



Girls' Education and Skills Partnership



Yaba College of Technology

# WEB DEVELOPMENT

**Frontend Development  
and WordPress Authoring**



GESP

POWERED BY:





NATIONAL SKILLS QUALIFICATION

LEVEL  
**III**

# Web Development

**Frontend Development and  
WordPress Authoring**



**Hi, my name is Stella, I shall be guiding you through this material developed to support your hands-on activities in the workplace environment.**

**I encourage you not to entertain any kind of fear that could hinder you from learning Website Development.**

**In addition to the support you shall enjoy from your trained instructors, I shall be here to guide you through the resources.**

**Come with me and lets go right in.**



# Copyright



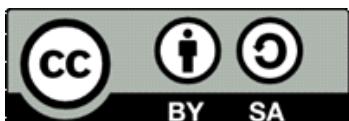
This gender-responsive resource for the Web Development in Girls' Education and Skills Partnership Project has been developed by Yaba College of Technology (Yabatech), Lagos, Nigeria. This project is financed by the Foreign, Commonwealth & Development Office (FCDO), facilitated by UNICEF, and implemented by Yabatech.

The content has been created based on the National Occupational Standard (NOS) for Skills Qualification Programme under the auspices of the National Board for Technical Education (NBTE) to prepare learners for the certification of NSQ level II.

The Foreign, Commonwealth & Development Office (FCDO) is a department of the UK government responsible for promoting British interests worldwide. It works to eradicate global poverty, address global challenges, and support sustainable development.

The United Nations International Children's Emergency Fund (UNICEF) is a United Nations agency responsible for providing humanitarian and developmental aid to children worldwide. UNICEF works in over 190 countries and territories to save children's lives, defend their rights, and help them fulfill their potential.

Yaba College of Technology is a fully accredited polytechnic offering National Diploma, Higher National Diploma, and non-formal courses through various modalities, including open distance and flexible learning systems. These systems use multiple communication methods such as face-to-face instruction, online courses, correspondence, seminars, e-learning, and blended learning modes. This resource is intended to complement the learning activities conducted at workplace centers



© 2024 by FCDO, UNICEF and Yabatech. Unless otherwise noted, the content is available under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) License:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode>.



# Acknowledgement



The development of this resource for Web Development has been a collaborative effort made possible by the contributions and dedication of numerous individuals and organizations. This project, financed by the Foreign, Commonwealth & Development Office (FCDO), facilitated by UNICEF, and implemented by Yaba College of Technology (YCT), has benefited greatly from the expertise and support of professionals from the following categories of the resource development team:

- Subject Matter Expert
- Learning Designers
- Course Developers
- Media Designer
- Reviewers
- Moderators
- Coordinator
- Gender Specialist

Special recognition to the International Network for Advancing Science and policy (INASP), for their critical role as the gender specialist, ensuring the gender-responsiveness of the developed resources during both the planning and implementation stages of the resource development.

We cannot but mention and appreciate the supervisory role played by the GESP project team in Yabatech. Hence, we say thank you to the following project team member:

- Dr. Ibrahim Abdul (Rector, Yabatech) / TVET Expert
- Dr. Funmilayo Doherty (Director, Centre for Research Support & Grant Management)
- Engr. Adebayo Adebari (Director, Center for Open Distance and Flexible Learning)
- Dr. Olusola Dada (Head, Final Account Department)
- Mr. Alade Amos (Centre Manager, Skills Development Centre)
- Dr. Idowu Aneyo (University of Lagos, Nigeria)

We also extend our gratitude to all the staff and students of Yaba College of Technology, the teams at FCDO and UNICEF, and everyone who contributed to this project. Your dedication and support have been instrumental in making this resource a success. Thank you for your commitment to advancing gender-responsive education and empowering young girls through web development skills.

# About Course Materials



## Welcome to Web Development, Level II

This qualification is about a Web Development Trade that is responsive to and reflects workers and employers' need in the work environment for all professional areas. It can be taken by all learners who wish to have competencies for the purpose of employment or enterprise creation. It provides the fundamental knowledge on health and safety standards in work environments, Communication System in a work environment, Team Work, Basic Computer Appreciation, Use of Computer for web Application, Understanding the World Wide Web (WWW), Use of online resources in learning, Understanding Development Environment, Introduction to Client Side Programming, Designing Website etc.

### Is this course for you?

This course is intended for the people who doesn't or already have introductory knowledge about computer, especially adolescent girls and young women, particularly those aged 13-24, who are eager to develop market-relevant skills and enhance their prospects for secure livelihoods. This program specifically targets young women who may be disadvantaged by socio-economic barriers, limited access to quality education, or societal norms that restrict their opportunities.

It is ideal for:

- **Young Women Seeking Skills Development:** Those who are passionate about gaining hands-on experience and practical skills in ICT fields such as Computer Hardware Repair, Mobile Phone Repair, Website Development, and Mobile Application Development.
- **Adolescent Girls and School Leavers:** Girls who are at a critical juncture in their educational journey and are looking for opportunities to acquire skills that can lead to immediate employment or entrepreneurial ventures.
- **Aspiring Female Entrepreneurs:** Young women interested in starting their own businesses within the ICT sector and seeking the knowledge and support necessary to succeed.
- **Women from Underserved Communities:** Those who may not have had access to traditional educational resources but are determined to break through barriers and achieve economic independence.

# Course Overview



# Course Outcome



Upon successful completion of the course, learners will be able to:

- Observe health and safety standards in work environments and hazard.
- Communicate in a work environment.
- Perform in a team work
- Carry out basic Computer Appreciation
- Understand the World Wide Web (WWW) and Development Environment
- Perform basic client side programming
- Design a Website and conveniently Author a WordPress site

## Time Frame



This is a 30 credit qualification, to achieve this qualification; Learners are required to achieve a minimum of 24 credits from mandatory units and 3 credits from the optional units. Each Credit is equivalent to approximately 10 Guided Learning Hours (GLH).

The Total Learning Hours will therefore consist of the GLH plus the independent learning hours of the candidate, which is generally 50% – 150% of the GLH. The actual Total Learning Hours for each Credit will then be a minimum of 15 hours.

# Assessment

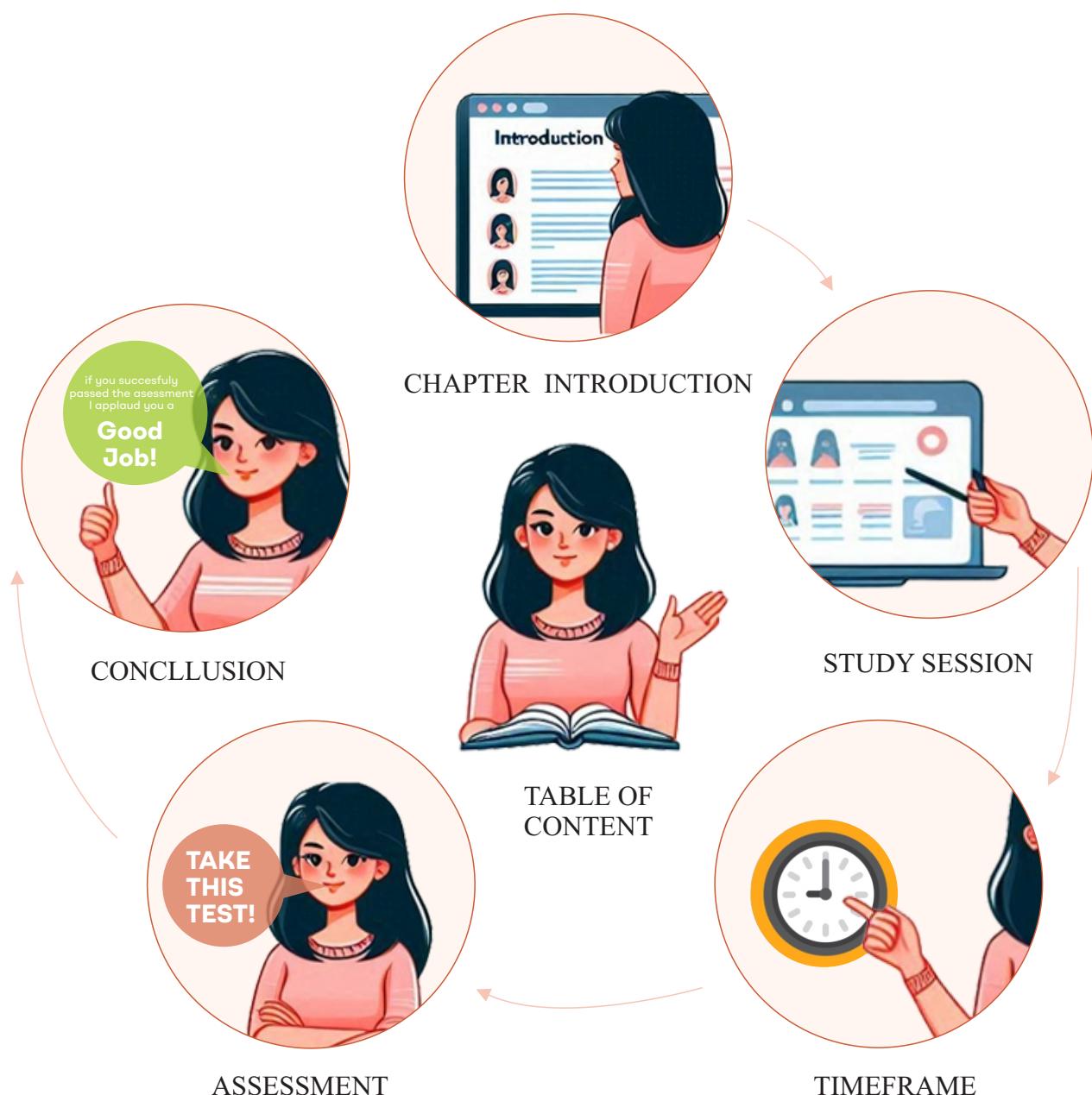
Learner's performance will be assessed using knowledge and vocational assessment methods which will be administered by the assigned Quality Assurance Assessor (QAA) personnel. We refer you to the National Occupational Standard (NOS), where assessment methods to be used for each unit are specified.

However, there are practical exercises which are meant to be carried out with the Mastercraft or instructor to ensure your competency before final assessment by qualified assessor.



# Getting around this Course Material

While using this course material, you will notice the frequent use of my images but of different postures. These postures serve to “signpost” some specific tasks or activity; they have been included to help you to find your way around this course material. A list of my postures used in this material is shown below. I suggest that you familiarize yourself with the postures and their meaning before starting your study.





# TABLE OF CONTENTS

MODULE 01

## OCCUPATIONAL HEALTH AND SAFETY IN AN ICT ENVIRONMENT

- Safe Working Practices and Instructions in an ICT Environment
- Safety Hazards and Risks in an ICT Environment
  - Practical Exercise
  - Additional Resources

Page No.

1-6



MODULE 02

## COMMUNICATIONS AND INTERPERSONAL SKILL

- Importance of Good Communication
- Documented Instruction
  - Practical Exercise
  - Additional Resources

Page No.

7-12



MODULE 03

## TEAM WORK

- Positive Working Relations In Teamwork
- Responsibility In Teamwork
- Compliance With Organisational Policy
  - Practical Exercise
  - Additional Resources

Page No.

13-20



MODULE 04

## USE OF COMPUTER FOR WEB DEVELOPMENT

- Effective Use Of The Internet In Web Development
  - Practical Exercise
  - Additional Resources
- Types of Computer network and their Roles in Web Development
  - Practical Exercise
  - Additional Resources

Page No.

21-38





# TABLE OF CONTENTS

MODULE 05

## OCCUPATIONAL HEALTH AND SAFETY IN AN ICT ENVIRONMENT

- Safe Working Practices and Instructions in an ICT Environment
- Safety Hazards and Risks in an ICT Environment
  - Practical Exercise
  - Additional Resources

Page No.

1-6



MODULE 06

## COMMUNICATIONS AND INTERPERSONAL SKILL

- Importance of Good Communication
- Documented Instruction
  - Practical Exercise
  - Additional Resources

Page No.

7-12



MODULE 03

## TEAM WORK

- Positive Working Relations In Teamwork
- Responsibility In Teamwork
- Compliance With Organisational Policy
  - Practical Exercise
  - Additional Resources

Page No.

13-20



MODULE 04

## USE OF COMPUTER FOR WEB DEVELOPMENT

- Effective Use Of The Internet In Web Development
  - Practical Exercise
  - Additional Resources
- Types of Computer network and their Roles in Web Development
  - Practical Exercise
  - Additional Resources

Page No.

21-38





# TABLE OF CONTENTS

## MODULE 09

### OCCUPATIONAL HEALTH AND SAFETY IN AN ICT ENVIRONMENT

- Safe Working Practices and Instructions in an ICT Environment
- Safety Hazards and Risks in an ICT Environment
  - Practical Exercise
  - Additional Resources

Page No.

1-6



## MODULE 10

### COMMUNICATIONS AND INTERPERSONAL SKILL

- Importance of Good Communication
- Documented Instruction
  - Practical Exercise
  - Additional Resources

Page No.

7-12



# OCCUPATIONAL HEALTH AND SAFETY IN AN ICT ENVIRONMENT



This unit specifies the competencies required to demonstrate understanding of safe work practices.

It involves learning about workplace safety, correct use of signs and symbols, identifying and reducing risks of hazards in the work environment and knowing how to protect self in the work environment.

**At the end of this Module, learners should be able to:**

- demonstrate safe working practices and instructions
- demonstrate Understanding of Safety Hazards and risks
- demonstrate the ability to take appropriate actions during accident/injury
- demonstrate safe work habit and clean work environment



# Safe Working Practices and Instructions in an ICT Environment

## Safe Working Instruction

Safe working instructions are specific, detailed directives that guide employees on how to safely carry out particular tasks or use certain equipment. These instructions are often included in training manuals, posted near workstations, or provided during training sessions to ensure that employees are aware of the correct and safe way to perform their duties. In a software environment, examples of safe working instructions include Regular Code Reviews, Version Control Usage, Backup Procedures, Secure Authentication, Environment Isolation, Regular Updates, and Error Logging



## Safe Working Practice

Safe working practices refer to the guidelines and procedures established to ensure the safety and well-being of employees while they perform their tasks. These practices are designed to minimize the risk of injury and accidents in the workplace by promoting safe behaviors and the proper use of equipment. In a software environment, examples of safe working practice include Ergonomic Workspace Setup, Regular Breaks, healthy habits, collaborative culture, continuous learning, mental health awareness, secure physical environment



### Difference between Safe Working Instruction and Practice

- **Safe Working Practice** is the general guidelines and procedures aimed at promoting safety across various tasks and environments while
- **Safe Working Instruction** is the Specific directives related to the safe execution of particular tasks or the use of specific equipment.

### Safe Working Instructions in an ICT Environment

- Ensure all electrical equipment is properly grounded.
- Avoid overloading power sockets and extension cords.
- Regularly inspect and maintain equipment to prevent malfunctions.
- Use anti-static wrist straps when working with sensitive components.
- Keep liquids away from electronic devices to prevent spills and short circuits.

### Safe Working Practices in an ICT Environment

- Maintain an organized and clutter-free workspace.
- Follow ergonomic guidelines to prevent strain and injury.
- Take regular breaks to avoid eye strain and repetitive stress injuries.
- Ensure proper ventilation to avoid overheating of equipment.
- Adhere to safety protocols during the setup and maintenance of hardware.

## Safety Hazards and Risks in an ICT Environment

### Define Safety Hazards and Risks

**Safety Hazards:** Conditions or practices that have the potential to cause harm or injury.

**Risks:** The likelihood or probability that a person may be harmed or experience adverse health effects if exposed to a hazard.

### Hazards in an ICT Environment and Methods to Reduce the Risk

Hazard	Risk Reduction Method
Electrical Hazards	Ensure proper grounding, regular maintenance, use of surge protectors
Tripping Hazards (cables)	Use cable management systems, keep walkways clear
Ergonomic Issues	Provide ergonomic furniture, encourage proper posture
Fire Hazards	Maintain clean and dry areas around electrical equipment, use fire extinguishers
Eye Strain	Use anti-glare screens, encourage regular breaks

### Safety Equipment Applicable to the ICT Environment

1. Fire extinguishers
2. Anti-static wrist straps
3. First aid kits
4. Cable management tools
5. Ergonomic furniture



### How to Maintain a Hygienic, Safe, and Secure Workplace

Regularly clean and disinfect workstations.

Ensure proper waste disposal.

Implement security measures such as access controls and surveillance.

Conduct regular safety audits and drills.

Provide training on hygiene and safety protocols.

### First-Aid Treatments

- **Cuts and Scrapes:** Clean the wound with antiseptic, apply a bandage.
- **Burns:** Cool the burn with running water, cover with a sterile dressing.
- **Electrical Shock:** Turn off the power source, check for breathing, and call for emergency assistance.
- **Sprains and Strains:** Apply ice, elevate the injured area, and rest.

**End of Study Session**

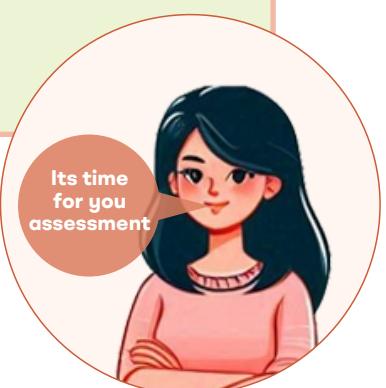
**CLICK! TO RESTART THIS LESSON**



## PRACTICAL EXERCISE

*Divide participants into small groups, let them:*

- Sets up an ergonomic workstation, ensuring correct monitor height, chair position, and keyboard/mouse placement.
- Inspect a pre-arranged workspace with various hazards (e.g., overloaded power sockets, tangled cables, improper equipment setup). Participants identify and list the hazards, then propose methods to mitigate the risks.



Its time  
for you  
assessment



*Now, can you confidently say repeat these statements to yourself:*

**I can:**

- demonstrate safe working practices and instructions
- demonstrate Understanding of Safety Hazards and risks
- demonstrate the ability to take appropriate actions during accident/injury
- demonstrate safe work habit and clean work environment

**CLICK! FOR YOUR NEXT MODULE**



# COMMUNICATIONS AND INTERPERSONAL SKILL

This unit seeks to develop the competency of the learner to be able to express self fluently in a well-defined manner understandable to the client with problems to solve and with group of colleagues.



**At the end of this Module, learners should be able to:**

- know of the importance of good
- demonstrate ability to follow documented instructions



## Importance of Good Communication

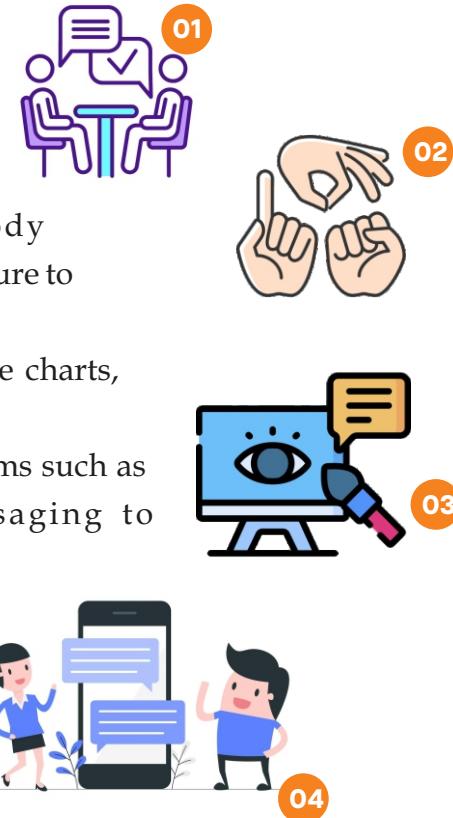
### What is Communication?

Communication is the process of exchanging information, ideas, thoughts, and feelings between individuals or groups through various means such as speaking, writing, gestures, or digital channels.

### Types of Communication

- Verbal Communication:** Using spoken or written words to convey messages.
- Non-verbal Communication:** Using body language, facial expressions, gestures, and posture to communicate.
- Visual Communication:** Using visual aids like charts, graphs, and images to share information.
- Digital Communication:** Using digital platforms such as email, social media, and instant messaging to communicate.

### Images:



### Why Good Communication is Important

- Enhances Collaboration:** Promotes teamwork and cooperation among colleagues.
- Prevents Misunderstandings:** Clear communication reduces the chances of errors and confusion.
- Builds Relationships:** Effective communication fosters trust and understanding in professional relationships.
- Improves Productivity:** Streamlined communication processes lead to more efficient workflows.
- Promotes Problem-Solving:** Open communication channels encourage the sharing of ideas and solutions.

### Ways to Communicate Effectively

- **Active Listening:** Pay attention to the speaker, show empathy, and provide feedback.
- **Clarity and Conciseness:** Be clear and to the point to avoid misunderstandings.
- **Non-verbal Cues:** Use appropriate body language and facial expressions.
- **Feedback:** Provide constructive feedback and be open to receiving it.
- **Adaptability:** Tailor your communication style to your audience.

### Communicating with Colleagues, Managers, and Clients

- **Colleagues:** A good communication fosters a collaborative environment, so you must be respectful, and offer support in your approach.
- **Managers:** When you communicate with a manager of a project, unit or team be clear and concise, provide regular updates, and respect their time.
- **Clients:** Communication with clients requires you to always listen to their needs, provide clear information, and maintain professionalism.

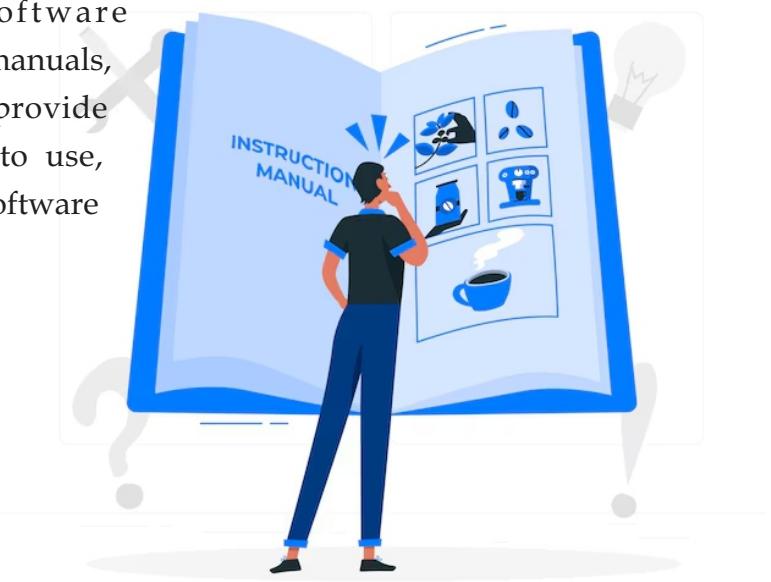
### Respectful Body Language in a Bad Mood and Under Pressure

- **Maintain Eye Contact:** Shows you are engaged and attentive.
- **Relaxed Posture:** Avoid crossing arms or appearing closed off.
- **Nod and Smile:** Acknowledge the other person's points.
- **Calm Tone:** Speak slowly and avoid raising your voice.
- **Take Deep Breaths:** Helps manage stress and maintain composure.



### Documented Instruction

Documented instruction in software development refers to the detailed manuals, guides, and documentation that provide step-by-step instructions on how to use, install, configure, and troubleshoot software applications and systems.



### How to Read and Accurately Follow Steps in a Web Framework/Plugin Installation Manual

- **Read Thoroughly:** Review the entire manual before starting the installation.
- **Follow Step-by-Step:** Execute each step in the given order without skipping.
- **Cross-Check Requirements:** Ensure all prerequisites are met before proceeding.
- **Refer to Diagrams:** Use visual aids in the manual to better understand the process.

### How to Find Feature Descriptions in the Plugin Framework Documentation

- **Navigate to the Features Section:** Most documentation has a dedicated section for features.
- **Use Search Tools:** Use search functionality to find specific features.
- **Refer to Examples:** Look for example use cases or tutorials in the documentation.

- **Consult FAQs:** Check the frequently asked questions section for additional insights.



**End of Study Session**  
**CLICK! TO RESTART THIS LESSON**



## PRACTICAL EXCERCISE

- In your study session with your instructor, observe how the instructor is using gesture to support verbal communication.
- With the support of your instructor search for the following web development frameworks read through the features from the provided documentation
  - a. WordPress
  - b. Bootstrap
- With the support of your instructor search for five WordPress relevant plugins and read through their understanding.

*Related icon to exercise*





*Now, can you confidently say repeat these statements to yourself:*

*I can:*

- demonstrate the importance of good
- demonstrate ability to follow documented instructions

**CLICK! FOR YOUR NEXT MODULE**



# TEAM WORK

This unit is meant to impact learners with the necessary skills, knowledge and understanding required to develop team spirit and positive working relationship with colleagues.



**At the end of this Module, learners should be able to:**

- demonstrate positive working relation with team member
- take responsibility in a teamwork
- Comply with organization policy



### Positive Working Relations in Teamwork

Teamwork refers to the collaborative effort of a group of individuals working together to achieve a common goal. It involves the coordinated and cooperative actions of team members, leveraging their diverse skills and strengths to enhance productivity and performance.



#### Importance of Developing Positive Working Relationships in a Team

- **Enhances Collaboration:** Positive relationships foster open communication and cooperation.
- **Increases Morale:** A supportive environment boosts team morale and job satisfaction.
- **Improves Productivity:** Good relationships facilitate smoother workflows and efficient task completion.
- **Encourages Innovation:** Trust and mutual respect enable team members to share creative ideas.
- **Reduces Conflict:** Positive relationships help in resolving conflicts amicably and maintaining a harmonious work environment.

### Information Communication in Teamwork

Effective information communication in teamwork is essential for ensuring that all team members are aligned, informed, and able to contribute to the team's objectives. Clear and open communication fosters collaboration, reduces misunderstandings, and enhances overall team productivity. Here are key practices for maintaining effective information communication in a team:

- **Regular Updates:** Keep team members informed about project progress and changes through regular updates. This can be done through scheduled meetings, status reports, or project management tools.
- **Clear Instructions:** Provide concise and understandable directives to ensure everyone knows their tasks and responsibilities. This reduces confusion and increases efficiency.
- **Open Channels:** Encourage open and transparent communication among team members. Use tools like instant messaging, collaboration platforms, and regular check-ins to facilitate ongoing dialogue.

- **Feedback Mechanism:** Implement a system for providing and receiving constructive feedback. This helps in addressing issues promptly and improving processes and outcomes. Regular feedback sessions can be formal or informal, depending on the team's needs.

### Reporting in Teamwork

Effective reporting in teamwork is critical for maintaining transparency, accountability, and coordination among team members. It ensures that everyone is aware of the team's progress, challenges, and milestones, allowing for informed decision-making and timely interventions. Here are key practices for maintaining effective reporting in a team:

1. **Scheduled Meetings:** Hold regular team meetings to discuss progress, challenges, and solutions. These meetings should have a clear agenda, be time-bound, and include opportunities for all members to contribute.
2. **Status Reports:** Submit periodic status reports to keep everyone updated on individual and team performance. These reports should highlight completed tasks, ongoing work, upcoming activities, and any issues encountered.
3. **Documentation:** Maintain thorough documentation of project details, decisions, and outcomes. This ensures that all team members have access to the same information and can refer back to it as needed.
4. **Responsibility Allocation:** Clearly define reporting responsibilities for each team member. This includes who reports what information, to whom, and how frequently. Clear guidelines help prevent overlaps and ensure that all necessary information is covered.



## Responsibility In Teamwork

Understanding and fulfilling responsibilities within a team is crucial for achieving collective goals efficiently and effectively. When team members recognize their roles and responsibilities, it leads to better coordination, accountability, and a stronger sense of ownership over their work. Here are key aspects of recognizing and executing responsibilities in teamwork:



### Recognizing Role and Responsibility in a Team

- **Role Definition:** Clearly define each team member's role based on their skills and expertise. This helps in leveraging individual strengths and ensuring that all necessary functions are covered.
- **Responsibility Assignment:** Assign specific tasks and responsibilities to ensure accountability. Clear assignments help team members understand their duties and what is expected of them.
- **Role Awareness:** Ensure team members understand their roles and how they contribute to the team's objectives. This promotes a sense of purpose and alignment with the team's goals.

### Rules and Regulations of Teamwork

- **Respect and Trust:** Foster a culture of mutual respect and trust among team members. Respect for each other's opinions and contributions is fundamental for effective collaboration.
- **Communication Protocol:** Establish clear communication protocols to avoid misunderstandings. This includes guidelines on how and when to communicate within the team.
- **Conflict Resolution:** Implement strategies for resolving conflicts in a constructive manner. Addressing conflicts promptly and fairly maintains team harmony.
- **Team Meetings:** Regularly schedule and conduct team meetings to discuss progress and address issues. These meetings should be structured to ensure productive discussions.

## How to Participate Effectively in Teamwork

- **Active Listening:** Pay attention to others' ideas and feedback. Active listening shows respect and help in understanding different perspectives.
- **Contribution:** Actively contribute ideas and efforts towards team goals. Engagement and participation from all members drive team success.
- **Reliability:** Be dependable and complete assigned tasks on time. Reliability builds trust and ensures the team meets its deadlines.
- **Flexibility:** Adapt to changing circumstances and be willing to help where needed. Flexibility is essential for navigating unforeseen challenges.
- **Support:** Offer assistance and support to team members when necessary. Providing help fosters a collaborative and supportive team environment.

## Compliance with Organizational Policy

Compliance with organizational policies is essential for maintaining a professional, ethical, and efficient workplace. Adhering to these policies ensures that all team members operate within the established guidelines, which helps to uphold the organization's values and standards. Compliance promotes consistency, accountability, and a safe working environment. Here are key aspects of understanding and adhering to organizational policies



## Organizational Code of Conduct

The code of conduct is a set of guidelines and principles that outline the expected behaviors and responsibilities of employees within the organization.

### Purpose

It aims to ensure a professional, respectful, and ethical work environment.

### Components of Organizational Code of Conduct

Typically includes:

- Policies on integrity
- Confidentiality
- Respect and
- Professionalism

### How to Participate Effectively in Teamwork

- **Active Listening:** Pay attention to others' ideas and feedback. Active listening shows respect and help in understanding different perspectives.
- **Contribution:** Actively contribute ideas and efforts towards team goals. Engagement and participation from all members drive team success.
- **Reliability:** Be dependable and complete assigned tasks on time. Reliability builds trust and ensures the team meets its deadlines.
- **Flexibility:** Adapt to changing circumstances and be willing to help where needed. Flexibility is essential for navigating unforeseen challenges.
- **Support:** Offer assistance and support to team members when necessary. Providing help fosters a collaborative and supportive team environment.

### Working in Line with Organizational Standards

- **Understanding Policies:** Familiarize yourself with the organization's policies and standards.
- **Adhering to Procedures:** Follow established procedures and protocols in your daily tasks.
- **Ethical Behavior:** Maintain high ethical standards in all professional interactions.
- **Compliance Training:** Participate in training sessions to stay updated on organizational policies.
- **Reporting Violations:** Know the process for reporting any violations of the code of conduct.



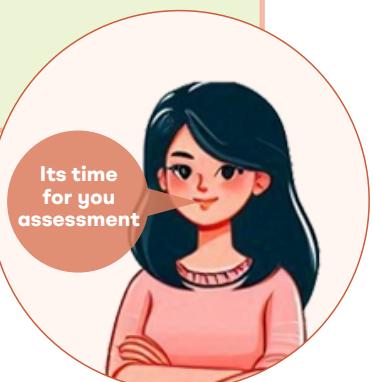
**End of Study Session**

**CLICK! TO RESTART THIS LESSON**



## PRACTICAL EXCERCISE

- Instructor divide the class session into groups
- Assign specific tasks to each group
- Assign the responsibility of selecting a group leader to each group
- From this time to the end of the training:
  - Maintain the group created in solving in class and out of class task
  - Engage the group leaders in a briefing meeting before or at the end of each class for any information or instruction for the group.





*Now, can you confidently say repeat these statements to yourself:*

*I can:*

- demonstrate positive working relation with team member
- take responsibility in a teamwork
- Comply with organization policy

**CLICK! FOR YOUR NEXT MODULE**

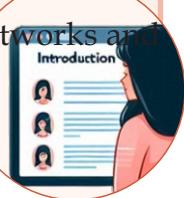
# USE OF COMPUTER FOR WEB DEVELOPMENT

This unit aims to equip learners with the skills to effectively use the Internet for web development, understand essential software and hardware, and grasp how computer networks function. It covers identifying different types of computer networks and their roles, providing a comprehensive foundation for web development.



## At the end of this Module, learners should be able to:

- The learners will utilize the Internet efficiently for web development tasks.
- The learners will explain the functioning of computer networks in the context of web development.
- The learners will identify and describe various types of computer networks and their specific functions in web development etc.



### Effective Use Of The Internet In Web Development

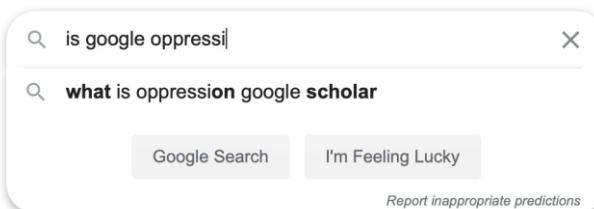
#### What Is Internet?

The Internet is a vast, interconnected network that links millions of private, public, academic, business, and government computers across the globe. This network enables computers to communicate and share information through standardized protocols, allowing users to access a wide range of resources, such as websites, emails, and files, regardless of their location.



#### What is a Search Engine:

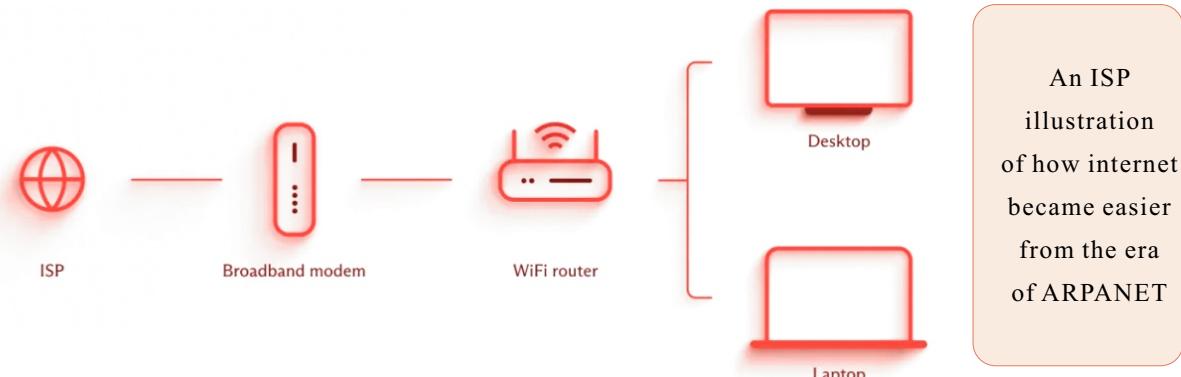
A search engine is a powerful tool designed to help users find information on the Internet quickly and efficiently. By entering keywords or phrases into the search engine, users can retrieve a list of relevant websites, documents, images, videos, and other types of content. Popular search engines like Google, Bing, and Yahoo use complex algorithms to rank and display the most relevant results based on the user's query.



#### History of the Internet

**Pre-Internet (1960s-1980s):** The roots of the Internet can be traced back to early networking experiments like ARPANET, which was developed by the U.S. Department of Defense to connect various research institutions. During this period, basic networking protocols and technologies were established, laying the groundwork for future development.

**Early Internet (1990s):** The Internet became accessible to the general public, leading to the rise of commercial Internet service providers (ISPs). This era saw the introduction of web browsers, such as Netscape Navigator, which made it easier for users to navigate the web. Email became a popular means of communication, and the World Wide Web Consortium (W3C) was founded to develop web standards.



**Web 2.0 (2000s):** This period marked the emergence of social media platforms like Facebook, Twitter, and YouTube, which allowed users to create and share content interactively. Websites became more dynamic and user-friendly, enabling richer user experiences. The term "Web 2.0" refers to this shift towards greater user participation and collaboration on the web.

**Mobile Internet (2010s):** The widespread adoption of smartphones and tablets revolutionized how people access the Internet. Mobile applications (apps) became prevalent, and connectivity improved with the advent of 4G and later 5G networks. This era emphasized the importance of mobile-friendly websites and services.



An illustration showing link between WEB2 and Apps



**Future Trends (2020s and beyond):** The future of the Internet is being shaped by advancements in artificial intelligence (AI), the Internet of Things (IoT), 5G technology, and beyond. These innovations promise to enhance connectivity, enable smarter devices, and offer new ways of interacting with the digital world, such as through augmented reality (AR) and virtual reality (VR).

**End of Study Session**  
**CLICK! TO RESTART THIS LESSON**



## PRACTICAL EXERCISE

In a collaborative session with peers and guided by an instructor, learners will:

1. Use search engines for basic and advanced search techniques (keywords, filters, Boolean operators).
2. Create and manage an email account, practice sending/receiving emails, and organize the inbox.
3. Set up a social media profile, manage privacy settings, and explore basic features (posting, connecting).
4. Navigate the internet safely, identify online risks, use secure browsing practices (HTTPS, privacy settings), and recognize phishing attempts.
5. Practice advanced skills such as using search operators, composing emails with attachments, and configuring detailed social media privacy settings.



### Additional Resources:

Video Links: <https://www.youtube.com/watch?v=UMpEkZxeVcY>

Credit: [@blastertechnology](#)

### Software And Its Evolution

#### What Is Software?

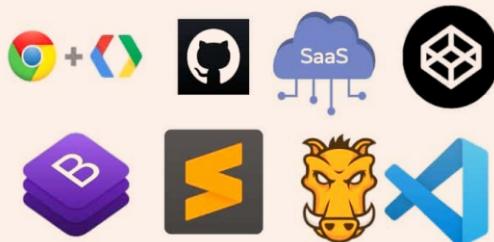
Software refers to the programs, instructions, and data that enable computers to perform specific tasks or functions. It encompasses both applications, like word processors and web browsers, and system software, such as operating systems and device drivers.

#### TYPES OF SOFTWARES

##### SYSTEM SOFTWARE

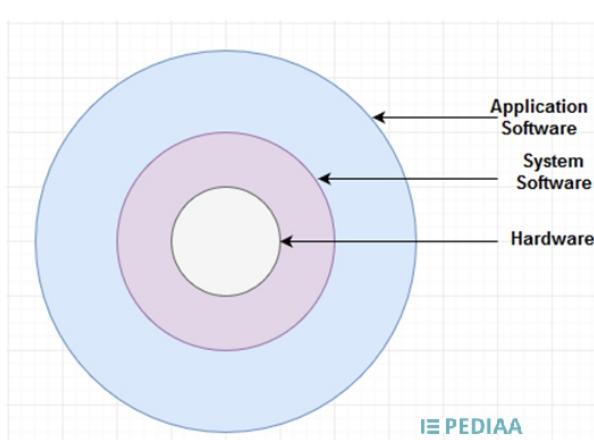


##### APPLICATION SOFTWARE



#### History of Software Evolution:

The history of software evolution traces back to the early days of computing when programs were hand-coded and stored on punch cards. Over time, advancements in programming languages, software engineering methodologies, and computing technologies led to the development of more sophisticated software systems, including graphical user interfaces (GUIs), multimedia applications, and cloud-based services.



#### Types of Software

Software can be categorized into two main types: application software and system software. Application software includes programs designed for specific tasks, such as word processing, spreadsheet management, and graphic design. System software comprises essential programs that manage computer hardware and provide a platform for running applications, including operating systems, device drivers, and utility software.

### Functionality of Software

Understanding how to work with software involves tasks such as installation and implementation. Installation involves setting up software on a computer system, which typically includes downloading or inserting installation media, following prompts, and configuring settings. Implementation involves using the software to perform desired tasks effectively, which may require customization, training, and troubleshooting.

### Understanding hardware

#### What Is Hardware?

Hardware refers to the physical components of a computer system, including the central processing unit (CPU), memory (RAM), storage devices (hard drives, SSDs), input devices (keyboard, mouse), output devices (monitor, printer), and networking devices (routers, switches).

### TYPES OF COMPUTER HARDWARE

01



1. **Cooling Fan:** Used to cool the computer by circulating air to prevent overheating of internal components.

01



1. **Microphone:** Used to input audio into the computer.

01



1. **Speakers:** Output device for producing sound from the computer.

01



1. **Webcam:** Used for capturing video and images, often used for video conferencing.

01



1. **Printer:** Output device that produces hard copies of documents and images.

### TYPES OF COMPUTER HARDWARE (cont'd)

01



1. **SD Card:** Storage device used to store data like photos, videos, and documents.

01



1. **Raspberry Pi:** A small, affordable computer used for learning programming and building hardware projects.

01



1. **CPU (Central Processing Unit):** The brain of the computer where most calculations take place.

01



1. **Monitor:** Output device that displays information in visual form.

01



1. **Keyboard:** Input device used to enter text and commands into the computer.

01



1. **Joystick:** Input device used for gaming and controlling applications.

01



1. **External Hard Drive:** Portable storage device used to store and back up data.

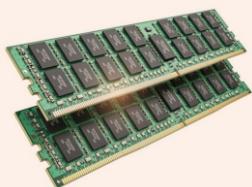
### TYPES OF COMPUTER HARDWARE (cont'd)

**01**



1. **Mouse:** Input device used to interact with the computer's graphical user interface.

**01**



1. **RAM (Random Access Memory):** Temporary storage used by the computer to store data that is being used and processed.

**01**



1. **Headphones:** Output device worn on the ears to listen to audio privately.

**01**



1. **USB Flash Drive:** Portable storage device used for data transfer and storage.

**01**



1. **Graphics Card (GPU):** Hardware used to render images, video, and animations, particularly important for gaming and video editing.

### HISTORY OF COMPUTERS



#### History of Hardware Evolution:

The evolution of hardware dates back to the invention of mechanical calculating devices and early electronic computers. Over time, advancements in semiconductor technology, miniaturization, and manufacturing processes led to the development of more powerful, compact, and energy-efficient hardware components, including microprocessors, memory chips, and storage devices.

## Hardware Components

Hardware components can be classified into input devices (e.g., keyboard, mouse, scanner) and output devices (e.g., monitor, printer). Input devices allow users to input data and commands into a computer system, while output devices display or provide output from the computer system.

### Importance of Hardware in a Computer System:

Hardware forms the foundation of a computer system, providing the physical infrastructure necessary for processing data, executing programs, and interacting with users. Each hardware component plays a crucial role in the overall functionality and performance of the computer system.

#### Scanner

**Function:** An input device that converts physical documents into digital format.

#### Monitor

**Function:** An output device that displays information in visual form, allowing the user to interact with the computer.

#### External Hard Drive

**Function:** A portable storage device used to store and back up data.

#### Webcam:

**Function:** An input device used for capturing video and images, often used for video conferencing.

#### Printer:

**Function:** An output device that produces hard copies of documents and images from digital form.

#### Graphics Tablet:

**Function:** An input device used primarily for digital drawing and graphic design.

#### Desktop Computer Tower:

**Function:** The main unit of the computer containing the CPU, memory, and other critical components that perform processing tasks.

#### Keyboard:

**Function:** An input device used to enter text, numbers, and commands into the computer.

#### Mouse:

**Function:** An input device used to interact with the computer's graphical user interface by controlling the movement of the cursor on the screen.



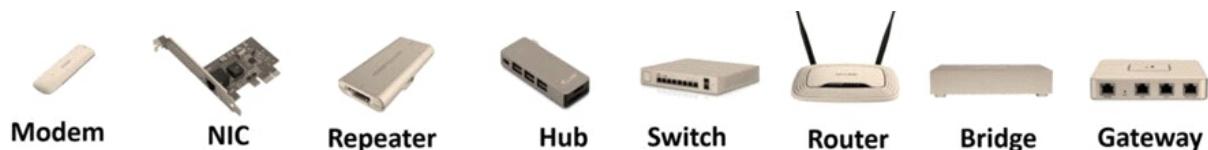
### Understanding hardware

#### What Is Hardware Networking?

Hardware networking involves the use of physical devices to connect computers and other devices within a network, allowing them to communicate and share resources. Networking hardware includes routers, switches, hubs, gateway devices, cables, adapters, servers, and firewalls.

#### Appropriate Networking Hardware

Networking hardware includes routers, which direct data traffic between different networks; switches, which connect multiple devices within a local area network (LAN); hubs, which distribute network signals to multiple devices; gateway devices, which enable communication between different types of networks; cables and adapters for physical connections; servers for hosting network services and resources; and firewalls for network security.



#### Difference Between Software and Hardware:

Software refers to programs, instructions, and data that enable computers to perform specific tasks, while hardware refers to the physical components of a computer system. Software interacts with hardware to execute instructions and manipulate data, whereas hardware provides the physical infrastructure for software to run.



**End of Study Session**  
CLICK! TO RESTART THIS LESSON



## PRACTICAL EXERCISE

In a collaborative session with peers and guided by an instructor, learners will:

1. Engage in hands-on activities to install and implement various software applications on computer systems. Follow step-by-step instructions to download, install, and configure software, and troubleshoot common installation issues.
2. Participate in practical exercises to identify and connect different hardware parts, such as storage drives, printer and mouse. Practice proper handling and installation techniques while following safety guidelines. Gain practical experience in troubleshooting hardware issues and optimizing system performance.



### Additional Resources:



Video Link: <https://www.youtube.com/watch?v=xnyFYiK2rSY>

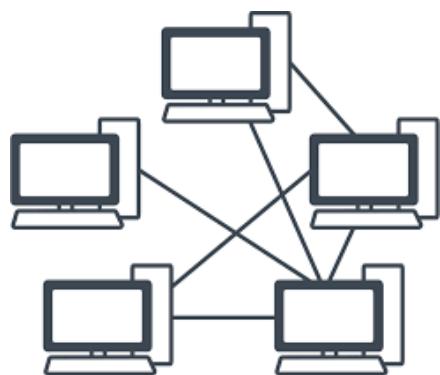
Credit: code.org

Video Link: <https://www.youtube.com/watch?v=DuYF3uD9AYQ>

Credit: [@makeiteasyeducation](#)

### What Is Computer Network?

A computer network is a system of interconnected computers and other devices that communicate and share resources, such as files, printers, and internet connections. Networks can vary in size from a small local area network (LAN) in a home or office to a vast wide area network (WAN) that spans the globe.



#### Basic Components of a Computer Network:

The essential components of a computer network include:

##### ● **Routers**

Routers help direct internet traffic between different networks. They act as a central hub where multiple networks connect, allowing devices in those networks to communicate with each other and access the internet.

##### ● **Modems**

Modems are devices that connect your home to the internet. They take digital signals from your computer, turn them into signals that can travel over phone or cable lines, and then convert them back to digital signals for your computer to use.

##### ● **Switches**

Switches connect multiple devices within a local network, like computers and printers in an office. They make the network more efficient by ensuring that data gets sent directly to the right device without causing slowdowns.

##### ● **Hubs**

Hubs connect several devices in a network so they can communicate. However, they are now considered outdated because they are less efficient than switches.

##### ● **Access Points**

Access points let wireless devices, like smartphones and laptops, connect to a wired network. They broadcast a Wi-Fi signal that allows these devices to access the network without needing cables.

##### ● **Network Interface Cards (NICs)**

NICs are hardware components that let computers connect to a network. They provide a way for computers to send and receive data over a network.

##### ● **Repeaters**

Repeaters extend the range of a network by boosting the signal, ensuring that data can travel further without losing quality.

##### ● **Bridges**

Bridges connect two different parts of a network, helping to manage traffic and improve performance by filtering and directing data.

##### ● **Wireless Adapters**

Wireless adapters let devices like laptops and desktop computers connect to a Wi-Fi network, enabling internet access without the need for cables. They can be built-in or come as USB dongles or PCI cards.

These devices are essential for creating, managing, and extending network connections, ensuring efficient and secure communication within and between networks.

## NETWORK TOPOLOGIES

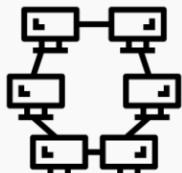
Network topology refers to the arrangement of different elements (links, nodes, etc.) in a computer network. Common network topologies include:

01



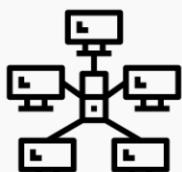
**Bus Topology:** All devices are connected to a single central cable, called the bus. Data travels in both directions along the bus.

02



**Ring Topology:** Each device is connected to two other devices, forming a circular data path. Data travels in one direction around the ring.

03



**Star Topology:** All devices are connected to a central hub or switch. Data passes through the central device before reaching its destination.

04



○ **Mesh Topology:** Devices are interconnected, with each device connected to several other devices. This provides multiple paths for data to travel.

### Basic Components of a Computer Network:

**Network Interface Card (NIC):** Allows a computer to physically connect to a network.

**Hub:** Connects multiple network devices, broadcasting data to all devices in the network.

**Switch:** Connects devices within a network and uses MAC addresses to forward data to the correct destination.

**Router:** Connects different networks, directing data between them, and often provides Internet connectivity.

**Network Cables:** Physical media that transmit data between devices within a network.

**End of Study Session**

[CLICK! TO RESTART THIS LESSON](#)



## PRACTICAL EXERCISE

In a collaborative session with peers and guided by an instructor, learners will:

Participate in hands-on activities to set up and configure a basic computer network. This includes connecting devices using network cables, and configuring routers. Gain practical experience in setting up network topologies and understanding their advantages and disadvantages.



### Additional Resources:

Video Links: <https://www.youtube.com/watch?v=tSodBEAJz9Y>

<https://www.youtube.com/watch?v=eMamgWlIRFY>

Credit: [CertBros](#)



Video Links: <https://www.youtube.com/watch?v=uuOytXJhuk0>

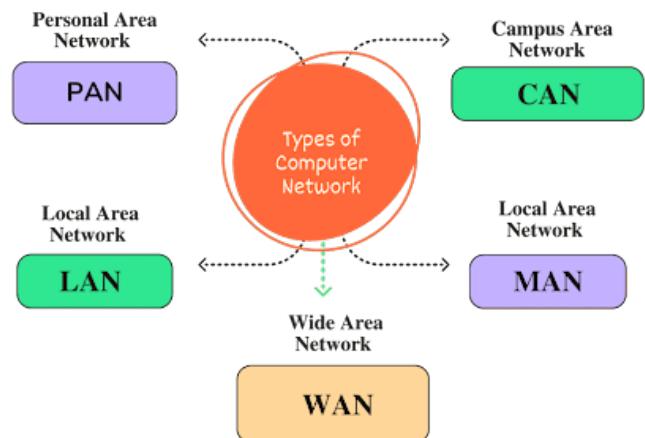
Credit: [IT-Made-Easy](#)

<https://www.youtube.com/watch?v=yYe6Mh9fig0>

Credit: [Simply Coding](#)

## Types Of Computer Networks And Their Roles in Web Development

In the ever-evolving landscape of technology, computer networks play a crucial role in facilitating communication, data exchange, and resource sharing among devices. Understanding the various types of computer networks and their specific configurations is fundamental to mastering web development.



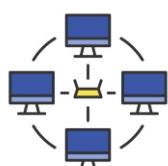
**01**



### Personal Area Network (PAN):

Connects personal devices such as smartphones, tablets, and laptops within a small area, typically using Bluetooth or USB connections.

**02**



### Local Area Network (LAN):

Connects computers and devices within a single building or a small group of buildings. Used for sharing resources like files, printers, and Internet connections within a limited area.

**03**



### Campus Area Network (CAN):

Connects multiple LANs within a campus or corporate facility. Used to link departments, buildings, or labs within a specific area, facilitating resource sharing and communication.

**04**



### Metropolitan Area Network (MAN):

Spans a city or a large campus, connecting multiple LANs and CANs. Often used by organizations to connect multiple branches or facilities within the same metropolitan area.

**05**



### Wide Area Network (WAN):

Covers a broad geographic area, often a country or continent. Used by businesses and organizations to connect regional offices and allow data communication over long distances.

### Virtual Private Network (VPN):

Creates a secure network connection over the Internet, allowing remote users to access a private network. Often used for secure communication and data transfer between remote offices and mobile workers.



By grasping the intricacies of these network types, web developers can optimize the performance, scalability, and reliability of their web applications, ensuring seamless connectivity and enhanced user experiences.

### Differences Between Different Types of Computer Networks

#### SCOPE AND RANGE:

PANs cover the smallest range (a few meters), while

WANs cover the largest (entire countries or continents).

LANs and CANs cover intermediate ranges, such as single buildings or campuses, respectively.

MANs cover cities or metropolitan areas.

#### PURPOSE AND USE CASE:

PANs are for personal device connectivity,

LANs are for localized resource sharing,

CANs link buildings within a campus, MANs connect networks within a city,

WANs connect geographically dispersed networks, and

VPNs provide secure remote access over the Internet.

#### COMPLEXITY AND COST:

PANs and LANs are relatively simple and inexpensive to set up.

CANs and MANs require more infrastructure and management.

WANs are the most complex and costly due to their vast coverage and need for extensive hardware and network management.

VPNs add complexity in terms of security and encryption but offer cost-effective remote connectivity.

End of Study Session  
CLICK! TO RESTART THIS LESSON



## PRACTICAL EXERCISE

*In a collaborative session with peers and guided by an instructor, learners will:*

Set up and configure three types of networks. This includes creating a PAN using Bluetooth devices, setting up a LAN in a small office and practice establishing a secure VPN connection for remote access.



### Additional Resources:



Video Links: <https://www.youtube.com/watch?v=LQEczD939jI>

Credit: [Learning DigiTech](#)

Video Links: [https://www.youtube.com/watch?v=4\\_zSIXb7tLQ](https://www.youtube.com/watch?v=4_zSIXb7tLQ)

<https://www.youtube.com/watch?v=zbqrNg4C98U>

Credit: PowerCert Animated Videos



*Now, can you confidently say repeat these statements to yourself:*

**I can:**

- Utilize the Internet efficiently for web development tasks.
- Understand and apply knowledge of essential web development software.
- Recognize the necessary hardware components for web development.
- Explain the functioning of computer networks in the context of web development.
- Identify and describe various types of computer networks and their specific functions in web development.

**CLICK! FOR YOUR NEXT MODULE**

A graphic of an orange computer cursor arrow pointing downwards, positioned below the text "CLICK! FOR YOUR NEXT MODULE".

# THE WORLD WIDE WEB (WWW)

This unit aims to provide a clear understanding of the World Wide Web (WWW), including its historical development, types of websites, web browsers, web threats and security measures, domain registration, web hosting, and e-commerce.



## At the end of this Module, learners should be able to:

- Demonstrate knowledge of the categories of websites.
- Identify different web browsers and their functionalities.
- Demonstrate knowledge of the different types of websites.
- Recognize common web threats and implement security measures.
- Highlight the process of domain registration and web hosting.
- Demonstrate knowledge of e-commerce principles and practices.



### Eras and Trends of the Web

#### History of the World Wide Web (WWW)

The World Wide Web, invented in 1989 by Tim Berners-Lee at CERN, revolutionized information sharing on the internet. Initially a tool for scientists to access and link research resources, the first web browser and server were created in 1990. By 1991, the first website was launched, explaining the web project. Its ease of use led to rapid adoption in the mid-1990s with browsers like Mosaic and Netscape Navigator. The web quickly expanded beyond academia to encompass diverse topics, transforming global communication, learning, work, and entertainment. Today, billions use the web daily on various devices, making it an integral part of modern life.



#### Websites vs Webpages

The table below captures the key differences between websites and webpages, highlighting their definitions, structures, content, examples, publication methods, purposes, and thematic roles.

<b>W E B P A G E</b>	<b>STRUCTURE</b>	<b>CONTENT</b>	<b>EXAMPLES</b>	<b>PUBLICATIONS</b>
A collection of related web pages identified by a common domain name.				
Provides an overall structure and context.				
Educational websites, news websites, online forums, e-commerce platforms.				
Published on at least one web server.				



### WEB PAGE

Serves as the complete publication for various topics and purposes.

Yes, it has an organizational structure and common theme.

[www.example.com](http://www.example.com) (overall website)

### PURPOSE

#### COMMON THEME

#### EXAMPLE IN CONTEXT



### WEBSITE

Provides detailed content and information on specific topics.

No, each page is specific to its content but part of the overall theme.

[www.example.com/article1](http://www.example.com/article1) (specific webpage within the website).

## Stages of Web Development

The World Wide Web has evolved significantly since its inception. Its development can be broadly categorized into three generations:

### Web 1.0

Static Web



Web 1.0, the first generation of the internet from 1991 to 2004, featured static websites with basic HTML pages that allowed users to view and read information but offered limited interaction. Its primary function was to provide information for consumption, similar to traditional media like newspapers and television, with examples including simple text-and-image websites.

### Web 2.0

Dynamic Web



Starting around 2004, Web 2.0 introduced a more interactive and collaborative internet, driven by social media, blogs, wikis, and other platforms that emphasize user participation and sharing. Users could not only consume content but also create, modify, and critique it. Popular examples include social networks, media sharing sites, online communities, and collaborative knowledge bases.

### Web 3.0

Semantic Web



This era, Web 3.0, focuses on making the web more intelligent and intuitive through AI and machine learning, allowing for personalized user experiences. It aims to blur the lines between the digital and physical worlds, integrating data and services for richer experiences, though it remains an active area of research and development.

## Eras and Trends of the Web

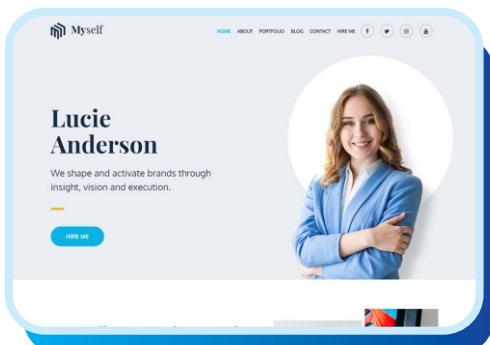
### Types and Categories of Websites

The terms "types" and "categories" of websites are often used interchangeably, but they can have slightly different connotations.

Types of websites typically refer to the fundamental underlying structure, technology, or purpose of a website. For example, the two main types of websites are static and dynamic websites. This classification is based on how the website delivers and updates its content.

Types of Websites		Categories of Websites
Static	Dynamic	
<ul style="list-style-type: none"> <li>• Simple</li> <li>• Pre-built</li> <li>• No interactivity</li> <li>• HTML,CSS, Basic JS</li> <li>• Brochures</li> <li>• Portfolios</li> </ul>	<ul style="list-style-type: none"> <li>• Complex</li> <li>• Real-time updates</li> <li>• Interactive</li> <li>• Uses databases</li> <li>• Server-side</li> <li>• Languages</li> <li>• E-commerce sites</li> <li>• Social media</li> <li>• Online banking</li> </ul>	E-commerce Educational News & Media Social Media Business Entertainment Government, etc.

Categories of websites, on the other hand, usually refer to a broader grouping or classification based on the website's specific content, purpose, or target audience. **For example, some common categories of websites include:**



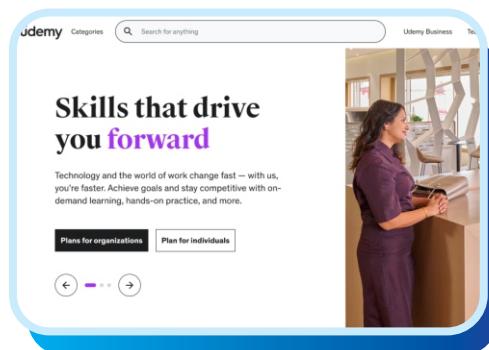
#### Personal Websites:

These are websites created by individuals to share personal information, thoughts, experiences, and creative works. Common examples are personal blogs, portfolios showcasing an artist/photographer's work, and self-promotion sites.



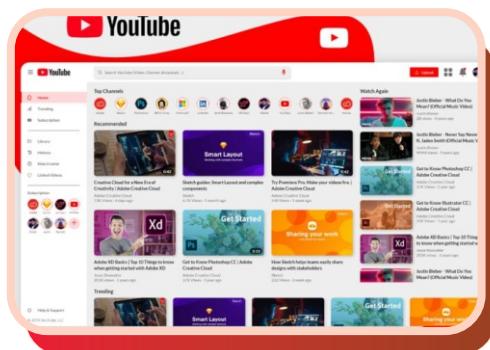
## COMMERCIAL WEBSITES:

Businesses and companies create commercial websites to advertise, market, and sell their products or services online. This includes e-commerce stores, company branding sites, and sites that allow online bookings/reservations.



## EDUCATIONAL WEBSITES:

These websites focus on providing educational content, resources, courses, and training materials. Examples are online learning platforms, tutorial sites, digital libraries, and official school/university websites.



## ENTERTAINMENT WEBSITES:

Entertainment websites are designed for online recreation and amusement. This category includes video/music streaming services, online gaming sites, humor/comedy sites, and fan communities.



## INFORMATIONAL WEBSITES:

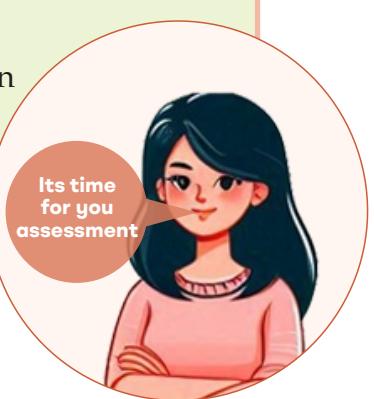
The main purpose of informational websites is to provide helpful facts, data, instructions, and general knowledge on various topics of interest. News sites, encyclopedias, how-to guides, and research databases fall into this category.

While many websites may incorporate elements of multiple categories, most can be primarily categorized by their main function and type of content offered to users. Understanding these website types provides clarity on the diverse range of information and services available on the World Wide Web.

## PRACTICAL EXERCISE

*In a collaborative session with peers and guided by an instructor, learners will:*

- Create a simple timeline to show the evolution of the web and identify key milestones.
- Compare websites from each era, noting differences in design and functionality.
- Understand practical differences between a web page and a website.



### Additional Resources:

Video Links: <https://www.youtube.com/watch?v=2uToPjU-ODI>



Credit: [LearnAwesome](#)

Video Links: [https://www.youtube.com/watch?v=RX\\_6nM11wGs](https://www.youtube.com/watch?v=RX_6nM11wGs)

Credit: Sky Tech

Video Links: <https://www.youtube.com/watch?v=X6DIH7b82Qw>

Credit: by ChaseJin

### Introduction To Web Browsers

Web browsers are software applications that allow users to access and view websites on the Internet. They interpret HTML and other web technologies to display web pages.

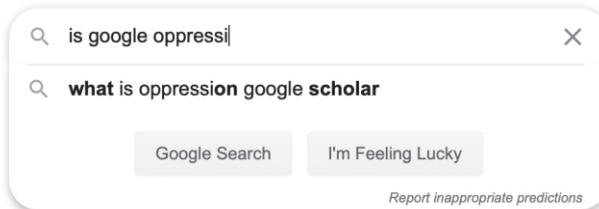
Understanding different web browsers and their functionalities is essential for effective web development and internet usage.

### Types of Web Browsers

- **Google Chrome**

**Description:** Google Chrome is known for its speed, simplicity, and extensive range of extensions. It is widely used and offers robust security features.

**Key Features:** Fast performance, synchronization with Google services, **extensive** library of extensions, built-in developer tools.



- **Mozilla Firefox**

**Description:** Mozilla Firefox is valued for its privacy features, customization options, and strong performance.

**Key Features:** High privacy standards, customizable interface, wide range of add-ons, developer-friendly tools.



- **Microsoft Edge**

**Description:** Microsoft Edge is integrated with Windows and offers good performance and compatibility, especially for enterprise environments.

**Key Features:** Fast and secure browsing, integration with Microsoft services, built-in reading mode, collection of productivity tools.



- **Safari**

**Description:** Safari is Apple's web browser, optimized for macOS and iOS devices, providing fast and energy-efficient browsing.

**Key Features:** Optimized for Apple devices, energy-efficient performance, strong privacy protections, seamless integration with the Apple ecosystem.

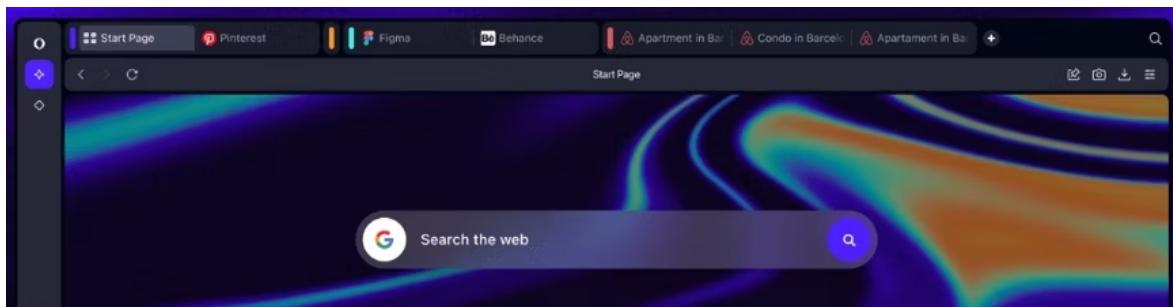


Safari Browser  
Mac OS X Yosemite

- **Opera**

**Description:** Opera offers unique features like a built-in VPN, ad blocker, and a battery saver mode, making it a versatile browser for various needs.

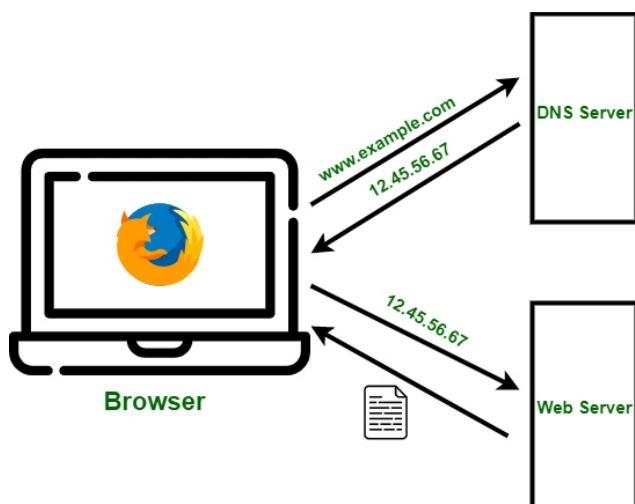
**Key Features:** Built-in VPN and ad blocker, customizable interface, battery saver mode, free access to various content through the browser.



### Functions of Web Browsers

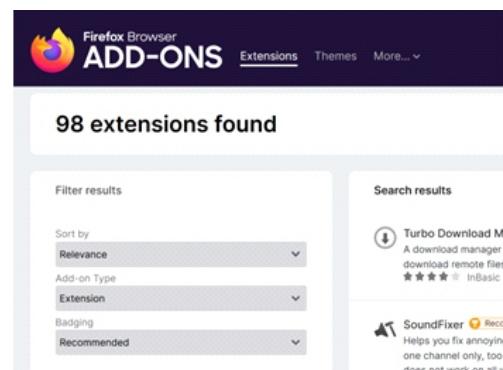
#### 1. Rendering Web Pages:

Web browsers render HTML, CSS, and JavaScript to display web pages. They fetch the necessary resources from web servers and assemble them to create the final user experience.



## 1. Managing Extensions and Add-ons

Extensions and add-ons enhance the functionality of web browsers by adding new features or improving existing ones. Users can install these to customize their browsing experience.



## 1. Ensuring Security and Privacy

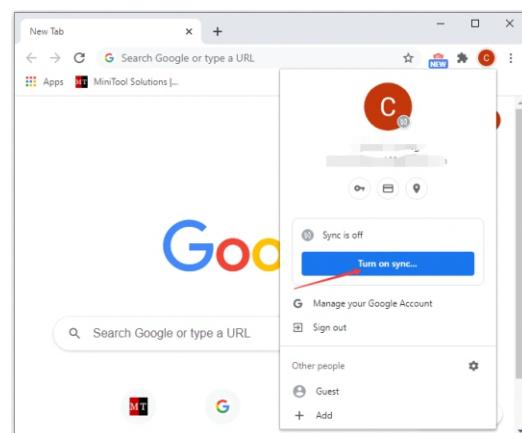
Modern web browsers incorporate various security features such as HTTPS, sandboxing, and phishing protection to safeguard user data and privacy.

Built-in features	Brave	Chrome	DDG	Edge	Firefox	Safari
Invasive ads blocked	✓	🟡	✓	✗	✗	✗
Cross-site trackers blocked	✓	✗	✓	🟡	🟡	✗
Cookies blocked	✓	✗	✓	🟡	✓	✓
Fingerprinting blocked (cookie-less trackers)	✓	✗	✗	✗	✗	✗
Malware & phishing protection*	✓	✓	✗	✓	✓	✓
Protections against bounce tracking	✓	✗	✗	✗	✗	✗
Anonymized network routing (Tor mode)	✓	✗	✗	✗	✗	✗

\* Google safe browsing + malware & phishing protection from crowd-sourced lists. Table describes default desktop browser configurations.

## 1. Synchronizing Data

Web browsers can synchronize bookmarks, history, passwords, and settings across multiple devices using cloud services. Data synchronization is the ongoing process of synchronizing data between two or more devices and updating changes automatically between them to maintain consistency within systems. This feature ensures a seamless browsing experience regardless of the device used.



## PRACTICAL EXCERCISE

*In a collaborative session with peers and guided by an instructor, learners will:*

- **Installing a Web Browser (e.g., Google Chrome):**

1. Download the browser installer from the official website.
2. Run the installer and follow the on-screen instructions to complete the installation.

- **Configuring Essential Settings:**

1. Set the homepage:
  - Open the browser settings.
  - Navigate to the "On startup" section.
  - Choose "Open a specific page or set of pages" and enter the desired URL.
2. Set the default search engine:
  - Open the browser settings.
  - Navigate to the "Search engine" section.
  - Select the preferred search engine from the drop-down menu.
3. Configure privacy preferences:
  - Open the browser settings.
  - Navigate to the "Privacy and security" section.
  - Adjust settings for cookies, site data, and permissions.

- **Exploring Browser Features:**

1. Open new tabs:
  - Click the "+" icon next to the open tabs.
2. Use incognito mode:
  - Open the menu (three dots in the upper-right corner).
  - Select "New incognito window."
3. Manage bookmarks:
  - Open the menu.
  - Select "Bookmarks" and then "Bookmark manager."
  - Add, organize, and edit bookmarks.
4. Install extensions:
  - Open the menu.
  - Select "More tools" and then "Extensions."
  - Visit the Chrome Web Store to browse and install extensions.



### Web Threats and Security Measures

Web threats are malicious activities that can harm computers, steal sensitive information, or disrupt online services. Understanding these threats and the measures to counter them is crucial for safe web browsing and web development.

#### Common Web Threats

##### 1. Malware

Malware is malicious software designed to infiltrate, damage, or disable computers. It includes viruses, worms, trojans, ransomware, and spyware. Malware can corrupt files, steal personal information, monitor user activity, and disrupt system performance.

**Example:** A virus that deletes or encrypts files on a computer, demanding a ransom to restore access.



##### 2. Phishing

Phishing involves fraudulent attempts to obtain sensitive information, such as usernames, passwords, and credit card details, by pretending to be a trustworthy entity. Victims of phishing can suffer identity theft, financial loss, and unauthorized access to their accounts.

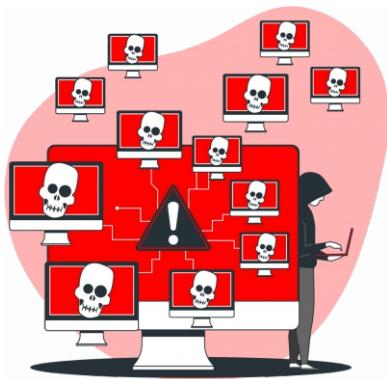
**Example:** An email that appears to be from a bank, asking the recipient to click a link and enter their account details.



##### 3. DDoS Attacks

Distributed Denial of Service (DDoS) attacks involve overwhelming a website with massive amounts of traffic from multiple sources, causing it to crash or become inaccessible. DDoS attacks can disrupt online services, leading to downtime, loss of revenue, and damage to reputation.

**Example:** An e-commerce site being flooded with traffic, making it impossible for legitimate users to make purchases.



### Web Threats and Security Measures

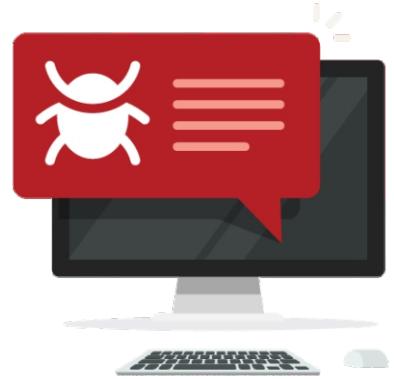
Web threats are malicious activities that can harm computers, steal sensitive information, or disrupt online services. Understanding these threats and the measures to counter them is crucial for safe web browsing and web development.

#### Common Web Threats

##### 1. Malware

Malware is malicious software designed to infiltrate, damage, or disable computers. It includes viruses, worms, trojans, ransomware, and spyware. Malware can corrupt files, steal personal information, monitor user activity, and disrupt system performance.

**Example:** A virus that deletes or encrypts files on a computer, demanding a ransom to restore access.



##### 2. Phishing

Phishing involves fraudulent attempts to obtain sensitive information, such as usernames, passwords, and credit card details, by pretending to be a trustworthy entity. Victims of phishing can suffer identity theft, financial loss, and unauthorized access to their accounts.

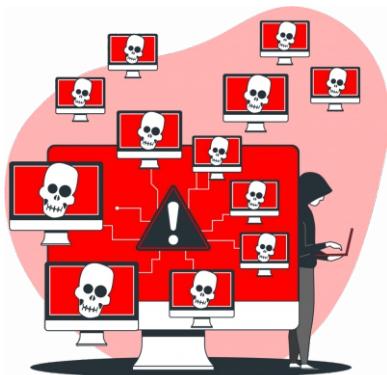
**Example:** An email that appears to be from a bank, asking the recipient to click a link and enter their account details.



##### 3. DDoS Attacks

Distributed Denial of Service (DDoS) attacks involve overwhelming a website with massive amounts of traffic from multiple sources, causing it to crash or become inaccessible. DDoS attacks can disrupt online services, leading to downtime, loss of revenue, and damage to reputation.

**Example:** An e-commerce site being flooded with traffic, making it impossible for legitimate users to make purchases.



## Security Measures to Counter Web Threats

### 1. Antivirus Software

Antivirus software scans and protects computers from malware by detecting and removing malicious programs.

- **Function:** Provides real-time protection, scheduled scans, and updates to defend against the latest threats.
- **Popular Options:** Norton, McAfee, Bitdefender, and Kaspersky.

### 2. Firewalls

Firewalls monitor and control incoming and outgoing network traffic based on predetermined security rules.

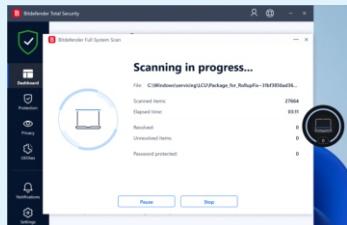
- **Function:** Acts as a barrier between a trusted network and untrusted external networks, blocking unauthorized access.
- **Types:** Hardware firewalls (standalone devices) and software firewalls (installed on computers).

### 3. SSL Certificates

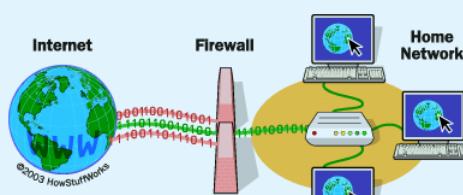
Secure Sockets Layer (SSL) certificates encrypt data transferred between a user's browser and a web server, ensuring secure communication.

- **Function:** Protects sensitive information, such as login credentials and payment details, from being intercepted by attackers.
- **Implementation:** Websites with SSL certificates display "https://" in the URL and a padlock icon in the address bar.

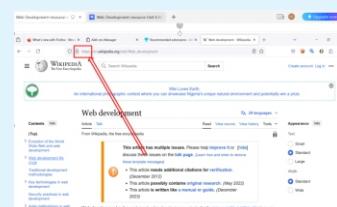
Antivirus Software



Firewall



SSL Certificate



## PRACTICAL EXCERCISE

**· Identify and Avoid Phishing Attempts:**

- Recognize suspicious emails.
- Verify website authenticity.

**· Install and Configure Antivirus Software:**

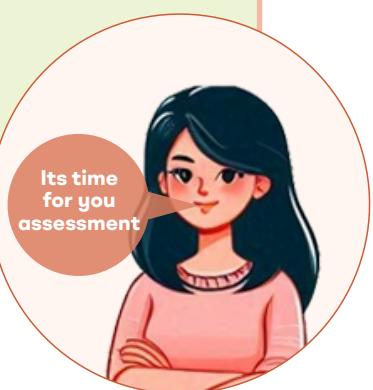
- Perform system scans.
- Remove malware.

**· Set Up and Manage Firewalls:**

- Monitor network traffic.
- Block unauthorized access.

**· Implement SSL Certificates:**

- Secure data transmission on websites.



**Additional Resources:**



Video Links: <https://www.youtube.com/watch?v=Dk-ZqQ-bfy4>

Credit: edureka!

Video Links: <https://www.youtube.com/watch?v=88-FENio9Yw>

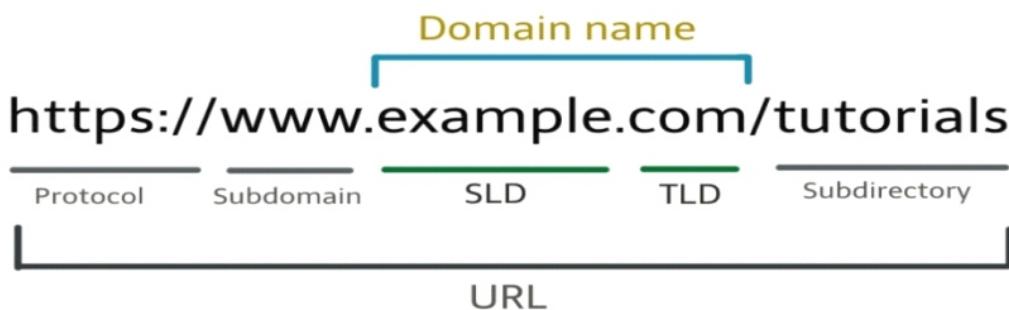
Credit: Kaspersky

### Domain Registration and Web Hosting

For any website to be accessible on the internet, it must have a registered domain name and be hosted on a server. Domain registration involves securing a unique name for the website, while web hosting provides the space and resources needed to store website files and serve them to visitors.

#### Domain Registration

A domain is a unique address that people use to access a website, such as [www.example.com](https://www.example.com). It represents the website's online identity. Domains consist of a name (e.g., "example") and a top-level domain (TLD) such as .com, .org, or .net.



I

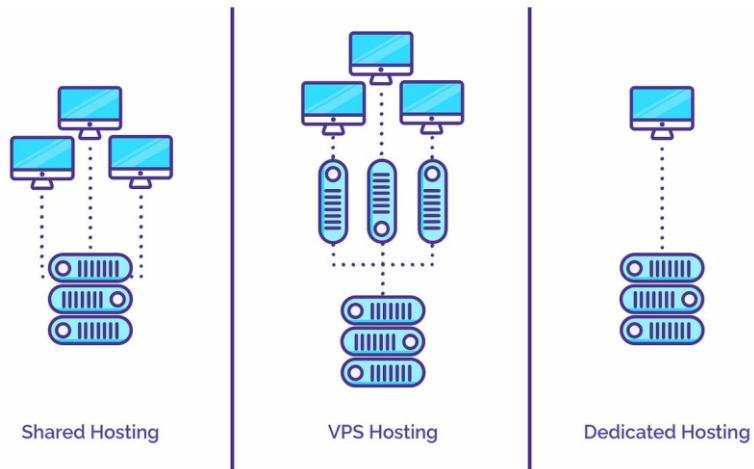
1. **Process:** Choose a domain name, check its availability through a domain registrar, and purchase it for a specified period.
2. **Domain Registrars:** Popular registrars include GoDaddy, Namecheap, and Google Domains.
3. **Considerations:** Choose a name that is easy to remember, relevant to the website's purpose, and preferably with a popular TLD.

### Web Hosting

Web hosting is a service that provides storage space and resources on a server for websites to be accessible online.

#### Types of Hosting

- Shared hosting
- VPS hosting
- Dedicated hosting
- Cloud hosting.



#### How to Host a Website

- 1. Process:** Choose a hosting provider, select a hosting plan based on website needs, upload website files, and configure DNS settings to link the domain to the hosting server.
- 2. Hosting Providers:** Popular providers include Bluehost, HostGator, and SiteGround.
- 3. Considerations:** Evaluate factors like uptime reliability, customer support, scalability, and pricing.

BASIC For those 'Getting Started'	PERSONAL For Average Size Website	SILVER No.1 Business Choice	ADVANCED Heavy Traffic Website
<p><b>\$ 0.99/mo</b></p> <p>Basic blogs, personal, or business website</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> HOST A SINGLE WEBSITE</li> <li><input checked="" type="checkbox"/> <b>500MB 1GB NVME SSD</b></li> <li><input type="checkbox"/> 10 E-MAIL IDs</li> <li><input type="checkbox"/> 5GB BANDWIDTH</li> <li><input checked="" type="checkbox"/> LITESPEED TECHNOLOGY</li> </ul> <p><small>3 years at \$ 0.99/month (30% off) ▾</small></p> <p><a href="#">VIEW MORE</a></p>	<p><b>\$ 1.49/mo</b></p> <p>Professional bloggers and active average traffic websites</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> HOST A SINGLE WEBSITE</li> <li><input checked="" type="checkbox"/> <b>UNLIMITED NVME SSD</b></li> <li><input type="checkbox"/> UNLIMITED EMAIL IDs</li> <li><input checked="" type="checkbox"/> UNLIMITED BANDWIDTH</li> <li><input checked="" type="checkbox"/> LITESPEED TECHNOLOGY</li> <li><input checked="" type="checkbox"/> PREMIUM SEO TOOL</li> <li><input type="checkbox"/> GIFT 0.5 TREE TO EARTH</li> </ul> <p><small>3 years at \$ 1.49/month (25% off) ▾</small></p> <p><a href="#">VIEW MORE</a></p>	<p><b>\$ 2.89/mo</b></p> <p>For heavy traffic or multiple average traffic websites</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> HOST A THREE WEBSITE</li> <li><input checked="" type="checkbox"/> <b>UNLIMITED NVME SSD</b></li> <li><input type="checkbox"/> UNLIMITED EMAIL IDs</li> <li><input checked="" type="checkbox"/> UNLIMITED BANDWIDTH</li> <li><input checked="" type="checkbox"/> LITESPEED TECHNOLOGY</li> <li><input checked="" type="checkbox"/> PREMIUM SEO TOOL</li> <li><input type="checkbox"/> GIFT 1 TREE TO EARTH</li> </ul> <p><small>3 years at \$ 2.89/month (24% off) ▾</small></p> <p><a href="#">VIEW MORE</a></p>	<p><b>\$ 3.99/mo</b></p> <p>For business, developers and startup with scalable features</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> FREE DOMAIN .IN/.COM</li> <li><input type="checkbox"/> HOST UNLIMITED WEBSITE</li> <li><input checked="" type="checkbox"/> ALL FEATURES SILVER PACK</li> <li><input checked="" type="checkbox"/> UNLIMITED BANDWIDTH</li> <li><input checked="" type="checkbox"/> LITESPEED TECHNOLOGY</li> <li><input checked="" type="checkbox"/> PREMIUM SEO TOOL</li> <li><input type="checkbox"/> GIFT 1 TREE TO EARTH</li> </ul> <p><small>3 years at \$ 3.99/month (25% off) ▾</small></p> <p><a href="#">VIEW MORE</a></p>

## PRACTICAL EXERCISE

### Identify and Avoid Phishing Attempts:

#### •Registering a Domain Name:

- Search for available domain names.
- Complete the registration process on a domain registrar's website.

#### •Setting Up Web Hosting:

- Choose a suitable hosting plan.
- Upload website files.
- Configure DNS settings to make the website live.



Video Links: <https://www.youtube.com/watch?v=sh7fe05mUfA>



Credit: [TechTerms](#)

Video Links: <https://www.youtube.com/watch?v=BDdACdrs1vE>

Credit: Ray DelVecchio

Video Links: <https://www.youtube.com/watch?v=se3ujLcBsAE>

Credit: Codecademy

### Introduction to E-commerce

E- Commerce, short for electronic commerce, is essentially doing business on the internet. It's like having a digital version of a traditional store or service, but instead of walking into a physical building, customers visit a website or use an app.

#### Key Aspects of E-Commerce:

##### ONLINE STORES

Think of these as digital shop windows. Just like how physical stores display products on shelves, online stores show products on web pages which include product descriptions, prices, and photos to help customers make decisions. E.g. Jumia, Konga

##### PAYMENT GATEWAYS

These are like digital cash registers. When you're ready to pay, the payment gateway securely processes your credit card or other payment information. They ensure your financial details are protected from hackers. E.g. PayPal, Paystack

##### SHOPPING CARTS

This is the digital equivalent of pushing a cart around a store and putting items in it. As you browse the online store, you can add items to your cart. You can usually view your cart at any time to see what you've selected and the total cost. You can also remove items or change quantities before you finalize your purchase.

##### ORDER MANAGEMENT

Once you've made a purchase, this system takes over. It's like having a team of people tracking your order from the moment you buy until it reaches your doorstep. It helps businesses keep track of inventory, process orders, and manage shipping. It also often provides updates to customers about their order status.

---

#### Additional aspects that could be helpful to understand:

---

##### CUSTOMER ACCOUNTS

Many e-commerce sites allow you to create a personal account. This stores your information (like shipping address) for faster checkout next time. It also allows you to view your order history and track current orders.

##### PRODUCT REVIEWS

Unlike physical stores, e-commerce sites often feature customer reviews. These can help you make informed decisions based on others' experiences.

##### MOBILE COMMERCE

Many e-commerce businesses now have mobile apps or mobile-friendly websites. This allows you to shop from your smartphone or tablet, not just a computer.

##### DIGITAL PRODUCTS

E-commerce isn't just for physical goods. You can also buy and download digital products like e-books, software, or music.

E-commerce has revolutionized how we shop, making it possible to buy almost anything from anywhere at any time. It offers convenience for customers and new opportunities for businesses of all sizes to reach a global market.

### E-commerce Flow Operation

#### •User Browsing:

- **Customer** visits an e-commerce website or app to browse products.
- **Website** displays various categories and products, often with search and filter options.

#### •Product Selection:

- **Customer** selects a product to view detailed information.
- **Website** provides product details, images, reviews, and specifications.

#### •Add to Cart:

- **Customer** decides to purchase and adds the product to the shopping cart.
- **Website** updates the shopping cart and displays the selected items.

#### •Checkout Process:

- **Customer** proceeds to checkout, where they review the order.
- **Website** asks for customer details like shipping address, payment information, and contact information.

#### •Payment Authorization:

- **Customer** enters payment details and submits the order.
- **Payment Gateway** processes the payment and authorizes the transaction.

#### •Order Confirmation:

- **Website** confirms the order and sends a confirmation email to the customer.
- **Customer** receives an order confirmation with the order number and details.

#### •Order Processing:

- **E-commerce Backend** processes the order, updating inventory and preparing for shipping.
- **Warehouse** picks, packs, and labels the products for shipment.

#### •Shipping and Delivery:

- **Logistics Provider** receives the package and updates the tracking information.
- **Customer** can track the shipment via the provided tracking number.

#### •Delivery:

- **Logistics Provider** delivers the package to the customer's specified address.
- **Customer** receives the package and confirms delivery.

#### •Post-Sale Service:

- **Customer** can leave reviews, request returns or refunds, or contact customer service for assistance.
- **E-commerce Platform** handles customer service queries and manages post-sale support.

**End of Study Session**  
CLICK! TO RESTART THIS LESSON



## PRACTICAL EXERCISE

IN A COLLABORATIVE BRAINSTORMING SESSION, GUIDE LEARNERS THROUGH:

**· Create an Account on an E-commerce Platform:**

- Use platforms like Shopify or WooCommerce.

**· Set Up an Online Store:**

- Add products, descriptions, and pricing.

**· Configure Payment Gateways:**

- Use secure options like Paystack or Flutterwave.

**· Manage Inventory:**

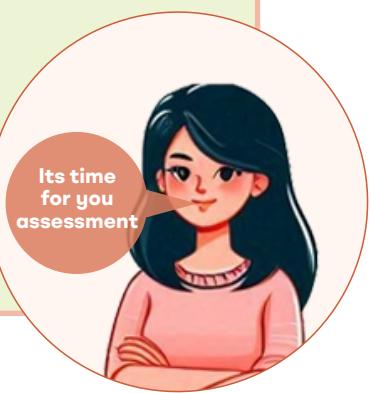
- Track and update stock levels.

**· Process Orders:**

- Handle order fulfillment and customer communication.

**· Implement Security Measures:**

- Use SSL certificates to protect customer data.



### Additional Resources:



Video Links: <https://www.youtube.com/watch?v=72MliXwjggY>

Credit: [ProfileTree](#)

Video Links: <https://www.youtube.com/watch?v=Bvg1MaxXV0E>

Credit: [ProfileTree](#)



*Now, can you confidently say repeat these statements to yourself:*

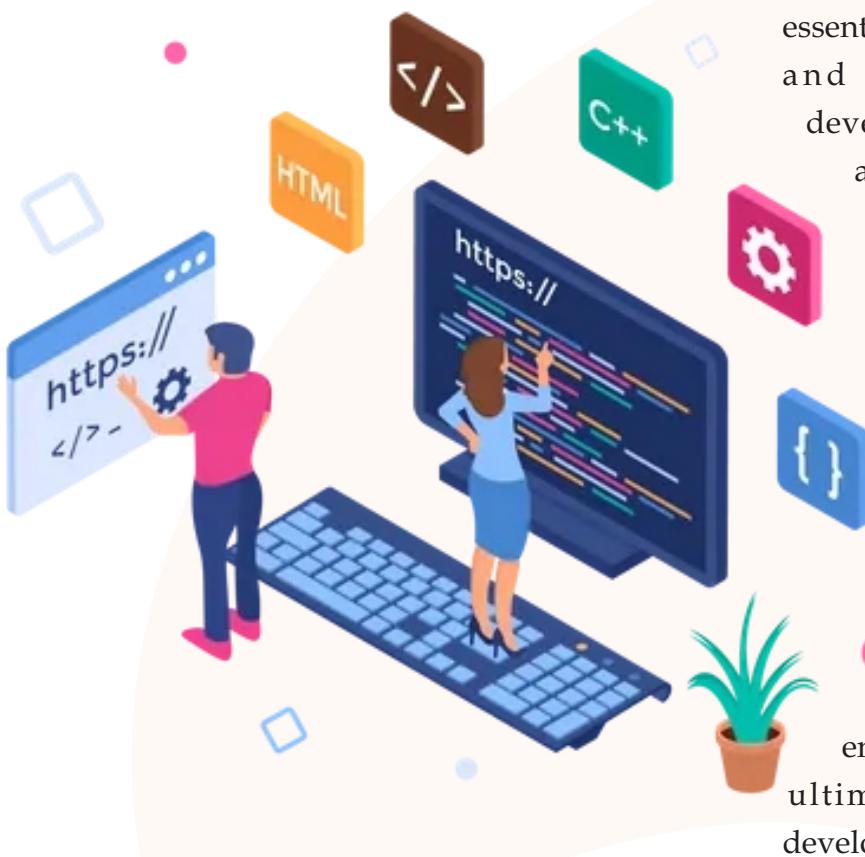
**I can:**

- Utilize the Internet efficiently for web development tasks.
- Understand and apply knowledge of essential web development software.
- Recognize the necessary hardware components for web development.
- Explain the functioning of computer networks in the context of web development.
- Identify and describe various types of computer networks and their specific functions in web development.

**CLICK! FOR YOUR NEXT MODULE**

# ONLINE & OFFLINE RESOURCES FOR WEB DEVELOPMENT

This unit aims to equip learners with essential skills to leverage both online and offline resources for web development. Learners will define and identify various online resources, understand their importance in learning, and register with programming communities and e-learning platforms. They will develop the ability to structure effective search queries, navigate e-learning platforms, set up personalized profiles, and use online tutorials and videos to enhance their learning experience, ultimately optimizing their web development skills.



## At the end of this Module, learners should be able to:

- The learners will demonstrate knowledge of online resources available to web developers.
- The learners will use online resources for web development.
- The learners will combine offline and online resources in learning web development.



### Eras and Trends of the Web

#### Online Resources

Online resources are tools, websites, and platforms available on the internet that provide information, tutorials, courses, and communities for learning web development. These resources are accessible from anywhere with an internet connection, allowing you to learn at your own pace and convenience. They include a wide range of materials such as articles, videos, interactive tutorials, forums, and more.



#### Types of Online Resources Available to Web Developers

- **Tutorial Websites:** Sites like W3Schools and MDN Web Docs offer step-by-step tutorials on HTML, CSS, JavaScript, and more. These sites provide foundational knowledge and hands-on coding practice.
- **Video Platforms:** YouTube channels like Traversy Media and Academind provide visual learning through video tutorials, which can be more engaging and easier to follow for visual learners.
- **E-Learning Platforms:** Platforms such as Udacity, Coursera, and Codecademy offer structured courses with quizzes and projects, allowing you to gain in-depth knowledge and practical experience.
- **Programming Communities:** Forums like StackOverflow and Reddit where you can ask questions and share knowledge with other developers. These communities are invaluable for troubleshooting and getting advice from experienced developers.
- **Documentation and Reference Sites:** Official documentation sites like Mozilla Developer Network (MDN) and W3C provide detailed technical information, which is essential for understanding how different web technologies work.



### Importance of Online Resources in Learning

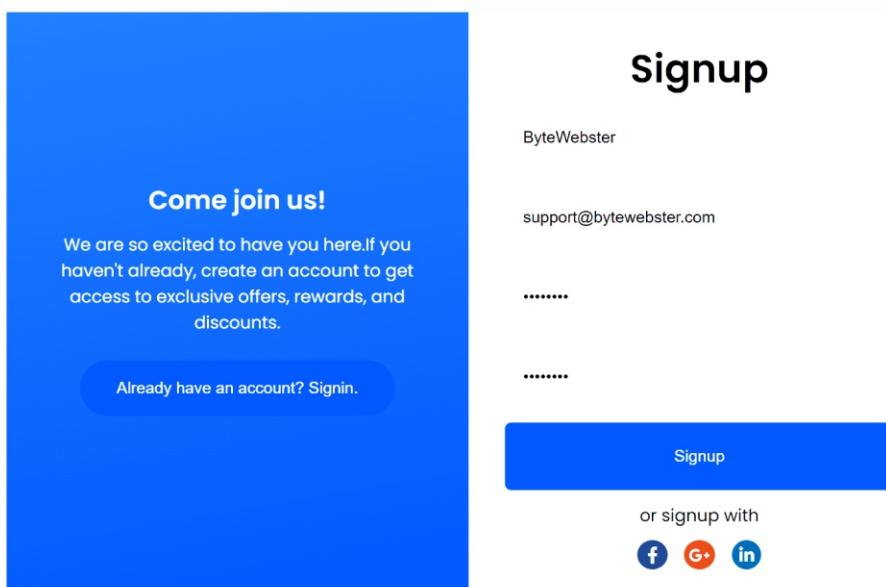
Online resources are crucial for continuous learning, staying updated with the latest trends, and finding solutions to specific problems. They offer flexibility, allowing you to learn at your own pace and according to your schedule. Additionally, online resources often provide the latest information and updates in the rapidly evolving field of web development. They also offer a diverse range of perspectives and teaching styles, catering to different learning preferences.

### Registering with Online Programming Communities

Joining communities like StackOverflow helps you connect with other developers, ask questions, and share your knowledge. These communities are supportive and can provide quick answers to your coding problems. Registering is simple: visit the site, click on "Sign Up," and follow the instructions to create an account. Being active in these communities also helps you stay engaged and motivated.

### Registering with E-Learning Platforms

Platforms like Udacity and Coursera offer comprehensive courses that include video lectures, quizzes, and hands-on projects. To register, visit the platform's website, click on "Sign Up," choose your preferred method (email, Google, etc.), and complete the registration process. These platforms often offer free courses as well as paid ones, with certifications that can enhance your resume.



*Sample of a signup page!!*

## PRACTICAL EXERCISE

*In a practical session guided by the instructor, learners will:*

1. Identify and Discuss Different Types of Online Resources Available to Web Developers.
2. Research and Present using the group formed in Module 3:
  - Each group member will research and present on one type of resource.
  - Highlight key examples and how they can be utilized in learning web development.
3. Receive Feedback:
  - Get feedback from the instructor on the relevance and applicability of each resource type in real-world scenarios.



### Additional Resources:



Video Links: <https://www.youtube.com/watch?v=rjL2zhtn4qM>

Credit: GreatStack

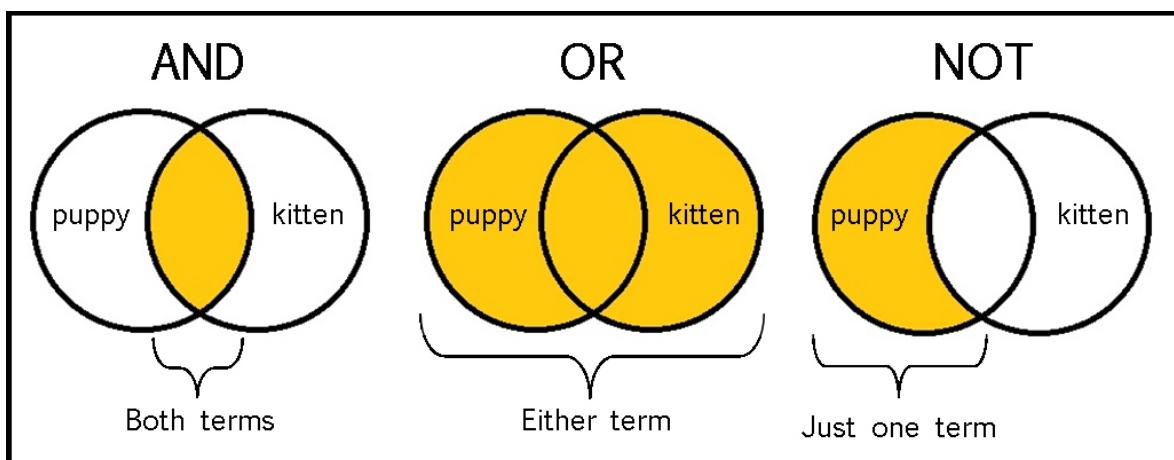
Video Links: <https://www.youtube.com/watch?v=JKHGs31MGKE>

Credit: Huy NG

## Effective use of Online Resources for Web Development

### Structuring Proper Search Queries

Effective search queries help you find relevant information quickly. Use specific keywords, phrases, and Boolean operators (AND, OR, NOT) to refine your searches. For example, "how to center a div in CSS" or "best JavaScript frameworks 2024." The more precise your query, the better the results you'll get. Practice refining your searches to become more efficient.



### Navigating E-Learning Platforms

Each platform has its own interface, but most follow a similar structure. After signing in, explore the course catalog, enroll in courses, and follow the syllabus. Platforms like Udacity and Coursera provide progress tracking, quizzes, and community forums. Take advantage of these features to stay organized and monitor your progress.

The screenshot shows the Coursera homepage with the following elements:

- Logo:** Coursera logo with a blue square icon.
- Search Bar:** "Explore" dropdown, "What do you want" input field, and a magnifying glass icon.
- Navigation Links:** "Online Degrees", "Find your New Career", "For Enterprise", "For Universities", "Log In", and a "Join for Free" button.
- Advertisement:** A banner for Google's Get job-ready skills for digital marketing and e-commerce careers in 6 months.

## Learn without limits

Get that "ready for anything" feeling with more than 5,000 courses, Professional Certificates, and degrees from world-class universities and companies.

[Join for Free](#)

[Try Coursera for Business](#)



## Setting Up a Profile Aligned with Your Learning Path

Creating a profile that reflects your learning goals helps you track progress and connect with like-minded learners. Include information about your interests, completed courses, and current learning objectives. A well-structured profile can also help you receive personalized recommendations and feedback from instructors and peers.



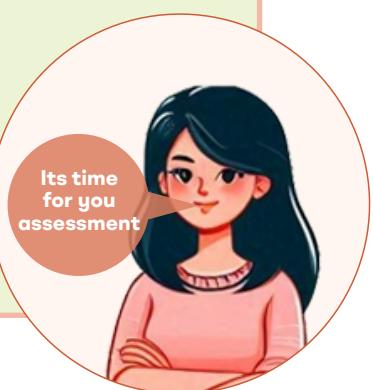
## Using Online Tutorials and Videos

Videos and tutorials offer visual and practical learning experiences. Follow along with coding examples, pause and rewind as needed, and practice by replicating the examples on your own. Interactive tutorials often provide hands-on coding environments where you can immediately apply what you've learned.

## PRACTICAL EXERCISE

*In a practical session guided by the instructor, learners will:*

- 1. Practice Structuring Proper Search Queries:**
  - Use search engines and e-learning platforms.
- 2. Choose a Specific Topic in Web Development:**
  - Examples: responsive design techniques or JavaScript frameworks.
- 3. Brainstorm a List of Search Queries:**
  - Develop queries that would yield relevant results.
- 4. Share and Discuss:**
  - Share your queries with peers and the industry mentor.
  - Discuss the effectiveness of each approach in finding accurate and useful information.



### Additional Resources:

Video Links: <https://www.youtube.com/watch?v=zK-LBukNzAs>

Credit: Traversy Media

## Online vs Offline Resources for Web Development

Combining offline and online resources means using both physical materials and internet-based tools to learn web development. Offline resources include textbooks, printed guides, coding bootcamps, and local study groups, while online resources include websites, video tutorials, e-learning platforms, and online communities. Using both types of resources together can greatly enhance your learning experience.

### Difference Between Online and Offline Resources

Aspect	Offline Resources	Online Resources
<b>Access Method</b>	Physical access, in-person attendance	Internet access, virtual attendance
<b>Examples</b>	<ul style="list-style-type: none"> <li>- Textbooks and Printed Guides: Physical books with in-depth explanations, examples, and exercises.</li> <li>- Coding Bootcamps: Intensive, in-person training programs with hands-on coding experience and mentorship.</li> <li>- Local Study Groups/Meetups: In-person gatherings for discussion, knowledge sharing, and collaboration on projects.</li> </ul>	<ul style="list-style-type: none"> <li>- Tutorial Websites: Step-by-step guides and coding examples (e.g., W3Schools, MDN Web Docs).</li> <li>- Video Tutorials: Platforms like YouTube providing video-based lessons and walkthroughs.</li> <li>- E-Learning Platforms: Websites like Udacity, Coursera, and Codecademy offering structured courses, quizzes, and certifications.</li> <li>- Online Communities: Forums and discussion boards like StackOverflow and Reddit for Q&amp;A, knowledge sharing, and problem-solving.</li> </ul>
<b>Interactivity</b>	High in-person interactivity	Varies; can be interactive through forums, live sessions, or self-paced with limited direct interaction
<b>Learning Environment</b>	Physical spaces such as classrooms, libraries, or meeting venues	Virtual spaces accessible from anywhere with an internet connection

Aspect	Offline Resources	Online Resources
<b>Mentorship/Guidance</b>	Direct mentorship and guidance from instructors and peers	Virtual mentorship through forums, online instructors, or peer interactions
<b>Cost</b>	May include costs for materials, travel, or attendance fees	Can range from free to paid subscriptions or course fees
<b>Flexibility</b>	Fixed schedules and locations	Flexible schedules and locations, allowing for self-paced learning

### Advantages of Using Both Online and Offline Resources

Aspect	Advantages of Online Resources	Advantages of Offline Resources
<b>Accessibility</b>	Available anytime and anywhere with an internet connection, allowing for flexible learning schedules.	Physical books and in-person training provide a distraction-free environment, helping you concentrate better.
<b>Up-to-Date Information</b>	Frequently updated with the latest trends, technologies, and best practices.	Textbooks and bootcamps often follow a well-organized curriculum, providing a clear path from beginner to advanced topics.
<b>Interactive Learning</b>	Many online platforms offer interactive coding environments and instant feedback.	Coding bootcamps and workshops offer practical, hands-on experience under the guidance of instructors.
<b>Diverse Content</b>	A wide range of formats including articles, videos, and interactive tutorials cater to different learning styles.	Face-to-face interactions in study groups and meetups allow for real-time discussions, networking, and collaboration.
<b>Structured Learning</b>	Online platforms can provide structured courses with a mix of interactive content, quizzes, and projects.	Textbooks and bootcamps often have a structured curriculum, offering a systematic approach to learning complex topics.
<b>Personal Interaction</b>	Online communities and forums allow for interaction and collaboration with a global audience.	Personal interaction with instructors and peers in a physical setting enhances learning through immediate feedback and support.

### Advantages of Combining Both:

- **Comprehensive Learning:** Blending the depth and structure of offline resources with the flexibility and interactivity of online resources creates a well-rounded learning experience.
- **Enhanced Understanding:** Using multiple resources to study the same topic can reinforce concepts and provide different perspectives, leading to a deeper understanding.
- **Flexibility and Support:** Online resources offer flexibility in learning schedules, while offline resources provide personal support and mentorship.
- **Broader Skill Set:** Access to a wide range of learning materials helps develop a more extensive and diverse skill set, making you a well-equipped web developer.



By combining online and offline resources, you can take advantage of the strengths of both to maximize your learning potential. For example, you might start with an online tutorial to get a basic understanding of a concept, then use a textbook to dive deeper into the theory, and finally attend a coding bootcamp or study group to apply what you've learned in a practical, hands-on environment. This integrated approach ensures a thorough and effective learning

## PRACTICAL EXCERCISE

### 1. Develop a Comprehensive Learning Plan for HTML Basics:

- Identify key online tutorials, videos, and offline resources (like textbooks) for learning HTML.
- Include practical exercises to create a "Hello World" webpage and insert an image using HTML.

### 2. Create a "Hello World" Webpage:

- Write HTML code to display a basic webpage with the text "Hello World".
- Incorporate CSS for basic styling if applicable.

### 3. Insert an Image:

- Use HTML to add an image to your webpage.
- Ensure the image is sourced correctly and displayed appropriately.

### 4. Present Your Learning Plan and Webpage to the Industry Professional:

- Explain how you utilized different resources to learn HTML.
- Demonstrate the "Hello World" webpage and image integration.

### Receive Feedback:

- Receive feedback on your learning plan, webpage design, and implementation of HTML basics.



### Additional Resources:

Video Links: <https://www.youtube.com/watch?v=9Kws0lFHJII>

Credit: Hitesh Choudhary

Video Links: <https://www.youtube.com/watch?v=PlxWf493en4>



*Now, can you confidently say repeat these statements to yourself:*

**I can:**

1. Demonstrate knowledge of various online resources available to web developers and understand their features and benefits.
2. Effectively use online resources such as search queries, e-learning platforms, tutorials, and videos for web development learning.
3. Combine online and offline resources to create comprehensive learning plans, reinforcing concepts and gaining diverse perspectives.

**CLICK! FOR YOUR NEXT MODULE**

# THE WEB DEVELOPMENT ENVIRONMENT



The Web Development Environment unit equips learners with comprehensive knowledge

and practical skills for web development, including custom frameworks and Command Line

Interfaces (CLI) for managing development environments using tools like npm and pip.

Learners will explore web application development tools, gaining proficiency with text editors and Integrated Development Environments (IDEs) such as Notepad++, Sublime Text,

NetBeans, and Microsoft Visual Studio. They will learn to differentiate between text editors

and IDEs, understand the importance of IDEs, and become adept at installing and configuring these tools.

## At the end of this Module, learners should be able to:

Understand the difference between text editors and IDEs.

Use various text editors and IDEs for web development.

Recognize the importance and features of IDEs.

Install and configure text editors and IDEs.

List and understand different types of CMS.

Understand what a web development framework is.

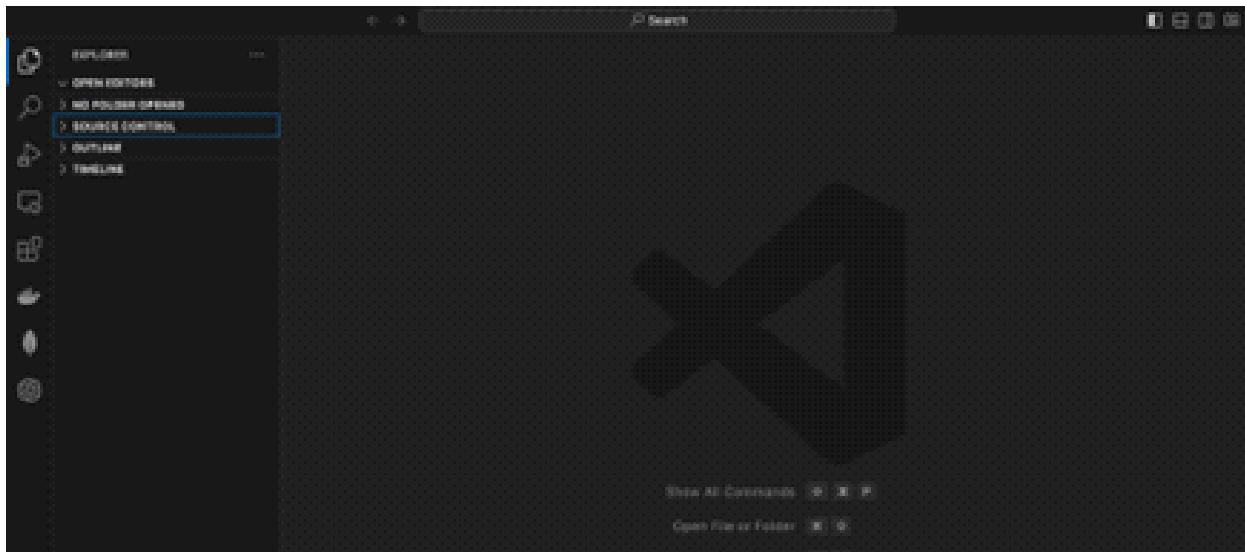


## How to use various web application development tools.

### Text Editors and Integrated Development Environments (IDE)

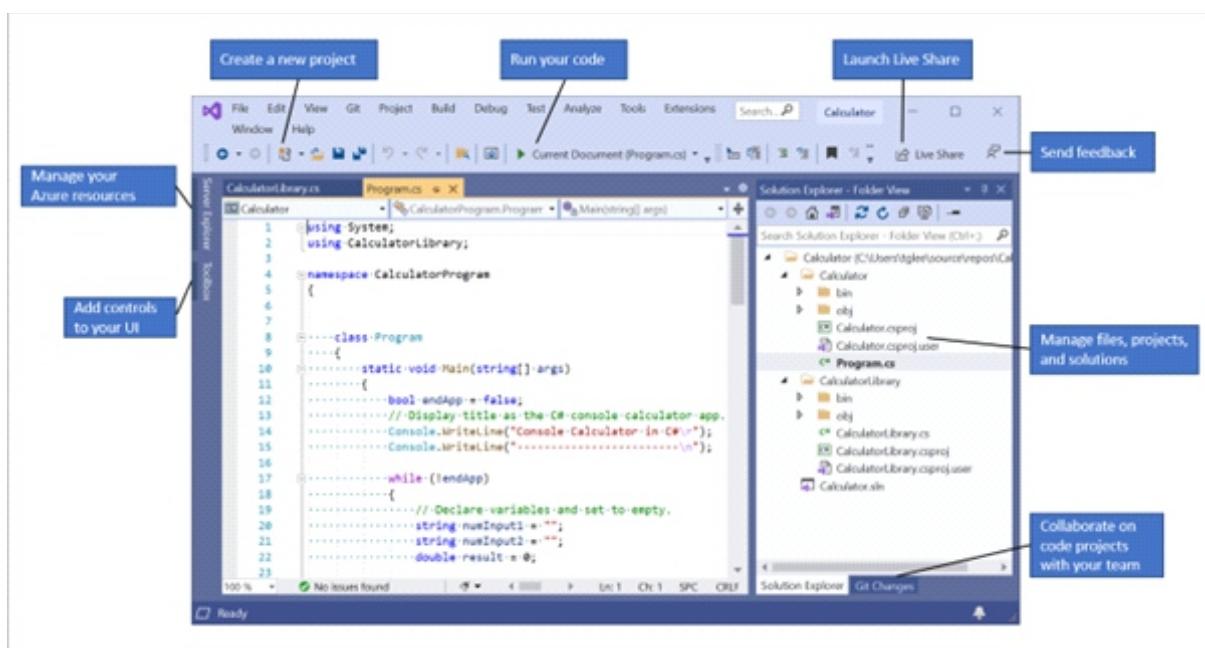
#### Text Editors

Text editors are simple programs designed to write and edit plain text. These tools are fundamental for coding because they allow developers to write, edit, and manipulate source code. Examples include Notepad++, Sublime Text, Visual Studio Code, and Atom.



#### Integrated Development Environments (IDE)

IDEs are comprehensive software applications that provide developers with tools for software development. An IDE typically includes a source code editor, build automation tools, and a debugger. Examples include Visual Studio, NetBeans, and IntelliJ IDEA.

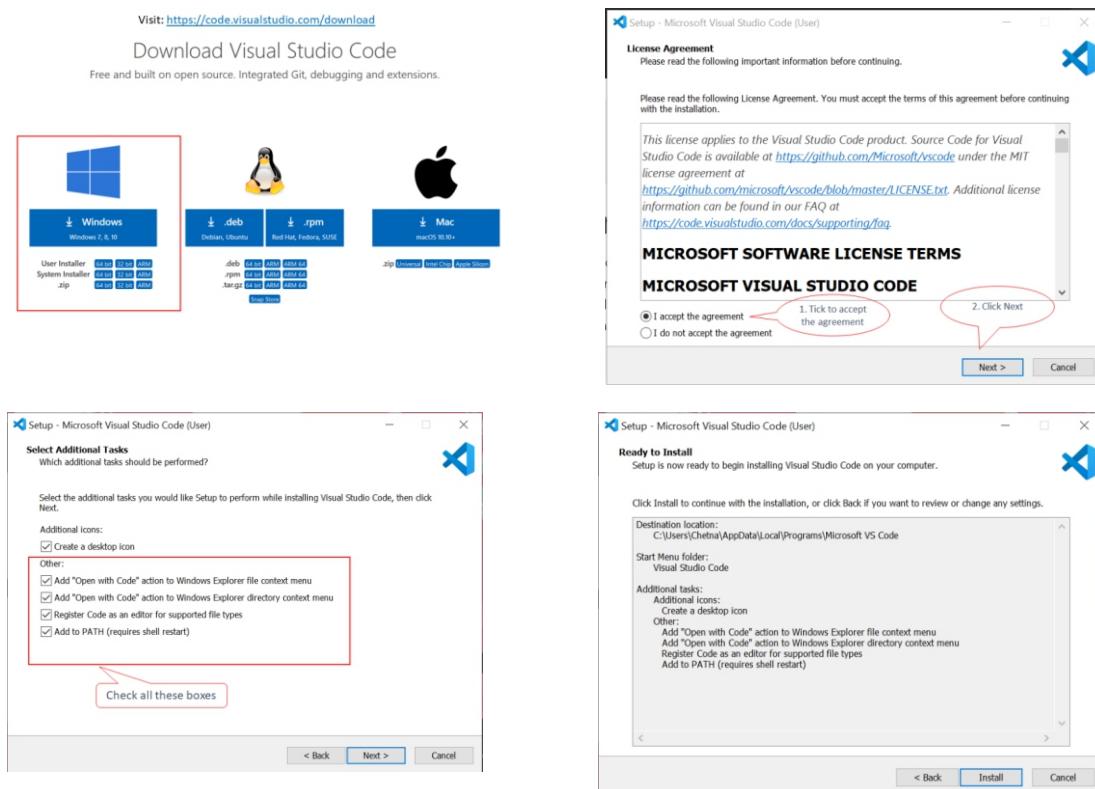


## Code Editor Installation

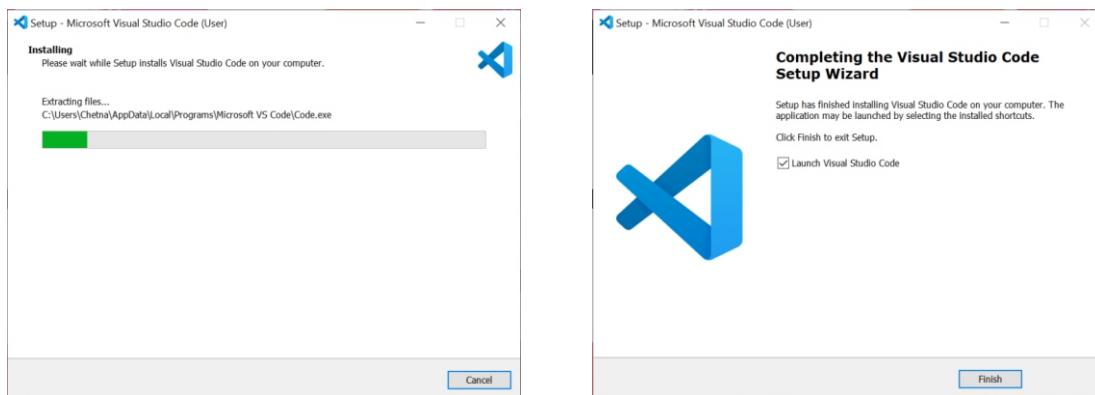
### Visual Studio Code

#### Installing Visual Studio Code on Windows

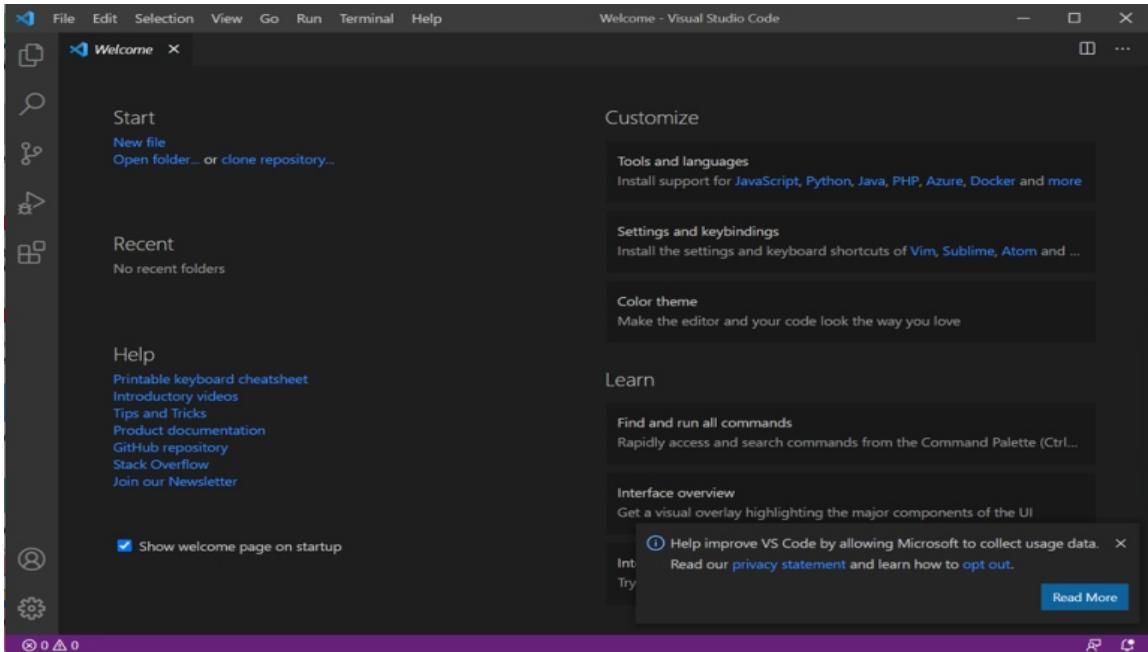
1. Download the VS Code file from the [Official Website](https://code.visualstudio.com/download)
2. Press the Windows button to download for Windows.
3. When the download finishes, click on the installer to begin the installation
4. Accept the agreement and press next to continue



5. Ensure the box for “Add to path” is checked and click Next to continue
6. Begin the installation by clicking on Install
7. Once it begins to install, wait for some minutes to complete the installation
8. Once the installation is complete, click on the Launch button to start it.



1. The environment will look like the image below



## The importance of using an Integrated Development Environment (IDE)

### Importance of IDEs

- Efficiency: IDEs streamline the development process by combining multiple tools into one application. This integration improves efficiency by providing everything a developer needs in one place.
- Debugging: IDEs come with powerful debugging tools that help developers quickly find and fix errors in their code.
- Productivity: Features like code completion, syntax highlighting, and project templates significantly enhance productivity.
- Collaboration: Many IDEs support version control systems (e.g., Git), making it easier for teams to collaborate on projects.

## Difference between text editors and IDEs

### Text Editors vs. IDEs

- Functionality: Text editors provide basic text editing capabilities with some added features like syntax highlighting, while IDEs offer a full suite of development tools.
- Usage Context: Text editors are ideal for lightweight editing and quick changes, whereas IDEs are better suited for larger, more complex projects that require comprehensive development and debugging tools.
- Features: IDEs typically include features such as a debugger, code completion, and project management tools that text editors do not have.

### Identification of the IDE features

#### Common Features of an IDE

- Code Editor: An advanced text editor with features like syntax highlighting, code folding, and code completion.
- Compiler/Interpreter: Converts written code into executable programs.
- Debugger: Helps in testing and debugging code by allowing the developer to run the code step-by-step and inspect variables.
- Project Management: Manages project files and resources, making it easier to navigate and organize the codebase.
- Version Control Integration: Supports version control systems like Git, facilitating collaboration and code versioning.

### The limitations of common IDEs

#### Limitations of IDEs

- Resource Intensive: IDEs require more system resources (CPU, memory) compared to text editors, which can slow down older computers.
- Complexity: The extensive features of IDEs can be overwhelming for beginners, leading to a steeper learning curve.
- Cost: Some IDEs are not free and require a paid license, which can be a barrier for some users.

### Configuration of the text editors

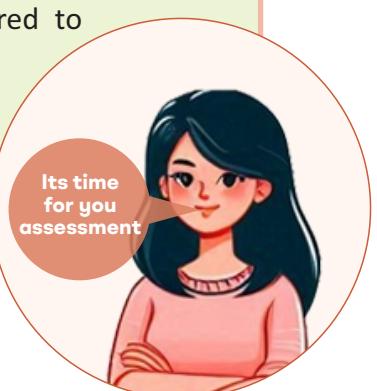
#### Configuring Text Editors

- Customization: Adjust settings such as themes, key bindings, and preferences.
- Plugins: Install and manage plugins to enhance functionality.

## PRACTICAL EXCERCISE

With the guardians of the instructor install and configure Software, code creation, and feature extraction:

- a. Install Notepad++ and Visual Studio on your computer.
- b. Configure Notepad++ by installing a theme and a code formatter plugin.
- c. Configure Visual Studio by adjusting the theme and installing an extension for web development.
- d. Create a simple HTML file using Notepad++ with a basic webpage structure.
- e. Open the same HTML file in Visual Studio, edit, and use the live server feature to preview the webpage.
- f. List the additional features available in Visual Studio compared to Notepad++.
- g. Identify how these features can enhance your development workflow.



### Additional Resources:

<https://notepad-plus-plus.org/downloads/>

<https://code.visualstudio.com/download>

<https://www.sublimetext.com/download>

### - Video Links:

<https://www.youtube.com/watch?v=Ozr46bBkXl8>

Credit: freeCodeCamp.org

### Use of command line interfaces (CLI) in setting web development environment

List common Command Line Interface (CLI) commands and their uses

#### Common CLI Commands

##### 1. Navigating the File System:

`cd [directory]`: Change the current directory.

Example: `cd Documents` navigates to the Documents directory.

```
Last login: Sat Jul 13 13:43:58 on console
(base) mac@MACs-MBP-4 ~ % cd ..
(base) mac@MACs-MBP-4 /Users %
```

`pwd`: Print the current working directory.

Example: `pwd` outputs `/home/user/Documents`.

```
(base) mac@MACs-MBP-4 /Users % pwd
/Users
(base) mac@MACs-MBP-4 /Users %
```

##### 2. Managing Directories:

`mkdir [directory]`: Create a new directory.

Example: `mkdir new\_project` creates a directory named new\_project.

```
(base) mac@MACs-MBP-4 /Users % mkdir new_project
```

`rmdir [directory]`: Remove an empty directory.

Example: `rmdir old\_project` removes the old\_project directory if it's empty.

```
(base) mac@MACs-MBP-4 /Users % rmdir new_project
```

##### 3. Managing Files:

`ls [options]`: List the contents of a directory.

Example: `ls -l` lists the contents in long format

```
(base) mac@MACs-MBP-4 /Users % ls
Shared  mac      new_app shuaib
(base) mac@MACs-MBP-4 /Users %
```

### Use of package manager to install frameworks and dependencies

#### Using Package Managers to Install Frameworks and Dependencies

##### 1. Node Package Manager (npm):

Command: `npm install [package\_name]`

Example: `npm install lodash` installs the lodash framework for Javascript.

```
(base) mac@MACs-MBP-4 /Users % npm install lodash
```

##### 2. Python Package Index (pip):

Command: `pip install [package\_name]`

Example: `pip install requests` installs the Requests library for Python

```
(base) mac@MACs-MBP-4 /Users % pip install requests
Requirement already satisfied: requests in /opt/anaconda3/lib/python3.11/site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2024.2.2)
```

### Update and upgrade installed frameworks

#### Updating and Upgrading Frameworks

pip:

Update a Package: `pip install --upgrade [package\_name]`

Example: `pip install --upgrade requests` upgrades the Requests library.

```
(base) mac@MACs-MBP-4 /Users % pip install --upgrade requests
Requirement already satisfied: requests in /opt/anaconda3/lib/python3.11/site-packages (2.31.0)
Collecting requests
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2024.2.2)
  Downloading requests-2.32.3-py3-none-any.whl (64 kB)
                                             64.9/64.9 kB 340.6 kB/s eta 0:00:00
Installing collected packages: requests
  Attempting uninstall: requests
    Found existing installation: requests 2.31.0
    Uninstalling requests-2.31.0:
      Successfully uninstalled requests-2.31.0
```

### Use of package manager to uninstall frameworks and dependencies

#### Uninstalling Frameworks and Dependencies

1. npm:

Command: `npm uninstall [package\_name]`

Example: `npm uninstall express` uninstalls the Express framework.

```
(base) mac@MACs-MBP-4 /Users % npm uninstall express
```

2. pip:

Command: `pip uninstall [package\_name]`

**Example: `pip uninstall requests` uninstalls the Requests library.**

```
(base) mac@MACs-MBP-4 /Users % pip uninstall requests
Found existing installation: requests 2.32.3
Uninstalling requests-2.32.3:
  Would remove:
    /opt/anaconda3/lib/python3.11/site-packages/requests-2.32.3.dist-info/*
    /opt/anaconda3/lib/python3.11/site-packages/requests/*
Proceed (Y/n)? █
```

### How to detect installed frameworks and dependencies and their versions using Command Line Interface (CLI)

Detecting Installed Frameworks and Dependencies

1. npm:

Command: `npm list --depth=0`

**Example: `npm list --depth=0` lists all installed packages and their versions at the root level.**

```
(base) mac@MACs-MBP-4 /Users % npm list --depth=0
Users@ /Users
└─ lodash@4.17.21

(base) mac@MACs-MBP-4 /Users % █
```

2. pip:

Command: `pip list`

**Example: `pip list` lists all installed packages and their versions.**

```
(base) mac@MACs-MBP-4 /Users % pip list
Package           Version
-----
aiobotocore       2.7.0
aiohttp           3.9.3
aioiterools       0.7.1
aiosignal         1.2.0
alabaster         0.7.12
altair             5.0.1
anaconda-anon-usage 0.4.3
anaconda-catalogs 0.2.0
anaconda-client   1.12.3
anaconda-cloud-auth 0.1.4
anaconda-navigator 2.5.2
anaconda-project  0.11.1
anyio              4.2.0
appdirs            1.4.4
applaunchservices 0.3.0
appnope            0.1.2
appscript          1.1.2
archspec           0.2.1
argon2-cffi       21.3.0
argon2-cffi-bindings 21.2.0
arrow               1.2.3
astroid             2.14.2
astropy             5.3.4
asttokens          2.0.5
```

## Popular package managers

### 1. npm (Node Package Manager):

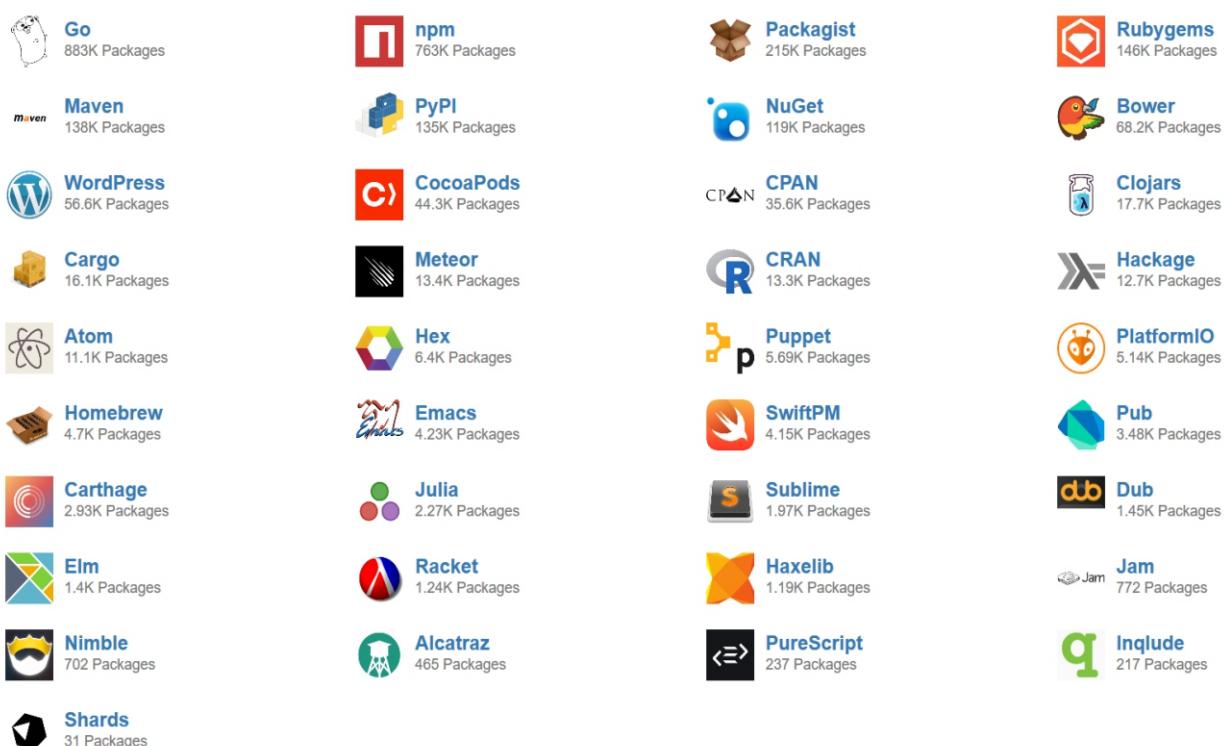
- Used for managing JavaScript packages.
- Common Use: Installing and managing packages for Node.js applications.

### 2. pip (Python Package Index):

- Used for managing Python packages.
- Common Use: Installing and managing libraries for Python applications.

### 3. Composer:

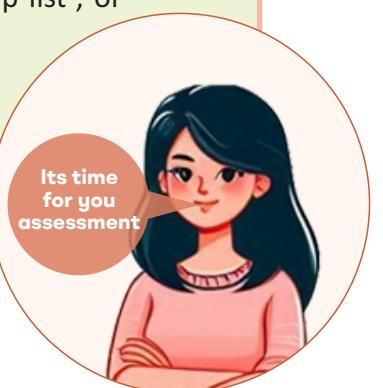
- Used for managing PHP packages.
- Common Use: Installing and managing dependencies for PHP projects.



## PRACTICAL EXERCISE

With the guide of your instructor, do the following exercise:

- Use `cd`, `mkdir`, `ls`, `cp`, `mv`, and `rm` commands to create and manage directories and files.
- Navigate the file system and explore different directories.
- Install a new framework using `npm`, `pip`, or `Composer`.
- Update an existing package to the latest version.
- Uninstall a package and verify its removal.
- List all installed packages and their versions using `npm list`, `pip list`, or `composer show` .
- Document the versions of the installed packages.



### Additional Resources:

- Video Links:

[https://www.youtube.com/watch?v=4aEETgIA\\_Wo](https://www.youtube.com/watch?v=4aEETgIA_Wo)  
<https://www.youtube.com/watch?v=P3aKRdUyrOs>



*Now, can you confidently say repeat these statements to yourself:*

**I can:**

1. Demonstrate knowledge of various online resources available to web developers and understand their features and benefits.
2. Effectively use online resources such as search queries, e-learning platforms, tutorials, and videos for web development learning.
3. Combine online and offline resources to create comprehensive learning plans, reinforcing concepts and gaining diverse perspectives.

**CLICK! FOR YOUR NEXT MODULE**

# INTRODUCTION TO CLIENT-SIDE DEVELOPMENT

This unit introduces learners to HTML, the fundamental language for creating web pages, where they will learn to identify and use HTML tags, write HTML code, understand various HTML concepts, use different text editors/IDEs, present code clearly, and create a basic web page. It also covers JavaScript, a powerful scripting language used to make websites dynamic and interactive. Learners will learn how JavaScript works, its importance, its syntax, and how to link JavaScript files to HTML pages, enabling them to enhance web pages with dynamic content. Additionally, this unit introduces learners to Content Delivery Networks (CDNs), focusing on their role in efficient web content delivery, including their definition, usage, importance, and advantages and disadvantages compared to local installations.

## At the end of this Module, learners should be able to:

- Write HTML for a website
- Use JavaScript to make website dynamic
- Implement Cascading Style Sheets (CSS) in a website
- Use Content Delivery Network (CDN).



### HTML for the Website

An HTML tag is a piece of code used to define the structure and content of a web page. HTML tags are the building blocks of HTML, which stands for HyperText Markup Language. Tags are usually written inside angle brackets (<>) and typically come in pairs:

- An opening tag and
- A closing tag.

The content in between these tags is affected by the tag.

Example of an HTML Tag:

```
<tagname>Content goes here</tagname>
```



- **<tagname>**: The opening tag, which starts the element.
- **Content goes here**: The content or data enclosed within the tag.
- **</tagname>**: The closing tag, which ends the element. The closing tag is the same as the opening tag but with a forward slash / before the tag name.

### Self-Closing Tags:

Some HTML tags do not require a closing tag and are self-closing. These tags often represent elements that do not have content, such as line breaks or images.

**Example:** <br> Tag

The <br> tag is used to insert a line break.

```
This is the first line.<br>This is the second line.
```

**OUTPUT:**

This is the first line.

This is the second line.



### Attributes

Attributes provide additional information about an element. They are always included in the start tag and usually consist of a name-value pair.

### Syntax

```
<element attribute="value">Content</element>
```



### Example

```
<a href="https://www.example.com" target="_blank">Visit Example</a>
    • href="https://www.example.com": Specifies the URL of the link.
    • target="_blank": Opens the link in a new tab or window.
```



### Common Attributes:

- **id:** Assigns a unique identifier to an element.
- **class:** Assigns one or more class names to an element, used for styling with CSS.
- **style:** Applies inline CSS styles to an element.
- **title:** Provides additional information about an element, typically displayed as a tooltip.

## Hypertext Mark-up Language (HTML) Syntax

HTML syntax refers to the rules and conventions for writing HTML documents. Here's a detailed description of HTML syntax:

### HTML Document Structure

An HTML document is structured in a hierarchical way, typically starting with a **<!DOCTYPE>** declaration, followed by an **<html>** element that contains two main sections: **<head>** and **<body>**.

### Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First Web Page</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
    <p>This is a paragraph of text.</p>
</body>
</html>
```



- **<!DOCTYPE html>**: Declares the document type and version of HTML. This is essential for ensuring that the document is parsed correctly by browsers.
- **<html lang="en">**: The **<html>** element is the root of the document. The **lang** attribute specifies the language of the document.
- **<head>**: Contains meta-information about the document, such as the title, character set, and links to external resources like CSS stylesheets and scripts.
- **<body>**: Contains the content of the document, such as text, images, links, tables, and other elements.

### Semantic HTML

Semantic HTML uses elements that clearly describe their meaning in a human- and machine-readable way. This improves accessibility, SEO, and code clarity.

#### Example:

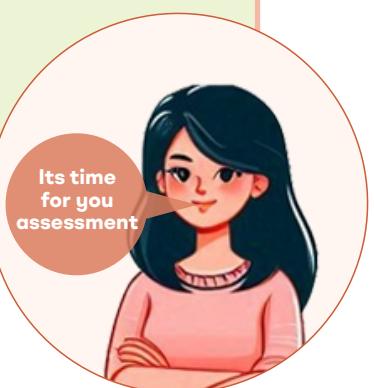
```
<header>
  <h1>Welcome to My Website</h1>
</header>
<main>
  <article>
    <h2>Article Title</h2>
    <p>This is the content of the article.</p>
  </article>
</main>
<footer>
  <p>© 2024 My Website</p>
</footer>
```



## PRACTICAL EXERCISE

With the guide of your instructor, do the following exercise:

1. Open VS Code and copy the HTML script above, save and name the file as “**html-structure.html**” and run on the browser.
2. Keep the code for further extension through subsequent exercise.
3. Copy and paste this URL on the browser, <https://www.lipsum.com/>
4. Discuss it uses with your instructor.



### Additional Resources:

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

- Video Links:

<https://www.youtube.com/watch?v=it1rTvBcfRg>

[https://www.youtube.com/watch?v=bBP0ckEln4Y&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=2](https://www.youtube.com/watch?v=bBP0ckEln4Y&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=2)

<https://youtu.be/bWPMSSsVdPk>

## Basic Hypertext Mark-up Language (HTML) Tags and their Uses

Below is a more comprehensive list of basic HTML tags, categorized by their use, along with examples:

### 1. Document Structure Tags

Tag	Description	Example
<code>&lt;html&gt;</code>	Root of the HTML document.	<code>&lt;html&gt;&lt;/html&gt;</code>
<code>&lt;head&gt;</code>	Contains meta-information and links to resources.	<code>&lt;head&gt;&lt;/head&gt;</code>
<code>&lt;title&gt;</code>	Sets the title of the document, shown in the browser tab.	<code>&lt;title&gt;My Website&lt;/title&gt;</code>
<code>&lt;body&gt;</code>	Contains the content of the HTML document.	<code>&lt;body&gt;&lt;/body&gt;</code>

### 2. Text Formatting Tags

Tag	Description	Example
<code>&lt;h1&gt; to &lt;h6&gt;</code>	Define headings, with <code>&lt;h1&gt;</code> as the highest level.	<code>&lt;h1&gt;Main Title&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	Defines a paragraph.	<code>&lt;p&gt;This is a paragraph.&lt;/p&gt;</code>
<code>&lt;br&gt;</code>	Inserts a line break.	This is a line <code>&lt;br&gt;</code> break.

<b>&lt;hr&gt;</b>	Inserts a horizontal rule.	<b>&lt;hr&gt;</b>
<b>&lt;strong&gt;</b>	Defines strong text, typically displayed as bold.	<b>&lt;strong&gt;Important&lt;/strong&gt;</b>
<b>&lt;em&gt;</b>	Defines emphasized text, typically displayed as italic.	<b>&lt;em&gt;Emphasized&lt;/em&gt;</b>
<b>&lt;b&gt;</b>	Defines bold text (without semantic emphasis).	<b>&lt;b&gt;Bold Text&lt;/b&gt;</b>
<b>&lt;i&gt;</b>	Defines italic text (without semantic emphasis).	<b>&lt;i&gt;Italic Text&lt;/i&gt;</b>
<b>&lt;u&gt;</b>	Defines underlined text.	<b>&lt;u&gt;Underlined Text&lt;/u&gt;</b>
<b>&lt;mark&gt;</b>	Highlights text.	<b>&lt;mark&gt;Highlighted Text&lt;/mark&gt;</b>
<b>&lt;small&gt;</b>	Defines smaller text.	<b>&lt;small&gt;Small Text&lt;/small&gt;</b>
<b>&lt;del&gt;</b>	Defines text that has been deleted.	<b>&lt;del&gt;Deleted Text&lt;/del&gt;</b>
<b>&lt;ins&gt;</b>	Defines text that has been inserted.	<b>&lt;ins&gt;Inserted Text&lt;/ins&gt;</b>
<b>&lt;sub&gt;</b>	Defines subscript text.	H <sub>2</sub> O

### 3. Links and Media Tags

Tag	Description	Example
<a>	Defines a hyperlink.	<a href="https://www.example.com">Link</a>
<img>	Embeds an image.	
<video>	Embeds a video.	<video src="movie.mp4" controls></video>
<audio>	Embeds audio content.	<audio src="sound.mp3" controls></audio>
<source>	Specifies media resources for <video> and <audio> elements.	<source src="video.mp4" type="video/mp4">
<iframe>	Embeds another HTML page within the current page.	<iframe src="https://www.example.com"></iframe>

#### 4. List Tags

Tag	Description	Example
<ul>	Defines an unordered list.	<ul><i>Item 1</i><i>Item 2</i></ul>
<ol>	Defines an ordered list.	<ol><i>First</i><i>Second</i></ol>
<i>	Defines a list item.	<i>List Item</i>
<dl>	Defines a description list.	<dl><dt>Term</dt><dd>Description</dd></dl>
<dt>	Defines a term in a description list.	<dt>HTML</dt>
<dd>	Defines a description of the term in a description list.	<dd>Hypertext Markup Language</dd>

### 5. Form and Input Tags

Tag	Description	Example
<form>	Defines an HTML form for user input.	<form action="/submit"></form>
<input>	Defines an input control.	<input type="text" name="name">
<label>	Defines a label for an element.	<input>    <label for="name">Name:</label>
<button>	Defines a clickable button.	<button type="submit">Submit</button>
<select>	Defines a drop -down list.	<select><option>Option 1</option></select>
<option>	Defines an option in a list.	<option value="1">Option 1</option>
<textarea>	Defines a multi -line text input control.	<textarea rows="4" cols="50"></textarea>
<fieldset>	Groups related elements in a form.	<fieldset><legend>Group</legend></fieldset>
<legend>	Defines a caption for a <fieldset>.	<legend>Form Group</legend>

## 6. Table Tags

Tag	Description	Example
<table>	Defines a table.	<table></table>
<tr>	Defines a row in a table.	<tr><td>Cell 1</td><td>Cell 2</td></tr>
<th>	Defines a header cell in a table.	<th>Header</th>
<td>	Defines a standard cell in a table.	<td>Data</td>
<thead>	Groups header content in a table.	<thead><tr><th>Head</th></tr></thead>
<tbody>	Groups body content in a table.	<tbody><tr><td>Body</td></tr></tbody>
<tfoot>	Groups footer content in a table.	<tfoot><tr><td>Footer</td></tr></tfoot>
<caption>	Adds a caption to a table.	<caption>Table Title</caption>

### 7. Comments and Miscellaneous Tags

Tag	Description	Example
<!-- ... -->	Defines a comment in the HTML document.	<!-- This is a comment -->
<span>	Defines an inline section of text within a document.	<span class="highlight">Highlighted</span>
<div>	Defines a block -level section or division in an HTML document.	<div class="container">Content here</div>
<blockquote>	Defines a block of text that is a quotation from another source.	<blockquote>Quoted text</blockquote>
<pre>	Defines preformatted text (preserves whitespace and line breaks).	<pre>Preformatted text</pre>
<code>	Defines inline code or a code snippet.	<code>console.log('Hello')</code>
<kbd>	Defines keyboard input.	<kbd>Ctrl + C</kbd>
<samp>	Defines sample output from a program or computing device.	<samp>Sample Output</samp>

## PRACTICAL EXERCISE

It is time to extend the result of the previous exercise! In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy and rename the initial file, “**htm-structure.html**”, into three different files as thus:
  - a. “**html-tag-1.html**”
  - b. “**html-tag-2.html**”
  - c. “**html-tag-3.html**”
  
2. Use the first file “**html-tag-1.html**” to practice the use of Text formatting, Links, Media and List Tags.
3. Use the first file “**html-tag-2.html**” to practice the use of form and input Tags.
4. Use the first file “**html-tag-3.html**” to practice the use of Table and accompany Tags.



### Additional Resources:

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

- Video Links:

[https://www.youtube.com/watch?v=yIoO52MdZFE&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=3](https://www.youtube.com/watch?v=yIoO52MdZFE&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=3)

[https://www.youtube.com/watch?v=yMX901oVtn8&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=4](https://www.youtube.com/watch?v=yMX901oVtn8&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=4)

[https://www.youtube.com/watch?v=9gHPpwq6IaY&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=5](https://www.youtube.com/watch?v=9gHPpwq6IaY&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=5)

[https://www.youtube.com/watch?v=qis4kAOThLw&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=6](https://www.youtube.com/watch?v=qis4kAOThLw&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=6)

[https://www.youtube.com/watch?v=HA6bByKdAQm&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=12](https://www.youtube.com/watch?v=HA6bByKdAQm&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=12)

[https://www.youtube.com/watch?v=FmoYRiepmOE&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=13](https://www.youtube.com/watch?v=FmoYRiepmOE&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=13)

[https://www.youtube.com/watch?v=e62D-aayveY&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=14](https://www.youtube.com/watch?v=e62D-aayveY&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=14)

[https://www.youtube.com/watch?v=VLeERv\\_dR6Q&list=PLP9IO4UYNF0VdAajP\\_5pYG-jG2JRrG72s&index=22](https://www.youtube.com/watch?v=VLeERv_dR6Q&list=PLP9IO4UYNF0VdAajP_5pYG-jG2JRrG72s&index=22)

### Basic webpage using HTML

This is an example of a basic HTML webpage. This webpage includes a title, a header, a paragraph, an image, a list, and a link.

#### Example I of a Basic Webpage:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First Webpage</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    margin: 0;
    padding: 20px;
  }
  header {
    background-color: #0073e6;
    color: white;
    padding: 10px 0;
    text-align: center;
  }
  h1 {
    margin: 0;
  }
  .content {
    margin-top: 20px;
  }
  img {
    max-width: 100%;
    height: auto;
  }
  ul {
    list-style-type: square;
  }

```

```
        }
    a {
        color: #0073e6;
        text-decoration: none;
    }
    a:hover {
        text-decoration: underline;
    }
    footer {
        text-align: center;
        margin-top: 40px;
        font-size: 0.9em;
        color: #555;
    }
}
</style>
</head>
<body>
<header>
    <h1>Welcome to My First Webpage</h1>
</header>

<div class="content">
    <h2>About This Page</h2>
    <p>This is a simple webpage created to demonstrate the basic structure and elements of HTML. It includes headings, paragraphs, images, lists, and links.</p>

    <h3>My Favorite Things</h3>
    <p>Here are a few of my favorite things:</p>
    <ul>
        <li>Learning new programming languages</li>
        <li>Exploring new technologies</li>
        <li>Traveling to new places</li>
    </ul>

    <h3>My Favorite Picture</h3>
    

```

```
<h3>Learn More</h3>
<p>If you want to learn more about web development, visit <a href="https://www.w3schools.com" target="_blank">W3Schools</a> for great tutorials and resources.</p>
</div>

<footer>
  <p>© 2024 My First Webpage | Created by [Your Name]</p>
</footer>
</body>
</html>
```



### Explanation:

- **DOCTYPE Declaration:** `<!DOCTYPE html>` tells the browser that this is an HTML5 document.
- **HTML Element:** `<html lang="en">` wraps the entire content and indicates that the language of the document is English.
- **Head Element:** `<head>` contains meta-information about the document, including the charset, viewport settings, title, and internal CSS styling.
  - **Title Element:** `<title>` defines the title of the webpage, which appears on the browser tab.
  - **Style Element:** `<style>` contains internal CSS to style the elements on the page.
- **Body Element:** `<body>` contains the visible content of the webpage.
  - **Header:** `<header>` includes a main heading `<h1>` with a background color.
  - **Content Div:** `<div class="content">` contains various content elements such as headings `<h2>`, `<h3>`, paragraphs `<p>`, an unordered list `<ul>`, and an image `<img>`.
  - **Footer:** `<footer>` provides footer information, such as a copyright notice.

### How to View the Webpage:

1. Copy the HTML code above.
2. Open a text editor (such as Notepad, Sublime Text, or VS Code).
3. Paste the HTML code into a new file.
4. Save the file with a .html extension, e.g., index.html.
5. Open the saved file in a web browser to view the webpage.

## Welcome to My First Webpage

### About This Page

This is a simple webpage created to demonstrate the basic structure and elements of HTML. It includes headings, paragraphs, images, lists, and links.

### My Favorite Things

Here are a few of my favorite things:

- Learning new programming languages
- Exploring new technologies
- Traveling to new places

### My Favorite Picture



600 x 400

### Learn More

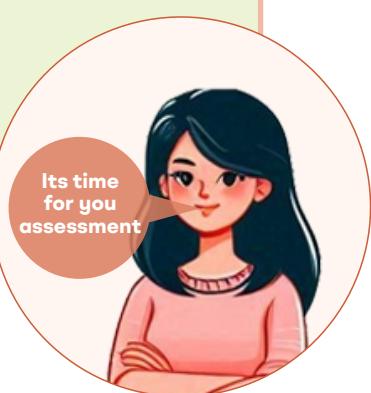
If you want to learn more about web development, visit [W3Schools](#) for great tutorials and resources.

© 2024 My First Webpage | Created by [Your Name]

## PRACTICAL EXCERCISE

In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy the code in an editor, save it and run it in a browser.
2. Listen to the instructor explain every part of the code to you.



Below is an example of another basic HTML webpage that demonstrates the use of table and form elements. This example includes a table for displaying data and a form that allows users to submit information.

### Example II of a Basic Webpage:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Webpage with Table and Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f9f9f9;
            margin: 0;
            padding: 20px;
        }
        header {
            background-color: #4CAF50;
            color: white;
            padding: 10px 0;
            text-align: center;
        }
        h1 {
            margin: 0;
        }
        .content {
            margin-top: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }
        table, th, td {
    
```

```
border: 1px solid #ddd;
}

th, td {
    padding: 12px;
    padding: 12px;
    text-align: left;
}

th {
    background-color: #4CAF50;
    color: white;
}

form {
    max-width: 600px;
    margin: 0 auto;
    background-color: #fff;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 5px;
}

form label {
    display: block;
    margin-bottom: 8px;
}

form input[type="text"],
form input[type="email"],
form input[type="submit"],
form textarea {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ddd;
    border-radius: 3px;
}

form input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    border: none;
}
```

```
        cursor: pointer;
    }
}

form input[type="submit"]:hover {
    background-color: #45a049;
}

</style>
</head>
<body>
    <header>
        <h1>Webpage with Table and Form</h1>
    </header>

    <div class="content">
        <h2>Sample Table</h2>
        <p>Below is a table showing a list of students and their scores:</p>
        <table>
            <thead>
                <tr>
                    <th>Student Name</th>
                    <th>Subject</th>
                    <th>Score</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>John Doe</td>
                    <td>Mathematics</td>
                    <td>85</td>
                </tr>
                <tr>
                    <td>Jane Smith</td>
                    <td>English</td>
                    <td>92</td>
                </tr>
                <tr>
                    <td>Tom Brown</td>
                    <td>Science</td>
                </tr>
            </tbody>
        </table>
    </div>

```

```

<td>78</td>
</tr>
</tbody>
</table>

<h2>Contact Us Form</h2>
<p>Please fill out the form below to get in touch with us:</p>
<form action="#" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="message">Message:</label>
  <textarea id="message" name="message" rows="5" required></textarea>

  <input type="submit" value="Submit">
</form>
</div>
</body>
</html>

```

**Explanation:**

- **Header:** The `<header>` section includes a main heading `<h1>` with a background color to give the page a title.
- **Table:**
  - The table is created using the `<table>` element.
  - The `<thead>` section contains the table header rows, with column headers defined by `<th>`.
  - The `<tbody>` section contains the data rows, with data cells defined by `<td>`.
  - Each row is represented by a `<tr>` element.

- **Form:**

- The form is enclosed within the `<form>` element, with the `action="#"` attribute indicating that the form submission will be handled within the same page (for this example).
- Form fields include `<input>` elements for text and email input, as well as a `<textarea>` for a multi-line text input.
- A submit button is created with `<input type="submit">` to allow users to submit the form.

Copy the code to an editor, save it like the first example and you should see the same result like the following image

**Webpage with Table and Form**

**Sample Table**

Below is a table showing a list of students and their scores:

Student Name	Subject	Score
John Doe	Mathematics	85
Jane Smith	English	92
Tom Brown	Science	78

**Contact Us Form**

Please fill out the form below to get in touch with us.

Name: \_\_\_\_\_

Email: \_\_\_\_\_

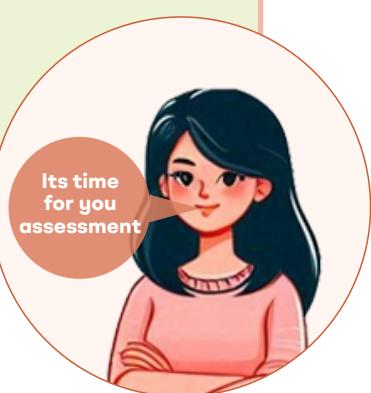
Message:  
\_\_\_\_\_

**Submit**

## PRACTICAL EXCERCISE

In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy the code in an editor, save it and run it in a browser.
2. Listen to the instructor explain every part of the code to you.

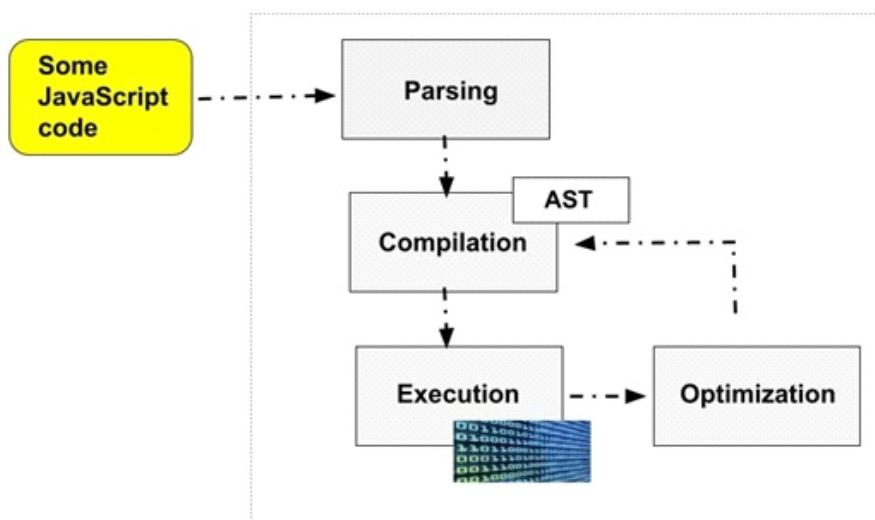


## How to make a website dynamic with Javascript

### How JavaScript works

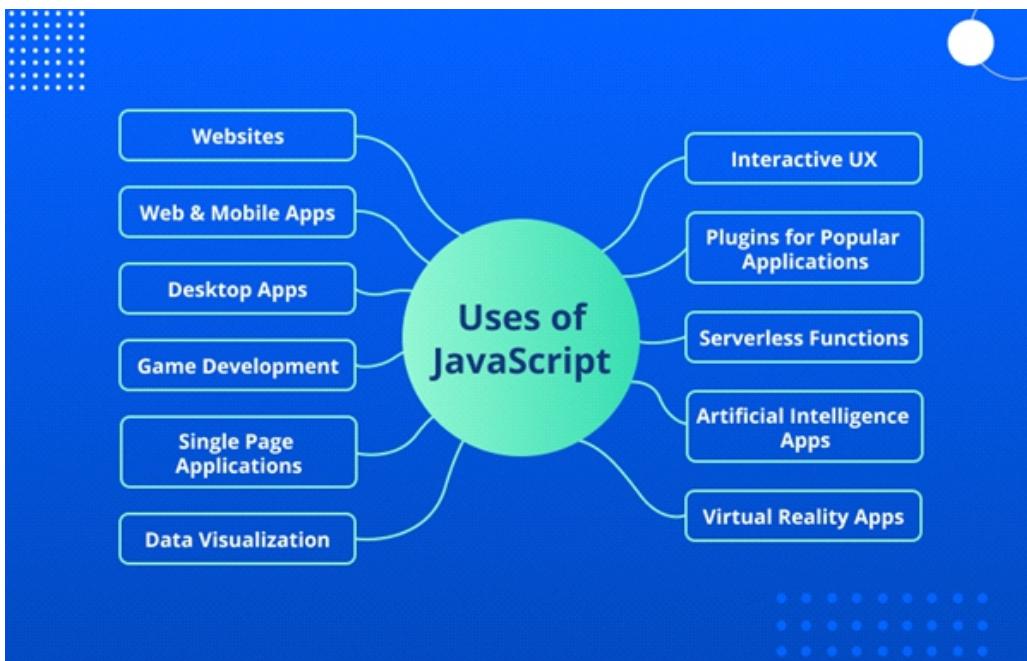
JavaScript is a high-level, interpreted programming language that enables interactive web pages. It runs on the client side (in the user's browser) and allows for dynamic content updates, interactive forms, animations, and more.

- Client-Side Execution: The browser executes JavaScript code, allowing it to respond to user actions without reloading the page.
- Event-Driven: JavaScript can respond to user interactions, such as clicks, mouse movements, and key presses.
- Integration with HTML/CSS: JavaScript works seamlessly with HTML and CSS to create interactive and dynamic web pages.



### Importance of JavaScript

1. **Interactivity:** JavaScript adds interactive elements to websites, enhancing the user experience.
2. **Dynamic Content:** It allows for dynamic content updates without reloading the page.
3. **Form Validation:** JavaScript can validate user inputs in real-time, ensuring data integrity.
4. **APIs and Third-Party Services:** JavaScript enables the integration of APIs and third-party services for added functionality.



### JavaScript code and syntax

#### 1. Data Types

- Primitive Types: `String`, `Number`, `Boolean`, `Undefined`, `Null`
- Non-Primitive Types: `Object`, `Array`, `Function`

#### 2. Data Structures

- Arrays: Ordered collections of items.
- Objects: Collections of key-value pairs.

#### 3. Functions

- Blocks of reusable code that can be called with parameters.

### How to define variables in JavaScript

1. Using `var`: Older way, function-scoped.
2. Using `let`: Modern way, block-scoped.
3. Using `const`: Modern way, block-scoped, cannot be reassigned.

### Functions in Javascript

#### 1. Function Declaration

```
function add(a, b) {
    return a + b;
}
```

#### 2. Function Expression

```
const multiply = function(a, b) {
    return a * b;
};
```

#### 3. Arrow Function (ES6)

```
const subtract = (a, b) => a - b;
```

- Practice this with your instructor

## Embedding JavaScript in HTML

JavaScript can be included in an HTML webpage in two main ways:

- **Inline JavaScript:** Directly within an HTML element using the onclick, onmouseover, onsubmit, etc., attributes.
- **Internal JavaScript:** Inside a <script> tag within the HTML document.
- **External JavaScript:** By linking an external .js file using the <script src="path/to/file.js"></script> tag.

## Basic Structure of an HTML Page with JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript in HTML</title>
</head>
<body>
    <h1>Welcome to My Webpage</h1>
    <p id="demo">Click the button to see a message.</p>
    <button onclick="displayMessage()">Click Me!</button>

    <script>
        // This is an internal JavaScript function
        function displayMessage() {
            document.getElementById("demo").innerText = "Hello, JavaScript!";
        }
    </script>

</body>
</html>
```

Practice this with your instructor

### Explanation of the Code

- **HTML Structure:**

- The `<!DOCTYPE html>` declaration defines the document type as HTML5.
- The `<html>` element is the root of the HTML document.
- The `<head>` section contains meta-information, like character encoding and the title of the webpage.
- The `<body>` section contains the content displayed on the webpage, including headings, paragraphs, and buttons.

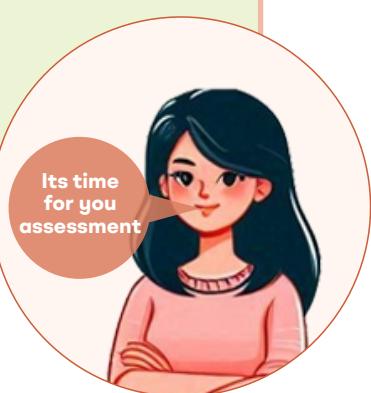
- **JavaScript Integration:**

- **Inline JavaScript:** The `onclick="displayMessage()"` attribute is attached to the button element. When the button is clicked, the `displayMessage()` function is called.
- **Internal JavaScript:** The `<script>` tag within the body of the HTML document contains a JavaScript function. This function changes the text content of the paragraph with the ID `demo`.

## PRACTICAL EXCERCISE

In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy the code in an editor, save it and run it in a browser.
2. Listen to the instructor explain every part of the code to you.



## Adding More Interaction, another example

Here's another example that adds a bit more interaction using JavaScript:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Interactive JavaScript Example</title>
    <style>
        .highlight {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <h1>Interactive JavaScript Example</h1>
    <p id="demo">Hover over this text to highlight it!</p>
    <button onclick="toggleHighlight()">Toggle Highlight</button>

    <script>
        // This function toggles the highlight class on the paragraph element
        function toggleHighlight() {
            document.getElementById("demo").classList.toggle("highlight");
        }

        // This function highlights the text when mouse is over the paragraph
        document.getElementById("demo").onmouseover = function() {
            this.style.backgroundColor = "yellow";
        };

        // This function removes the highlight when the mouse leaves the paragraph
        document.getElementById("demo").onmouseout = function() {
            this.style.backgroundColor = "";
        };
    </script>
</body>
</html>
```

### Explanation of the Code

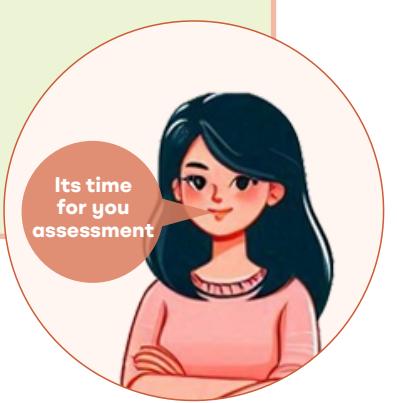
#### JavaScript for Interaction:

- **Toggle Highlight:** The `toggleHighlight()` function adds or removes the `highlight` class from the paragraph when the button is clicked.
- **Mouseover and Mouseout Events:** These events are used to change the background color of the paragraph when the mouse hovers over it or moves away from it.

## PRACTICAL EXCERCISE

In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy the code in an editor, save it and run it in a browser.
2. Listen to the instructor explain every part of the code to you.



It's time  
for you  
assessment

## Simple Form with JavaScript Validation

JavaScript can be used in an HTML webpage to create interactive forms, validate user input, and enhance the user experience. Below is an example that explains how JavaScript can be utilized to handle form operations on a basic HTML webpage

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Validation with JavaScript</title>
</head>
<body>
    <h1>Registration Form</h1>

    <form id="registrationForm" onsubmit="return validateForm()">
        <label for="username">Username:</label><br>
        <input type="text" id="username" name="username"><br><br>

        <label for="email">Email:</label><br>
        <input type="email" id="email" name="email"><br><br>

        <label for="password">Password:</label><br>
        <input type="password" id="password" name="password"><br><br>

        <input type="submit" value="Register">
    </form>

    <p id="errorMessages" style="color:red;"></p>

    <script>
        // JavaScript function to validate form input
        function validateForm() {
            var username = document.getElementById("username").value;
            var email = document.getElementById("email").value;
            var password = document.getElementById("password").value;
        }
    </script>
```

```
var errorMessages = "";

// Check if username is empty
if (username == "") {
    errorMessages += "Username is required.<br>";
}

// Check if email is valid
var emailPattern = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;
if (!email.match(emailPattern)) {
    errorMessages += "Please enter a valid email address.<br>";
}

// Check if password is empty
if (password == "") {
    errorMessages += "Password is required.<br>";
}

// Display error messages if any
if (errorMessages != "") {
    document.getElementById("errorMessages").innerHTML = errorMessages;
    return false; // Prevent form submission
}

alert("Form submitted successfully!");
return true; // Allow form submission
}

</script>
</body>
</html>
```

### Explanation:

#### 1. HTML Form Structure:

- The form consists of three fields: username, email, and password, each with its corresponding label.
- The form is given an id of registrationForm to reference it in the JavaScript code.
- The form's onsubmit attribute is set to call the validateForm() function when the user submits the form. The function will return false if the form validation fails, preventing the form from being submitted.

#### 2. JavaScript for Form Validation:

- **Function Definition:** The validateForm() function is defined within the <script> tag.
- **Input Validation:**
  - The function retrieves the values entered by the user in the username, email, and password fields using document.getElementById("elementID").value.
  - It then checks whether each field is filled out correctly. If a field is empty or invalid (e.g., an improperly formatted email address), an error message is added to the errorMessages string.
- **Email Validation:** The function uses a regular expression (emailPattern) to ensure the email is in a valid format.
- **Displaying Errors:** If there are any validation errors, they are displayed in the errorMessages paragraph (<p>) with the id="errorMessages". The form submission is blocked by returning false.
- **Successful Submission:** If all fields are valid, an alert message is displayed, and the form is submitted.

### How JavaScript Enhances the Form:

- **Real-time Validation:** JavaScript can provide immediate feedback to users before the form is submitted. This reduces the number of errors submitted to the server and enhances the user experience.
- **Preventing Invalid Submissions:** By blocking form submissions with invalid or incomplete data, JavaScript ensures that only correctly formatted data is sent to the server.

**Improved User Experience:** JavaScript allows for smoother interaction, such as showing error messages directly on the page without requiring a page reload.

Result

## Registration Form

Username:

Email:

Password:

Username is required.

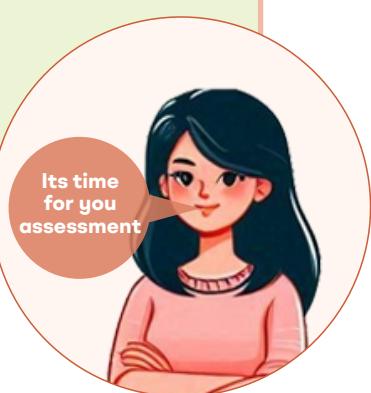
Please enter a valid email address.

Password is required.

## PRACTICAL EXCERCISE

In the presence of the instructor as a guide to you, carry out the following exercise:

1. Copy the code in an editor, save it and run it in a browser.
2. Listen to the instructor explain every part of the code to you.



### Implementation of Cascading Style Sheets (CSS) in a Website

#### What is Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS controls the layout, formatting, and overall visual appearance of web pages by specifying styles for elements like fonts, colors, spacing, and positioning. CSS is a powerful tool that helps web developers control the visual presentation of web pages, ensuring a consistent and appealing design across different browsers and devices.

#### The Importance of Cascading Style Sheets (CSS) in website design

- Separation of Concerns: CSS separates content (HTML) from presentation, making it easier to manage and update.
- Consistency: Ensures a consistent look and feel across multiple web pages.
- Design Flexibility: Provides extensive styling options for creative and responsive designs.
- Performance: External CSS files can be cached by browsers, improving page load times.

#### CSS Syntax Overview

CSS syntax consists of a selector and a declaration block. The selector targets the HTML elements to style, while the declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property and a value, specifying how the targeted element should be styled.

#### Example:

```
selector {  
    property: value;  
}
```



1. **Selectors:** Targets HTML elements to apply styles.

- ID Selector: `#id`
- Class Selector: `.class`
- Element Selector: `element`

### 2. Properties and Values:

- **Background:** `background-color`, `background-image`
- **Text:** `color`, `text-align`, `text-decoration`
- **Fonts:** `font-family`, `font-size`, `font-weight`
- **Links:** `a:link`, `a:visited`, `a:hover`, `a:active`
- **Tables:** `border-collapse`, `padding`, `text-align`
- **Box Model:** `border`, `margin`, `padding`, `width`, `height`
- **Positioning:** `position`, `top`, `bottom`, `left`, `right`
- **Floating:** `float`, `clear`
- **Alignment:** `vertical-align`, `text-align`
- **Colors:** Named colors, HEX codes (`#000000`), RGB (`rgb(0,0,0)`)

Here are some few examples example

#### ID & Class Selectors

```
#header {  
    background-color: blue;  
}  
  
.button {  
    background-color: green;  
}
```



#### Styling Backgrounds

```
body {  
    background-color: lightblue;  
    font-family: Arial, sans-serif;  
}  
  
body {  
    background-image: url('image.jpg');  
}  
  
body {  
    background-repeat: no-repeat;  
}
```





Practice this with your instructor

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Selectors Example</title>
    <style>
        /* ID Selector */
        h1 {
            background-color: #f1f1f1;
            padding: 20px;
            text-align: center;
        }

        /* Class Selector */
        button {
            background-color: #4CAF50;
            color: white;
            padding: 15px 32px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
        }

        /* Element Selector */
        p {
            font-size: 16px;
            line-height: 1.5;
        }
    </style>
</head>
<body>

    <!-- Header Section using ID Selector -->
    <h1 id="main-title">Welcome to My Webpage</h1>
```



Practice this with your instructor

```
<!-- Paragraph using Element Selector -->  
<p>This is a sample paragraph demonstrating the use of the element selector in CSS. The text is  
styled with a larger font size and increased line height for better readability.</p>  
  
<!-- Button using Class Selector -->  
<button class="btn">Click Me</button>  
  
</body>  
</html>
```

### Explanation:

#### 1. ID Selector (**h1**):

- Applies styles to the `<h1>` element, setting a light gray background color, adding padding, and centering the text.

#### 2. Class Selector (**button**):

- Applies styles to the `<button>` element. It sets a green background, white text, padding, and centers the text within the button.

#### 3. Element Selector (**p**):

- Targets all `<p>` elements in the document, setting the font size to 16px and the line height to 1.5 for improved readability.

### Result

# Welcome to My Webpage

This is a sample paragraph demonstrating the use of the element selector in CSS. The text is styled with a larger font size and increased line height for better readability.

**Click Me**

### Use of CSS in Webpage

CSS can be used in a webpage in three main ways:

- Inline CSS
- Internal CSS and
- External CSS.

Each method has its own use case and advantages.

#### 1. Inline CSS

Inline CSS is used directly within an HTML element's style attribute. This method applies CSS rules to a specific element on the page.

```
<p style="color: blue; font-size: 18px;">This is a paragraph with inline CSS.</p>
```



##### Advantages:

- Quick and easy for small changes.
- Useful for applying unique styles to a single element.
- No need for separate CSS files.

##### Disadvantages:

- Not reusable; styles must be repeated for each element.
- Can clutter HTML code and make it harder to maintain.
- Not efficient for larger projects.

#### 2. Internal CSS

Internal CSS is used within a `<style>` tag in the `<head>` section of an HTML document. This method applies CSS rules to the entire webpage.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {
            background-color: lightgray;
        }
        p {
            color: red;
        }
    </style>
</head>
<body>
    <h1>Hello World!</h1>
    <p>This is a paragraph with internal CSS.</p>
</body>
</html>
```

```
        color: green;  
        font-size: 20px;  
    }  
</style>  
<title>Internal CSS Example</title>  
</head>  
<body>  
    <p>This is a paragraph styled with internal CSS.</p>  
</body>  
</html>
```

### Advantages:

- Centralized control over the styles of a single webpage.
- Cleaner HTML since styles are separated into the `<style>` tag.
- No external file is required, making it easier to manage small projects.

### Disadvantages:

- Styles are limited to the specific webpage; not reusable across multiple pages.
- Not efficient for larger websites with multiple pages.
- Can lead to code duplication if the same styles are needed on different pages.

## External CSS

External CSS is written in a separate .css file and linked to the HTML document using the `<link>` tag in the `<head>` section. This method allows you to apply the same styles to multiple webpages.

### Example:

#### HTML file (index.html):

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <link rel="stylesheet" href="styles.css">  
<title>External CSS Example</title>  
</head>
```

```
<body>
  <p>This is a paragraph styled with external CSS.</p>
</body>
</html>
```

SS file (styles.css):

```
body {
  background-color: white;
}
p {
  color: purple;
  font-size: 22px;
}
```

### Advantages:

- Separation of concerns: HTML structure and CSS styling are kept separate.
- Reusability: The same CSS file can be used across multiple webpages.
- Easier to maintain and update styles for a large website.
- Reduces page load times because the CSS file can be cached by browsers.

### Disadvantages:

- Requires an extra HTTP request to load the CSS file, potentially increasing load times (though often negligible).
- More complex to set up initially compared to inline or internal CSS.

### Summary:

Inline CSS	Internal CSS	External CSS
<b>Is useful for small, one-off style changes directly within HTML tags.</b>	Is effective for styling a single webpage with multiple elements.	Is the most efficient and scalable method, ideal for larger websites where styles need to be consistent across multiple pages.

## PRACTICAL EXERCISE

1. With your instructor, practice the use of the following:

- Inline CSS
- Internal CSS and
- External CSS.

2. Download a simple webpage and carry out the following activities with your instructor:

- Observe and discuss the html structure off the page.
- Observe and discuss how the CSS is implemented.



### Additional Resources:



<https://www.w3schools.com/css/default.asp>

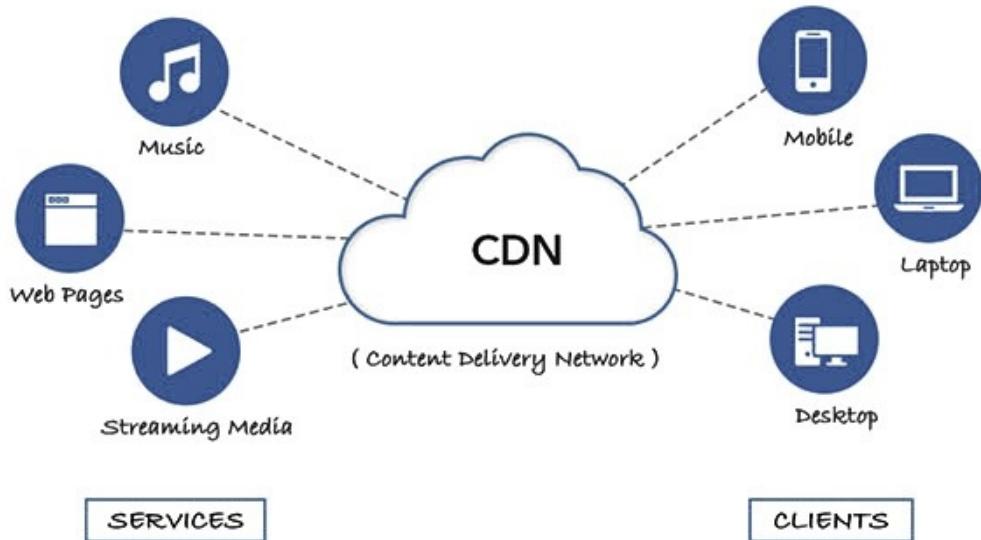
- Video Links:

<https://www.youtube.com/watch?v=AGDDdsiZ0Ko>

<https://www.youtube.com/watch?v=G8r00ZNopTE>

## Content Delivery Network

A Content Delivery Network (CDN) is a distributed network of servers strategically located worldwide. CDNs deliver web content and resources to users based on their geographical location, improving load times and performance.



## How to use CDN for CSS and JavaScript

### 1. Including CSS from a CDN

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is a paragraph of text on my webpage.</p>
</body>
</html>
```



### 2. Including JavaScript from a CDN

```
<!DOCTYPE html>
<html>
<head>
<title>My Web Page</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>
<h1>Welcome to My Website</h1>
<script>
$(document).ready(function(){
    alert("Hello, World!");
});
</script>
</body>
</html>
```



### 3. Including Bootstrap from a CDN

```
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
crossorigin="anonymous">
</head>
<body>
<div class="container">
<div class="row">
<div class="col border">Column 1</div>
<div class="col border">Column 2</div>
<div class="alert alert-primary mt-2">
```



```
This is an alert box using Bootstrap</div>
</div>
</div>

</body>
</html>
```



### Importance of CDN

- Improved Performance: Faster content delivery by serving users from the nearest server.
- Reliability: High availability and redundancy across multiple servers.
- Scalability: Efficiently handles high traffic volumes.
- Security: Enhanced security features like DDoS protection.

The advantages and disadvantages of Content Delivery Network (CDN) over Local Installation of front-end tools

### Advantages

- Speed: Faster load times due to proximity to users.
- Efficiency: Reduced bandwidth usage and server load.
- Global Reach: Consistent user experience worldwide.

### Disadvantages

- Cost: Potentially higher costs for premium CDN services.
- Dependency: Reliance on third-party services.
- Complexity: Additional configuration and management.

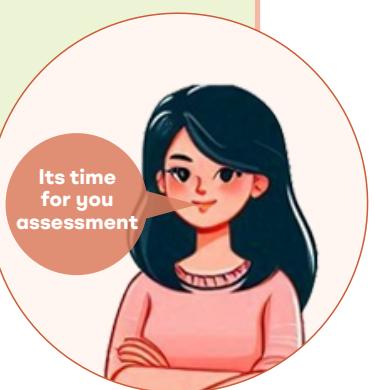
## PRACTICAL EXERCISE

### 1. Using CDNs:

- Include CSS and JavaScript libraries from CDNs in your web pages.
- Compare the load times with and without using CDNs.

### 2. Evaluating CDNs:

- Research and list different CDN providers.
- Analyze the advantages and disadvantages of each provider.



### Additional Resources:



#### - Video Links:

- <https://www.youtube.com/shorts/b6RYQkmRhho>
- <https://www.youtube.com/watch?v=SdVWAX2Js-g>
- <https://www.youtube.com/watch?v=uOxg3KZm7Fw>
- <https://www.w3schools.com/bootstrap5/index.php>



*Now, can you confidently say repeat these statements to yourself:*

*I can:*

- Write HTML for a website
- Use JavaScript to make website dynamic
- Implement Cascading Style Sheets (CSS) in a website
- Use Content Delivery Network (CDN).



**MODULE 09**

# DESIGNING A WEBSITE



This unit equips learners with fundamental web design knowledge and skills, including using web design software, researching and customizing templates, and understanding UI/UX principles. Learners will gain the ability to design simple websites, manage files across platforms, and differentiate between web design and development.

**At the end of this Module, learners should be able to:**

- Understand basic web design elements and layouts.
- Use various web design software tools.
- Research and customize website templates.
- Manage file imports/exports across platforms.
- Design a simple website with an understanding of UI/UX principles.



### Basics Of Web Design

#### The Basic Elements of Web Design

##### Elements of Web Design:

- **Layout:** The arrangement of visual elements on a page. It determines how content is structured and displayed.
- **Color:** Choice of colors to create visual interest and evoke emotions. Color schemes and palettes are crucial for brand identity.
- **Graphics:** Use of images, icons, and other visual elements to enhance the visual appeal and user experience.
- **Fonts:** Typography used to enhance readability and convey the brand's voice. Different fonts can set different tones and moods.
- **Content:** Information presented on the website, including text, images, and multimedia. Quality content is key to user engagement.
- **Navigation:** How users move through the site. Effective navigation ensures users can find what they're looking for easily.



#### Differences between Web Design and Web Development

Web Design	Web Development
Focuses on the aesthetics, layout, and usability of a website.	Involves coding and building the structure of the website.
It involves creating the visual elements and user interface	It includes both front-end (client-side) and back-end (server-side) development.

	<b>WEB DESIGNER</b>	<b>WEB DEVELOPER</b>
<b>SKILLS</b>		
<b>TOOLS</b>		
Layout design	✓	✗
Color theory	✓	✗
Typography	✓	✗
User Experience	✓	✗
HTML/CSS	Good to know	✓
JavaScript	Good to know	✓
Git + GitHub	✗	✓
Back-end code	✗	✓ (sometimes)
Photoshop	✓	Good to know
XD/Figma/Sketch	✓	✗
Code editor	✗	✓
Web frameworks	✗	✓

## Differences

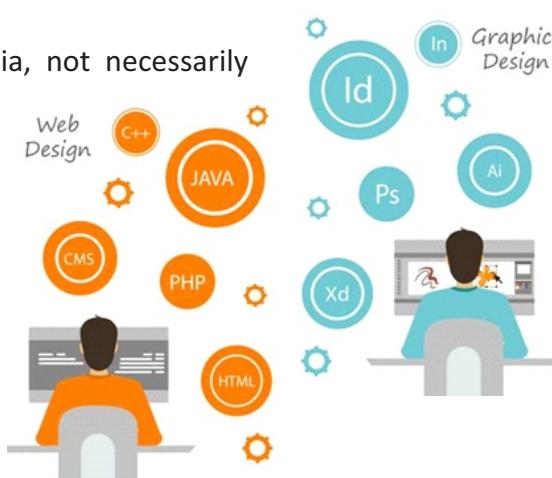
### Web Design

- Creating functional and user-friendly web pages that are interactive and responsive.

### Graphic Design

- Creating visual content for various media, not necessarily interactive.

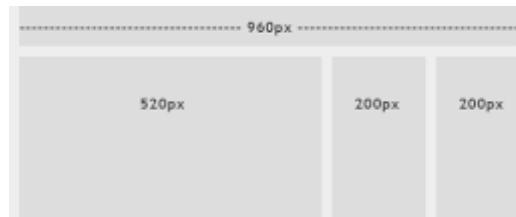
It focuses on aesthetics and communication through visuals



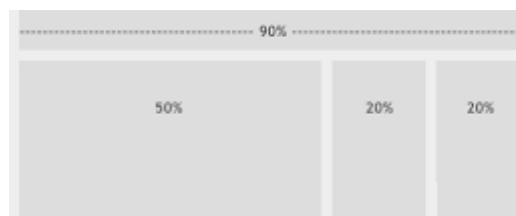
### Different Types of Web Design Layout

#### Types of Layouts:

- Fixed Layout: Fixed width, does not change with the browser size. Suitable for websites with a lot of fixed-width content.



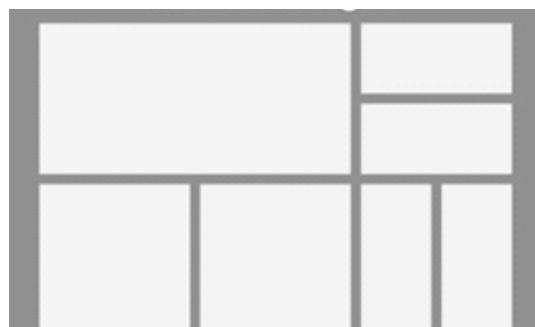
- Fluid Layout: Uses percentages for widths, adjusts with the browser size. Provides flexibility across different screen sizes.



- Responsive Layout: Adapts to different devices and screen sizes using media queries. Essential for modern web design to ensure accessibility on all devices.



- Grid Layout: Uses a grid system for structure and alignment. Helps in creating balanced and organized designs.



### The Design Grid System

Design Grid System: A framework that divides the page into columns and rows to ensure consistent alignment and spacing. It helps in creating balanced and visually appealing layouts.

#### Code Example with HTML and CSS of Design Grid System

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Design Grid System Example</title>
<style>
.container {
  display: grid;
  grid-template-columns: repeat(12, 1fr); / 12-column grid /
  gap: 20px; / Spacing between grid items /
  max-width: 1200px; / Maximum width of the container /
  margin: 0 auto; / Center align the container /
}
.item {
  background-color: f0f0f0;
  padding: 20px;
  border: 1px solid ccc;
  text-align: center;
}
.item-1 {
  grid-column: span 6; / This item spans 6 columns /
}
.item-2 {
  grid-column: span 4; / This item spans 4 columns /
}
.item-3 {
  grid-column: span 2; / This item spans 2 columns /
}
</style>
</head>
<body>
<div class="container">
<div class="item item-1">Content Area 1 (Span 6 columns)</div>
<div class="item item-2">Sidebar (Span 4 columns)</div>
<!--You can add more items as needed -->
</div>
</body>
</html>
```



#### RESULT

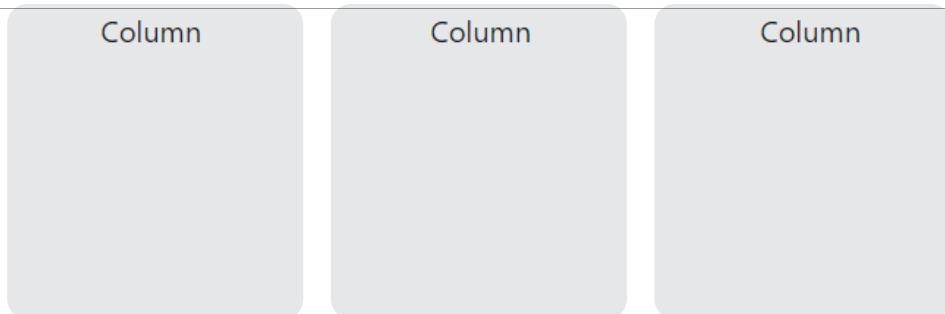
Content Area 1 (Span 6 columns)

Sidebar (Span 4 columns)

### Bootstrap Grid System

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bootstrap Design Grid System Example</title>
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container">
<div class="row">
<div class="col">
Column
</div>
<div class="col">
Column
</div>
<div class="col">
Column
</div>
<div class="col">
Column
</div>
</div>
</div>
</body>
</html>
```

### RESULT



### Basic Web Design and Use of Forms Using Bootstrap

Bootstrap is a powerful, open-source front-end framework used to create modern, responsive websites quickly and efficiently. It includes HTML, CSS, and JavaScript components that are reusable and customizable, making web development more streamlined.

#### Setting Up Bootstrap

##### 1. Using Bootstrap CDN

To start using Bootstrap, you can include the following links in your HTML file:

##### 2. Using Bootstrap Locally

Download Bootstrap from the official website, and include the downloaded CSS and JS files in your project.

### Basic Web Design with Bootstrap

#### Creating a Simple Layout

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <title>Basic Web Design</title>
</head>
<body>
    <div class="container">
        <header class="bg-primary text-white text-center py-3">
            <h1>Welcome to My Website</h1>
        </header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="">Navbar</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item active">
                        <a class="nav-link" href="">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="">Features</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="">Pricing</a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>
</body>
```



```
</div>
</nav>
<main class="my-4">
  <div class="row">
    <div class="col-md-8">
      <h2>Main Content</h2>
      <p>This is the main content area.</p>
    </div>
    <div class="col-md-4">
      <h2>Sidebar</h2>
      <p>This is the sidebar area.</p>
    </div>
  </div>
</main>
<footer class="bg-secondary text-white text-center py-3">
  <p>&copy; 2024 My Website</p>
</footer>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```



## RESULT

A screenshot of a website demonstrating the layout. The header is a blue bar with the text "Welcome to My Website". Below it is a light gray navigation bar containing the word "Navbar" on the left and a three-line menu icon on the right. The main content area contains the heading "Main Content" and the text "This is the main content area.". To the right is a sidebar with the heading "Sidebar" and the text "This is the sidebar area.". At the bottom is a dark gray footer bar with the copyright text "© 2024 My Website".

Welcome to My Website

Navbar

Main Content

This is the main content area.

Sidebar

This is the sidebar area.

© 2024 My Website

### Using Forms in Bootstrap

Bootstrap provides a variety of classes to create and style forms easily.

#### Basic Form Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <title>Bootstrap Form</title>
</head>
<body>
    <div class="container">
        <h2 class="my-4">Contact Us</h2>
        <form>
            <div class="form-group">
                <label for="name">Name</label>
                <input type="text" class="form-control" id="name" placeholder="Enter your name">
            </div>
            <div class="form-group">
                <label for="email">Email address</label>
                <input type="email" class="form-control" id="email" placeholder="Enter your email">
            </div>
            <div class="form-group">
                <label for="message">Message</label>
                <textarea class="form-control" id="message" rows="3" placeholder="Enter your message"></textarea>
            </div>
            <button type="submit" class="btn btn-primary">Submit</button>
        </form>
    </div>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```



## RESULT

## Contact Us

Name

Email address

Message



### Customizing Forms with Bootstrap

You can further customize forms using Bootstrap classes:

- **Form Control Size:** Use `form-control-lg` or `form-control-sm` to change the size of inputs.
- **Inline Forms:** Use `form-inline` to create inline forms.
- **Form Grid:** Use Bootstrap grid classes like `row` and `col` to create complex form layouts.

### Example of Custom Form Layout: Registration Form

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
  <title>Custom Form Layout</title>
</head>
<body>
  <div class="container">
    <h2 class="my-4">Registration Form</h2>
```



```
<form>

<div class="form-row">
    <div class="form-group col-md-6">
        <label for="firstName">First Name</label>
        <input type="text" class="form-control" id="firstName" placeholder="First Name">
    </div>
    <div class="form-group col-md-6">
        <label for="lastName">Last Name</label>
        <input type="text" class="form-control" id="lastName" placeholder="Last Name">
    </div>
</div>
<div class="form-group">
    <label for="inputEmail4">Email</label>
    <input type="email" class="form-control" id="inputEmail4" placeholder="Email">
</div>
<div class="form-group">
    <label for="inputPassword4">Password</label>
    <input type="password" class="form-control" id="inputPassword4" placeholder="Password">
</div>
<div class="form-group">
    <label for="inputAddress">Address</label>
    <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St">
</div>
<div class="form-group">
    <label for="inputCity">City</label>
    <input type="text" class="form-control" id="inputCity">
</div>
<div class="form-group">
    <label for="inputState">State</label>
    <select id="inputState" class="form-control">
        <option selected>Choose...</option>
        <option>...</option>
    </select>
</div>
```



```
<div class="form-group">
    <label for="inputZip">Zip</label>
    <input type="text" class="form-control" id="inputZip">
</div>
<button type="submit" class="btn btn-primary">Register</button>
</form>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```



### RESULT

**Registration Form**

First Name

Last Name

Email

Password

Address

City

State

Zip

**Register**

### Login Form

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <title>Login Form</title>
</head>
<body>
    <div class="container">
        <h2 class="my-4">Login Form</h2>
        <form>
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" class="form-control" id="email" placeholder="Enter your email">
            </div>
            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" class="form-control" id="password" placeholder="Enter your password">
            </div>
            <div class="form-group form-check">
                <input type="checkbox" class="form-check-input" id="rememberMe">
                <label class="form-check-label" for="rememberMe">Remember Me</label>
            </div>
            <button type="submit" class="btn btn-primary">Login</button>
        </form>
    </div>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```



**RESULT**

The screenshot shows a simple login form. At the top is a large title 'Login Form'. Below it is a label 'Email' followed by an input field containing placeholder text 'Enter your email'. Below that is a label 'Password' followed by another input field with placeholder text 'Enter your password'. Underneath the password field is a small checkbox labeled 'Remember Me'. At the bottom is a blue rectangular button with the word 'Login' in white.

Both forms are contained within a **.container** class to provide padding and centering.

The registration form uses a **.form-row** class and multiple **.form-group** classes to create a grid layout.

The login form is simpler, with a few **.form-group** classes to organize the inputs and a **.form-check** class for the "Remember Me" checkbox.

Bootstrap's **.form-control** class is used to style input fields consistently.

Submit buttons are styled using Bootstrap's **.btn** and **.btn-primary** classes.

**How to use Bootstrap to create containers, rows, and columns, and how to insert images and videos.**

#### **Basic Structure with Containers, Rows, and Columns**

Bootstrap's grid system allows you to create responsive layouts using a series of containers, rows, and columns.

#### **Inserting Images**

Bootstrap makes it easy to include and style images. You can use the **.img-fluid** class to make images responsive, ensuring they scale with the parent element.

## Example Code

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <h2>Image Example</h2>
      
    </div>
  </div>
</div>
```



## Inserting Videos

Bootstrap doesn't include specific classes for videos, but you can make videos responsive by embedding them within a responsive iframe container.

## Example with Video

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <h2>Video Example</h2>
      <div class="embed-responsive embed-responsive-16by9">
        <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/dQw4w9WgXcQ" allowfullscreen></iframe>
      </div>
    </div>
  </div>
</div>
```



HTML   CSS   JS   Result

⚙️ ⏮

```
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <h2>Video Example</h2>
      <div class="embed-responsive embed-responsive-16by9">
        <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/dQw4w9WgXcQ" allowfullscreen></iframe>
      </div>
    </div>
  </div>
</div>
```

**Video Example**

A screenshot of a browser showing a YouTube video player. The video is titled 'Rick Astley - Never Gonna Give You Up'. The player has a play button and a progress bar. The video frame shows Rick Astley in his signature suit and striped shirt.

Container: The **.container** class centers your content and adds some padding.

Row: The **.row** class is used to create a horizontal group of columns.

Columns: The **.col-\*** classes are used to define the width of each column, where `\*` can be **1-12** to specify the width in a 12-column grid system.

Images: The **.img-fluid** class makes an image responsive (it will scale nicely to the parent element).

Videos: The **.embed-responsive** class along with `embed-responsive-16by9` (or other aspect ratios like `4by3`, `1by1`, etc.) creates a responsive video container.

## PRACTICAL EXCERCISE

- Create a fixed layout and a fluid layout web page.
- Design a simple responsive web page using a grid layout.



Its time  
for you  
assessment

### Additional Resources:



<https://getbootstrap.com/docs/5.0/layout/grid/>

### Video links:

<https://www.youtube.com/watch?v=Hjl6gbg9kmk>

<https://www.youtube.com/watch?v=F4fbwKV9dBU>

<https://www.youtube.com/watch?v=AaVy9UR30Dc>

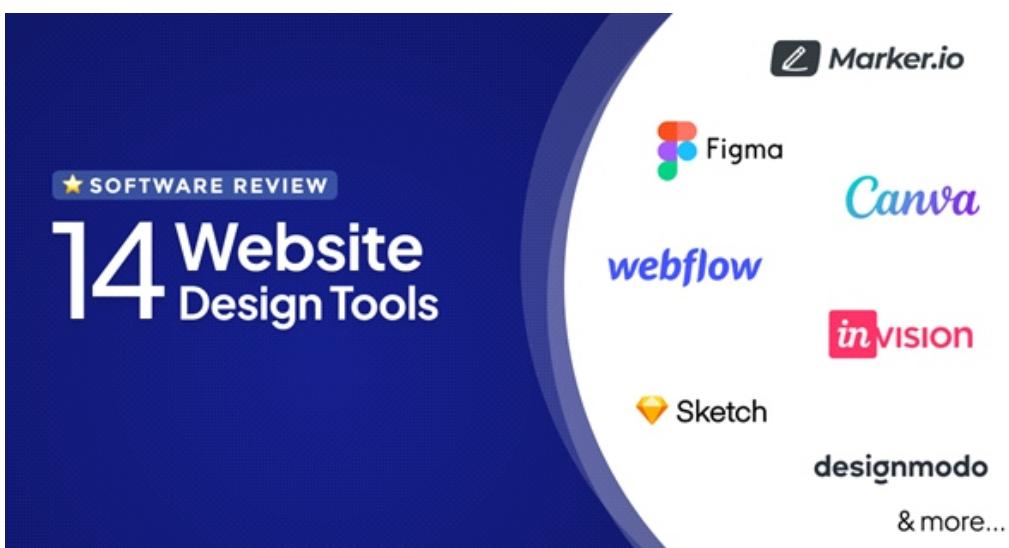
[https://www.youtube.com/watch?v=Wqu-d\\_b3K-0](https://www.youtube.com/watch?v=Wqu-d_b3K-0)

## Different Web Design Software

### Web Designing Software Tools

Common Tools:

- **Figma:** Collaborative interface design tool that allows real-time collaboration and prototyping.
- **Sketch:** Vector graphics editor primarily used for UI and UX design.
- **Adobe XD:** Tool for designing and prototyping user experiences with features for creating wireframes, prototypes, and screen designs.



### Features and Options of Different Designing Software

#### Features to Explore:

- **Prototyping:** Creating interactive models of your design to test functionality and user experience.
- **Collaboration:** Sharing designs and getting feedback from team members in real time.
- **Plugins:** Extending the functionality of the software with additional tools and integrations.

### Installation of Different Web Design software on a Computer System

Installing web design software on a computer system typically involves downloading the installer from the software's official website and running it. Here's a general guide on installing some commonly used web design tools:

#### 1. Adobe Creative Cloud (Adobe XD, Photoshop, Illustrator):

- Visit Adobe's website and sign in or create an Adobe ID.
- Download the Creative Cloud installer.
- Run the installer and follow the prompts to select which applications (e.g., Adobe XD, Photoshop, Illustrator) you want to install.

- Once installed, sign in with your Adobe ID to activate the software.

### 2. Sketch:

- Go to the Sketch website and download the installer.
- Open the downloaded installer package (.dmg file on macOS).
- Drag the Sketch icon into your Applications folder.
- Open Sketch from your Applications folder to complete the installation.

### 3. Figma:

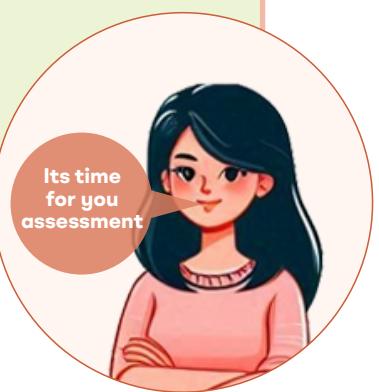
- Figma is primarily web-based, so you can use it directly in your web browser.
- Optionally, you can install the Figma desktop app:
- Go to the Figma website and download the desktop app for your operating system.
- Run the installer and follow the on-screen instructions to complete the installation.

### 5. Visual Studio Code (VS Code):

- Go to the Visual Studio Code website and download the installer for your OS.
- Run the installer and follow the prompts to install VS Code.
- After installation, launch VS Code and install any necessary extensions for web development (e.g., HTML, CSS, JavaScript).

## PRACTICAL EXCERCISE

- Install Figma and Adobe XD on your computer.
- Create a simple prototype using Figma or Adobe XD.
- Create a simple wordpress site.



### Additional Resources:



#### Video Links:

- <https://www.youtube.com/watch?v=D4NyQ5iOMF0>
- <https://www.youtube.com/watch?v=FO5TM0spOZk>
- <https://www.youtube.com/watch?v=dXQ7IHkTiMM>

### Searching for Website Templates and Custom Web Design

#### What is a Website Template?

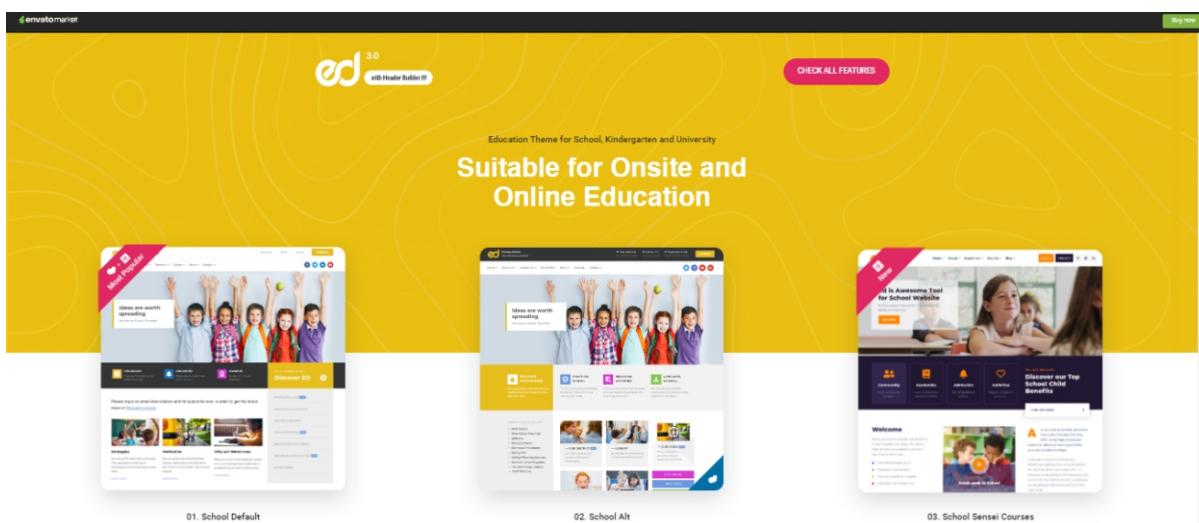
Website Template is a pre-designed webpage or set of webpages that you can customize with your content and images. Templates save time and provide a starting point for website design.

#### The Website Template Suited for a Particular Medium

##### Examples:

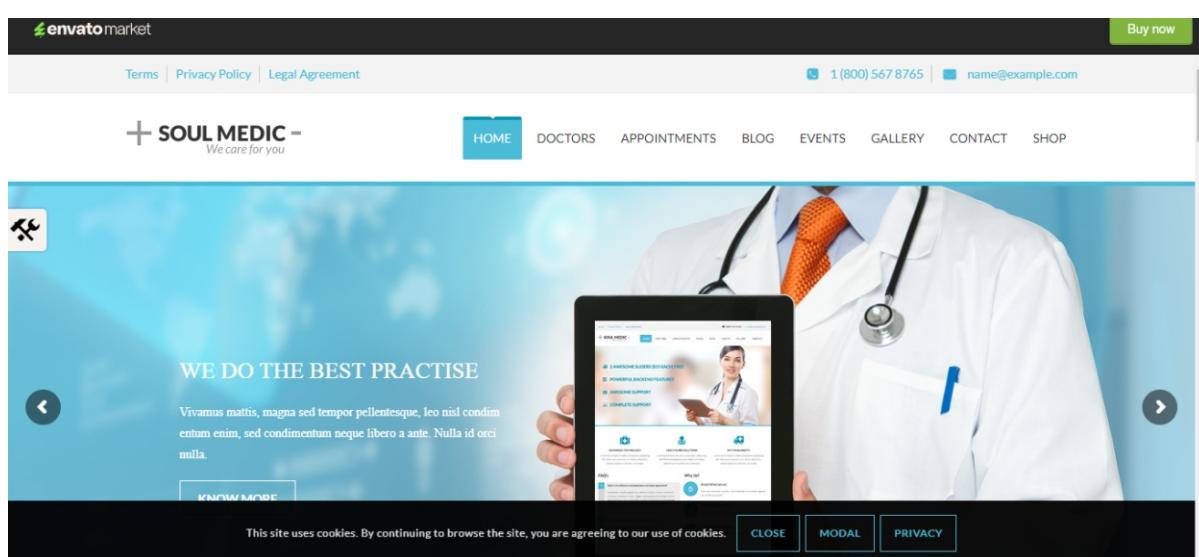
##### School Template:

Educational resources, class schedules, and student information.



##### Hospital Template:

Medical services, appointment booking, and patient information.



### How to search Free and Paid Website Templates on the Internet

#### Searching Templates:

- Use search engines or template marketplaces.
- Filter results based on your needs (e.g., free vs. paid, industry-specific). Evaluate templates based on features, responsiveness, and customization options.

### How to download Website Templates to a Computer

#### Download Steps:

- Find a template you like.
- Click the download button.
- Save the file to your computer and unzip it if necessary.

### How to test Website Templates Using Any Web Browser

#### Testing Templates:

- Open the template files in a web browser.
- Check for responsiveness, compatibility across different browsers, and overall functionality.

### How to edit different Website Templates in Any Code Editor

#### Editing Templates:

- Open the template files in a code editor like Sublime Text or Visual Studio Code.
- Make necessary changes to HTML, CSS, and JavaScript files to customize the template according to your needs.

### How to use custom Web Design Tools and Programs

#### Custom Design Process:

- Use tools like Figma or Adobe XD to create your own design from scratch.
- Implement the design using HTML, CSS, and JavaScript, ensuring that it meets the project requirements.

### The difference between Web Template and Custom Web Design

- Web Template: Pre-made design, faster setup, limited customization.
- Custom Web Design: Unique design, more control, more time-consuming, but allows for complete customization and uniqueness.

### Identification of different Copyrights and Intellectual Property Right Licenses

#### Types of Licenses:

- Creative Commons: Free to use with some conditions, often used for educational or creative content.
- Commercial License: Purchased for commercial use, providing more rights and flexibility.

Open Source: Free to use, modify, and distribute, often used in software and web development.

## PRACTICAL EXCERCISE

- Find and download a website template.
- Edit the template using a code editor and customize it according to a specific theme (e.g., a school website).



### Additional Resources:



- Video Links:

<https://www.youtube.com/watch?v=zBpuKybGUfs>

<https://www.youtube.com/watch?v=tmeOOL58KLM&t=74s>

<https://www.youtube.com/watch?v=bqnmDc2FdKg>

## How to Import and Export Files Across Different Platforms

### Types of Files Commonly Used in Web Development

- Image Files: JPG, PNG, GIF – used for graphics and photos.
- Document Files: PDF, DOCX – used for downloadable content and documentation.



### Platform-Specific Procedures for Importing and Exporting Files

Importing and exporting files can vary based on the platform and the specific software you are using. Here are the general procedures for common platforms:

#### Windows

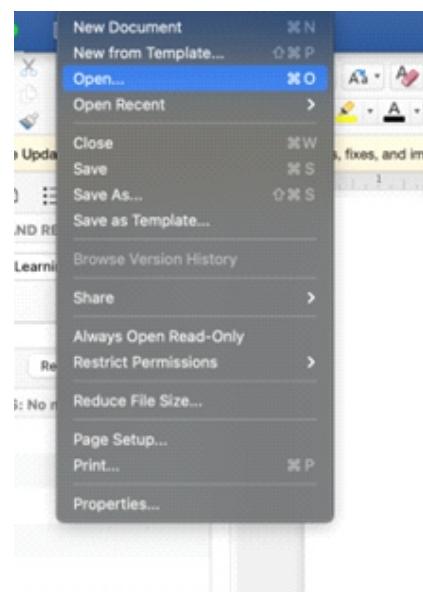
Importing Files:

##### 1. Using File Explorer:

- Navigate to the folder where you want to import files.
- Copy or move the files from their source location (e.g., another folder, or external drive) into the desired folder using drag-and-drop or copy-paste.

##### 2. Using Software:

- Open the software application (e.g., Adobe Photoshop, Microsoft

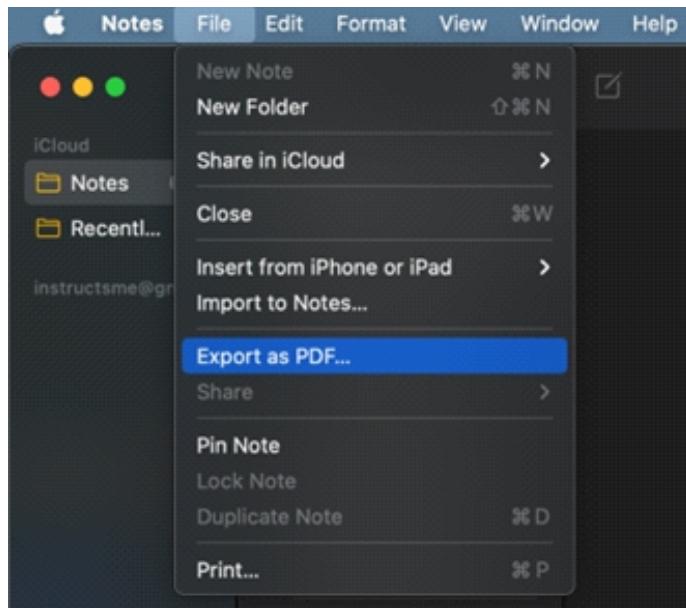


- Go to the "File" menu and choose "Open" or a similar option.
- Navigate to the location of the file you want to import and select it.

## Exporting Files:

### 1. Using Software:

- Open the software application (e.g., Adobe Illustrator, Notepad).
- Create or open the file you want to export.
- Go to the "File" menu and choose "Save As" or "Export as".
- Choose the file format and location where you want to save the exported file.
- Optionally, adjust export settings such as quality or resolution.



## Common Software Specific Procedures:

Microsoft Office (Word, Excel, PowerPoint)

- Importing: Open the application, go to "File" > "Open" and select the file.
- Exporting: Open the application, go to "File" > "Save As" and choose the format and location.

## Compatibility Issues While Exporting a File from One Software Application to Another

### Compatibility Issues:

- Different file formats and how they are handled by various software.
- Ensuring that exported files maintain their integrity and functionality across different platforms.

## Knowledge of File Types and Extensions Effectively in Practical Web Development Tasks

Understanding file types and extensions is crucial in web development for handling various content and ensuring compatibility across different platforms and browsers.

### Some key file types and extensions commonly used in practical web development tasks:

#### 1. HTML (Hypertext Markup Language)

- Extension: ` `.html` , ` .htm`
- Description: HTML files are the backbone of web pages, defining the structure and content using markup tags.

#### 2. CSS (Cascading Style Sheets)

- Extension: ` `.css`
- Description: CSS files define the visual presentation and layout of HTML elements, including styles for fonts, colors, margins, and more.

#### 3. JavaScript

- Extension: ` `.js`
- Description: JavaScript files contain scripts that add interactivity and dynamic behavior to web pages, enabling functions like form validation, animations, and AJAX requests.

#### 4. Images

- Extensions: ` `.jpg` , ` `.jpeg` , ` `.png` , ` `.gif` , ` `.svg` , etc.
- Description: Image files are used for adding visual content to web pages. Each format (JPEG, PNG, GIF, SVG) has its strengths (e.g., JPEG for photos, PNG for transparency, SVG for scalable vector graphics).

#### 5. Fonts

- Extensions: ` `.ttf` , ` `.otf` , ` `.woff` , ` `.woff2` , etc.
- Description: Font files are used to define typography on web pages. Web fonts like WOFF and WOFF2 are optimized for web use, providing cross-browser compatibility and faster load times.

#### 6. Video

- Extensions: ` `.mp4` , ` `.webm` , ` `.ogg`
- Description: Video files can be embedded in web pages to provide multimedia content. Formats like MP4, WebM, and OGG are widely supported across browsers.

#### 7. Audio

- Extensions: ` `.mp3` , ` `.ogg` , ` `.wav`
- Description: Audio files are used for embedding sound content on web pages. MP3, OGG, and WAV are common formats for web-based audio.

#### 8. Markup Formats

- Extensions: ` `.xml` , ` `.json`
- Description: XML (eXtensible Markup Language) and JSON (JavaScript Object Notation) are used for structured data interchange between web servers and clients, often in APIs and data handling.

### 9. Document Files

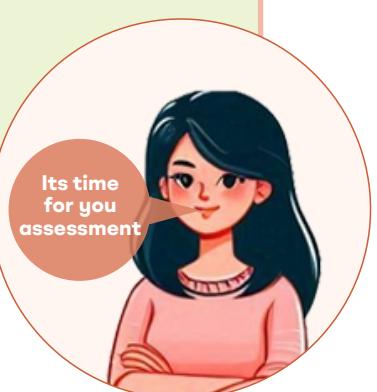
- Extensions: `pdf`, `docx`, `xlsx`, `pptx`, etc.
- Description: Document files are often linked or downloaded from web pages. PDF for documents, DOCX/XLSX/PPTX for Microsoft Office files.

### 10. Archive and Compression

- Extensions: `zip`, `tar.gz`, `rar`
- Description: Archive and compression formats are used to bundle multiple files and reduce file size for faster downloads. They are commonly used for distributing resources or backups.

## PRACTICAL EXERCISE

- Create a simple webpage that integrates HTML, CSS, and JavaScript files
- Understand the differences between image file formats by using them in a webpage. **Task:** Collect the same image in different formats: .jpg, .png, .gif, and .svg. Create an HTML file and embed each image format. Write a brief comparison in the HTML file about the file sizes and quality differences.
- Embed and control audio and video files on a webpage. **Task:** Create an HTML file and embed audio files (.mp3, .ogg, .wav) and video files (.mp4, .webm, .ogg). Add controls to play, pause, and adjust the volume. Write a short description of the compatibility and quality of each file format in the HTML file.



### Additional Resources:



[https://www.w3schools.com/html/html\\_images.asp](https://www.w3schools.com/html/html_images.asp)

#### - Video Links:

<https://www.youtube.com/watch?v=emLHW0VhzeM>

[https://www.youtube.com/watch?v=\\_w6N\\_nplmAw](https://www.youtube.com/watch?v=_w6N_nplmAw)

<https://www.youtube.com/watch?v=QP3PVpvDnks>

## How to Design a Simple Website

### How to produce Web Design Wireframes

Creating web design wireframes is an essential step in the web design process. Wireframes are basic visual guides that represent the skeletal framework of a website.

#### Guides to producing web design wireframes:

##### Step 1: Define Objectives and Requirements

- Identify the Purpose: Determine the main goals of the website.
- Gather Requirements: Collect information about the target audience, features, content, and functionality needed.

##### Step 2: Research and Inspiration

- Analyze Competitors: Look at competitor websites for inspiration and understand what works well in your industry.
- Gather Ideas: Note down ideas and features you want to incorporate into your design.

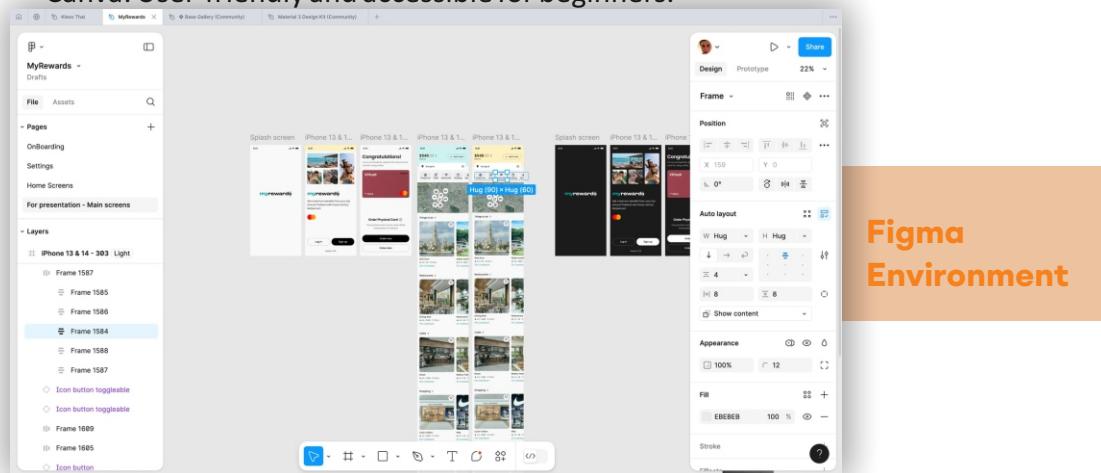
##### Step 3: Create a Sitemap

- Outline the Structure: Create a sitemap that outlines the main pages and subpages of the website. This helps in organizing the content and defining the navigation structure.

##### Step 4: Choose a Wireframing Tool

Select a Tool: Choose a wireframing tool that suits your needs. Some popular options include:

- Sketch: Great for macOS users, offers extensive plugins.
- Adobe XD: Powerful design and prototyping tool.
- Figma: Excellent for collaborative design with real-time updates.
- Balsamiq: Focuses on low-fidelity wireframes with a hand-drawn feel.
- Canva: User-friendly and accessible for beginners.



### Step 5: Start with Basic Sketches

- Draw Layouts: Create simple sketches on paper or a whiteboard to outline the layout of each page.
- Focus on Structure: Concentrate on the placement of key elements like headers, footers, navigation menus, content areas, and buttons.



### Step 6: Add Detail to Your Wireframes

- Include Navigation: Add a navigation bar with links to the main sections of your website.
- Create Multiple Pages: Design wireframes for each major page or section of the website.
- Use Placeholders: Add placeholders for images, videos, and other media content.

### Step 7: Incorporate Interaction Elements

- Buttons and Links: Add buttons, links, and call-to-action elements.
- Form Fields: Include form fields for contact forms, search bars, and other interactive elements.

### Step 9: Review and Refine

- Review for Consistency: Ensure that design elements are consistent across all pages.
- Gather Feedback: Share your wireframes with stakeholders, team members, or potential users and gather feedback.
- Make Improvements: Refine the wireframes based on feedback and ensure they align with the project objectives.

### How to produce a Web Page Flow Diagram

Producing a web page flow diagram involves creating a visual representation of the navigation and structure of a website.

#### Guides:

1. Define the Scope: Identify the main pages and sections that will be included in the website.
2. Start with Main Pages: List the main pages of the website (e.g., Home, About Us, Services, Contact Us).
3. Outline Subpages: For each main page, outline any subpages or sections (e.g., Services may have subpages like Web Design, Development, SEO).
4. Map Navigation: Draw arrows to show how users navigate between pages (e.g., from Home to About Us, from Services to Web Design).
5. Include Interactions: If needed, indicate interactions such as dropdown menus, links to external pages, or forms.
6. Review and Refine: Share the diagram with stakeholders for feedback and make adjustments to ensure clarity and completeness.
7. Document and Share: Once finalized, document the web page flow diagram and share it with the development team and stakeholders for reference.

### Design a Web Page

Page Design Principles: Applying design principles such as visual hierarchy, typography, and color theory to create aesthetically pleasing and functional web pages.

A simple website about Girls' Education using Bootstrap:

#### Step 1: Set Up Your Project

1. Create a project folder: GirlsEducation
2. Create the necessary files and folders:
  - o index.html
  - o css/style.css
  - o img/ (for images)

#### Step 2: Include Bootstrap

Add the Bootstrap CSS and JS links in the <head> and before the closing <body> tag respectively in your index.html file.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Girls Education</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
</head>
<body>

  <!-- Your content will go here -->

  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.2/dist/umd/popper.min.js"><scri
pt>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```



### Step 3: Create the Navbar

Add a navigation bar using Bootstrap classes.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Girls Education</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home</a>
      </li>
    </ul>
  </div>
</nav>
```



```
<li class="nav-item">
    <a class="nav-link" href="#">About</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="#">Programs</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="#">Contact</a>
</li>
</ul>
</div>
</nav>
```



### Step 4: Add a Jumbotron

Use a Jumbotron for the hero section.

```
<div class="jumbotron text-center">
    <h1 class="display-4">Empowering Girls Through Education</h1>
    <p class="lead">Unlocking potential and creating opportunities for a brighter future.</p>
    <a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a>
</div>
```



### Step 5: Create the About Section

Add an about section using Bootstrap grid.

```
<div class="container mt-5">
    <div class="row">
        <div class="col-md-6">
            <h2>About Us</h2>
            <p>Our mission is to provide quality education to girls worldwide, enabling them to achieve their full potential. We believe education is the key to a brighter future.</p>
        </div>
    </div>
```



```
<div class="col-md-6">
    
</div>
</div>
</div>
```



### Step 6: Add Programs Section

List the programs offered.

```
<div class="container mt-5">
    <h2 class="text-center">Our Programs</h2>
    <div class="row">
        <div class="col-md-4">
            <div class="card">
                
                <div class="card-body">
                    <h5 class="card-title">STEM Education</h5>
                    <p class="card-text">Encouraging girls to pursue studies in Science, Technology, Engineering, and Mathematics.</p>
                    <a href="#" class="btn btn-primary">Learn More</a>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="card">
                
                <div class="card-body">
                    <h5 class="card-title">Scholarships</h5>
                    <p class="card-text">Providing financial assistance to girls in need to help them continue their education.</p>
                    <a href="#" class="btn btn-primary">Learn More</a>
                </div>
            </div>
        </div>
    </div>
</div>
```



```
</div>
<div class="col-md-4">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Mentorship</h5>
      <p class="card-text">Connecting girls with mentors to guide them in their academic and professional journeys.</p>
      <a href="#" class="btn btn-primary">Learn More</a>
    </div>
  </div>
</div>
</div>
```



### Step 7: Add Contact Section

Provide a contact form.

```
<div class="container mt-5 mb-5">
  <h2 class="text-center">Contact Us</h2>
  <div class="row">
    <div class="col-md-8 offset-md-2">
      <form>
        <div class="form-group">
          <label for="name">Name</label>
          <input type="text" class="form-control" id="name" placeholder="Your Name">
        </div>
        <div class="form-group">
          <label for="email">Email</label>
          <input type="email" class="form-control" id="email" placeholder="Your Email">
        </div>
        <div class="form-group">
```

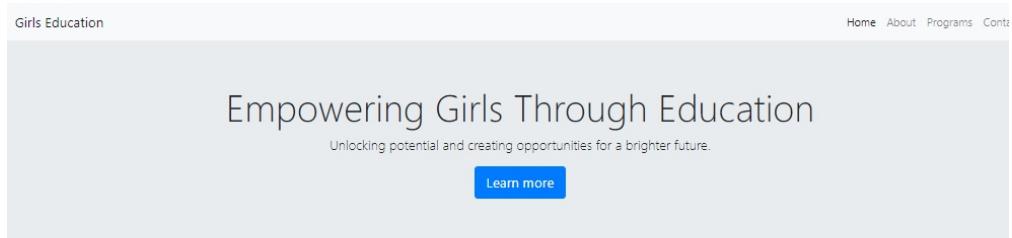


```
<label for="message">Message</label>
<textarea class="form-control" id="message" rows="4" placeholder="Your
Message"></textarea>
</div>
<button type="submit" class="btn btn-primary btn-block">Submit</button>
</form>
</div>
</div>
</div>
```

## Step 8: Add Footer

Add a simple footer.

```
<footer class="bg-light text-center py-4">
<p>© 2024 Girls Education. All Rights Reserved.</p>
</footer>
```



## About Us

Our mission is to provide quality education to girls worldwide, enabling them to achieve their full potential. We believe education is the key to a brighter future.



## Our Programs



### STEM Education

Encouraging girls to pursue studies in Science, Technology, Engineering, and Mathematics.

[Learn More](#)



### Scholarships

Providing financial assistance to girls in need to help them continue their education.

[Learn More](#)



### Mentorship

Connecting girls with mentors to guide them in their academic and professional journeys.

[Learn More](#)

## Contact Us

Name

Your Name

Email

Your Email

Message

Your Message

Submit

© 2024 Girls Education. All Rights Reserved.

### How to Build a Website Prototype

Building a website prototype involves creating a preliminary model of the website to visualize and test its design and functionality before the final development stage.

#### Step 1: Define Objectives and Requirements

- Identify the Purpose: Determine the main goals of the website.
- Gather Requirements: Collect information about the target audience, features, content, and functionality needed.

#### Step 2: Create a Sitemap

- Outline the Structure: Create a sitemap that outlines the main pages and subpages of the website. This helps in organizing the content and defining the navigation structure.

#### Step 3: Sketch Wireframes

- Draw Layouts: Create simple sketches or digital wireframes to outline the layout of each page.
- Focus on Structure: Focus on the placement of key elements like headers, footers, navigation menus, content areas, and buttons without worrying about design details.

#### Step 4: Use Prototyping Tools

- Choose a prototyping tool to create interactive wireframes. Here are some popular tools:
- Figma: Great for collaborative design with real-time updates.
- Adobe XD: Provides powerful design and prototyping features.
- Sketch: Preferred for macOS users with extensive plugins and integrations.
- InVision: Focuses on collaboration and feedback with interactive prototypes.

#### Step 5: Build Prototypes

- Create Basic Layouts: Using your wireframes, build low-fidelity prototypes in the chosen tool.
- Add Navigation: Link the wireframes to simulate basic navigation between pages.
- Focus on User Flow: Ensure the user can navigate through the site as intended.

#### Step 6: Add Interactivity

- Incorporate Clickable Elements: Make buttons, links, and interactive elements clickable.
- Simulate User Actions: Add interactions like hover effects, pop-ups, and form submissions to test functionality.

### Step 7: Collect Feedback

- Share the Prototype: Share the interactive prototype with stakeholders, team members, and potential users.
- Gather Input: Collect feedback on usability, design, and functionality to identify areas for improvement.

### Step 8: Iterate and Refine

- Make Improvements: Based on feedback, refine the prototype to address any issues or suggestions.
- Repeat Testing: Continue testing and gathering feedback until the prototype meets the desired requirements.

### Step 9: Create High-Fidelity Prototypes

- Design Details: Add detailed design elements, such as colors, fonts, images, and content.
- Enhance Interactions: Improve the interactions to closely mimic the final product.

### Step 10: Final Review and Approval

- Present the Prototype: Present the high-fidelity prototype to stakeholders for final approval.
- Prepare for Development: Once approved, prepare the prototype and assets for handoff to the development team.

### Example Using Figma

#### 1. Define Objectives and Requirements:

Objective: Create a website prototype for a Girls' Education platform.

Requirements: Home page, About Us, Programs, Contact Us, responsive design.

#### 2. Create a Sitemap:

- Home
- About Us
- Programs
  - STEM Education
  - Scholarships
  - Mentorship
- Contact Us



**3. Sketch Wireframes:**

- Use paper or a digital tool to sketch the layout of the Home, About Us, Programs, and Contact Us pages.

**4. Set Up Figma Project:**

- Create a new project in Figma and start with blank frames for each page.

**5. Build Low-Fidelity Prototypes:**

- Design the basic layout of each page using rectangles and text blocks to represent headers, content areas, and buttons.
- Link the frames to simulate navigation.

**6. Add Interactivity:**

- Make buttons clickable to navigate between pages.
- Add hover states for interactive elements.

**7. Collect Feedback:**

- Share the Figma link with stakeholders for feedback.

**8. Iterate and Refine:**

- Update the prototype based on feedback.
- Repeat the testing process.

**9. Create Prototypes:**

- Add colors, fonts, and images.
- Enhance interactions to reflect the final design.

**10. Final Review and Approval:**

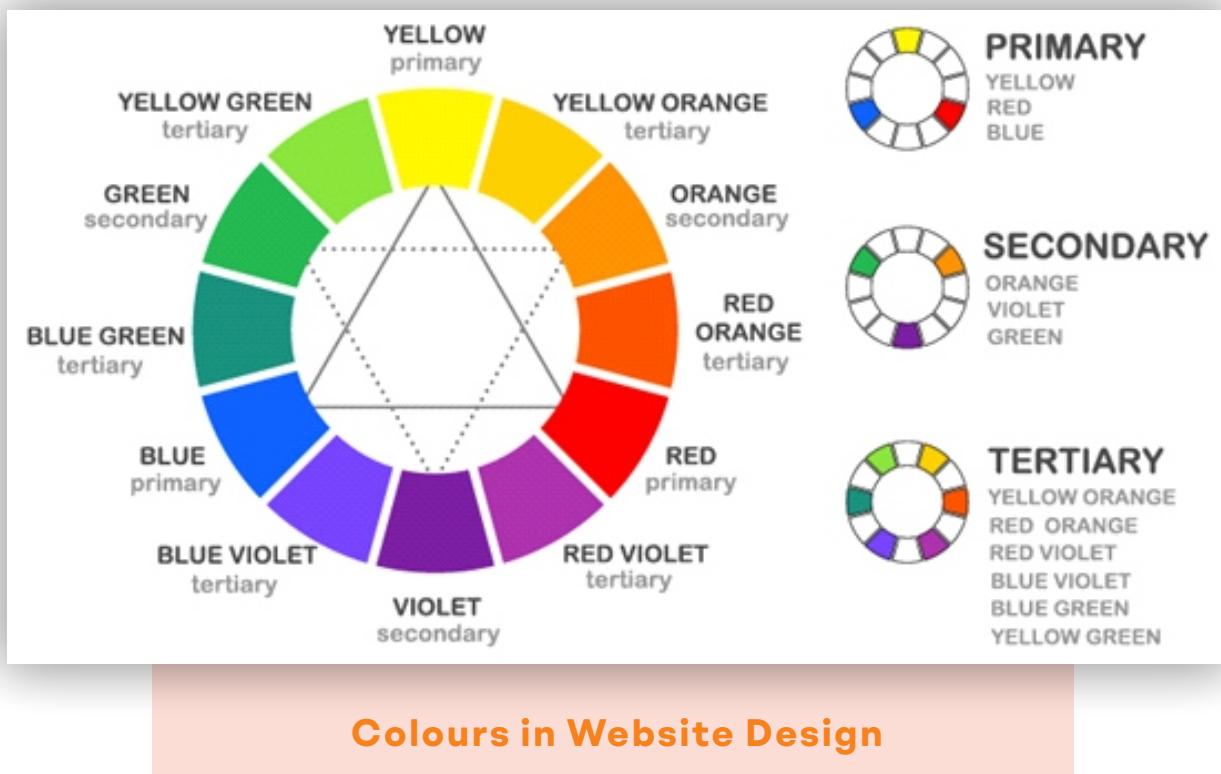
- Present the refined prototype to stakeholders.
- Prepare the design assets for the development team.

### The Importance of Themes and Color Coding in Web Design

Themes and color coding play crucial roles in web design, influencing user experience, branding, and overall aesthetics.

**Here is why they are important:**

- Visual Consistency and Branding: Themes and color schemes establish a visual identity for the website that aligns with the brand. Consistent use of colors and themes across pages helps users recognize and remember the brand easily. This consistency builds trust and reinforces brand recognition.
- User Experience and Navigation: Color coding can aid navigation by highlighting important elements such as links, buttons, and navigation menus. For example, using a distinct color for clickable links or active navigation tabs guides users through the website and enhances usability. Clear visual cues reduce confusion and improve the overall user experience.
- Aesthetic Appeal and User Engagement: A well-chosen theme and color scheme enhance the overall aesthetic appeal of the website. A visually appealing design can captivate users, encourage longer visits, and increase the likelihood of return visits. It reflects professionalism and attention to detail, contributing to a positive perception of the brand.



**Additional Resources:**



- Video Links:

- [https://www.youtube.com/watch?v=D4NyQ5iOMF0&list=PLKId0A0XCIbUYx3c\\_NYn13W9Z\\_kkIiA2m](https://www.youtube.com/watch?v=D4NyQ5iOMF0&list=PLKId0A0XCIbUYx3c_NYn13W9Z_kkIiA2m)
- <https://www.youtube.com/watch?v=jk1T0CdLxwU&t=97s>
- <https://www.youtube.com/watch?v=cCNLD5IZY34&t=530s>
- <https://www.youtube.com/watch?v=pijzYKAOluw>

### The UI/UX in Web Design

#### What is User Interface (UI)

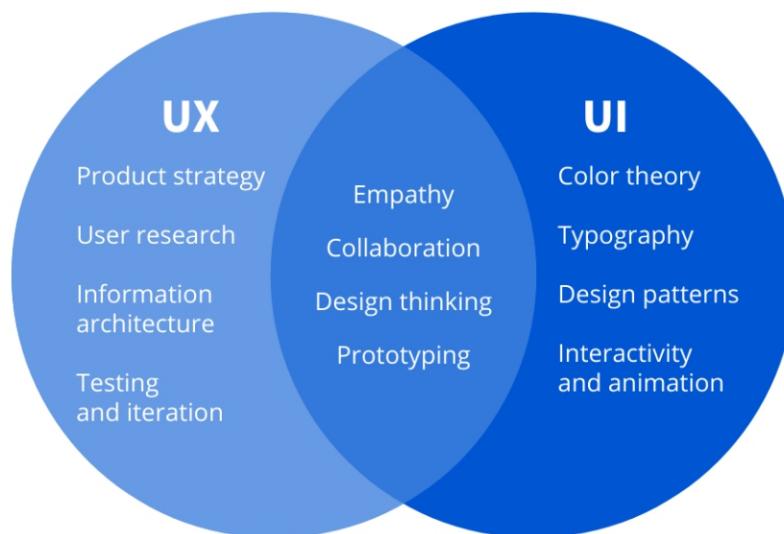
UI is the Identifying and designing of interactive elements and visual components that facilitate user interaction and navigation within a website.

#### What is User Experience (UX)

UX is evaluating and optimizing the overall user journey and interaction flow to ensure a seamless and engaging experience across all touchpoints.

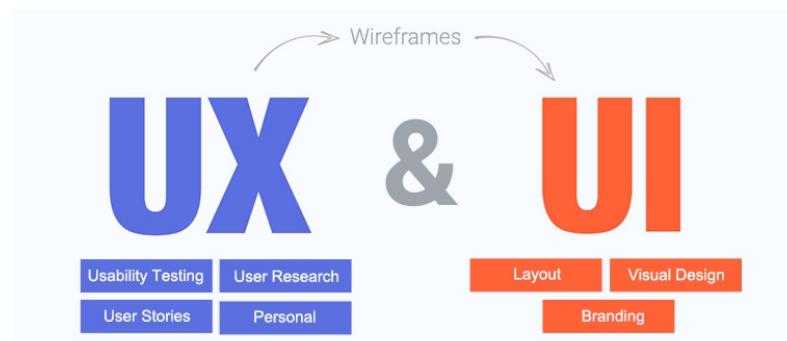
#### Difference between UI and UX

- UI vs. UX Comparison: Recognizing the distinct roles of UI (visual design and layout) and UX (user journey and experience optimization) in web design.



#### The Importance of UI/UX in Web Design

- UI/UX Significance: Emphasizing the critical role of intuitive interface design and user-centric experience in maximizing usability, satisfaction, and conversion rates.



## PRACTICAL EXERCISE

- Design a user interface prototype for a web application.
- Conduct usability testing and gather feedback to enhance user experience design.



### Additional Resources:



#### Video Links:

<https://www.youtube.com/watch?v=55NvZjUZIO8>



*Now, can you confidently say repeat these statements to yourself:*

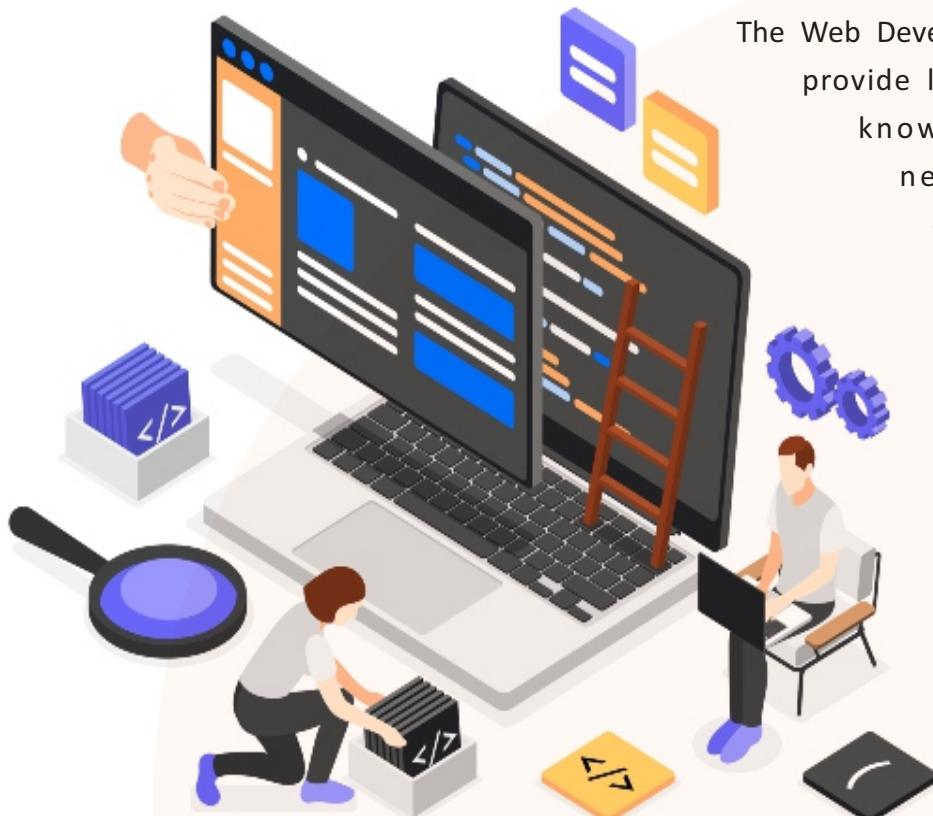
*I can:*

- Write HTML for a website
- Use JavaScript to make website dynamic
- Implement Cascading Style Sheets (CSS) in a website
- Use Content Delivery Network (CDN).

**CLICK! FOR YOUR NEXT MODULE**



# DESIGNING A WEBSITE



The Web Development Environment aims to provide learners with comprehensive knowledge and practical skills necessary for effective web development. The unit will discuss Content Management Systems (CMS), where learners will know various types of CMS, understand the criteria for selecting a CMS, and learn to install and utilize these systems or frameworks for web applications.

### At the end of this Module, learners should be able to:

- List and understand different types of CMS.
- Install and configure CMS for web application development.
- Select appropriate CMS based on specific criteria.
- Choose relevant database technologies for web applications.
- Compare CMS frameworks with custom web frameworks.
- Understand what a web development framework is.
- List different types of web development frameworks with examples.
- Recognize the importance of frameworks in web design/development.
- Compare CMS frameworks with custom web frameworks.
- Install and configure a web development framework (WordPress)



### Content Management System (CMS)

A Content Management System (CMS) is software that enables users to create, manage, and modify content on a website without specialized technical knowledge.

#### Types of Content Management Systems (CMS)

##### 1. Open Source CMS:

- Examples: WordPress, Joomla, Drupal.
- Characteristics: Free to use, community-supported, highly customizable.

##### 2. Proprietary CMS:

- Examples: Adobe Experience Manager, Sitecore.
- Characteristics: Requires a license, offers professional support, usually more secure.

##### 3. Cloud-Based CMS:

- Examples: Wix, Squarespace, Shopify.
- Characteristics: Hosted on the cloud, easy to set up, often includes hosting and domain services.

##### 4. Headless CMS:

- Examples: Strapi, Contentful.
- Characteristics: Focuses on managing content. decoupled front-end. API-driven.



#### Web Development Framework Using CMS: A Focus on WordPress and Database

Web development frameworks and Content Management Systems (CMS) have revolutionized the way websites are created and managed. This learning material focuses on using WordPress, a popular CMS, along with database management, to create and manage websites efficiently.

### Web Development Framework

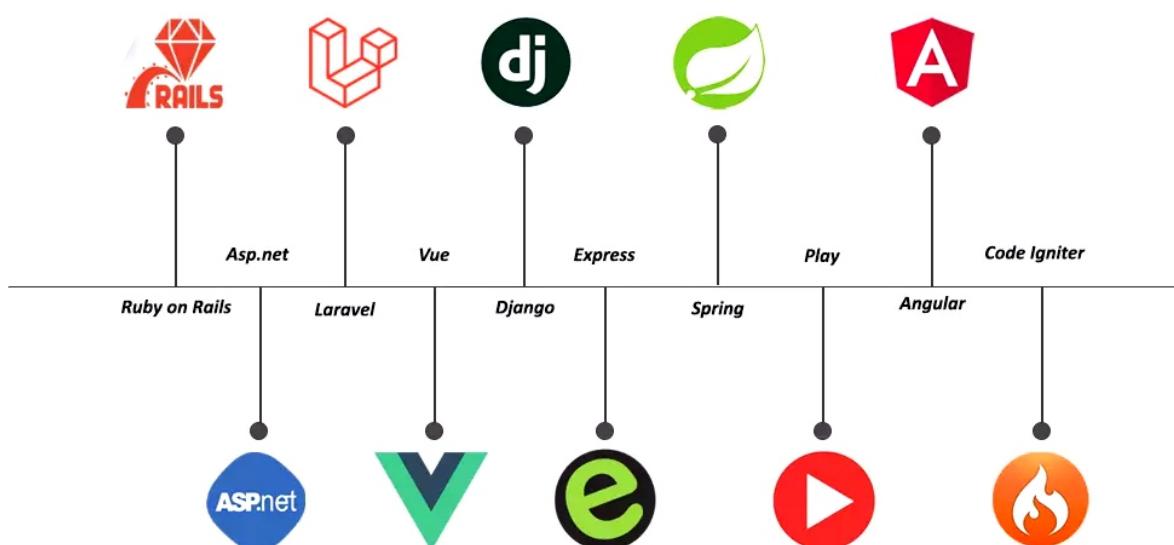
A web development framework is a collection of tools, libraries, and best practices that provide a structured and efficient way to build web applications. These frameworks streamline the development process by offering reusable code components and functionalities, enabling developers to focus on writing custom features.

#### Benefits of Using Frameworks:

- Consistency: Provides a standard way of developing applications.
- Efficiency: Reduces development time with pre-built modules.
- Maintainability: Easier to maintain and update code.
- Security: Often includes built-in security features.

#### Examples of Popular Frameworks:

- Frontend: Angular, React, Vue.js
- Backend: Django, Ruby on Rails, Laravel



### What is a CMS?

A Content Management System (CMS) is software that allows users to create, manage, and modify digital content without requiring specialized technical knowledge. CMS platforms are designed to simplify the management of websites and digital content.

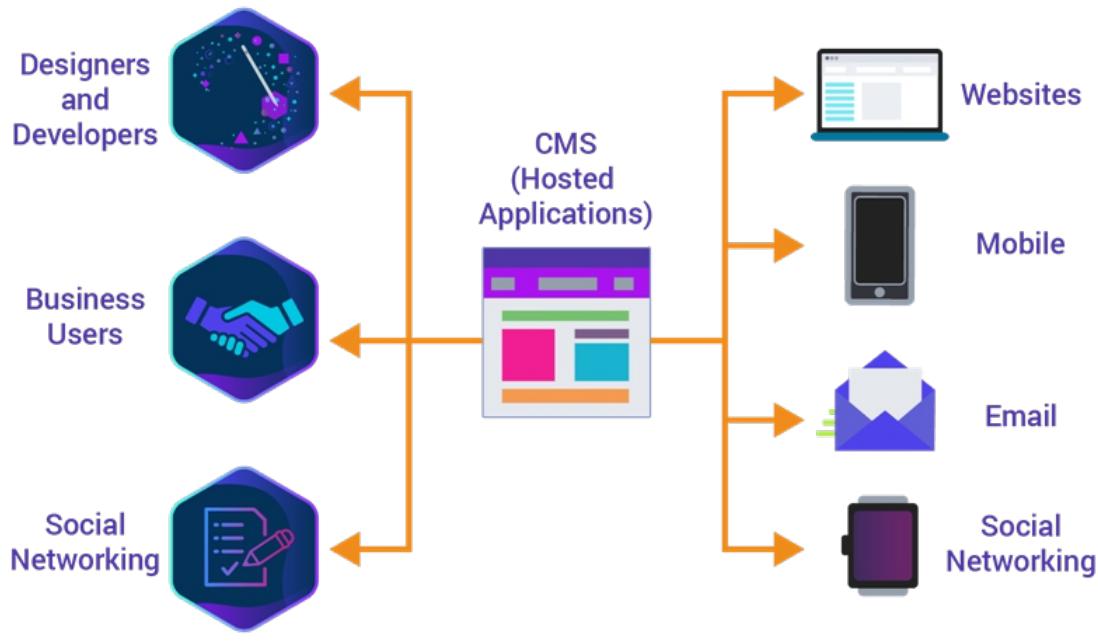
### How CMS Differs from Traditional Web Development:

- User-Friendly: No need for coding skills to create and manage content.
- Dynamic Content Management: Easily update and manage content through an admin interface.
- Extensibility: Plugins and themes extend functionality without altering the core codebase.

### Some Popular CMS Platforms:

- **WordPress**: Most widely used, flexible, and user-friendly.
- **Joomla**: Offers more complexity and flexibility than WordPress.
- **Drupal**: Highly customizable and suited for complex websites.

## What's a CMS?



### WordPress

WordPress started as a simple blogging platform in 2003 and has evolved into a full-fledged CMS capable of handling complex websites.

#### **Advantages of Using WordPress:**

- Ease of Use: Intuitive interface that's easy for beginners.
- Customization: Thousands of themes and plugins.
- Community Support: Large community for support and development.
- SEO-Friendly: Built-in SEO features and plugins.

#### **Use Cases and Examples of Websites Built with WordPress:**

- Personal blogs
- Corporate websites
- E-commerce sites (with WooCommerce)
- Portfolios and Galleries

### WordPress.org vs WordPress.com

WordPress.org: Self-hosted, full control over customization, requires domain and hosting.

WordPress.com: Hosted solution, limited customization, managed by WordPress.

### WordPress Features and Benefits

#### **Customizability and Flexibility**

WordPress allows users to create unique websites through customizable themes and plugins.

#### **Themes and Plugins:**

- Themes: Control the appearance and layout of the website.
- Plugins: Extend functionality, such as SEO, security, and e-commerce.

#### **User-Friendly Interface**

The WordPress dashboard is designed for ease of use, even for non-technical users.

#### **SEO-Friendly**

WordPress is designed with SEO in mind, featuring clean code, customizable permalinks, and SEO plugins like Yoast SEO.

### What are the terminologies in WordPress

#### Posts vs Pages:

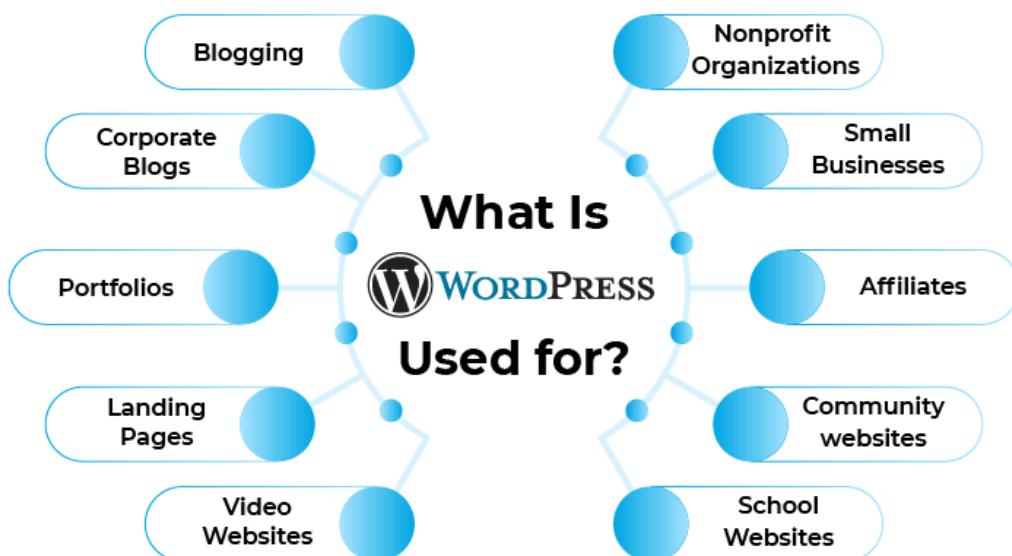
- Posts: Time-stamped content for blogs.
- Pages: Static content like "About Us" and "Contact."

#### Categories vs Tags:

- Categories: Broader topics than group-related posts.
- Tags: Specific keywords describing the post's details.

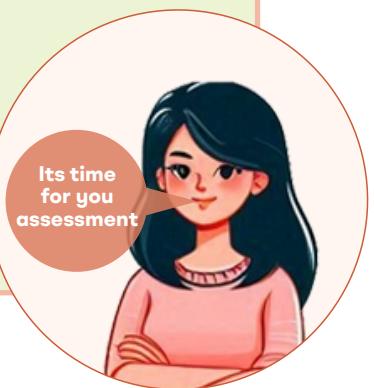
#### Themes and Plugins:

- Themes: Visual and structural template of a site.
- Plugins: Add-on features and functionalities.



## PRACTICAL EXCERCISE

1. Search online for different uses of CMS.
2. Understand CMS and CMS Frameworks.
3. Watch and Read from different tutors about the use of CMS and Wordpress
4. Configure the basic settings and explore the dashboard.



### Additional Resources:



- Video Links:

<https://www.youtube.com/watch?v=deqX0gMeUVc>

# Setting Up a Local Development Environment

## What is a Local Server?

A local server is a server environment installed on your local machine, allowing you to develop and test websites offline.

### Benefits of Using a Local Server for Development:

- Speed: Local development is faster without internet delays.
- Security: Safe testing environment without affecting the live site.
- Cost-Effective: No need for hosting costs during development.

## Tools for Setting Up a Local Server

### XAMPP:

Cross-platform, easy to install, includes Apache, MySQL, PHP, and Perl



### MAMP:

Similar to XAMPP but specifically for macOS.



### WAMP:

Windows-based, includes Apache, MySQL, and PHP.



### Local by Flywheel:

User-friendly interface, designed specifically for WordPress development.



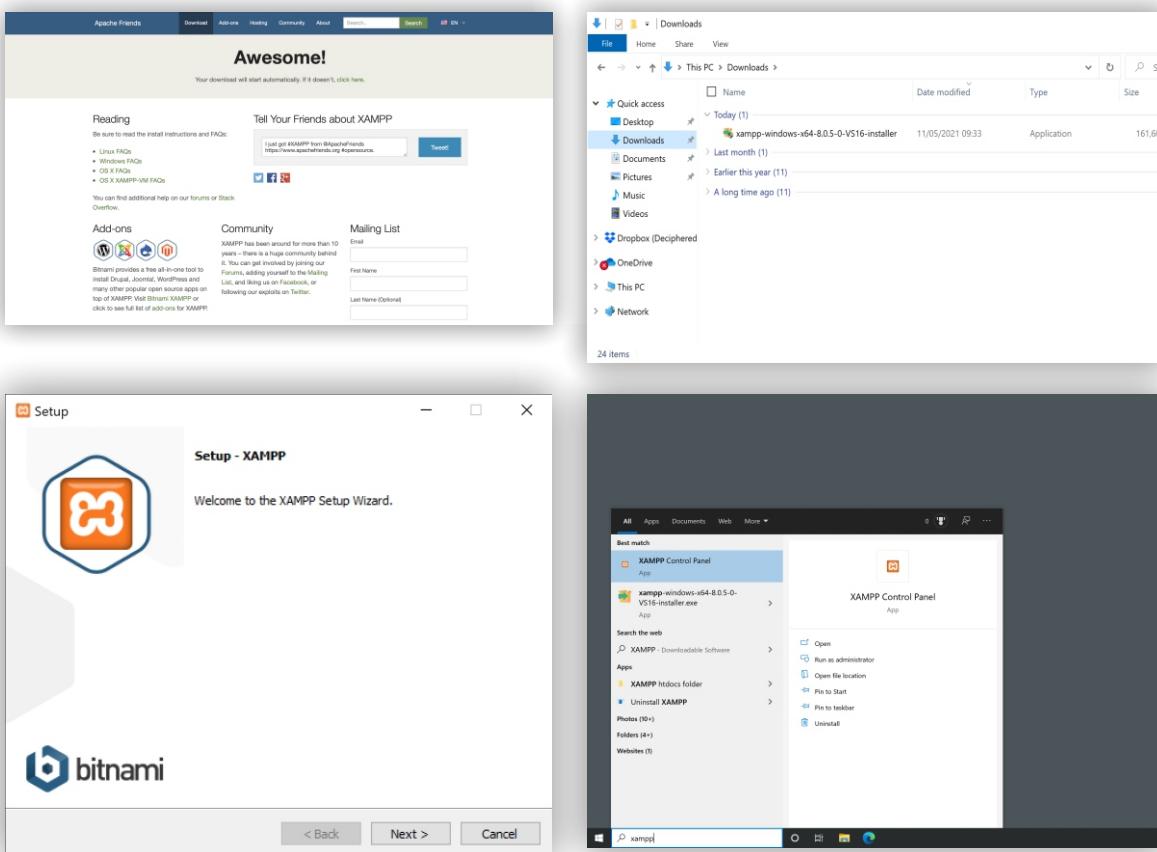
## Installing XAMPP

### Downloading and Installing XAMPP

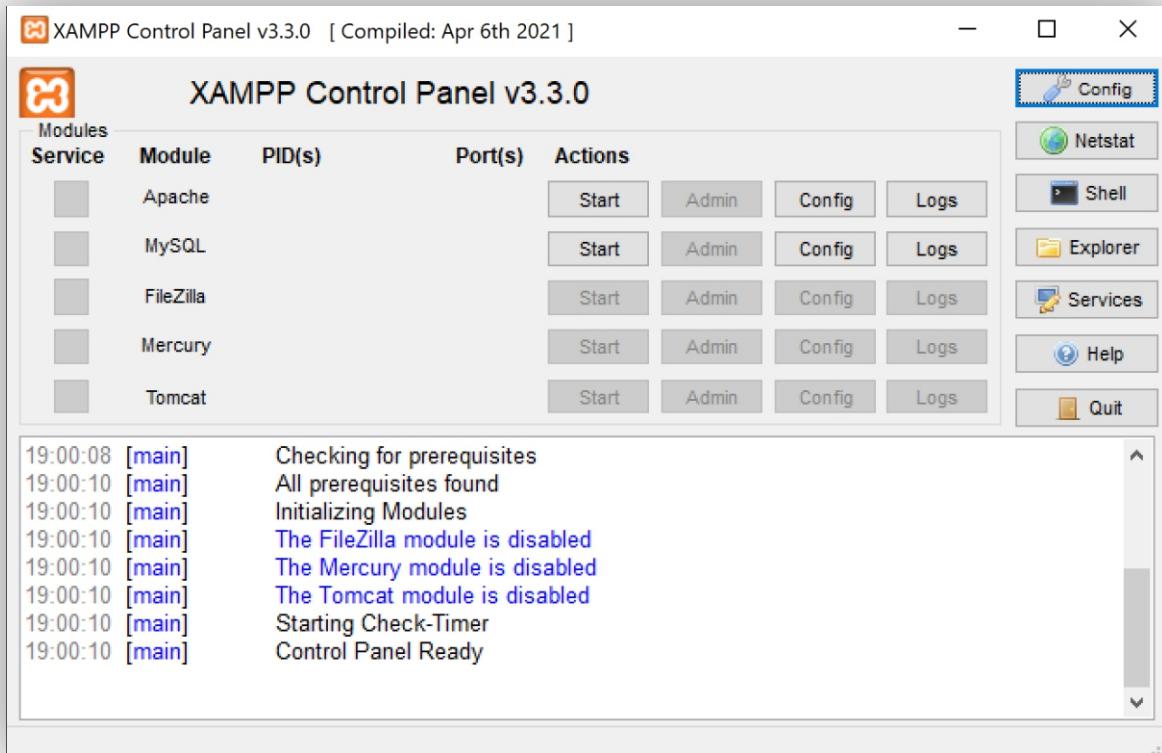
Visit the XAMPP [website](#) and download the installer and click on XAMPP for Windows to download.

A screenshot of the XAMPP website's download page. The page has a dark blue header with navigation links like 'Apache Friends', 'Download', 'Add-ons', 'Hosting', 'Community', 'About', 'Search...', and a language selector. The main content area features the XAMPP logo and the text 'XAMPP Apache + MariaDB + PHP + Perl'. Below this is a section titled 'What is XAMPP?' with a brief description. To the right is a large orange button with a play icon and the word 'XAMPP'. At the bottom are four download links: 'Download' (with a note 'Click here for other versions'), 'XAMPP for Windows 8.0.6 (PHP 8.0.6)', 'XAMPP for Linux 8.0.6 (PHP 8.0.6)', and 'XAMPP for OS X XAMPP-VM (PHP 8.0.6)'.

1. Wait for installation to complete
2. Once the download has been completed, go to your Downloads folder and double-click on the installer to start the installation.
3. Follow the instructions to install the XAMPP
4. Choose Language and press Next to continue
5. Wait for the Installation to complete
6. Once all the components are unpacked and installed, you can close the setup wizard by clicking on the 'finish' button.
7. To start XAMPP, type XAMPP into the search bar. Then select the XAMPP Control Panel to start XAMPP.



## The Xampp Environment



1. Here you can see the XAMPP interface and actions you can perform
2. The **Start** button for Apache is used to start the Apache web server and then stop it when no longer needed.
3. The **Start** button for MySQL will start the MySQL database and then stop it when it is no longer needed.
4. The port numbers for Apache and MySQL are shown when the servers have started and running.
5. Click on your Apache server's Admin button to go to your web server's web address. The cPanel will now start in your standard browser, and you will be led to the dashboard of your XAMPP's localhost. Click on phpMyAdmin at the right top corner to open database environment.
6. You can create and handle the database of your application

The left screenshot shows the XAMPP Control Panel welcome page. It includes a header with links for Apache Friends, Applications, FAQs, HOW TO Guides, PHPInfo, and phpMyAdmin. The main content area says 'Welcome to XAMPP for Windows 8.1.6'. It provides information about the installation and links to various resources. The right screenshot shows the phpMyAdmin interface. It has a sidebar with 'Recent' and 'Favorites' sections. The main area has tabs for 'Databases', 'User accounts', 'Export', 'Import', and 'Settings'. The 'Databases' tab is currently active. It contains sections for 'General settings' (with a dropdown for 'Server connection collation' set to 'utf8mb4\_unicode\_ci') and 'Appearance settings' (with dropdowns for 'Language' set to 'English' and 'Theme' set to 'pmahome').

### Configuring XAMPP Settings

- Ensure Apache and MySQL ports do not conflict with other applications.
- Adjust memory limits and execution times in php.ini if needed.

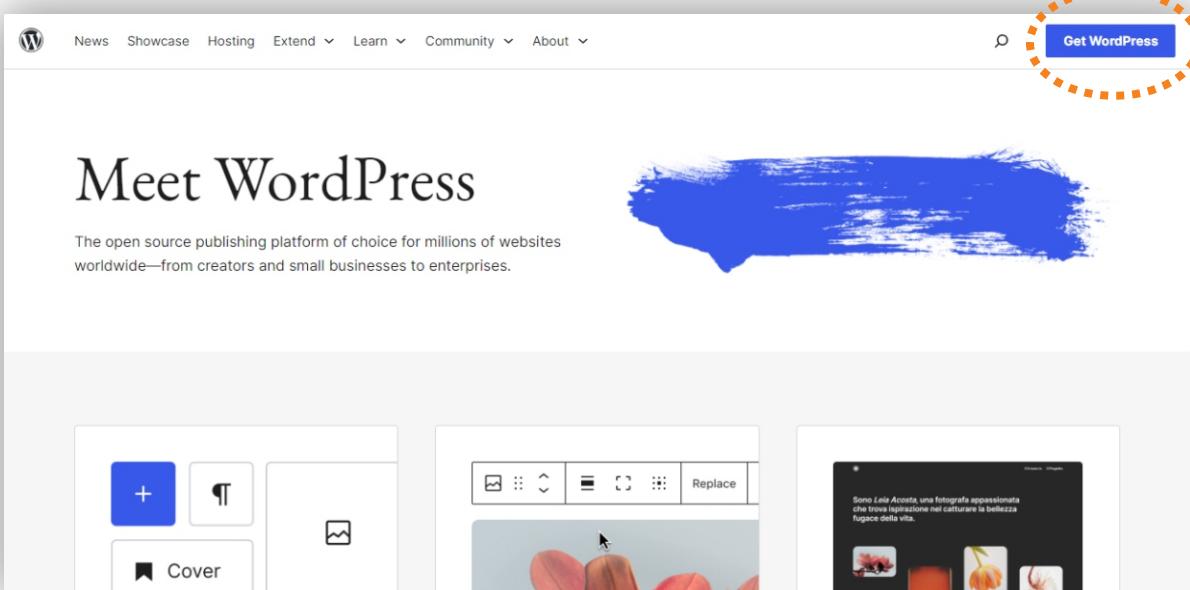
### Starting and Stopping the Server

- Open the XAMPP control panel.
- Click 'Start' next to Apache and MySQL.
- To stop, click 'Stop' next to Apache and MySQL.

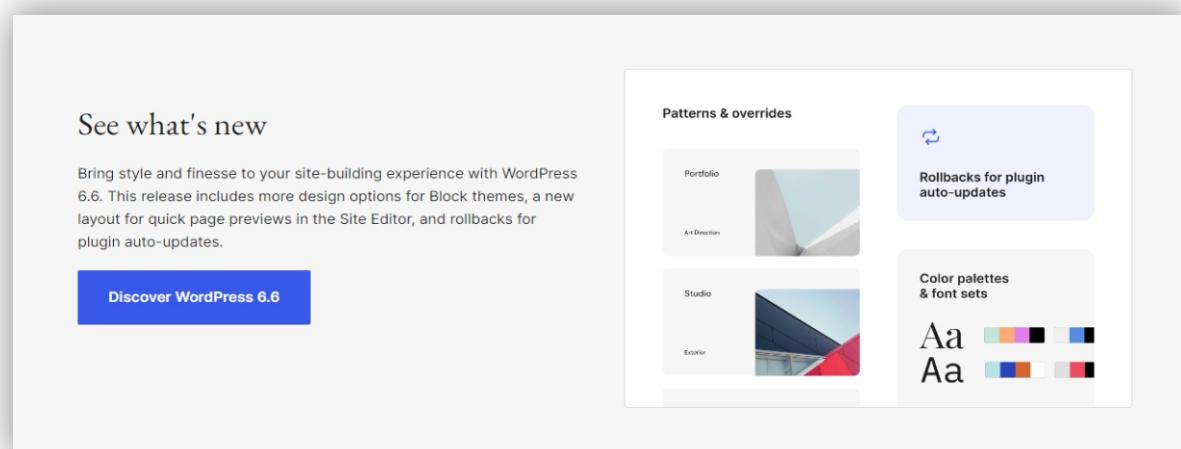
### Setting Up a Local WordPress Site

#### Downloading WordPress

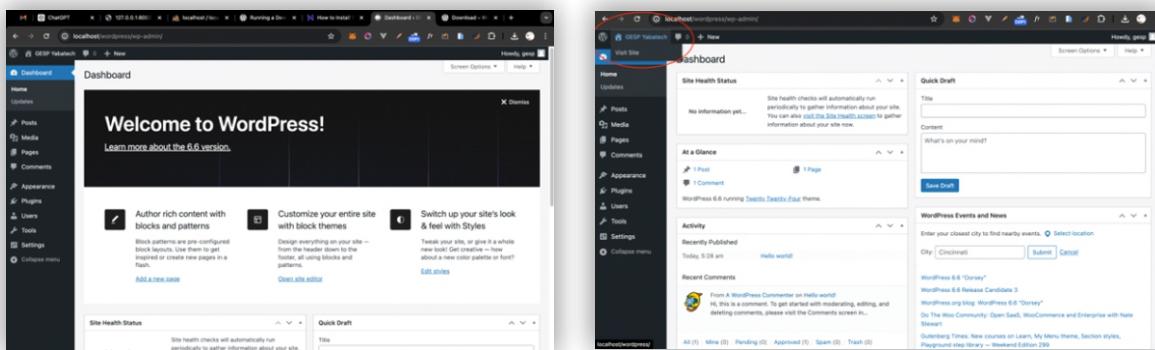
1. Go to the official WordPress at [wordpress.org](https://wordpress.org) and choose Get WordPress.



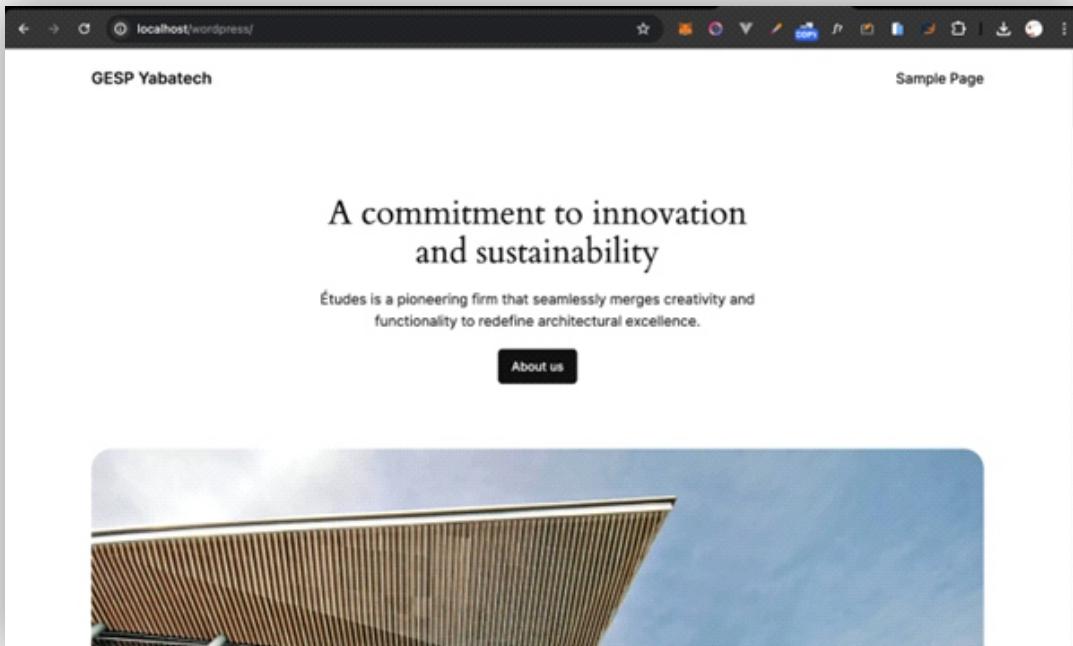
2. Download the latest version of WordPress.



3. Go to your file explorer and unzip the WordPress archive.
4. Move the unzipped WordPress folder to your Xampp htdocs in C:\Xampp\htdocs or C:\Xampp\www.
5. Go to your browser, type localhost/wordpress and press enter to launch the WordPress website and choose language, press continue.
6. Know the following requirements as instructed below, click on Let's Go to continue.
7. Enter the database information and the database name you created earlier, the username is usually root, and the password is root (or sometimes empty, depending on the Xampp configuration), click submit to continue.
8. If everything is correct, you will interface below. Click 'run the installation' to continue.
9. Provide the site information with your login details as administrator, and provide a strong password. Now click on "Install wordpress"
10. Congrats. Now click on the "Login link" to log in to the WordPress admin.
11. Enter your site login details to log in.
12. Dashboard to manage your site.
13. Visit your WordPress site by clicking on the site name at the left corner and then clicking on 'Visit site'



### 14. Welcome to your WordPress Website.



## PRACTICAL EXERCISE

Carry out the following Installation and Configuration with your instructor:

1. Install Xampp on your computer system.
2. Configure the basic settings and explore the localhost environment.
3. Download and Install WordPress on your computer system.
4. Configure the basic settings and explore WordPress Admin



### Additional Resources:



- Video Links:

<https://www.youtube.com/watch?v=SoKCB41tKW4>

<https://www.youtube.com/watch?v=71EZb94AS1k>

[https://www.youtube.com/results?search\\_query=how+to+install+wordpress](https://www.youtube.com/results?search_query=how+to+install+wordpress)

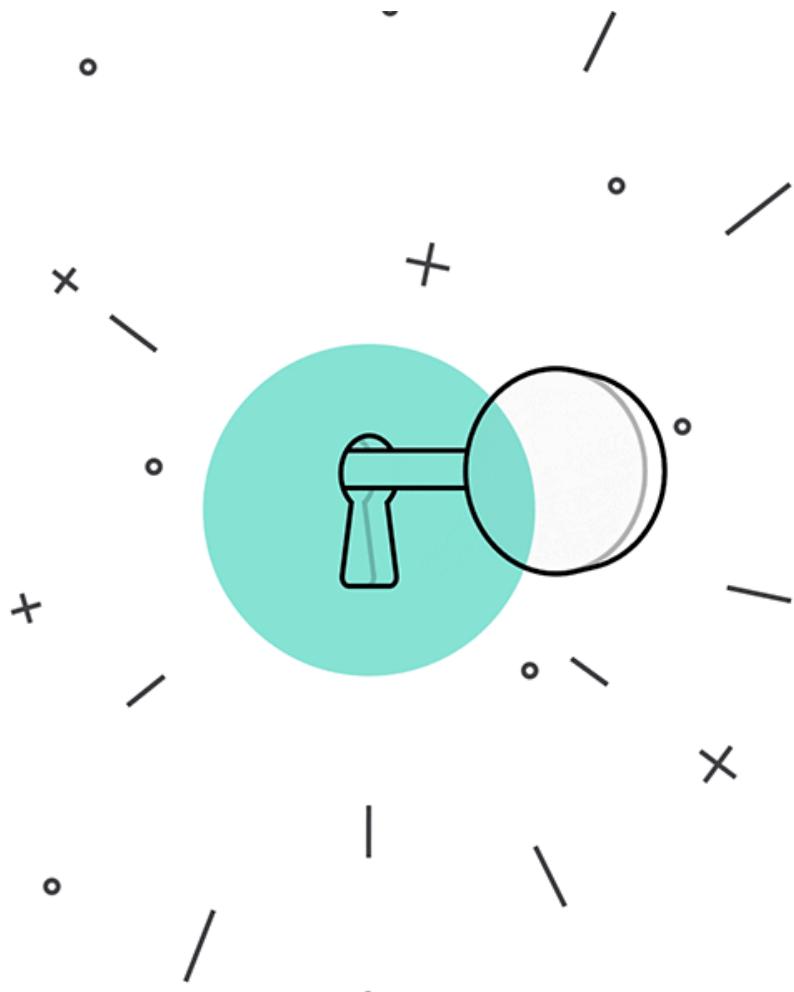


BUT,  
ONE MORE  
THING!

CLICK! TO VIEW WHAT YOU EARNED



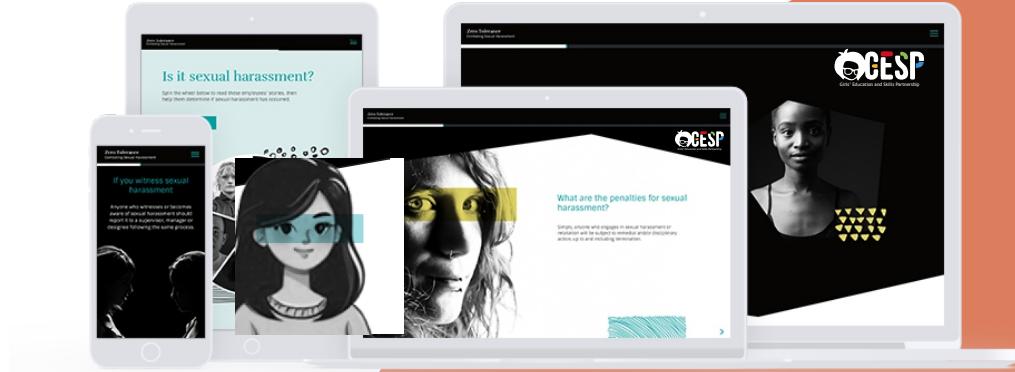




**YOU HAVE UNLOCKED  
END OF COURSE PROJECT**

# GESP WEB DEV PROJECTS

## GIRLS' EDUCATION ADVOCACY WEBSITE



1  
PROJECT

The focus of this project is to build a simple and informative website to promote girls' education, using HTML, CSS, and BOOTSTRAP.

The purpose of this website is to create awareness about the importance of girls' education. The site will feature different sections that highlight statistics, stories, and resources related to girls' education globally.



### KEY FEATURES



**Homepage:** Introduces the importance of girls' education with key statistics and a banner image.



**Success Stories:** Displays real-life stories of girls who have benefited from education programs.



**Contact Page:** Contains a contact form and links to social media platforms.



**About Section:** Provides an overview of the issue, with key facts and infographics.



**Resources Page:** Offers downloadable resources, links to educational programs, and government initiatives supporting girls' education.

### STEP-BY-STEP BREAKDOWN

#### 1. HTML Structure:

- Create a homepage (`index.html`) that includes sections for introduction, success stories, and resources.
- Use semantic HTML tags like ``, `

`, and `` to structure the content.
- Add a navigation bar at the top with links to different sections (About, Success Stories, Resources, Contact).

#### 2. CSS Design:

- Use CSS for a clean, professional design with a focus on readability.
- Define colors, fonts, and spacing that align with a global education theme (soft colors like blue or green).
- Style the navigation bar to be fixed at the top, with hover effects on the menu links.

#### 3. Responsive Design with BOOTSTRAP:

- Integrate Bootstrap to make the website responsive.
- Use Bootstrap's grid system to create a flexible layout.
- Add a Bootstrap carousel on the homepage to display key facts about girls' education.
- Use Bootstrap cards for the success stories section.

#### 4. Contact Form:

- Create a contact form with fields like name, email, and message using Bootstrap form controls.
- Style the form with CSS to match the overall theme.

#### 5. Design:

- Add animation on scroll using CSS or a Bootstrap library like `wow.js` for interactive design.

# SUBMIT

# GESP WEB DEV PROJECTS

PROJECT  
2

## SEXUAL HARASSMENT AWARENESS BLOG WITH WORDPRESS (HOSTED WEBSITE)



This project will involve setting up a WordPress blog dedicated to raising awareness about sexual harassment and providing support resources.

This WordPress site will feature informative blog posts, articles, and resources about sexual harassment. The goal is to educate users on the topic, provide legal resources, and offer support services for victims.



### KEY FEATURES



**Blog Section:** Regular posts that discuss the signs of harassment, how to report it, and personal stories.



**Resource Library:** Links to helplines, legal advice, and organizations that offer support.



**Interactive Forum:** Allows users to share their experiences and ask for advice in a safe environment.



**Newsletter Signup:** Encourages users to subscribe for updates on new posts and resources.

## STEP-BY-STEP BREAKDOWN

### 1. Install WordPress and Theme Setup:

- Set up a WordPress site using a user-friendly hosting platform.
- Choose a clean and responsive theme suited for a blog (e.g., Astra or OceanWP).
- Customize the theme to match the branding of the awareness campaign (soft colors, clear typography).

### 2. Content Structure:

- Create pages like "About Us," "Resources," and "Get Help Now."
- Set up the homepage to feature recent blog posts and a call to action encouraging readers to seek help or share their stories.

### 3. Plugins for Additional Functionality:

- Install the Elementor plugin to create custom page layouts for more control over design.
- Use the Contact Form 7 plugin to create a confidential form where users can report harassment anonymously.
- Add Yoast SEO to optimize blog posts for search engines, increasing visibility.

### 4. Blog Post Creation:

#### Write posts covering the following topics:

- What sexual harassment is and how to recognize it.
- Legal rights and steps to take after experiencing harassment.
- Stories from survivors (with consent) to create awareness and solidarity.
- Use categories and tags to organize posts (e.g., "Survivor Stories," "Legal Support," "Prevention Tips").

### 5. Resource Library:

- Create a dedicated page for resources, linking to organizations like local helplines, legal aid offices, and counseling services.
- Use a table or list format to make the information easy to access.

### 6. Interactive Forum (Optional):

- Install a forum plugin like bbPress to create a space where users can anonymously discuss their experiences, ask for advice, and support each other.
- Moderate the forum to ensure safety and confidentiality.

### 7. Include a downloadable PDF guide for sexual harassment prevention and reporting procedures.