Frontend /Backend

Lesson Time: 10 Minutes

BROWSER IS THE FRONT END.
SERVER IS THE BACK END.

Web development can be grouped into two categories--the frontend and the backend of a web site.

The frontend is always made up of HTML, CSS, and Javascript technologies and involves all of the tasks that happen in the web browser. Frontend tasks include things like designing the layout of the web page, creating CSS stylings, updating the content of HTML pages, and writing javascript for the web browser to execute.

The backend involves tasks that happen on the web server. There are many more programming languages to choose from on the backend. PHP, Node.JS, JSP, and ASP.NET are all examples of backend technology, and websites typically only use one of these languages for the backend. Backend tasks might include programming the server to processing input from user forms or return data to the web browser from a database.

Web developers can choose to specialize in the frontend or the backend. Skilled frontend developers work with graphic designers to implement beautiful site layouts and have a focus on creating the best experience possible for the user. Experienced backend developers have a focus on deep knowledge of programming languages.

A full stack developer is a developer who can implement both the frontend and the backend of a website, as well as implement the database portions of the site. A full stack developer should also have good project management and be able to work with a team.

Key Terms	
Lesson Files	
Additional Resources	
Further Learning	

What is the Stack?

Lesson Time: 10 Minutes

SOFTWARE TOOLS FOR EVERY BROWSERS, SERVERS, AND DATABASES.

So what is the “stack” in Full Stack Web Developer? The stack is the set of technologies we use to create a complete web app. There is no one software tool we use create web apps. Instead, we use a suite of tools, which we call the stack.

There are many different versions of the stack. Popular Stacks include the LAMP stack (Linux, Angular, MongoDB, PHP), the MEAN stack (MongoDB, Express.JS, AngularJS, and Node.JS) and the Microsoft stack. This course teaches the Microsoft Stack.

Let’s look at the Microsoft stack more closely.

Area	**Technology**	**Where**
USER INTERFACE LEVEL	HTML, CSS, Javascript	Happens in Web Browser
MIDDLE TIER LEVEL/ SERVER LEVEL	ASP.NET Framework & C#	Happens on Web Server
DATABASE LEVEL	Microsoft SQL Server	Happens in the Database

To make a website work, we have 3 “problem” areas to solve.
1. User Interface - How will my users interact with my app?
Answer: In the web browser, using HTML, CSS, and Javascript.

2. How will my app send/receive data to/from the browser?
Answer: On my Web Server, using ASP.NET and C# programming language

3. How will my app permanently store data it needs?
Answer: In my Database, using Microsoft SQL Server

This course focuses on the User Interface technologies.

Key Terms	
Lesson Files	
Additional Resources	
Further Learning	

Websites vs Web Apps

Lesson Time: 10 Minutes

YOU SAY TOMATO I SAY TOMATO

All web apps are web sites, but not all web sites are web apps. The difference is typically around how a site is planned to be used. A basic website might only contain information about a company for the user to read. A Web application is typically intended for the user to perform a task.

However, there is a bit of a grey area here. The folks at Airbnb would call their site a web app, but we call amazon an ecommerce website. People say they visit walmart's website, but you can shop there.

For this class, we'll consider a web app and website that allows a user to create, read, edit, and delete data. A web app is meant to compete with a traditional desktop or mobile app. At the end of the day, the same skills and tools are used to build web apps and web sites.

Key Terms	
Lesson Files	
Additional Resources	
Further Learning	

Benefits of Web Apps

Lesson Time: 10 Minutes

NO OS REQUIREMENTS.
WRITE THE CODE ONCE.

There are many benefits to building your application as a web app. Benefits include

- No need to write code for different operating systems
- Nothing for users to install or update
- All users are using the same version of the app
- The only thing a user needs is a browser and an internet connection
- Lower cost to develop

Web apps are not perfect. Some of the problems with web apps are:

- Can be slower than desktop and mobile apps
- The internet is not 100% reliable, internet goes out
- Have to support different web browsers
- Not as well suited to graphic intensive apps like gaming, CAD, graphic arts

Key Terms	
Lesson Files	
Additional Resources	

Further Learning	