7th - May - Day 4

Topics:
Structure Of Web Application

Before we are going to web application we need to know the difference

between **WEB SITE AND WEB APPLICATION & API**

1) **Website :**

What exactly is a Web Site is if you go for any landing page , like

restaurant or textile or any industry they what would they have? They

eventually have

1.home page

2.about page

3.career

4. Contact page

Content on these things will never change. It is like an information

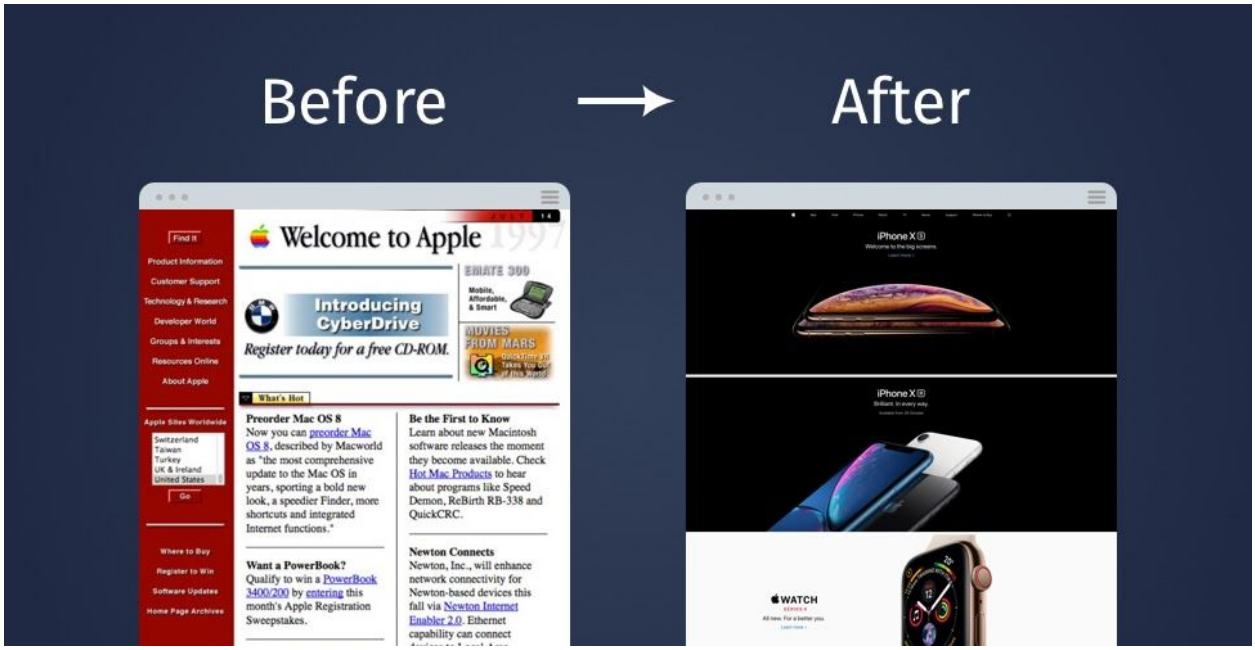
web site.

Static → data stand still means data doesn't change.

The main purpose of the website was to provide information which

later one evolved to provide information . Websites during this classic

HTML era were just static pages with content and media that anyone could access by visiting the URL address. The reason these websites are called static is that the content remains the same each and every time a visitor accesses it. And any changes in the page could be made by adding the content in the HTML On the contrary modern websites use more than just HTML. Languages like CSS and JavaScript are prominently used and even there is a heaven-high and hell-deep of difference in the designs.



Apple website then and now

However, the concept of the website is still the same – *to provide information to the user*

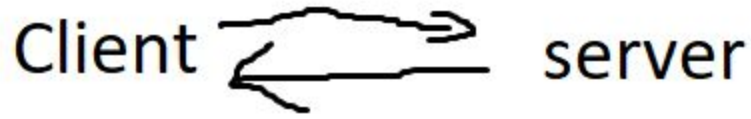
HTML - *HYPERTEXT MARKUP LANGUAGE*

CSS - *CASCADING STYLE SHEETS*

JS - JAVA SCRIPT

HTML is basic structure of web site and css its give beauty to website, & java script which adds functionality of that website

Website workflow :



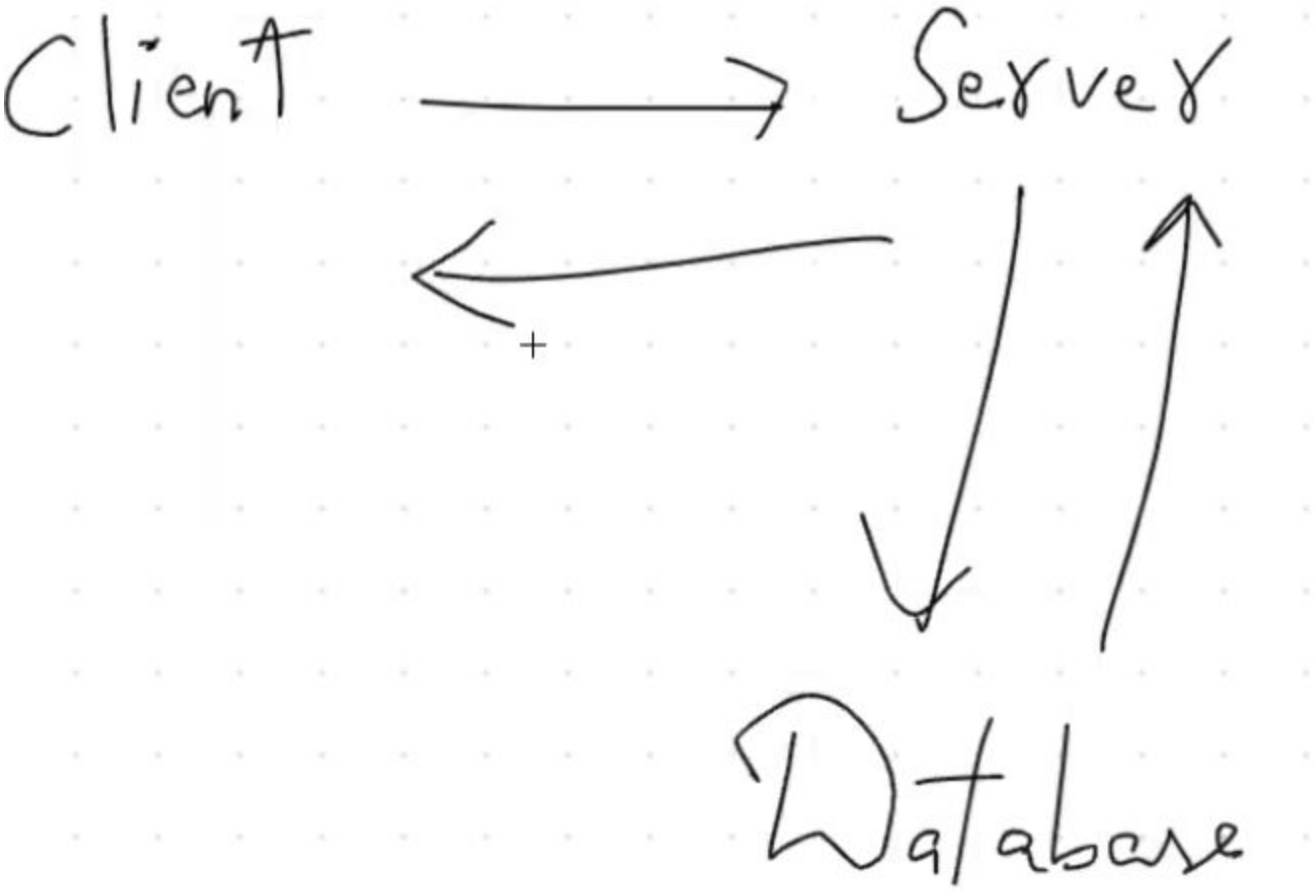
It doesn't require database bcoz its static

2) What is a Web Application?

Traditionally software was installed on personal computers. But the proliferating reach of the internet and perks of web technology bred a new concept of catering software. These software are called web applications. The term “Web Application” itself makes the things clear – it is just like any other software or mobile application which catered over the internet and used in the browser. But unlike websites, they provide *more functionalities and features*. We everyday use numerous web applications such as Gmail,

Google Drive, Facebook, LinkedIn, Trello, Slack, etc.

Web application workflow:



What exactly happens when u visit a domain & how website works :

Resource :

1. https://www.freecodecamp.org/news/what-happens-when-you-hit-url-in-your-browser/

2. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

API :

The difference between API and web application

The major difference in web app any file received from the server like HTML CSS and But in API data will be received from the server.

Example:

When you visit www.amazon.com and if you click on a product it will show product image and content and description, again if you click on other products it will show product image and content and description, there only data changing but the markup is the same.

API DATA Example:

https://jsonplaceholder.typicode.com/posts

Resource:

https://www.freecodecamp.org/news/what-is-an-api-in-english-please-b880a3214a82/

Single Page Application: (SPA).

Single page application is a single page (hence the name) where a lot of information stays the same

and only a few pieces need to be updated at a time. For example, when you browse through your

email you'll notice that not much changes during navigation - the sidebar and header remain untouched as you go through your inbox.

The SPA only sends what you need with each click, and your browser renders that information. This is different to a traditional page load where the server re-renders a full page with every click you make and sends it to your browser.

This piece by piece, client side method makes load time must faster for users and makes the amount of information a server has to send a lot less and a lot more cost efficient. A win-win.

Topics which will be learning in next months:

Front end:

1. Html
2. Css
3. Java script
4. React
5. Redux
6. Redux thunk
7. Redux saga

Back end:

1. Node js
2. MongoDB
3. Express js
4. Sql
5. Heroku
6. Aws

Session - End

Thank You