dimming desktop
    d. Never notify (least secure)
4. Move the slider to the desired level
5. Click OK → Enter admin password if prompted
6. Test by opening a setup file or trying to change a system setting



Result:

User Access Control (UAC) was successfully configured. It now prompts the user before any software or system-level change, ensuring greater control over administrative actions and protecting the system from potential threats.

Experiment 16

Websites Security

Aim:
To learn how to identify safe and unsafe websites using visual and technical cues like HTTPS, padlock icon, website reputation, and domain name authenticity.

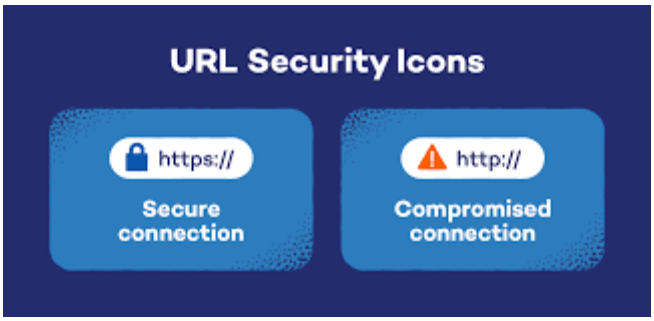
Theory:
Internet users often encounter both safe and unsafe websites. Recognizing the difference helps prevent phishing, malware downloads, and identity theft.

🔹 Characteristics of a Safe Website:

Feature	Example
✅ Uses HTTPS	URL starts with https://
✅ Padlock icon in address bar	Indicates secure SSL connection
✅ Trusted domain name	Like www.amazon.in, www.google.com
✅ Professional appearance	Clear layout, contact info, no errors
✅ Verified by antivirus	Antivirus shows a green tick or "Safe"

🔻 Signs of an Unsafe or Phishing Website:

Feature	Example
❌ Uses only HTTP	No encryption: http:// only
❌ Misspelled domain	www.go0gle.com, paypal-secure.xyz
❌ Pop-ups and ads	Too many pop-ups asking for clicks
❌ Poor design	Broken links, typos, or fake logos
❌ Antivirus warning	Browser or antivirus blocks the page



Result:
The experiment successfully demonstrated how to distinguish between safe and unsafe websites by observing their URLs, security icons, content quality, and browser warnings. This helps users avoid phishing, scams, and malware attacks.

Experiment 17

Encryption and Decryption

Aim:

To understand the process of protecting files using encryption and decrypting them using tools available in the operating system (Windows or Linux).

Theory:

What is File Encryption?

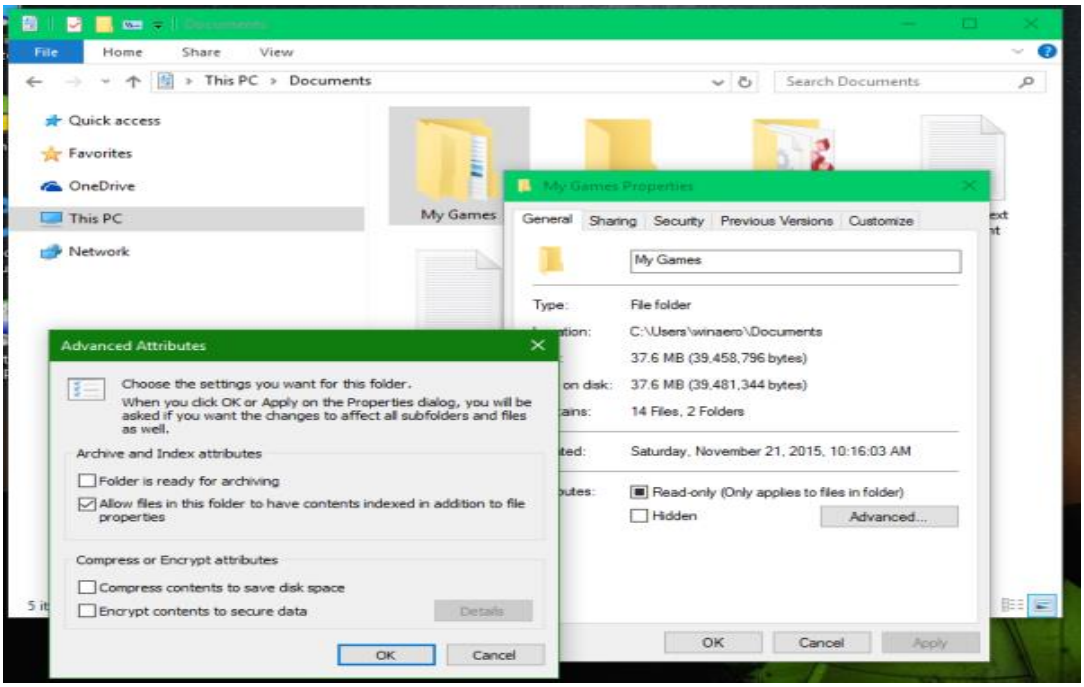
File encryption is a method of converting a file into an unreadable format to protect it from unauthorized access. Only users with the right key or credentials can decrypt it.

File Encryption in Windows (NTFS)

1. Right-click on the file or folder you want to encrypt.
2. Click Properties → Advanced...
3. Check the box "Encrypt contents to secure data"
4. Click OK → Apply
5. Windows encrypts the file using your user account credentials.
6. Only your Windows user profile can open the file.

To Decrypt:

- Go back to Properties → Advanced
- Uncheck "Encrypt contents to secure data" → Apply



Result:

Files were successfully encrypted and decrypted using built-in OS tools in both Windows and Linux. This enhances data security by preventing unauthorized access to confidential files.

Experiment 18

Data Backup

Aim:
To demonstrate the process of backing up important files and folders to an external drive, cloud storage, or another location to ensure data safety in case of system failure or loss.

Theory:
What is Data Backup?
Backup is the process of copying and storing data in a different location (external device or cloud) so it can be restored if lost or corrupted.

🔻 Types of Backup Methods:

- Manual backup: Copy files to USB drive, external HDD
- Cloud backup: Use Google Drive, OneDrive, Dropbox
- Automated backup: Use built-in tools like Windows Backup

Steps to Perform Basic Manual Backup (Windows):

1. Connect an external USB drive or open your cloud storage folder (e.g., Google Drive).
2. Select the files or folders to back up (e.g., Documents, Pictures).
3. Right-click → Copy
4. Navigate to the destination (USB drive or cloud folder).
5. Right-click → Paste
6. Confirm files are successfully copied by opening a few.

Backup to Google Drive:

1. Open https://drive.google.com
2. Click "+ New" → File upload or Folder upload
3. Select files and wait for upload to finish

Result:
Important files were successfully backed up to both an external drive and cloud storage. This ensures that the data remains safe and can be restored in case of hardware failure, accidental deletion, or malware attack.

Experiment 19

Fake Websites and Phishings

Aim:

To identify the key signs of fake websites and phishing emails to avoid falling victim to online scams and cyberattacks.

Theory:

What is Phishing?

Phishing is a cyberattack where attackers send fake emails or websites to trick users into sharing confidential information like passwords, bank details, or OTPs.

How to Spot a Fake Website:

Checkpoint	Real Website	Fake Website Example
🔒 HTTPS/Padlock	✓ https://www.amazon.in	✗ http://amaz0n-offer.xyz
✓ Correct Domain Name	paypal.com	paypal-secure-login.net (fake)
🖌 Clean Design	Professional UI, working links	Poor design, typos, broken links
🎯 Look for Trust Seals	Norton, McAfee, etc.	Often missing or fake icons

How to Spot a Phishing Email:

Clue	Description
🎣 Suspicious Sender	support@bank-secure-mail.com instead of @hdfcbank.com
⚠ Urgent Message	"Your account will be blocked! Click now to verify."
🔗 Fake Links	Hover reveals mismatched URLs (hdfc-login.ru)
📎 Attachments	Unknown file formats (e.g., .exe, .zip)
✗ Spelling Errors	Poor grammar and fake logos

Result:

The experiment successfully demonstrated how to spot fake websites and phishing emails by checking domain names, links, security indicators, and language. Recognizing these signs helps prevent online fraud and identity theft.

Experiment 20

Real vs. Fake Websites

Aim:
To compare a real website with a fake (phishing) website and identify key differences in domain name, security, layout, and trust indicators to improve cyber awareness.

Theory:
Fake websites are created to mimic real ones with the goal of stealing login credentials, banking details, or personal information. Spotting the differences between them can prevent serious cyber threats.

Step-by-Step Analysis:

Feature	Real Website (e.g., https://www.amazon.in)	Fake Website (e.g., http://amaz0n-offer.xyz)
HTTPS with Padlock	✓ Yes (encrypted)	✗ No padlock / "Not Secure" warning
Domain Name	Proper and short	Misspelled or long subdomains
Website Design	Clean, professional, with proper logos	Poor layout, pixelated images, fake logos
Login URL	/login or /signin under main domain	Redirects to strange URLs
Grammar/Spelling	Professional language	Often has typos and awkward phrases
Trust Symbols	Verified badges, privacy policy	Fake seals or missing entirely
Pop-ups/Ads	Minimal and controlled	Multiple pop-ups urging urgent action

Result:
Real and fake websites were analyzed side-by-side. Key differences in domain names, site layout, and security features were observed. This activity helps students become more alert while browsing and logging into websites.

Experiment 21

Identify Phishing Emails

Aim:

To recognize phishing emails by analyzing their structure, content, and sender details, and to understand how to avoid falling victim to such scams.

Theory:

What is a Phishing Email?

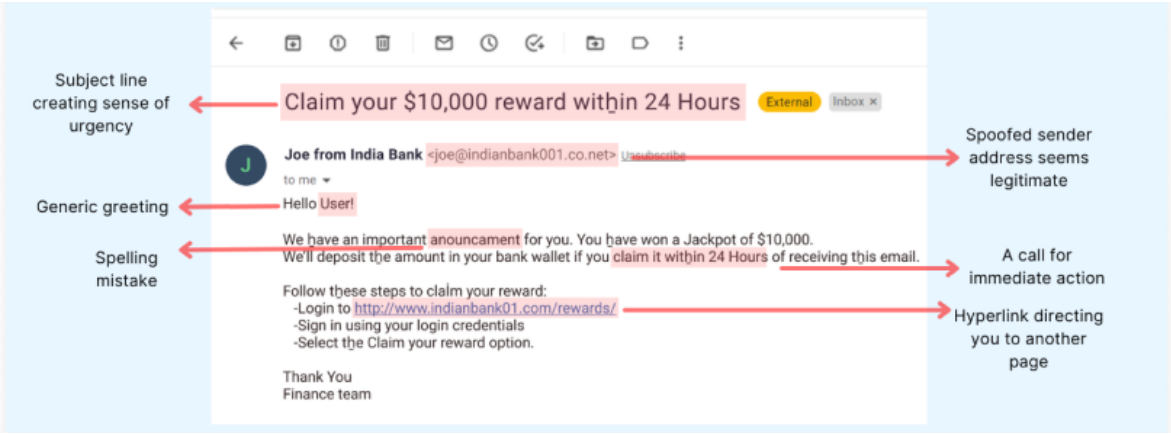
A phishing email is a fake message that appears to be from a trusted organization (bank, social media, company) but is designed to steal personal or financial information.

Common Signs of a Phishing Email:

Feature	Phishing Indicator
Suspicious sender	E.g., security@hdfcbank-alerts.xyz
Urgent language	"Your account will be blocked!" or "Verify now!"
Fake links	Hovering reveals a non-matching or unusual URL
Unexpected attachments	.exe, .zip, or fake invoices
Spelling/grammar errors	Unprofessional content or formatting
Requests sensitive info	Asks for passwords, OTPs, or card details

Steps Performed:

1. Opened sample phishing emails.
2. Checked sender's email ID and domain.
3. Hovered over links to preview real URLs.
4. Looked for spelling errors and threats (e.g., "Your account is suspended").
5. Compared with a legitimate email from the same company.
6. Reported or marked the phishing mail as spam.



Result:

Phishing emails were successfully identified based on sender details, links, content structure, and urgency tone. Awareness of these indicators helps users avoid cyber fraud and data theft.

Experiment 22

Check OS Updates and Remove Unnecessary Applications

Aim:

To ensure system performance and security by checking for operating system and software updates, and removing unused or unwanted applications.

Theory:

Keeping the operating system and installed software up to date helps protect the system from vulnerabilities, improves performance, and adds new features. Removing unused applications frees up memory and improves speed.

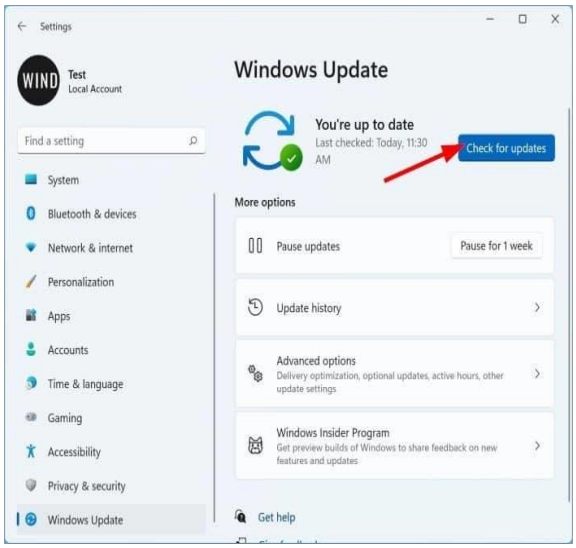
Part A: Check for OS & Software Updates

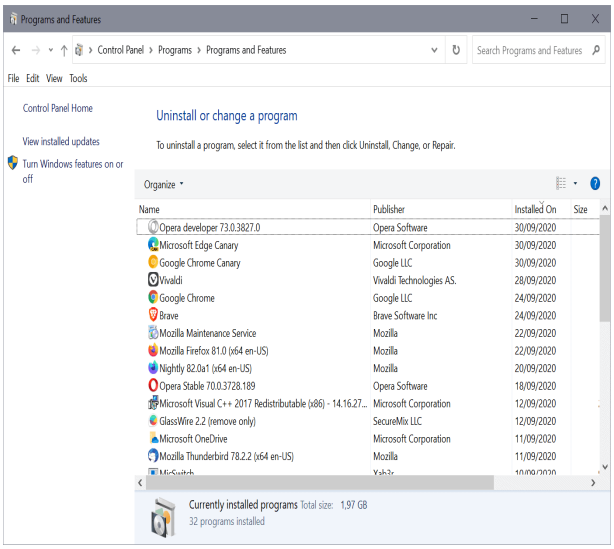
Windows OS Update:

1. Open Settings → Click on Update & Security
2. Select Windows Update
3. Click Check for updates
4. If updates are available, click Download & Install
5. Restart the system if prompted

Part B: Uninstall Unnecessary Applications

1. Open Control Panel → Programs → Uninstall a Program
2. Review the list of installed applications
3. Select unused software (e.g., toolbars, outdated games)
4. Click Uninstall and follow prompts
5. Restart if required





Result:

The operating system and key software were successfully updated, and unnecessary applications were uninstalled. This improved system performance, freed up disk space, and enhanced overall security.

Experiment 23

Recognize Scam Calls and Messages

Aim:

To identify characteristics of scam calls and fraudulent messages, and learn how to respond safely to avoid being tricked into revealing personal or financial information.

Theory:

📱 What Are Scam Calls and Messages?

Scam calls and messages are fraudulent attempts by cybercriminals to trick individuals into sharing confidential information such as OTPs, passwords, bank details, or installing malicious apps.

🔻 Common Signs of a Scam Call:

- The caller pretends to be from a bank, telecom provider, or government agency
- They demand urgent action (e.g., "Your account will be blocked!")
- Ask for sensitive data like PIN, OTP, or Aadhar number
- Offer too-good-to-be-true prizes or job offers
- Speak in a threatening or overly friendly tone

🔻 Common Signs of a Scam SMS or WhatsApp Message:

- Contains suspicious links (e.g., bit.ly/freedata)
- Promises lottery wins, cashback, free recharge
- Asks to forward the message to others to "claim reward"
- Contains spelling errors and generic greetings ("Dear Customer")

📃 How to Respond:

- Do not share personal info over phone or message
- Do not click on suspicious links
- Report the number to your service provider or cybercrime portal
- Use Truecaller or spam filters to block scam numbers
- Verify claims through official channels only

Result:

Scam call and message patterns were successfully identified by examining their content, tone, and demands. Awareness of such fraud attempts helps avoid financial and identity theft, promoting safer digital communication.

Experiment 24

Explore Block Coding Tool Interface (Using Scratch)

Aim:

To explore the user interface of a block-based programming environment like Scratch and understand its major components.

Theory:

What is Block Coding?

Block coding is a visual programming method where users create programs by dragging and dropping code blocks instead of writing text-based code. This approach is ideal for beginners and helps build logical thinking.

Scratch is a popular, free block coding tool available at https://scratch.mit.edu.

🔹 Steps to Explore Scratch Interface:

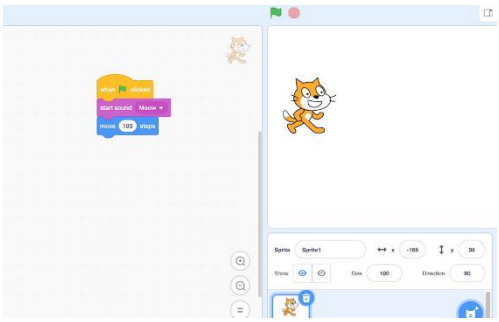
1. Open a browser and go to: https://scratch.mit.edu
2. Click "Start Creating" to launch the editor
3. Familiarize yourself with the following interface components:

Interface Section	Description
Stage Area	Where animations and output appear
Sprite Pane	Contains characters (sprites) that perform actions
Blocks Palette	Categories of code blocks (Motion, Looks, Events, Control, etc.)
Script Area	Drag and drop blocks here to build code
Toolbar	File options, tutorials, and login info
Green Flag / Red Stop	Start and stop buttons for the project

4. Click on different block categories and observe how blocks snap together.

📚 Main Block Categories in Scratch:

- Motion – Move, turn, glide sprites
- Looks – Change costumes, say something
- Sound – Play sounds or music
- Events – Start program using flags or keys
- Control – Loops, conditions (if, repeat)
- Sensing – Detect touches, mouse position, etc.



Result:

The block coding tool Scratch was successfully explored. Major components of its interface and code block categories were identified, providing a foundation to create animations, stories, and games through visual programming.

Experiment 25

Develop Algorithms and Flowcharts for:
a) Arithmetic Operations
b) Metric Conversions

Aim:
To develop basic algorithms and draw flowcharts to solve arithmetic and metric conversion problems using logical steps and structured diagrams.

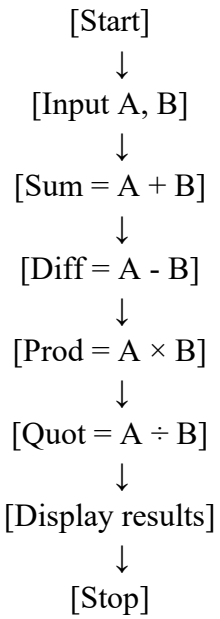
🧮 Part A: Arithmetic Operations

📝 Problem:
Design an algorithm and flowchart to perform basic arithmetic operations (Addition, Subtraction, Multiplication, Division) on two numbers.

✅ Algorithm:

1. Start
2. Input two numbers: A, B
3. Calculate:
    a. Sum = A + B
    b. Difference = A - B
    c. Product = A × B
    d. Quotient = A ÷ B
4. Display all results
5. Stop

Flowchart:



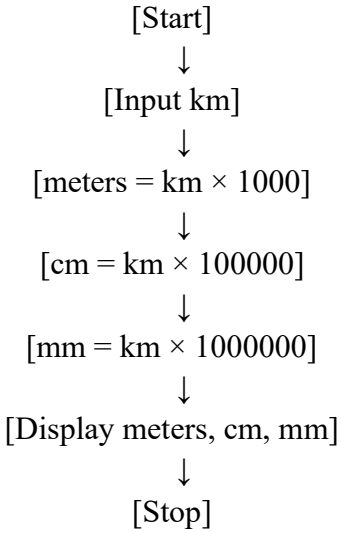
Part B: Metric Conversions

Problem:
Convert kilometers to meters, centimeters, and millimeters.

✓ Algorithm:

1. Start
2. Input distance in kilometers (km)
3. Convert to:
    a. Meters = km × 1000
    b. Centimeters = km × 100000
    c. Millimeters = km × 1000000
4. Display all converted values
5. Stop

Flowchart:



Result:
Algorithms and flowcharts were successfully developed for arithmetic operations and metric conversions. This practice enhanced logical thinking and understanding of structured problem-solving techniques.

Experiment 26

Create an Animated Sequence Using Scratch

Aim:
To design and execute a basic animated scene using Scratch, involving sprite movement, background changes, and visual effects.

Theory:
Scratch is a block-based visual programming language ideal for creating animations, games, and stories.

🔹 Steps to Create an Animated Scene in Scratch:

1. Open https://scratch.mit.edu
2. Click “Create” to launch the editor
3. Delete the default cat sprite if needed and choose new sprites (e.g., Butterfly, Sun, Boy)
4. Choose a background from the Scratch library (e.g., Garden or Beach)
5. Drag motion blocks to move the sprite:
    a. go to x: -100 y: 0
    b. glide 2 secs to x: 100 y: 0
6. Use Looks blocks to create animation effects:
    a. say "Hello!" for 2 seconds
    b. next costume for sprite animation
7. Add Control blocks to loop the movement or wait:
    a. repeat, wait, or forever
8. Click the Green Flag to start the animation

📜 Sample Code (in Blocks):

```
when green flag clicked
go to x: -150 y: 0
repeat 5
  glide 1 secs to x: (x + 50) y: 0
  next costume
end
say "Animation Complete!" for 2 seconds
```

Result:
An animated sequence was successfully created using Scratch. Sprites moved across the screen with costume changes and text effects, demonstrating understanding of animation logic and block coding.

Experiment 27

Design Flowchart for a Real-World Task (e.g., Making a Cup of Tea)

Aim:

To design a flowchart representing the logical steps involved in completing a real-world task using standard flowchart symbols.

Steps:

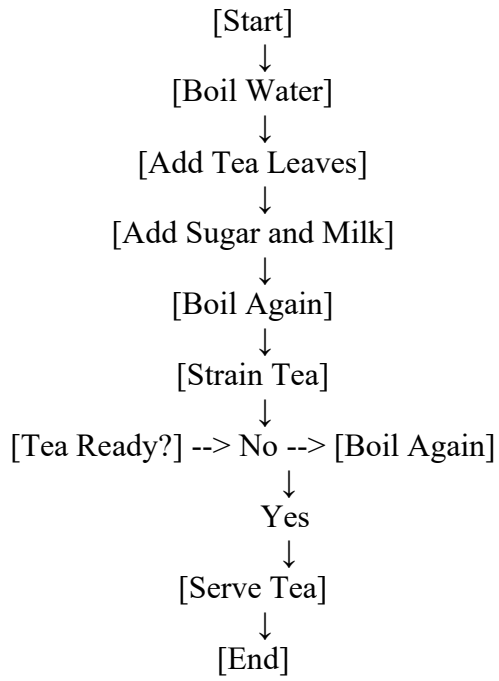
Chosen Real-World Task:

Making a Cup of Tea

✓ Steps to Represent:

1. Start
2. Boil water
3. Add tea leaves to boiling water
4. Add sugar and milk
5. Boil again
6. Strain tea
7. Is the tea ready?
    a. Yes → Serve tea
    b. No → Go back to boil
8. End

Flowchart:



Result:

A clear and structured flowchart was created for a real-world task. This helped understand how logical thinking and sequential representation can simplify problem-solving in both computing and daily life activities.

Experiment 28

Create a Score Counter for a Simple Game Using Scratch

Aim:

To implement a score counter in a basic Scratch game where the score increases when the sprite touches a target.

Steps:

Scratch allows the creation of interactive games with scorekeeping, using variables to store and display the score during gameplay.

🔹 Steps to Create the Game with Score Counter:

1. Open https://scratch.mit.edu and click Create
2. Add two sprites:
    a. Player sprite (e.g., Cat)
    b. Target sprite (e.g., Apple or Ball)
3. Click Variables → Make a Variable → Name it score
4. Drag set score to 0 under the when green flag clicked block
5. Move the player using arrow key events:

```
when key [right arrow] pressed
change x by 10
```

6. Add code to check collision with target:

```
if <touching [Apple]> then
    change [score v] by 1
    go to [random position]
end
```

7. Display the score on screen during the game.

📜 Sample Script (Player Sprite):

```
when green flag clicked
set [score v] to 0
forever
  if <touching [Apple]> then
    change [score v] by 1
    go to [random position v]
  end
end
```

Result:

A simple interactive game was created with a score counter. Each time the player touches the target, the score increases, demonstrating basic event handling, collision detection, and variable use in Scratch.

Experiment 29

Create an Interactive Greeting App with User Input Using Scratch

Aim:

To create a Scratch-based interactive program that takes the user's name as input and displays a personalized greeting on the screen.

Steps:

Scratch allows interactive programming using the ask block to receive input from the user and say blocks to respond dynamically.

🔹 Steps to Create the Interactive App:

1. Open https://scratch.mit.edu and click Create
2. Use the default Cat sprite or choose another sprite for greeting
3. Go to the Sensing category → Drag out the ask [] and wait block
4. In the ask block, type:
   What is your name?
5. Go to Looks → Use the say block and insert a join operation:
   say (join [Hello,] (answer)) for 2 seconds
6. You can also add a second ask (e.g., "How are you?") and respond again.

📃 Sample Script:

```
when green flag clicked
ask [What is your name?] and wait
say (join [Hello, ] (answer)) for 2 seconds
ask [How are you today?] and wait
say (join [Nice to hear you're ] (answer)) for 2 seconds
```

Result:

A personalized greeting app was successfully created in Scratch. The program took input from the user and responded with a friendly, interactive message, demonstrating the use of input, variables, and output.

Experiment 30

Create an Algorithm and Flowchart with Conditions — Compare Two Numbers

Aim:

To design an algorithm and draw a flowchart that compares two numbers and displays which one is greater, or if they are equal, using conditional statements.

Steps:

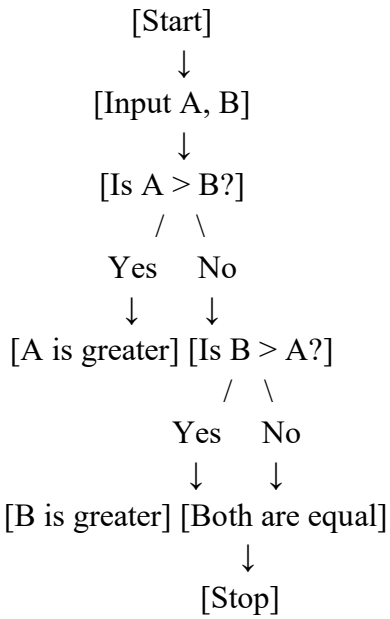
Conditional logic (like if–else) is fundamental in programming. It helps decision-making based on comparisons such as "greater than", "less than", or "equal to".

This experiment uses a simple decision box in the flowchart to evaluate which number is greater.

✓ Algorithm: Compare Two Numbers

1. Start
2. Input two numbers: A and B
3. If A > B, display “A is greater”
4. Else if B > A, display “B is greater”
5. Else, display “Both are equal”
6. Stop

Flowchart:



Result:

An algorithm and flowchart using conditional logic were successfully developed to compare two numbers. The output varies based on the values entered, demonstrating logical branching and decision-making in computing.

Experiment 31

Create an Interactive Story with Yes/No Choices Using Scratch

Aim:

To design a simple interactive story in Scratch that asks the user yes/no questions and changes the storyline based on the response.

Steps:

In Scratch, you can use the ask [ ] and wait block along with if...else conditions to allow the user to make decisions that affect how the story unfolds.

This promotes branching logic, a key programming concept in interactive applications and games.

🔹 Steps to Create the Story:

1. Go to https://scratch.mit.edu → Click Create
2. Choose a character sprite (e.g., Knight, Cat, or Fairy)
3. Choose a background (e.g., Forest, Castle)
4. Use the Sensing block ask [Do you want to go left?] and wait
5. Use Control → if/else to branch the story:

```
if <(answer) = [yes]> then
   say [You find a treasure chest!] for 2 secs
else
   say [You meet a dragon!] for 2 secs
end
```

6. Add more steps by nesting or adding more ask blocks for continued story branching.

📜 Sample Script:

```
when green flag clicked
say [Welcome to the magic forest!] for 2 seconds
ask [Do you want to go left? (yes/no)] and wait
if <(answer) = [yes]> then
   say [You find a hidden cave!] for 2 seconds
else
   say [A wild creature appears!] for 2 seconds
end
```

Result:

An interactive story was successfully created using yes/no input from the user. The story branches based on the choices made, demonstrating use of input handling, conditions, and storytelling in Scratch.

Experiment 32

Create a Traffic Light Simulator Using Conditions in Scratch

Aim:
To create a traffic light simulator in Scratch that uses conditions and timing to control the change of lights in a sequence (Red → Green → Yellow → Red...).

Steps / Theory:
Traffic lights operate in a cycle with specific timing for each light. Using Scratch, we simulate this using costume switching, timing, and conditional logic.

🔷 Steps to Build the Simulator:

1. Go to https://scratch.mit.edu → Click Create
2. Add or draw a traffic light sprite with 3 costumes:
    a. Red light on
    b. Green light on
    c. Yellow light on
3. Create a when green flag clicked block to start the simulation
4. Use a forever loop to change costumes with delays:

```
when green flag clicked
forever
  switch costume to [Red]
  wait 3 seconds
  switch costume to [Green]
  wait 3 seconds
  switch costume to [Yellow]
  wait 2 seconds
end
```

📃 Concepts Applied:

- Conditional logic (if...then)
- Loops (forever)
- Costume switching
- Timing and sequencing

Result:
A working traffic light simulator was successfully created in Scratch. The program uses conditions and timing to simulate a real-life traffic signal, demonstrating automation logic and sequence control. Bouncing ball animation using loops.

Experiment 33

Create a Bouncing Ball Animation Using Loops in Scratch

Aim:
To simulate a bouncing ball animation in Scratch using motion blocks and loops for continuous movement and boundary detection.

Steps:
In animation, a bouncing object changes direction when it touches an edge or another surface. In Scratch, this can be implemented using:

- Motion blocks
- Edge detection
- Forever loops
- Direction control

🔷 Steps to Build the Bouncing Ball Animation:

1. Open https://scratch.mit.edu → Click Create
2. Delete the default cat sprite and choose Ball from the sprite library
3. Set the starting position:

   when green flag clicked
   go to x: 0 y: 0
   point in direction (45)
4. Use a forever loop with bounce logic:

   forever
     move 10 steps
     if on edge, bounce
   end
5. You can enhance with effects:
   a. change color effect
   b. play sound when bouncing

Result:
A bouncing ball animation was successfully created using loops and motion controls. The ball continues to move and bounce off screen edges, demonstrating continuous looping and simple physics simulation.

Experiment 34

Counting Program (1 to 20 using loops)

Aim:

To create a Scratch program that counts numbers from 1 to 20 using the repeat loop.

Steps:

Loops in Scratch allow blocks to run repeatedly without duplication. By combining variables and repeat blocks, we can display a sequence of numbers.

🔻 Steps in Scratch:

1. Open Scratch → Create a new project.
2. Delete the Cat sprite if not required.
3. Create a variable named count.
4. Drag and arrange the following blocks:

```
when green flag clicked
set [count v] to (1)
repeat (20)
say (count) for (1) second
change [count v] by (1)
end
```

Explanation:

set count to 1 initializes the counter.

repeat (20) ensures the block runs 20 times.

say (count) displays the current number.

change count by 1 increases the number each loop.

Result:

The sprite successfully displayed numbers from 1 to 20 in order using a loop and variable control in Scratch.

Experiment 35

Create a Free Cloud Account (AWS, Microsoft Azure, or Google Cloud Platform)

Aim:

To create and explore a free-tier cloud computing account on a major platform (Amazon Web Services - AWS, Microsoft Azure, or Google Cloud Platform - GCP) for basic cloud learning and services.

Steps:

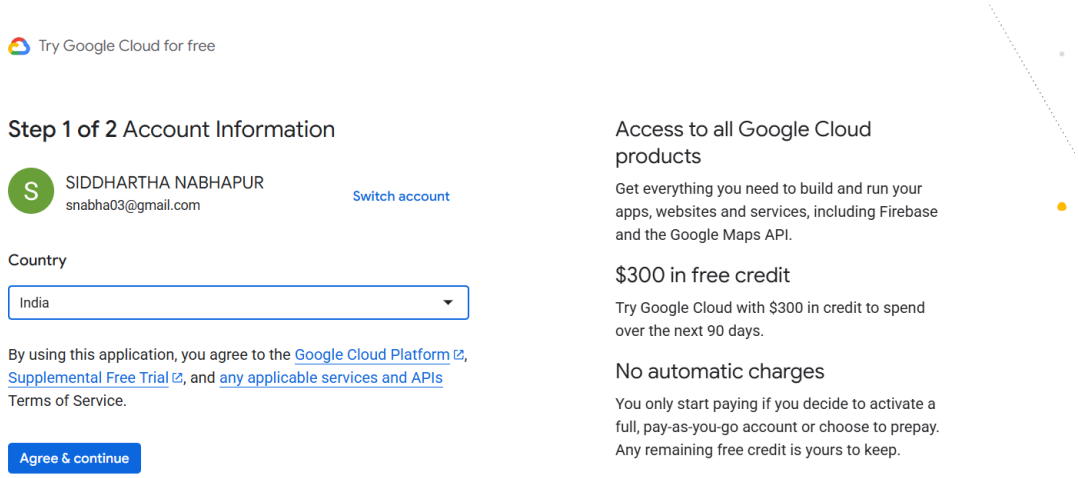
Cloud platforms provide computing, storage, and networking services over the internet. Leading providers like AWS, Azure, and GCP offer free-tier accounts for students and beginners to practice cloud computing.

🔷 Steps to Create a Free Cloud Account (GCP Example):

1. Go to: https://cloud.google.com
2. Click on "Get started for free"
3. Sign in with a Google account
4. Accept the terms and conditions
5. Enter basic contact information
6. Add payment method (only for identity verification – no charge)
7. Receive $300 free credits valid for 90 days
8. Dashboard opens → Explore services (e.g., VM, Storage, APIs)

Similar steps apply for:

- AWS: https://aws.amazon.com/free
- Azure: https://azure.microsoft.com/en-in/free



Result:

A cloud computing account was successfully created using GCP's free tier. The platform allows exploration of cloud services like virtual machines, cloud storage, and APIs, supporting hands-on learning in cloud technologies.

Experiment 36

Explore Google Cloud Console and Services

Aim:

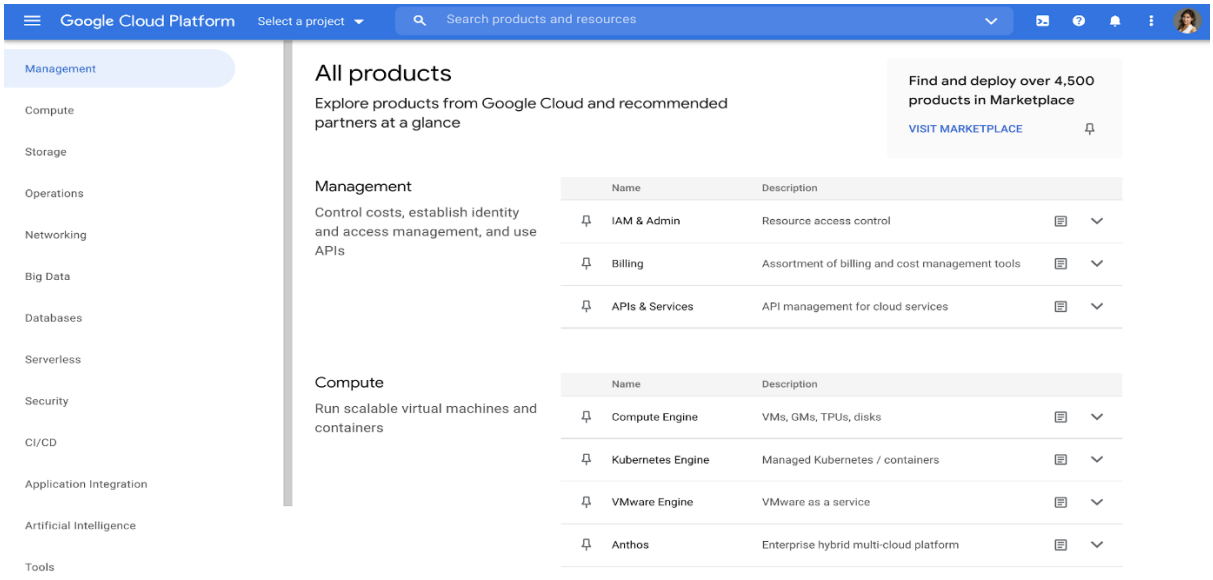
To explore the Google Cloud Platform (GCP) Console and understand its services such as compute, storage, networking, and AI tools.

Steps:

Google Cloud Platform (GCP) is a cloud computing service offered by Google that provides resources like virtual machines, cloud storage, APIs, and machine learning services. The Cloud Console is a web-based interface to manage these services.

🔹 Steps to Explore GCP Console:

1. Go to https://console.cloud.google.com
2. Log in with your Google account
3. Accept terms → Start Free Trial ($300 credits for 90 days)
4. Dashboard opens → Explore services:
5. Compute Engine → Create Virtual Machines
6. Cloud Storage → Upload/Download files
7. BigQuery → Analyze large datasets
8. AI & Machine Learning APIs → Use Vision API, Translation API
9. Networking → Configure VPCs, load balancers
10. Navigate using the left-side menu and search bar for specific services.



Result:

The Google Cloud Console was successfully accessed and explored. Key services like Compute Engine, Storage, and AI APIs were identified, demonstrating how GCP enables cloud computing for real-world applications.

Experiment 37

Upload and Download Files via Cloud Storage (e.g., Google Drive, OneDrive, Dropbox)

Aim:
To understand and perform the process of uploading and downloading files using a cloud storage service like Google Drive, OneDrive, or Dropbox for secure and remote file access.

Steps:
Cloud storage is a service where data is stored, managed, and backed up remotely and made available to users over the internet.
Popular free cloud platforms:

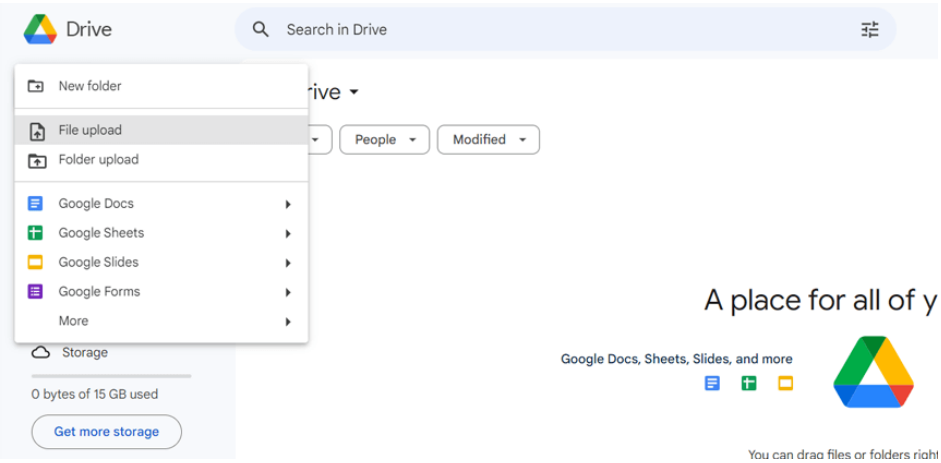
- Google Drive (15 GB free): https://drive.google.com
- Microsoft OneDrive (5 GB free): https://onedrive.live.com
- Dropbox (2 GB free): https://www.dropbox.com

🔺 Steps to Upload Files (Google Drive Example):

1. Go to https://drive.google.com
2. Log in using a Google account
3. Click "+ New" → "File upload" or "Folder upload"
4. Select the file (e.g., PDF, DOCX, image) from your device
5. Wait for the upload to complete
6. File appears in the main drive window

🔻 Steps to Download Files:

1. Right-click on any file in Google Drive
2. Click "Download"
3. The file is saved to your local system (Downloads folder)



Result:
Files were successfully uploaded to and downloaded from Google Drive. The experiment demonstrated secure file handling through cloud storage, an essential skill for digital file management and remote collaboration.

Experiment 38

Create Online Forms or Surveys Using Cloud Services (e.g., Google Forms)

Aim:
To design and publish an online form or survey using cloud-based tools such as Google Forms, enabling digital data collection and analysis.

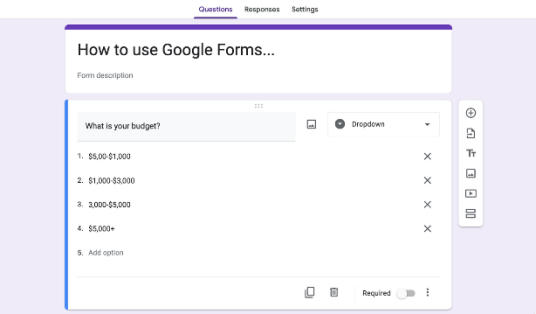
Steps:
Online forms are digital tools used to collect information, conduct surveys, or receive feedback. Cloud-based form builders like Google Forms make it easy to create, distribute, and manage responses in real time.

🔻 Steps to Create a Form (Google Forms Example):

1. Go to https://forms.google.com
2. Click “Blank” or choose a template
3. Add a title and description for your form (e.g., "Student Feedback")
4. Click “+” to add questions
5. Choose question types:
    a. Multiple choice
    b. Short answer
    c. Checkboxes
    d. Dropdown
6. Enable “Required” for mandatory questions
7. Click on the palette icon to change theme or background color
8. Click “Preview” (eye icon) to test the form
9. Click “Send” to share the form via link, email, or embed

📊 Responses:

- Click “Responses” tab to view answers
- Click the green Google Sheets icon to open the data in a spreadsheet



Result:
An online form was successfully created, customized, and shared using Google Forms. Responses were collected and viewed in spreadsheet format, demonstrating how cloud services simplify digital data collection and analysis.

Experiment 39

Blink an LED Using Visual Block Programming with Arduino

Aim:

To use block-based visual programming (e.g., via mBlock or Tinkercad Circuits) to blink an LED connected to an Arduino, demonstrating hardware interaction through simple logic.

Steps:

Arduino is an open-source electronics platform. Using visual programming tools like mBlock or Tinkercad, we can control hardware (like LEDs) with simple drag-and-drop coding blocks instead of writing complex code.

🔻 Tools Required:

- Arduino Uno board (or virtual simulator like Tinkercad)
- LED
- Resistor (220 ohm)
- Breadboard
- Jumper wires
- Visual programming tool (e.g., mBlock, Tinkercad)

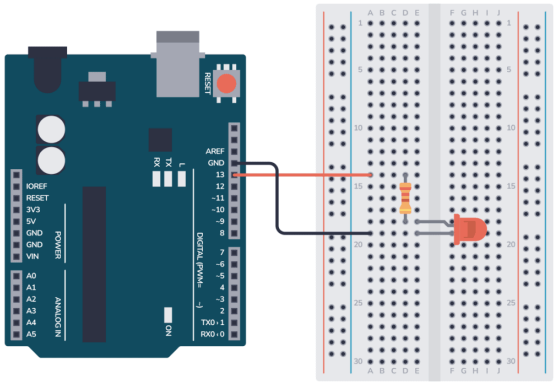
🔻 Wiring Connections:

- Connect LED anode (+) to Arduino Pin 13
- Connect cathode (-) through resistor to GND

🔻 Steps in Visual Programming (mBlock/Tinkercad):

1. Open https://www.tinkercad.com → Circuits
2. Drag Arduino and LED to the breadboard
3. Connect the LED as shown above
4. Use code blocks:

```
when Arduino starts up
forever
  set digital pin 13 output as HIGH
  wait 1 seconds
  set digital pin 13 output as LOW
  wait 1 seconds
```



Result:

The LED was successfully programmed to blink using visual block coding. This activity demonstrated basic control of electronic components using logical sequences and introduced physical computing concepts.

Experiment 40

Create a 3-LED Traffic Signal Controller Using Arduino and Visual Block Programming (Tinkercad/mBlock)

Aim:

To simulate a traffic light system using three LEDs (Red, Yellow, Green) connected to an Arduino and controlled through block-based programming.

Steps:

A traffic signal operates in a cycle (Red → Green → Yellow → Red...). Using visual block coding platforms like Tinkercad or mBlock, we simulate this sequence using an Arduino board and three LEDs.

🔷 Tools/Components Required:

- Arduino Uno (physical or virtual via Tinkercad)
- 3 LEDs (Red, Yellow, Green)
- 3 × 220-ohm resistors
- Breadboard
- Jumper wires
- Visual block-based IDE (mBlock or Tinkercad's code blocks)

🔷 Circuit Connections:

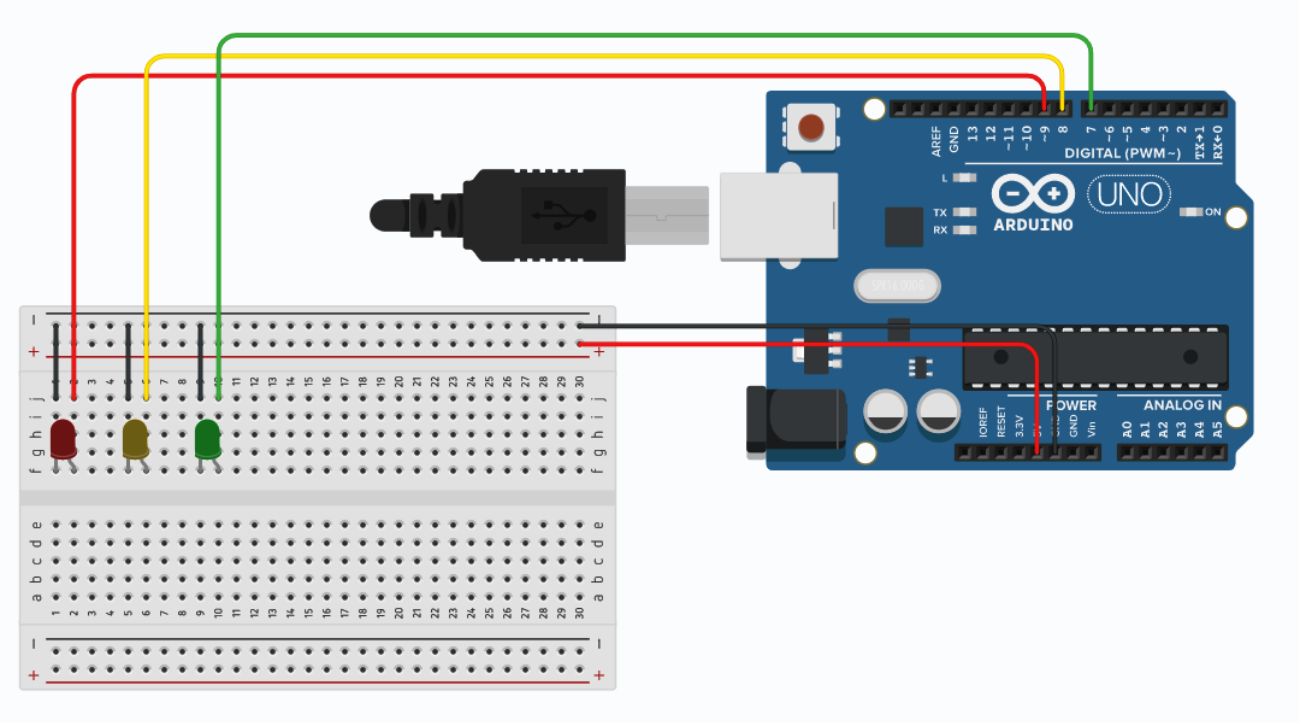
LED Color	Arduino Pin	Connection Details
Red	D2	Through resistor to GND
Yellow	D3	Through resistor to GND
Green	D4	Through resistor to GND

🔷 Visual Block Logic (mBlock or Tinkercad):

```
when Arduino starts up
forever
   set digital pin 2 output HIGH   // Red ON
   wait 3 seconds
   set digital pin 2 output LOW

   set digital pin 4 output HIGH   // Green ON
   wait 3 seconds
   set digital pin 4 output LOW

   set digital pin 3 output HIGH   // Yellow ON
   wait 1 seconds
   set digital pin 3 output LOW
```



Result:

A working 3-LED traffic signal system was created using Arduino and block-based programming. The experiment helped visualize real-world automation logic and reinforced digital output control using microcontrollers.

Experiment 41

Explore AI Tools: ChatGPT, Gemini, Grok, GitHub Copilot, Sora, and More

Aim:

To explore and understand various modern Artificial Intelligence (AI) tools such as ChatGPT, Google Gemini, xAI's Grok, GitHub Copilot, and Sora, and their real-world applications in productivity, learning, and creativity.

Steps:

Artificial Intelligence (AI) tools are designed to simulate human intelligence and perform tasks like reasoning, learning, writing, coding, generating media, and more. These tools are widely used in education, software development, and automation.

🔹 Explored Tools:

AI Tool	Platform	Purpose/Use
ChatGPT	OpenAI	Answering questions, writing code, essays, tutoring
Gemini	google	Summarizing docs, generating content, search assist
Grok	xAI (by Elon Musk)	AI chatbot in X (formerly Twitter)
GitHub Copilot	GitHub + Microsoft	AI coding assistant for developers (VS Code)
Sora	OpenAI (video AI)	Generates realistic videos from text (early access)

🔹 Steps Taken:

1. Visited each official platform
2. Explored features, such as:
    a. Asking questions (ChatGPT, Gemini)
    b. Generating summaries
    c. Writing simple code with Copilot
    d. Watching Sora video demos
3. Compared their strengths in creativity, accuracy, and coding assistance

📄 Applications of AI Tools:

- Writing essays, emails, or stories
- Debugging or generating code
- Explaining complex topics simply
- Designing educational or promotional content
- Video generation and editing (Sora)

Result:

Multiple AI tools were explored. Each has unique strengths: ChatGPT for conversational help and learning, Gemini for Google-integrated research, Copilot for code generation, Sora for visual creativity, and Grok for social media intelligence. These tools enhance productivity and digital skills.

Experiment 42

Test Basic Prompts and Analyze Responses Using AI Tools

Aim:
To interact with AI models using basic text prompts and evaluate their responses for accuracy, clarity, creativity, and usefulness.

Steps:
Prompts are user inputs (questions or commands) given to AI models like ChatGPT, Claude, or Gemini. The quality and structure of prompts directly affect the relevance and quality of the AI's responses.

🔻 Steps Performed:

1. Opened ChatGPT, Claude, and Gemini (Bard)
2. Entered a set of basic prompts:
    a. Prompt 1: “What is the Internet?”
    b. Prompt 2: “List 5 uses of cloud computing”
    c. Prompt 3: “Explain the water cycle in simple words”
    d. Prompt 4: “Write a short story about a robot”
    e. Prompt 5: “Solve: What is 15% of 240?”
3. Observed and analyzed the answers across each model
4. Compared accuracy, detail, language simplicity, and response tone

Observations:

Prompt	ChatGPT	Claude	Gemini
Internet definition	Clear and user-friendly	Technical, detailed	Search-like, concise
Cloud uses	Structured and explained	Well-listed, less verbose	Short but updated
Water cycle	Kid-friendly explanation	Clear, less creative	Very concise
Short story	Creative and engaging	Polite but less vivid	Quick and imaginative
Math problem	Accurate and explained	Accurate, brief	Accurate, shortest

Result:
All models gave correct and relevant responses to basic prompts. ChatGPT was most balanced in clarity and creativity, Claude was formal and structured, and Gemini was concise and search-like. The experiment demonstrated the importance of prompt design and model selection for effective AI interaction.

Experiment 43

Research and Report on Popular IT Certifications

Aim:

To research widely recognized IT certifications and understand their benefits, issuing organizations, skill focus, and career impact.

Steps:

IT certifications validate an individual's knowledge and skills in specific areas such as networking, cloud computing, cybersecurity, programming, and support. These certifications are offered by leading tech companies and organizations.

🔹 Steps Performed:

1. Researched online about top IT certifications using official websites and tech job portals
2. Collected data on:
    a. Certification name
    b. Issuing body
    c. Skill area
    d. Recommended level (Beginner/Intermediate/Expert)
    e. Career relevance
3. Compiled findings into a summary table

📄 Popular IT Certifications:

Certification	Issuing Body	Skill Focus	Level	Career Benefit
CompTIA A+	CompTIA	IT support, hardware	Beginner	Entry-level IT technician jobs
Cisco CCNA	Cisco	Networking basics	Intermediate	Network engineer, admin roles
Microsoft Azure Fundamentals	Microsoft	Cloud services (Azure)	Beginner	Cloud career starter
AWS Certified Cloud Practitioner	Amazon (AWS)	Cloud concepts (AWS)	Beginner	Entry to cloud computing jobs
Google IT Support	Google/Coursera	IT support skills	Beginner	Helpdesk, support technician
Certified Ethical Hacker (CEH)	EC-Council	Cybersecurity, hacking	Advanced	Cybersecurity analyst, penetration tester
Python Institute PCAP	Python Institute	Programming in Python	Beginner	Developer, scripting, automation roles

Result:

Several popular IT certifications were studied, covering support, networking, cloud, programming, and cybersecurity. Certifications like AWS Cloud Practitioner, CompTIA A+, and Google IT Support are ideal starting points for students in IT and Computer Science.