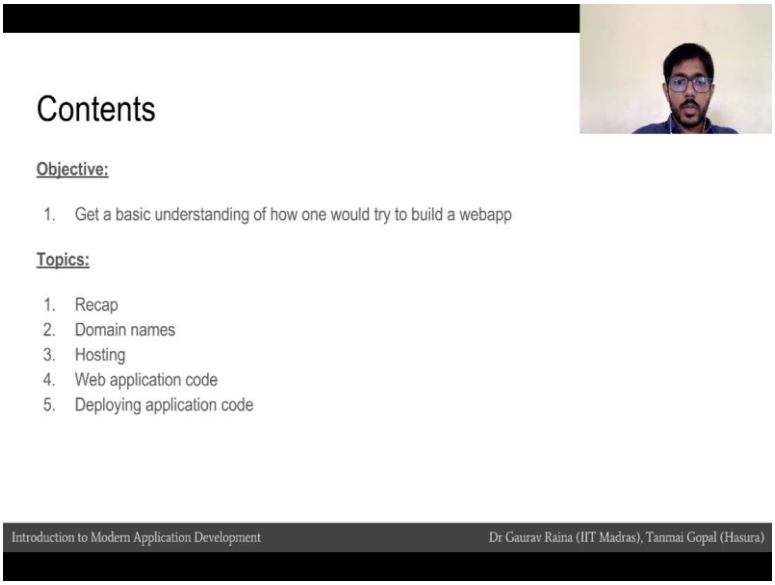
Introduction to Modern Application Development
Prof. Tanmai Gopal
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 03
Lecture – 03
Building a web app

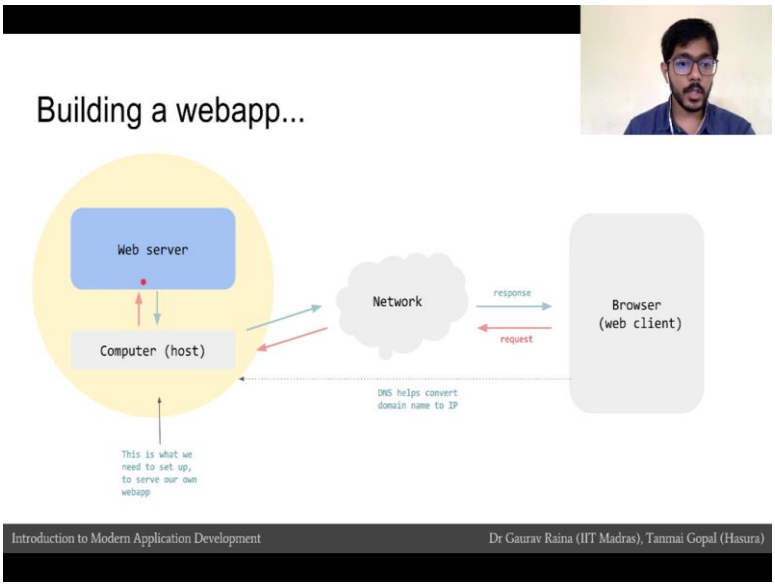
Hi everybody. Welcome to Module 3, we are going to talk about Building a web app.

(Refer Slide Time: 00:05)



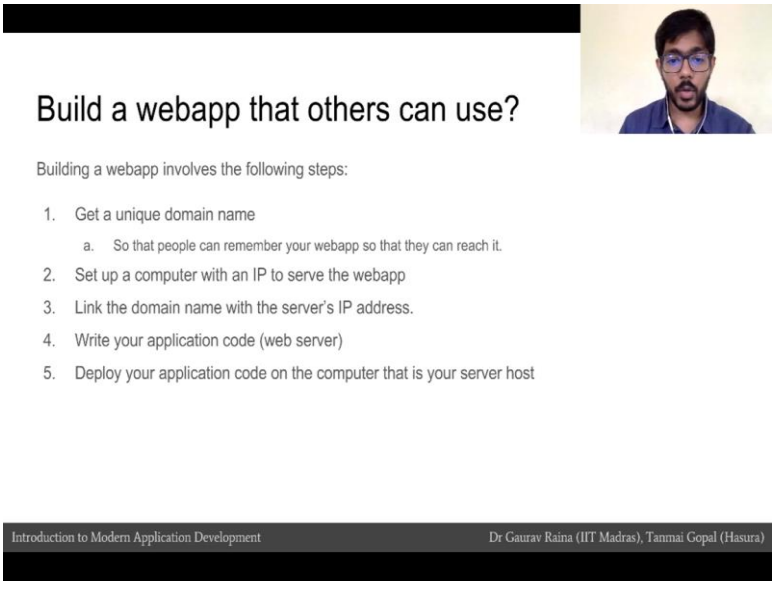
The main objective in this course is to get a basic understanding of how one would even try to build a web app.

(Refer Slide Time: 00:11)



Let us do a quick recap of what we covered in the last module. When a browser makes request to a web server the browser makes a request through the network on to the computer host, the web server process the request returns the response through the network and back to the browser.

(Refer Slide Time: 00:29)

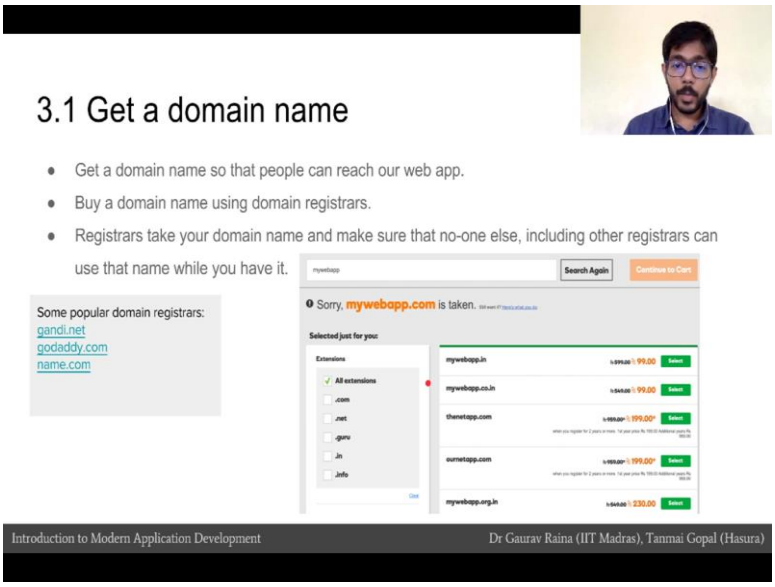


If we have to build our own webapp we would have to proceed with the following steps. First thing we would do is get a unique domain name, so that people can remember our

webapp so that they can reach it. The next thing we would do is get a computer with an IP to serve web.

Step three would be to link the domain name to the servers IP address. Once we have our server host setup and we can reach it we would write the application code which is the web server software that will sit on the host. The last step is to deploy the application code on to the computer which is our server host.

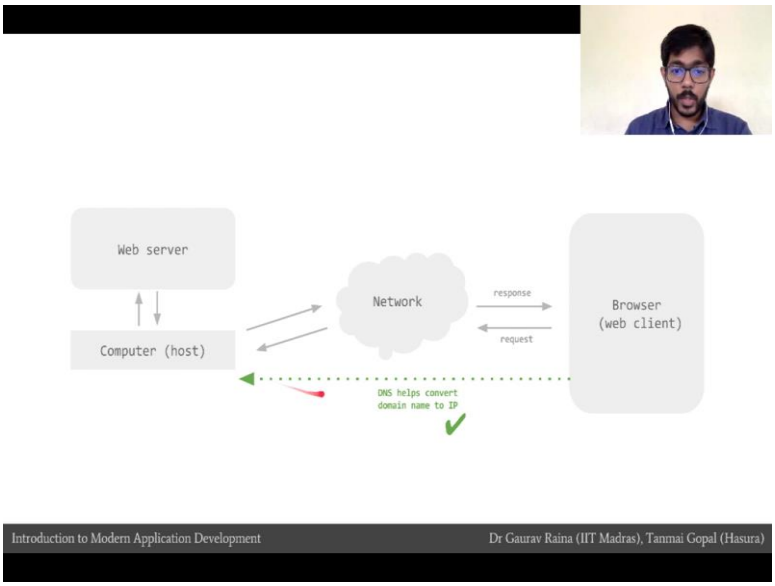
(Refer Slide Time: 01:00)



Let us look at the first step of buying a domain name. We typically buy a domain names from domain registrars. Registrars take our domain name and ensure that no one else including other registrars can use that name while we have it.

For example, in this particular screenshot, I have searched for the domain name my webapp and the registrars console is telling me that my webapp.com is already taken. It is often is suggestions to me saying why do not you purchase my webapp.in or my webapp.co.in or the netapp.com. Some popular domain registrars today are Gandi.net, Godaddy name.

(Refer Slide Time: 01:38)



Once we have a domain name, we can solve the first step of the problem. The browser can now reach our computer if the user remembers the domain name of our webapp.

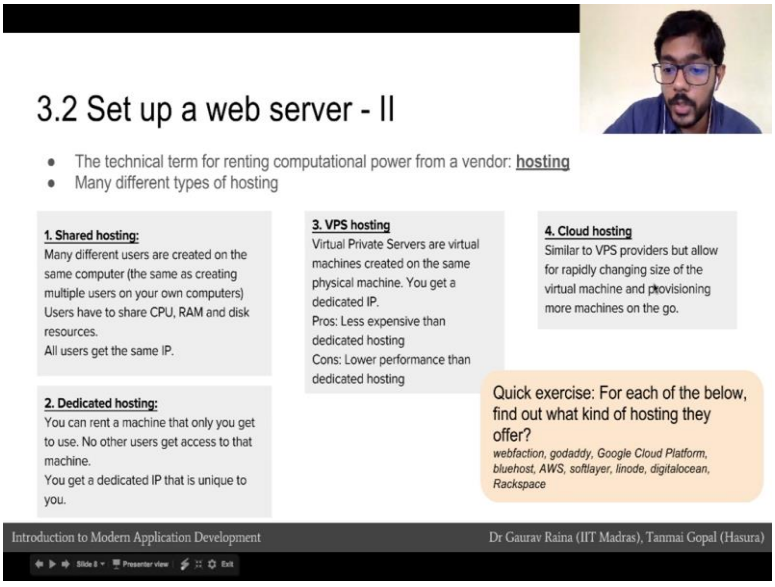
(Refer Slide Time: 01:48)

Step two, is to set up the web server. We need to get a computer and then IP. Now the first option that we have is that we can buy a computer we can connect it to the ISP, we can get a unique IP for that computer, unique IP that would not change it is called a Static IP. I would then install the operating system then the web server software on to this computer. Now this is very very hard to do. Imagine the amount of stuff that we would

have to learn about hardware, we would have to talk to the ISP's, we would have to ensure that we have power backup to make sure that our website is live 24 x 7.

Instead of doing all this we can just rent a computer from a particular vendor who might be anywhere in the world, and the vendor would give us a computer and would give us an IP attached to that computer which would let us reached our computer.

(Refer Slide Time: 02:34)



This business of renting a computer or renting computational power from a vendor is called Hosting. And there are many different types of hosting available. The most common and perhaps the cheapest kind of hosting is Shared hosting. In shared hosting many different users are created on the same computer. This process is quite similar to as creating different users on the same computer on our own machines.

For example, in your labs you might have multiple users who login to the same computer to do their own different work. The vendor does the same thing, the vendor creates separate users on the same physical machine and gives user access, and users have to share CPU, RAM and Disk resources. All users in this case get the same IP.

The second kind of hosting is Dedicated hosting. Unlike shared hosting where many users are created on the same physical machine, in this case the vendor gives their client or user a single machine to use. And in this case the user has dedicated access to the CPU, RAM and Disk resources of the particular machine. Dedicated hosting is

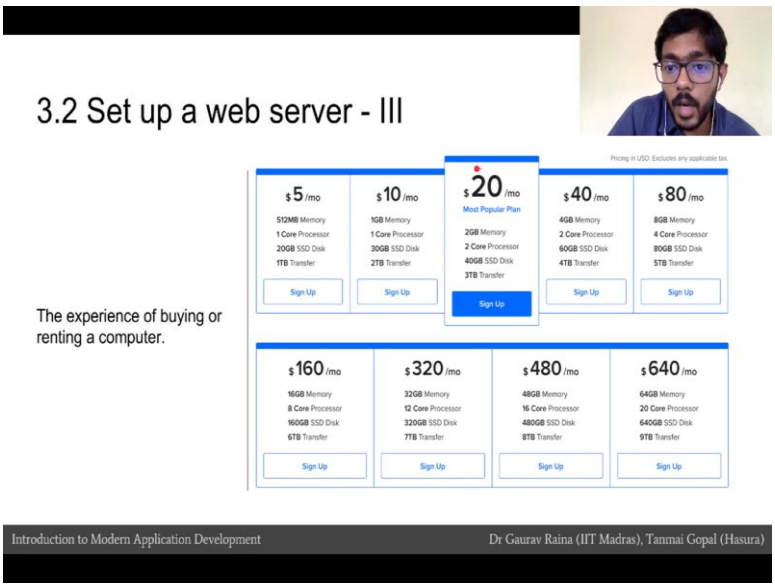
significantly more expensive then shared hosting. If we had go for dedicated hosting the vendor also gives us the dedicated IP that is unique to us and that is not shared by us with any other user.

The third kind of hosting which is become increasingly popular over the last few years is VPS hosting. Virtual Private Servers are virtual machines that are created on the same physical machine. It is like a combination of shared hosting and dedicated hosting, where we get the benefit of dedicated hosting which are dedicated IP and a guaranteed amount of resource. It is however like shared hosting because multiple virtual machines are created on the same physical machine. Hence, it is significantly cheaper than dedicated hosting.

Cloud hosting is very similar to VPS hosting, but in this case the vendor allows us to resize our compute resources dynamically. For example, we can change the size of the machines very quickly, we can provision new machines quickly, we can add more dedicated IP's for our machines quickly.

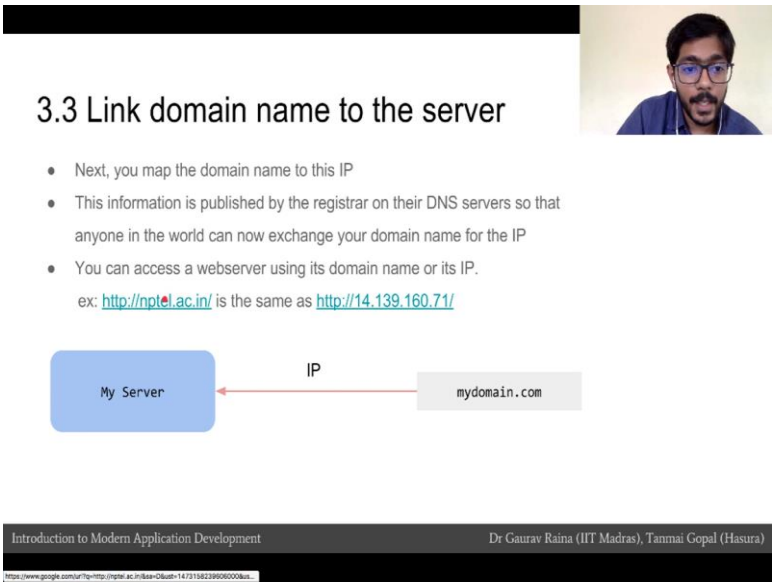
Here is a quick exercise, look at all of these hosting providers and figure out what kind of hosting they offer; webfaction, Godaddy, Google Cloud, AWS, softlayer, digitalocean.

(Refer Slide Time: 04:54)



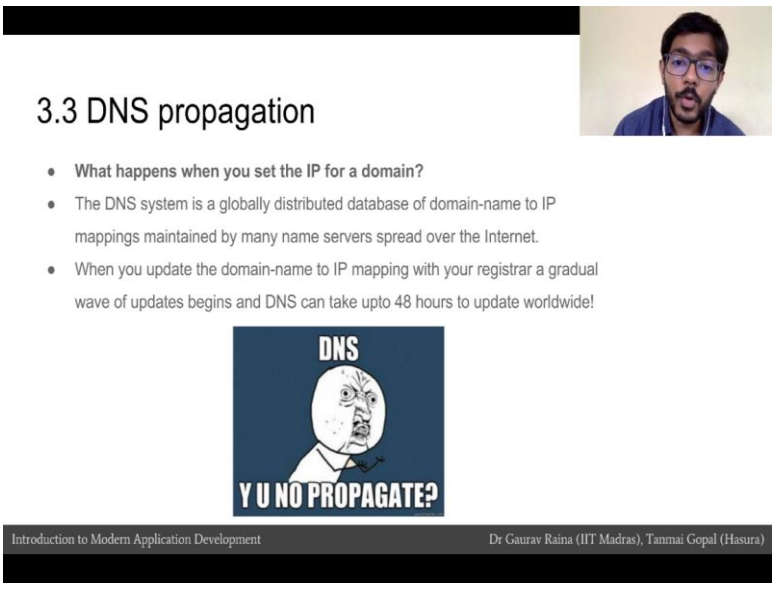
This is what a typical hosting providers console looks like; you can see that they are offering different computational resources for a different monthly rental.

(Refer Slide Time: 05:03)



Now that we have set up a machine the next step is to link our domain name to our server. This link is done on the DNS providers' console, but this is actually an optional step. Because even if you do not have a domain name and do not map the domain name to your IP you can still use the IP to directly contact the web server. For example in this case, instead of going to nptel.ac.in you could also just go to the IP and that would also serve the NPTEL website.

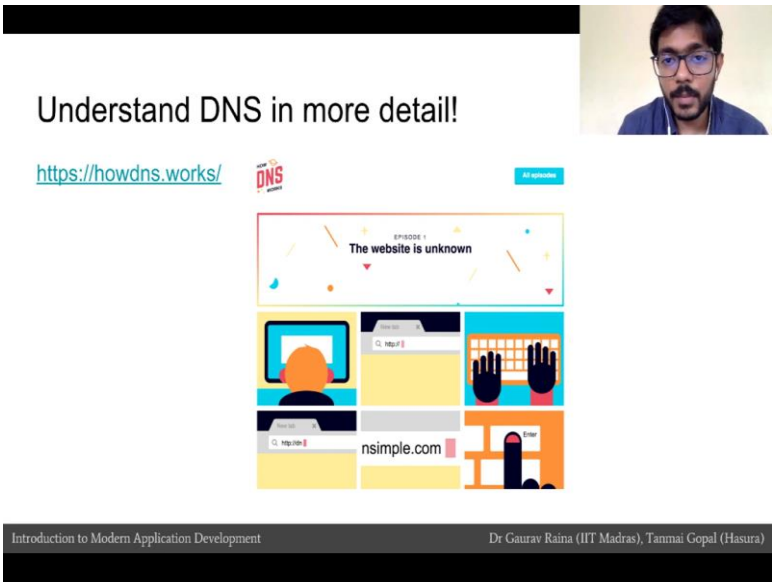
(Refer Slide Time: 05:34)



Once you do link the domain name to an IP, DNS has to update throughout the world. And so let us try to understand what happens when you set the IP for a domain. The DNS system is actually a set of globally distributed data basis that contain the domain name to IP mappings. And this is maintained by many name servers spread out over the internet.

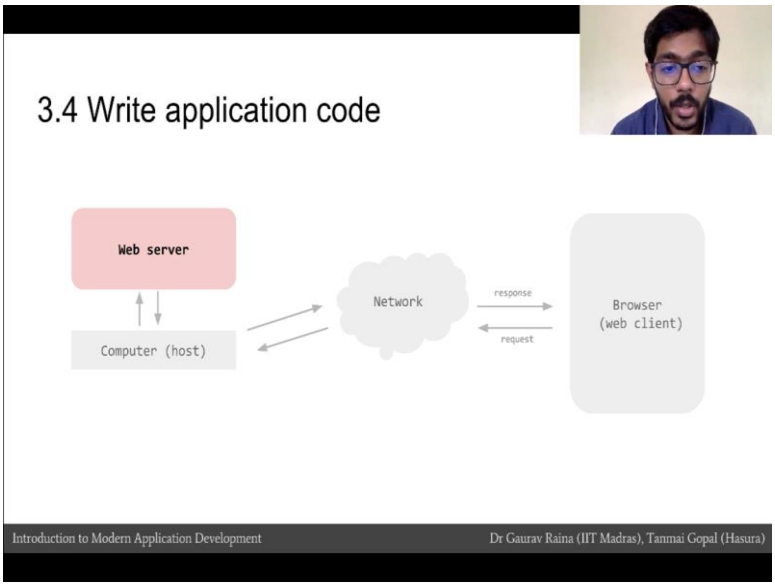
When you update the domain name to IP mapping with your registrar a gradual wave of updates begins and DNS updates are propagated throughout the world. DNS can sometimes take 48 hours to update worldwide, which is why very frequently when you are developing you will notice that you are not able to access your server with the domain name for sometimes even up to 4 or 5 hours till the DNS is updated.

(Refer Slide Time: 06:22)



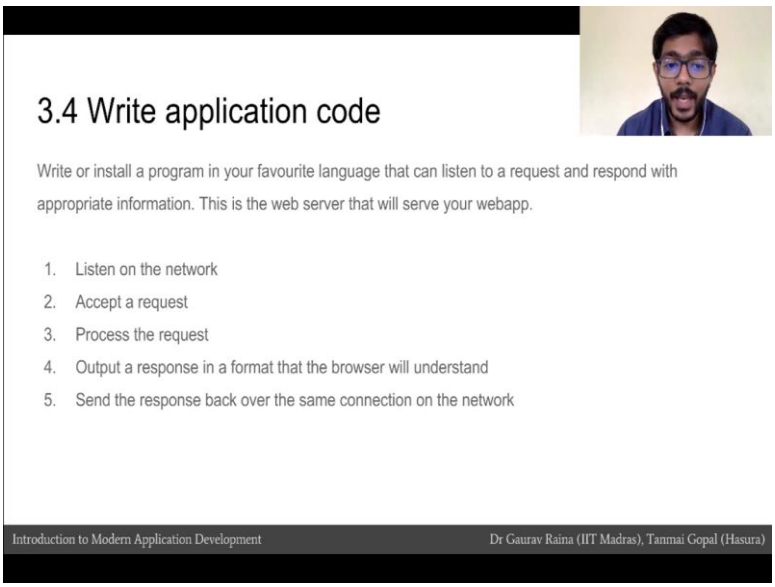
To understand how DNS works in more detail check this website out - howdns.works. This is a lovely resource to understand how your browser finally manages to get the IP of the webapp that you are trying to reach.

(Refer Slide Time: 06:36)



The last step now, is to actually write our application code which is the web server on our computer host.

(Refer Slide Time: 06:44)

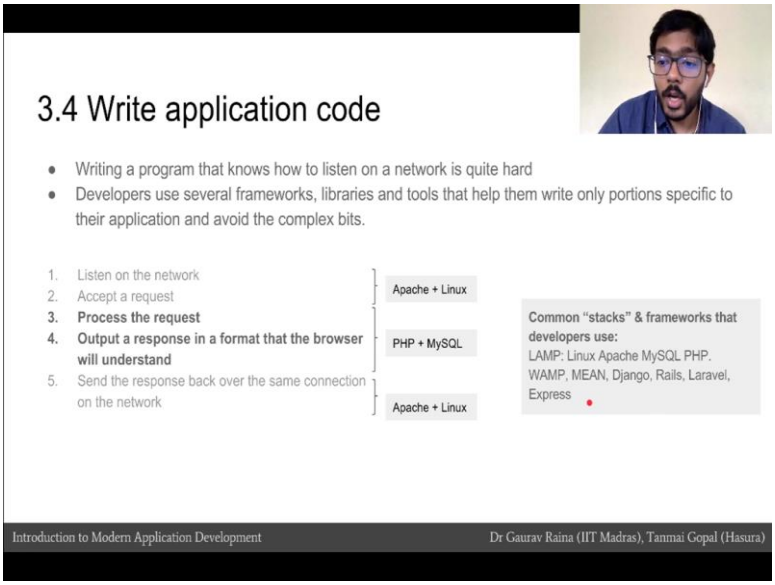


We will have to write or install a program in a language of our choice that can request or respond to the appropriate information. This software will be the web server that will serve our webapp.

This is the 5 things that a webapp does, it listens on the network, it accepts a request from the network, and it processes the request and then outputs of response in the format

that the browser or a web client can understand. It then sends that response back over the same connection on the network.

(Refer Slide Time: 07:12)



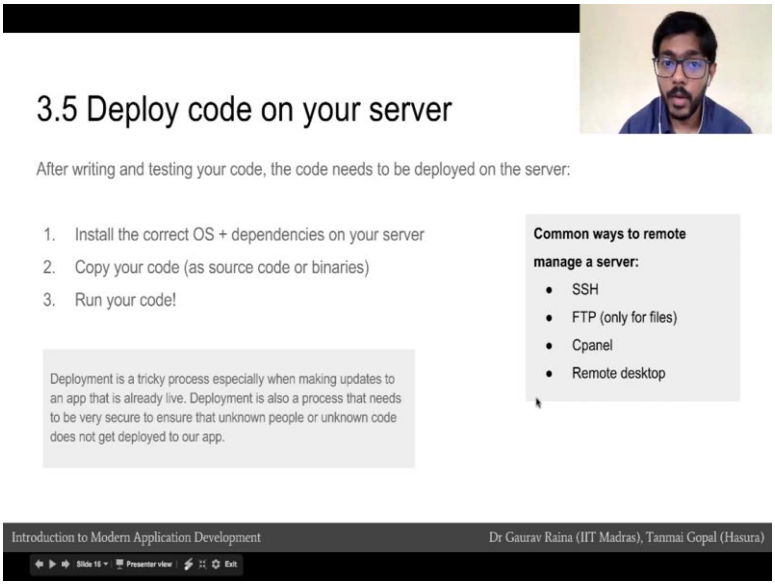
Now if you think about it this is quite complicated. Writing a program that knows how to listen to the network and how to respond in a network is pretty complicated. Developers often use several frameworks, libraries, and tools to help them write portions that are specific to their application and avoid writing the complex bits. For example, in our 5 step process, we only want to care about step 3 and 4, because that is what is unique by a webapp. Steps 1, 2 and 5 are common for all webapps and we should use a framework or a library to help this tool.

For example, the lamp stack is a very common stack that developers use. The lamp stack is composed of Linux, Apache, MySQL and PHP. Apache and Linux handle portions 1 and 2, PHP and MySQL handle portions 3 and 4, and Apache and Linux again handle the 5th portion. In this case Linux is your operating system, Apache is the web server, PHP is the language in which we write our application code and MySQL is the data base. We will go into detail and understand what each of these things are actually doing, but for now it is important to understand that the bit we want to deal with has webapp developers is only step 3 and 4.

Some other common stacks that developers use are the WAMP stack, which is windows Apache (Refer Time: 08:35) PHP, the MEAN stack which is the mongodb data base, the

express framework angular js and known Django, Rails, Laravel and Express are also common web frameworks that help us to achieves step 3 and 4.

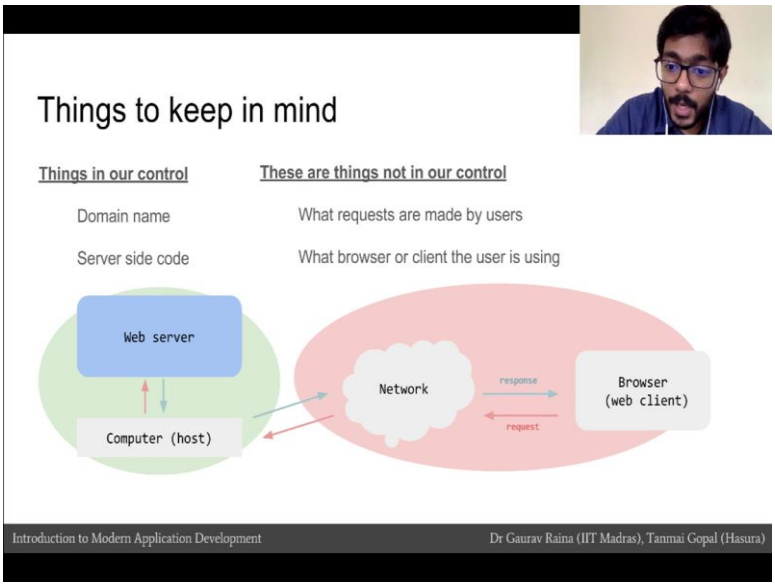
(Refer Slide Time: 08:47)



Once we have written our application code and we have chosen our web server, we have to deploy this code on to our server host. Now the server host is not a machine that we can access directly or physically. There are many ways to access a server remotely or remotely manages server. Some of the most common ways are SSH, FTP, Cpanel and Remote desktop.

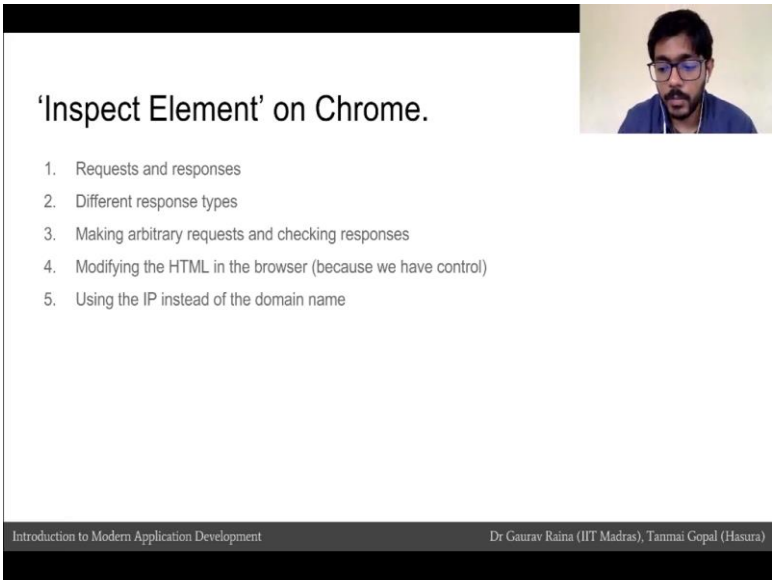
It is important to understand the deployment is a very tricky process, especially when you are making updates to an app that is already live. Deployment is also a process that needs to be very secure, because you do not want random people or you do not want to deploy a random code on to your server.

(Refer Slide Time: 09:26)



It is important to remember that there are some things in our control and there are some things that are not in our control. For example, the domain name and the server side code that we deploy on our server is in our control, but what is not in our control is the network; what requests browsers make, how they make the requests, and how they process the response.

(Refer Slide Time: 09:48)



Let us do up with quick exercise to see some of these concepts in action. So, among my Chrome browser on the Wikipedia website, I go to google.com and I open up a

google.com right click on the page and you will see inspect another window will pop up. Now go to the network tab on it is right element, you will see that a request is made to www.google.co.in and if you look at the response you will the some data has been sent back to the browser.

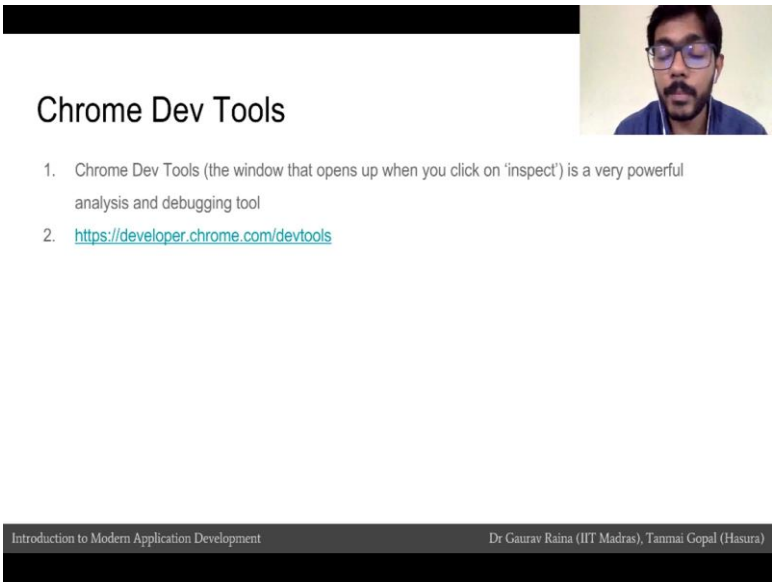
This data is in fact the html, and this html is understood by the browser and rendered. If you head over to the elements stack you will notice that this is the passed html where the browser has passed the response and understood the html and displayed their html as a web page. If you go over on html elements you will see that that is get highlighted.

Now, the interesting thing and the important thing to understand is that, once the data from google.com is on our browser that data is outside the control of the google.com server. For example, in this case this html element represents the lower; I can edit this html and remove the logo. In fact, I can go a step further, and insert a new html element. And you can see that this html element has now got inserted on the web page. Of course, this is the change that I have made temporarily on to my own and if I go back to google.com, you will notice that google.com serves a fresh page.

Let us take a look at what kind of (Refer Time: 12:17) and responses happen when you make a request to google.com. I refresh my page, and now you notice that a lot of request you can made. In fact, a total of 16 request have been made and 61 kilobytes of data been transferred from google.com to my browser, and the total load time is about 424 millisecond.

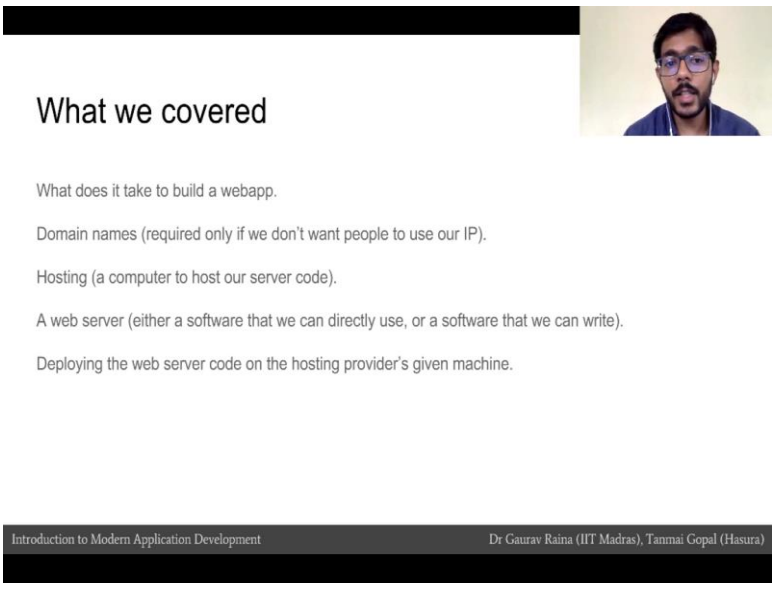
If you look at the various types of data that the google.com (Refer Time: 12:42) I responded with. Let us look at the type column and you can see that it is responded with many kinds of data, there are png images for example the Google logo, you click on this image and you can see the preview so that is in fact the image. In fact, I can take this image and open it up in a different tab and so I can get just the Google image that is being used on this web page.

(Refer Slide Time: 13:09)



I did in small experiments when we right clicked on the web page on chrome and clicked on inspect, the window that opened up is called Chrome Dev Tools. This is a very powerful analysis and debugging tool and you should definitely read more about it and use it as you serve the web.

(Refer Slide Time: 13:26)



Just to summarize quickly, in this session we took a high level look at, what it would take for us to build a webapp? We looked at domains, we looked at hosting, we looked at what it would take to build a web server, and what deployment also means.