

DIGITAL TECHNOLOGY LESSON NOTES

Class: JSS3 | First Term |

WEEK 1: REVISION OF JSS2 WORK

Duration: 40 minutes

Topic: Recall Networking, Databases, HTML, and Cybersecurity Basics

INTRODUCTION (5 minutes)

Welcome students to their final year of junior secondary education. Explain that this revision will reinforce JSS2 concepts and prepare them for more advanced topics in JSS3.

STEP 1: TOPIC (2 minutes)

Main Topic: Comprehensive Review of JSS2 Digital Technology Foundations **Sub-topics:**

- Computer networking fundamentals and connectivity
- Database concepts and data organization
- HTML structure and basic web development
- Cybersecurity principles and protective measures
- Integration of learned concepts for advanced applications

STEP 2: EXPLANATION (15 minutes)

Networking Basics Review:

- **Network Definition:** Two or more connected computers sharing resources and information
- **Network Types:**
 - **LAN (Local Area Network):** Computers in same building or campus
 - **WAN (Wide Area Network):** Networks covering large geographical areas
 - **Internet:** Global network connecting millions of devices worldwide
- **Network Components:** Cables, wireless signals, routers, switches, modems
- **Data Transmission Methods:** Wired (Ethernet, fiber optic) vs Wireless (Wi-Fi, Bluetooth)
- **Network Advantages:** Resource sharing, communication, centralized data storage, cost efficiency

Database Concepts Recap:

- **Database:** Organized collection of related information stored electronically
- **Key Components:**
 - **Tables:** Store data in rows and columns
 - **Records:** Individual entries (rows) containing complete information sets
 - **Fields:** Individual data categories (columns) like Name, Age, Address
- **Database Examples:** Student records, library catalogs, customer information, inventory systems
- **Database Benefits:** Efficient storage, quick searching, data relationships, reduced redundancy
- **Simple Operations:** Adding records, editing information, deleting entries, searching data

HTML Foundation Review:

- **HTML Purpose:** Markup language for creating web page structure and content
- **Basic Document Structure:** <!DOCTYPE html>, <html>, <head>, <body> tags
- **Essential Tags Recap:**
 - **Headings:** <h1> to <h6> for different heading levels
 - **Paragraphs:** <p> for text content organization
 - **Images:** with src and alt attributes
 - **Line Breaks:**
 and horizontal rules <hr>
- **Attributes:** Additional information providing specific instructions to tags
- **File Organization:** Proper naming conventions, folder structures for web projects

Cybersecurity Fundamentals Review:

- **Digital Threats Understanding:**
 - **Malware:** Viruses, worms, trojans designed to damage or steal data
 - **Phishing:** Fake websites or emails trying to steal personal information
 - **Identity Theft:** Unauthorized use of personal information for fraudulent purposes
- **Protection Strategies:**
 - **Strong Passwords:** Complex combinations of letters, numbers, symbols
 - **Antivirus Software:** Programs detecting and removing malicious software
 - **Safe Browsing:** Verifying website security, avoiding suspicious downloads
 - **Privacy Settings:** Controlling who can access personal information online
- **Digital Citizenship:** Responsible, ethical behavior in digital environments

STEP 3: COMPARISON (8 minutes)

Standalone vs Networked Computers:

- **Standalone:** Limited to local resources, no communication capability
- **Networked:** Shared resources, communication, collaborative work, internet access

Traditional vs Digital Data Storage:

- **Traditional:** Paper files, physical storage, manual organization, limited search
- **Digital Databases:** Electronic storage, automated organization, powerful search, data relationships

Static vs Dynamic Web Content:

- **HTML Pages:** Fixed content, basic structure, foundational understanding
- **Advanced Web:** Interactive elements, database integration, user input processing

Reactive vs Proactive Security:

- **Reactive:** Responding to threats after they occur
- **Proactive:** Preventing threats through education, protective software, safe practices

STEP 4: CONCLUSION (5 minutes)

Students should demonstrate solid understanding of networking principles, database organization, HTML structure, and cybersecurity awareness, providing the foundation needed for advanced JSS3 topics including programming, advanced web development, and complex database operations.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the difference between LAN and WAN networks with examples
2. Define database, record, and field, giving a practical example of each
3. Write the basic HTML structure for a simple webpage with a heading and paragraph
4. List five cybersecurity best practices for safe internet use
5. How do networking, databases, HTML, and cybersecurity work together in modern web applications?

WEEK 2: COMPUTER HARDWARE MAINTENANCE

Duration: 40 minutes

Topic: Explain Maintenance Practices; Cleaning and Safe Use of Devices

INTRODUCTION (5 minutes)

Explain that proper hardware maintenance extends computer lifespan, improves performance, prevents costly repairs, and ensures safe operation for users.

STEP 1: TOPIC (2 minutes)

Main Topic: Computer Hardware Care and Maintenance Procedures **Sub-topics:**

- Preventive maintenance principles and scheduling
- Physical cleaning techniques and safety procedures
- Environmental factors affecting hardware longevity
- Safe handling and operation practices
- Troubleshooting common maintenance issues

STEP 2: EXPLANATION (15 minutes)

Preventive Maintenance Principles:

- **Definition:** Regular care activities performed to prevent hardware problems before they occur
- **Maintenance Schedule:**
 - **Daily:** Proper shutdown procedures, checking for unusual sounds or smells
 - **Weekly:** External cleaning, checking cable connections
 - **Monthly:** Internal cleaning, software updates, disk cleanup
 - **Quarterly:** Deep cleaning, hardware inspection, backup verification
- **Benefits:** Extended hardware life, improved performance, reduced repair costs, fewer unexpected failures
- **Documentation:** Keeping maintenance logs, recording issues, tracking performance changes

Physical Cleaning Procedures: External Cleaning:

- **Screen/Monitor:** Use microfiber cloth with appropriate screen cleaner, avoid harsh chemicals
- **Keyboard:** Turn upside down to remove debris, use compressed air between keys, disinfect with alcohol wipes
- **Mouse:** Clean sensor area, wipe surface with disinfectant, check for smooth movement
- **Exterior Surfaces:** Wipe with slightly damp cloth, avoid getting moisture into ports or openings

Internal Cleaning (Advanced/Supervised):

- **Dust Removal:** Use compressed air to blow dust from fans, heat sinks, and components
- **Fan Maintenance:** Ensure fans rotate freely, replace if making unusual noise
- **Ventilation:** Keep air vents clear, ensure proper airflow around computer
- **Cable Management:** Organize cables to prevent airflow restriction, secure loose connections

Environmental Factors:

- **Temperature Control:** Keep computers in cool, well-ventilated areas (ideal: 18-24°C)

- **Humidity Management:** Avoid extremely dry or humid conditions (ideal: 30-50% humidity)
- **Dust Prevention:** Use dust covers when not in use, keep area clean, avoid carpet placement
- **Power Protection:** Use surge protectors, UPS for power outages, avoid overloaded circuits
- **Placement Considerations:** Stable surfaces, away from heat sources, adequate space around vents

Safe Handling Practices: Power Safety:

- **Proper Shutdown:** Use operating system shutdown procedures, never force power off
- **Power Connection:** Ensure secure connections, avoid damaged cables, use appropriate voltage
- **Static Electricity:** Ground yourself before touching internal components, use anti-static wrist straps

Physical Handling:

- **Lifting Techniques:** Lift with legs not back, get help for heavy equipment
- **Transportation:** Use original packaging when possible, secure all components
- **Component Installation:** Handle by edges, avoid touching circuits, ensure proper seating

Operational Safety:

- **Ventilation:** Never block air vents, ensure adequate space for airflow
- **Liquid Protection:** Keep drinks away from computers, use spill-resistant containers
- **Electrical Safety:** Check cables regularly, replace damaged cords immediately

Common Maintenance Tasks:

- **Disk Cleanup:** Remove temporary files, empty recycle bin, uninstall unused programs
- **Defragmentation:** Reorganize hard drive data for improved performance (HDD only)
- **Software Updates:** Keep operating system and drivers current for optimal performance
- **Antivirus Scanning:** Regular system scans to detect and remove threats
- **Backup Procedures:** Regular data backup to prevent loss from hardware failure

STEP 3: COMPARISON (8 minutes)

Preventive vs Corrective Maintenance:

- **Preventive:** Regular scheduled care, prevents problems, lower cost, planned activities
- **Corrective:** Fix problems after they occur, higher cost, unexpected downtime, emergency repairs

Professional vs User Maintenance:

- **User Level:** External cleaning, basic software maintenance, environmental care
- **Professional Level:** Internal repairs, component replacement, advanced diagnostics

Different Hardware Requirements:

- **Desktop Computers:** More accessible for cleaning, better ventilation options
- **Laptops:** More compact, requires careful handling, limited user maintenance
- **Mobile Devices:** Primarily external care, professional service for internal issues

Maintenance Frequency:

- **High-Use Environments:** More frequent cleaning, stricter environmental controls
- **Low-Use Environments:** Standard maintenance schedules, basic care procedures

STEP 4: CONCLUSION (5 minutes)

Students should understand that proper hardware maintenance involves regular cleaning, environmental control, safe handling practices, and preventive care that extends equipment life and ensures reliable operation.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Create a weekly maintenance schedule for a school computer lab
2. Explain the proper procedure for cleaning a computer keyboard safely
3. List five environmental factors that can damage computer hardware
4. What is the difference between preventive and corrective maintenance?
5. Describe three safety precautions to follow when handling computer components

WEEK 3: SYSTEM TROUBLESHOOTING

Duration: 40 minutes

Topic: Identify Common Faults in Computers; Demonstrate Simple Solutions

INTRODUCTION (5 minutes)

Explain that troubleshooting is a systematic approach to identifying and solving computer problems, an essential skill for anyone working with technology.

STEP 1: TOPIC (2 minutes)

Main Topic: Computer Problem Diagnosis and Resolution Techniques **Sub-topics:**

- Common hardware and software problems
- Systematic troubleshooting methodology
- Basic diagnostic tools and techniques
- Simple repair procedures and solutions
- When to seek professional assistance

STEP 2: EXPLANATION (15 minutes)

Systematic Troubleshooting Approach:

- **Step 1: Problem Identification:** Clearly define what is not working correctly
- **Step 2: Information Gathering:** When did problem start? What changed recently?
- **Step 3: Hypothesis Formation:** Develop possible explanations for the problem
- **Step 4: Testing Solutions:** Try simplest solutions first, test one change at a time
- **Step 5: Documentation:** Record what worked, what didn't, for future reference
- **Step 6: Verification:** Ensure problem is completely resolved, not just temporarily fixed

Common Hardware Problems and Solutions:

Computer Won't Start:

- **Symptoms:** No power, no lights, no sounds when power button pressed
- **Simple Solutions:**
 - Check power cable connections to wall and computer
 - Verify power switch on power supply (desktop computers)
 - Try different power outlet or power cable
 - Check for loose internal connections (if comfortable doing so)
- **Advanced Checks:** Power supply testing, motherboard issues (professional help needed)

Overheating Issues:

- **Symptoms:** Computer shuts down unexpectedly, very hot to touch, loud fan noise
- **Simple Solutions:**
 - Clean dust from vents and fans using compressed air
 - Ensure adequate ventilation around computer
 - Check that all fans are working properly
 - Close resource-intensive programs
- **Prevention:** Regular cleaning, proper placement, monitoring temperatures

Peripheral Device Problems:

- **Symptoms:** Keyboard, mouse, or printer not responding
- **Simple Solutions:**
 - Check cable connections and USB ports

- Try different USB port or cable
 - Restart computer to refresh device connections
 - Update or reinstall device drivers
- **Advanced:** Device manager troubleshooting, hardware replacement

Common Software Problems and Solutions:

Slow Performance:

- **Symptoms:** Programs take long time to open, sluggish response, freezing
- **Simple Solutions:**
 - Close unnecessary programs and browser tabs
 - Restart computer to clear memory
 - Run disk cleanup to free storage space
 - Check for malware using antivirus software
 - Disable startup programs not needed immediately
- **Advanced:** Add more RAM, upgrade to SSD, reinstall operating system

Application Crashes:

- **Symptoms:** Programs close unexpectedly, error messages, frozen applications
- **Simple Solutions:**
 - Update the problematic application to latest version
 - Restart the application and try again
 - Restart computer to clear memory issues
 - Check if problem occurs with specific files or actions
- **Advanced:** Reinstall application, check system logs, hardware diagnostics

Internet Connectivity Issues:

- **Symptoms:** Cannot access websites, slow browsing, connection drops frequently
- **Simple Solutions:**
 - Check network cable or Wi-Fi connection status
 - Restart modem and router (unplug for 30 seconds)
 - Try connecting different device to test network
 - Run built-in network troubleshooter
 - Check with internet service provider for outages

Basic Diagnostic Tools:

- **Built-in Windows Tools:** Device Manager, System Information, Event Viewer, Network Troubleshooter
- **Third-party Tools:** Antivirus scanners, disk cleanup utilities, system monitoring software
- **Hardware Tests:** Memory diagnostic, hard drive health check, temperature monitoring
- **Network Tools:** Ping command, speed tests, connection status indicators

STEP 3: COMPARISON (8 minutes)

Hardware vs Software Problems:

- **Hardware:** Physical component failures, often consistent symptoms, may require replacement
- **Software:** Program conflicts, corrupted files, often intermittent, usually fixable with software solutions

Simple vs Complex Problems:

- **Simple:** User-fixable with basic knowledge and tools (cable connections, restarts, cleaning)
- **Complex:** Require professional diagnosis, specialized tools, or component replacement

Temporary vs Permanent Solutions:

- **Temporary:** Quick fixes that may not address root cause (restart, close programs)
- **Permanent:** Address underlying problem (driver updates, hardware replacement, system reinstall)

User-Level vs Professional Repairs:

- **User-Level:** Safe procedures, external connections, basic software troubleshooting
- **Professional:** Internal hardware work, advanced diagnostics, warranty considerations

STEP 4: CONCLUSION (5 minutes)

Students should understand systematic troubleshooting methodology, be able to identify and resolve common computer problems using simple solutions, and recognize when professional assistance is needed for complex issues.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Describe the six steps of systematic troubleshooting methodology
2. What are three simple solutions to try when a computer won't start?
3. List five symptoms that might indicate a computer is overheating
4. How would you troubleshoot a mouse that suddenly stops working?
5. When should you seek professional help instead of attempting repairs yourself?

WEEK 4: ADVANCED WORD PROCESSING

Duration: 40 minutes

Topic: Create Reports with Table of Contents, References, and Page Layouts

INTRODUCTION (5 minutes)

Explain that advanced word processing skills are essential for creating professional academic and business documents, including research papers, reports, and formal publications.

STEP 1: TOPIC (2 minutes)

Main Topic: Professional Document Creation and Advanced Formatting **Sub-topics:**

- Table of contents generation and management
- Advanced reference and citation systems
- Professional page layout and design principles
- Document structure and organization techniques
- Collaborative editing and review processes

STEP 2: EXPLANATION (15 minutes)

Table of Contents Creation:

- **Purpose:** Provides document navigation, shows document structure, enables quick access to sections
- **Automatic Generation Process:**
 1. **Apply Heading Styles:** Use built-in heading styles (Heading 1, Heading 2, Heading 3) consistently
 2. **Insert Table of Contents:** References tab → Table of Contents → Choose style
 3. **Update Automatically:** Right-click TOC → Update Field when document changes
- **Heading Style Hierarchy:**
 - **Heading 1:** Main chapters or major sections
 - **Heading 2:** Sub-sections within chapters
 - **Heading 3:** Sub-sub-sections for detailed breakdown
- **Customization Options:** Modify styles for consistent formatting, adjust tab leaders, set page number alignment
- **Navigation Benefits:** Ctrl+Click to jump to sections, print TOC for document overview

Advanced Reference Systems: Citation Management:

- **In-Text Citations:** Brief source references within document text
- **Bibliography Creation:** Complete source list automatically generated from citations
- **Citation Styles:** APA (psychology, education), MLA (literature, humanities), Chicago (history, arts)
- **Source Types:** Books, journal articles, websites, interviews, government documents

Reference Insertion Process:

1. **References Tab:** Insert Citation → Add New Source
2. **Source Information:** Fill complete details (author, title, publisher, date)
3. **Citation Placement:** Insert at appropriate points in text
4. **Bibliography Generation:** Insert Bibliography at document end
5. **Style Management:** Change citation style affects entire document automatically

Footnotes and Endnotes:

- **Footnotes:** Additional information at bottom of each page
- **Endnotes:** Collected at document end for comprehensive reference
- **Insertion:** References → Insert Footnote/Endnote
- **Automatic Numbering:** Sequential numbering updates automatically when notes added/removed

Professional Page Layout: Page Setup and Margins:

- **Standard Margins:** 1 inch on all sides for academic papers
- **Custom Margins:** Adjust for binding requirements or specific formats
- **Orientation:** Portrait (standard) vs Landscape (wide tables, charts)
- **Paper Size:** A4 (international standard), Letter (US standard)

Section Breaks and Page Layout:

- **Section Breaks:** Allow different formatting in same document
- **Page Breaks:** Control where pages end for better organization
- **Column Layout:** Multi-column text for newsletter or academic formats
- **Headers/Footers by Section:** Different headers for chapters or sections

Advanced Formatting Features:

- **Drop Caps:** Large first letter for chapter beginnings
- **Text Boxes:** Callout information, sidebar content
- **Watermarks:** Draft stamps, confidential markings, background images
- **Page Borders:** Decorative elements for certificates or formal documents

Document Structure Best Practices: Professional Document Elements:

- **Title Page:** Document title, author information, date, institutional affiliation
- **Executive Summary/Abstract:** Brief overview of document contents and conclusions
- **Table of Contents:** Navigation aid for readers
- **Main Body:** Organized with clear headings and logical flow
- **Conclusion:** Summary of key points and recommendations
- **References/Bibliography:** Complete source documentation
- **Appendices:** Supporting materials, detailed data, additional information

Collaborative Features:

- **Track Changes:** Monitor edits from multiple authors
- **Comments:** Add feedback without changing document text
- **Version Control:** Compare documents, accept/reject changes
- **Sharing Options:** Cloud collaboration, email review, permission settings

STEP 3: COMPARISON (8 minutes)

Manual vs Automatic Features:

- **Manual TOC:** Time-consuming, error-prone, requires manual updates
- **Automatic TOC:** Generated from styles, updates automatically, consistent formatting

Simple vs Professional Documents:

- **Simple:** Basic formatting, manual page numbers, limited structure
- **Professional:** Automatic features, consistent styling, comprehensive references

Academic vs Business Documents:

- **Academic:** Strict citation requirements, formal structure, research focus
- **Business:** Practical information, executive summaries, action-oriented content

Individual vs Collaborative Work:

- **Individual:** Simple document creation, basic editing tools
- **Collaborative:** Version control, change tracking, comment management, shared access

STEP 4: CONCLUSION (5 minutes)

Students should understand how to create professional documents using advanced word processing features including automatic table of contents, proper reference systems, and sophisticated page layouts suitable for academic and business environments.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the process of creating an automatic table of contents using heading styles
2. What is the difference between footnotes and endnotes? When would you use each?
3. List the essential elements of a professional business report in correct order
4. How do you insert and manage citations in different academic styles (APA, MLA)?
5. Create a document outline for a school project report, including all advanced formatting elements discussed

WEEK 5: SPREADSHEET ADVANCED FUNCTIONS

Duration: 40 minutes

Topic: Use IF Statements, Conditional Formatting, and Nested Functions

INTRODUCTION (5 minutes)

Explain that advanced spreadsheet functions transform simple data tables into powerful analytical tools capable of complex decision-making and automated data analysis.

STEP 1: TOPIC (2 minutes)

Main Topic: Complex Spreadsheet Analysis and Automated Decision Making **Sub-topics:**

- IF statement logic and practical applications
- Conditional formatting for visual data analysis
- Nested functions for complex calculations
- Advanced data validation and error handling
- Real-world business and academic applications

STEP 2: EXPLANATION (15 minutes)

IF Statement Fundamentals:

- **Purpose:** Make decisions based on conditions, automate analysis, categorize data
- **Basic Syntax:** =IF(condition, value_if_true, value_if_false)
- **Logical Operators:**
 - **Equal to:** = (e.g., A1=100)
 - **Greater than:** > (e.g., B1>75)
 - **Less than:** < (e.g., C1<50)
 - **Greater/equal:** >= (e.g., D1>=60)
 - **Less/equal:** <= (e.g., E1<=100)
 - **Not equal:** <> (e.g., F1<>"")

Practical IF Statement Examples:

- **Grade Classification:** =IF(A1>=70,"Pass","Fail")
- **Sales Commission:** =IF(B1>10000,B10.15,B10.10)
- **Attendance Status:** =IF(C1>=75%,"Regular","Irregular")
- **Budget Analysis:** =IF(D1>E1,"Over Budget","Within Budget")
- **Inventory Management:** =IF(F1<50,"Reorder","Sufficient")

Advanced IF Statements: Multiple Conditions (Nested IF):

- **Grade Levels:** =IF(A1>=90,"A",IF(A1>=80,"B",IF(A1>=70,"C",IF(A1>=60,"D","F"))))
- **Salary Ranges:**
=IF(B1<30000,"Entry",IF(B1<50000,"Mid",IF(B1<80000,"Senior","Executive")))

Logical Functions:

- **AND Function:** =IF(AND(A1>60,B1>60),"Both Pass","At Least One Fail")
- **OR Function:** =IF(OR(C1="Excellent",C1="Good"),"Satisfactory","Needs Improvement")
- **NOT Function:** =IF(NOT(D1="Absent"),"Present","Absent")

Conditional Formatting:

- **Purpose:** Visual highlighting of data based on values, trends, or conditions
- **Application Process:**
 1. Select data range to format
 2. Home tab → Conditional Formatting
 3. Choose rule type (greater than, between, top 10, etc.)
 4. Set conditions and formatting (colors, fonts, borders)

Conditional Formatting Types:

- **Highlight Cell Rules:** Values greater than, less than, between, equal to specific values
- **Top/Bottom Rules:** Highest values, lowest values, above/below average
- **Data Bars:** Horizontal bars showing relative values within cells
- **Color Scales:** Color gradients from lowest to highest values
- **Icon Sets:** Traffic lights, arrows, stars to show performance levels

Practical Formatting Examples:

- **Student Grades:** Green for A grades, yellow for B-C, red for D-F
- **Sales Performance:** Data bars showing monthly sales relative to targets
- **Budget Variance:** Red for over-budget, green for under-budget items
- **Project Status:** Icons showing on-time, at-risk, or delayed projects

Nested Functions Applications: Complex Business Calculations:

- **Commission Structure:** =IF(A1>50000,A1*0.20,IF(A1>25000,A1*0.15,A1*0.10))
- **Shipping Costs:** =IF(B1>100,0,IF(B1>50,5,10))
- **Performance Ratings:**
=IF(AND(C1>=8,D1>=85),"Excellent",IF(OR(C1>=6,D1>=70),"Good","Needs Improvement"))

Academic Applications:

- **Scholarship Eligibility:** =IF(AND(GPA>=3.5,Income<50000),"Eligible","Not Eligible")

- **Course Prerequisites:** =IF(AND(Math>=70,English>=65),"Can Enroll","Missing Prerequisites")

Data Validation and Error Handling:

- **IFERROR Function:** =IFERROR(A1/B1,"Cannot divide by zero")
- **Data Validation:** Restrict input to specific ranges, lists, or formats
- **Error Prevention:** Check for common calculation errors before they occur

STEP 3: COMPARISON (8 minutes)

Simple vs Complex Formulas:

- **Simple:** Basic calculations (+, -, *, /, SUM, AVERAGE)
- **Complex:** IF statements, nested functions, multiple conditions, logical operations

Manual vs Automated Analysis:

- **Manual:** Review each data point individually, prone to errors, time-consuming
- **Automated:** IF statements and formatting highlight important data automatically

Static vs Dynamic Formatting:

- **Static:** Fixed colors and formats that don't change with data
- **Dynamic:** Conditional formatting that updates automatically as data changes

Single vs Multiple Criteria:

- **Single:** One condition determines outcome (IF statement)
- **Multiple:** AND/OR functions combine several conditions for complex decisions

STEP 4: CONCLUSION (5 minutes)

Students should understand how to use IF statements for automated decision-making, apply conditional formatting for visual data analysis, and create nested functions for complex business and academic calculations.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Write an IF statement that assigns letter grades based on numerical scores (A: 90+, B: 80-89, C: 70-79, D: 60-69, F: below 60)
2. Explain how conditional formatting can help identify trends in sales data
3. Create a nested IF statement for employee bonus calculation based on years of service and performance rating

4. What is the difference between AND and OR logical functions? Give examples
 5. Design a spreadsheet system for tracking student attendance with automatic status updates and visual indicators
-

WEEK 6: DATABASE QUERIES

Duration: 40 minutes

Topic: Explain and Run Simple Queries in MS Access or Excel

INTRODUCTION (5 minutes)

Explain that database queries are powerful tools for extracting specific information from large datasets, enabling users to find answers to questions and generate reports efficiently.

STEP 1: TOPIC (2 minutes)

Main Topic: Database Information Retrieval and Data Analysis Techniques **Sub-topics:**

- Query concepts and purposes in database management
- Simple query creation in MS Access
- Excel database functions and filtering techniques
- Query criteria and result optimization
- Practical applications in academic and business contexts

STEP 2: EXPLANATION (15 minutes)

Database Query Fundamentals:

- **Definition:** Structured requests for specific information from database tables
- **Purpose:** Extract relevant data, answer specific questions, generate reports, analyze trends
- **Query Types:**
 - **Select Queries:** Retrieve specific records meeting certain criteria
 - **Parameter Queries:** Ask for input values when query runs
 - **Calculated Queries:** Perform calculations on retrieved data
 - **Summary Queries:** Group data and calculate totals, averages, counts

MS Access Query Operations:

Simple Select Queries:

- **Query Design View:** Visual interface for building queries without coding
- **Steps to Create:**

1. Create → Query Design
2. Select table(s) containing needed data
3. Drag fields from table to query grid
4. Add criteria in Criteria row
5. Run query to see results

Query Criteria Examples:

- **Exact Match:** "Lagos" (finds records exactly matching Lagos)
- **Numerical Comparison:** >18 (finds records greater than 18)
- **Date Ranges:** Between #1/1/2024# And #12/31/2024#
- **Text Patterns:** Like "Smith*" (finds names starting with Smith)
- **Multiple Criteria:** Use AND (both conditions) or OR (either condition)

Practical Access Query Examples:

- **Student Records:** Find all students with GPA > 3.5
- **Inventory Management:** Show products with quantity < 50
- **Sales Analysis:** Display sales between specific dates
- **Employee Database:** List employees in specific department

Parameter Queries:

- **Interactive Queries:** Prompt user for input when query runs
- **Syntax:** [Enter Student Name] in criteria field
- **Benefits:** Reusable queries, flexible searching, user-friendly interface

Excel Database Functions:

Data Filtering:

- **AutoFilter:** Data tab → Filter → Click dropdown arrows on column headers
- **Custom Filters:** Text filters, number filters, date filters with specific criteria
- **Advanced Filter:** More complex criteria using separate criteria range
- **Filter Benefits:** Hide irrelevant data, focus on specific records, maintain original data integrity

Excel Database Functions:

- **COUNTIF:** Count cells meeting specific criteria
 - Example: =COUNTIF(A:A,">50") counts values greater than 50
- **SUMIF:** Sum values meeting specific criteria
 - Example: =SUMIF(B:B,"Electronics",C:C) sums sales for Electronics category
- **AVERAGEIF:** Calculate average of values meeting criteria
 - Example: =AVERAGEIF(D:D,">=70") averages scores 70 and above

Advanced Excel Queries:

- **VLOOKUP:** Find information in table based on lookup value
 - Example: =VLOOKUP(student_id,student_table,grade_column,FALSE)
- **Pivot Tables:** Summarize and analyze large datasets interactively
- **Data Validation:** Ensure data entry meets specific criteria

Query Optimization Techniques:

- **Select Necessary Fields Only:** Improves performance, reduces result complexity
- **Use Specific Criteria:** Narrow results to relevant information
- **Sort Results:** Organize output for easier analysis
- **Index Important Fields:** Speeds up query performance (Access)

Real-World Query Applications:

Academic Examples:

- **Student Performance:** Find students needing academic support
- **Course Analysis:** Identify most/least popular courses
- **Attendance Tracking:** Students with attendance below threshold
- **Grade Distribution:** Statistical analysis of class performance

Business Examples:

- **Customer Analysis:** Find high-value customers for special promotions
- **Inventory Reports:** Items needing reorder or overstocked items
- **Sales Performance:** Top-performing products or sales representatives
- **Financial Analysis:** Expenses exceeding budget categories

Report Generation:

- **Query Results as Reports:** Use query output for formatted reports
- **Regular Updates:** Queries refresh automatically with new data
- **Data Export:** Save query results to Excel, PDF, or other formats
- **Automated Reporting:** Schedule queries to run at specific intervals

STEP 3: COMPARISON (8 minutes)

Access vs Excel for Database Operations:

- **Access:** Purpose-built for databases, complex relationships, better for large datasets
- **Excel:** Familiar interface, good for smaller datasets, better calculation features

Simple vs Complex Queries:

- **Simple:** Single table, basic criteria, straightforward results
- **Complex:** Multiple tables, calculated fields, grouped results, parameter input

Manual vs Query-Based Data Analysis:

- **Manual:** Time-consuming, error-prone, difficult with large datasets
- **Query-Based:** Fast, accurate, repeatable, handles large amounts of data efficiently

Static vs Dynamic Queries:

- **Static:** Fixed criteria, same results each time
- **Dynamic:** Parameter queries, criteria can change, interactive user input

STEP 4: CONCLUSION (5 minutes)

Students should understand database query concepts, be able to create simple queries in both MS Access and Excel, and recognize how queries can efficiently extract meaningful information from large datasets for decision-making.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the difference between a query and a table in database terminology
2. Write the steps to create a simple query in MS Access that finds all students with grades above 85
3. What Excel function would you use to count how many products have a price greater than ₦5,000?
4. Compare the advantages of using queries versus manual data searching
5. Design a query scenario for a school library database that would help librarians track overdue books

GENERAL ASSESSMENT FOR WEEKS 1-6

Practical Activities:

1. **System Maintenance Lab:** Perform complete computer cleaning and maintenance checklist
2. **Troubleshooting Simulation:** Diagnose and solve common hardware/software problems
3. **Professional Document Creation:** Create multi-page report with TOC, references, and advanced formatting
4. **Advanced Spreadsheet Project:** Build grade book with IF statements and conditional formatting

5. **Database Query Exercise:** Create and run queries to extract specific information from sample databases
6. **Integration Project:** Combine word processing, spreadsheet, and database skills in comprehensive assignment

Cumulative Project Assignment: "Digital Solutions for School Management" Students will create a comprehensive project including:

- **Database Component:** Student information system with query capabilities
- **Spreadsheet Analysis:** Grade calculations using advanced functions and conditional formatting
- **Professional Report:** Document the entire system with proper formatting, TOC, and references
- **Maintenance Guide:** Computer care procedures and troubleshooting manual
- **Presentation:** Demonstrate all components and explain design decisions

Assessment Criteria:

- **Technical Proficiency (40%):** Correct use of advanced features and functions
- **Problem-Solving (30%):** Effective troubleshooting and system design
- **Professional Presentation (20%):** Document formatting and project organization
- **Integration Skills (10%):** Seamless combination of different software tools

Learning Outcomes Achieved: By the end of Week 6, students should demonstrate:

- Comprehensive understanding of JSS2 foundation concepts
- Competent computer maintenance and troubleshooting skills
- Advanced word processing capabilities for professional documents
- Complex spreadsheet analysis using logical functions and formatting
- Database query skills for information extraction and analysis
- Integration of multiple software applications for comprehensive solutions

WEEK 7: INTRODUCTION TO PYTHON (TEXT-BASED PROGRAMMING)

Duration: 40 minutes

Topic: Write Simple Python Commands (Print, Input)

INTRODUCTION (5 minutes)

Introduce Python as a powerful, beginner-friendly programming language used worldwide for web development, data analysis, artificial intelligence, and automation.

STEP 1: TOPIC (2 minutes)

Main Topic: Fundamentals of Python Programming and Basic Commands Sub-topics:

- Python language overview and applications
- Programming environment setup and interface
- Basic syntax rules and command structure
- Print statements for output display
- Input commands for user interaction

STEP 2: EXPLANATION (15 minutes)

Python Programming Fundamentals:

- **What is Python:** High-level programming language created by Guido van Rossum in 1991
- **Why Python is Popular:**
 - Easy to read and write (English-like syntax)
 - Versatile applications (websites, games, data analysis, AI)
 - Large community support and extensive libraries
 - Free and open-source
- **Real-World Applications:** Google, YouTube, Instagram, Netflix use Python extensively
- **Career Relevance:** High demand for Python programmers in technology industry

Programming Environment Setup:

- **Python Installation:** Download from python.org or use online platforms
- **Code Editors:** IDLE (built-in), Visual Studio Code, PyCharm, online editors (repl.it, trinket.io)
- **Interactive Shell:** Test commands immediately, see results instantly
- **Script Files:** Save programs as .py files for reuse and sharing

Basic Python Syntax Rules:

- **Case Sensitivity:** Python distinguishes between uppercase and lowercase (Print vs print)
- **Indentation:** Spaces at beginning of lines show code structure (crucial in Python)
- **Comments:** Use # symbol for explanatory text that doesn't run
- **String Quotes:** Use single (') or double (") quotes for text
- **No Semicolons:** Unlike some languages, Python doesn't require semicolons at line ends

Print Statement Mastery: Basic Print Syntax:

python

```
print("Hello, World!")
```

```
print("Welcome to Python programming")
```

Advanced Print Features:

python

```
# Multiple items in one print
```

```
print("Student Name:", "John", "Age:", 15)
```

```
# Print with variables
```

```
name = "Sarah"
```

```
print("Hello", name)
```

```
# Print with formatting
```

```
print("My favorite number is", 42)
```

```
# Print special characters
```

```
print("Line 1\nLine 2") # \n creates new line
```

```
print("Column1\tColumn2") # \t creates tab space
```

Print Applications:

- **Welcome Messages:** Greet users when programs start
- **Information Display:** Show results, status updates, instructions
- **Debugging:** Display variable values to check program logic
- **User Feedback:** Confirm actions, show progress, display errors

Input Command Fundamentals: Basic Input Syntax:

```
python
```

```
name = input("What is your name? ")
```

```
print("Nice to meet you,", name)
```

Input with Different Data Types:

```
python
```

```
# Text input (default)
```

```
favorite_color = input("What's your favorite color? ")
```

```
# Number input (requires conversion)
```

```
age = int(input("How old are you? "))
```

```
height = float(input("What's your height in meters? "))
```

```
# Using input in calculations
```

```
num1 = int(input("Enter first number: "))
```

```
num2 = int(input("Enter second number: "))
```

```
total = num1 + num2
```

```
print("The sum is:", total)
```

Interactive Program Examples:

python

Simple greeting program

```
print("Welcome to the Greeting Program!")
name = input("Please enter your name: ")
age = input("Please enter your age: ")
print("Hello", name + "!")
print("You are", age, "years old.")
```

Basic calculator

```
print("Simple Calculator")
first = float(input("Enter first number: "))
second = float(input("Enter second number: "))
result = first + second
print("The answer is:", result)
```

Common Beginner Mistakes:

- **Forgetting Quotes:** print(Hello) instead of print("Hello")
- **Case Errors:** Print instead of print
- **Input Type Issues:** Not converting input to numbers for calculations
- **Syntax Errors:** Missing parentheses, colons, or quotes

STEP 3: COMPARISON (8 minutes)

Python vs Other Programming Languages:

- **Python:** Easy syntax, readable code, slower execution
- **Java:** More complex syntax, faster execution, widely used in enterprise
- **C++:** Very fast, complex syntax, used for system programming

Print vs Input Functions:

- **Print:** Sends information FROM program TO user (output)
- **Input:** Receives information FROM user TO program (input)
- **Interaction:** Programs become dynamic when combining both

Text vs Numeric Input:

- **Text Input:** Default input() function, stores as string

- **Numeric Input:** Requires int() or float() conversion for calculations
- **Mixed Programs:** Combine text and numbers for comprehensive applications

Interactive vs Static Programs:

- **Static:** Same output every time program runs
- **Interactive:** Different results based on user input, more engaging

STEP 4: CONCLUSION (5 minutes)

Students should understand Python programming basics, be able to write simple programs using print and input commands, and recognize how these fundamental skills form the foundation for more complex programming concepts.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Write a Python program that asks for your name and favorite subject, then displays a personalized message
2. What's the difference between print("5 + 3") and print(5 + 3)? Explain the output for each
3. Create a program that asks for two numbers and displays their sum
4. Why do we need to use int() when asking for a user's age if we want to do calculations?
5. List three real-world applications where Python programming is commonly used

WEEK 8: CONTROL STRUCTURES IN PYTHON

Duration: 40 minutes

Topic: Apply If...Else Statements and Loops

INTRODUCTION (5 minutes)

Explain that control structures allow programs to make decisions and repeat actions, transforming simple linear programs into intelligent, adaptive applications.

STEP 1: TOPIC (2 minutes)

Main Topic: Python Decision-Making and Repetition Control Structures **Sub-topics:**

- Conditional statements and logical decision-making
- If, elif, and else statement applications
- Loop structures for repetitive tasks
- Practical applications in problem-solving
- Combining control structures for complex programs

STEP 2: EXPLANATION (15 minutes)

Conditional Statements (If...Else):

Basic If Statement Structure:

python

```
if condition:
    # Code to execute if condition is True
    action1
    action2
```

If...Else Structure:

python

```
if condition:
    # Code if condition is True
    action_when_true
else:
    # Code if condition is False
    action_when_false
```

If...Elif...Else Structure:

python

```
if condition1:
    action1
elif condition2:
    action2
elif condition3:
    action3
else:
    default_action
```

Comparison Operators for Conditions:

- **Equal to:** == (e.g., age == 18)
- **Not equal:** != (e.g., name != "guest")
- **Greater than:** > (e.g., score > 70)
- **Less than:** < (e.g., price < 100)
- **Greater or equal:** >= (e.g., grade >= 60)

- **Less or equal:** <= (e.g., attempts <= 3)

Practical If...Else Examples:

python

Grade classification

```
score = int(input("Enter your test score: "))
```

```
if score >= 90:
```

```
    print("Excellent! Grade A")
```

```
elif score >= 80:
```

```
    print("Very Good! Grade B")
```

```
elif score >= 70:
```

```
    print("Good! Grade C")
```

```
elif score >= 60:
```

```
    print("Fair! Grade D")
```

```
else:
```

```
    print("Needs improvement! Grade F")
```

Simple authentication

```
password = input("Enter password: ")
```

```
if password == "secret123":
```

```
    print("Access granted! Welcome!")
```

```
else:
```

```
    print("Access denied! Wrong password.")
```

Age-appropriate content

```
age = int(input("Enter your age: "))
```

```
if age >= 18:
```

```
    print("You can access all content")
```

```
elif age >= 13:
```

```
    print("You can access teen content")
```

```
else:
```

```
    print("You can access kid-friendly content only")
```

Loop Structures:

For Loops (Definite Repetition):

python

Basic for loop with range

```
for i in range(5):  
    print("Count:", i)
```

For loop with custom range

```
for number in range(1, 11):  
    print("Number:", number)
```

For loop with step

```
for even in range(0, 21, 2):  
    print("Even number:", even)
```

For loop with lists

```
fruits = ["apple", "banana", "orange"]  
for fruit in fruits:  
    print("I like", fruit)
```

While Loops (Indefinite Repetition):

python

Basic while loop

```
count = 1  
while count <= 5:  
    print("Counting:", count)  
    count = count + 1
```

While loop with user input

```
answer = ""  
while answer.lower() != "quit":  
    answer = input("Enter 'quit' to exit or anything else to continue: ")  
    if answer.lower() != "quit":  
        print("You entered:", answer)
```

While loop for validation

```
password = ""  
while password != "correct":  
    password = input("Enter the correct password: ")  
    if password != "correct":
```

```
    print("Wrong password, try again!")
print("Access granted!")
```

Combining Control Structures:

python

Number guessing game

```
import random
```

```
secret_number = random.randint(1, 10)
```

```
attempts = 0
```

```
max_attempts = 3
```

```
while attempts < max_attempts:
```

```
    guess = int(input("Guess a number between 1 and 10: "))
```

```
    attempts = attempts + 1
```

```
if guess == secret_number:
```

```
    print("Congratulations! You guessed correctly!")
```

```
    break
```

```
elif guess < secret_number:
```

```
    print("Too low!")
```

```
else:
```

```
    print("Too high!")
```

```
if attempts == max_attempts:
```

```
    print("Game over! The number was", secret_number)
```

Simple menu system

```
while True:
```

```
    print("\n--- Calculator Menu ---")
```

```
    print("1. Addition")
```

```
    print("2. Subtraction")
```

```
    print("3. Exit")
```

```
choice = input("Choose an option (1-3): ")
```

```
if choice == "1":
```

```
    a = float(input("First number: "))
```

```

    b = float(input("Second number: "))
    print("Result:", a + b)
elif choice == "2":
    a = float(input("First number: "))
    b = float(input("Second number: "))
    print("Result:", a - b)
elif choice == "3":
    print("Goodbye!")
    break
else:
    print("Invalid choice! Please try again.")

```

Logical Operators for Complex Conditions:

python

AND operator

```

age = int(input("Enter age: "))
has_license = input("Do you have a license? (yes/no): ")
if age >= 18 and has_license.lower() == "yes":
    print("You can drive!")
else:
    print("You cannot drive.")

```

OR operator

```

day = input("What day is it? ").lower()
if day == "saturday" or day == "sunday":
    print("It's weekend!")
else:
    print("It's a weekday.")

```

NOT operator

```

logged_in = False
if not logged_in:
    print("Please log in to continue")
else:
    print("Welcome back!")

```

STEP 3: COMPARISON (8 minutes)

If vs If...Else vs If...Elif...Else:

- **If only:** Single condition, may or may not execute action
- **If...Else:** Two possibilities, always executes one action
- **If...Elif...Else:** Multiple conditions, handles various scenarios

For vs While Loops:

- **For loops:** Known number of repetitions, definite iteration
- **While loops:** Unknown number of repetitions, continues until condition false

Linear vs Control Structure Programs:

- **Linear:** Executes every line in order, same result every time
- **Control Structures:** Can skip lines, repeat sections, different outcomes based on conditions

Simple vs Complex Conditions:

- **Simple:** Single comparison (age > 18)
- **Complex:** Multiple conditions with AND/OR (age > 18 and has_license == True)

STEP 4: CONCLUSION (5 minutes)

Students should understand how to use conditional statements for decision-making and loops for repetition, enabling them to create interactive, intelligent programs that respond to different situations and user inputs.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Write a Python program that asks for a student's age and determines if they're eligible for junior (13-15) or senior (16-18) secondary school
2. Create a while loop that counts from 1 to 10 and displays each number
3. What's the difference between a for loop and a while loop? When would you use each?
4. Write an if...elif...else statement that converts numerical grades (0-100) to letter grades (A, B, C, D, F)
5. Create a simple program that asks users to guess a number between 1 and 5, giving them 3 attempts

WEEK 9: WEB DESIGN WITH HTML & CSS

Duration: 40 minutes

Topic: Build a Simple Webpage Using Basic HTML with Inline CSS Styles

INTRODUCTION (5 minutes)

Explain that combining HTML structure with CSS styling creates visually appealing, professional websites, moving beyond basic text to engaging, formatted web content.

STEP 1: TOPIC (2 minutes)

Main Topic: Integrated Web Development Using HTML Structure and CSS Styling **Sub-topics:**

- HTML review and advanced structure elements
- CSS fundamentals and styling properties
- Inline CSS implementation and syntax
- Color schemes and typography for web design
- Creating cohesive, professional webpage layouts

STEP 2: EXPLANATION (15 minutes)

HTML Structure Review and Enhancement:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First Styled Webpage</title>
</head>
<body>
  <header>
    <h1>Welcome to My Website</h1>
    <nav>
      <p>Home | About | Contact</p>
    </nav>
  </header>

  <main>
    <section>
      <h2>About Me</h2>
      <p>This is a paragraph about myself.</p>
    </section>
```

```
</main>

<footer>
  <p>&copy; 2024 My Website. All rights reserved.</p>
</footer>
</body>
</html>
```

Advanced HTML Elements:

- **Semantic Tags:** header, nav, main, section, article, footer for better structure
- **div Elements:** Generic containers for grouping and styling
- **span Elements:** Inline containers for styling specific text portions
- **Lists:** ul (unordered), ol (ordered), li (list items) for organized content

CSS Fundamentals:

- **What is CSS:** Cascading Style Sheets - language for describing HTML presentation
- **Why Use CSS:** Separates content from presentation, consistent styling, professional appearance
- **CSS Syntax:** selector { property: value; property: value; }

Inline CSS Implementation:

html

```
<!-- Inline CSS directly in HTML elements -->
<h1 style="color: blue; font-size: 36px; text-align: center;">
  Main Heading
</h1>

<p style="color: #333; font-family: Arial, sans-serif; line-height: 1.5;">
  This is a styled paragraph with custom font and color.
</p>

<div style="background-color: lightblue; padding: 20px; margin: 10px; border: 2px solid navy;">
  This is a styled container with background, padding, and border.
</div>
```

Essential CSS Properties:

Text and Font Styling:

html

```
<h2 style="
  color: darkgreen;
  font-family: 'Times New Roman', serif;
  font-size: 28px;
  font-weight: bold;
  text-align: center;
">Section Heading</h2>
```

```
<p style="
  color: #555;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 16px;
  line-height: 1.6;
  text-align: justify;
">Paragraph with comprehensive text styling.</p>
```

Background and Color Properties:

html

```
<div style="
  background-color: #f0f0f0;
  background-image: linear-gradient(to right, lightblue, lightgreen);
  color: darkblue;
  padding: 15px;
">Styled container with gradient background</div>
```

```
<section style="
  background-color: white;
  border: 1px solid #ccc;
  border-radius: 10px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
">Modern card-style section</section>
```

Layout and Spacing Properties:

html

```
<div style="
  width: 80%;
```

```
height: 200px;
margin: 20px auto;
padding: 30px;
border: 2px solid #333;
border-radius: 15px;
">Controlled size and spacing container</div>
```

```

```

Complete Styled Webpage Example:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My School Project</title>
</head>
<body style="font-family: Arial, sans-serif; line-height: 1.6; margin: 0; padding: 20px; background-color: #f4f4f4;">

  <header style="background-color: #333; color: white; padding: 20px; text-align: center; margin-bottom: 20px;">
    <h1 style="margin: 0; font-size: 2.5em;">My Digital Portfolio</h1>
    <p style="margin: 5px 0 0 0; font-style: italic;">Showcasing my learning journey</p>
  </header>

  <main style="background-color: white; padding: 30px; border-radius: 10px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);">
    <section style="margin-bottom: 30px;">
      <h2 style="color: #333; border-bottom: 2px solid #007acc; padding-bottom: 10px;">About This Project</h2>
      <p style="color: #666; font-size: 16px;">This webpage demonstrates my understanding of HTML structure combined with CSS styling to create professional, visually appealing web content.</p>
```

```

</section>

<section style="margin-bottom: 30px;">
  <h2 style="color: #333; border-bottom: 2px solid #28a745; padding-bottom: 10px;">Skills Learned</h2>
  <ul style="color: #666;">
    <li style="margin-bottom: 8px; padding: 5px; background-color: #f8f9fa; border-radius: 5px;">HTML semantic structure</li>
    <li style="margin-bottom: 8px; padding: 5px; background-color: #f8f9fa; border-radius: 5px;">CSS inline styling</li>
    <li style="margin-bottom: 8px; padding: 5px; background-color: #f8f9fa; border-radius: 5px;">Color theory and typography</li>
    <li style="margin-bottom: 8px; padding: 5px; background-color: #f8f9fa; border-radius: 5px;">Layout and spacing control</li>
  </ul>
</section>

<section>
  <h2 style="color: #333; border-bottom: 2px solid #dc3545; padding-bottom: 10px;">Future Goals</h2>
  <div style="background-color: #fff3cd; border: 1px solid #ffeea7; padding: 15px; border-radius: 8px;">
    <p style="margin: 0; color: #856404;">Continue learning advanced CSS, JavaScript, and responsive web design to create even more impressive websites!</p>
  </div>
</section>
</main>

<footer style="text-align: center; margin-top: 30px; padding: 20px; background-color: #333; color: white;">
  <p style="margin: 0;">©copy; 2024 Student Portfolio. Created with HTML & CSS.</p>
</footer>

</body>
</html>

```

Color Theory for Web Design:

- **Color Codes:** Hexadecimal (#FF0000 for red), RGB (rgb(255,0,0)), named colors (red, blue)
- **Color Schemes:** Complementary (opposite colors), analogous (adjacent colors), monochromatic (shades of one color)
- **Accessibility:** Ensure sufficient contrast between text and background colors
- **Psychology:** Blue (trust, professional), green (nature, growth), red (energy, urgency)

STEP 3: COMPARISON (8 minutes)

Plain HTML vs Styled HTML:

- **Plain HTML:** Basic structure, browser default styling, unprofessional appearance
- **Styled HTML:** Custom appearance, brand consistency, professional presentation

Inline vs External CSS:

- **Inline CSS:** Styles directly in HTML, quick for small projects, harder to maintain
- **External CSS:** Separate file, better organization, reusable across multiple pages

Basic vs Advanced Styling:

- **Basic:** Colors, fonts, simple layouts
- **Advanced:** Gradients, shadows, animations, responsive design

Static vs Dynamic Design:

- **Static:** Fixed appearance, same on all devices
- **Dynamic:** Responsive design, adapts to different screen sizes

STEP 4: CONCLUSION (5 minutes)

Students should understand how to combine HTML structure with CSS styling to create professional webpages, demonstrating competency in color application, typography, layout control, and overall web design principles.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Create an HTML page with a header, main content section, and footer, each with different background colors and text styling
2. What's the difference between inline CSS and the CSS we might put in a separate file?
3. Write the inline CSS to make text blue, 24px size, and centered
4. Design a color scheme for a school website and justify your color choices
5. Create a styled navigation menu using HTML list elements with CSS formatting

WEEK 10: ICT IN EVERYDAY LIFE

Duration: 40 minutes

Topic: Discuss ICT Applications in Banking, Education, and Healthcare

INTRODUCTION (5 minutes)

Explain how ICT has transformed traditional industries, making services more efficient, accessible, and user-friendly while creating new opportunities and challenges in modern society.

STEP 1: TOPIC (2 minutes)

Main Topic: ICT Integration and Impact Across Essential Service Sectors **Sub-topics:**

- Digital transformation in banking and financial services
- Technology revolution in educational systems and learning
- ICT innovations in healthcare delivery and management
- Societal benefits and challenges of digital integration
- Future trends and emerging applications

STEP 2: EXPLANATION (15 minutes)

ICT Applications in Banking:

Digital Banking Services:

- **Online Banking:** Account access, balance checking, transaction history via internet browsers
- **Mobile Banking Apps:** Smartphone applications for banking on-the-go, biometric security
- **ATM Networks:** Automated cash withdrawal, deposit, and transfer services 24/7
- **Point of Sale (POS):** Card payment systems in shops, restaurants, and businesses
- **Internet Payment Platforms:** PayPal, Stripe, local solutions like Paystack, Flutterwave

Advanced Banking Technologies:

- **Blockchain and Cryptocurrencies:** Secure, decentralized transaction systems
- **Artificial Intelligence:** Fraud detection, customer service chatbots, personalized financial advice
- **Contactless Payments:** NFC technology, tap-to-pay cards and mobile wallets
- **Biometric Authentication:** Fingerprint, facial recognition, voice verification for security
- **Cloud Banking:** Data storage and processing in secure remote servers

Benefits of ICT in Banking:

- **Convenience:** 24/7 access, no physical branch visits required
- **Speed:** Instant transfers, real-time balance updates, quick loan approvals
- **Security:** Advanced encryption, multi-factor authentication, fraud monitoring
- **Cost Reduction:** Lower operational costs, reduced paperwork, automated processes
- **Financial Inclusion:** Banking access for remote areas, microfinance solutions

ICT Applications in Education:

Digital Learning Platforms:

- **Learning Management Systems (LMS):** Moodle, Canvas, Google Classroom for course organization

- **Educational Software:** Subject-specific programs for mathematics, science, language learning
- **Virtual Classrooms:** Zoom, Microsoft Teams, Google Meet for remote instruction
- **Educational Games:** Gamified learning experiences, interactive simulations
- **Online Libraries:** Digital books, research databases, multimedia resources

Educational Technology Tools:

- **Interactive Whiteboards:** Smart boards for dynamic presentations and student interaction
- **Tablets and Laptops:** Personal devices for research, assignments, and digital literacy
- **Educational Apps:** Khan Academy, Duolingo, coding platforms for skill development
- **Virtual Reality (VR):** Immersive field trips, historical recreations, scientific simulations
- **Artificial Intelligence:** Adaptive learning systems, automated grading, personalized curricula

Educational Benefits:

- **Accessibility:** Learning opportunities for students with disabilities, remote areas
- **Personalization:** Adaptive learning paths, individual pace progression
- **Engagement:** Interactive content, multimedia presentations, gamification elements
- **Global Collaboration:** International student exchanges, cultural learning projects
- **Resource Sharing:** Open educational resources, teacher collaboration platforms

ICT Applications in Healthcare:

Medical Technology Systems:

- **Electronic Health Records (EHR):** Digital patient files, medical history tracking, prescription management
- **Telemedicine:** Remote consultations via video calls, health monitoring apps
- **Medical Imaging:** Digital X-rays, MRI, CT scans with computer analysis
- **Hospital Management Systems:** Patient registration, appointment scheduling, billing automation
- **Wearable Health Devices:** Fitness trackers, heart monitors, diabetic glucose meters

Advanced Healthcare ICT:

- **Artificial Intelligence Diagnostics:** AI-powered disease detection, medical image analysis
- **Robotic Surgery:** Precision surgical procedures, minimally invasive operations
- **Genetic Analysis:** DNA sequencing, personalized medicine based on genetic profiles
- **Drug Discovery:** Computer modeling for new medication development
- **Health Information Systems:** Epidemiological tracking, public health monitoring

Healthcare ICT Benefits:

- **Improved Accuracy:** Reduced human errors in diagnosis and treatment
- **Faster Treatment:** Quick access to patient information, rapid test results
- **Remote Care:** Healthcare access for rural areas, elderly patients, mobility challenges
- **Cost Effectiveness:** Reduced hospital stays, preventive care, efficient resource allocation

- **Research Advancement:** Big data analysis, clinical trial management, medical discoveries

Societal Impact and Challenges:

Positive Impacts:

- **Increased Efficiency:** Faster service delivery, automated processes, reduced waiting times
- **Better Access:** Services available to previously underserved populations
- **Improved Quality:** Standardized processes, reduced errors, enhanced monitoring capabilities
- **Economic Growth:** New job creation, business opportunities, increased productivity

Challenges and Concerns:

- **Digital Divide:** Unequal access to technology based on location, income, education
- **Privacy and Security:** Data breaches, identity theft, unauthorized access to personal information
- **Job Displacement:** Automation replacing traditional jobs, need for reskilling workers
- **Dependence on Technology:** System failures causing service disruptions, reduced human skills

STEP 3: COMPARISON (8 minutes)

Traditional vs Digital Services:

- **Banking:** Physical branches vs online platforms, paper transactions vs digital records
- **Education:** Textbook learning vs multimedia resources, classroom-only vs anywhere access
- **Healthcare:** Paper records vs electronic systems, in-person only vs telemedicine options

Developed vs Developing Countries:

- **Developed:** Advanced infrastructure, widespread adoption, sophisticated applications
- **Developing:** Leapfrogging technologies, mobile-first solutions, addressing basic access needs

Benefits vs Challenges:

- **Individual Level:** Convenience vs privacy concerns, efficiency vs technology dependence
- **Societal Level:** Economic growth vs job displacement, improved services vs digital divide

Present vs Future Applications:

- **Current:** Established systems, proven technologies, widespread adoption
- **Future:** Emerging technologies, AI integration, more personalized and intelligent services

STEP 4: CONCLUSION (5 minutes)

Students should understand how ICT has revolutionized banking, education, and healthcare, recognizing both the tremendous benefits and important challenges that come with digital transformation in essential service sectors.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Compare traditional banking with modern digital banking, listing three advantages and two challenges of each
 2. How has ICT changed the way students learn compared to education 20 years ago?
 3. Describe three ways ICT has improved healthcare delivery and patient outcomes
 4. What is the digital divide and how does it affect access to ICT services in different sectors?
 5. Choose one sector (banking, education, or healthcare) and predict three future ICT developments that might transform it further
-

WEEK 11: COMPREHENSIVE REVISION

Duration: 40 minutes

Topic: Review and Consolidate First Term Advanced Learning

INTRODUCTION (5 minutes)

Explain that this revision integrates all advanced concepts learned, preparing students for examinations while demonstrating how different technologies work together in real-world applications.

STEP 1: TOPIC (2 minutes)

Main Topic: Integrated Review of Advanced Digital Technology Concepts and Applications

Sub-topics:

- Hardware maintenance and troubleshooting mastery
- Advanced software applications and programming skills
- Web development and design principles
- Real-world ICT applications and societal impact
- Problem-solving using integrated technology solutions

STEP 2: EXPLANATION (15 minutes)

Technical Skills Integration:

Hardware and System Management:

- **Maintenance Knowledge:** Preventive care schedules, cleaning procedures, environmental controls
- **Troubleshooting Skills:** Systematic problem diagnosis, common fault identification, solution implementation

- **Safety Practices:** Proper handling techniques, electrical safety, component protection
- **Performance Optimization:** System maintenance for improved efficiency and longevity

Advanced Software Proficiency:

- **Word Processing Mastery:** Professional document creation with TOC, references, advanced formatting
- **Spreadsheet Analysis:** Complex functions (IF, nested formulas), conditional formatting, data analysis
- **Database Operations:** Query creation, data extraction, information management systems
- **Integration Skills:** Combining multiple applications for comprehensive solutions

Programming and Web Development:

- **Python Programming:** Basic commands, control structures, interactive program creation
- **HTML/CSS Web Design:** Structured content creation, professional styling, visual design principles
- **Problem-Solving Logic:** Algorithm development, systematic approach to coding challenges
- **Creative Application:** Using technical skills for innovative project solutions

Real-World Application Understanding:

- **Industry Applications:** Banking technology, educational systems, healthcare innovations
- **Societal Impact:** Benefits and challenges of digital transformation
- **Career Awareness:** ICT professional roles, skill requirements, preparation pathways
- **Ethical Considerations:** Digital citizenship, privacy protection, responsible technology use

Integrated Problem-Solving Scenarios:

Scenario 1: School Management System Challenge: Design a comprehensive system for managing student information *Required Skills:*

- Database design for student records
- Spreadsheet analysis for grade calculations
- Professional reporting with advanced word processing
- Web interface for information access
- System maintenance procedures

Scenario 2: Small Business Automation Challenge: Create digital solutions for local business operations *Required Skills:*

- Inventory management using spreadsheets with advanced functions
- Customer communication through professional documents
- Simple website for business promotion
- Data backup and system maintenance procedures
- Financial tracking and analysis

Class: JSS3 | First Term | Weeks 1-6

Aligned with NERDC New Curriculum & Oyo State TESCOE Method

WEEK 1: REVISION OF JSS2 WORK

Duration: 40 minutes

Topic: Recall Networking, Databases, HTML, and Cybersecurity Basics

INTRODUCTION (5 minutes)

Welcome students to their final year of junior secondary education. Explain that this revision will reinforce JSS2 concepts and prepare them for more advanced topics in JSS3.

STEP 1: TOPIC (2 minutes)

Main Topic: Comprehensive Review of JSS2 Digital Technology Foundations **Sub-topics:**

- Computer networking fundamentals and connectivity
- Database concepts and data organization
- HTML structure and basic web development
- Cybersecurity principles and protective measures
- Integration of learned concepts for advanced applications

STEP 2: EXPLANATION (15 minutes)

Networking Basics Review:

- **Network Definition:** Two or more connected computers sharing resources and information
- **Network Types:**
 - **LAN (Local Area Network):** Computers in same building or campus
 - **WAN (Wide Area Network):** Networks covering large geographical areas
 - **Internet:** Global network connecting millions of devices worldwide
- **Network Components:** Cables, wireless signals, routers, switches, modems
- **Data Transmission Methods:** Wired (Ethernet, fiber optic) vs Wireless (Wi-Fi, Bluetooth)
- **Network Advantages:** Resource sharing, communication, centralized data storage, cost efficiency

Database Concepts Recap:

- **Database:** Organized collection of related information stored electronically

- **Key Components:**
 - **Tables:** Store data in rows and columns
 - **Records:** Individual entries (rows) containing complete information sets
 - **Fields:** Individual data categories (columns) like Name, Age, Address
- **Database Examples:** Student records, library catalogs, customer information, inventory systems
- **Database Benefits:** Efficient storage, quick searching, data relationships, reduced redundancy
- **Simple Operations:** Adding records, editing information, deleting entries, searching data

HTML Foundation Review:

- **HTML Purpose:** Markup language for creating web page structure and content
- **Basic Document Structure:** <!DOCTYPE html>, <html>, <head>, <body> tags
- **Essential Tags Recap:**
 - **Headings:** <h1> to <h6> for different heading levels
 - **Paragraphs:** <p> for text content organization
 - **Images:** with src and alt attributes
 - **Line Breaks:**
 and horizontal rules <hr>
- **Attributes:** Additional information providing specific instructions to tags
- **File Organization:** Proper naming conventions, folder structures for web projects

Cybersecurity Fundamentals Review:

- **Digital Threats Understanding:**
 - **Malware:** Viruses, worms, trojans designed to damage or steal data
 - **Phishing:** Fake websites or emails trying to steal personal information
 - **Identity Theft:** Unauthorized use of personal information for fraudulent purposes
- **Protection Strategies:**
 - **Strong Passwords:** Complex combinations of letters, numbers, symbols
 - **Antivirus Software:** Programs detecting and removing malicious software
 - **Safe Browsing:** Verifying website security, avoiding suspicious downloads
 - **Privacy Settings:** Controlling who can access personal information online
- **Digital Citizenship:** Responsible, ethical behavior in digital environments

STEP 3: COMPARISON (8 minutes)

Standalone vs Networked Computers:

- **Standalone:** Limited to local resources, no communication capability
- **Networked:** Shared resources, communication, collaborative work, internet access

Traditional vs Digital Data Storage:

- **Traditional:** Paper files, physical storage, manual organization, limited search
- **Digital Databases:** Electronic storage, automated organization, powerful search, data relationships

Static vs Dynamic Web Content:

- **HTML Pages:** Fixed content, basic structure, foundational understanding
- **Advanced Web:** Interactive elements, database integration, user input processing

Reactive vs Proactive Security:

- **Reactive:** Responding to threats after they occur
- **Proactive:** Preventing threats through education, protective software, safe practices

STEP 4: CONCLUSION (5 minutes)

Students should demonstrate solid understanding of networking principles, database organization, HTML structure, and cybersecurity awareness, providing the foundation needed for advanced JSS3 topics including programming, advanced web development, and complex database operations.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the difference between LAN and WAN networks with examples
2. Define database, record, and field, giving a practical example of each
3. Write the basic HTML structure for a simple webpage with a heading and paragraph
4. List five cybersecurity best practices for safe internet use
5. How do networking, databases, HTML, and cybersecurity work together in modern web applications?

WEEK 2: COMPUTER HARDWARE MAINTENANCE

Duration: 40 minutes

Topic: Explain Maintenance Practices; Cleaning and Safe Use of Devices

INTRODUCTION (5 minutes)

Explain that proper hardware maintenance extends computer lifespan, improves performance, prevents costly repairs, and ensures safe operation for users.

STEP 1: TOPIC (2 minutes)

Main Topic: Computer Hardware Care and Maintenance Procedures **Sub-topics:**

- Preventive maintenance principles and scheduling
- Physical cleaning techniques and safety procedures
- Environmental factors affecting hardware longevity
- Safe handling and operation practices

- Troubleshooting common maintenance issues

STEP 2: EXPLANATION (15 minutes)

Preventive Maintenance Principles:

- **Definition:** Regular care activities performed to prevent hardware problems before they occur
- **Maintenance Schedule:**
 - **Daily:** Proper shutdown procedures, checking for unusual sounds or smells
 - **Weekly:** External cleaning, checking cable connections
 - **Monthly:** Internal cleaning, software updates, disk cleanup
 - **Quarterly:** Deep cleaning, hardware inspection, backup verification
- **Benefits:** Extended hardware life, improved performance, reduced repair costs, fewer unexpected failures
- **Documentation:** Keeping maintenance logs, recording issues, tracking performance changes

Physical Cleaning Procedures: External Cleaning:

- **Screen/Monitor:** Use microfiber cloth with appropriate screen cleaner, avoid harsh chemicals
- **Keyboard:** Turn upside down to remove debris, use compressed air between keys, disinfect with alcohol wipes
- **Mouse:** Clean sensor area, wipe surface with disinfectant, check for smooth movement
- **Exterior Surfaces:** Wipe with slightly damp cloth, avoid getting moisture into ports or openings

Internal Cleaning (Advanced/Supervised):

- **Dust Removal:** Use compressed air to blow dust from fans, heat sinks, and components
- **Fan Maintenance:** Ensure fans rotate freely, replace if making unusual noise
- **Ventilation:** Keep air vents clear, ensure proper airflow around computer
- **Cable Management:** Organize cables to prevent airflow restriction, secure loose connections

Environmental Factors:

- **Temperature Control:** Keep computers in cool, well-ventilated areas (ideal: 18-24°C)
- **Humidity Management:** Avoid extremely dry or humid conditions (ideal: 30-50% humidity)
- **Dust Prevention:** Use dust covers when not in use, keep area clean, avoid carpet placement
- **Power Protection:** Use surge protectors, UPS for power outages, avoid overloaded circuits
- **Placement Considerations:** Stable surfaces, away from heat sources, adequate space around vents

Safe Handling Practices: Power Safety:

- **Proper Shutdown:** Use operating system shutdown procedures, never force power off
- **Power Connection:** Ensure secure connections, avoid damaged cables, use appropriate voltage
- **Static Electricity:** Ground yourself before touching internal components, use anti-static wrist straps

Physical Handling:

- **Lifting Techniques:** Lift with legs not back, get help for heavy equipment
- **Transportation:** Use original packaging when possible, secure all components
- **Component Installation:** Handle by edges, avoid touching circuits, ensure proper seating

Operational Safety:

- **Ventilation:** Never block air vents, ensure adequate space for airflow
- **Liquid Protection:** Keep drinks away from computers, use spill-resistant containers
- **Electrical Safety:** Check cables regularly, replace damaged cords immediately

Common Maintenance Tasks:

- **Disk Cleanup:** Remove temporary files, empty recycle bin, uninstall unused programs
- **Defragmentation:** Reorganize hard drive data for improved performance (HDD only)
- **Software Updates:** Keep operating system and drivers current for optimal performance
- **Antivirus Scanning:** Regular system scans to detect and remove threats
- **Backup Procedures:** Regular data backup to prevent loss from hardware failure

STEP 3: COMPARISON (8 minutes)

Preventive vs Corrective Maintenance:

- **Preventive:** Regular scheduled care, prevents problems, lower cost, planned activities
- **Corrective:** Fix problems after they occur, higher cost, unexpected downtime, emergency repairs

Professional vs User Maintenance:

- **User Level:** External cleaning, basic software maintenance, environmental care
- **Professional Level:** Internal repairs, component replacement, advanced diagnostics

Different Hardware Requirements:

- **Desktop Computers:** More accessible for cleaning, better ventilation options
- **Laptops:** More compact, requires careful handling, limited user maintenance
- **Mobile Devices:** Primarily external care, professional service for internal issues

Maintenance Frequency:

- **High-Use Environments:** More frequent cleaning, stricter environmental controls
- **Low-Use Environments:** Standard maintenance schedules, basic care procedures

STEP 4: CONCLUSION (5 minutes)

Students should understand that proper hardware maintenance involves regular cleaning, environmental control, safe handling practices, and preventive care that extends equipment life and ensures reliable operation.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Create a weekly maintenance schedule for a school computer lab
 2. Explain the proper procedure for cleaning a computer keyboard safely
 3. List five environmental factors that can damage computer hardware
 4. What is the difference between preventive and corrective maintenance?
 5. Describe three safety precautions to follow when handling computer components
-

WEEK 3: SYSTEM TROUBLESHOOTING

Duration: 40 minutes

Topic: Identify Common Faults in Computers; Demonstrate Simple Solutions

INTRODUCTION (5 minutes)

Explain that troubleshooting is a systematic approach to identifying and solving computer problems, an essential skill for anyone working with technology.

STEP 1: TOPIC (2 minutes)

Main Topic: Computer Problem Diagnosis and Resolution Techniques **Sub-topics:**

- Common hardware and software problems
- Systematic troubleshooting methodology
- Basic diagnostic tools and techniques
- Simple repair procedures and solutions
- When to seek professional assistance

STEP 2: EXPLANATION (15 minutes)

Systematic Troubleshooting Approach:

- **Step 1: Problem Identification:** Clearly define what is not working correctly
- **Step 2: Information Gathering:** When did problem start? What changed recently?
- **Step 3: Hypothesis Formation:** Develop possible explanations for the problem
- **Step 4: Testing Solutions:** Try simplest solutions first, test one change at a time
- **Step 5: Documentation:** Record what worked, what didn't, for future reference
- **Step 6: Verification:** Ensure problem is completely resolved, not just temporarily fixed

Common Hardware Problems and Solutions:

Computer Won't Start:

- **Symptoms:** No power, no lights, no sounds when power button pressed
- **Simple Solutions:**
 - Check power cable connections to wall and computer
 - Verify power switch on power supply (desktop computers)
 - Try different power outlet or power cable
 - Check for loose internal connections (if comfortable doing so)
- **Advanced Checks:** Power supply testing, motherboard issues (professional help needed)

Overheating Issues:

- **Symptoms:** Computer shuts down unexpectedly, very hot to touch, loud fan noise
- **Simple Solutions:**
 - Clean dust from vents and fans using compressed air
 - Ensure adequate ventilation around computer
 - Check that all fans are working properly
 - Close resource-intensive programs
- **Prevention:** Regular cleaning, proper placement, monitoring temperatures

Peripheral Device Problems:

- **Symptoms:** Keyboard, mouse, or printer not responding
- **Simple Solutions:**
 - Check cable connections and USB ports
 - Try different USB port or cable
 - Restart computer to refresh device connections
 - Update or reinstall device drivers
- **Advanced:** Device manager troubleshooting, hardware replacement

Common Software Problems and Solutions:

Slow Performance:

- **Symptoms:** Programs take long time to open, sluggish response, freezing
- **Simple Solutions:**
 - Close unnecessary programs and browser tabs
 - Restart computer to clear memory
 - Run disk cleanup to free storage space
 - Check for malware using antivirus software
 - Disable startup programs not needed immediately
- **Advanced:** Add more RAM, upgrade to SSD, reinstall operating system

Application Crashes:

- **Symptoms:** Programs close unexpectedly, error messages, frozen applications
- **Simple Solutions:**
 - Update the problematic application to latest version
 - Restart the application and try again
 - Restart computer to clear memory issues
 - Check if problem occurs with specific files or actions
- **Advanced:** Reinstall application, check system logs, hardware diagnostics

Internet Connectivity Issues:

- **Symptoms:** Cannot access websites, slow browsing, connection drops frequently
- **Simple Solutions:**
 - Check network cable or Wi-Fi connection status
 - Restart modem and router (unplug for 30 seconds)
 - Try connecting different device to test network
 - Run built-in network troubleshooter
 - Check with internet service provider for outages

Basic Diagnostic Tools:

- **Built-in Windows Tools:** Device Manager, System Information, Event Viewer, Network Troubleshooter
- **Third-party Tools:** Antivirus scanners, disk cleanup utilities, system monitoring software
- **Hardware Tests:** Memory diagnostic, hard drive health check, temperature monitoring
- **Network Tools:** Ping command, speed tests, connection status indicators

STEP 3: COMPARISON (8 minutes)

Hardware vs Software Problems:

- **Hardware:** Physical component failures, often consistent symptoms, may require replacement
- **Software:** Program conflicts, corrupted files, often intermittent, usually fixable with software solutions

Simple vs Complex Problems:

- **Simple:** User-fixable with basic knowledge and tools (cable connections, restarts, cleaning)
- **Complex:** Require professional diagnosis, specialized tools, or component replacement

Temporary vs Permanent Solutions:

- **Temporary:** Quick fixes that may not address root cause (restart, close programs)
- **Permanent:** Address underlying problem (driver updates, hardware replacement, system reinstall)

User-Level vs Professional Repairs:

- **User-Level:** Safe procedures, external connections, basic software troubleshooting
- **Professional:** Internal hardware work, advanced diagnostics, warranty considerations

STEP 4: CONCLUSION (5 minutes)

Students should understand systematic troubleshooting methodology, be able to identify and resolve common computer problems using simple solutions, and recognize when professional assistance is needed for complex issues.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Describe the six steps of systematic troubleshooting methodology
2. What are three simple solutions to try when a computer won't start?
3. List five symptoms that might indicate a computer is overheating
4. How would you troubleshoot a mouse that suddenly stops working?
5. When should you seek professional help instead of attempting repairs yourself?

WEEK 4: ADVANCED WORD PROCESSING

Duration: 40 minutes

Topic: Create Reports with Table of Contents, References, and Page Layouts

INTRODUCTION (5 minutes)

Explain that advanced word processing skills are essential for creating professional academic and business documents, including research papers, reports, and formal publications.

STEP 1: TOPIC (2 minutes)

Main Topic: Professional Document Creation and Advanced Formatting **Sub-topics:**

- Table of contents generation and management
- Advanced reference and citation systems
- Professional page layout and design principles
- Document structure and organization techniques
- Collaborative editing and review processes

STEP 2: EXPLANATION (15 minutes)

Table of Contents Creation:

- **Purpose:** Provides document navigation, shows document structure, enables quick access to sections
- **Automatic Generation Process:**
 1. **Apply Heading Styles:** Use built-in heading styles (Heading 1, Heading 2, Heading 3) consistently
 2. **Insert Table of Contents:** References tab → Table of Contents → Choose style
 3. **Update Automatically:** Right-click TOC → Update Field when document changes
- **Heading Style Hierarchy:**
 - **Heading 1:** Main chapters or major sections
 - **Heading 2:** Sub-sections within chapters
 - **Heading 3:** Sub-sub-sections for detailed breakdown
- **Customization Options:** Modify styles for consistent formatting, adjust tab leaders, set page number alignment
- **Navigation Benefits:** Ctrl+Click to jump to sections, print TOC for document overview

Advanced Reference Systems: Citation Management:

- **In-Text Citations:** Brief source references within document text
- **Bibliography Creation:** Complete source list automatically generated from citations
- **Citation Styles:** APA (psychology, education), MLA (literature, humanities), Chicago (history, arts)
- **Source Types:** Books, journal articles, websites, interviews, government documents

Reference Insertion Process:

1. **References Tab:** Insert Citation → Add New Source
2. **Source Information:** Fill complete details (author, title, publisher, date)
3. **Citation Placement:** Insert at appropriate points in text
4. **Bibliography Generation:** Insert Bibliography at document end
5. **Style Management:** Change citation style affects entire document automatically

Footnotes and Endnotes:

- **Footnotes:** Additional information at bottom of each page
- **Endnotes:** Collected at document end for comprehensive reference
- **Insertion:** References → Insert Footnote/Endnote
- **Automatic Numbering:** Sequential numbering updates automatically when notes added/removed

Professional Page Layout: Page Setup and Margins:

- **Standard Margins:** 1 inch on all sides for academic papers
- **Custom Margins:** Adjust for binding requirements or specific formats
- **Orientation:** Portrait (standard) vs Landscape (wide tables, charts)
- **Paper Size:** A4 (international standard), Letter (US standard)

Section Breaks and Page Layout:

- **Section Breaks:** Allow different formatting in same document
- **Page Breaks:** Control where pages end for better organization
- **Column Layout:** Multi-column text for newsletter or academic formats
- **Headers/Footers by Section:** Different headers for chapters or sections

Advanced Formatting Features:

- **Drop Caps:** Large first letter for chapter beginnings
- **Text Boxes:** Callout information, sidebar content
- **Watermarks:** Draft stamps, confidential markings, background images
- **Page Borders:** Decorative elements for certificates or formal documents

Document Structure Best Practices: Professional Document Elements:

- **Title Page:** Document title, author information, date, institutional affiliation
- **Executive Summary/Abstract:** Brief overview of document contents and conclusions
- **Table of Contents:** Navigation aid for readers
- **Main Body:** Organized with clear headings and logical flow
- **Conclusion:** Summary of key points and recommendations
- **References/Bibliography:** Complete source documentation
- **Appendices:** Supporting materials, detailed data, additional information

Collaborative Features:

- **Track Changes:** Monitor edits from multiple authors
- **Comments:** Add feedback without changing document text
- **Version Control:** Compare documents, accept/reject changes
- **Sharing Options:** Cloud collaboration, email review, permission settings

STEP 3: COMPARISON (8 minutes)

Manual vs Automatic Features:

- **Manual TOC:** Time-consuming, error-prone, requires manual updates
- **Automatic TOC:** Generated from styles, updates automatically, consistent formatting

Simple vs Professional Documents:

- **Simple:** Basic formatting, manual page numbers, limited structure
- **Professional:** Automatic features, consistent styling, comprehensive references

Academic vs Business Documents:

- **Academic:** Strict citation requirements, formal structure, research focus
- **Business:** Practical information, executive summaries, action-oriented content

Individual vs Collaborative Work:

- **Individual:** Simple document creation, basic editing tools
- **Collaborative:** Version control, change tracking, comment management, shared access

STEP 4: CONCLUSION (5 minutes)

Students should understand how to create professional documents using advanced word processing features including automatic table of contents, proper reference systems, and sophisticated page layouts suitable for academic and business environments.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the process of creating an automatic table of contents using heading styles
2. What is the difference between footnotes and endnotes? When would you use each?
3. List the essential elements of a professional business report in correct order
4. How do you insert and manage citations in different academic styles (APA, MLA)?
5. Create a document outline for a school project report, including all advanced formatting elements discussed

WEEK 5: SPREADSHEET ADVANCED FUNCTIONS

Duration: 40 minutes

Topic: Use IF Statements, Conditional Formatting, and Nested Functions

INTRODUCTION (5 minutes)

Explain that advanced spreadsheet functions transform simple data tables into powerful analytical tools capable of complex decision-making and automated data analysis.

STEP 1: TOPIC (2 minutes)

Main Topic: Complex Spreadsheet Analysis and Automated Decision Making **Sub-topics:**

- IF statement logic and practical applications
- Conditional formatting for visual data analysis
- Nested functions for complex calculations
- Advanced data validation and error handling
- Real-world business and academic applications

STEP 2: EXPLANATION (15 minutes)

IF Statement Fundamentals:

- **Purpose:** Make decisions based on conditions, automate analysis, categorize data
- **Basic Syntax:** =IF(condition, value_if_true, value_if_false)
- **Logical Operators:**
 - **Equal to:** = (e.g., A1=100)
 - **Greater than:** > (e.g., B1>75)
 - **Less than:** < (e.g., C1<50)
 - **Greater/equal:** >= (e.g., D1>=60)
 - **Less/equal:** <= (e.g., E1<=100)
 - **Not equal:** <> (e.g., F1<>"")

Practical IF Statement Examples:

- **Grade Classification:** =IF(A1>=70,"Pass","Fail")
- **Sales Commission:** =IF(B1>10000,B10.15,B10.10)
- **Attendance Status:** =IF(C1>=75%,"Regular","Irregular")
- **Budget Analysis:** =IF(D1>E1,"Over Budget","Within Budget")
- **Inventory Management:** =IF(F1<50,"Reorder","Sufficient")

Advanced IF Statements: Multiple Conditions (Nested IF):

- **Grade Levels:** =IF(A1>=90,"A",IF(A1>=80,"B",IF(A1>=70,"C",IF(A1>=60,"D","F"))))
- **Salary Ranges:** =IF(B1<30000,"Entry",IF(B1<50000,"Mid",IF(B1<80000,"Senior","Executive")))

Logical Functions:

- **AND Function:** =IF(AND(A1>60,B1>60),"Both Pass","At Least One Fail")
- **OR Function:** =IF(OR(C1="Excellent",C1="Good"),"Satisfactory","Needs Improvement")
- **NOT Function:** =IF(NOT(D1="Absent"),"Present","Absent")

Conditional Formatting:

- **Purpose:** Visual highlighting of data based on values, trends, or conditions
- **Application Process:**
 1. Select data range to format
 2. Home tab → Conditional Formatting
 3. Choose rule type (greater than, between, top 10, etc.)
 4. Set conditions and formatting (colors, fonts, borders)

Conditional Formatting Types:

- **Highlight Cell Rules:** Values greater than, less than, between, equal to specific values
- **Top/Bottom Rules:** Highest values, lowest values, above/below average
- **Data Bars:** Horizontal bars showing relative values within cells
- **Color Scales:** Color gradients from lowest to highest values
- **Icon Sets:** Traffic lights, arrows, stars to show performance levels

Practical Formatting Examples:

- **Student Grades:** Green for A grades, yellow for B-C, red for D-F
- **Sales Performance:** Data bars showing monthly sales relative to targets
- **Budget Variance:** Red for over-budget, green for under-budget items
- **Project Status:** Icons showing on-time, at-risk, or delayed projects

Nested Functions Applications: Complex Business Calculations:

- **Commission Structure:** `=IF(A1>50000,A10.20,IF(A1>25000,A10.15,A1*0.10))`
- **Shipping Costs:** `=IF(B1>100,0,IF(B1>50,5,10))`
- **Performance Ratings:** `=IF(AND(C1>=8,D1>=85),"Excellent",IF(OR(C1>=6,D1>=70),"Good","Needs Improvement"))`

Academic Applications:

- **Scholarship Eligibility:** `=IF(AND(GPA>=3.5,Income<50000),"Eligible","Not Eligible")`
- **Course Prerequisites:** `=IF(AND(Math>=70,English>=65),"Can Enroll","Missing Prerequisites")`

Data Validation and Error Handling:

- **IFERROR Function:** `=IFERROR(A1/B1,"Cannot divide by zero")`
- **Data Validation:** Restrict input to specific ranges, lists, or formats
- **Error Prevention:** Check for common calculation errors before they occur

STEP 3: COMPARISON (8 minutes)

Simple vs Complex Formulas:

- **Simple:** Basic calculations (+, -, *, /, SUM, AVERAGE)
- **Complex:** IF statements, nested functions, multiple conditions, logical operations

Manual vs Automated Analysis:

- **Manual:** Review each data point individually, prone to errors, time-consuming
- **Automated:** IF statements and formatting highlight important data automatically

Static vs Dynamic Formatting:

- **Static:** Fixed colors and formats that don't change with data
- **Dynamic:** Conditional formatting that updates automatically as data changes

Single vs Multiple Criteria:

- **Single:** One condition determines outcome (IF statement)
- **Multiple:** AND/OR functions combine several conditions for complex decisions

STEP 4: CONCLUSION (5 minutes)

Students should understand how to use IF statements for automated decision-making, apply conditional formatting for visual data analysis, and create nested functions for complex business and academic calculations.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Write an IF statement that assigns letter grades based on numerical scores (A: 90+, B: 80-89, C: 70-79, D: 60-69, F: below 60)
 2. Explain how conditional formatting can help identify trends in sales data
 3. Create a nested IF statement for employee bonus calculation based on years of service and performance rating
 4. What is the difference between AND and OR logical functions? Give examples
 5. Design a spreadsheet system for tracking student attendance with automatic status updates and visual indicators
-

WEEK 6: DATABASE QUERIES

Duration: 40 minutes

Topic: Explain and Run Simple Queries in MS Access or Excel

INTRODUCTION (5 minutes)

Explain that database queries are powerful tools for extracting specific information from large datasets, enabling users to find answers to questions and generate reports efficiently.

STEP 1: TOPIC (2 minutes)

Main Topic: Database Information Retrieval and Data Analysis Techniques **Sub-topics:**

- Query concepts and purposes in database management
- Simple query creation in MS Access
- Excel database functions and filtering techniques
- Query criteria and result optimization
- Practical applications in academic and business contexts

STEP 2: EXPLANATION (15 minutes)

Database Query Fundamentals:

- **Definition:** Structured requests for specific information from database tables
- **Purpose:** Extract relevant data, answer specific questions, generate reports, analyze trends
- **Query Types:**

- **Select Queries:** Retrieve specific records meeting certain criteria
- **Parameter Queries:** Ask for input values when query runs
- **Calculated Queries:** Perform calculations on retrieved data
- **Summary Queries:** Group data and calculate totals, averages, counts

MS Access Query Operations:

Simple Select Queries:

- **Query Design View:** Visual interface for building queries without coding
- **Steps to Create:**
 1. Create → Query Design
 2. Select table(s) containing needed data
 3. Drag fields from table to query grid
 4. Add criteria in Criteria row
 5. Run query to see results

Query Criteria Examples:

- **Exact Match:** "Lagos" (finds records exactly matching Lagos)
- **Numerical Comparison:** >18 (finds records greater than 18)
- **Date Ranges:** Between #1/1/2024# And #12/31/2024#
- **Text Patterns:** Like "Smith*" (finds names starting with Smith)
- **Multiple Criteria:** Use AND (both conditions) or OR (either condition)

Practical Access Query Examples:

- **Student Records:** Find all students with GPA > 3.5
- **Inventory Management:** Show products with quantity < 50
- **Sales Analysis:** Display sales between specific dates
- **Employee Database:** List employees in specific department

Parameter Queries:

- **Interactive Queries:** Prompt user for input when query runs
- **Syntax:** [Enter Student Name] in criteria field
- **Benefits:** Reusable queries, flexible searching, user-friendly interface

Excel Database Functions:

Data Filtering:

- **AutoFilter:** Data tab → Filter → Click dropdown arrows on column headers
- **Custom Filters:** Text filters, number filters, date filters with specific criteria
- **Advanced Filter:** More complex criteria using separate criteria range
- **Filter Benefits:** Hide irrelevant data, focus on specific records, maintain original data integrity

Excel Database Functions:

- **COUNTIF:** Count cells meeting specific criteria
 - Example: =COUNTIF(A:A,">50") counts values greater than 50
- **SUMIF:** Sum values meeting specific criteria
 - Example: =SUMIF(B:B,"Electronics",C:C) sums sales for Electronics category
- **AVERAGEIF:** Calculate average of values meeting criteria
 - Example: =AVERAGEIF(D:D,">=70") averages scores 70 and above

Advanced Excel Queries:

- **VLOOKUP:** Find information in table based on lookup value
 - Example: =VLOOKUP(student_id,student_table,grade_column,FALSE)
- **Pivot Tables:** Summarize and analyze large datasets interactively
- **Data Validation:** Ensure data entry meets specific criteria

Query Optimization Techniques:

- **Select Necessary Fields Only:** Improves performance, reduces result complexity
- **Use Specific Criteria:** Narrow results to relevant information
- **Sort Results:** Organize output for easier analysis
- **Index Important Fields:** Speeds up query performance (Access)

Real-World Query Applications:

Academic Examples:

- **Student Performance:** Find students needing academic support
- **Course Analysis:** Identify most/least popular courses
- **Attendance Tracking:** Students with attendance below threshold
- **Grade Distribution:** Statistical analysis of class performance

Business Examples:

- **Customer Analysis:** Find high-value customers for special promotions
- **Inventory Reports:** Items needing reorder or overstocked items
- **Sales Performance:** Top-performing products or sales representatives
- **Financial Analysis:** Expenses exceeding budget categories

Report Generation:

- **Query Results as Reports:** Use query output for formatted reports
- **Regular Updates:** Queries refresh automatically with new data
- **Data Export:** Save query results to Excel, PDF, or other formats
- **Automated Reporting:** Schedule queries to run at specific intervals

STEP 3: COMPARISON (8 minutes)

Access vs Excel for Database Operations:

- **Access:** Purpose-built for databases, complex relationships, better for large datasets
- **Excel:** Familiar interface, good for smaller datasets, better calculation features

Simple vs Complex Queries:

- **Simple:** Single table, basic criteria, straightforward results
- **Complex:** Multiple tables, calculated fields, grouped results, parameter input

Manual vs Query-Based Data Analysis:

- **Manual:** Time-consuming, error-prone, difficult with large datasets
- **Query-Based:** Fast, accurate, repeatable, handles large amounts of data efficiently

Static vs Dynamic Queries:

- **Static:** Fixed criteria, same results each time
- **Dynamic:** Parameter queries, criteria can change, interactive user input

STEP 4: CONCLUSION (5 minutes)

Students should understand database query concepts, be able to create simple queries in both MS Access and Excel, and recognize how queries can efficiently extract meaningful information from large datasets for decision-making.

STEP 5: EVALUATION (5 minutes)

Questions:

1. Explain the difference between a query and a table in database terminology
2. Write the steps to create a simple query in MS Access that finds all students with grades above 85
3. What Excel function would you use to count how many products have a price greater than ₦5,000?
4. Compare the advantages of using queries versus manual data searching
5. Design a query scenario for a school library database that would help librarians track overdue books

GENERAL ASSESSMENT FOR WEEKS 1-6

Practical Activities:

1. **System Maintenance Lab:** Perform complete computer cleaning and maintenance checklist

2. **Troubleshooting Simulation:** Diagnose and solve common hardware/software problems
3. **Professional Document Creation:** Create multi-page report with TOC, references, and advanced formatting
4. **Advanced Spreadsheet Project:** Build grade book with IF statements and conditional formatting
5. **Database Query Exercise:** Create and run queries to extract specific information from sample databases
6. **Integration Project:** Combine word processing, spreadsheet, and database skills in comprehensive assignment

Cumulative Project Assignment: "Digital Solutions for School Management" Students will create a comprehensive project including:

- **Database Component:** Student information system with query capabilities
- **Spreadsheet Analysis:** Grade calculations using advanced functions and conditional formatting
- **Professional Report:** Document the entire system with proper formatting, TOC, and references
- **Maintenance Guide:** Computer care procedures and troubleshooting manual
- **Presentation:** Demonstrate all components and explain design decisions

Assessment Criteria:

- **Technical Proficiency (40%):** Correct use of advanced features and functions
- **Problem-Solving (30%):** Effective troubleshooting and system design
- **Professional Presentation (20%):** Document formatting and project organization
- **Integration Skills (10%):** Seamless combination of different software tools

Learning Outcomes Achieved: By the end of Week 6, students should demonstrate:

- Comprehensive understanding of JSS2 foundation concepts
- Competent computer maintenance and troubleshooting skills
- Advanced word processing capabilities for professional documents
- Complex spreadsheet analysis using logical functions and formatting
- Database query skills for information extraction and analysis
- Integration of multiple software applications for comprehensive solutions