INTRODUCTION TO WEB DEVELOPMENT

Objectives

By the end of this lesson, you should be able to answer these questions:

What is the Web?

How does the web work?

What is the possible usage of the web?

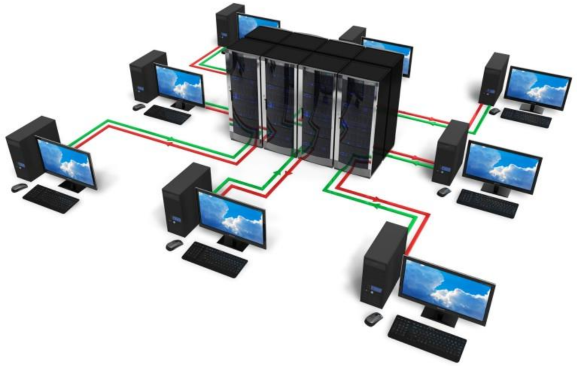
what exactly is the web?

Think of it as a gigantic network of computers all around the world.

It's like a huge library where you can find all sorts of information, watch videos, play games, and more.

Definition of the Web?

The **Web** or **World Wide Web** is a collection of electronic documents linked together like a spider web.



These documents are stored on computers called **servers** located around the world.

What Is the Web Made of?

- The Web consists of:
  - Your personal computer
  - **Web browser** software to access the Web
  - A connection to an **Internet service provider** (ISP)
  - **Servers** to host the data
  - **Routers** and **switches** to direct the flow of **data**

BROWSER SOFTWARE
ISP
ROUTERS
YOUR COMPUTER
SERVERS

How the Web Works?

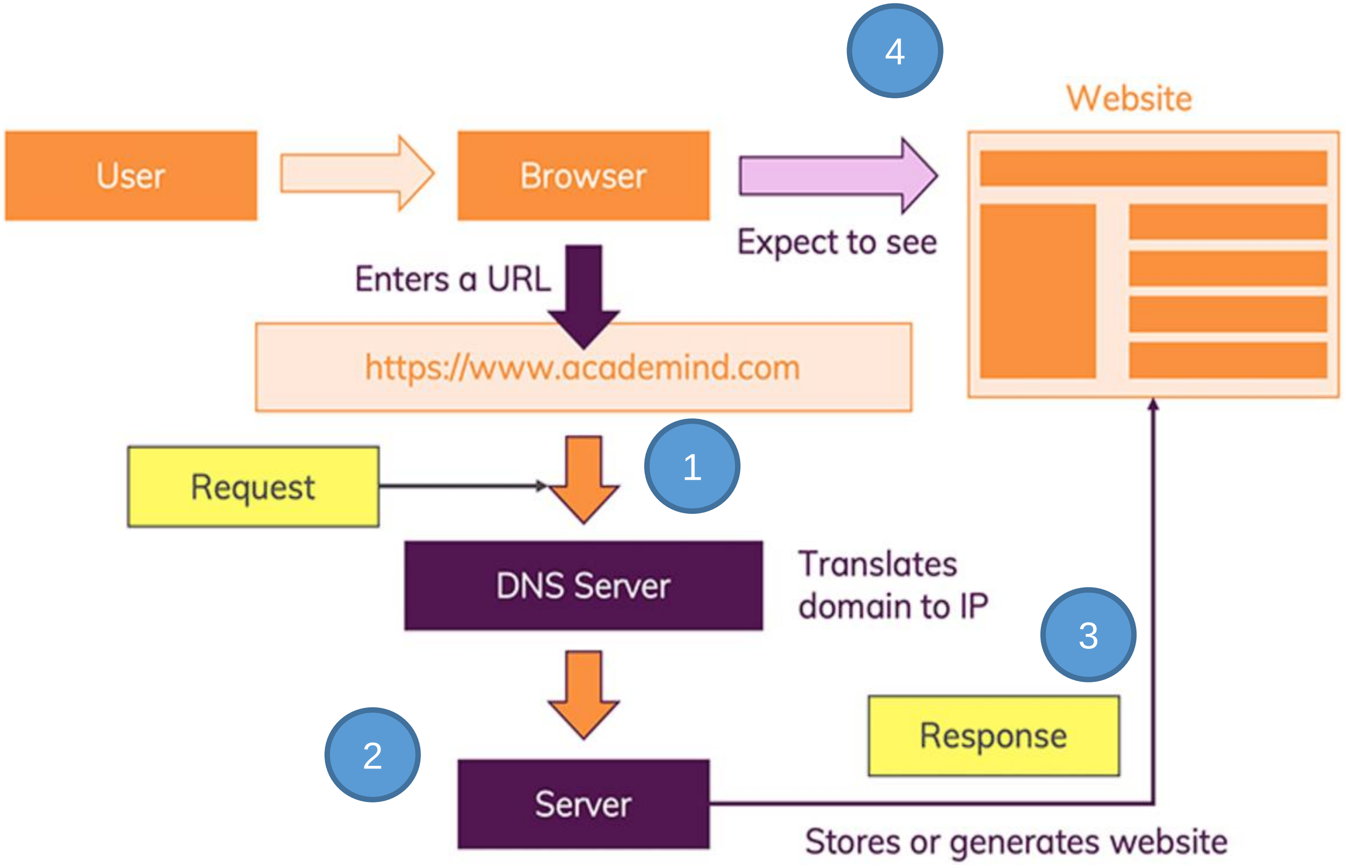
The moment you enter the address in your browser and you hit ENTER, a lot of different things happen:

1. The URL gets resolved.

2. A Request is sent to the server of the website.

3. The response of the server is parsed.

4. The page is rendered and displayed.



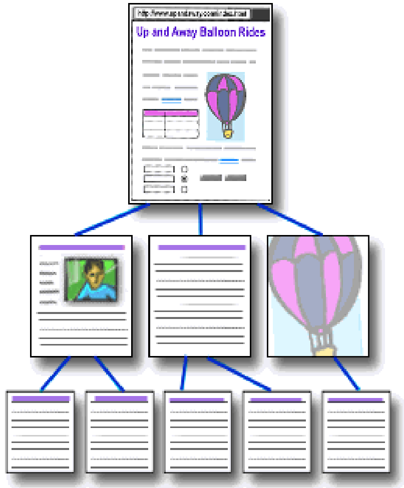
Web Pages

- A **web page** is an electronic document written in a computer language called **HTML** (**Hypertext Markup Language**).
- Web pages can contain text, graphics, audio, video, and animation, as well as interactive features, such as data entry forms and games.
- Each page has a unique address known as a **URL** (**Uniform Resource Locator**), which identifies its location on the **server**.
- Web pages usually contain **hyperlinks** to other web pages. Hyperlinks are text and images that reference the addresses of other web pages.

http://www.upandaway.com/index.html
Up and Away Balloon Rides

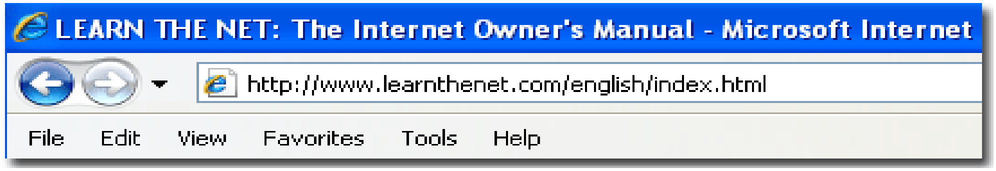
Websites

- A **website** consists of one or more web pages that relate to a common theme, such as a person, business, organization, or a subject, such as news or sports.
- The first page is called the **home page**, which acts like an **index**, indicating the content on the site.
- From the home page, you can click **links** to access other pages on the site or other resources on the Web.



Using Web URLs

- A **URL** indicates where the web page is stored on the Internet.
- URLs never use back slashes (\). All slashes are forward slashes (/).
- You need to type a URL *exactly* for your browser to locate the desired web page, otherwise you will access the wrong site or get an error message.
- URLs may not contain spaces between characters.
- The **location box** or **address field** on your browser indicates the URL of the page you arrived at after clicking a link.



LEARN THE NET: The Internet Owner's Manual - Microsoft Internet

http://www.learnthenet.com/english/index.html

Typical uses of websites

- There are millions of websites on the internet. They are used for almost every conceivable purpose including business, personal and entertainment purposes. Some of the more common purposes of websites are described in the following sections.
  - Presenting information
  - Storing information
  - Browsing and searching for information
  - Improving productivity
  - Making decisions
  - Communication with people
  - Media sharing
  - E-commerce
  - Education
  - Downloading Information

Presenting Information

- Websites are commonly used both for presenting information, such as news, and for product advertising.

BBC NEWS

sky NEWS

Storing Information

- It involves using the internet to store information rather than using local storage (such as your computers hard disk drive or a USB memory stick).

Browsing and searching for information

- The internet search engine Google has become such a part of our lives that the verb ‘to Google’ has been included in the Oxford English Dictionary.
- If you ask someone a question, you may get a response, ‘Just Google it’.

I don't know,

Google it.

Improving productivity

- Email has Improved personal and business communication, enabling people to share information and send document to each other quickly and so improving productivity.

- Facilities such as video conferencing allow people to have ‘virtual’ meetings, enabling them to work collaboratively without having to spend time travelling.

Making decisions

- Many websites provide information which helps individuals and businesses to make decisions.

Communication with people

- Social networking sites, such as Facebook, have become very popular as they allow people to communicate with their friends.

Media sharing

- Another area that the internet has revolutionized is how we listen to music or live radio, and how we watch films.
- such as Apple iTunes allow consumers to purchase and download music, while sites such as YouTube let people view and share videos on a wide range of subjects.

E-commerce

- There has also been a huge increase in the number of goods being brought online, both from online-only retailers like Amazon and high street chains, such as Tesco and John Lewis.

- A large proportion of trade business is now conducted online.

Education

- The internet contains a vast resource of information on just about every subject imaginable. You can find information on encyclopaedia websites or specialist websites.

Downloading Information

- Many websites provide digital content for you to download. iTunes is one of the best known websites from which you can buy and download music.

Summary

Web : is a collection of electronic documents linked together like a spider web.

Webpage : is an electronic document written in a computer language called HTML.

Website : consists of one or more web pages connected to gather.

URL : unique address which identifies the web page location on the server.

DNS Server : A DNS server is like a phone book for the internet. It translates website names (like "google.com") into their actual IP addresses.

Web Server : its where we store websites. It receives requests from your browser for web pages, finds those pages, and sends them back to your browser so you can see them.

see you in the next lesson

Full Stack Developer
Role Description

Objectives

By the end of this lesson, you should be able to answer these questions:

What is the Architecture of a Web Application?

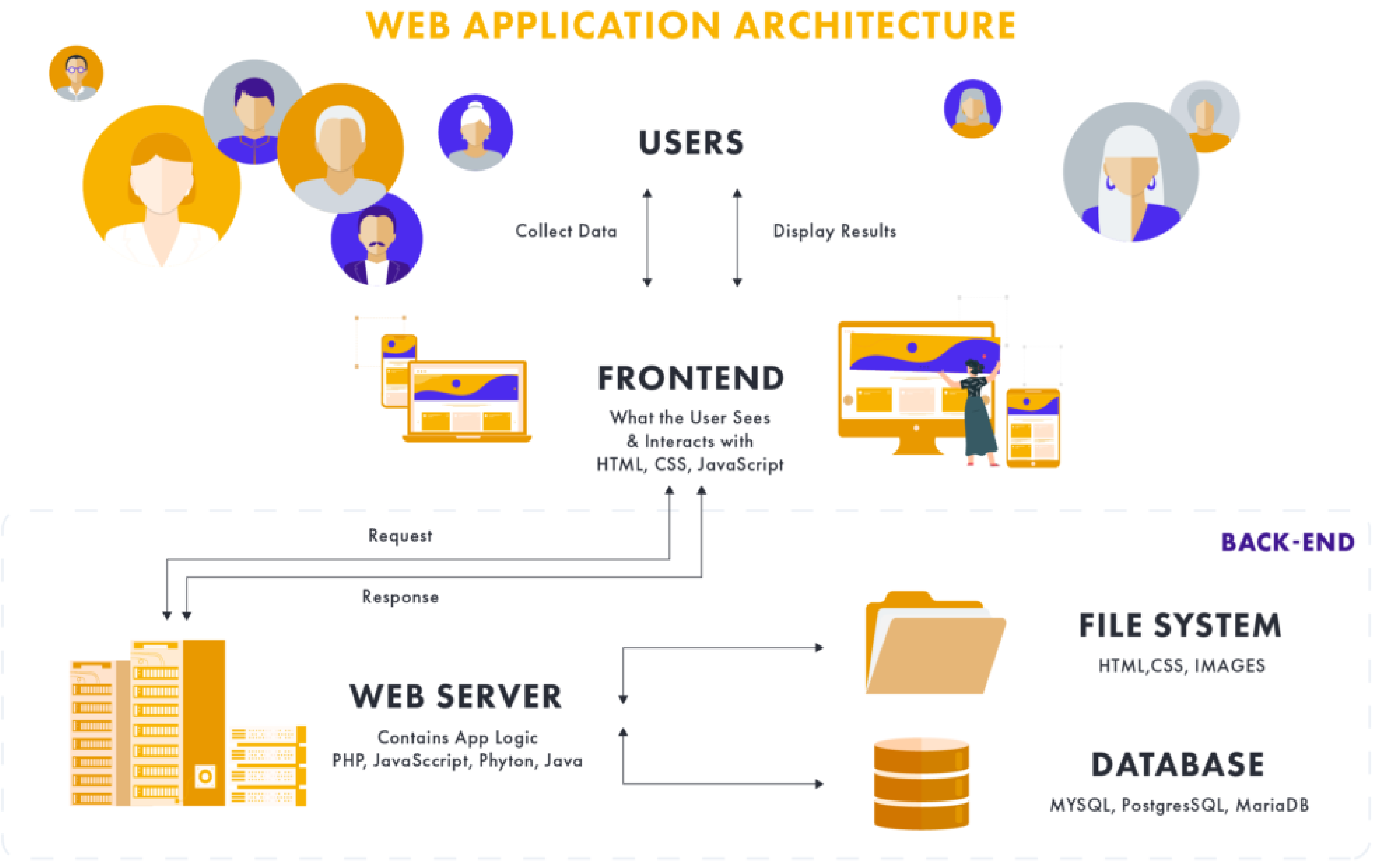
What is Differences between frontend, backend and full stack?

What are the responsibilities of a full stack developer ?

How To Become a Full-Stack Developer?

Frontend

Backend



WEB APPLICATION ARCHITECTURE

USERS

Collect Data

Display Results

FRONTEND

What the User Sees
& Interacts with
HTML, CSS, JavaScript

Request

Response

BACK-END

WEB SERVER

Contains App Logic
PHP, JavaSccript, Phyton, Java

FILE SYSTEM

HTML,CSS, IMAGES

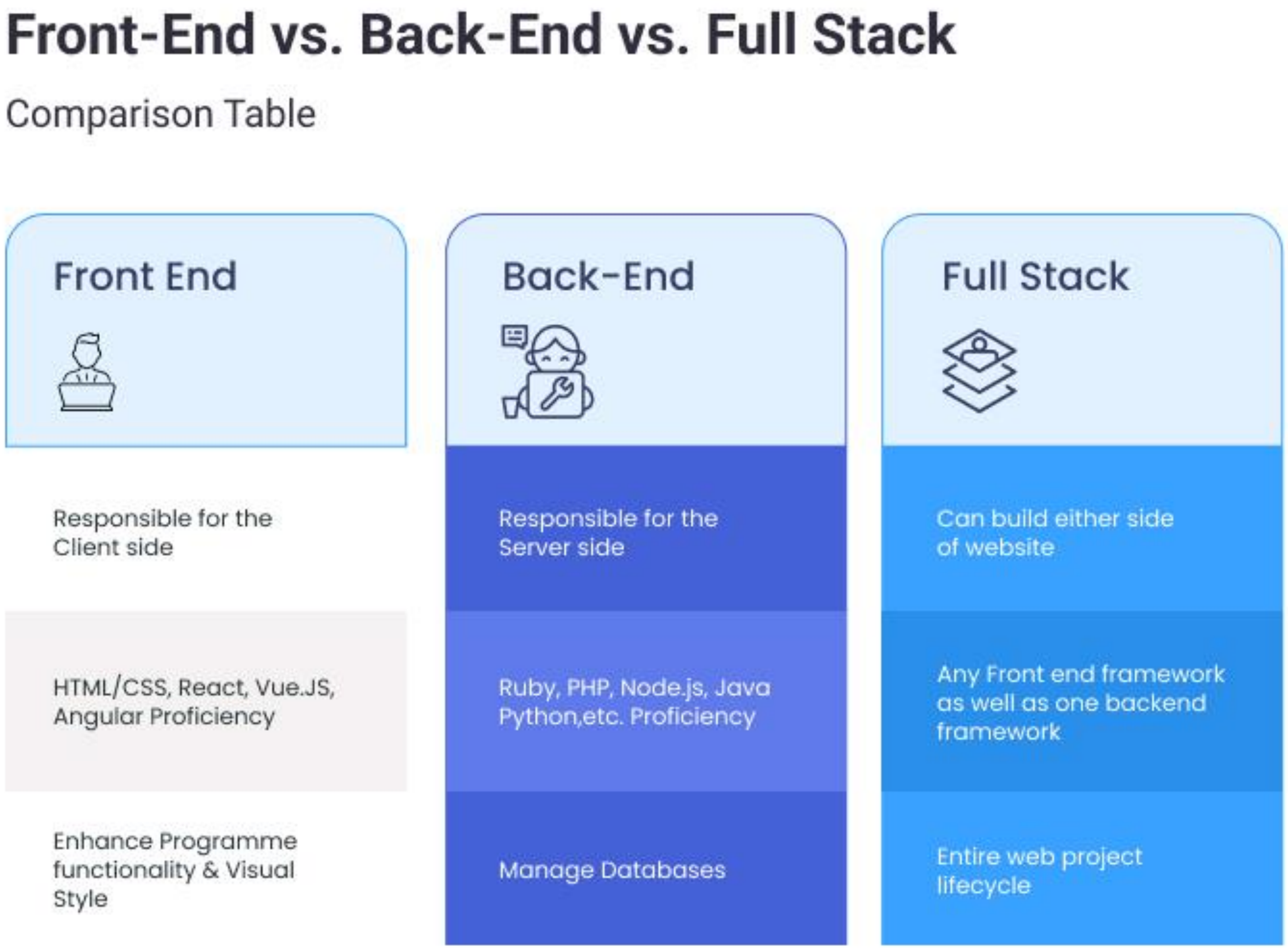
DATABASE

MYSQL, PostgresSQL, MariaDB

Full stack development

means working on both the parts of a website or app that users see (front end) and the parts they don't (back end).

A **full stack developer** knows how to do everything from designing what users interact with to managing the data and servers behind the scenes. They're like all-in-one developers who can handle every aspect of building an application.

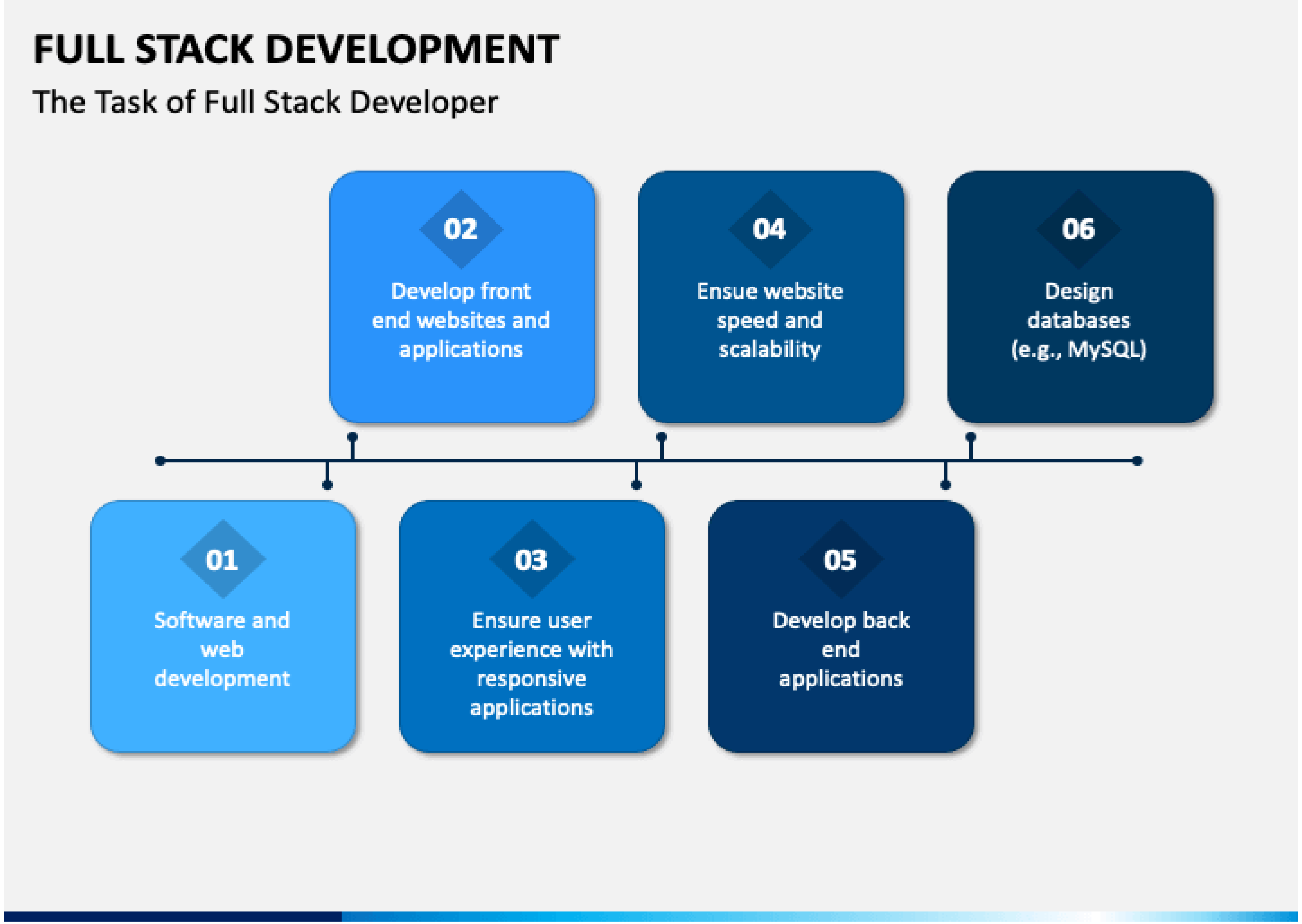


Front-End vs. Back-End vs. Full Stack

Comparison Table

Front End	Back-End	Full Stack
Responsible for the Client side	Responsible for the Server side	Can build either side of website
HTML/CSS, React, Vue.JS, Angular Proficiency	Ruby, PHP, Node.js, Java Python,etc. Proficiency	Any Front end framework as well as one backend framework
Enhance Programme functionality & Visual Style	Manage Databases	Entire web project lifecycle

Understanding the Full Stack Developer responsibilities



FULL STACK DEVELOPMENT

The Task of Full Stack Developer

01 Software and web development

02 Develop front end websites and applications

03 Ensure user experience with responsive applications

04 Ensue website speed and scalability

05 Develop back end applications

06 Design databases (e.g., MySQL)

How To Become a Full-Stack Developer: A 10-Step Guide

Here are the steps to take to launch a career as a full-stack developer:

1. Obtain the Requisite Education
2. Master the Essential Programming Languages and Tools
3. Develop Your Skillset
4. Pursue Volunteer, Open-Source, or Freelance Work and Participate in Coding Challenges To Build Your Portfolio
5. Create a GitHub Profile
6. Your Network Is Key
7. Pursue a Full-Stack Development Internship
8. Find a Mentor
9. Ensure Your Resume Is Up-to-Date and Start Applying for Jobs
10. Do Coding Exercises Daily

How To Become a Full-Stack Developer: A 10-Step Guide

Obtain the Requisite Education

- While a solid education will lay the groundwork for your full-stack development career, there are many different paths you can take. Let’s take a look at some of them.

- Formal Education / University Degree
- Bootcamp
- Self-Taught Route

How To Become a Full-Stack Developer: A 10-Step Guide

Languages and Tools

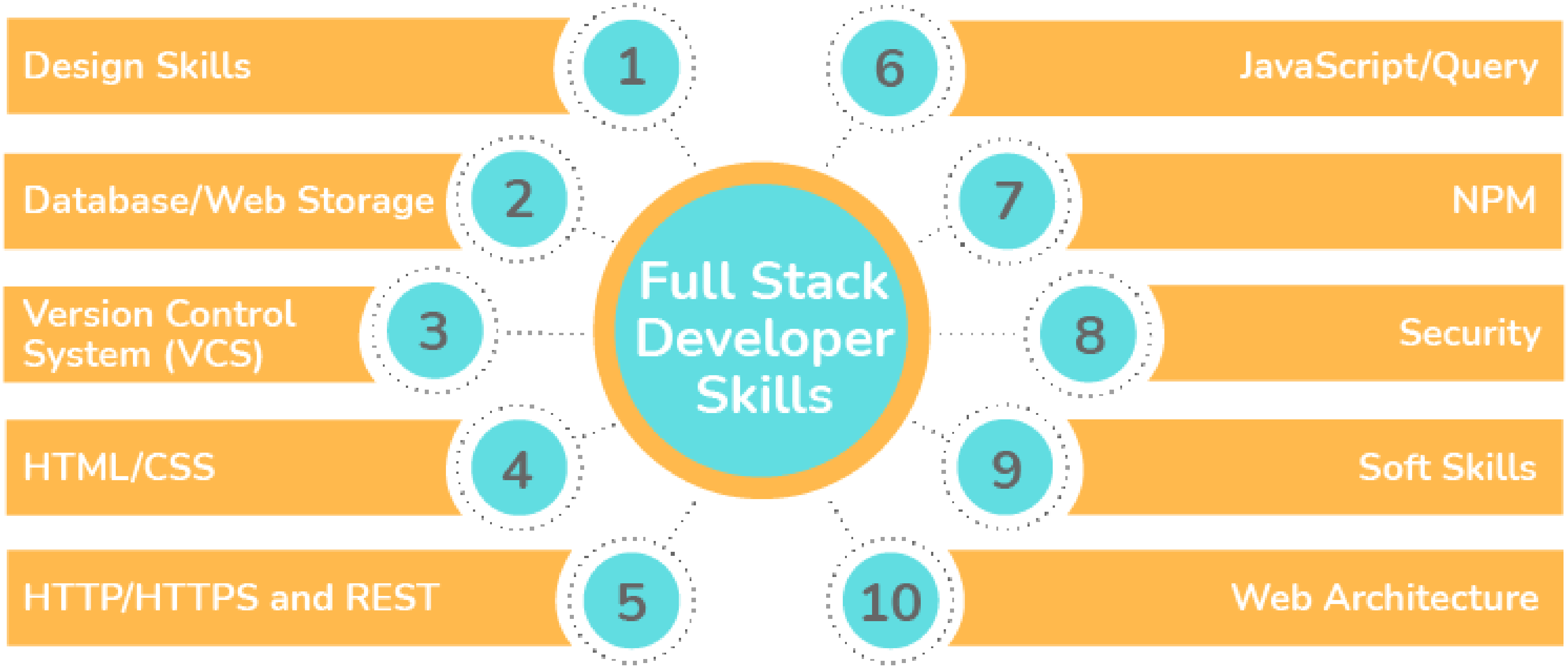
- Let’s take a look at the languages and tools you’ll need to master:

- **Languages:** HTML, CSS, JavaScript, Bootstrap, React, Node.js, Databases.

- **Tools:** Text editors, code editors, Terminals or consoles to run your commands and code, Database storage for the websites or mobile applications you are trying to build, GitHub.

How To Become a Full-Stack Developer: A 10-Step Guide

Develop Your Skillset



Full Stack Developer Skills

1 Design Skills

2 Database/Web Storage

3 Version Control System (VCS)

4 HTML/CSS

5 HTTP/HTTPS and REST

6 JavaScript/Query

7 NPM

8 Security

9 Soft Skills

10 Web Architecture

How To Become a Full-Stack Developer: A 10-Step Guide

Pursue Volunteer, Open-Source, or Freelance Work and Participate in Coding Challenges To Build Your Portfolio

- A full-stack developer portfolio is proof of your expertise. To build this, volunteer to create apps and websites for local businesses. You can also take up small freelancing jobs. Keep an eye out on websites like Hackathon and HackerEarth for coding challenges.

Create a GitHub Profile

- A GitHub profile is a great way to exhibit your portfolio. You can also connect with other developers, and collaborate with them. Through GitHub, you can share code files easily and track different versions within each project.

How To Become a Full-Stack Developer: A 10-Step Guide

Your Network Is Key

- Your network can be a shortcut to landing internships and jobs. Here are some ways to build that:
- LinkedIn.
- Online communities like Reddit and GitHub.

Pursue a Full-Stack Development Internship

- Once you have a portfolio and some education, you can start to apply for internships, where you'll be able to apply your knowledge.

Find a Mentor

- Mentors can be an incredible resource. Finding one is a bit trickier. Rather than pursuing a stranger, let this sort of relationship develop organically over time—an internship is a great place for this to happen.

How To Become a Full-Stack Developer: A 10-Step Guide

Ensure Your Resume Is Up-to-Date and Start Applying for Jobs

- Once you’ve got some experience—either in the form of internships or a portfolio—you can start applying for jobs. Make sure that you’re tailoring your resume for each position so that recruiters know that you’ve considered the role thoroughly and that you aren’t just applying at random.

Do Coding Exercises Daily

- At some point during the interview process, you’re going to need to do a coding exercise, which is usually about 30-40 minutes long. So as you’re applying for jobs, practice coding every day. It’s a great way to solidify your knowledge and can help alleviate the nervousness that often comes with job hunting.

That's all for this lesson. I hope you enjoyed it. Thank you for listening, and see you in the next lesson!

Development Environment

Objectives

By the end of this lesson, you should be able to answer these questions:

What is Code Editors?

What is Version Control?

Why Version Control systems are useful?

What is GitHub?

How to Create a GitHub Repository and host your first Webpage?

Text Editors Vs Code Editors

- Text editor is used to edit text files as the name suggest and there is nothing great about it (as per my knowledge). Example of a text editor is notepad.

- Code editor on the other hand is specifically meant to edit code. They provide you with various features like auto indent, bracket matching, different colors for variables, key words, comments, pre-processor directives, strings, etc.

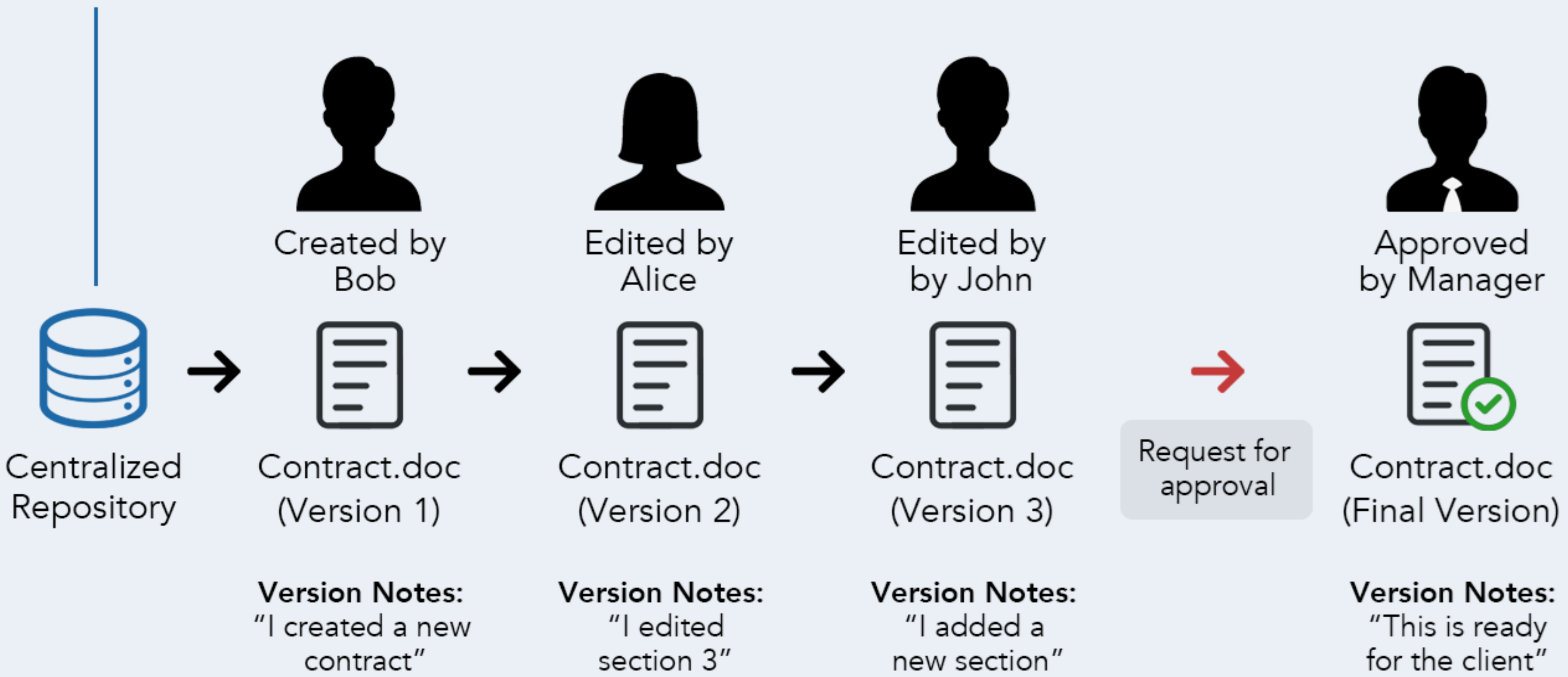
- Code editors are mostly integrated with the IDE (Integrated Development Environment).

What is Version Control?

- Version control systems are software that help track changes make in code over time. As a developer edits code, the version control system takes a snapshot of the files. It then saves that snapshot permanently so it can be recalled later if needed.

- Popular version control systems for developers include SVN, Git (not to be confused with GitHub).

How Version Control Works



Centralized Repository

Created by Bob

Contract.doc (Version 1)

Version Notes: "I created a new contract"

Edited by Alice

Contract.doc (Version 2)

Version Notes: "I edited section 3"

Edited by by John

Contract.doc (Version 3)

Version Notes: "I added a new section"

Request for approval

Approved by Manager

Contract.doc (Final Version)

Version Notes: "This is ready for the client"

Why Version Control systems are useful

Learn by Real Life Scenarios

I’ve made a change I didn’t mean to make

- the core concept of a Version Control System is the **creation of versions** of your code over time. Implicit in this functionality is the ability to revert to an old version of your code and undo changes that have been made.

I’ve made some changes, but I don’t remember what they were

- With your code being managed by a Version Control System, you can easily **see the changes you have made since the last time you have committed your code** (“committing” your code is when you save the changes to code you have been working on.

Why Version Control systems are useful

Learn by Real Life Scenarios

My computer crashed and now I don’t have access to my code

- Most **VCS can run on a server**, so your changes can be tracked and backed up remotely, In the event of a computer crash, you can simply fetch your codebase from that server and get to work.

We can’t all work on the same file, can we?

- you **can set up a remote Version Control System, where developers can git fetch a copy of the code** onto their machine and work independently from other developers.

- Once everyone is done with their work, it is easy to merge all the different versions into a common file that incorporates everyone’s modifications.

Why Version Control systems are useful

Learn by Real Life Scenarios

I’m not sure what I accomplished today

- Version Control systems allow you to have a good **overview of what you’ve accomplished for a given period of time**.
- a **history** also allows you to track down changes to specific lines of code, so you can source why a change was made, or how a bug first appeared.

I want to try something, but I’m scared I’ll break something else

- A powerful concept known as “**branching**” is widely used in the software development industry. Branching is a great way to isolate changes to your code and allows for easy reversing of problematic code.

What is GitHub?

- GitHub is a company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

- anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

That's all for this lesson. I hope you enjoyed it. Thank you for listening, and see you in the next lesson!