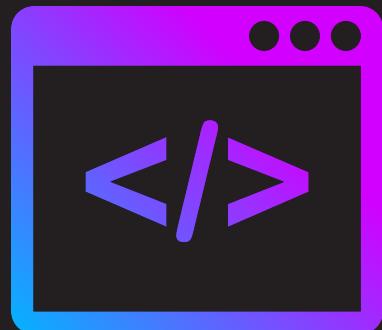


Introduction to Web Development



Learning Objectives



- Discuss web development and some statistics
- Discuss the scope and power of full stack development
- Explain the history of the World Wide Web
- Explain how does the World Wide Web work
- Describe the history of the web browser
- Explain client-server and request-response cycle

Learning Objectives

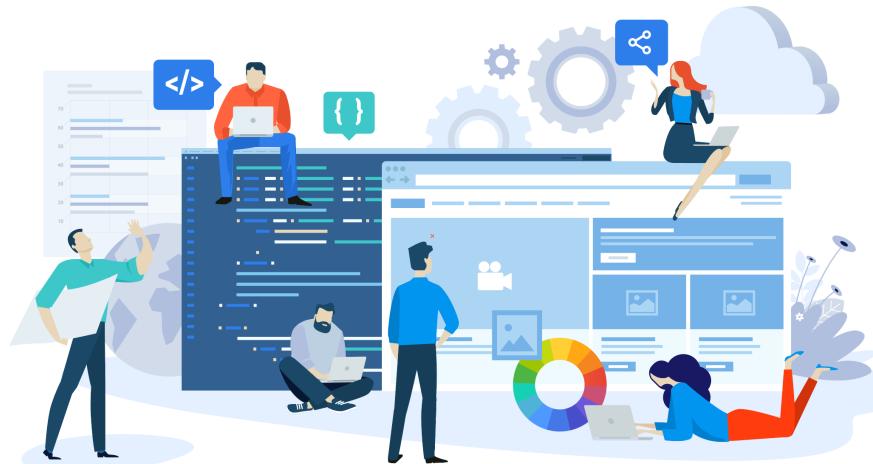


- Describe the evolution from web sites to mobile apps
- Discuss how does a website work
- Explain frontend and backend technologies
- Discuss stack with examples
- Discuss MERN stack
- Explain algorithms and pseudocode
- Illustrate the pseudocode exercises

What is Web Development ?

Creating a web-based application that helps solve a problem. This could be a product or a service.

Includes: web design, web development, database management, deployment, etc.



What is Web Development ?



Web development does not necessarily refer to developing a static website; rather, it deals with providing dynamic solutions to any industry problem.

Learning Management Solutions (LMS) provides online courses for upskilling and learning. Hence, solving the problem of learning anytime, anywhere!

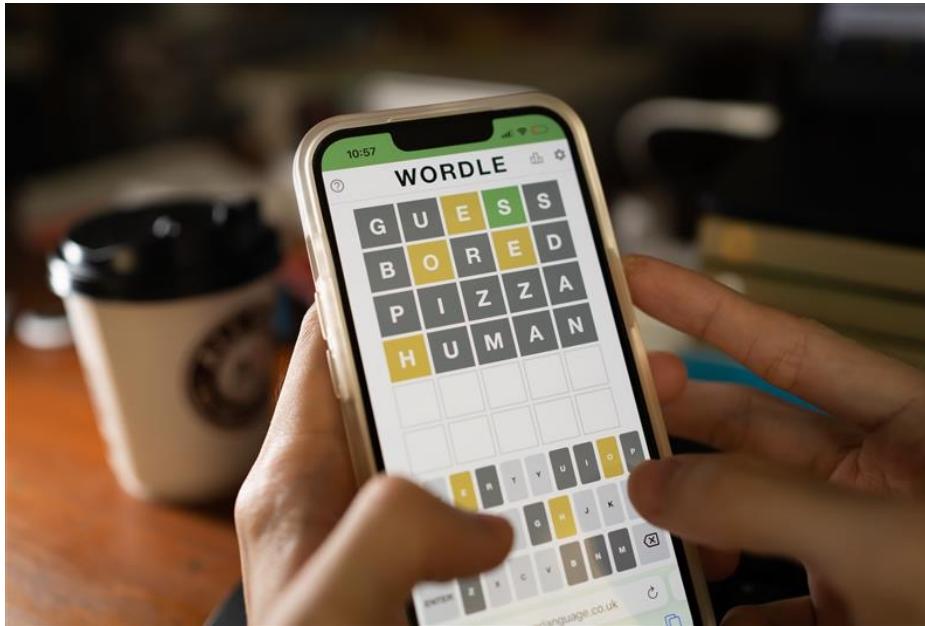


Customer Resource Management systems (CRM) make it easy for companies to manage customers and their requirements. Thus, solving the problem of relationship management in a structured manner.



Power of the Web

Even a simple game designed by a developer for personal use can turn out to be a huge success and make a huge difference.



Power of the Web

- Wordle is a word game initially developed by a web developer Josh Wardle, in Brooklyn for his wife.
- Wardle made the game public in October 2021, and it became viral in December 2021.
- In January 2022, the game was purchased by "The New York Times" company.



Growth in Web and Web Development

"Web development has a wide reach with the booming IT industry not only in India but also abroad."

- Internet users worldwide comprise more than 5.4 billion ([Source](#)).
- There are more than 1.9 billion websites worldwide ([Source](#)).
- 62% of companies increased their sales by designing responsive mobile platforms for their websites, according to SAG ipl.
- 68% of the companies that developed a website after using it as a mobile app saw a rise in sales, according to SAG ipl.



5,471,427,466

Internet Users in the world



1,990,778,803

Total number of Websites

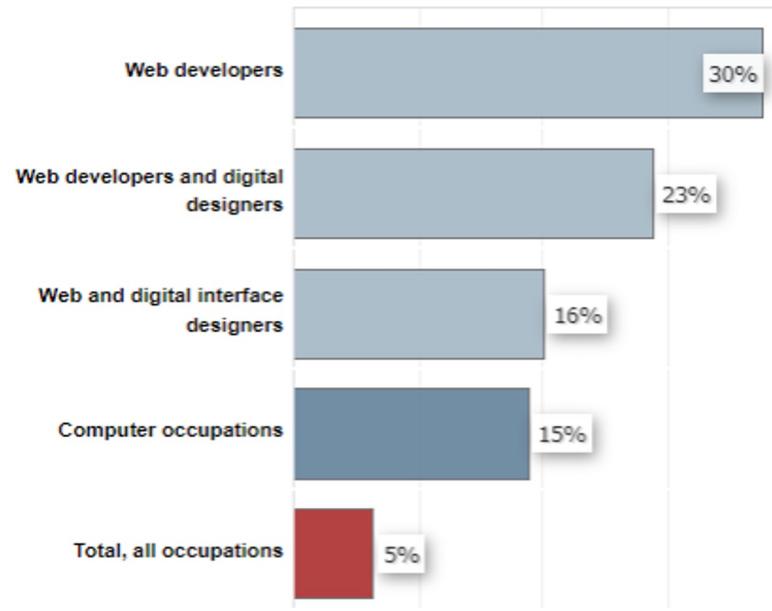


223,651,708,502

Emails sent today

Growth in Web and Web Development

Percent change in employment, projected 2021-31

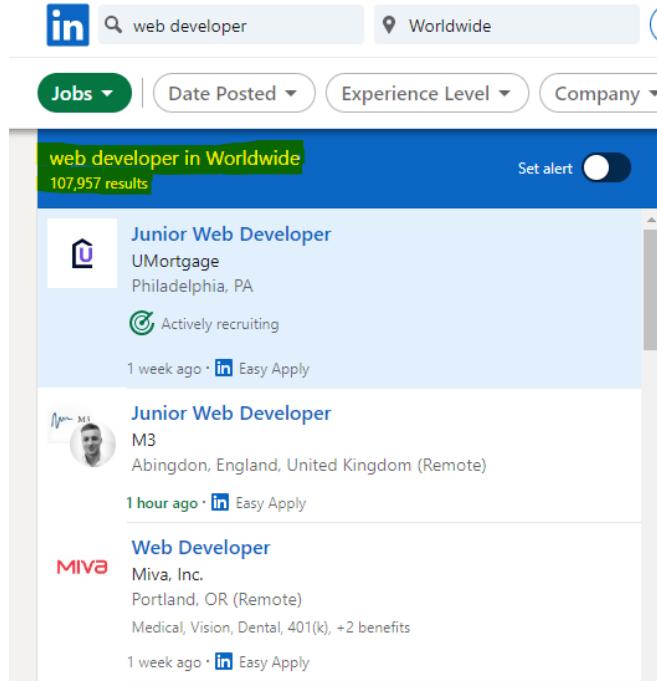


Note: All Occupations includes all occupations in the U.S. Economy.

Source: U.S. Bureau of Labor Statistics, Employment Projections program

Jobs for a Web developer

LinkedIn October 2022 Stats: More than 100 thousand jobs for web developers worldwide.



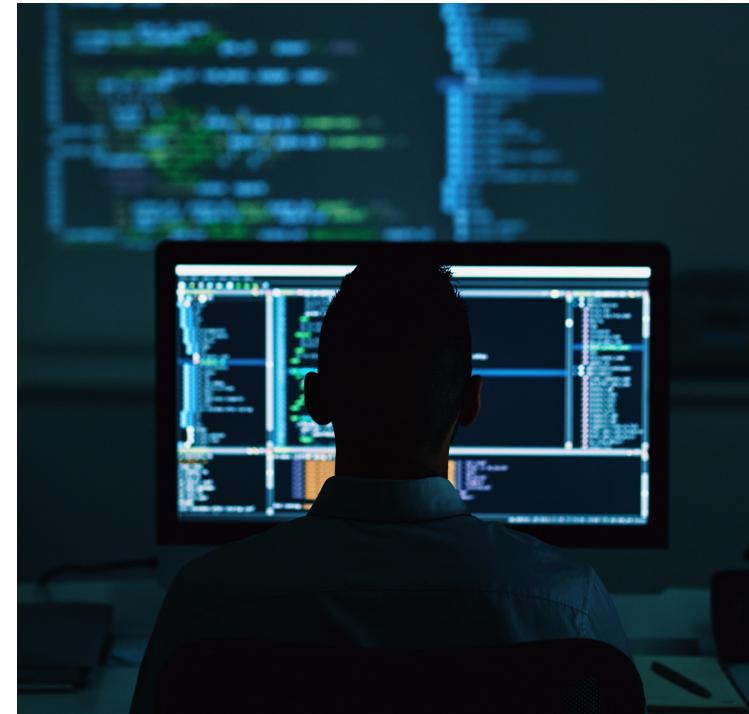
The screenshot shows the LinkedIn search interface with the following details:

- Search bar: web developer, Worldwide
- Filter buttons: Jobs (selected), Date Posted, Experience Level, Company
- Search results summary: web developer in Worldwide, 107,957 results
- First result: Junior Web Developer at UMortgage, Philadelphia, PA. Status: Actively recruiting. Posted 1 week ago, Easy Apply.
- Second result: Junior Web Developer at M3, Abingdon, England, United Kingdom (Remote). Status: 1 hour ago, Easy Apply.
- Third result: Web Developer at Miva, Inc., Portland, OR (Remote). Benefits: Medical, Vision, Dental, 401(k), +2 benefits. Posted 1 week ago, Easy Apply.

A person who creates a web-based application based on a business requirement.

Job Responsibilities

1. Website or webapp development
2. Analysis
3. Research
4. Development
5. Core implementation
6. Maintenance
7. Testing



Opportunities for Web Developers

Opportunities and skills which are highly in demand:

- Frontend developer
- Backend developer
- Full stack developer
- Web designer
- UX designer
- Architect
- Project manager



Opportunities for Full Stack Developer

Full-stack developers are needed in the following areas for creating various products:

- Blockchain
- Internet of Things (IoT)
- Augmented Reality
- Virtual Reality
- Chatbots
- Low Code Development



World Wide Web (WWW)

- It was **developed for the scientists** of different universities and institutes to share information automatically worldwide.
- www or web has a **collection of websites and webpages** containing information like hyperlinks and hypertexts.
- Anybody, anywhere, anytime in the world, can access the web to gather information on any topic.
- Information can be in the form of electronic images, text, sounds, animations, and movies.

History of WWW

- WWW is a project at the European Particle Physics Laboratory called CERN.
- Tim Berners-Lee, a British Scientist, published World Wide Web in 1991.
- In 1993, CERN declared that WWW would be available for free to be used by anyone in the world. They didn't copyright it in their name.



How Does WWW Work?

- It is based on several technologies that help make information available to people
- Information and services can be accessed by users using web browsers
- The user end is known as a *client* which makes requests to *servers* where web content and applications are stored
- These *requests* travel over the Internet using communication protocols such as TCP/IP. The language used to talk is HTTP or HyperText Transfer Protocol. In a nutshell, an HTTP request is sent over the Internet using the TCP/IP protocol to the server.
- The *server* intercepts and processes the *request* and sends back a *response*. This travels in the same fashion as a *request*.
- At the *client*, the *response* is received and shown to the user as content such as a webpage, video, audio etc.

Client and Server



Servers



Clients

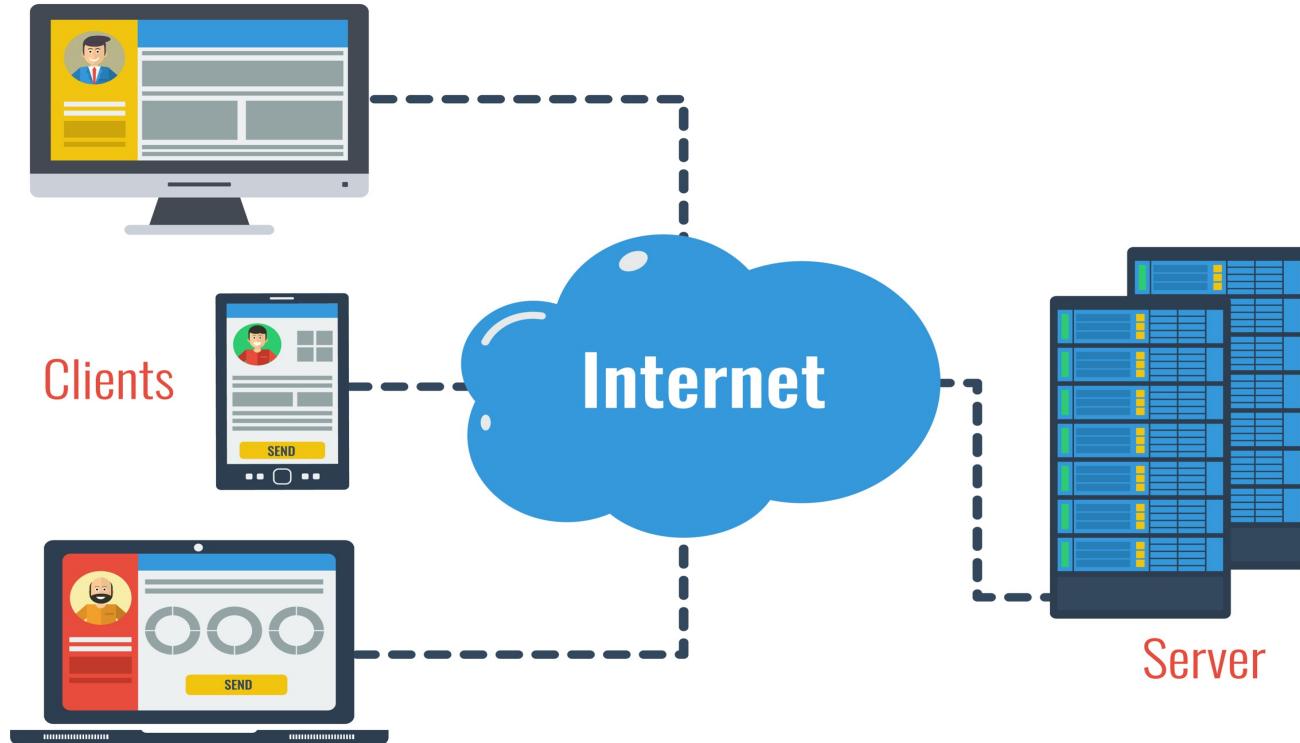
A Server is simply a service provider who services a request and returns a response

Clients request data and services from the servers

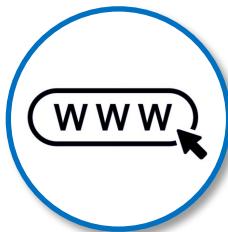
Types of Servers

- Application Server
- Cloud Server
- Database Server
- Dedicated Server
- Domain Name Service
- File Server
- Mail Server
- Print Server
- Proxy Server
- Web Server

Client-Server Model



Let us type a URL and hit enter!



Type URL
in the
browser



Browser
checks DNS



Request - Response
cycle

- URL (Uniform Resource Locator) is the address of the website that you are looking for on the internet. For example: <https://www.knowledgehut.com/>
- DNS (Domain Name Server) is like the address book of the internet. It maps the URL with the I.P. address of the host of that URL.

Request - Response Cycle



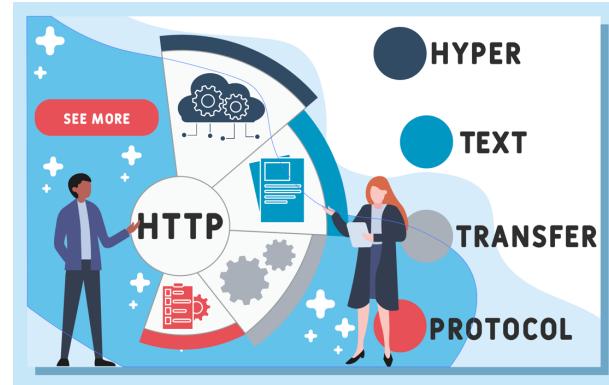
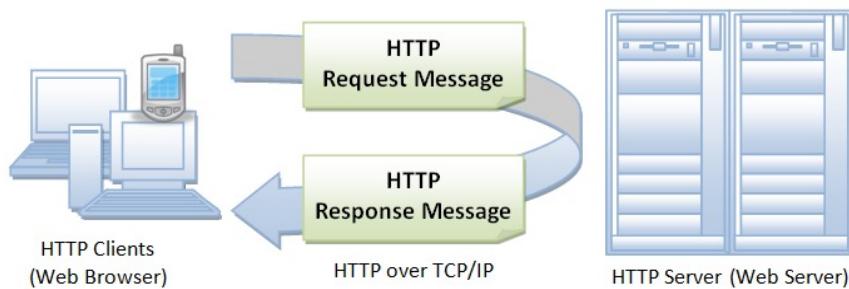
When a client (your computer) requests a web page, that information is processed by the web server (responder). If the request is a server-side script before the information is returned to the client, the script is executed on the server, and the result of the script is returned to the client.



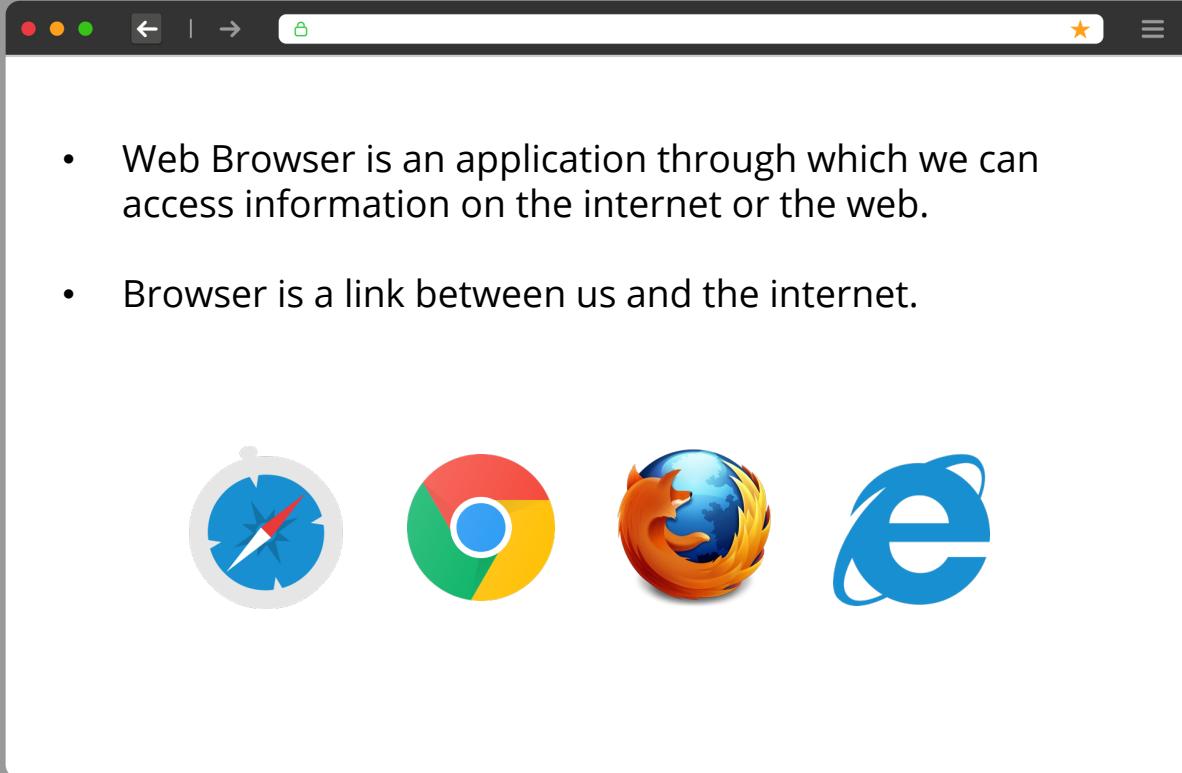
Once the client receives the returned information from the server, if it contains a client-side script (Ex: JavaScript), your computer browser executes that script before displaying the web page.

Request - Response Cycle

HTTP provides rules to define the exchange of information in the request-response cycle.



The Web Browser - History and Workings



The image shows a screenshot of a web browser window with a light gray header bar containing standard icons for window control, navigation, and search. The main content area displays two bulleted points:

- Web Browser is an application through which we can access information on the internet or the web.
- Browser is a link between us and the internet.



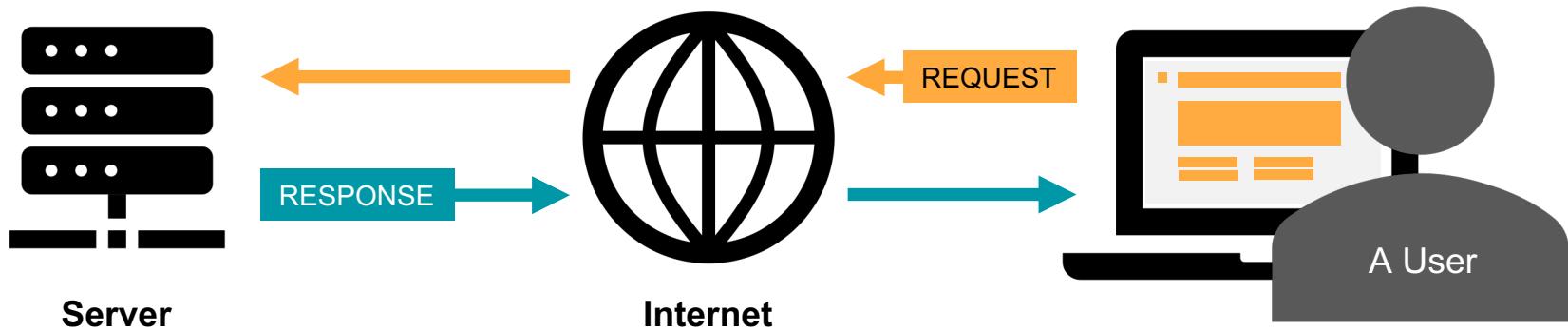
Below the text, there are four circular logos representing different web browsers: Safari (blue compass rose), Chrome (red, green, and yellow segments), Firefox (blue globe with orange flames), and Internet Explorer (blue 'e' with a blue circle).

The Web Browser - History and Workings

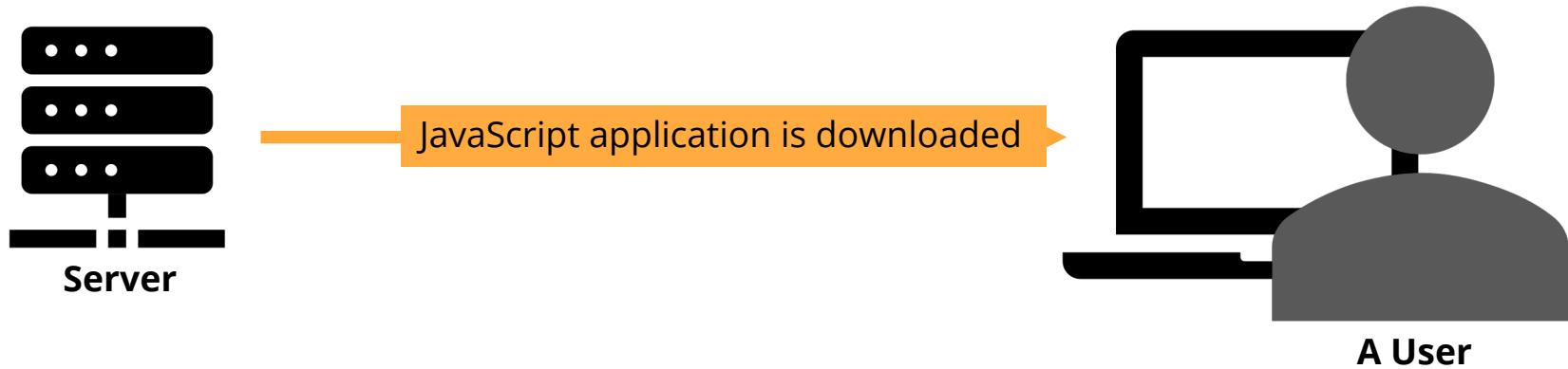
- 1990 Worldwide Web, renamed to Nexus
- 1994 Netscape
- 1995 Microsoft's Internet Explorer
- 1996 Opera
- 2003 Apple Safari
- 2004 Mozilla
- 2007 Mobile Safari ➡ Very successful browser was introduced for Apple Phones
- 2008 Google Chrome ➡ Soon took over all the browsers till date
- 2011 Opera Mini ➡ Made for Android Mobile
- 2015 Microsoft Edge ➡ Designed in competition to Google Chrome

How do Websites work?

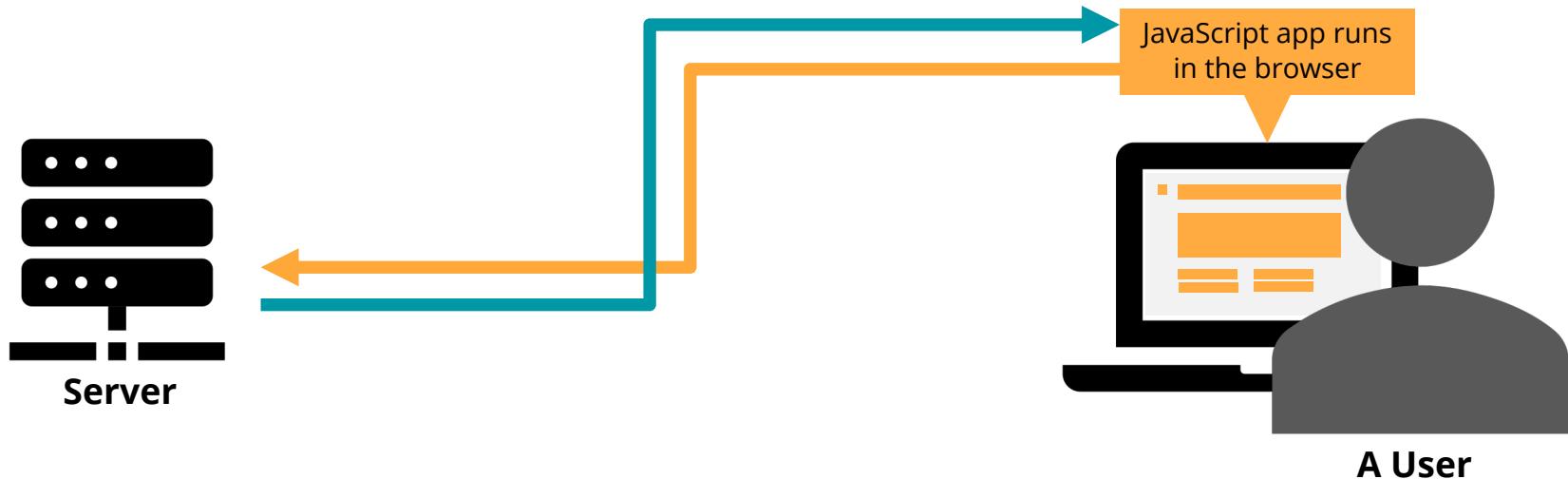
Request gets a Response



Modern Websites - Think Gmail, Office 365



Modern Websites - Think Gmail, Office 365



WEB APPLICATIONS

Built using HTML5, CSS3 and JavaScript



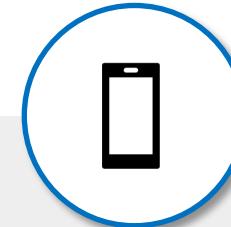
Website

- A collection of web pages having information in the form of images, text, videos, etc.
- Website has to be published on at least one web browser to get access.
- A website has to have a domain name.
- Example: amazon.com, google.com, etc.



Web Applications

Like Spreadsheets, Google documents, Gmail, Photo editing apps etc., are examples of web applications which can run on a web server.



Mobile Application

- An application designed to run on a mobile phone.
- We can say it's a mobile view of a website.
- Example: amazon.com, google.com, etc., have their websites and have mobile apps also.

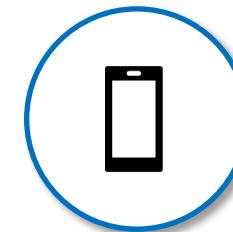
From Websites to Web Applications to Mobile Apps



Website



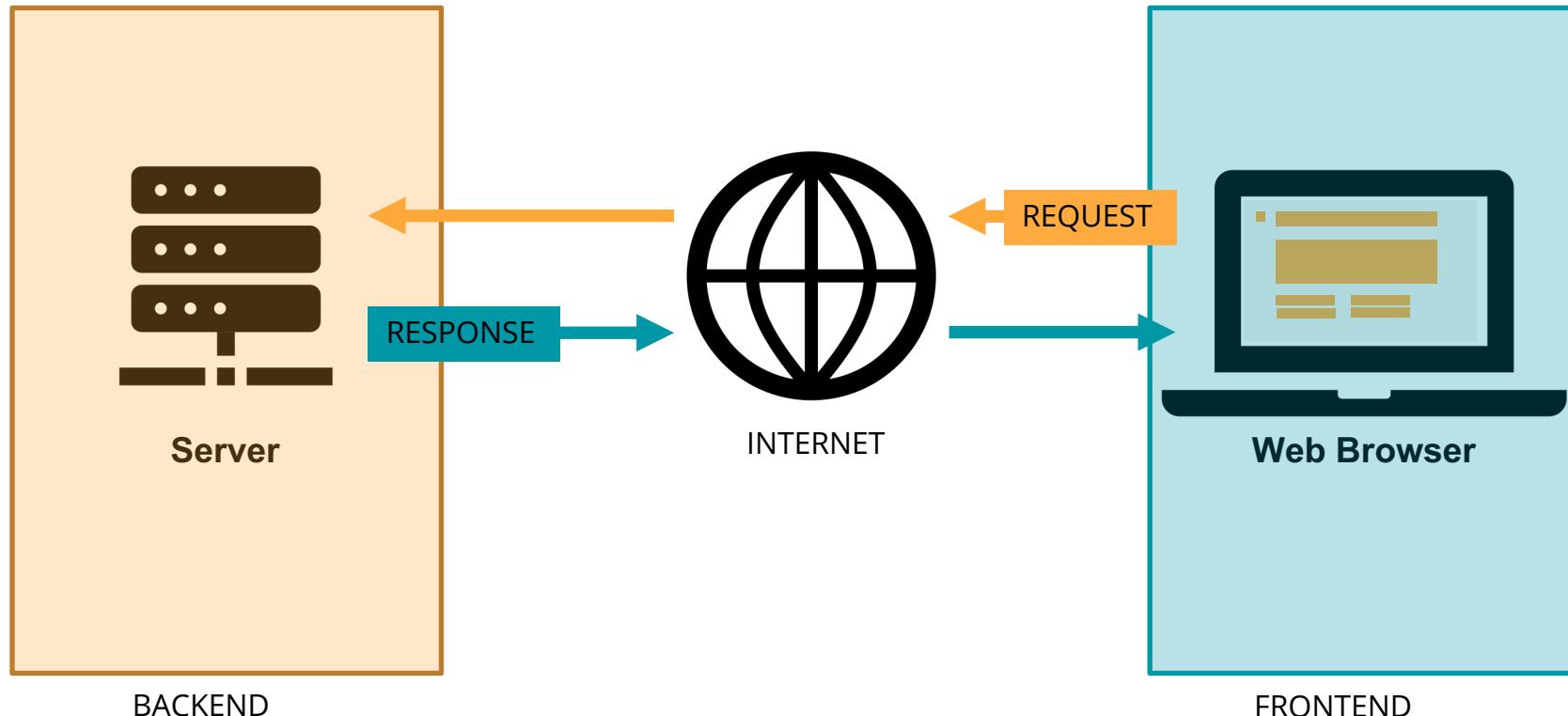
Web Applications



Mobile Application

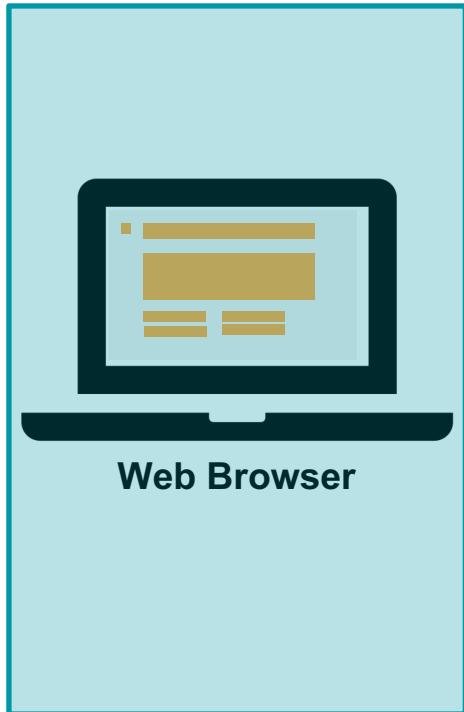
- Companies have shifted from websites to web applications to mobile apps to increase accessibility among people.
- This evolution leads to more business and increased customer satisfaction.
- All the above things co-exist and are used by their required user as per their comfort.

Two Sides of a Web Application



Understanding Technologies: Frontend

The Frontend - Your Web Browser



FRONTEND

HTML5

HyperText Markup Language

Used for describing the structure and contents of a webpage.

CSS3

Cascading Style Sheets

Used for describing the presentation & visual look and feel of a webpage.

JavaScript (ECMAScript®)

Is Not Java

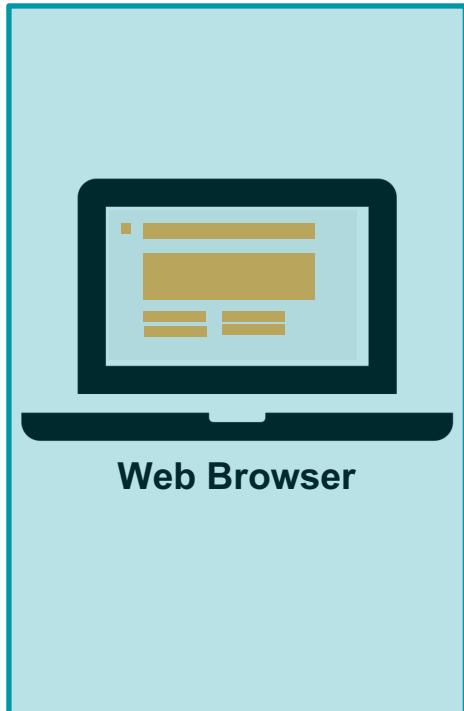
Used for adding logic, interactivity, and a lot more.

The Frontend - Introducing Libraries and Framework



- Libraries and frameworks make it easy for developers to build sophisticated JavaScript web applications.
- These frameworks ensure that your app offers great performance.
- They are basically like tools written using JavaScript.

The Frontend - Angular



FRONTEND

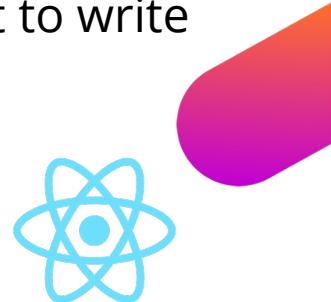
- Angular is a framework for building JavaScript apps by Google.
- One of the most popular frameworks, especially in Asia.
- You can build: Web applications, Mobile Apps (Using the iconic framework) and even desktop apps.
- Developers are strongly advised to learn TypeScript before learning Angular.



The Frontend - React



- React is a library for building user interfaces and web applications. Made by Facebook.
- Globally, the most popular library for building high-performance JavaScript applications.
- You can build: Web applications, Mobile apps (Using React Native) and even desktop apps.
- Developers can use JavaScript or TypeScript to write applications.



The Frontend - Vue

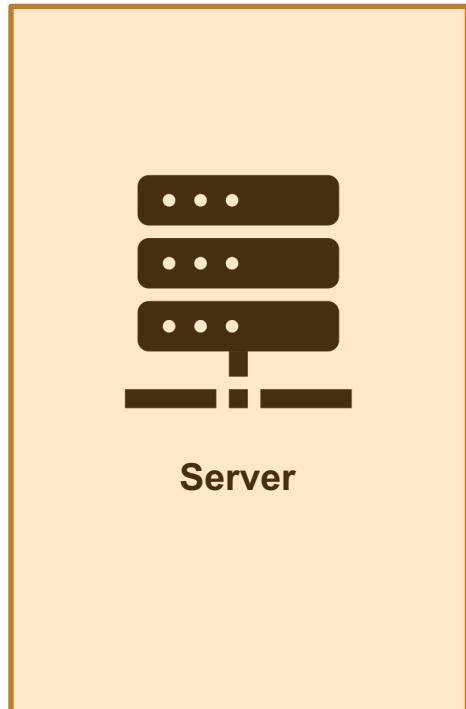


- Vue is a library for building user interfaces and applications. Made by Evan You.
- Very popular framework and easy to learn & use.
- You can build: Web applications and even desktop apps
- Developers can use JavaScript or TypeScript for writing applications.

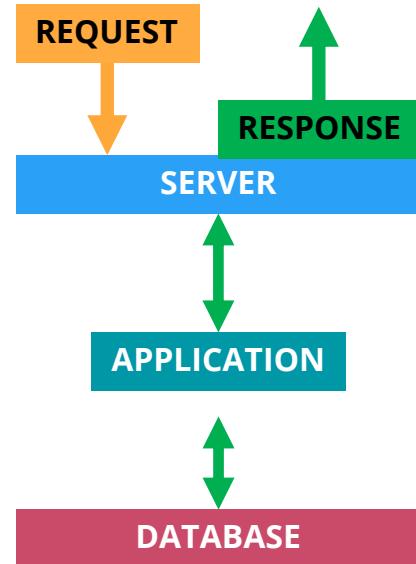


Understanding Technologies: Backend

The Backend - Components



BACKEND

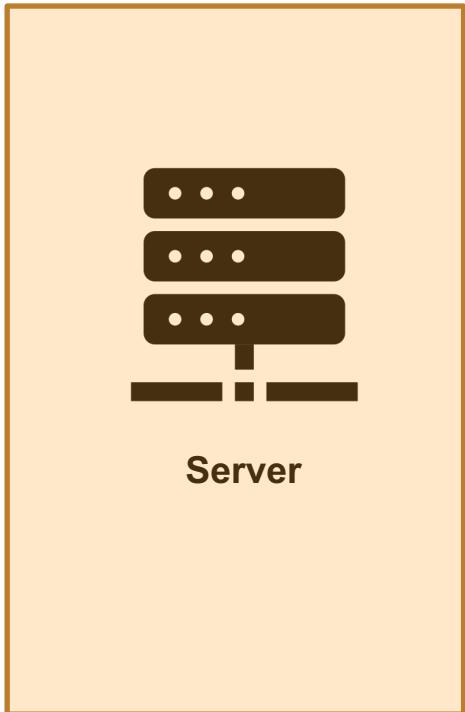


Listens to requests and serves responses

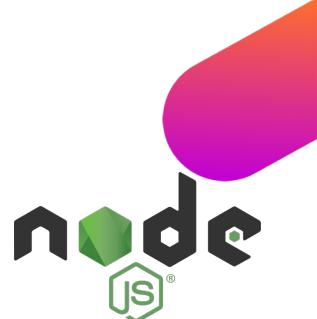
The brain of the back end

Stores all your data

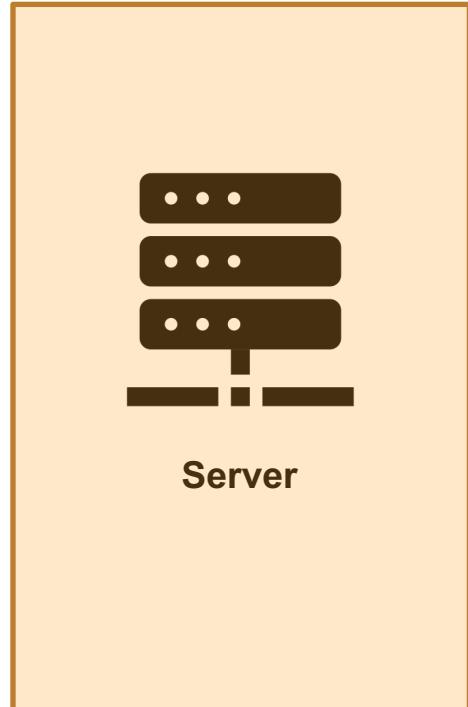
The Backend - Server and Application



- Node.js allows you to write software that runs on a server using JavaScript.
- Developers who build front-end applications using React, Angular, etc., can also learn to write Node.js apps that work at the back end.
- You use Node.js to build your server and application code.
- You write both codes using one common language – JavaScript frontend and backend.



The Backend - Server and Application

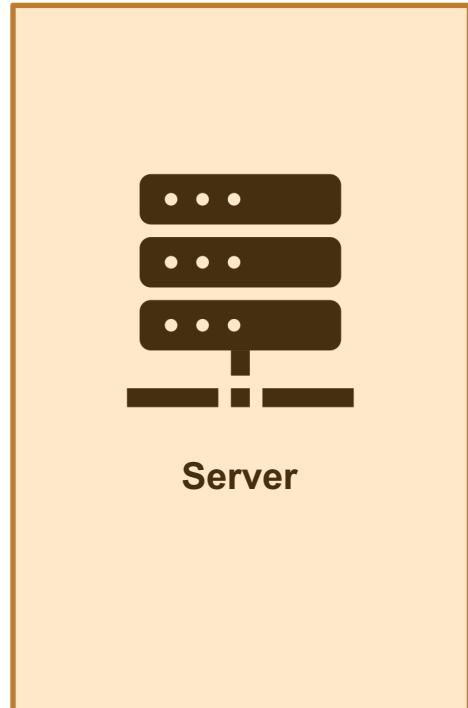


BACKEND

- Express is a framework that allows you to easily build Node.js applications.
- You can build websites, backend for React/Angular applications, mobile apps, and lots more.
- Easy to learn.
- Massively speeds up building apps with Node.js.
- Used by a ton of big companies and enterprises.



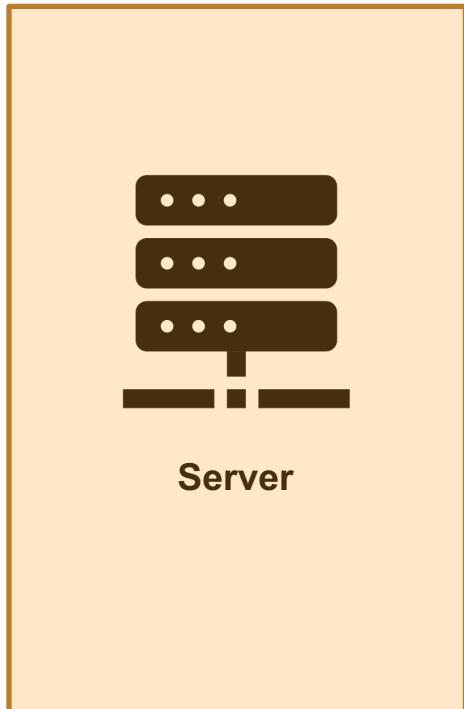
The Backend - Server and Application



- Java is a language for building computer software.
- Spring Framework & Spring Boot make it really easy for Java developers to build backend applications.
- You can build a backend for React/Angular applications, mobile apps, and lots more.
- Used by big names like Accenture, Cognizant, Intuit, and more.



The Backend - Server and Application



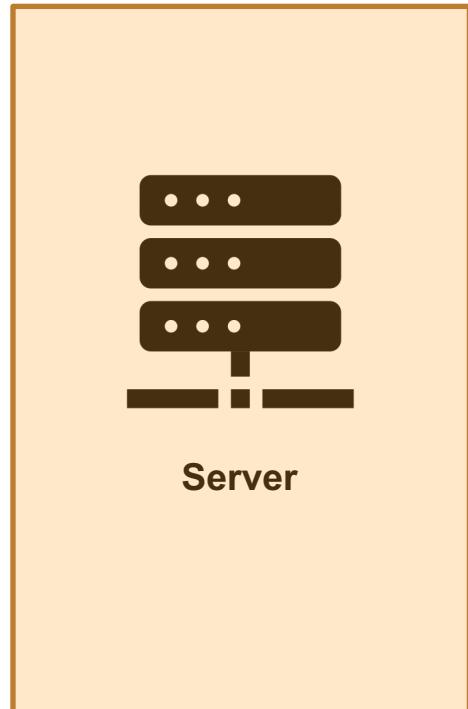
BACK END

- Python is a general-purpose computer language.
- Django framework makes it easy for Python developers to build websites.
- The Flask framework allows developers to build backend apps for web applications made using React/Angular etc., as well as for mobile apps.

 python™ + **django** or 



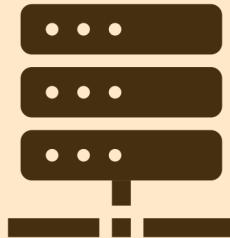
The Backend - Server and Application



- Ruby is a general-purpose computer language.
- Ruby is also used for building web applications and has been quite popular in the West.
- Ruby on Rails is a framework that makes it easy for Ruby developers to build web applications and backend services quickly.



The Backend - Server and Application



Server

BACK END

- PHP is a language for building websites and back-end applications.
- You need Apache or Nginx web server to run PHP.
- Laravel is a popular framework for rapidly building PHP websites.



The Backend - Database

A database stores data such as bank records, user accounts, social media posts, blog posts, etc.



What is a Stack?

What is a Stack?

A **Stack** is a group of technologies and products that work great as a team.



A Stack - Think of a Recipe

Spaghetti Aglio e Olio



+



=



Spaghetti Aglio e Olio



+



+



+



=



CALAVITA

FIGARO

HARDNECK

NATURESMITH

Spaghetti Aglio e Olio



AGNESI



+



+



+

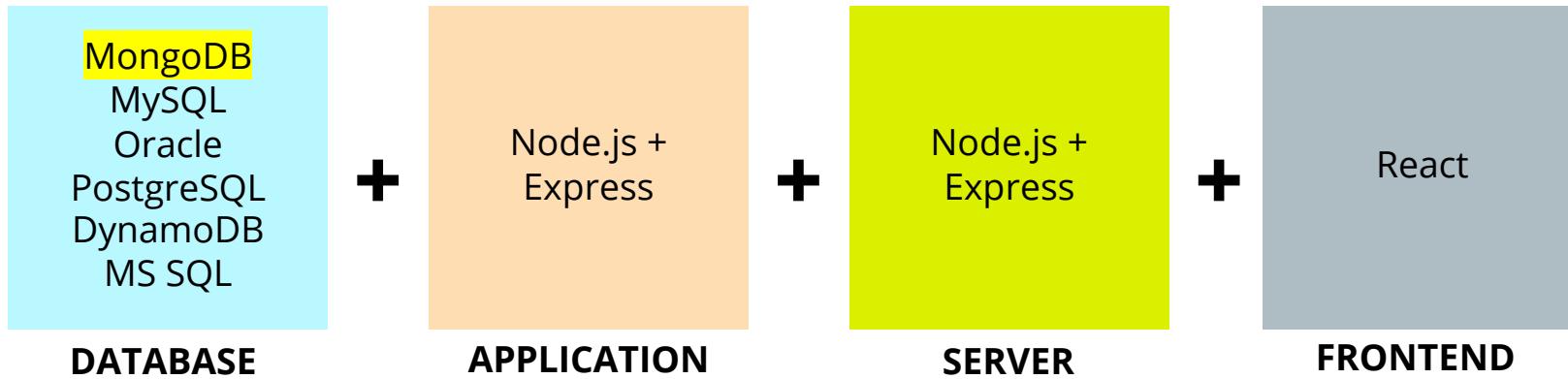
=



SOFTNECK

EASY LIFE

A Stack - Think of a Recipe



Who are Full Stack Developers?

Full Stack Developers are involved in both the frontend development and the backend development.

They build software that runs in your web browser as well as on a server.



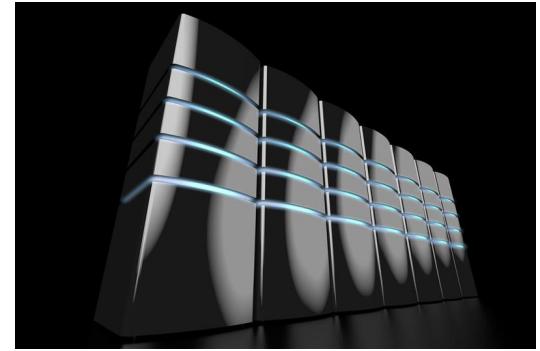
Server-Side Rendered Application (SSR) vs. Single Page Application (SPA)

In SSR:

- The user types any URL on the web browser.
- The web server interprets the user's **request** and HTML based **response** is generated.
- With each URL click, a new HTML page is sent to the client.
- A new Request – Response cycle takes place.

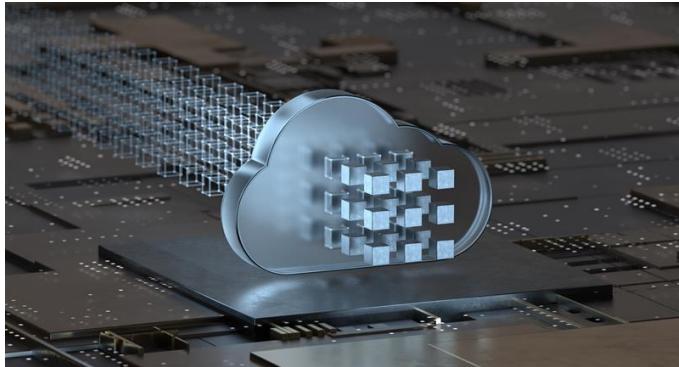
In SPA:

- The user types any URL. For the first time, request-response cycle generates the HTML response for the client's web browser.
- When other events like page sleep, page refresh, clicking on a link or button on the same page happen, the web browser stays on the same page.
- It updates the response on the same page instead of creating a new HTML page as a response. Hence, the name Single Page Application.



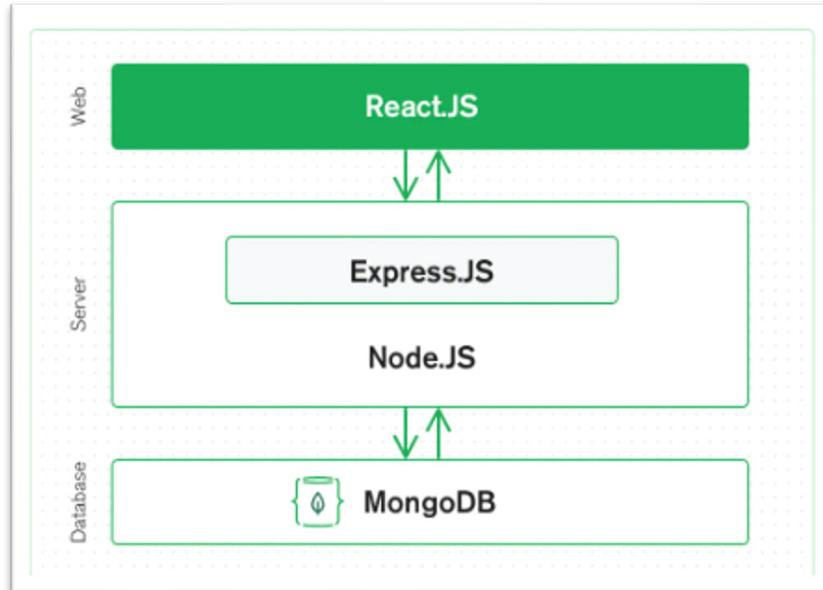
Hybrid Approach

- For large websites, some parts can be pre-rendered, while others keep changing dynamically.
- Modern approaches, like the hybrid approach, help in switching between client-side rendering and server-side rendering whenever needed.
- Server Rendered + Client Rendered applications use React, Angular, Vue.js, on both the client-side and the server-side.
- We will use the MERN stack for developing a web applications using JavaScript.
- MERN refers to MongoDB, ExpressJS, ReactJS, and NodeJS.
- It provides support for frontend as well as a strong backend



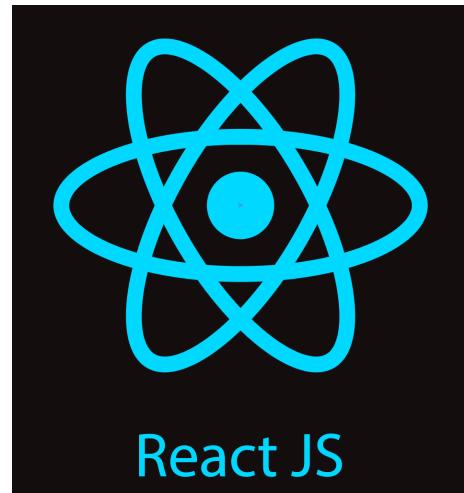
MongoDB and ExpressJS

- MongoDB is a NoSQL document-based database
- MongoDB stores any kind of data like comments, user profiles, content, documents, uploads etc.
- ExpressJS is a framework for rapidly building Node.js apps



express

- React is a JavaScript library that is a collection of functions used to simplify development and introduce powerful features and functionality to a website.
- React was designed by Facebook to **solve common design problems and performance roadblocks.**
- React is used to build interactive user interfaces and create reusable UI components.



Node.js for Backend

Node.js – A JavaScript runtime based on Google's V8 JavaScript engine that is also found in the Chrome Browser.



Algorithms and Pseudocode

Algorithm – Pseudocode

- An algorithm is a sequence of steps the computer needs to achieve a task. The sequence is very important for correct output.
- The purpose of an algorithm is to solve a problem in the most efficient manner. After developing an algorithm for any problem statement, a pseudocode based on that is created next.



ALGORITHM

Algorithm – Pseudocode

- Pseudocode is an artificial language that aims to resemble a real programming language in terms of core features and abilities but gives you ample flexibility to express an algorithm in a way closest to what real code might look like.
- Let us now have a look at some algorithms and pseudocodes.

Consider the case of preparing instant milk coffee in the simplest of steps:

Step 1: Pour a mug full of milk into a saucepan

Step 2: Place the saucepan on the stove/cooktop.

Step 3: Switch on the stove (heat)

Step 4: Pour a spoonful of instant coffee into a coffee mug

Step 5: Check if the milk is hot enough

a. If **YES**

Step 6: Turn off the heat

Step 7: Pour milk into the coffee mug

Step 8: Add sugar to taste

Step 9: Stir till mixed

Step 10: Enjoy

b. If **NO**

Step 6: Go back to Step 5

Let's now see a simple case of displaying even numbers from a list of random numbers

Step 1: Setup a list named L

Step 2: Store 10 random numbers in the list named L

Step 3: Run through every number in the list

Step 4: If the number is even

Step 5: Print the number

Step 6: If the number is odd

Step 7: Continue running through the list

Let us now convert this into a pseudocode.

Pseudocode

Pseudocode: Displaying even numbers from a list of random numbers

```
SET L = array of 10 random numbers
FOR each number in L
    IF number is even
        PRINT number
    ENDIF
ENDFOR
```

Pseudocode

Example

1. Write pseudocode that computes the sum of all even numbers up to a given point and displays it on the screen

```
SET SUM = 0
FOR every number between 1 and N
    IF number is even THEN
        SET SUM = SUM + number
    ENDIF
ENDFOR
PRINT SUM
```

Example

2. Write pseudocode that prints multiples of 7 between 1 and 200

```
FOR every number between 1 and 200
    IF number % 7 is 0 THEN
        PRINT number
    ENDIF
ENDFOR
```

Example

3. Write pseudocode that finds the smallest number from a given Array of 5 numbers.

```
SET numArray = array of 5 given numbers
SET S to the first element of numArray
FOR each number in numArray
    IF S is greater than the number THEN
        SET S = number
    ENDIF
ENDFOR
PRINT S
```

Example

4. Write pseudocode that finds the largest number from a given Array of 5 numbers.

```
SET numArray = array of 5 given numbers
SET L to the first element of numArray
FOR each number in numArray
    IF L is smaller than the number
        SET L = number
    ENDIF
ENDFOR
PRINT L
```

Example

5. Write pseudocode to sort an Array into two Arrays: Odd & Even numbers

```
SET L = array to be sorted
SET EVEN = empty array
SET ODD = empty array
FOR each number in L
    IF L % 2 is 0 THEN
        STORE number in EVEN
    ELSE
        STORE number in ODD
    ENDIF
ENDFOR
PRINT EVEN
PRINT ODD
```

Example

6. Basic Authentication Workflow (username, password) as Pseudocode.

```
GET the username from the user and store as U  
GET the password from the user store as P  
IF U is valid and P is valid THEN  
    DISPLAY 'U is a valid user'  
ELSE IF U is valid and P is invalid THEN  
    DISPLAY 'Password is invalid'  
ELSE  
    DISPLAY 'U is not a valid user'  
ENDIF
```

Example

7. Authentication where a one-time password (OTP) is sent

```
GET the phoneNumber from the user and store as P  
SET OTP as a unique number  
SEND OTP to phoneNumber  
GET the OTP from the user store as userOTP  
IF userOTP is the same as OTP THEN  
    DISPLAY 'User is valid'  
ELSE  
    DISPLAY 'Invalid User'  
ENDIF  
SET OTP as null
```

Example

8. Computing the distance between two points ($[x_1, y_1]$ and $[x_2, y_2]$).

```
SET distance as square root of  $(x_2 - x_1)^2 + (y_2 - y_1)^2$ 
PRINT distance
```

9. Write pseudocode to check if a word is a palindrome or not?

```
GET a word from the user and store as WORD
```

```
SET REVERSED = reverse of WORD
```

```
IF WORD is equal to REVERSED THEN
```

```
    DISPLAY 'Word is a palindrome'
```

```
ELSE
```

```
    DISPLAY 'Word is not a palindrome'
```

```
ENDIF
```

Q&A



Thank You!