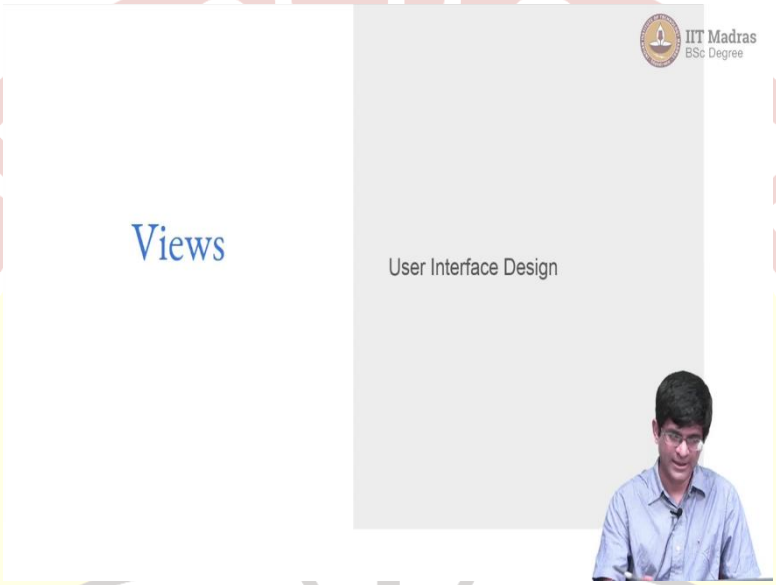
IIT Madras

ONLINE DEGREE

Modern Application Development - I
Professor. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Views

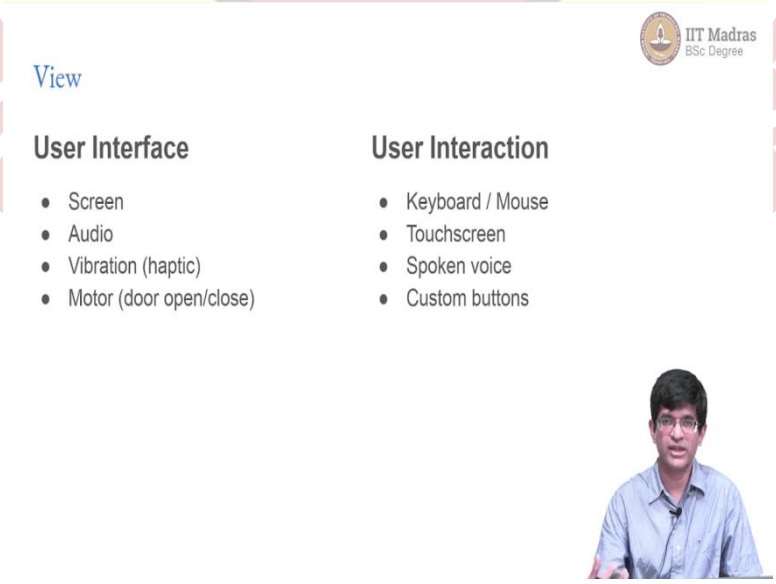
Hello, everyone, and welcome to this course on modern application development.

(Refer Slide Time: 0:17)



We are now going to look a little bit more in detail on views. And what we are keeping in mind over here is the whole idea of user interface design.

(Refer Slide Time: 0:28)



So, there are a couple of things that need to be kept in mind over here. One of them is that a view consists of two parts in some sense. One is the user interface. This could be a screen, which basically displays something for the person to look at. It could be audio, where you are listening to something, it could be completely audio feedback that you get from the application. It could be vibration, or haptic, haptic feedback basically means touch based feedback.

So, for example, that could be applications where just based on the vibration of phone or some other kind of system that you are working with, you get some feedback on whether what you are doing is correct or not. Now, the interesting thing is, we need to sort of when we are abstracting this out, you can take this one step further, I could even go so far as saying that the whole idea of opening or closing a door is a user interaction.

So, a door by its very nature, we do not normally think of it as something being a user interface, or something that is associated with computers or user interaction in any way. It is not associated with computers, well, nowadays, you might have smart doors, so to say, but even a simple door has a user interface, it gives you a handle. And the idea is you should be able to, if necessary, turn the handle and push or pull the door in order to open it or close it.

So, the functionality that the door performs is basically to be opened or closed. And this is a sort of motor interface. Once again, it is interfacing with a person. The other side is that user interaction. So, the interface is what it provides to you what you see or what you feel. Whereas, the user interaction is how do I actually interact? How do I give inputs to the system? In the case of a computer, a desktop computer, for example, the most common would be a keyboard or a mouse. With mobile phones, or tablets, it will be a touchscreen.

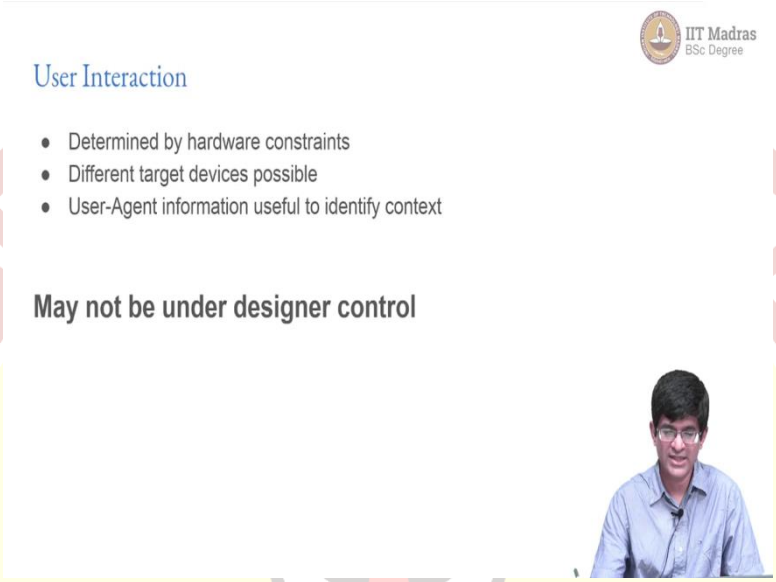
With increasing numbers of devices, especially the smart assistants, Google Assistant, Alexa, Siri and so on, we are talking about spoken voice, you say a command word or a control word. And the device sort of wakes up. It is constantly listening for that word, it wakes up when you say that and it starts listening and responds based on what you have said. And people actually find this useful, especially in a context where perhaps your hands are busy doing something else.

You might also have custom buttons, or like I said, in the case of the door, the door handle is a user interaction device. So, it could be something which is not what you normally think of as a user interface. But the principles are the same. And the reason why it is important to keep

that in mind is because many of the principles of user interaction or user interface designs that are made in the real world also have applications or counterparts in web applications.

So, you need to sort of understand what is it that makes something easy to use, and see if you can bring those similar principles to play when you are designing an application.

(Refer Slide Time: 3:38)



So, the user interaction is largely determined by hardware constraints. So, I might, for example, find that, you know, I am targeting a desktop computer that has a keyboard and mouse, in which case automatically, I might have ruled out people who are using mobile phones. If I expect that you have a keyboard, and you need to type in things using a keyboard, that makes it essentially impossible, a physical keyboard that essentially makes it impossible to use a mobile phone, unless perhaps you have some kind of a hacky way of you know, emulating the more physical keyboard on the mobile phone.

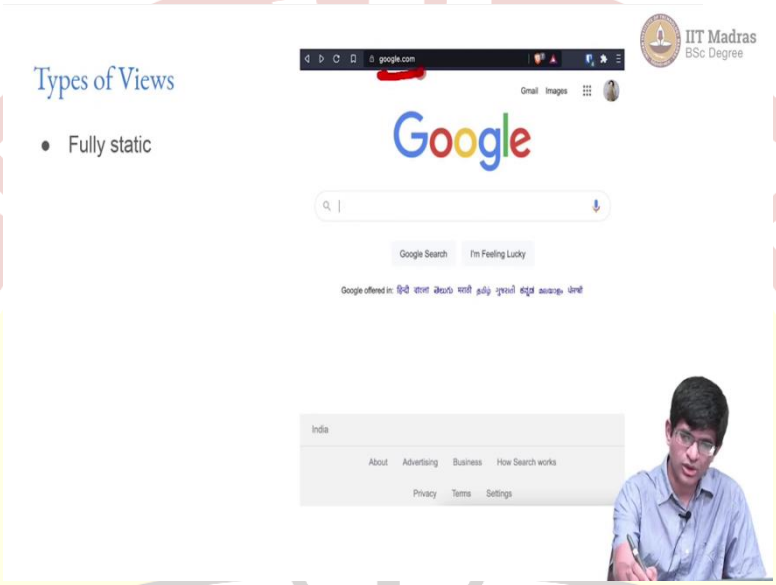
So, increasingly, as a designer, you need to keep in mind the fact that you might have different target devices, you cannot just assume that everybody has a keyboard with the same character set even, you might have people from different countries having different keyboards. And in a lot of these cases, there is usually information that comes from the client, from the person who is actually interacting with your application. It is usually called the user agent, because that is the information that is sent from the client to the server and that can be used in order to identify the context and very often, the server can then adapt to that.

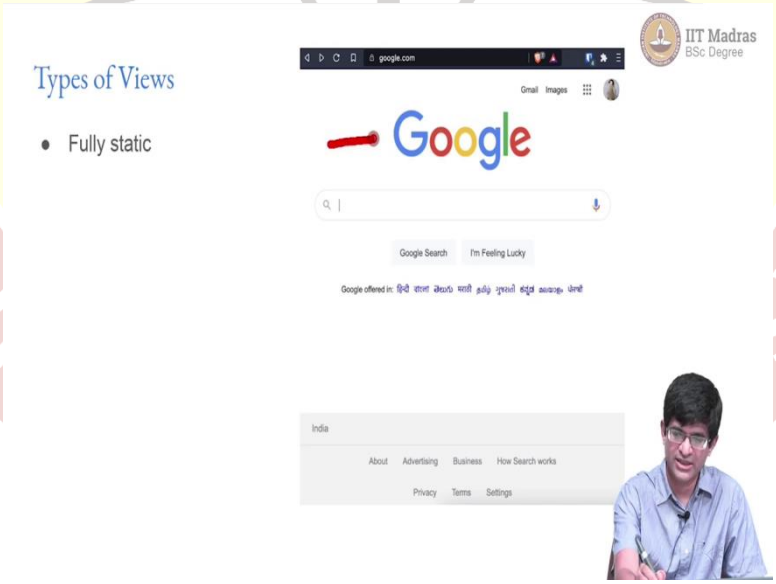
The important point to keep in mind over here is that this kind of information, what kind of hardware constraints do you have? What kind of context information can you get may not

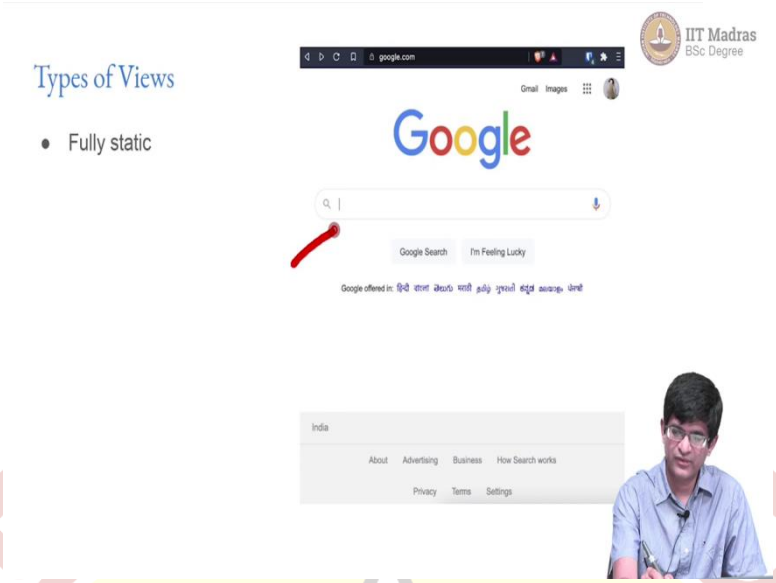
always be under the control of the designer. Over here, the way to think about it is, each of you who is going through this course, is a potential designer of a web application. That is why you are doing this course you want to be able to design applications.

And in such a context, you need to keep in mind that you may not always have all parts of it under your control. And how would you actually go about implementing certain things.

(Refer Slide Time: 5:29)







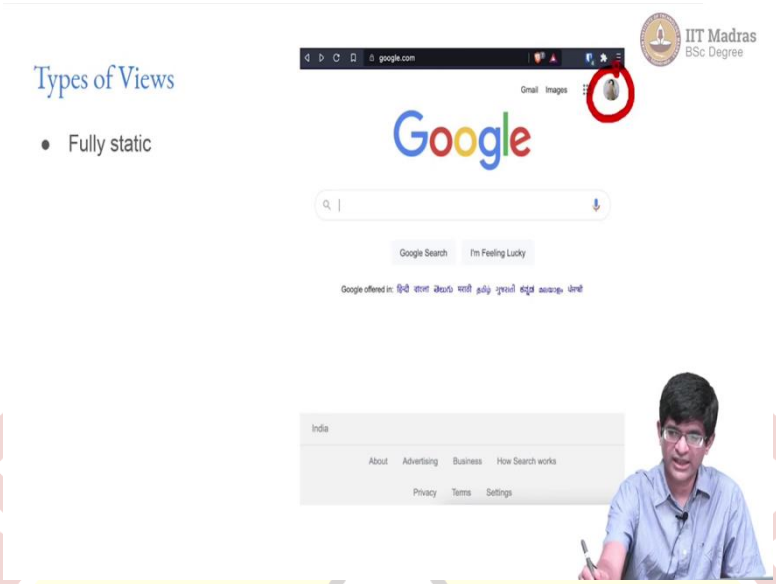
We can also think in terms of different types of views, an example is that you could have something which is a fully static page in HTML page. So, now I am increasingly going to be focusing on web pages, because after all, we have decided that the web is going to be the platform that we are using for our application development. And based on that, most of the examples and other kinds of ways of building up, the views that I am going to talk about are going to be based around the idea of web pages.

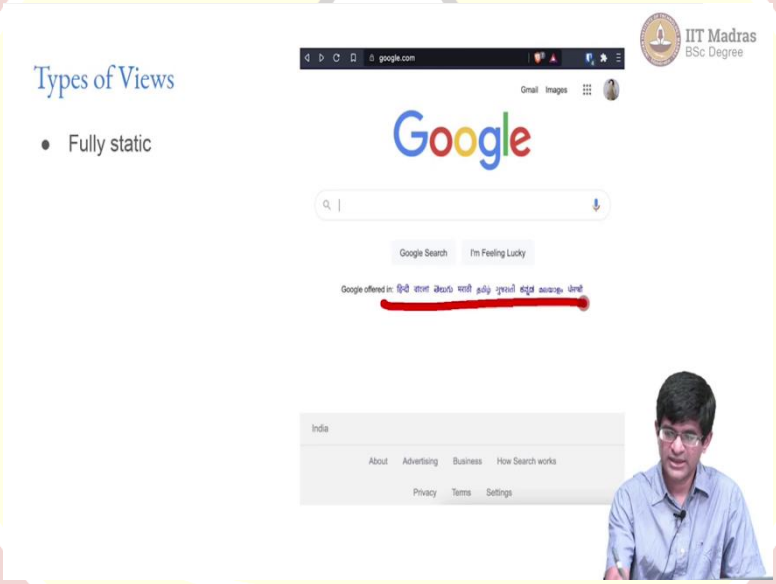
So, an example could be a fully static webpage, the Google web page, if you just go to google.com, what appears before you is a largely static webpage. You look at it, there is very little content on it, it actually just has one big image saying Google, and it has one very nice simple looking text box out there where you can enter information.

And if you think about it, pretty much this entire thing is static any time any day, any year, you go to the Google website, apart from maybe some cosmetic changes, like maybe the shape of the font, or something of that sort had, maybe it may have evolved with time. Most of it has remained pretty much the same. So, potentially, this could have been made a completely static website. Pure HTML that is delivered to you when you try connecting to it.

Now, in practice, this is actually a bad example, because this web page actually does have a lot of dynamic information on it.

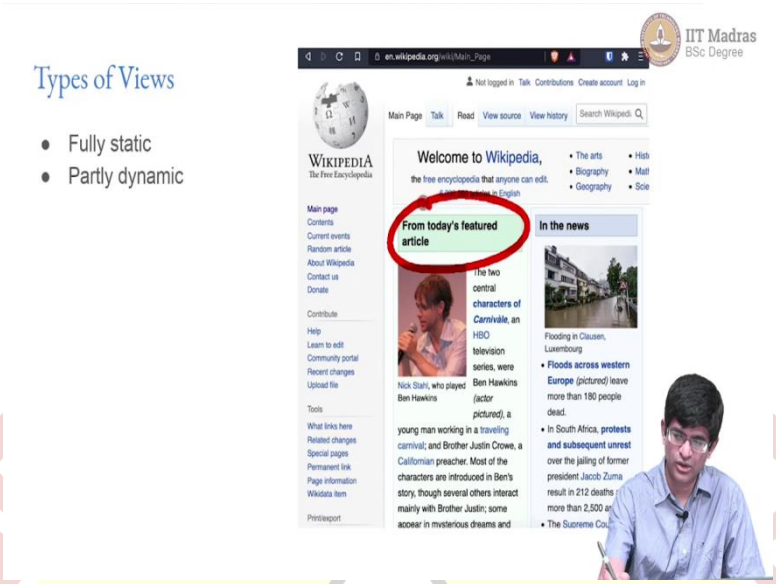
(Refer Slide Time: 7:00)

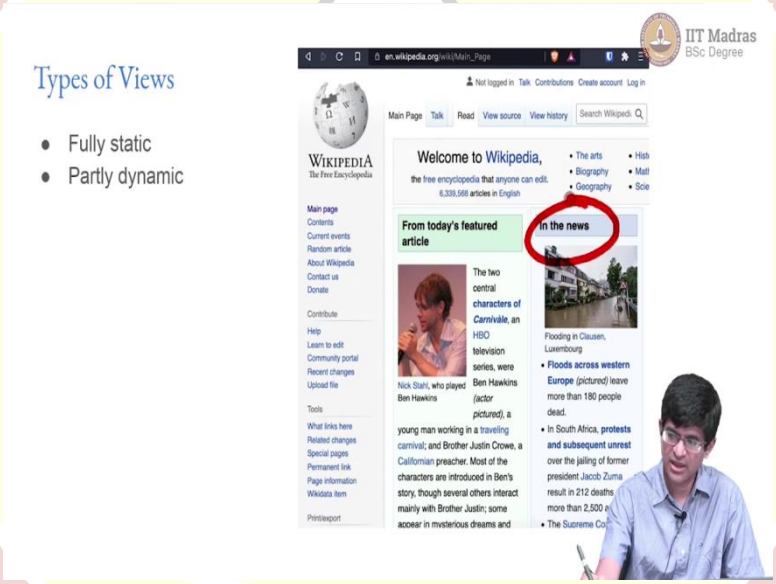


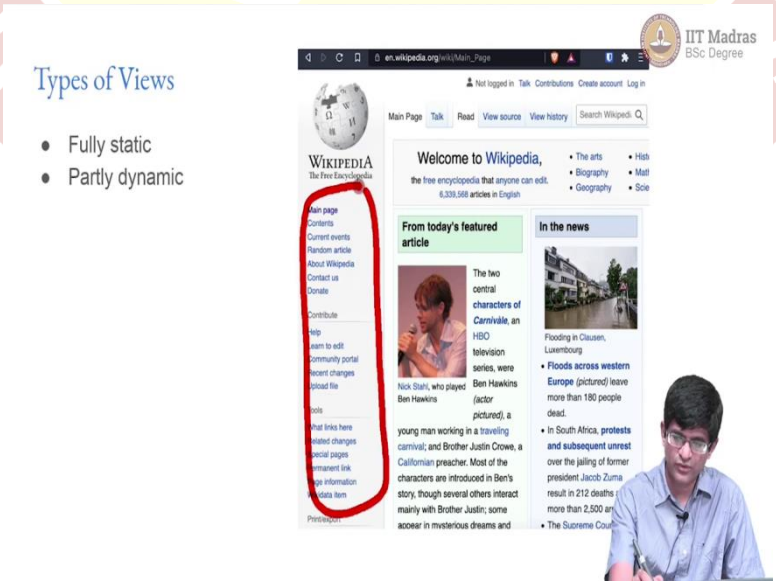


For example, it has found out that I have logged in and has managed to get my… the icon and put it out there. It also knows that I am logging in from India. And hence, it shows me a bunch of possible Indian languages that I might be interested in using. So, clearly, there has been something dynamically done in order to generate this page to customize it for me. But in principle, this should have been a static website.

(Refer Slide Time: 7:27)



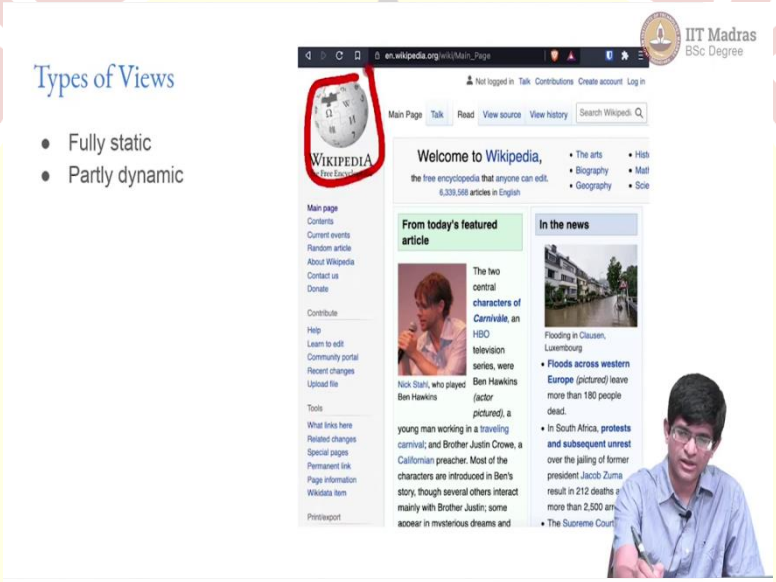


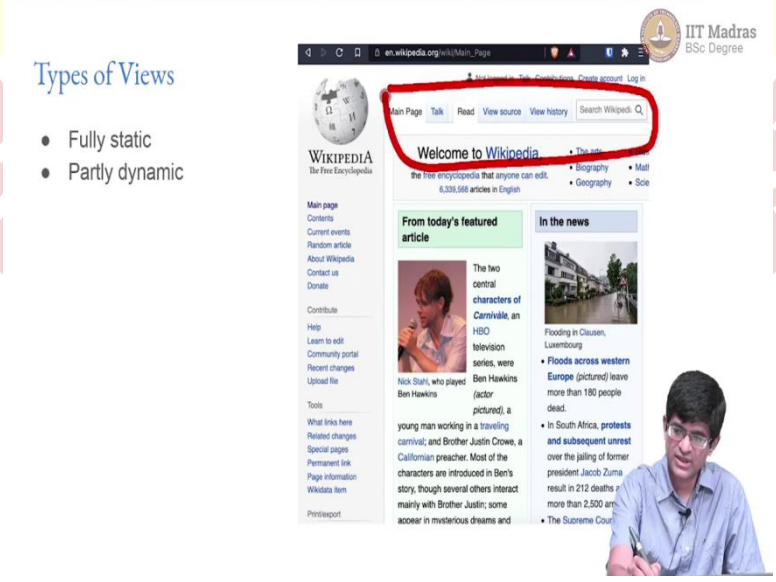


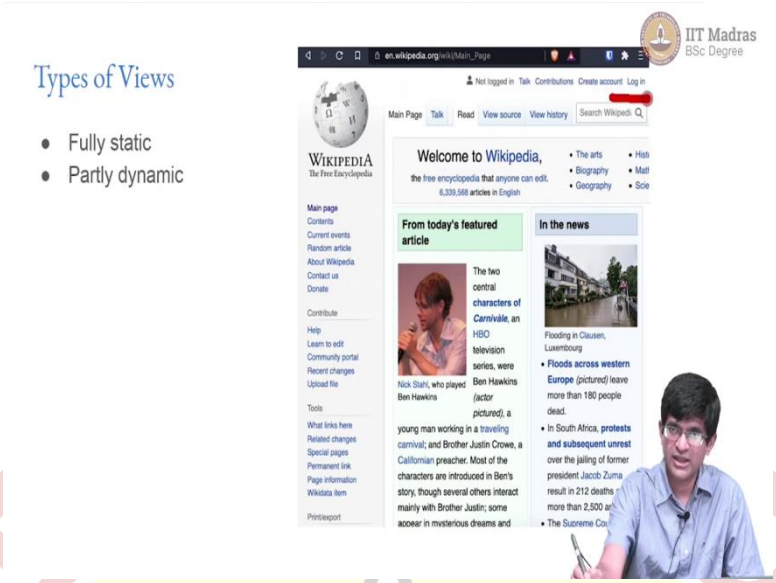
Wikipedia, on the other hand, is an example of something which by its very nature is at least partly dynamic. Now, the content of most of the wiki pages is mostly static, here, again, we have something where once you have entered the information, you know, you have created a wiki page for some material, you do not expect it to change very often.

But if you look at, for example, the front page of Wikipedia, this is what it looked like today, it essentially has something called today's featured article, it has something about the news, it says something about the various items from Europe from Africa. And it also has a large part of the rest of the thing is basically static.

(Refer Slide Time: 8:09)

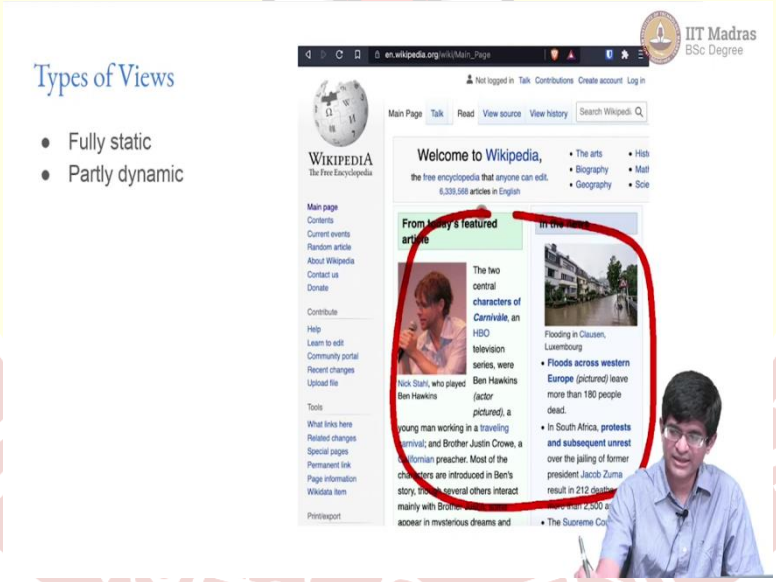






This icon, for example, or the logo remains essentially the same. Most of this rest of this traffic pretty much remain the HTML out here remains the same, it does allow you to login. And if you had logged in, it might have done something a little bit different.

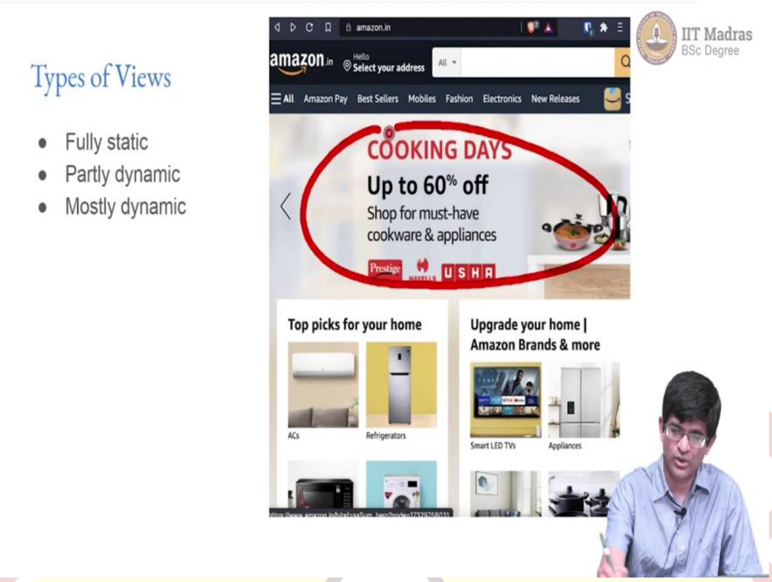
(Refer Slide Time: 8:26)

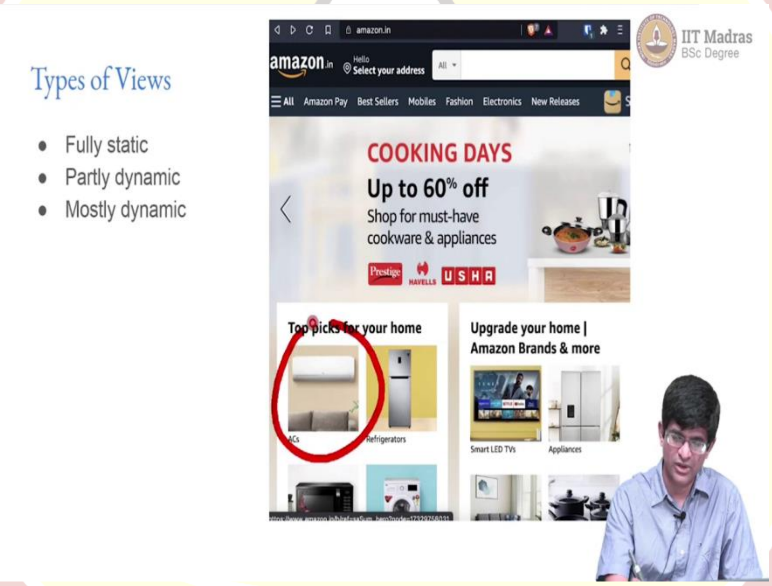


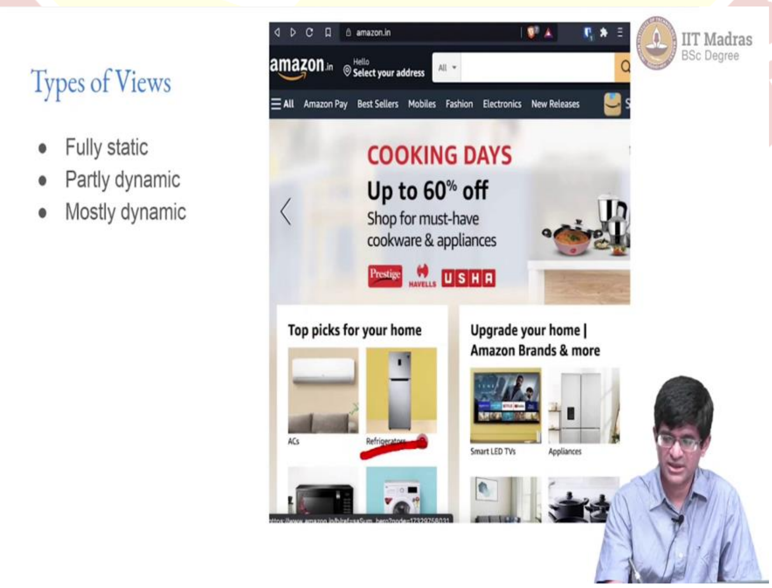
But the fact that there are these sections out here that are definitely meant to change, each time you load it essentially means that this is at least partly dynamic page. So, it is a combination, there is a lot of static text, which can just be pulled out from files or databases and dumped over the network and sent to you.

But there are also things that the server needs to do it needs to sort of look up, is this a new user? Have they logged in? If so, then, you know, how should I react and what information should I send them?

(Refer Slide Time: 8:58)





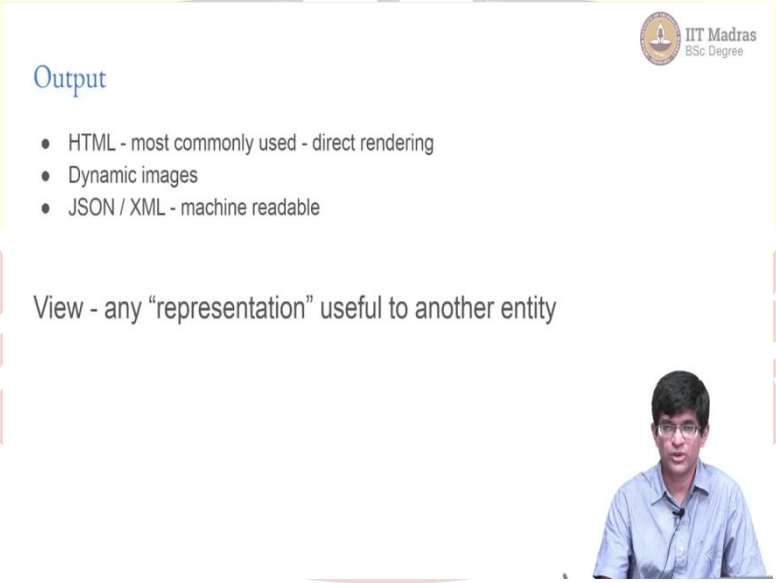


And finally, we come to a page like for example, amazon.in, which is almost completely dynamic, you look at it, every part of this page out here, this is the cooking days or something which is probably just, these few days that we have out here. It has sort of updates on AC's, which is clearly something to do with the fact that we are in summer, similarly refrigerators, it has various different kinds of things on appliances, and TVs.

And pretty much almost the entire page, as you can see is dynamically generated. There are sort of chunks of text that are sort of static, but this whole page, if you look at it, maybe 90 percent of it is going to be changing, some part of it or the other could change with it time. There is no such thing as a fully dynamic page. I mean, even if you had something like you know, a PHP script, which is generating every page, which after all is what is happening over here. It is also pulling out some chunks of text.

So, it is not like randomly generating every single character that comes out to you. That is why I am calling this a mostly dynamic page. Rather than fully dynamic. So, anyway, the point is you could have these different kinds of views when you are interacting with a web application.

(Refer Slide Time: 10:10)



Now, it is not just that, you might also find that you have different kinds of output that are coming to you. The most common one, which is used for direct rendering. And the one that we normally think of when we are looking when we are thinking of a view is direct HTML. The HTML is just sent across from the server to the client, and the client, which is the browser is just going to render it, render meaning it decides if this is a heading, it needs to be

displayed with this font in this location, if it is a paragraph, it needs to be done with this font in this location, and so on. So, that is the part that is called rendering.

Now, on the other hand, you could for example, remember the example I said about histogram of marks. That can also be thought of as a view. And that histogram is something which is actually going to get dynamically generated. Anytime that you go and select the course.

Of course, I might do certain things like for example, once I have generated the image, I save it and cache it. So, that next time around, no, I do not need to generate it again, that is an optimization, we are not thinking about that, in principle, at least, it is being generated fresh every time you ask for the histogram, of a course.

So, an image, for example, could be a view, it is not normally what we think of as a view. But it could be an example. And yet another example could be JSON or XML, which is purely machine readable. It is not meant for a human to read or directly interact with, but is important, because at the end of the day, the JSON output which is generated could be used by some other application in order to do further processing.

In other words, you need to keep in mind that when we are talking about view, a view is any representation that is useful to another entity. So, any representation meaning that it could be HTML, it could be plain text, it could be an image, it could be formatted structured data, like JASON. And it should be useful to another entity. It might be useful to a human being it might be useful to another machine. Any of those constitutes a view.