**Error Handling**

* **Error Handling -**
* Error handling is one of the most important part of any web application.
* Each error has to be caught and suitable action has to be taken to resolve that problem.
* ASP.NET provides a simple yet powerful way to deal with errors that occur in your web applications.
* Every application should have error handling.
* By default,ASP.NET displays error messages that include .NET error description along with the stack trace.
* ASP.NET provides three main methods that allow to respond to errors when they occur.

**[1]Page\_Error()**

- event of individual .aspx file

[**2]Application\_Error()**

`

- event in global.aspx file

**[3]<customErrors>**

- section of the application configuration files(Web.config)

* **Sequence of error events-**

In above paragraph we have discussed there are three ways to provide error handling.

The sequence of errors are executed exactly in the same way.

Page\_Error()

event of .aspx file

**LEVEL-1**

**If Error is not cleared,control goes to Level-2**

Application\_Error() enent of Global.asax

**LEVEL-2**

**If Error is not cleared,control goes to level-3**

<customErrors>

section of Web.Config

**LEVEL-3**

**Level-1 :** First of all Page\_Error () event of perticular .aspxfile(Web Form) is executed if you have written and handled it.

**Level-2 :** After going into Page\_Error() next,control goes to Application\_Error() event which is written Global.asax file.

**Level-3 :** As the last sequence,if you have written <custom errors> sction in Web.config,it goes to specified page.

* **Server.ClearError()-**
* It goes from Page\_Error() to Application\_Error() and finally to <custom Errors>.
* At any point of error if you want to stop error going further,is possible by using Server.ClearError().
* **Server.GetLastError()-**
* At any point of time,for proper error handling it is important to know which error /exception has occured so far.
* Server.GetLastError() gives you the most recent error /exception so far.