**React Js**

2. A 7m Overview of Web Development and React

All of this relates to react since where react is a javascript

framework for web applications. As a web app framework reacts optimizes how javascript is used on the

Web browser to make the app dynamic.

HTML CSS JS

|  |  |  |
| --- | --- | --- |
| **Skeleton** | **CLoths** | **Magical body** |

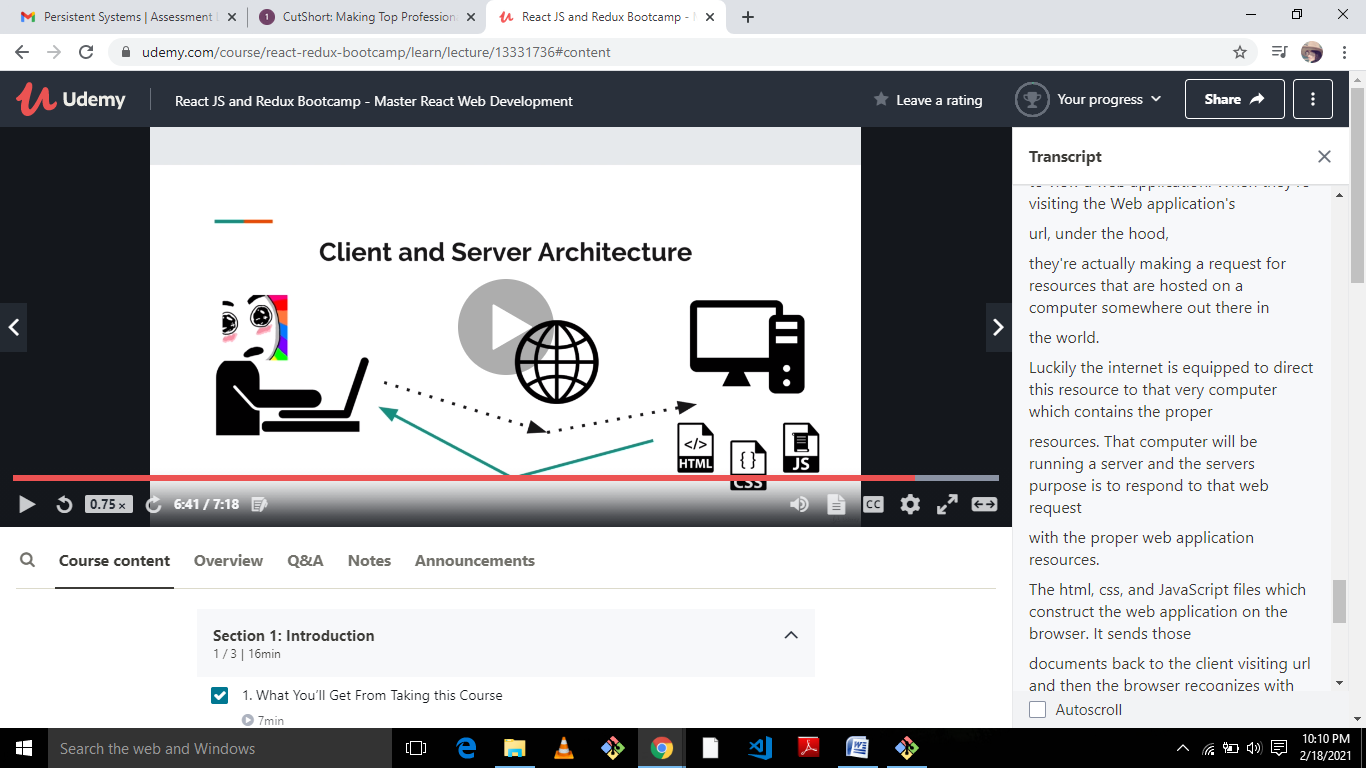
**Client-Server Architecture:**

First the user of the web application is the client. As a client,

they use their computer's web browser to view a web application. When they're visiting the Web application's url, under the hood, they're actually making a request for resources that are hosted on a computer somewhere out there in the world.

Luckily the internet is equipped to direct this resource to that very computer which contains the proper resources. That computer will be running a server and the servers purpose is to respond to that web request with the proper web application resources.

The html, css, and JavaScript files which construct the web application on the browser. It sends those documents back to the client visiting url and then the browser recognizes with the html, css and js that is now viewing a web application and it should present something amazing for the user.



[IMPORTANT] Course Repo and Software Installations

**Important! This is required reading to be successful in the course.**

**Here is the official course repository:**

<https://github.com/15Dkatz/react-bootcamp>

**Software Installations:**

**1. A Code Editor**

* For a code editor, I recommend Visual Studio Code: https://code.visualstudio.com/. I also recommend configuring it to open via the command line:
  + CMD-SHIFT-P
  + Enter “Shell Command”, and choose “Install ‘code’ command in path.
  + Then you can run $ code /folder to open folders in VSCode from your shell.

**2. A Command Line Application**

* For the command line application, on MacOS or Linux, I recommend the native Terminal application (or Iterm2)..

**WINDOWS USERS: I absolutely recommend downloading Git Bash. This course was recorded using a bash environment. Using Git Bash will make doing this course a smooth process. Here’s a great stackoverflow thread for configuring Git Bash on on Windows:**<https://stackoverflow.com/questions/42606837/how-do-i-use-bash-on-windows-from-the-visual-studio-code-integrated-terminal>

**3. Node.js.**

If you don’t already have Node.js installed, here is the download page: https://nodejs.org/en/download/. Once you have Node.js installed, run the following commands on the command line

* $ node -v
  + See a value like `v8.8.1` or higher
* $ npm -v
  + See a value like `v5.4.2` or higher

\*\*\*\*\*

With a Code Editor, Command Line App, and Node.js installed, you should be ready to go!

**Set up portfoilio app**

1.mkdir react-bootcamp

Cd react-bootcamp

Npx –version

npx create-react-app portfolio //create react app

goto command-pallete in visual studio and type “shell command”- add path

Run React app and project Structure

1.goto portfolio

npm run start :open localhost:3000

|  |  |  |
| --- | --- | --- |
| Directory/file name | | Description |
| Public/index.html | | The browser receives this document and then  uses it to display the content on the page. |
| Index.js | Import react | import sets up the file to write react code  by importing global values and methods that are defined within the react library. |
| Import react-dom | library takes care of actual application rendering meaning displaying the application on the browser  itself. |
| Sericeworker | | Use offline application |
| ReactDOM.render | | ReactDOM.render is a function which takes your react code and then inserts it into  the html. |
| App.js | | ReactDOM.render inserts this components JSX as HTML elements within the root div in the H.HTML. |
| Manifest.json | | Information about the project |
| Src/app.js,index.css,app.css,gitignore,  package.json | | React-script-build: The build script builds a production |

ReactDOM, Elements, and JSX

But we can create an element rather quickly. A react element is the smallest object within a react application.

They're pretty much copies of html elements.

So think of div tags, h1 tags, image tags etc. Those are all each simple elements, and a react element is

very much the same.

They represent an object at that level, like a div, or an H1, or an image in react.

We represent those elements with a syntax called JSX. Meaning javascript, JS,

and XML-like syntax, X. So XML is this other markup language like html which is based off the

pattern of using arrow like tags. For our first element let's add a div, and this will say react element.

So that becomes what we want to render within the application.

now we have the second argument of the render method, and in the second argument we specify where we want

this react element to get inserted. To do that we can access the core javascript document object, this

document object is provided to any javascript file running in the browser and there's way more to this

document object than meets the eye.

Src/index.js--🡪index.html(id=root)