UNIT 1 –INTRODUCTION TO WEB SITE DEVELOPMENT & JAVA SCRIPT

1. Web Site Development

- Web development is the act of building, creating and maintaining websites.
- The field encompasses a broad range of tasks including everything from coding, to technical design, to the performance of a website or application running on the internet.
- Web development consists of front-end and back-end components.

➢ **Front-end development** is responsible for the aspects of a website that users see and interact with: the user interface (UI).

- Front-end developers are well-versed in HTML, CSS and JavaScript, often working closely with design and UX teams to capture both the intended look and feel of the site, while also creating a quality user experience across multiple device types.

➢ **Back-end development** is responsible for all the aspects of a website that users do not see.

- This is also known as server-side development because back-end developers focus primarily on the behind-the-scenes logic, APIs and database interactions that power the site.

➢ **Full-stack development** is a more holistic approach where the developers responsible for the site or app take care of the entire development stack, from the inner workings typically performed on the back-end to the presentation layer normally handled by front-end developers.

2. Web Site Development Phases

- From creating a plan to launch, your team needs to perform many tasks in order to design and develop the website that will be able to attract customers.
- To ensure success, some companies tend to request assistance from IT staffing companies to augment the in-house team with a dedicated team of developers. Others try to create a website by their own efforts.
- Regardless of the choice, the whole development life cycle will take time and resources, so let's take a closer look at how to develop a website and what has to be done.

Step 1. Gathering Information: Purpose, Main Goals, and Target Audience

- Set Goals For The Website
- Define Sites Target Audience

Step 2. Planning: Sitemap and Wireframe Creation

- Create A Sitemap Sketch
- Create A Wireframe/Mock-Up
- Select Technology Stack

Step 3. Design: Page Layouts, Review, and Approval Cycle

- Create Colorful Page Layouts
- Review The Layouts
- Get Client's Feedback
- Change The Layout If Required

Step 4. Content Writing and Assembly

- Create New Content
- Get It Ready For Migration

Step 5. Coding

- Build And Deploy Website
- Add Special Features And Interactivity
- Use Seo For Your Website

Step 6. Testing, Review, and Launch

- Test The Created Website
- Upload The Website To Server
- Final Testing And Launch.

Step 7. Maintenance: Opinion Monitoring and Regular Updating

- Add User Report System
- Fix Bugs Asap
- Keep Website Up-To-Date

3. Web Site Authoring tools

- Web site authoring tools are software applications or online platforms that enable individuals to create and manage websites without extensive coding knowledge.
- These tools typically provide a user-friendly interface, templates, and features that simplify the website development process.
- Here are some popular web site authoring tools:

1. WordPress:

- WordPress is a widely-used content management system (CMS) that powers a significant portion of the web. It offers a user-friendly interface, a plethora of themes and plugins, and supports both beginners and advanced users.

2. Wix:

- Wix is a cloud-based website builder that allows users to create websites through a drag-and-drop interface. It offers a variety of templates, and users can customize their sites with ease.

3. Squarespace:

- Squarespace is a website builder known for its elegant and professional templates. It caters to users who want visually appealing websites without delving into the technical aspects of coding.

4. Weebly:

- Weebly is a user-friendly website builder with a drag-and-drop interface. It's suitable for beginners and small businesses, offering various themes and e-commerce capabilities.

5. Shopify:

- Shopify is a specialized platform for e-commerce websites. It provides a simple way to set up online stores, manage products, and handle transactions.

6. Adobe Dreamweaver:

- Dreamweaver is a professional web development tool by Adobe. It supports both visual design and coding, making it suitable for users with various levels of expertise.

7. Google Sites:

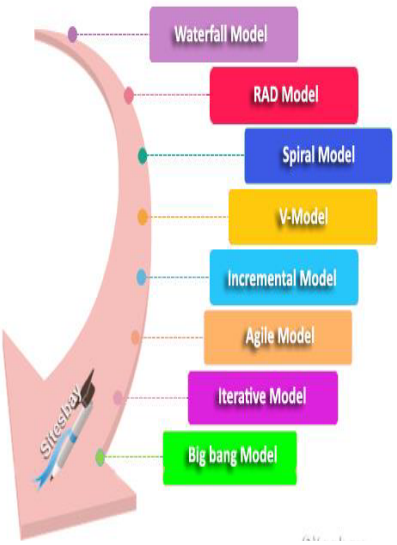
- Google Sites is a free and straightforward tool for creating websites. It's part of the Google Workspace and is suitable for creating intranet sites, project wikis, and simple websites.

8. GitHub Pages:

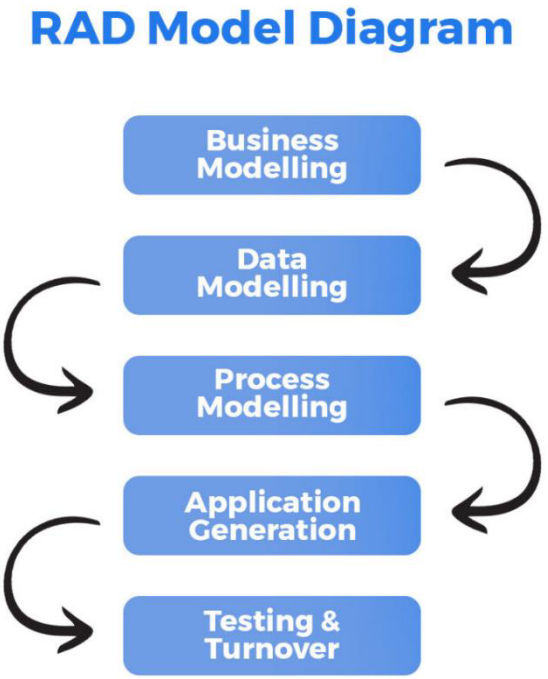
- GitHub Pages is a free hosting service that leverages the power of Git and GitHub repositories. It's suitable for developers who want to host static websites directly from their GitHub repositories.

4. Web Site Development Model

RAD	Waterfall	Agile
Builds a functional, working model of the application in the fastest way possible	Emphasizes intensive planning and follows through on set objectives	Builds the app by breaking down large objectives into smaller 'sprints'
Perfect for projects that require the shortest time to complete	Projects are thoroughly planned, and execution is typically time-consuming	Helps develop projects in periodical milestones or 'sprints'
Can adjust to changing requirements	Does not allow for changes once planning is done	Can quite easily adjust to changes even at later stages
Involves clients throughout the development cycle	Only involve clients during the planning stage	Involves clients throughout the development



➢ Rapid Application Development Model (RAD)



The five rapid application development (RAD) phases include

1. **Business modeling**
2. **Data modeling**
3. **Process modeling**
4. **Application generation**
5. **Testing and turnover**

- Developers can add new features and functionalities to the application at any given time.
- RAD also gets rid of the planning phase in favour of prioritizing speed. Software becomes ready for use in a shorter period. Multiple testing ensures that each application fully meets the users' needs. These are the five RAD phases:

Phase 1: Business modelling

- At this stage, information flow between different business functions is defined by answering the following questions:
  - ✓ What data drives the business process?
  - ✓ Who generates the data?
  - ✓ Where does the information go?
  - ✓ Who processes it?

- Information is gathered through many business-related sources. This information is combined to create a useful description of how the data will be used when it is processed.

Phase 2: Data modeling

- The information gathered during the Business Modeling phase is analyzed and categorized into different groups (data objects) that are needed to support the business.
- The attributes of each group are identified, and the relationship between them is defined.

Phase 3: Process modeling

- Process modeling is the third phase of the RAD methodology. Here, all the information groups from the Data Modeling stage are converted into usable data models.
- These models help extract information from the data objects for making the changes required for proper processing and are responsible for executing business functions.
- At this stage, changes and optimization in development can be done to increase value and quality.

Phase 4: Application generation

- Here, all the data gathered is coded, and the system that will be used to build the prototype is developed.
- The data models become actual prototypes that can be tested.
- Automated tools are used to speed up construction of the software.

Phase 5: Testing and turnover

- This RAD methodology phase ensures less time is spent testing the prototypes created.
- Every prototype is tested separately to modify the components quickly and to create the best product.
- The overall testing time is reduced since many of the programming components have already been tested.

5. Meaning of Scripting Language

- A scripting language is a type of programming language that is designed for writing scripts, which are typically short programs that automate tasks or perform specific functions.
- Unlike compiled languages, where the code is translated into machine code before execution, scripting languages are often interpreted at runtime.
- This means that the source code is executed line by line by an interpreter, rather than being translated into machine code beforehand.
- Scripting languages are commonly used for tasks such as automation, system administration, web development, and other applications where rapid development and ease of use are important.
- They are often more high-level and abstracted than lower-level languages, making them easier to learn and use for certain tasks.

6. Types of Scripting Language

6.1 JavaScript

- JavaScript is primarily known as a client-side scripting language, meaning it is executed by web browsers on the client's side (i.e., in the user's browser).
- However, JavaScript has evolved over the years, and it is now used in various environments. Here are the primary types of JavaScript usage:

1. **Client-Side JavaScript (CSJS):**

- **Web Browsers:** JavaScript is most commonly associated with web development and is executed by web browsers to enhance the interactivity and dynamic behavior of websites. It can manipulate the Document Object Model (DOM) to modify the content and structure of a webpage on the client side.

2. **Server-Side JavaScript (SSJS):**

- **Node.js:** JavaScript can also be used on the server side with the help of environments like Node.js. Node.js allows developers to run JavaScript server-side, enabling them to build scalable and efficient network applications.

3. **Mobile App Development:**

- **React Native, NativeScript:** JavaScript, along with frameworks like React Native or NativeScript, can be used for developing mobile applications that run on both iOS and Android platforms. This enables developers to write code once and deploy it on multiple platforms.

4. **Game Development:**

- **Game Engines (e.g., Phaser):** JavaScript is used in game development, particularly for web-based games. Game development frameworks and engines like Phaser make it possible to create interactive games using JavaScript.

5. **Desktop Applications:**

- **Electron:** JavaScript, along with HTML and CSS, can be employed for building cross-platform desktop applications using frameworks like Electron. Applications like Visual Studio Code and Slack use Electron.

6. **Embedded Systems:**

- **Internet of Things (IoT):** JavaScript can be used in embedded systems and IoT devices to control and manage connected devices.

7. **Serverless Computing:**

- **AWS Lambda, Azure Functions:** JavaScript can be utilized in serverless computing environments where code is executed in response to events without the need for dedicated servers. Platforms like AWS Lambda and Azure Functions support JavaScript.

8. **API Development:**

- **Express.js:** JavaScript can be used on the server side for building APIs. Express.js is a popular web application framework for Node.js that simplifies the process of creating robust APIs.

JavaScript's versatility and broad range of applications contribute to its widespread use across different domains in the software development landscape.

6.2 VBScript

The VBScript stands for Visual Basics Script language. Basically it is the combination of Visual Basic programming language and JavaScript language. VBScript was invented and maintained by Microsoft. It is used to develop dynamic web pages. It is much lighter compared to Visual Basic programming language but works as a scripting language like JavaScript. To run VBScript on the client-side, the client must have to use Internet Explorer because other browsers are still not supported by the VBScript.

VBScript currently runs on below mentioned environments:

- Internet Information Server (IIS) – It is a Microsoft web server.
- Windows Script Host(WSH) – It is a native hosting environment of Windows operating system.
- Internet Explorer (IE) – It is the simplest hosting environment where we can run VBScript code.

6.3 ASP

ASP.NET is a web application framework designed and developed by Microsoft. ASP.NET is open source and a subset of the .NET Framework and successor of the classic ASP(Active Server Pages). With version 1.0 of the .NET Framework, it was first released in January 2002. So a question comes to mind that which technology we were using before the year 2002 for developing web applications and services? Answer is Classic ASP. So before .NET and ASP.NET there was Classic ASP.

ASP.NET is built on the CLR(Common Language Runtime) which allows the programmers to execute its code using any .NET language(C#, VB etc.). It is specially designed to work with HTTP and for web developers to create dynamic web pages, web applications, web sites, and web services as it provides a good integration of HTML, CSS, and JavaScript.

.NET Framework is used to create a variety of applications and services like Console, Web, and Windows, etc. But ASP.NET is only used to create web applications and web services. That's why we termed ASP.NET as a subset of the .NET Framework.

6.4 PHP

The term PHP is an acronym for PHP: Hypertext Pre-processor. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension ".php".

Rasmus Lerdorf inspired the first version of PHP and participated in the later versions. It is an interpreted language and it does not require a compiler.

- PHP code is executed in the server.
- It can be integrated with many databases such as Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, and Informix.

- It is powerful to hold a content management system like WordPress and can be used to control user access.
- It supports main protocols like HTTP Basic, HTTP Digest, IMAP, FTP, and others.
- Websites like www.facebook.com and www.yahoo.com are also built on PHP.
- One of the main reasons behind this is that PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file.
- The thing that differentiates PHP from the client-side language like HTML is, that PHP codes are executed on the server whereas HTML codes are directly rendered on the browser. PHP codes are first executed on the server and then the result is returned to the browser.
- The only information that the client or browser knows is the result returned after executing the PHP script on the server and not the actual PHP codes present in the PHP file. Also, PHP files can support other client-side scripting languages like CSS and JavaScript.

7. Differences between Client-Side &Server-Side Scripting

1. Client-side scripting:

1. Web browsers execute client-side scripting.
2. It is used when browsers have all code.
3. Source code is used to transfer from webserver to user's computer over the internet and run directly on browsers.
4. It is also used for validations and functionality for user events.
5. It allows for more interactivity.
6. It usually performs several actions without going to the user.
7. It cannot be basically used to connect to databases on a web server.
8. These scripts cannot access the file system that resides in the web browser..
9. Pages are altered on basis of the user's choice. It can also be used to create "cookies" that store data on the user's computer.

2. Server-side scripting :

1) Web servers are used to execute server-side scripting.
2) They are basically used to create dynamic pages.
3) It can also access the file system residing at the webserver.
4) A server-side environment that runs on a scripting language is a web server.
5) Scripts can be written in any of a number of server-side scripting languages available.
6) It is used to retrieve and generate content for dynamic pages.
7) In this load times are generally faster than client-side scripting.
8) When you need to store and retrieve information a database will be used to contain data.
9) The server sends pages to the request of the user/client.

Difference between client-side scripting and server-side scripting :

Client-side scripting	Server-side scripting
Source code is visible to the user.	Source code is not visible to the user because its output of server-sideside is an HTML page.
Its main function is to provide the requested output to the end user.	Its primary function is to manipulate and provide access to the respective database as per the request.
It usually depends on the browser and its version.	n this any server-side technology can be used and it does not depend on the client.
It runs on the user's computer.	It runs on the webserver.
It does not provide security for data.	It provides more security for data.
It is a technique used in web development in which scripts run on the client's browser.	It is a technique that uses scripts on the webserver to produce a response that is customized for each client's request.
HTML, CSS, and javascript are used.	PHP, Python, Java, Ruby are used.
It reduces load on processing unit of the server.	It surge the processing load on the server.

8. Introduction to JavaScript

- JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

Javascript History

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time.

9. Advantages of JavaScript

- **Simple Syntax**: JavaScript has a simple syntax that makes it easy to learn and use. This makes it an ideal choice for novice developers as well as experienced JavaScript developers.
- **Platform Independent**: JavaScript is platform independent, meaning it can run on any platform such as Windows, Linux, or Mac OS.
- **Interactive Web Pages**: JavaScript can be used to create interactive web pages that can react to user input. This makes it easier for users to interact with websites and applications.
- **Cross-browser Compatibility**: JavaScript is compatible with all major browsers such as Chrome, Firefox, Safari, and Internet Explorer.

- **Rich Interface**: JavaScript can be used to create a rich interface with animations, graphics, and interactive elements.
- **Easy to Debug**: JavaScript has tools that make it easy for developers to debug code and find errors quickly.

10.Limitation of JavaScript

- **Security Issues**: JavaScript is client-side scripting language which means the code runs on the user's computer. This can lead to security issues if not implemented properly.
- **Browser Support**: Not all browsers support the latest version of JavaScript. This can lead to compatibility issues when developing cross-browser applications.
- **Speed Issues**: JavaScript can be slower than other languages due to its interpreted nature. This can cause performance issues in applications that require high performance.
- **Poor Error Handling**: JavaScript does not have a good error handling mechanism which can make debugging difficult.
- **Lack of Libraries**: JavaScript has limited libraries compared to other languages such as Java or Python.