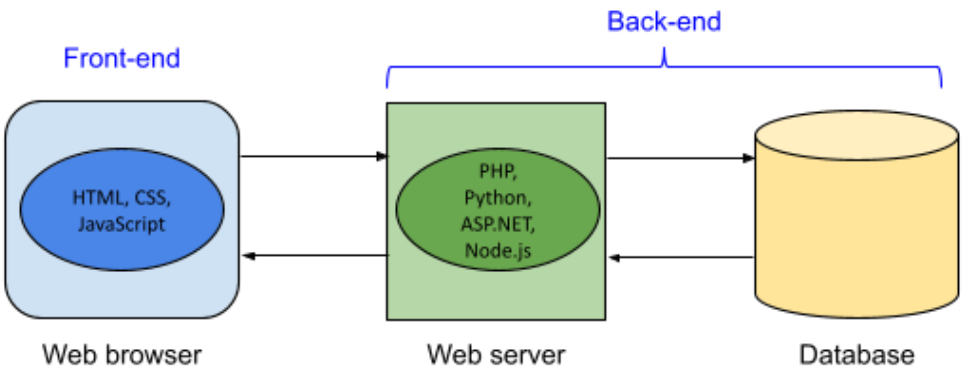
11.1 Full-stack development (Node)

Overview of front-end and back-end development

Most websites and web applications require the development of client-side technologies that interact with server-side technologies. ***Client-side*** (or ***front-end***) refers to those technologies that run in the web browser like HTML, CSS, and JavaScript. ***Server-side*** (or ***back-end***) refers to those technologies that run on the web server like PHP, Python, Node.js, etc. and databases. Ex: Amazon uses server-side technologies to store information on millions of products and a client-side search interface that interacts with the web server so customers can find and purchase products.

Figure 11.1.1: Front-end and back-end technologies.



Feedback?

A ***front-end developer*** is a developer that is proficient in client-side technologies. A ***back-end developer*** is a developer that is proficient in server-side technologies. Many developers strive to be proficient in both front-end and back-end technologies and how the two sides work together. A ***full-stack developer*** is a developer who has expertise in all aspects of a website or web application's development, including client technologies, server technologies, data modeling, and user interfaces. The "stack" in "full-stack" refers to the various layers that compose websites and web applications. Technology stacks have increased in complexity over the years, so even "full-stack" developers typically specialize in a few areas of the technology stack.

PARTICIPATION ACTIVITY 11.1.1: Primary layers of the full stack.

If unable to drag and drop, refresh the page.

Server and hosting environment	Issues regarding network throughput, cloud storage, virtualization, hardware constraints, multithreading, and data redundancy. Developers should understand how the server components interact and how the components can be scaled-up to support large volumes of traffic.	**Correct**
Data modeling	Representing, storing, and retrieving application data in relational and non-relational databases. Developers should know the tradeoffs of choosing one type of database over another.	**Correct**
Business logic	Programming logic on the front or back-end that determines how data can be created, displayed, stored, and changed. Developers need solid programming skills and the ability to use frameworks and libraries to simplify common tasks.	**Correct**
Application Programming Interface (API)	Programmable actions that may be performed on the underlying data. Often used by the front-end to interact with the back-end. Developers should understand how to create clear and concise programming interfaces.	**Correct**
User interface (UI)	Visual part of the application that users interact with. Developers are often aided by graphic designers and usability experts to create effective UIs, but developers need mastery of front-	**Correct**

	end technologies to implement UI designs.	
Testing framework	Automated tests that verify the web application components are working properly, independently and together. Developers should write tests to insure the business logic is implemented correctly and write tests that automate interaction with the web application in the web browser.	**Correct**

Reset

Feedback?

Web hosting

When creating a web application, developers must decide where the application and application data are going to be hosted. Large companies like Google, Amazon, and Facebook have the resources to host their web applications on their own servers. Smaller companies and individuals often outsource their server hosting to web hosting companies. A ***web hosting company*** is a company that hosts others' websites on the company's servers, usually for a fee. Factors to consider when choosing a web hosting company include:

- **Reliability**: Many web hosting companies guarantee a certain level of uptime, and the level can be increased by paying more. Some companies backup data daily, and others provide little to no backups.
- **Flexibility**: Websites that become popular may need to quickly scale-up to handle more users. Web hosting companies may offer a virtual private server that can quickly be duplicated on other servers to meet high demand. A ***virtual private server (VPS)*** is an autonomous server that is hosted on a physical server with other virtual servers. Amazon Web Services (AWS) allows organizations to host virtual servers in the Amazon cloud that can quickly scale-up hosted websites when necessary.
- **Security**: Hackers may attempt to access a website's data or upload malware to a hosted website that attacks the website, other hosted websites, or the website's users. ***Malware*** is malicious software designed to cripple a computer system or perform unwanted actions. Some hosting companies offer extra security measures like encrypting web traffic or providing dedicated servers in heavily-guarded data centers.
- **Price**: Some web hosting companies offer limited services for free and subsidize lost income with advertising. Prices go up depending on reliability, services provided, security, amount of traffic, etc. The most expensive plans usually involve dedicated hosting where the customer is given full control over the web server.

The choice of platform dictates many of the web application's implementation decisions, since certain server-side technologies are only offered on certain platforms. Most web hosting companies provide a Linux or Windows server to host the website. Linux servers typically use open-source software: Apache web server with support for PHP, Python, Ruby, or Perl, and the MySQL database system. Windows servers generally run Microsoft's IIS web server, which supports ASP.NET and the SQL Server database system. Linux servers usually cost less than Windows servers because of the use of open-source software.

PARTICIPATION ACTIVITY 11.1.2: Web hosting.

1) A small company may host the company's own website on the company's own web server.
   - (•) True
   - ( ) False

   Correct
   Some small companies and organizations have the expertise to run their own web server, but many lack the technical information needed to do so or just prefer not to.

2) A web hosting company that hosts websites for free is likely to provide services like backup, unlimited email addresses, and 24-hour customer support.
   - ( ) True
   - (•) False

   Correct
   Free hosting generally provides limited services.

3) A VPS generally runs slower than a dedicated host.
   - (•) True
   - ( ) False

   Correct
   A VPS is easier to create and configure than a dedicated host, but a VPS is often slower because multiple VPSs are running on the same physical server and must share resources like disk access.

4) Web hosting companies provide various levels of security.
   - (•) True
   - ( ) False

   Correct
   Most web hosting companies are quick to install software patches for security vulnerabilities, but providing extra layers of security costs money.

5) Web hosting companies generally

   Correct

charge more for hosting on Linux servers than for hosting on Windows servers.

- True
- False

Linux servers generally use open source software that does not cost money, but Microsoft servers generally run commercial software with the exception of ASP.NET, which is open source.

Feedback?

Server-side programming

Web developers have a wide range of options when choosing a server-side programming platform or language. When choosing a server-side programming platform, developers must consider:

- **Server platform**: Some web servers support certain languages and not others. Ex: IIS supports ASP.NET, and Apache supports PHP.
- **Tool support**: Some tools are ideal for working with certain programming languages. Ex: PhpStorm is ideal for PHP development, and Visual Studio is ideal for ASP.NET.
- **Developer experience**: JavaScript developers may choose Node.js instead of learning a new language like C#. Developers who are new to web development might already know Java or Python and prefer those languages.
- **Library support**: Some languages may have pre-built libraries that support some web applications better than others.

PARTICIPATION ACTIVITY 11.1.3: Server-side programming platforms and languages.

If unable to drag and drop, refresh the page.

PHP	Scripting language created in 1994 by Rasmus Lerdorf. Currently the most popular server-side language in use. Facebook was originally created with PHP, and PHP remains one of the easiest server-side languages to learn.	Correct
		Correct

ASP.NET	Collection of web development technologies first released in 2002 by Microsoft that uses the C# or VB.NET programming languages. ASP.NET is a powerful platform generally used on Windows servers.	
Python	General-purpose scripting language created by Guido van Rossum in the 1990s that uses frameworks like Django, web2py, and Flask to create web applications. Python is Google's language of choice. Python uses an easy-to-learn syntax without { curly braces }.	**Correct**
Ruby on Rails	Web application framework written in Ruby and created by David Heinemeier Hansson in 2004. Twitter was originally created with Rails. Rails emphasizes Convention over Configuration (CoC), meaning developers are encouraged to write code in a manner that follows a specific convention.	**Correct**
Java	Used to create applets on the front-end and servlets, JavaServer Pages, and web APIs on the back-end. Java has been used for web development since the late 1990s. Java is still a popular server-side language, but most developers now use Java primarily for creating web APIs.	**Correct**
	Runtime environment that uses modules written in	**Correct**

Node.js

JavaScript. Originally created in 2009 by Ryan Dahl.

Node.js is a relative new-comer, but many notable companies like Walmart and LinkedIn tout Node's ability to scale-up efficiently.

Reset

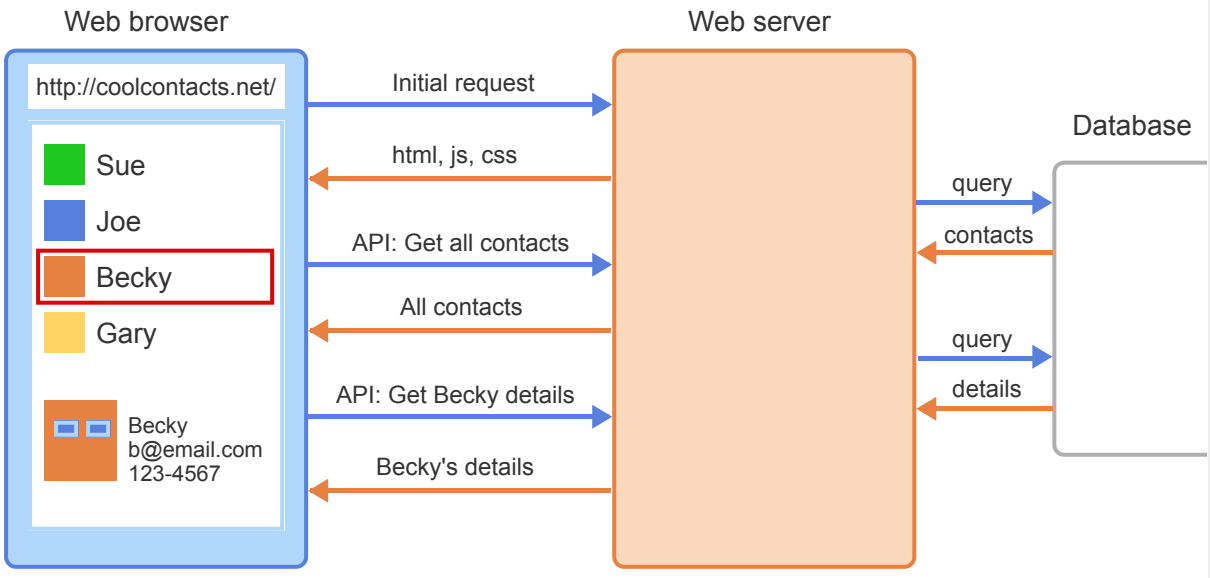
Feedback?

Developers have traditionally used server-side technologies to generate dynamic webpages. A ***dynamic webpage*** is a webpage that is generated on the web server when requested, typically personalized to the user who requested the page. With advances in web browsers, developers have begun creating static webpages that are dynamically altered by JavaScript. In this new paradigm, server-side technologies are used primarily to respond to Ajax requests and send data to the front-end for rendering.

Single Page Applications are an example of modern web development. A ***Single Page Application (SPA)*** is a web application that provides a similar user experience as a desktop application, all in a single webpage. Ex: Gmail, Google Docs, and Google Calendar are all SPAs. An SPA initially loads all of the application's resources so subsequent user interaction results in loading small pieces of content dynamically. Much of an SPA's programming logic is written in JavaScript, which loads data via Ajax calls to a web API. A ***web API*** is a collection of functions that are invoked using HTTP. Ex: An HTTP GET request to the URL https://linkedin.com/api/contacts may retrieve a list of all contacts from the web server.

PARTICIPATION ACTIVITY 11.1.4: Cool Contacts SPA.

1 2 3 4 5 6 7 2x speed



Web server requests Becky's details from the database and sends the details back to the web ap

displaying.

Captions

1. Initial request for CoolContacts web app sent to the web server.
2. All resources needed for app are downloaded in multiple request-response messages.
3. JavaScript uses web API to request all contacts.
4. All contacts are retrieved from the database and sent back to the web app for displaying.
5. User selects a contact from the web app.
6. JavaScript uses web API to request details for selected contact.
7. Web server requests Becky's details from the database and sends the details back to the web app for displaying.

Feedback?

PARTICIPATION ACTIVITY 11.1.5: Server-side programming platforms.

1) A dynamic webpage might look different for two different users who are accessing the same page.

- (•) True
- ( ) False

Correct

Dynamic webpages are often personalized to the user who requested the page.

2) The business logic of an SPA should generally be encoded in the front-end.

- ( ) True
- (•) False

Correct

A well-designed SPA implements as little business logic in the front-end as possible. The web API should implement the business logic.

3) SPAs generally result in less data being sent over the network than web applications developed with dynamically generated webpages.

- (•) True
- ( ) False

Correct

SPAs use web APIs to pass data between the web browser and web server, which are generally more network efficient.

4) Developers use ASP.NET, Java, PHP,

Correct

Python, Node.js, and Ruby on Rails to create web APIs.

- True
- False

All server-side programming platforms or languages have libraries and frameworks that allow developers to create web APIs.

Feedback?

Databases

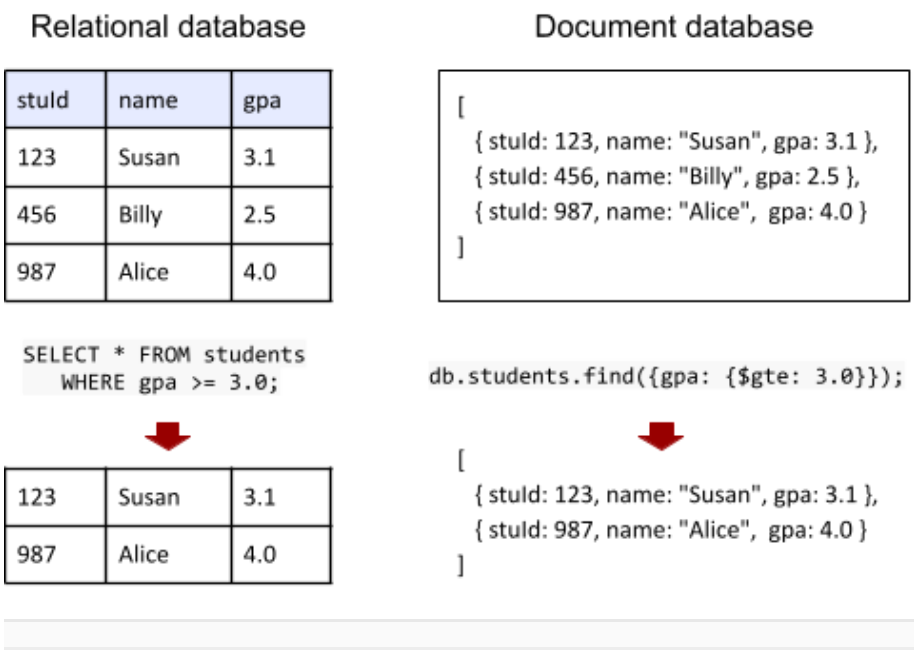
Websites and web applications normally store and retrieve information from a database and have traditionally used relational databases. A ***relational database*** stores data in relations (usually called tables). The ***Structured Query Language (SQL)*** is a language for creating, editing, selecting, and deleting data in a relational database. Popular relational database management systems (RDBMS) include MySQL, PostgreSQL, Oracle, and SQL Server.

Non-relational databases, sometimes called ***non-SQL*** or ***NoSQL*** databases, have become increasingly popular over the last few years. Non-relational databases use different methods to store and retrieve data using a variety of data access languages. Non-relational databases come in several flavors:

- Document database: For storing documents in JSON format with many levels of nesting. Ex: MongoDB.
- Key-value database: For storing values that are associated with unique keys. Ex: Redis.
- Object database: For storing objects created in object-oriented programming languages. Ex: Caché.
- Column database: For storing and processing large amounts of data using pointers that link to columns distributed over a cluster. Ex: HBase.
- Graph database: For storing graph structures with nodes and edges. Ex: Neo4j.

The figure below illustrates how information about students might be stored in a relational database with a table versus a document database using JSON-like documents. The "SELECT" statement is an SQL statement used to extract students with a 3.0 GPA or above from the table. The "db.students.find" statement is a MongoDB function used to extract the same information from the document database.

Figure 11.1.2: Relational database vs. document database for student data.



Feedback?

PARTICIPATION ACTIVITY 11.1.6: Databases.

1) Relational databases will likely not be used for many web applications in the future.
   - ○ True
   - ◉ False

Correct

The popularity of relational databases and relative ease of using SQL ensures that relational databases will not go away soon.

2) A relational database can be used to store documents, objects, graphs, and key-value pairs.
   - ◉ True
   - ○ False

Correct

Relational databases are very flexible and can store a wide range of data. Some developers prefer non-relational databases for documents, objects, graphs, and key-value pairs because the programming required to manipulate such data is often decreased, and in some cases non-relational databases are faster.

3) Column databases are generally faster than relational databases for accessing vast amounts of data.
   - ◉ True
   - ○ False

Correct

Column databases excel at certain types of queries, such as finding aggregations like totals and averages. However, column databases are slow when inserting a single row.

4) Both relational and non-relational databases have been implemented with open source software.

- (●) True
- ( ) False

Correct

MySQL and PostgreSQL are popular open-source relational databases. MongoDB and Cassandra are popular open-source non-relational databases.

Feedback?

Client-side technologies

The user interface (UI) governs the interaction between users and web applications. Developers use HTML, CSS, and JavaScript to create the UI. Various tools exist to aid UI development:

- An ***HTML preprocessor*** is a program that converts a markup language into HTML. The markup languages supported by HTML preprocessors are generally easier to use and read than HTML. Ex: Haml, Markdown, Slim, Pug.
- A ***CSS preprocessor*** is a program that converts a CSS-like language into CSS. CSS-like languages simplify the development of CSS stylesheets used in large projects. Ex: Sass, Less, Stylus.
- A ***UI library*** is a library that creates UI widgets like sliders, dialog boxes, and drop-downs or simplify DOM manipulation. Ex: jQuery UI, Bootstrap, YUI, Ext JS. Libraries like React and Vue.js support more extensive UI management.
- A ***CSS front-end framework*** is a framework that uses CSS or CSS pre-processors to aid in developing responsive websites that work well on every screen size. Ex: Bootstrap, YAML 4, Skeleton, Foundation.

Most modern web applications use an extensive amount of JavaScript, so developers use various tools to aid in JavaScript development:

- A ***compile-to-JavaScript language*** is a programming language that is compiled into JavaScript. Compile-to-JavaScript languages provide benefits lacking in JavaScript like type safety, simplified class creation, and module creation. Ex: TypeScript, CoffeeScript, and Haxe.
- A ***JavaScript framework*** is a JavaScript environment that dictates the organization of the application's JavaScript to simplify many programming tasks. JavaScript frameworks often dictate how UI widgets receive data or send data to the web server. Ex: AngularJS, Backbone, Ember.

Figure 11.1.3: Example use of HTML and CSS preprocessors and compile-to-JavaScript.

Haml	Resulting HTML

nav
  ul
  li
  a href='/home'
Home
  li
  a href='/about'
About
  li
  a href='/sales'
Sales
	<nav>
	
	  Home
	
	  About
	
	  Sales
	
	</nav>

Less	Resulting CSS

@nice-green: #097911;
@light-green: @nice-green + #111;
 	
header {
color: @light-green;
.logo {
width: 250px;
}
}
	header {
	color: #1a8a22;
	}
	
	header .logo {
	width: 250px;
	}

CoffeeScript	Resulting JavaScript

math =
root: Math.sqrt
square: (x) -> x * x
	math = {
	root: Math.sqrt,
	square: function(x) {
	return x * x;
	}
	};

Feedback?

PARTICIPATION ACTIVITY 11.1.7: Client-side technologies.

1) Haml code can be rendered directly in a web browser.

- True
- False

Correct

Haml must first be converted by an HTML preprocessor into HTML, which is then rendered in a web browser.

2) CSS preprocessors allow developers to write much less code compared to writing straight CSS.

- (•) True
- ( ) False

Correct

CSS preprocessors convert special syntax into more verbose CSS rules.

3) UI libraries always use JavaScript to govern the behavior of the UI widgets.

- ( ) True
- (•) False

Correct

Some widgets require JavaScript, but many use special HTML tags to govern the widget behavior.

4) CSS front-end frameworks are required to build responsive websites that work well on mobile devices.

- ( ) True
- (•) False

Correct

CSS front-end frameworks simplify the creation of responsive websites, but some developers write their own CSS code.

5) TypeScript code is executed by the web browser.

- ( ) True
- (•) False

Correct

TypeScript is compiled into JavaScript by the developer, and the compiled JavaScript is executed by the web browser. JavaScript is the only native language that all major browsers are capable of executing.

6) JavaScript frameworks often simplify the use of web APIs in the browser.

- (•) True
- ( ) False

Correct

JavaScript frameworks like AngularJS often provide built-in functionality for working with web APIs.

Feedback?

Testing

Developers must test the full technology stack used by web applications. A variety of testing frameworks exist to automate the testing of web applications. Ex: Selenium is used to

automate a test user's interaction with a web application and verify that the UI behaves correctly.

PARTICIPATION ACTIVITY | 11.1.8: Testing a web application.

If unable to drag and drop, refresh the page.

Functionality testing	Verifying that each individual application function is working as expected. Functional testing includes verifying all links resolve properly, testing data validation on form inputs, and verifying the database is storing the correct information.	**Correct**
Interface testing	Testing the interaction between the front-end and back-end and the interactions between the server-side programs and the database. Each component must handle error conditions properly and immediately alert developers to any problems.	**Correct**
Usability testing	Testing the user's ability to properly use the web application for specific purposes. Usability testing includes checking the content for correctness and ensuring the application is accessible to all users.	**Correct**
Compatibility testing	Testing the web application's ability to work on various browsers, operating systems, and platforms. Developers often use third-party libraries that aid in developing	**Correct**

	code that runs properly on all platforms.	
Performance testing	Verifying the web server is able to respond reasonably under various load conditions. Performance testing involves pushing the back-end to capacity and ensuring the system is able to withstand high loads.	**Correct**
Security testing	Ensuring the integrity and privacy of the user's data and interactions with the web application. Security testing involves looking for ways that unauthorized users may access the user's data. Security breaches can have drastic consequences for users and organizations.	**Correct**

Reset

Feedback?

Exploring further:

- Ranking of database systems
- 6 Current Options for CSS Preprocessors
- Best languages that compile to JavaScript
- Top JavaScript Frameworks, Libraries and Tools and When to Use Them
- Summary of web application testing methodologies and tools

How was this section? **Provide section feedback**