Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #19
CGI Scripts

In this lecture, we would be talking about CGI scripts.

(Refer Slide Time: 00:57)

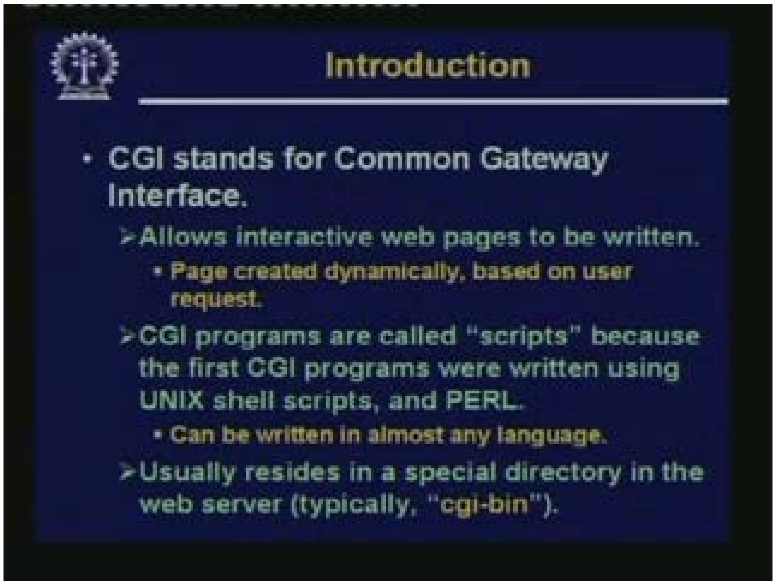
Indian Institute of Technology Kharagpur

CGI Scripts

Prof. Indranil Sen Gupta
Dept. of Computer Science & Engg.
I.I.T. Kharagpur, INDIA

Now we had been talking about the necessity for having CGI scripts. The necessity of having some kind of processing capability on the side of the web server. So that some kind of processing can be done at that end. We gave some typical examples that of submitting a form data. And the data going on the server side and some processing taking place on that side before the results coming back. So in this lecture we would talk about the basic technology behind CGI scripts. How we can write a CGI script program and what are the different things we need to remember when we are trying to write such a program.

(Refer Slide Time: 01:46)



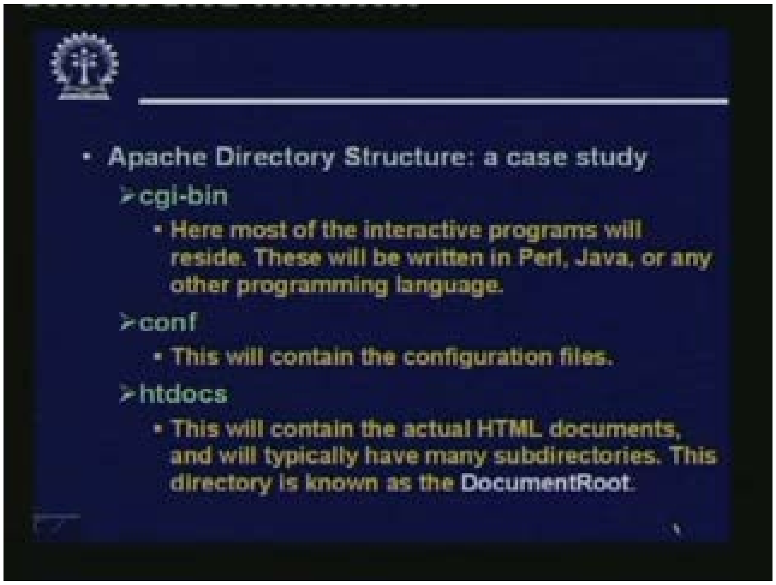
Introduction

- CGI stands for Common Gateway Interface.
  - ➢Allows interactive web pages to be written.
    - Page created dynamically, based on user request.
  - ➢CGI programs are called "scripts" because the first CGI programs were written using UNIX shell scripts, and PERL.
    - Can be written in almost any language.
  - ➢Usually resides in a special directory in the web server (typically, "cgi-bin").

The first thing is CGI is an acronym. CGI stands for Common Gateway Interface. CGI stands for Common Gateway Interface. Now in CGI we can do a few things. Now using CGI, we can create interactive webpages. We can also provide two way interaction and dynamic web page creation using CGI. Pages can be created dynamically. Of course this will be based on what the user wants. And one thing you may be thinking that the CGI's are nothing but some programs which are running on the side of the server. Why are they called scripts? They are called scripts due to traditional reasons. Traditional reason means that the first CGI programs that were written they were written using UNIX shell scripts or Perl scripts.

So for those of you who are not familiar with UNIX shell scripts let me tell you. UNIX shell scripts provide us with a very powerful string handling capability. It is a very small and simple language using which you can combine the UNIX commands in a very powerful way and you can do certain things in a very efficient way and repetitively. So UNIX shell scripts and Perl scripts were used to write the first CGI programs. But in actuality, CGI programs can be written in almost any language: C, Java, C++, anything. Depending on the server you are using CGI scripts or CGI programs will be residing in some special directory. Typically a directory called CGI bin is created under which these CGI script programs are stored.

(Refer Slide Time: 04:00)

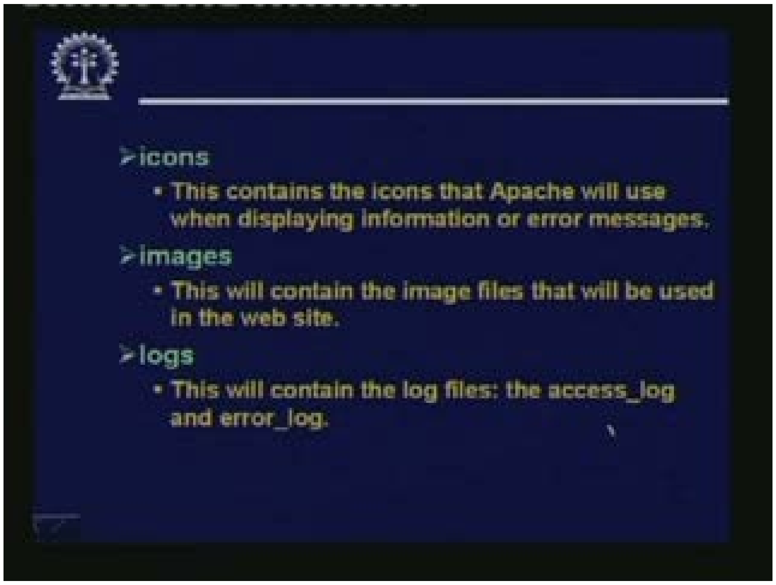


- Apache Directory Structure: a case study
  - cgi-bin
    - Here most of the interactive programs will reside. These will be written in Perl, Java, or any other programming language.
  - conf
    - This will contain the configuration files.
  - htdocs
    - This will contain the actual HTML documents, and will typically have many subdirectories. This directory is known as the DocumentRoot.

Now, let us talk of a very popularly used web server program apache. Now in apache, see apache is a web server which is freely available on the net. And I strongly recommend you should down load apache sometime and try to install it on your machine. When you install apache you will see a few things. There is a particular apache home directory that is created, that is called the apache root. And under the root directory some default subdirectories get created. Let us see what are the different subdirectories that get created in apache. Well, one is of course the CGI bin that I had mentioned. So under CGI bin most of the server side or interactive programs will be residing. These programs may be written in Perl, java, scripting language or any other programming language. There is another directory called conf which will contain some apache configuration files.

There is another directory htdocs under which all html and other documents will get stored. This htdocs usually will contain many subdirectories. This directory is called the document root. Document root means say in many cases when we access website, we give something like www.yahoo.com. We do not give any path or any particular file name after that. So what this means is that I am trying to refer to a default file in the directory root of the particular server. So here when apache is installed, a directory root gets created and if any one wants to access it default file typically default file is named as index.html or default.html in some cases. You need not have to specify that that will be taken by default so htdocs is there.

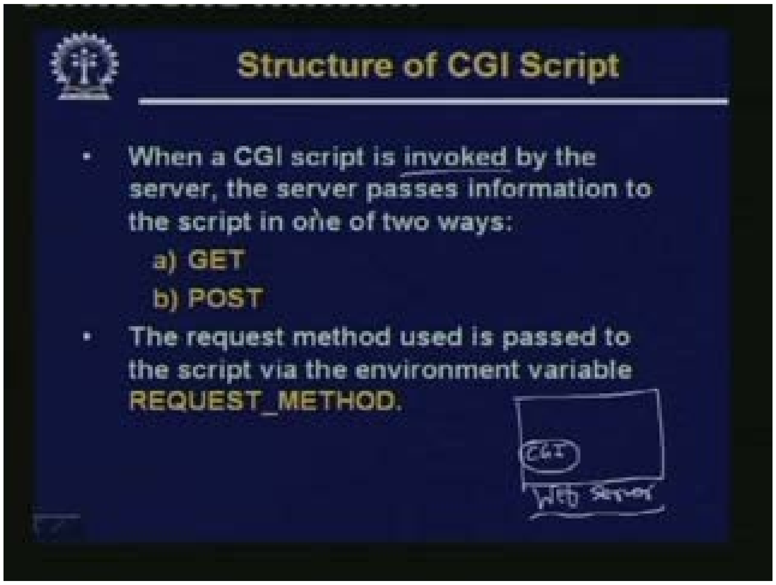
(Refer Slide Time: 06:26)



- icons
  - This contains the icons that Apache will use when displaying information or error messages.
- images
  - This will contain the image files that will be used in the web site.
- logs
  - This will contain the log files: the access_log and error_log.

Then there are directical icons, where all these small images or icons which apache uses are stored. Then there is another folder which is called images this can contain all the images files that will be used in the web site. And of course, there is another directory called logs which will contain all the log files information about the accesses and also about the error. Messages we have got.

(Refer Slide Time: 06:58)



Structure of CGI Script

- When a CGI script is invoked by the server, the server passes information to the script in one of two ways:
  a) GET
  b) POST
- The request method used is passed to the script via the environment variable REQUEST_METHOD.

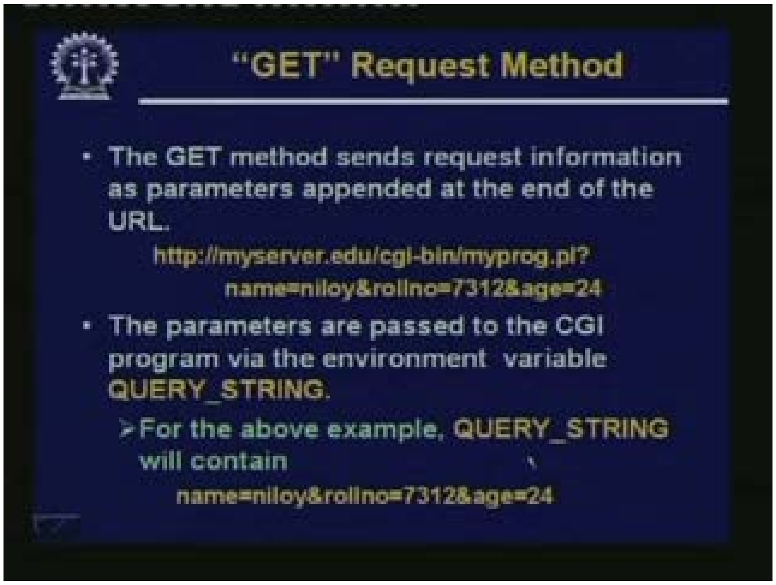
CGI
Web server

Now let us come back to the structure of a CGI script. Now when a CGI script is invoked by the server. Because, CGI script is a program which is stored on the web server. This is the CGI script and this is the web server. So CGI script is normally not executing the web server can wake it up. It can invoke it. So when the CGI script is invoked by the server, the server has to pass some information to the CGI script. Those are the input data to the program. Now input data can be passed in two ways GET and POST. GET and POST, we have already seen earlier in the context of http protocol we have seen that. So exactly very the same thing happens here http also we are trying to send something to the server in a particular way. Here also we are doing the same thing get and post tells you exactly in what form the data are coming to the server.

Now whether we are using GET or POST which is called the request method, this can be known from the environment variable called request method. Here let me talk a little bit about environment variable. See, you know when you are writing a program you can use variables. You can store some values in the variables if you are using a shell script or a shell programming if you are familiar with UNIX. You can define shell variables and you can assign some values to it. But these values are all local when as long as your session is continuing, you can access these variables their values. But some other user, you log out again login you will not get that values anymore. In contrast environment variables are those which some sort of you can say they do not get destroyed they remain permanently until or unless you change them.

Suppose I assign some value to an environment variable that will remain as long as the machine is running. So I can log out, I can again log in, I can get that same environment variable. Here when we are talking about passing some values from one program to another. Actually we are talking about environment variables. CGI script is a separate program some other program will be forwarding some value to it. It can do it through the environment variable. The web server understands the request method from the method attribute of the possibly and it can set the environment variable accordingly. The CGI script program whenever it requires, it can read the value of the environment variable and it can it can find out which method is being used GET or POST. So here the request method environment variable is used.

(Refer Slide Time: 10:32)



"GET" Request Method

- The GET method sends request information as parameters appended at the end of the URL.

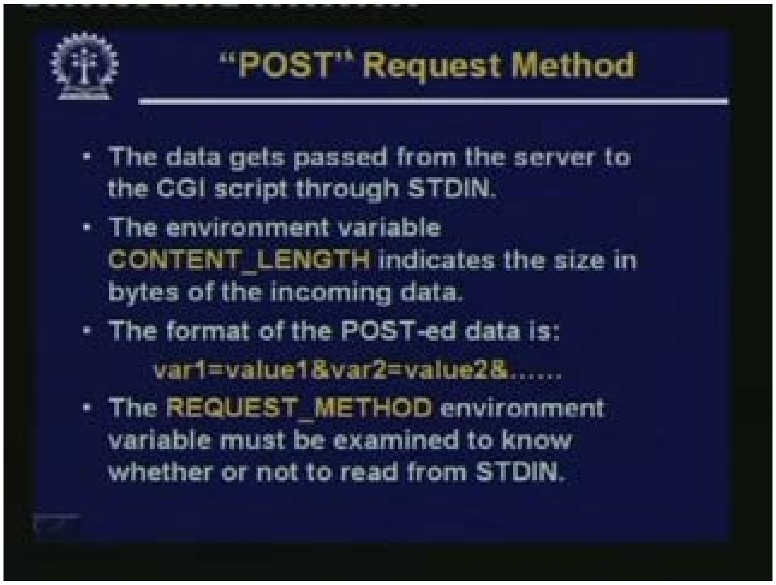
  http://myserver.edu/cgi-bin/myprog.pl?
  name=niloy&rollno=7312&age=24

- The parameters are passed to the CGI program via the environment variable QUERY_STRING.
  - For the above example, QUERY_STRING will contain
    name=niloy&rollno=7312&age=24

Now let us look at the detail of the GET request method. The get request method sends request information as parameters appended at the end of the URL. So if you have defined a form with method equal to get you have filled up the form and submitted exactly what goes to the server. What goes to the server is this. In the first part you have the URL of the server and the path of the CGI script program which was there as part of the form definition from where you have clicked submits. So this will tell you exactly what is the name of these server side CGI program prog.pl followed by a question mark. After question mark the values that you had entered in the form will come. Suppose in the form there were three forms name, roll number and age. Suppose you have filled them up and submitted and you if you can assign some names to each of these form elements. Suppose the names were name, rollno and age.

So the values will be going to be web server in this way name equal to niloy, this is what you have entered. Ampersand roll number is equal to 7312. This is what you have entered and age equal to 24. So whatever you have entered will be going as a string, concatenated string of name and value pairs, name equal to value and name equal to value and name equal to value and so on, as a single string. Now when this kind of a request goes to the webserver the web server receives it identifies this question mark and whatever is there after the question mark that gets stored in another environment variable called query string. For this example query string will contain this particular value. So you can understand if this CGI program now wants to read the data that was entered in the form it can directly read it from the query string environment variable, it stores a string containing the name value pairs with equal to and ampersand acting as delimiters.

(Refer Slide Time: 13:31)



"POST" Request Method

- The data gets passed from the server to the CGI script through STDIN.
- The environment variable CONTENT_LENGTH indicates the size in bytes of the incoming data.
- The format of the POST-ed data is:

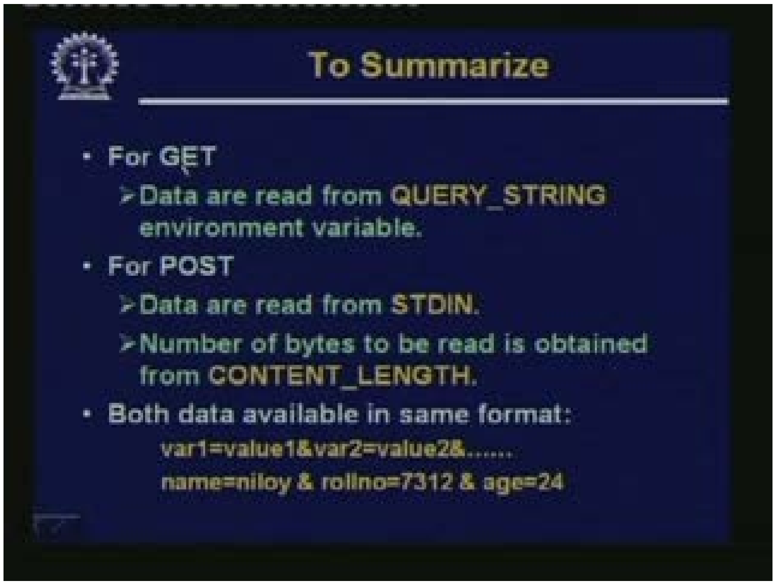
  var1=value1&var2=value2&......
- The REQUEST_METHOD environment variable must be examined to know whether or not to read from STDIN.

Before coming to POST, let me tell you a little bit about get one more thing. In GET you are getting the data as a string which is stored in an environment variable called POST. Now the difficult here is that in most machines there is a maximum length to the size of the string you can use. This I had also mentioned earlier, so the use of GET is very limited if you have only few number of parameters and their values are small in a form. Then only you can use; otherwise it can cross that limit of 256 characters and you can lose some information. Post is more widely used. In POST there are some differences. Here, the web server does not store the data in some environment variable. Rather it makes some arrangement, so that the CGI script can read the data from the standard input. This means that if you are writing this, CGI script program in C. Then you are you can use a normal scanf statement to read the data.

So as if the data you are reading from the standard input, in the same way you can do it. But you need some information like you need that the size of the incoming data. How many total characters are there in the string that you want to read from this standard input? So this is stored in an environment variable called content length. See here also whatever you are reading from the standard input, if format is the same name equal to value ampersand name equal to value ampersand and so on. So you can read it character by character in whatever way you want you can read it and you can parse it. The thing is that you must know that how many characters are there in the total list. That is stored separately in another environment variable called content length. Now here you can understand why I had mentioned earlier that this CGI script programming requires good string handling capabilities.

Because you can imagine that if you want to write this program in C, you need to do a lot of things. You need to read the characters one after the other you need to see where there is an equal to sign. That means the name has ended and the value is starting. Again you have to read it character by character wait for the ampersand. If it is a number you will have to convert it into a value and so on, languages like Perl and some other languages have a very simple facility to read them directly. We will see that later. So as I mentioned the format of the posted data which can be read through this standard input is like this. Var, variable 1 equal to value 1, ampersand variable 2 equal to value 2 and so on. So when you are writing a generalized CGI program it is good to read the request method to understand or to know that from where you are expecting the input data to come from the query string environment variable or from the standard input. Depending on whether the value is GET or POST the request method can be used to take this decision.

(Refer Slide Time: 17:21)



To Summarize

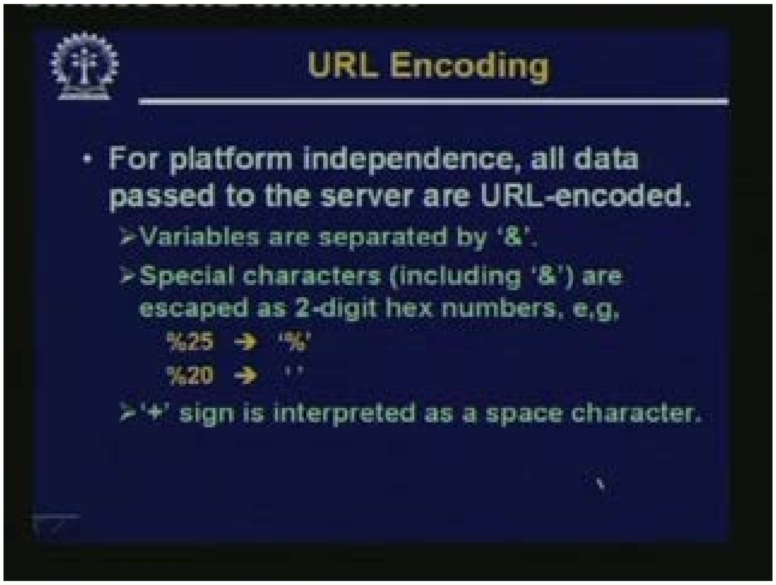
- For GET
  - Data are read from QUERY_STRING environment variable.
- For POST
  - Data are read from STDIN.
  - Number of bytes to be read is obtained from CONTENT_LENGTH.
- Both data available in same format:

  var1=value1&var2=value2&......

  name=niloy & rollno=7312 & age=24

So to summarize whatever we have said, for GET, we can read the data from the query string variable. For POST you can read data from the standard input. But the number of bytes we can get from content length. However in both the cases data are available in the same format variable is equal to this one and in the example like this. So this is what you should remember.

(Refer Slide Time: 17:52)



URL Encoding

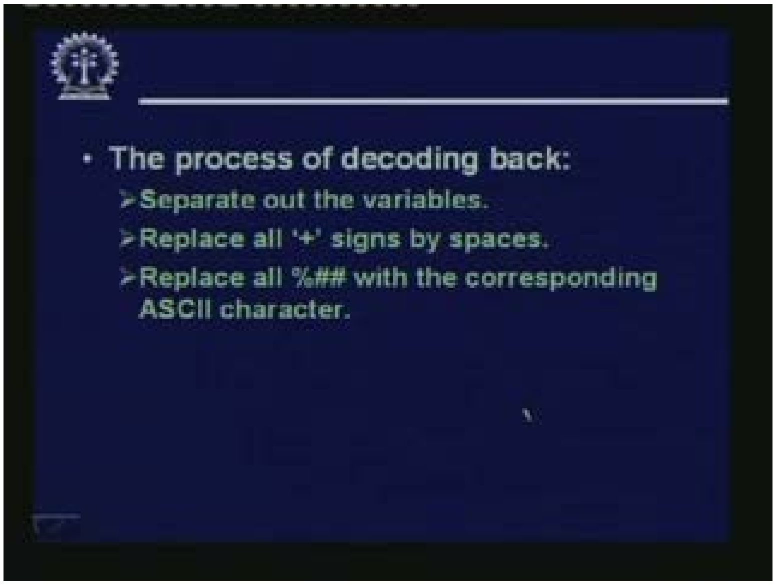
- For platform independence, all data passed to the server are URL-encoded.
  - ➢Variables are separated by '&'.
  - ➢Special characters (including '&') are escaped as 2-digit hex numbers, e.g,
    - %25 → '%'
    - %20 → ' '
  - ➢'+' sign is interpreted as a space character.

There is something called URL encoding that you needs to do whenever some something is being passed to the server. See, you click on the submit button something goes to the server. Even if it is a get URL question mark followed by something, name equal to value name equal to value and so on. Now in the value well I can type my name Indranil blank Sen blank Gupta. There will be blanks in my URL. But normally in a URL you cannot give blank spaces because blank is taken as a delimiter. So there has to be something which you need to do before the URL can be sent to the server. This is something called URL encoding. This is used mainly for platform independence.

Because in some platforms some symbols may have some special meaning. So if you use those special symbols as part of the URL, then some systems may recognize it properly. Some system may not. So you should be careful. So URL encoding there are a few simple rules. Variables are separated by ampersand. This you already know. So this ampersand is a special character. So naturally the question arises, what will happen if this ampersand appears as part of my string. Suppose in the form there was an entry called movie name I have typed Tom and Jerry, that ampersand was there in between. So the receiver may incorrectly interpret that ampersand as the end of the name value pair.

So the special characters this ampersand is one of them as I had just mentioned they can be specified as escape sequences by their 2-digit hexadecimal equivalence. For some examples the percentage symbol can be replaced by percentage 25. See, percentage is the special symbol because this is the symbol you are using to escape this is the escape sequence character. Blank you replace by percentage 20, this is the ASCII equivalent of the blank. And sometimes instead of percentage 20 you can also use plus. Because in many cases this plus sign is also interpreted as a space character.

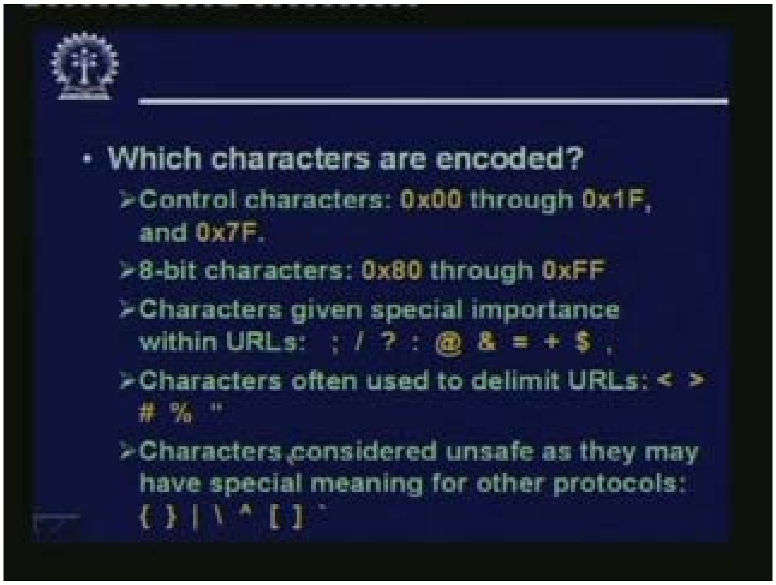
(Refer Slide Time: 20:40)



- The process of decoding back:
  - Separate out the variables.
  - Replace all '+' signs by spaces.
  - Replace all %## with the corresponding ASCII character.

So when you are decoding back at the server side the process is just the reverse. You can separate out the variables, you can replace the plus sign by spaces; you can replace all percentage hash, hash with their corresponding ASCII characters. But now the question arises that what are my special symbols, what are the characters that need to be escaped replaced by their escape sequences? In fact we will see that there are many characters need to be replaced by the escape sequences. That is why this URL encoding is important.

(Refer Slide Time: 21:18)

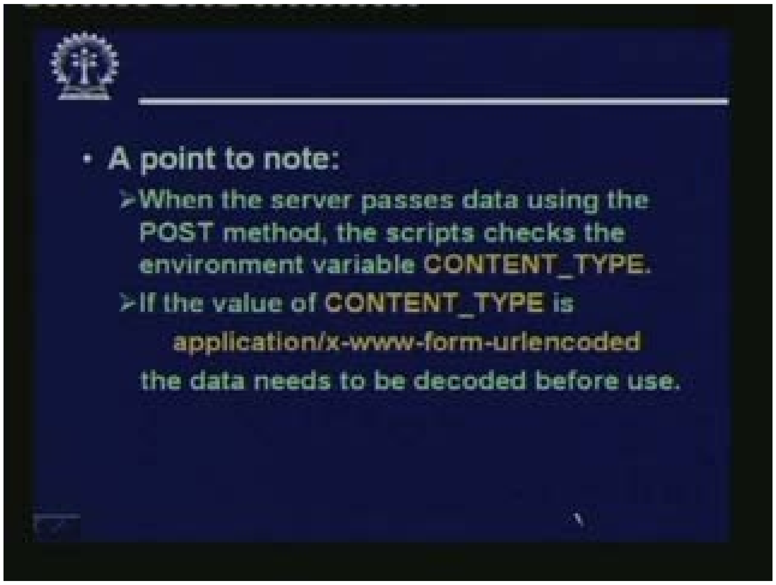


- Which characters are encoded?
  - Control characters: 0x00 through 0x1F, and 0x7F.
  - 8-bit characters: 0x80 through 0xFF
  - Characters given special importance within URLs: ; / ? : @ & = + $,
  - Characters often used to delimit URLs: < > # % "
  - Characters considered unsafe as they may have special meaning for other protocols: { } | \ ^ [ ] `

Which characters are encoded? Firstly, well here I am talking about ASCII, this 0x00 means in hexadecimal 0 0 up to 0 x1f. 1f in hexadecimal 00 to 1f. These are control characters they will have to be en encoded by the escape sequences because they cannot be represented by any printable characters and as a special character 7f. Normally, we work with 7bit ASCII characters. We do not use the MSB, Most Significant Bit is used for parity or it is made 0 permanently. But if you are using 8 bit characters, then we can use some special symbols by utilizing the 8 bit also. So those 8 bit characters are normally not on the keyboard are not easily printable; some systems may not recognize them. So they will have to be escaped also. Characters are given special importance like I told you ampersand, question mark, semicolon, slash, dollar, there are many such plus they will also have to be escaped.

Some characters which are used to delimit URLs like less than, greater than, percentage, hash, double quote, they will also have to be escaped some characters are considered unsafe. Because, there are some protocols for example MIME and other protocols Email, SMTP, MIME. There some other characters may have some special meaning, so they will also be escaped. So actually you see that other than the few alpha numeric characters most other characters need to be escaped. So actually you see that other than the few alpha numeric characters, most other characters need to be escaped because they have some special meaning in some context or the other. So this URL encoding is an elaborate process many of the characters need to be replaced by their by their ASCII or Hexical equivalence.

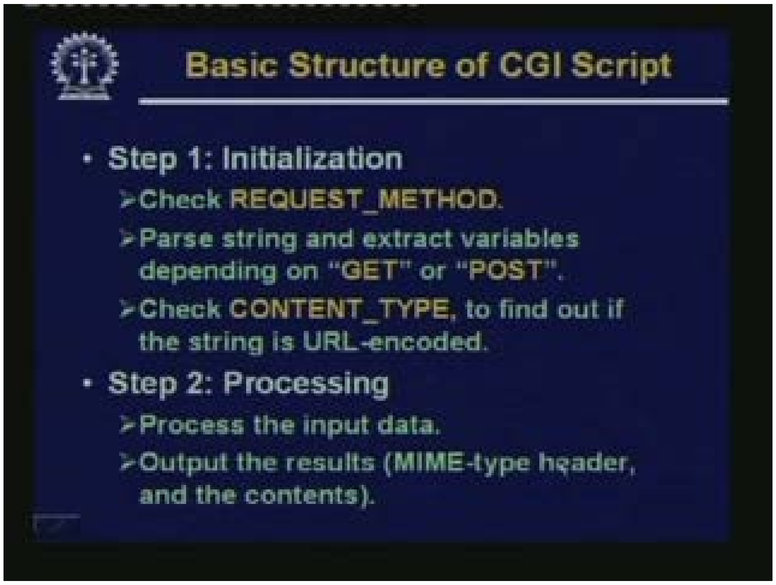
(Refer Slide Time: 23:21)



- A point to note:
  - ➢When the server passes data using the POST method, the scripts checks the environment variable CONTENT_TYPE.
  - ➢If the value of CONTENT_TYPE is application/x-www-form-urlencoded the data needs to be decoded before use.

So there is one point to note when this server is passing the data using the POST method to the CGI script program. There is an environment variable again called content type. Content type will tell you that whatever the server is giving me, is it URL encoded? or it is in the proper form? I need not do anything with it. I need not have to decode it separately. So if the value of the content type is this particular string, this means that what ever is coming has been URL encoded and needs to be decoded. Application slash x www form URL encoded. This is part of the MIME header which comes. So it can it is straight away stored in the environment variable content type, so that you can know by looking at the value of the variable whether such an encoding has taken place or it is not required.

(Refer Slide Time: 24:34)



Basic Structure of CGI Script

- Step 1: Initialization
  - Check REQUEST_METHOD.
  - Parse string and extract variables depending on "GET" or "POST".
  - Check CONTENT_TYPE, to find out if the string is URL-encoded.
- Step 2: Processing
  - Process the input data.
  - Output the results (MIME-type header, and the contents).

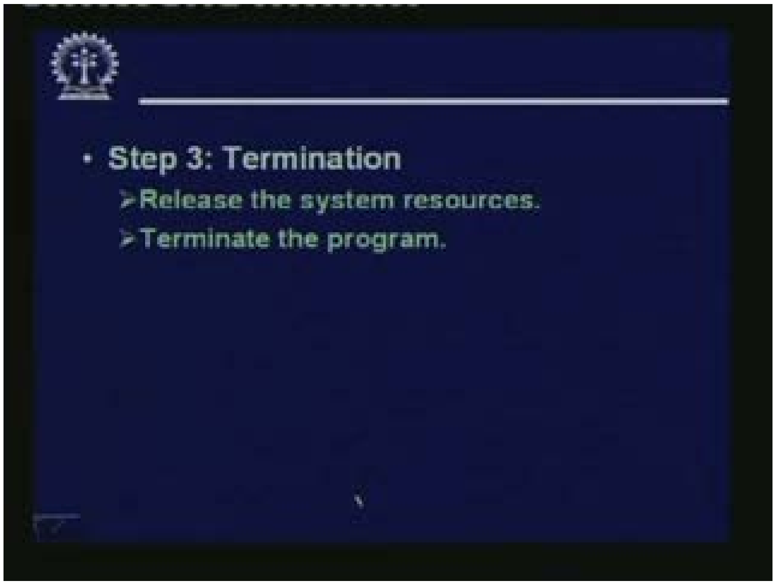
So the basic structure of a CGI script as we have seen it looks something like this. Now in the first step you do some kind of initialization. First you can check for the request method. Now the request method can be GET or POST. If it is GET, you will have to read the data from the query string environment variable. If it is POST, then you will have to read it from the standard input. If you are reading from the standard input the content length will tell you how many characters you need to read total. So depending on whatever you identify; the request method you can read the name and value pairs. But you will have to finally parse this string, the characters and extract the variables and their names.

Because ultimately they are coming as name equal to nillay, like that you will have to separate out name and nillay. What is the name of that form element and what is the value associated with that form element, those will be required for actually processing of the data inside your CGI program. Then you will have to check the content type variable to find out if you need to decode the string which is coming to if required. You will have to decode. In the second step we need to do some processing here you need to process the input data of course this depends on the context. So you are writing a CGI script program if it is a search engine, you may need to consult the database extract some data return it back to you.

If it is an email gate like say rediff, yahoo or Gmail, then you are typing some thing email address the body of the male subject then you are pressing go or send that is like submitting the form. So there again if the form is submitted the CGI script program will be sending the data to the email server for further processing and sending. So it really depends on the context in which you are actually using the CGI script. Is it ready and CGI script after processing the data we will half to output which will be coming back to the web server not to the client to the browser. Now the way the results are sent back they are sent back using MIME type headers and the contents.

So the data that will be sent back as the output from the CGI script will contain a header portion. That is the MIME type header which we had already seen earlier followed by the body. So if you are writing the CGI script in some language say C, then when you are outputting the result you need to make sure that you are outputting all the MIME headers everything in the proper format. If it is an html then the MIME header will tell you it is an html file. Then the actual body will contain all those html head body all those tags. So to be exactly in that format you will have to create or output the data in that particular format to be processed in a proper way.

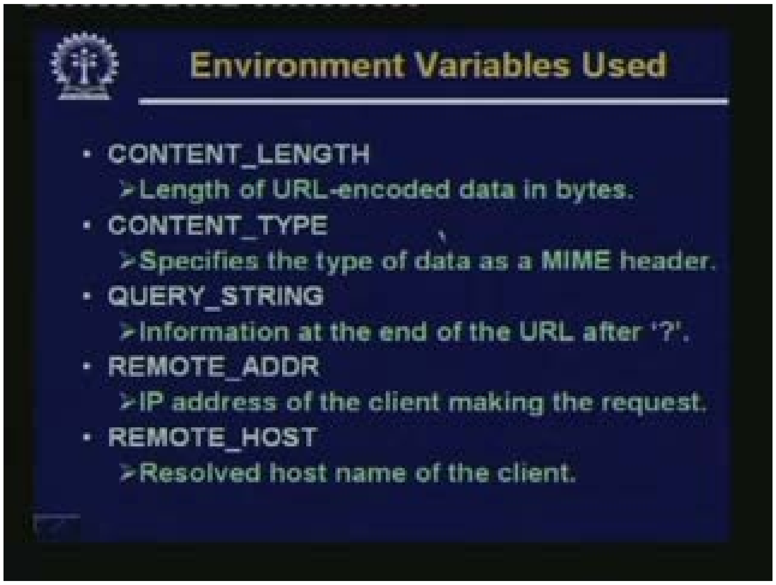
(Refer Slide Time: 28:25)



- Step 3: Termination
  - ➢Release the system resources.
  - ➢Terminate the program.

And of course the third step is the termination. After the CGI script finishes it releases the system resources, it terminates the program. Now this is how typically a CGI script program works. A CGI script program is residing on the web server, you submit a form the web server receives the form data, the web server invokes the CGI script program. The CGI script program can read the data do something and stop and if I submit the form again, the CGI script program will be woken up. And if there are ten simultaneous requests the older versions of the browsers and the servers they invoke multiple copies of the CGI script program and that increase the load on the server quite a bit. But of course, now a days with the sophisticated technology of threading and other things, modern browsers can use multithreading to create multiple threads to handle multiple user requests. So even the CGI script program can be threaded by the web server when new requests come.

(Refer Slide Time: 29:58)



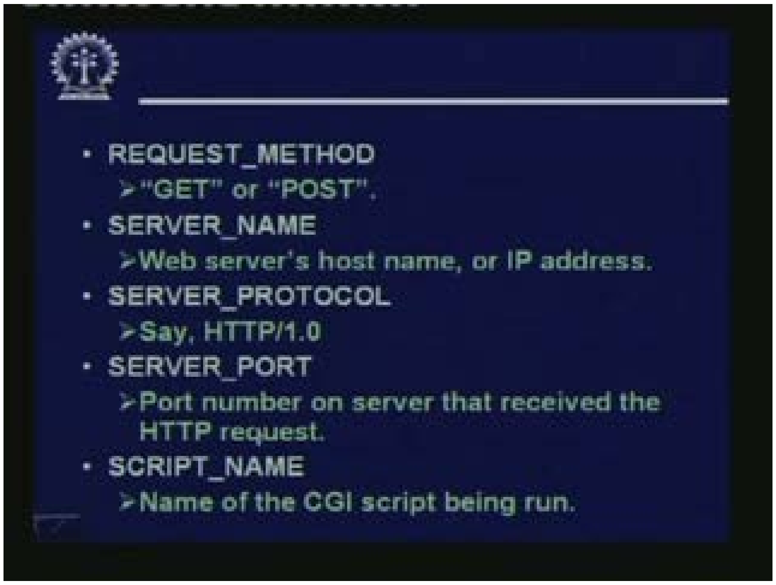
Environment Variables Used

- CONTENT_LENGTH
  - Length of URL-encoded data in bytes.
- CONTENT_TYPE
  - Specifies the type of data as a MIME header.
- QUERY_STRING
  - Information at the end of the URL after '?'.
- REMOTE_ADDR
  - IP address of the client making the request.
- REMOTE_HOST
  - Resolved host name of the client.

Now let us look at some of the environment variables that are used in conjunction with CGI scripts. Some of this we have already seen. For example content length. Content length is used in conjunction with the post method of submission of data. This gives you the length of the URL encoded data in bytes. See, whatever comes from in the standard input, the total length. It is not equal to the number of characters sent originally after URL encoding how many characters is there that is stored. So hash followed by number all those numbers will go on increasing then content type. Content type is a standard MIME header it can tell you that it is an html file it is a plain text or what type of content it is then query string.

Query string is used in conjunction with get here in the URL whatever is after the question mark that will get stored in the query string variable. There is some other environment variables which are there which you have not talked about. There is something called remote ADDR. This stores the ip address of the client which has made the request. Remote host is the host name of the client. Sometimes the server needs to store in some log that who is the person who has sent me the request which ip address from which machine the request has come. So these environment variables can be used for that purpose.

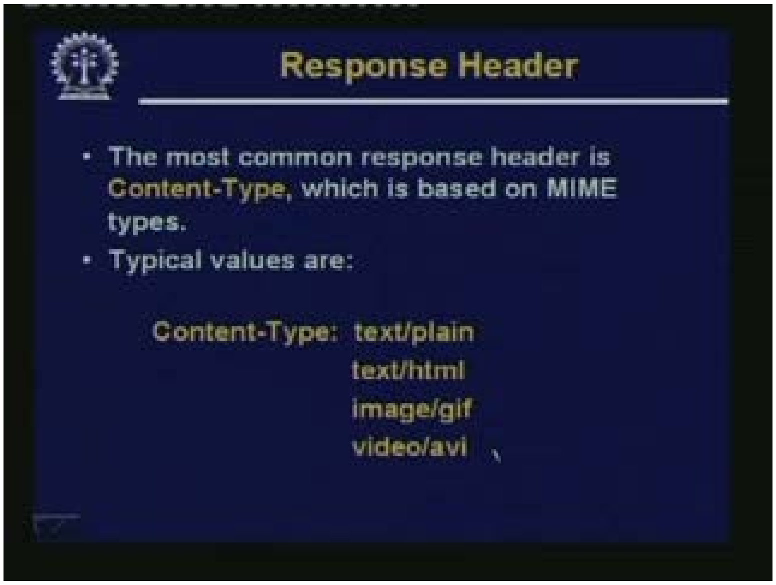
(Refer Slide Time: 31:43)



- REQUEST_METHOD
  - "GET" or "POST".
- SERVER_NAME
  - Web server's host name, or IP address.
- SERVER_PROTOCOL
  - Say, HTTP/1.0
- SERVER_PORT
  - Port number on server that received the HTTP request.
- SCRIPT_NAME
  - Name of the CGI script being run.

Of course request method GET or POST, server name, the web server's host name or IP address is also available. Server protocol which version of http is running server port on which port number on the server. This particular request has arrived and script name which CGI script is being executed.

(Refer Slide Time: 32:10)



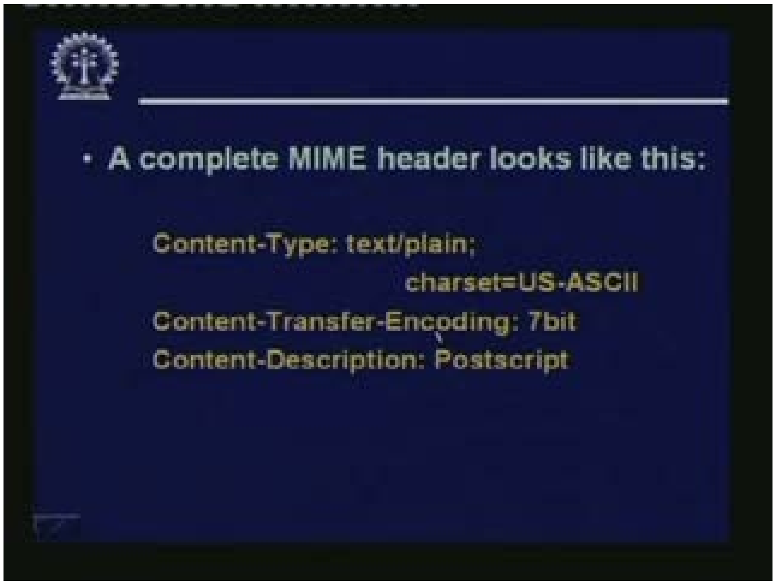
Response Header

- The most common response header is Content-Type, which is based on MIME types.
- Typical values are:

```
Content-Type:  text/plain
               text/html
               image/gif
               video/avi
```

Now, so many environment variables are there which you can use. Now coming to the response header. The most common response header is the content type. Because as I mentioned whenever the CGI scripts outputs the data back to the browser. There will be two part one will be the header part other will be the body part. The header will be MIME encoded. MIME encoded header looks like this. Content type colon text equal to plain text slash plain or text html or image gif or video, AVI or there are many other alternatives. So this content type is a very important MIME type which is present in the header which the CGI script will be generating as the output. So that the browser after receiving it can understand what type of data it is. If it says that it is text dot plain, then it will not try to compose it. It will display whatever it comes in text on the screen. If it is text slash html, then it is expecting that an html page will be coming back with all those tags, html, body; etcetera. So it will try to compose it as html page and so on. If it is an image jpeg, then it will try to display it as an image alright.

(Refer Slide Time: 33:45)

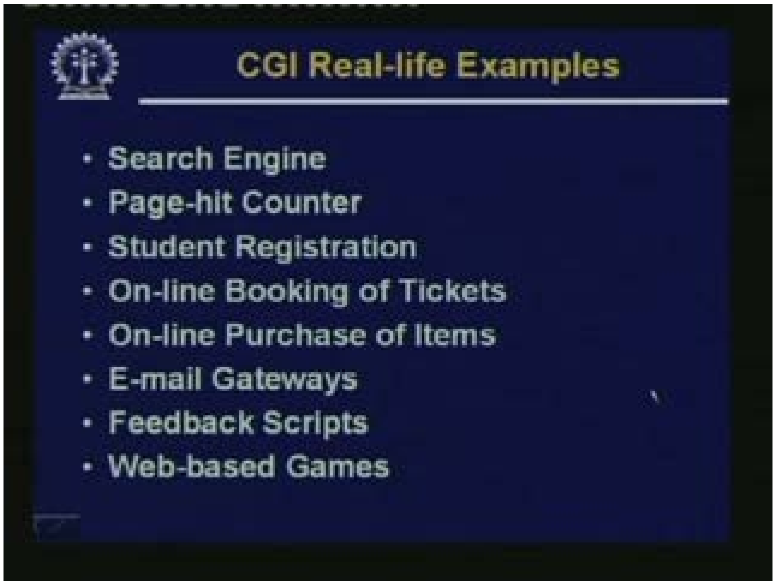


- A complete MIME header looks like this:

```
Content-Type: text/plain;
                    charset=US-ASCII
Content-Transfer-Encoding: 7bit
Content-Description: Postscript
```

So a typical complete MIME header may look like this. Content type, text plain character set us ASCII content transfer encoding 7bit, content description postscript. So depends on the application of the CGI script you may not give all the details. But at least you can give some minimum information which can be handled by the browser. So content type is one such minimum information which is required.

(Refer Slide Time: 34:19)



CGI Real-life Examples

- Search Engine
- Page-hit Counter
- Student Registration
- On-line Booking of Tickets
- On-line Purchase of Items
- E-mail Gateways
- Feedback Scripts
- Web-based Games

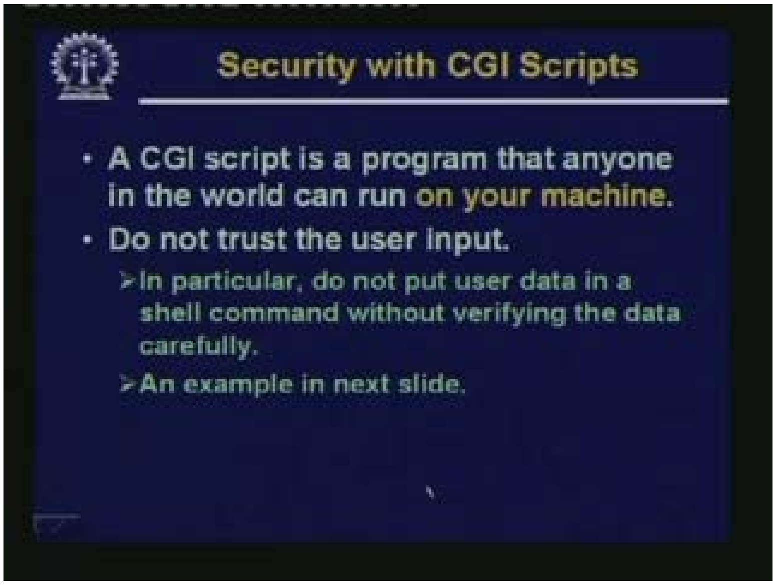
So let us look at some of the CGI real life examples. Some which we have already mentioned. Let us try to look at some of the salient points of this. Search engines I have already mentioned. So I need not spend much time on it you have a small box you type in the keyword you want to search for press the button search or go request goes back to the web server. Some database is searched, some data retrieved. They are composed as an html page and sent back to me. So, on my browser I can view it as html page that is the search engine application of a CGI script. Page hit counter is slightly more involved. See you may have encountered the case there are some websites where if you visit the web sites you will get some information that this page has been visited so many times.

So this is called a page hit counter. There are a number of ways to implement page hit counter. Page hit counter can be stored as some number in an environment variable or in a database which will be updated every time that page is visited or some form is submitted that you go on incrementing or updating. But there is one thing that you should understand. Page hit counter is a variable and what ever you are viewing that is an html page. See if it is a dynamically generated html page, then I have no issue I have no problem. Then while generating the html page the page hit value can also be included as part of the html page. But suppose I have a fixed page nothing else changes only one number at the bottom of the page changes.

How I can implement that without modifying my html page which is stored on the server. There are number of ways in which people have tried to do it. One way is very simple. There are small gif images corresponding to the digits 0 to 9. The CGI script composes an image based on the value of the page hit counter. Suppose it is 935, it puts 9, 3 and 5. And it composes an image. In the html file there is a hyperlink to that particular image, so that image gets loaded. And that image is continuously changed or modified by the CGI script program. So in this way the content of the image gets changed and you can see that count value is also changing. So this is an application. Student registration is very simple. There are forms you fill up and submit and information goes into the database.

Online booking and online purchasing are also very similar. There is facility through forms you can fill up relevant information and you can submit them. This email gateway I have mentioned. Feedback scripts any there are many websites you can have, you can also do it in your own website if you design one. You can have a box, a text box at the end where you can invite some feedback from visitors. Please write how do you like my page you type something and press submit that is like a page you have submitted like a form you are submitting. This will go to the html you can say html CGI script on the other side and the CGI script will wake up it will take back the value which is entered and may be it will store in the database. And web based games is another application where CGI scripts and similar technologies are also used.

(Refer Slide Time: 38:53)

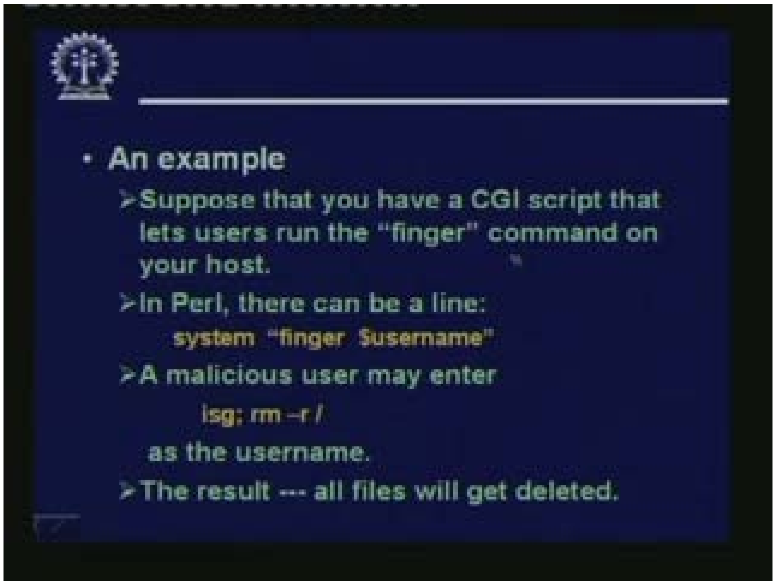


Security with CGI Scripts

- A CGI script is a program that anyone in the world can run on your machine.
- Do not trust the user input.
  - In particular, do not put user data in a shell command without verifying the data carefully.
  - An example in next slide.

Security is one problem with CGI script if you do not use it in a judicious way. Let us try to understand what this means. Suppose I have floated my web server. This is my computer. I am running apache on this computer. I have created some web pages. I have written some CGI script programs which are stored on my computer. Someone from anywhere in the world comes and accesses my page. There will be some forms in some of the pages, the user should filling up the forms and submitting them. Submitting means the CGI script program will be running and they will be running on my computer. So someone else sitting somewhere I do not know he or she is an anonymous entity for me is sending something and making some program on my machine executing. So I should be very careful about this. Why? This is related to the user input. Normally what is recommended is that, do not put user data directly in a shell command without verifying the data carefully. A very classical example and simple example is being illustrated next. Well, this example concerns a particular command that is available under the UNIX or Linux operating system called finger. Finger is a command which takes one argument finger user name like I can give a command finger blank xyz. If xyz is the name of the user on that machine, then I can get all the details of the user. What is the name of the user, the contact telephone number, last when he logged in, if he has left any message that he is out of the station out of office that also I can see. Normally finger command is used to know some status information about a particular user. May be in a web page I have kept a facility like that. I have kept a facility so that remote user can type the name of some user and my CGI script will run a finger command and give back the result to that user that, what is the status of that user. But this is a very dangerous thing. Why?

(Refer Slide Time: 41:41)



- An example
  - ➢Suppose that you have a CGI script that lets users run the "finger" command on your host.
  - ➢In Perl, there can be a line:

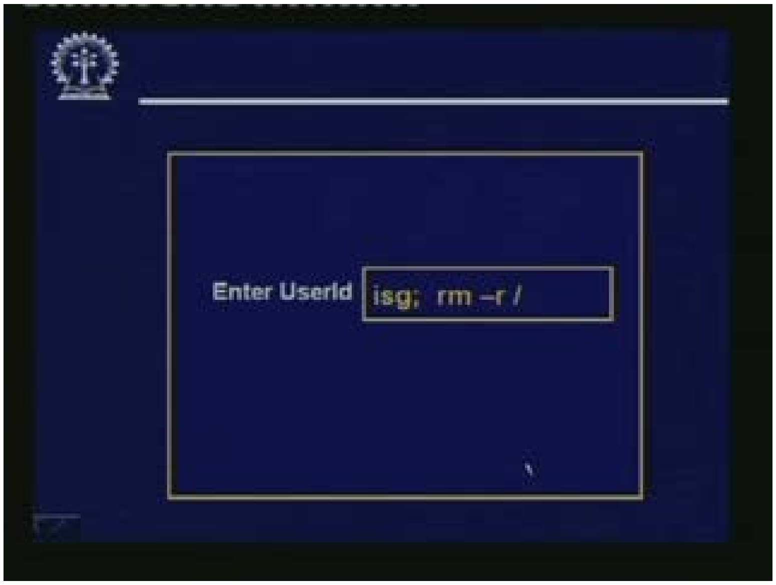
    `system "finger $username"`
  - ➢A malicious user may enter

    `isg; rm –r /`

    as the username.
  - ➢The result --- all files will get deleted.

So as I said that basically the CGI script is running the finger command, if you are writing the CGI script in Perl, well we will be learning Perl later. But let us see how statement in Perl looks like. System finger username this is a typical Perl statement. This means finger is a command which you are asking the operating system to run. That means the Linux or UNIX to run with a parameter dollar user name which is coming from the form. Dollar user name is something which you are getting or which you are reading from the form. So finger something that something is coming from the form. Now let us see what are the possibilities here? Normally in place of user name say we can give user name like isg. So the command will translate to finger isg and the remote user can see the details of the information about the user isg. But suppose the user is malicious and instead of typing isg the user types, this entire string, isg colon rm minus r slash. What does this mean? This means you are trying to execute a statement like this finger isg semicolon rm minus r slash. Now in UNIX, multiple statements can be given on the same line separated by semicolon. So actually this means there are two statements you are asking to execute. First one is finger isg that is fine. Second one rm minus r slash means from the slash means from the root you recursively go on deleting all the files. So this will delete all your files on the machine. So a remote user has been able to delete all the files in my machine. But this is because I was not careful I did not check what the user typed in place of the user name. Was it isg? Or something more after semicolon something else. So this is one very simple example which illustrates that whenever you are passing some form data directly to a script for execution, you need to be doubly sure that exactly what you are passing and what are different possibilities that you may encounter.

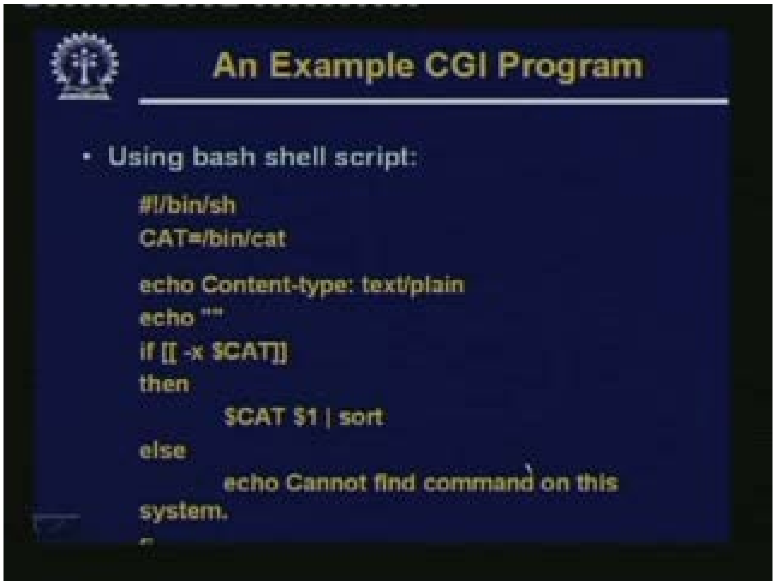
(Refer Slide Time: 44:23)



Enter UserId isg; rm –r /

So this is the example I told you.

(Refer Slide Time: 44:31)



An Example CGI Program

- Using bash shell script:

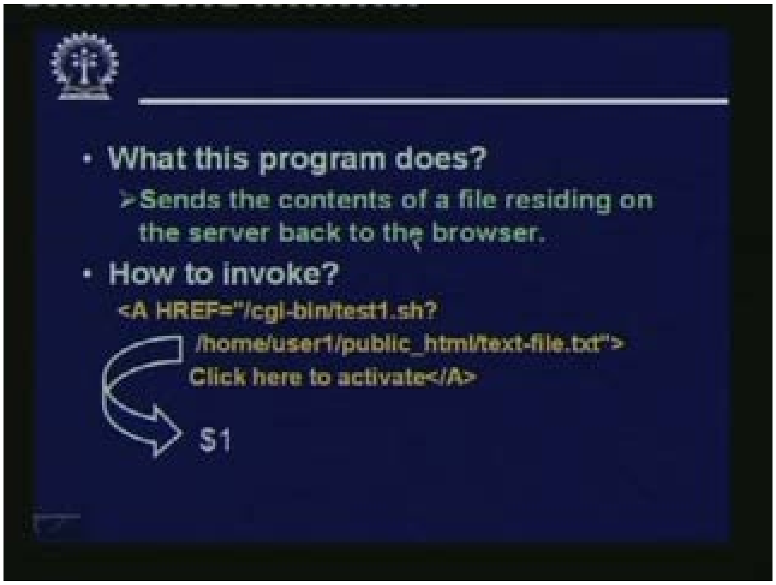
```
#!/bin/sh
CAT=/bin/cat

echo Content-type: text/plain
echo ""
if [[ -x $CAT]]
then
        $CAT $1 | sort
else
        echo Cannot find command on this
system.
```

Now let us look at a couple of example CGI programs. Well we had not talking about form data at present. Because form data will require some string manipulation that we shall discuss, when we talk about Perl. Here we talk about some simple shell scripts and using this simple shell scripts, how you can write a CGI program. The first example, this is the CGI script. First this is a variable called CAT where I am storing the string slash bin slash CAT for those who are familiar with UNIX in shell script Bourne shell or bash it will be easy for you to understand. For the others I ask that you should look into the syntax of the UNIX shell scripts like bash and you will be able to understand this.

This echo means displaying on the screen so you are displaying this content type colon text plain on the screen. Echoing null means a blank line on the screen. This statement means dollar CAT means the content of the CAT whether this CAT is an executable file exits minus x means whether this exists on the system whether this is an executable file bin cat; CAT means concatenate. This CAT command is sometimes used to view a file. If it exists then you give a command that CAT dollar one means the first parameter. This is coming from somewhere type sort. So whatever is coming, you sort it and then you display it on the screen else.

If it is not there you output an error message cannot find command on this system followed by end if fi. Fi means end if. Now if you see what is coming on in the output, the output will be content type text plain followed by a blank followed by content of some file. This is actually a MIME encoded file. There is a MIME header between MIME header and body that has a blank line that blank line is also there. So whenever this comes back to my browser I will see the contents of the file in textual form not composed as html. But in text form because it has been mentioned as text plain. But the question is how to invoke this?

(Refer Slide Time: 47:25)



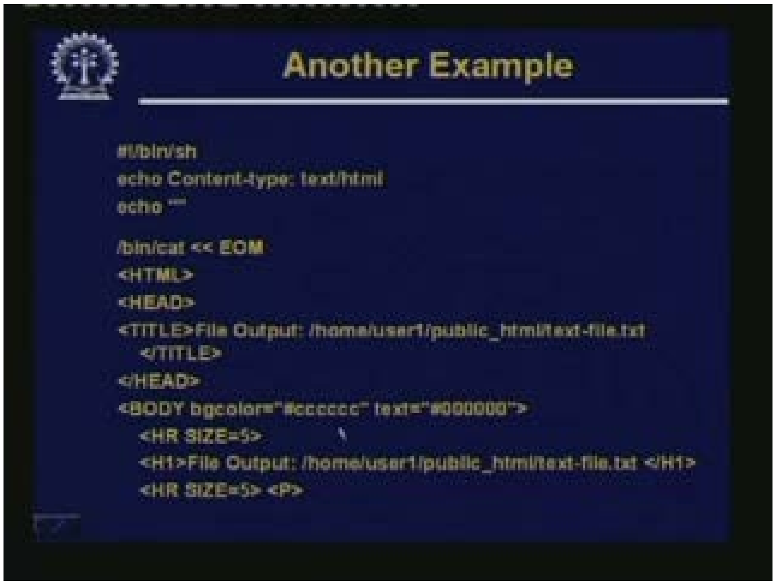
- What this program does?
  - Sends the contents of a file residing on the server back to the browser.
- How to invoke?

```
<A HREF="/cgi-bin/test1.sh?
    /home/user1/public_html/text-file.txt">
    Click here to activate</A>

    $1
```

This can be invoked as follows. This program as I mentioned, sends the contents of the file in the sorted order back to the browser. How to invoke? You can invoke it directly by specifying this URL in html file href. This is the name of the shell script. This means that it is a get method of submission you specify the name of the file. Here you specify which file you want to display of course this is the hyperlink. Here I have shown example. Well this first parameter, this will be treated as that dollar one in the previous example. If you go back this dollar one which over is whatever is they are here, this is the first parameter.

This will be actually this. This entire thing will be taken as dollar one, the particular file on the server which you want to execute. Now the same thing you can also invoke using a form with a get method. It is a dummy form. There are no forms to fill up or you can have one form where the entire path names of the file you are filling up then submit. This same thing will happen. It is getting submitted the shell script can access the first parameter of the form as dollar one and it simply discuss the content of the file on screen.

(Refer Slide Time: 49:08)

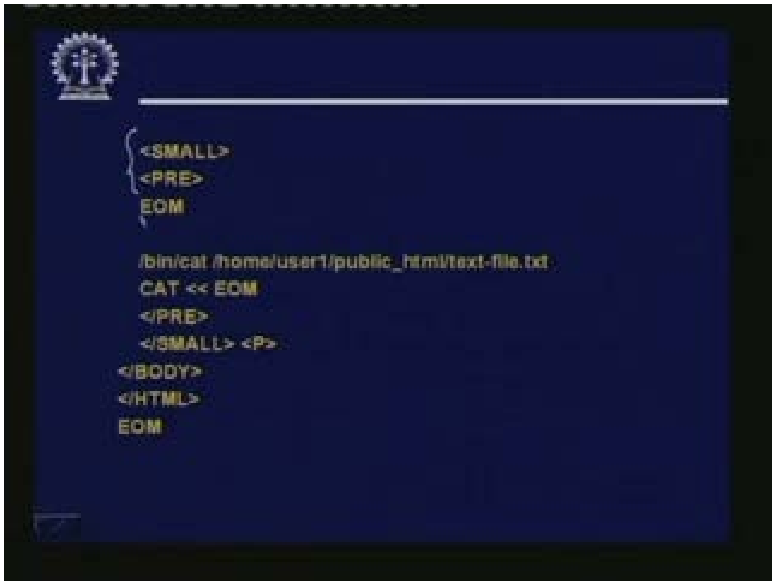


Another Example

```
#!/bin/sh
echo Content-type: text/html
echo ""

/bin/cat << EOM
<HTML>
<HEAD>
<TITLE>File Output: /home/user1/public_html/text-file.txt
  </TITLE>
</HEAD>
<BODY bgcolor="#cccccc" text="#000000">
  <HR SIZE=5>
  <H1>File Output: /home/user1/public_html/text-file.txt </H1>
  <HR SIZE=5> <P>
```

Let us take another example which is little more complex. Here we are composing as an html file not as a text. Here we are outputting content text html in a blank line. Now since we are trying to output html, we have to output all the tags of the html also this you are doing here. See this statement bin CAT less than less than a means, you simply concatenate all the statements which follow blindly until you encounter a string called EOM. This you could replace by CAT, CAT, CAT on each of these lines it makes it much shorter. So here you put it on each and every of the lines. This entire thing gets displayed. This entire thing gets outputted bin cat.

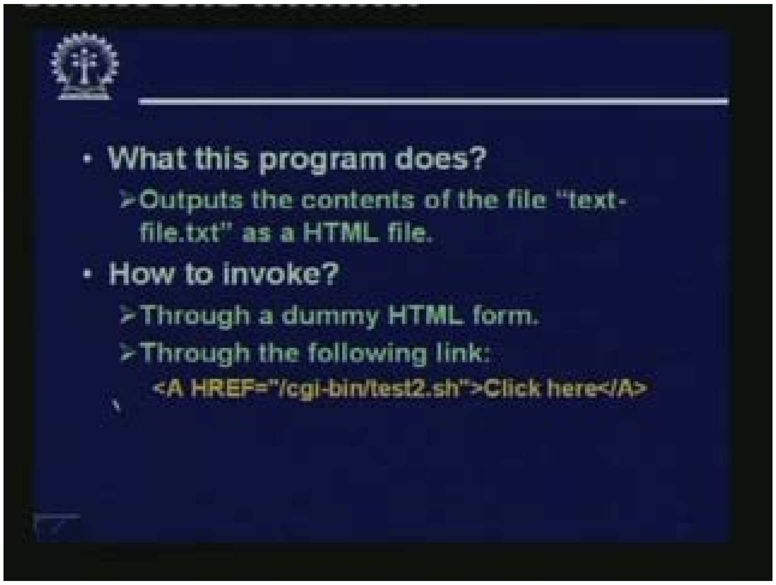
(Refer Slide Time: 50:08)



```
   <SMALL>
   <PRE>
   EOM

   /bin/cat /home/user1/public_html/text-file.txt
   CAT << EOM
   </PRE>
   </SMALL> <P>
</BODY>
</HTML>
EOM
```

This entire thing gets outputted till this EOM comes. Then the actually the contents of the files will come here. Then again of course there is a space here. CAT this file name. So this file will get displayed on the screen again CAT less than less than up to EOM, the end of the html will come here. So actually what will come back or what will be outputted will be the total html file with the content of the file embedded in it.

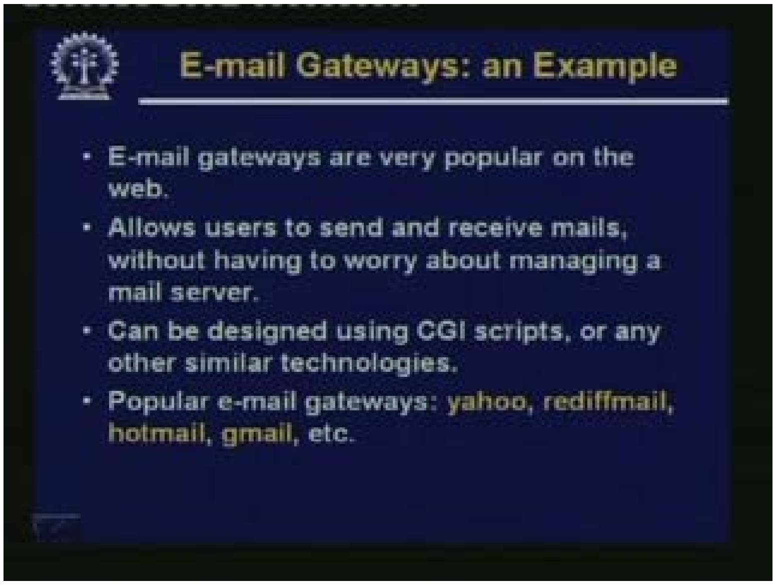
(Refer Slide Time: 50:43)



- What this program does?
  - ➢Outputs the contents of the file "text-file.txt" as a HTML file.
- How to invoke?
  - ➢Through a dummy HTML form.
  - ➢Through the following link:
    `<A HREF="/cgi-bin/test2.sh">Click here</A>`

So this program actually outputs the contents of the file as html file it can be invoked through a dummy html form or through a link like this. CGI bin test two dot sh.

(Refer Slide Time: 50:59)

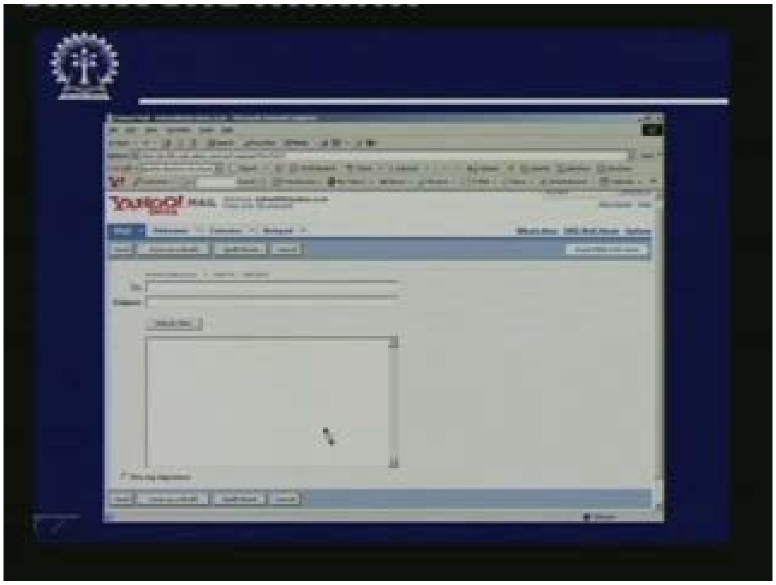


E-mail Gateways: an Example

- E-mail gateways are very popular on the web.
- Allows users to send and receive mails, without having to worry about managing a mail server.
- Can be designed using CGI scripts, or any other similar technologies.
- Popular e-mail gateways: yahoo, rediffmail, hotmail, gmail, etc.

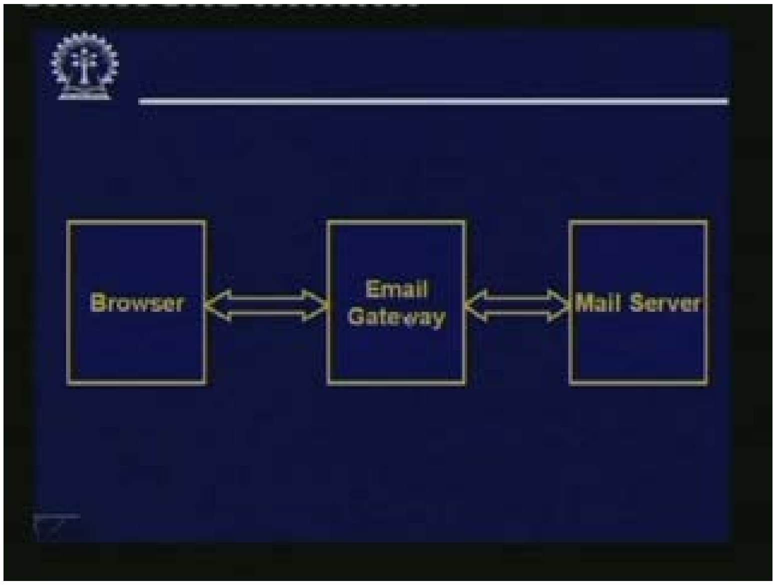
So this gives you some simple idea how you can invoke other form CGI script and some simple example email gateways. This I have already mentioned. Some popular email gateways are there which you can use.

(Refer Slide Time: 51:15)



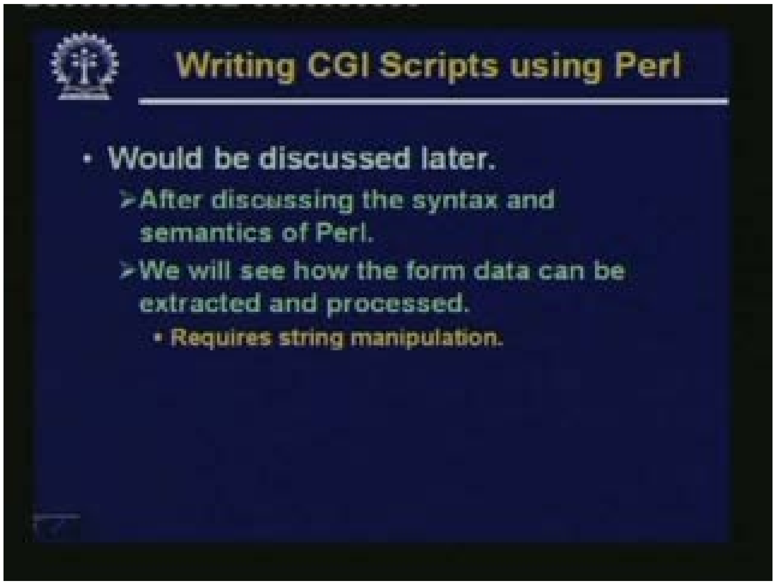
And one such email gateway yahoo looks like this. You can see that there are so many forms in which you need to fill up and send means you are submitting the form. So it is just that.

(Refer Slide Time: 51:26)



Browser is connected to the email gateway. Email gateway is connected to the mail server.

(Refer Slide Time: 51:34)

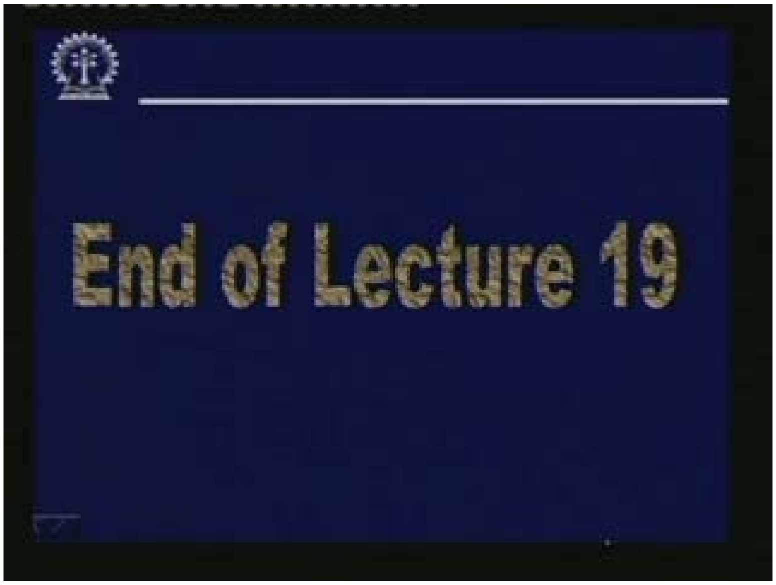


Writing CGI Scripts using Perl

- Would be discussed later.
  - After discussing the syntax and semantics of Perl.
  - We will see how the form data can be extracted and processed.
    - Requires string manipulation.

So writing CGI script using Perl would be discussing later when we talk about the language Perl. And there we shall take complete examples and see that how actually data can be sent to a script within the name value pairs and the like.

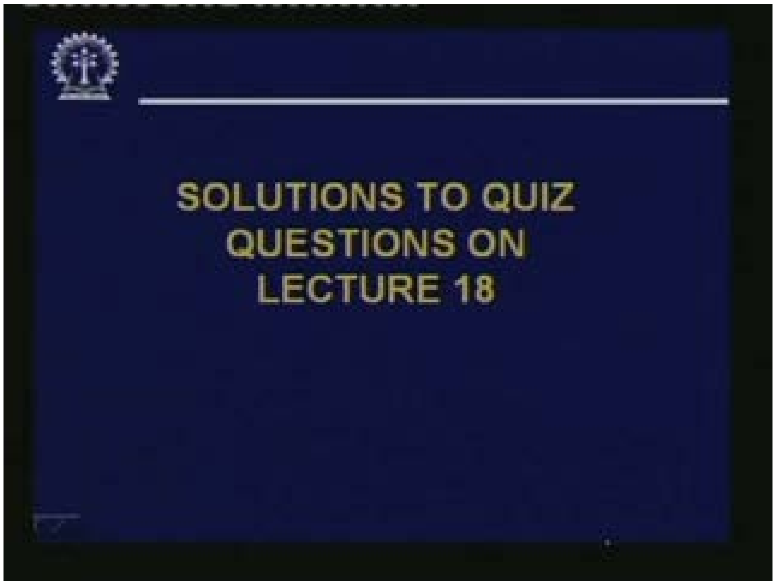
(Refer Slide Time: 51:57)



End of Lecture 19

So with this we come to the end of today’s discussion on the basic technology behind CGI script.

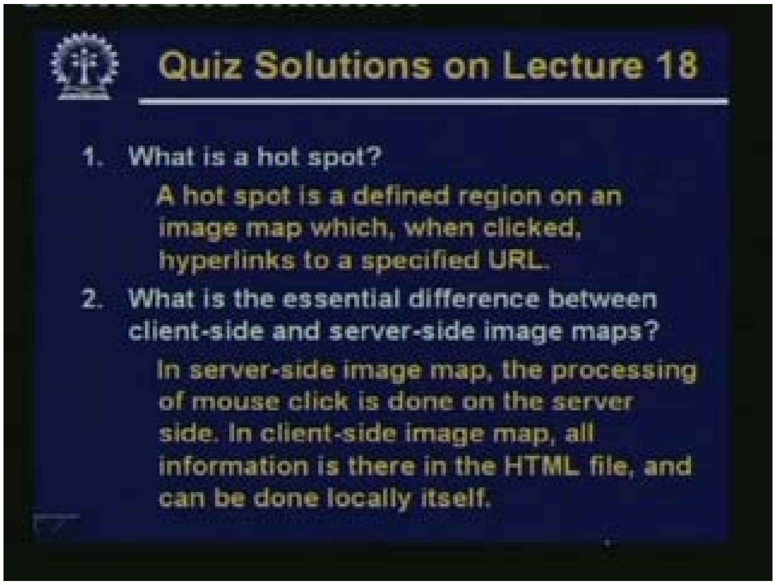
(Refer Slide Time: 52:04)



SOLUTIONS TO QUIZ QUESTIONS ON LECTURE 18

Let us very quickly go through the solutions to last day’s lecture.

(Refer Slide Time: 52:08)



Quiz Solutions on Lecture 18

1. What is a hot spot?

   A hot spot is a defined region on an image map which, when clicked, hyperlinks to a specified URL.

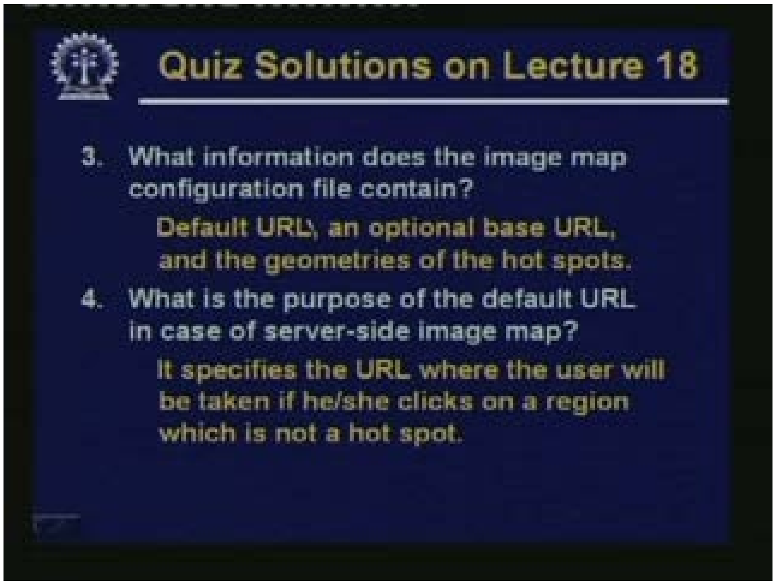
2. What is the essential difference between client-side and server-side image maps?

   In server-side image map, the processing of mouse click is done on the server side. In client-side image map, all information is there in the HTML file, and can be done locally itself.

First question was what is a hot spot?
It is a defined region on an image map which when clicked hyperlinks to a specified URL.

Difference between client side and server side maps.
In server side map the processing of mouse click is done on the server side. There is a program executing there. In client side map all information are present in html in the html file and it is done locally by the browser itself. So you need not disturb the server for this.

(Refer Slide Time: 52:43)



Quiz Solutions on Lecture 18

3. What information does the image map configuration file contain?

   Default URL, an optional base URL, and the geometries of the hot spots.

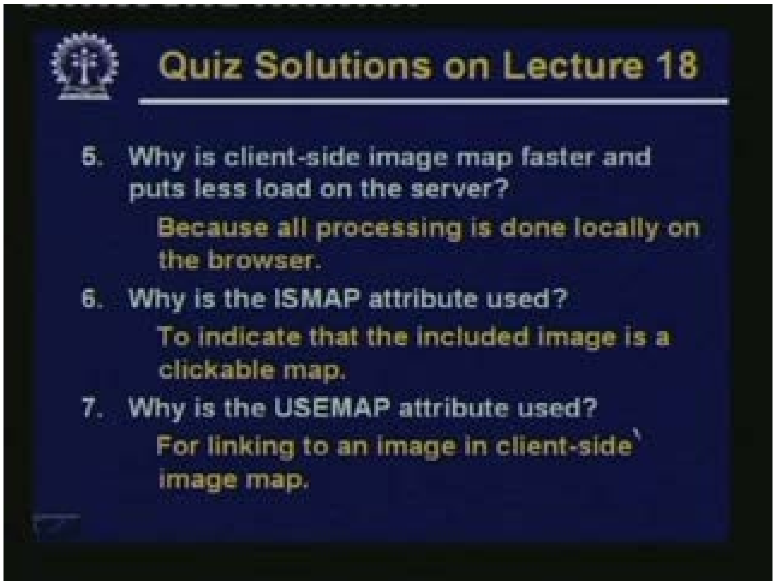
4. What is the purpose of the default URL in case of server-side image map?

   It specifies the URL where the user will be taken if he/she clicks on a region which is not a hot spot.

What information does the image map configuration file contain?
Default URL, optional URL, geometries of the hot spot and their corresponding URLs.

What is the purpose of default URL?
As I mentioned, it specifies the URL where the user will be taken if he or she clicks on a region which is not a hot spot. That is the purpose of default URL.

(Refer Slide Time: 53:10)



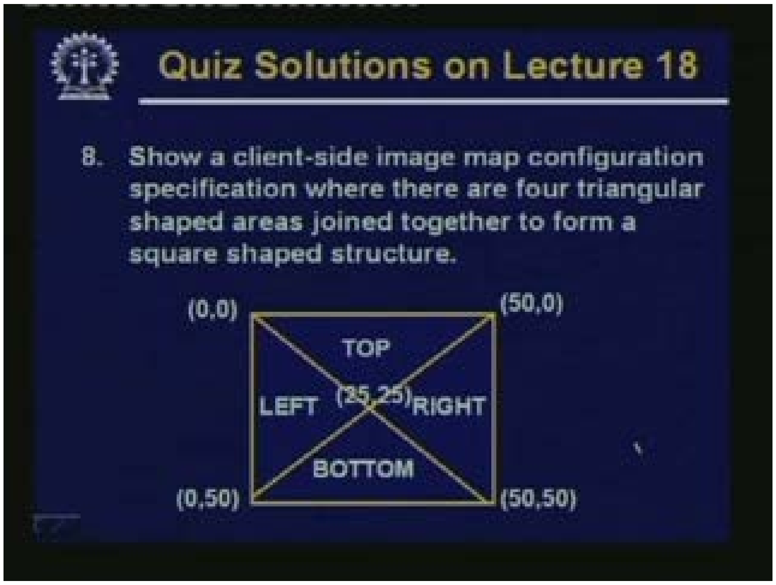
Quiz Solutions on Lecture 18

5. Why is client-side image map faster and puts less load on the server?
   Because all processing is done locally on the browser.
6. Why is the ISMAP attribute used?
   To indicate that the included image is a clickable map.
7. Why is the USEMAP attribute used?
   For linking to an image in client-side image map.

Why is client side image map faster and puts less load on server?
Because, all processing is done locally.

Why is ISMAP used?
To indicate that the included image is a clickable map an ISMAP is used for the server side maps. USEMAP is also used to indicate that the image is a clickable map but it is used for client side.

(Refer Slide Time: 53:34)

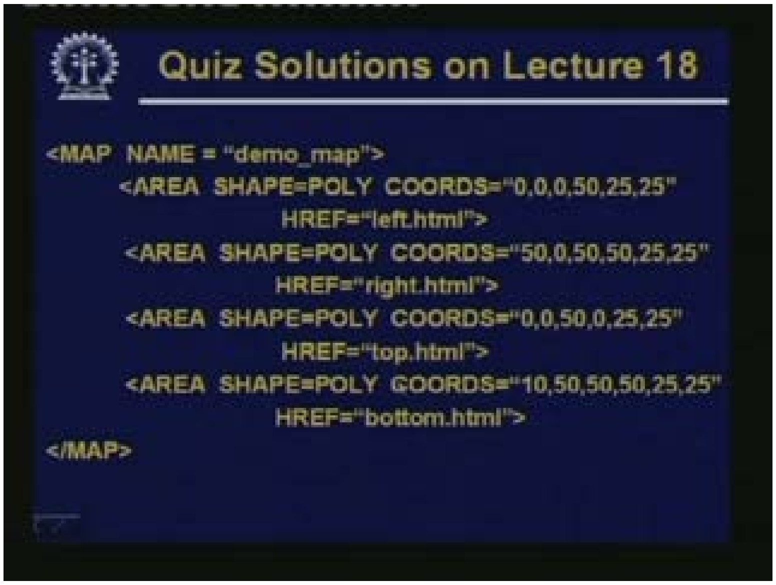


Quiz Solutions on Lecture 18

8. Show a client-side image map configuration specification where there are four triangular shaped areas joined together to form a square shaped structure.

The last question was for geometrical shape like this, an image with four regions to show the client side image map configuration specification. So as an example we have taken some coordinates (0, 0) here (0, 50), (50, 50), (50, 0) and the center coordinate is (25, 25) and the four zones we just referred as left right top and bottom. So with these coordinates we can straight away we can write specification of the four polygons which will be triangles as follows.

(Refer Slide Time: 54:15)

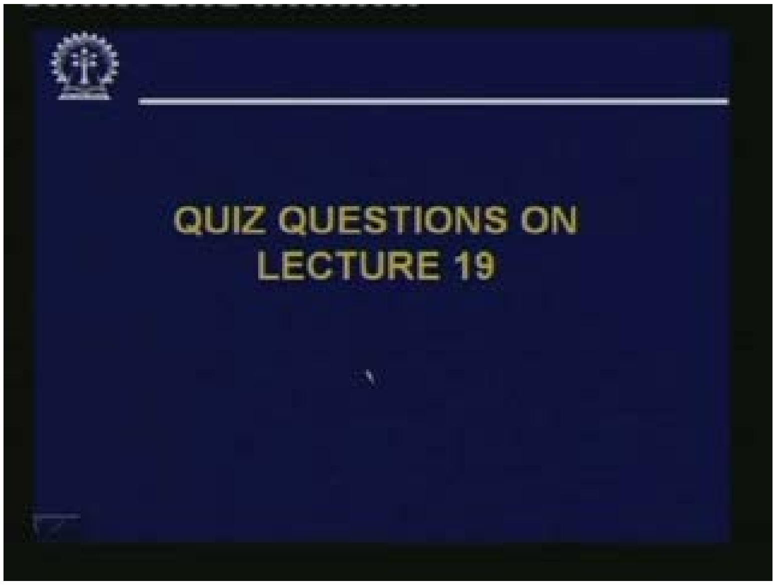


Quiz Solutions on Lecture 18

```
<MAP  NAME = “demo_map”>
      <AREA  SHAPE=POLY  COORDS=“0,0,0,50,25,25”
                    HREF=“left.html”>
       <AREA  SHAPE=POLY  COORDS=“50,0,50,50,25,25”
                   HREF=“right.html”>
       <AREA  SHAPE=POLY  COORDS=“0,0,50,0,25,25”
                    HREF=“top.html”>
       <AREA  SHAPE=POLY  GOORDS=“10,50,50,50,25,25”
                   HREF=“bottom.html”>
</MAP>
```

You can give some name. There will actually four polygons there will be 6 coordinates each. 6 coordinate values 3 3 points and the references will mean which one left right top and bottom. So this you can easily relate with this. How it comes?

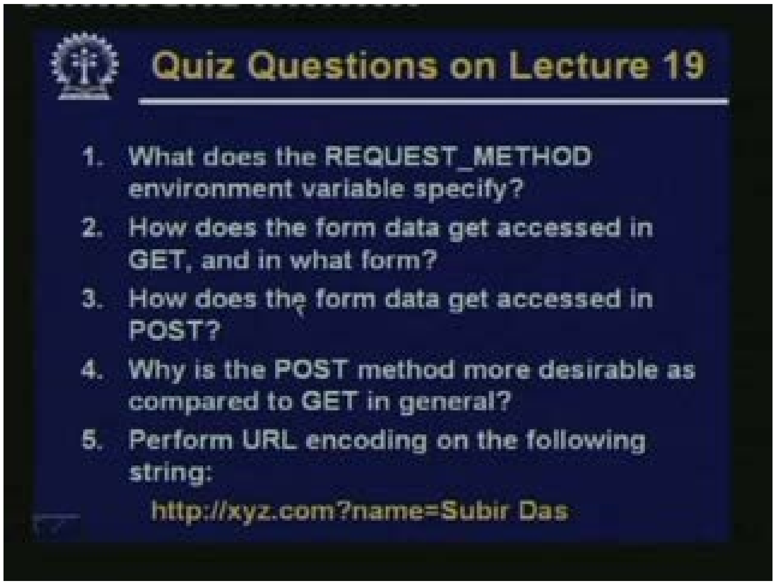
(Refer Slide Time: 54:43)



QUIZ QUESTIONS ON
LECTURE 19

Now, for the questions from today’s lecture.

(Refer Slide Time: 54:47)



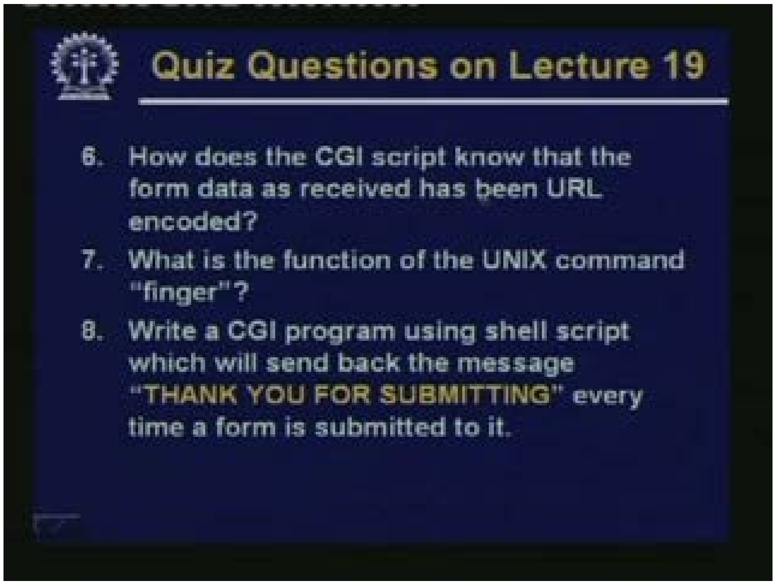
Quiz Questions on Lecture 19

1. What does the REQUEST_METHOD environment variable specify?
2. How does the form data get accessed in GET, and in what form?
3. How does the form data get accessed in POST?
4. Why is the POST method more desirable as compared to GET in general?
5. Perform URL encoding on the following string:
   http://xyz.com?name=Subir Das

What does the request method environment variable specify?
How does the form data get accessed in GET and in what form?
How does the form data get accessed in POST?

Why is the POST method more desirable as compared to get?
Perform URL encoding on the following string. Try to do this, where there are some special symbols out here.

(Refer Slide Time: 55:11)



Quiz Questions on Lecture 19

6. How does the CGI script know that the form data as received has been URL encoded?
7. What is the function of the UNIX command "finger"?
8. Write a CGI program using shell script which will send back the message "THANK YOU FOR SUBMITTING" every time a form is submitted to it.

How does the CGI script know that the form data as received has been URL encoded?
What is the function of the UNIX command finger?
Write a CGI program using shell script which will send back the message, “Thank you for submitting” every time a form is submitted to it.

So with this we come to the end of today’s lecture. In our next lecture we shall be looking at some other technologies not directly CGI script which can also be used to write server side applications, take data from forms and so on. Thank you.

(Refer Slide Time: 55:51)

Lecture # 20

Other Technologies

In the last few lectures, we had talked about html forms CGI scripts, image maps the different ways of designing interactive web scripts. Now in addition to CGI scripts which traditionally is written in Perl or similar language with strong character or string manipulation capabilities there are some alternate solutions also. Today in this lecture we would be talking about a couple of such alternatives which we have when you want to design such interactive web pages.

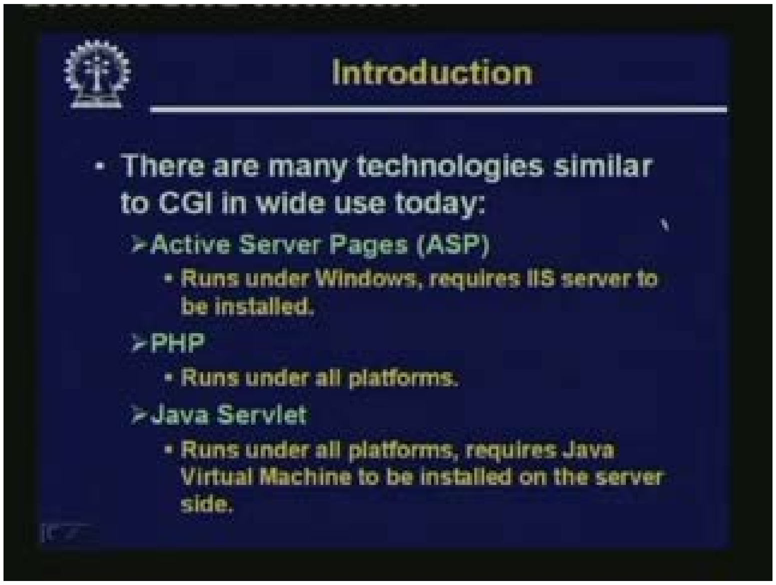
(Refer Slide Time: 56:37)

Indian Institute of Technology Kharagpur

Other Technologies

Prof. Indranil Sen Gupta
Dept. of Computer Science & Engg.
I.I.T. Kharagpur, INDIA

(Refer Slide Time: 56:42)

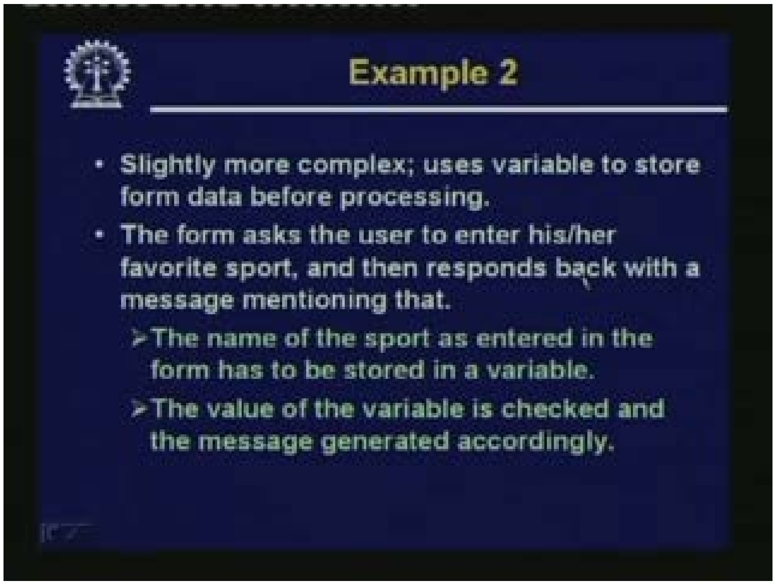


Introduction

- There are many technologies similar to CGI in wide use today:
  - Active Server Pages (ASP)
    - Runs under Windows, requires IIS server to be installed.
  - PHP
    - Runs under all platforms.
  - Java Servlet
    - Runs under all platforms, requires Java Virtual Machine to be installed on the server side.

So under other technologies we would basically talking about active server pages ASP and this PHP. Well active server pages this is a Microsoft based technology. This typically runs under the windows operating system and requires their own proprietary web server IIS to be running on your system in order to have ASP installed. Of course you can have any other web server with ASP support. But IIS is the recommended one. However PHP, this is a solution which is available almost under all platforms. And now as the other alternative you can also have java servlets. Many people develop their interactive web pages using java, using (…..) (Refer Slide Time: 57:38) (…) of the html page. But in this second example we would see how we can use ASP variables. We can store the form data in variables which you can later on check or process in whatever way we want. Well we are not formally showing you the syntax of ASP language. But rather we are giving some examples through which you will understand how an ASP code will look like and you can easily modify this kind of programs to create the program of your choice.

(Refer Slide Time: 58:14)



Example 2

- Slightly more complex; uses variable to store form data before processing.
- The form asks the user to enter his/her favorite sport, and then responds back with a message mentioning that.
  - ➢The name of the sport as entered in the form has to be stored in a variable.
  - ➢The value of the variable is checked and the message generated accordingly.

So the example two is an application where we are designing a form. The form will ask the user to enter his or her favorite sport. There is a choice of course cricket or tennis or chess and then it should respond back to the user with some suitable message mentioning that. So we will see how we are doing this. Now the way you have implemented or written this application is that the name of the sport as the user enters (….).