

# Embedding User Interactivity

Lecture no:

# Today's Aim:

- Interacting with the users using HTML forms
- Using various HTML form elements

# User Interactivity

- Earlier websites were one-sided, i-e we could view the content available at different web-servers but we couldn't give any feedback.
- User Interactivity adds to the efficiency of a website and makes the whole internet process a two way mechanism, i-e the User can not only retrieve data but it can also send data to the server.

# HTML Forms

- There might be many types of data that the user would send, the most common however is text and files.
- HTML forms allow web-designers to interact with their users by providing simple controls such as text boxes, combo boxes (Also called combination boxes and they are used when you set up a form where a question should normally be answered by selecting one of given alternatives but it is possible that none applies, in which case you'd like to ask the user to type his choice in his own words.), **radio controls** (`<input type="radio">` is used to display a radio button, and are arranged in groups of two or more and displayed on screen as, for example, a list of circular holes that can contain white space (for unselected) or a dot (for selected)).
- The user fills the form and clicks on a 'Submit' element and the form gets processed.
- Everyone who has used the Internet should have filled and submitted HTML forms.

# HTML Form: Example

**NWFP University of Engineering & Technology**  
**Entrance Test Result 2009**

Type in your ID Number in the search box

Text Field

Submit Field

 **Board of Intermediate & Secondary Education**

Home   Online Results   Position Holders   Roll Nos

HSSC (Annual) Result 2009

**Result Search by Roll No.**

Your Roll No:

Submit       Text Field

# How do Forms Work

- HTML forms work in a four step process.
  - i. The user fills the form and clicks on submit.
  - ii. A client side script checks for validity of the data (optional).
  - iii. Data is sent to the server using either POST or GET (this is an initial version of HyperText Transfer protocol and the requests send using GET should only retrieve the data, it has no other effect).
  - iv. A server side script handles the data accordingly and a response is sent to the user.

# Using HTML forms

- To construct an HTML form we use the `<form>` tag, and the 'action' attribute specifies the server side script, while 'method' specifies the data delivery method 'post' or 'get'.
- Example: `<form action="somescript.php" method="post"> ...all the form elements...</form>`

# Form submission methods

- There are three methods:
  - a. GET
  - b. POST
  - c. HEAD

GET sends the data by embedding it in the URL (Uniform Resource Locator. A URL is a formatted text string used by Web browsers, email clients and other software to identify a *network resource* on the Internet.) [or] the Address,

POST sends it as hidden in the actual HTTP message,  
and HEAD only gets information about documents located on the server, not the actual document.



# HTML Form: Example

```
<form action="somescript.php" method="post">  
Enter your name: <input type="text" name="name"><br>  
Enter your father's name: <input type="text"  
    name="fname"><br>  
Enter your Roll number: <input type="text" name="rollno"><br>  
<input type="submit" value="Find!">  
</form>
```

# HTML Form: Output

Enter your name:

Enter your father's name:

Enter your Roll number:

Find

Note: The 'Find' submit element will not work as we have not defined the script 'somescript.php' yet, server side scripting is beyond the scope of this course, you can study it yourself once you are familiarized with web-designing.

# HTML Form elements or tags

- Common form elements include:
  - i. Text fields (`<input type="text" ...>`)
  - ii. Text area (`<textarea name="...">`) (The `<textarea>` tag defines a multi-line text input control. A text area can hold an unlimited number of characters)  
)
  - iii. Checkbox( `<input type="checkbox" ...>`)
  - iv. Button (`<input type="button" ..>`)
  - v. Radio (`<input type="radio" ..>`)
  - vi. Password(`<input type="password" ..>`)
  - vii. File (`<input type="file" ..>`)
  - viii. Hidden( `<input type="hidden" ..>`)
  - ix. Submit & Reset (`<input type="submit/reset">`)

# HTML Form elements

- Note that constructing an HTML form does NOT give it any functionality, we have to do that manually, by creating a server side or client side script to handle the form before and/or after submission.