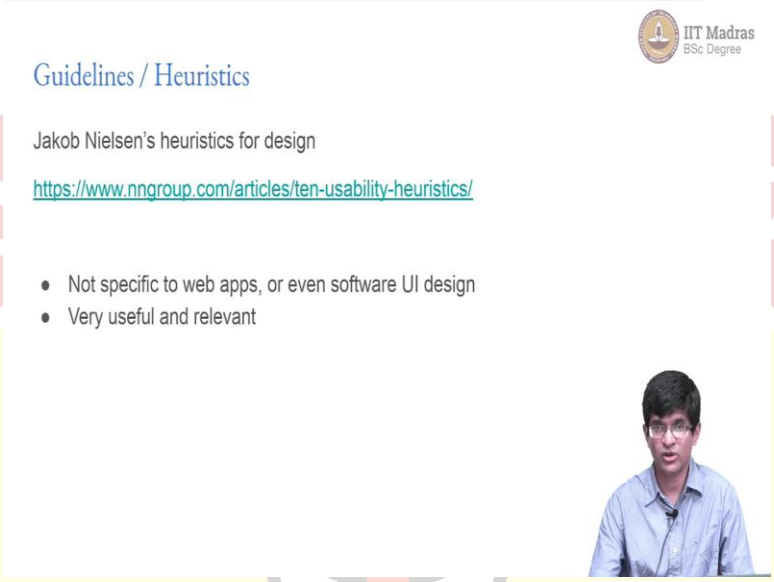
IIT Madras

ONLINE DEGREE

Modern Application Development – I
Professor Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Usability Heuristics

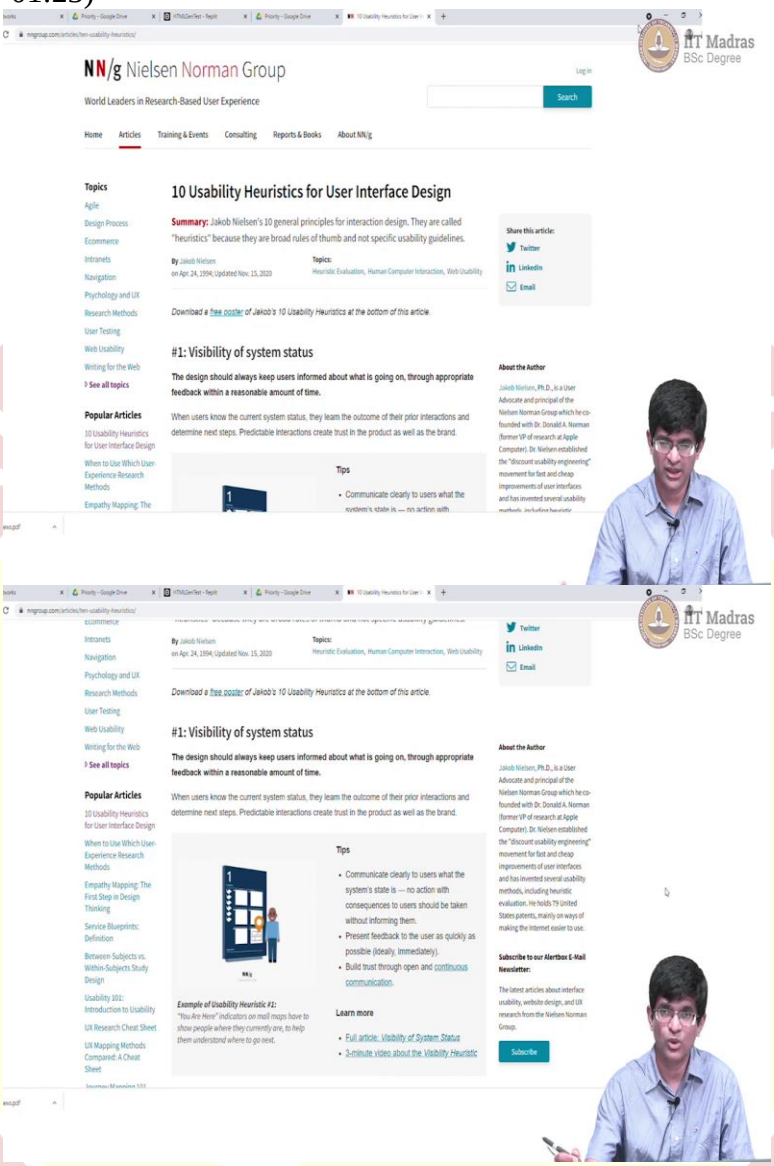
(Refer Slide Time: 00:17)



Hello, everyone, and welcome to this course on Modern Application Development. Now, user interface design, as I told you is something that has been studied extensively over the years. And one good set of heuristics, so to say, is what we have over here, Jakob Nielsen. He is a person who has been doing research on user interfaces, not just in the context of web applications, generally in computer user interfaces, yes.

But, the first set of sort of usability guidelines or heuristics that he had put out, were in 1994, when the web did not really exist in any big form. So, clearly, this is something that he had been working on well before many of these concepts came into place. Now, this website over here, the ten usability heuristics is something that pretty much has the information from 1994, but has again been updated over the years. Especially with, some kind of context for modern day. And it is worth sort of going into that in a little bit more detail.

(Refer Slide Time: 01:23)



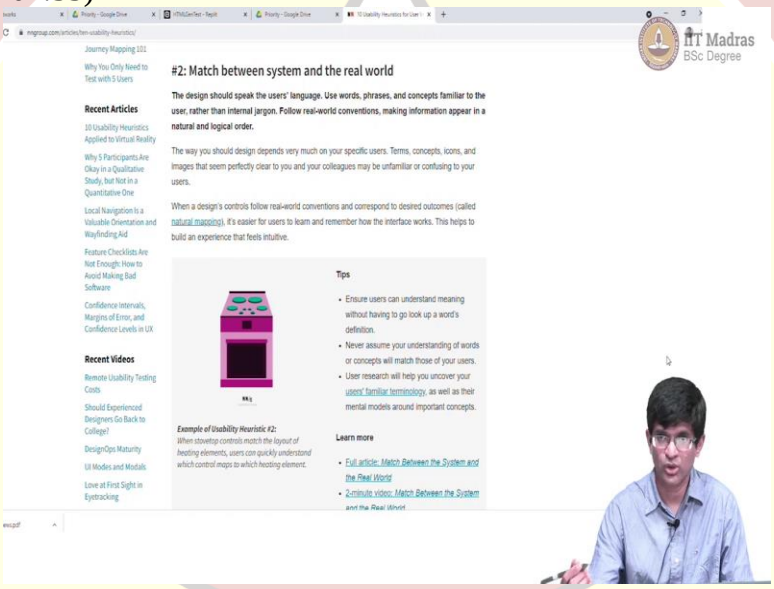
So, what I am going to do over here, rather than putting up the slides is to just sort of run you through the page itself, which talks about the usability heuristics. And some, why some of those ideas are good? And one of them, for example, is the very first one is the visibility of the system status. What do we mean by that? You need to communicate as the author of an application, you need to be able to communicate with the users, what this system is doing?

So, for example, even something like this, as you can see over here, there is a cursor over here that is moving around when I move the mouse, over the screen. Now, that cursor is giving me immediate feedback about what will, about where exactly, the computer now expects me to interact with it. And it also tells me something about what will happen if I do go ahead and click.

So, for example, when it is an arrow, it indicates that nothing much is going to happen, if I do go ahead and click on the screen.

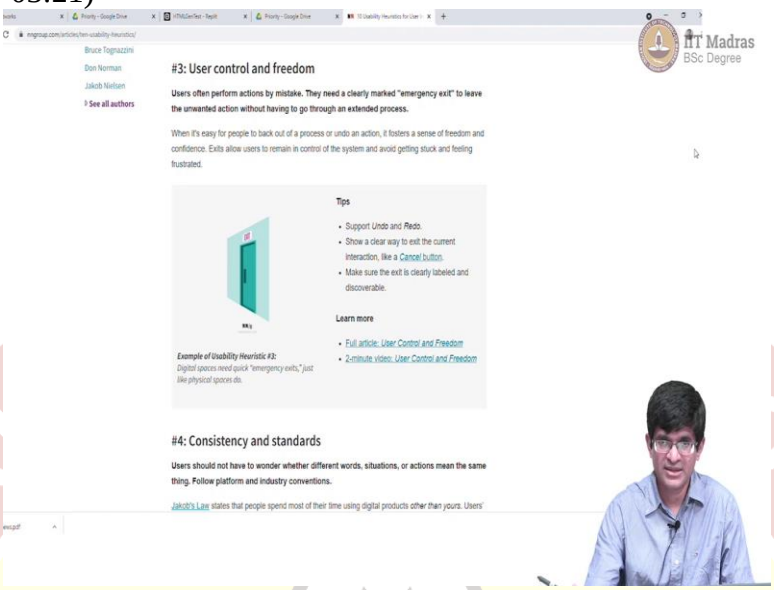
But, if it changes to this hand icon, for example, up here, it sort of tells me that this is a hyperlink. And if I click on it, there is a chance that now, something else will happen. So, that is an example of a system status. Similarly, on a map, which is let us say pasted on a wall somewhere or in a mall, you might find that there is a you are here, which basically tells you, this is the system status. It has nothing to do with computers again. But it is useful information to the person who is trying to use that map.

(Refer Slide Time: 02:55)



Match between the system and the real world, a simple example is you have a stove with multiple burners, the location of the burner knob should match with the location of the actual burner. That makes it clear, let us say that the knob on the right controls the burner on the left, that is not intuitive, I do not expect that to happen. And it becomes difficult to use such a device.

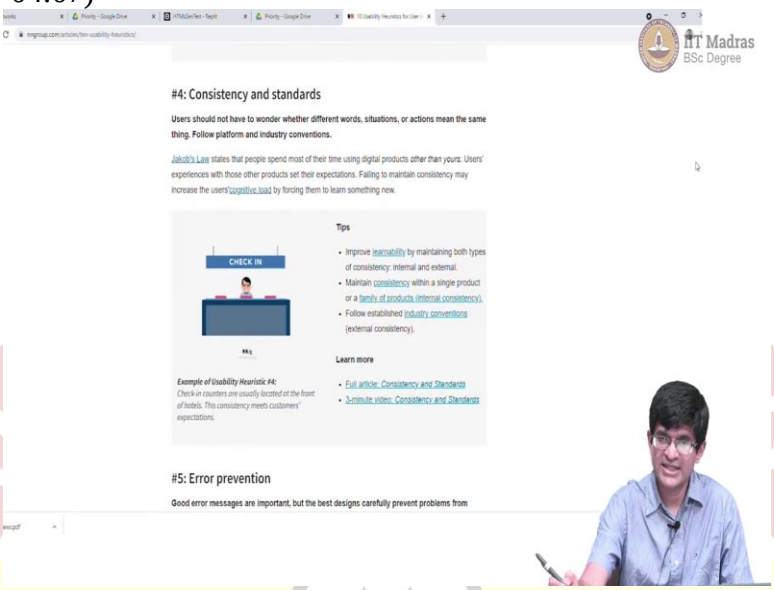
(Refer Slide Time: 03:21)



User control and freedom, one of the things is, you will find that pretty much any application that you are used to these days has some kind of undo support. You do something it, okay it was a mistake, I need to take that back, it allows you to go back, including to the point where for example, let us say you have deleted a file somewhere, it does not actually usually delete it, it puts it into a trash or, recycle bin or whatever it is called on the application on the OS that you have.

Why is that? Because, there is a chance it was done by mistake, and you might want to undo it. So, it gives the user that sort of safety net, some ability to do things without worrying too much that, something is going to be a completely destructive operation.

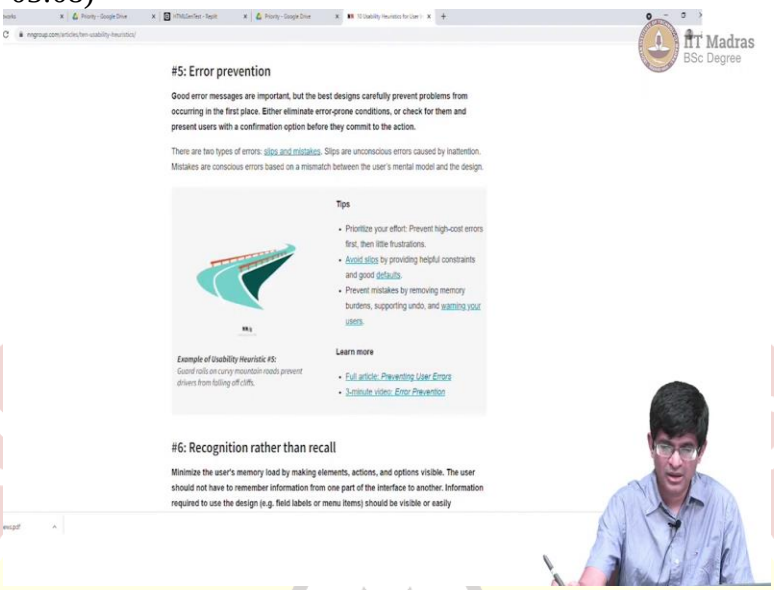
(Refer Slide Time: 04:07)



Now, this number four, consistency is one of the most important things to keep in mind. What consistency means is, anytime I look at a page, if I or when I am browsing through a website, or if I am going using browser for something, I expect certain consistency. An example would be that, whenever I hover over a link, I expect the icon, the cursor shape to change, or perhaps the link to get highlighted in some way.

Instead of that, if you did that, any time that you hover over this, if you hover over links on the right hand side, it would show you a particular kind of cursor. If you hover over links on the left hand side, it does something else. Maybe it automatically assumes you are clicking and just goes through the link, that is inconsistent. Though there are actually pages that do things of that sort. And those can be very disorienting to a user. It is difficult to get used to a website and be able to use it when such things happen.

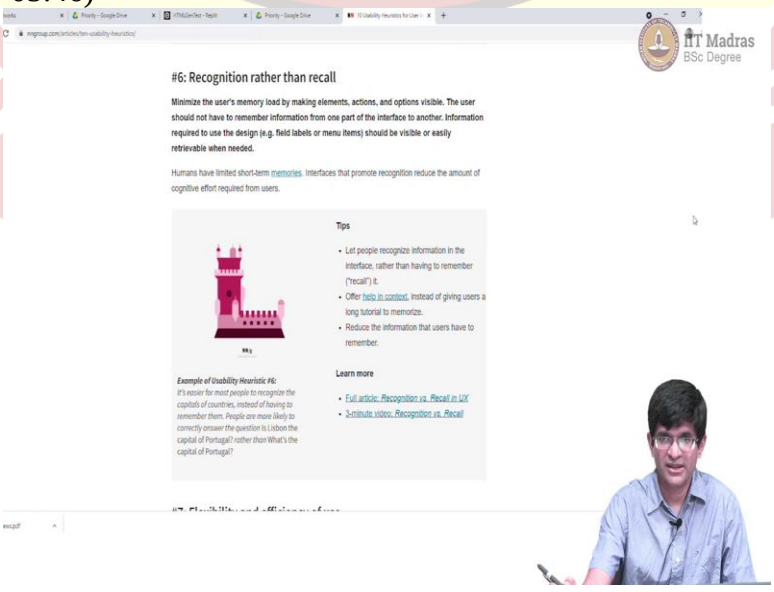
(Refer Slide Time: 05:08)



Error prevention, it is not just good enough to detect errors after they happen. Ideally, you should be able to prevent an error from happening in the first place. The example given over here, of course is, you put guardrails on roads when they are going close to a cliff edge or something like that, a mountain edge. Why? Because you want to make it very clear to the person that look, if you go too fast, you are likely to fall off the edge of the steep slope.

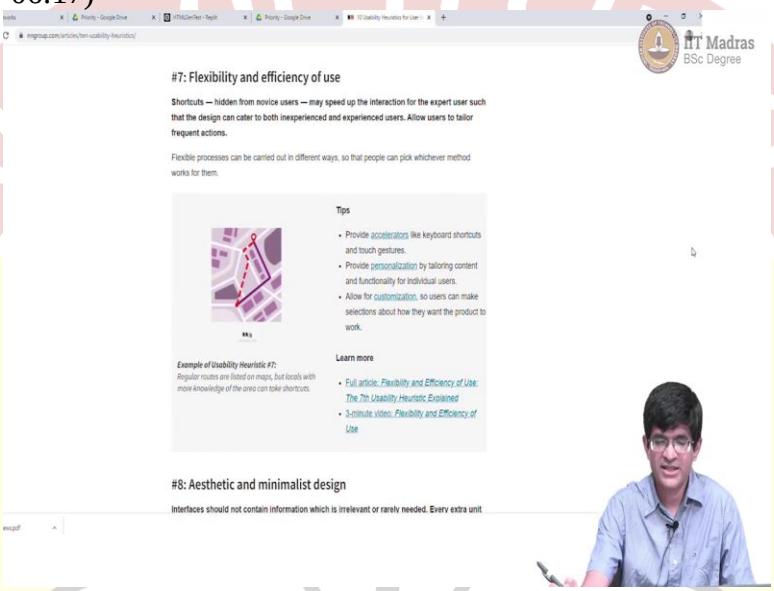
You would rather prevent that from happening rather than saying, if they do fall off, we have some way of catching them when they do, that is not good enough. Ideally, you want to prevent it from happening in the first place.

(Refer Slide Time: 05:46)



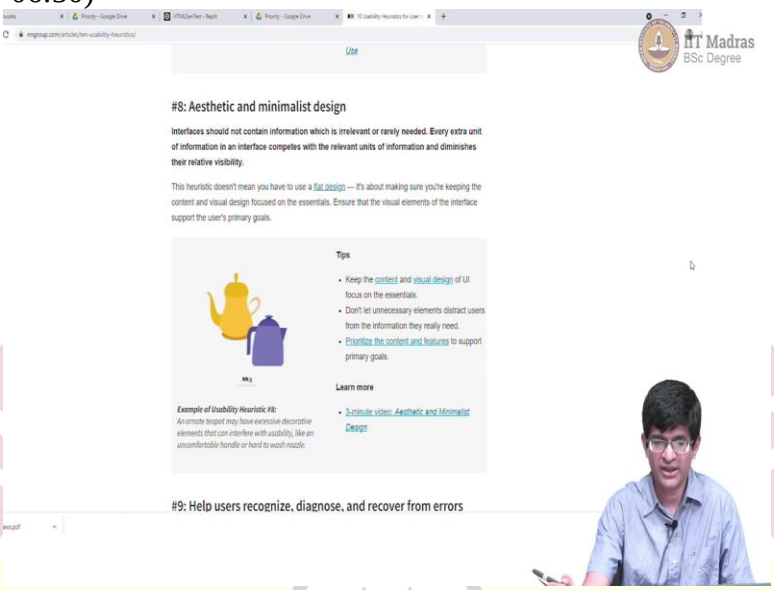
Recognition rather than recall. Once again, this is, related to the door handle. I should be able to look at the door and say if I press over here, I should be able to open the door, rather than saying, I remember this sign corresponds to push, and therefore, I, if I push the door will open. The recognition saying that, this looks obviously like something I need to push or to press, is more important than something which needs to be remembered so that I need to act on it correctly.

(Refer Slide Time: 06:17)



Flexibility efficiency of use, most of you who are using desktops, at least you would find that there are shortcuts, mostly using the keyboard for a lot of applications. In fact, when I use Gmail on my desktop, a lot of my navigation is done by using the keyboard shortcuts, I do not use the mouse for going back to the inbox. I do not use it for composing a new email. All of those have some kind of keyboard shortcuts that make it easier and faster for me to do certain things. Those kinds of things, improve the usability of the system.

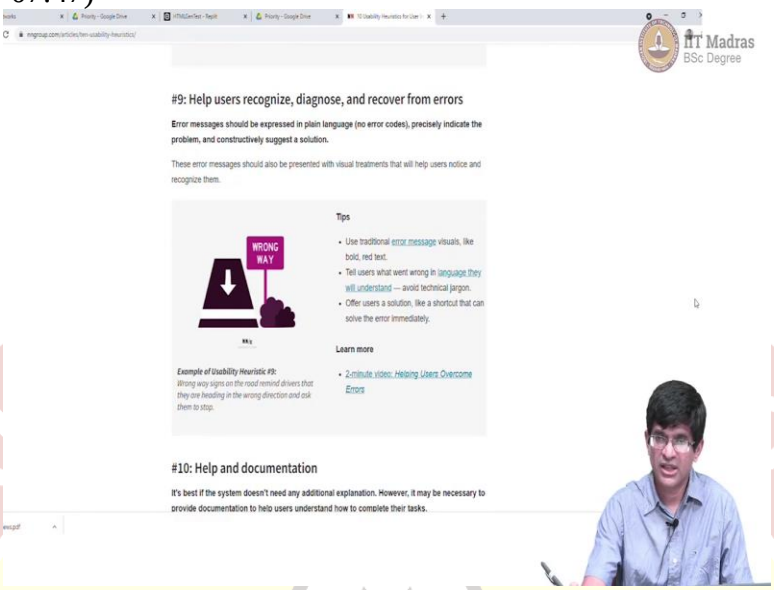
(Refer Slide Time: 06:50)



Aesthetic and minimalist design. Remember what I said about avoiding the use of excessive colors, and so on? This is something important, the interfaces should not contain information, which is irrelevant or rarely needed. In fact, even if you look at this web page, this looks quite nice. I have scrolled down to a point where it just has the bare minimum of information that is useful to convey the point that they are trying to make.

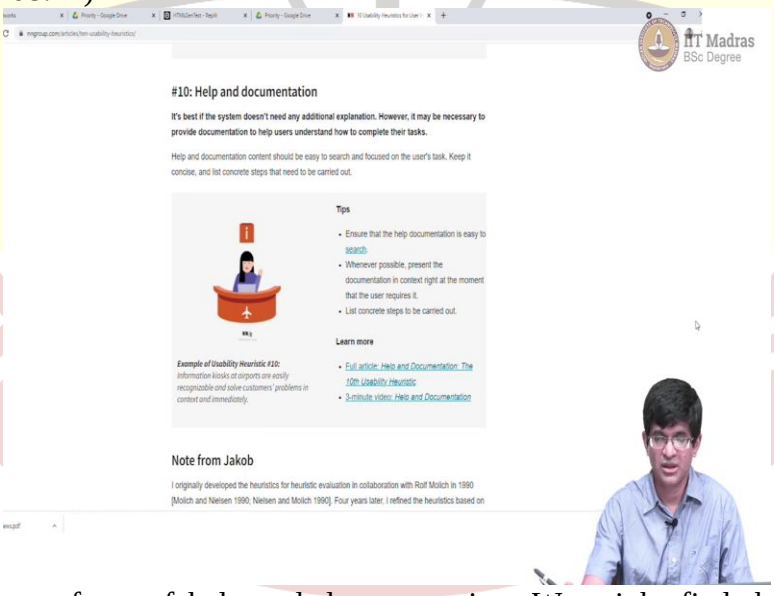
There is a lot of other navigation and other various kinds of information on this page. But it is not present right here. I know that if I scroll to the top of the page, I will be able to find it. Could this have been improved in some ways? Maybe, but it is not too bad. At the same time, it is aesthetic. It does not have sharp, jarring colors, there is some white, there is gray, clear black fonts, the headlines are in a larger font, the main point is in bold. The links are clearly differentiated in color. All of those are sort of useful things to keep in mind.

(Refer Slide Time: 07:47)



Similarly, help users recognize, diagnose and recover from errors. If you do find that you are doing something wrong, the system should try and catch it as early as possible and tell you wait, something is going wrong. Why do not you look at what you are doing? Rather than allowing them to make a lot of mistakes, and then sort of saying, oh, you messed up.

(Refer Slide Time: 08:11)

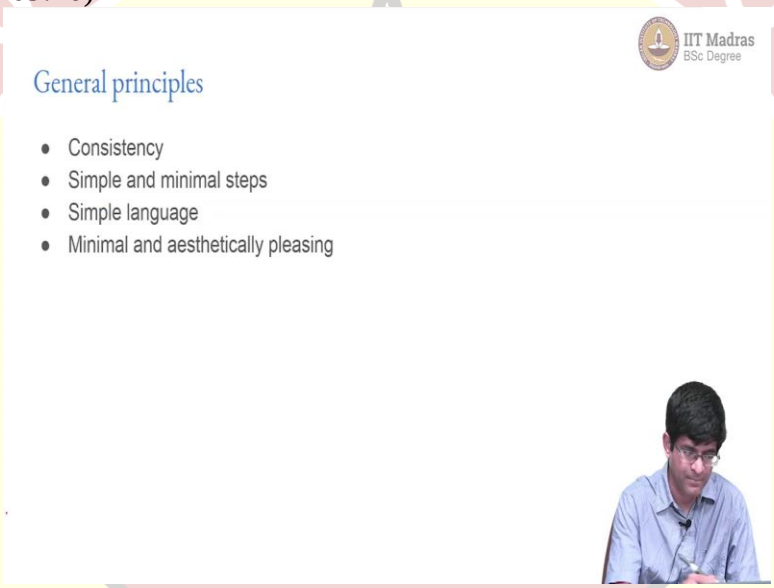


And, of course, some form of help and documentation. We might find that, the system we designed is perfectly usable and, very nice for us to use, that might not be the same viewpoint shared by a third person. So, how do you actually document in such a way that this becomes

actually usable and easy for a person to interact with your application, that is also something important to keep in mind.

So, like I said, these guidelines are useful, this particular website is a nice resource. I would strongly advise that you go through it, but there are a lot of other heuristics and other kinds of suggestions that you will find for UI design. There are increasingly many such heuristics for mobile based designs, for responsive designs and various sort of variants, Nielsen's initial heuristics are for any kind of user interface design, not just for web applications.

(Refer Slide Time: 09:10)



So, as I said, the core principles if you try to distill them down, consistency, simple and minimal, simple language, your language that you use in order to describe what something is doing, use simple words. Do not try to use a fancy word just because it is sort of, even though it may fit the bill perfectly. Because it throws the user off, they have to take a step back, try and understand what you are trying to say and then make use of the system.

And also minimal and aesthetically pleasing. Now the problem with all of these principles is that they are subjective, you might find that there are places where what you find to be aesthetically pleasing is different from what I do, for example. So, for all, for the most part, we are not going to be focusing too much on aesthetics here, except to try and keep this in mind, minimal as far as possible, do not unnecessarily complicate things beyond what is needed.