Frame in HTML :

- Frames divide a browser window into several pieces or panes, each pane containing a separate HTML document.
- The advantage of is that you can load and reload single panes without having to reload the entire contents of the browser window.
- A collection of frames in the browser window is known as a frameset.
- The simplest of framesets might just divide the screen into two rows, while a complex frameset could use several rows and columns.

Creating Frames - The <frameset> Element:

- The <frameset> tag defines how to divide the window into frames.
- Each frameset defines a set of rows **or** columns. If you define frames by using rows then horizontal frames are created. If you define frames by using columns then vertical frames are created.
- The values of the rows/columns indicate the amount of screen area each row/column will occupy.
- Each frame is indicated by <frame> tag and it defines what HTML document to put into the frame.

The <frameset> Element Attributes:

- Following are important attributes of <frameset> and should be known to you to use frameset.
  - **cols:** specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of four ways:
    - Absolute values in pixels. For example to create three vertical frames, use *cols="100, 500,100"*.
    - A percentage of the browser window. For example to create three vertical frames, use *cols="10%, 80%,10%"*.
    - Using a wildcard symbol. For example to create three vertical frames, use *cols="10%, *,10%"*. In this case wildcard takes remainder of the window.
    - As relative widths of the browser window. For example to create three vertical frames, use *cols="3*,2*,1*"*. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
  - **rows:** attribute works just like the cols attribute and can take the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use *rows="10%, 90%"*. You can specify the height of each row in the same way like columns.
  - **border:** attribute specifies the width of the border of each frame in pixels. For example border="5". A value of zero specifies that no border should be there.
  - **frameborder:** specifies whether a three-dimensional border should be displayed between frames. This attrubute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.
  - **framespacing:** specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Element:

- The <frame> element indicates what goes in each frame of the frameset. The <frame> element is always an empty element, and therefore should not have any content, although each <frame> element should always carry one attribute, src, to indicate the page that should represent that frame.
- **For example :**

```
<frame src="/html/top_frame.htm" />
<frame src="/html/main_frame.htm" />
<frame src="/html/bottom_frame.htm" />
```

The <frame> Element Attributes:

- Following are important attributes of and should be known to you to use frames.

  - **src:** indicates the file that should be used in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file avaible in html directory.
  - **name:** attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into a second frame, in which case the second frame needs a name to identify itself as the target of the link.
  - **frameborder:** attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> element if one is given, and the possible values are the same. This can take values either 1 (yes) or 0 (no).
  - **marginwidth:** allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".
  - **marginheight:** allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".
  - **noresize:** By default you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".
  - **scrolling:** controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
  - **longdesc:** allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

The <noframes> Element:

- If a user is using any old browser or any browser which does not support frames then <noframes> element should be displayed to the user.
- In XHTML you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser

does not understand the <frameset> element it should understand what is inside the <body> element contained in the <noframes> element.

- You can print message for your user having old browsers. For example *Sorry!! your browser does not support frames.*

- ***Example :***

```
<noframes>
    <body>
    Your browser does not support frames.
    </body>
</noframes>
```

Frame Example :

```
<html>
<head>
<title>Frames example</title>
</head>
  <frameset rows="10%,80%,10%">
    <frame src="/html/top_frame.htm" />
    <frame src="/html/main_frame.htm" />
    <frame src="/html/bottom_frame.htm" />
    <noframes>
    <body>
    Your browser does not support frames.
    </body>
    </noframes>
  </frameset>
</html>
```

5.1 Overview of internet security :

Internet security is a branch of computer security that deals specifically with Internet-based threats.

These include **hacking**, where unauthorized users gain access to computer systems, email accounts or websites. **Viruses** and other **malicious software** (malware), which can damage data or make systems vulnerable to other threats. And **identity theft**, where hackers steal personal details such as credit card numbers and bank account information. You can protect yourself from these threats with strong Internet security.

In simple words, Internet security is a protection to your personal computer from any harmful and malicious spyware. There are different kind of software to protect your computer in order your important documents that store in your computer are protected.

There are following internet security technique is used to protect your system against internet based attack.

1. Network layer security.
2. Internet Protocol security (IPSec).
3. Pretty good privacy (PGP).
4. Multipurpose internet mail extension (MIME).
5. Message Authentication Code(MAC).
6. Firewall.

Network Layer Security :

TCP/IP Internet protocol group can be made secure communications on the Internet. These protocols include Secure Sockets Layer SSL, succeeded by Transport Layer Security TLS for web traffic, Pretty Good Privacy PGP for email, and IPsec for the network layer security.

Internet Protocol security (IPSec) :

Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality, and replay protection. This protocol is developed by **I**nternet **E**ngineering **T**ask **F**orce(IETF).

Pretty Good Privacy:

It is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. It was created by Phil Zimmermann in 1991.

Multipurpose Internet Mail Extension (MIME)

MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet e-mail protocol that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs.

Message Authentication Code:

Message Authentication Code provide security against the modification and alteration of the message while sending and receiving by the two party. A Message authentication code (MAC) is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.

Firewall :

Firewall provide protection to your local system or local network (private network) against internet based attacks.

Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as *choke points*. Firewalls can create choke points based on IP source and TCP port number.

They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet.

Advantages and disadvantages of internet security :

Advantages:

- Protection from malicious attacks on your network.
- Deletion and/or guaranteeing malicious elements within a preexisting network.
- Prevents users from unauthorized access to the network.
- Securing confidential information.

Disadvantages:

- Difficult to work with for non-technical users.
- Restrictive to resources. It means that all the computer system don't support the internet security protocols or security tool. There is a possibility where the system resource is not enough to satisfy the requirement of internet security tool or protocol.
- Constantly needs Patching. Means you have to update your system regularly which have the security related technology like firewall etc.

Program 1. Create a calculator in Simple Calculator in Javascript.

```
<html>
<head>
	<title>Simple Javascript Calculator - Basic Arithmetic Operations</title>
		<script language="javascript" type="text/javascript">
		function multiply()
		{
			a=Number(document.calculator.number1.value);
			b=Number(document.calculator.number2.value);
			c=a*b;
			document.calculator.total.value=c;
		}
		</script>

		<script language="javascript" type="text/javascript">
		function addition()
		{
			a=Number(document.calculator.number1.value);
			b=Number(document.calculator.number2.value);
			c=a+b;
			document.calculator.total.value=c;
		}
		</script>
		<script language="javascript" type="text/javascript">
		function subtraction()
		{
			a=Number(document.calculator.number1.value);
			b=Number(document.calculator.number2.value);
			c=a-b;
			document.calculator.total.value=c;
		}
		</script>
		<script language="javascript" type="text/javascript">
		function division()
		{
			a=Number(document.calculator.number1.value);
			b=Number(document.calculator.number2.value);
			c=a/b;
			document.calculator.total.value=c;
		}
		</script>

		<script language="javascript" type="text/javascript">
		function modulus()
		{
			a=Number(document.calculator.number1.value);
			b=Number(document.calculator.number2.value);
			c=a%b;
			document.calculator.total.value=c;
		}
```

```
		</script>

</head>
<body>
<form name="calculator">
Number 1: <input type="text" name="number1">
Number 2: <input type="text" name="number2">
Get Result: <input type="text" name="total">
<input type="button" value="ADD" onclick="javascript:addition();">
<input type="button" value="SUB" onclick="javascript:subtraction();">
<input type="button" value="MUL" onclick="javascript:multiply();">
<input type="button" value="DIV" onclick="javascript:division();">
<input type="button" value="MOD" onclick="javascript:modulus();">
</form>
</body>
</html>
```

Program 2- write a javascript for email validation where username can not be blank and password at least six character long.

```
<html>
<head>
	<titile>Email Validation</title>

<SCRIPT LANGUAGE="JavaScript">

function validateform()
{
	if (document.form1.psw.value.length < 6)
	{
		alert("Your password must be at least six character long");
		return false;
	}
	if (document.form1.unm.value=="")
	{
		alert("enter username");
		return false;
	}
	return true;
}
</script>
</head>
<body>
	<form name="form1" onSubmit="validateform()">
	username<input type="text" name="unm"><br>
	password<input type="text" name="psw">
	<input type="submit" value="SUBMIT">
	</form>
	</body>
	</html>
```

Prog 3: write a script for standard calculator in javascript.

```
<html>
  <head>
      <title>Simple Javascript Calculator - Basic Arithmetic Operations</title>
 <script language="javascript" type="text/javascript">
  function multiply()
        {
        num1=Number(document.calculator.number1.value);
        document.getElementById("number1").value="";
        opr=3;
        }
  var num1,num2;
   var opr;

 function addition()
 {
    num1=Number(document.calculator.number1.value);
    document.getElementById("number1").value="";
   opr=1;
 }

  function subtraction()
  {
    num1=Number(document.calculator.number1.value);
        document.getElementById("number1").value="";
        opr=2;
  }

  function division()
  {
       num1=Number(document.calculator.number1.value);
        document.getElementById("number1").value="";
        opr=4;
   }

  function modulus()
  {
   num1=Number(document.calculator.number1.value);
        document.getElementById("number1").value="";
        opr=5;
  }

  function getdata1()
  {
        document.calculator.number1.value+=document.calculator.btn1.value;
  }
```

```
function getdata2()
	{
		document.calculator.number1.value+=document.calculator.btn2.value;

	}
	function getdata3()
	{
		document.calculator.number1.value+=document.calculator.btn3.value;

	}
	function getdata4()
	{
		document.calculator.number1.value+=document.calculator.btn4.value;

	}
	function getdata5()
	{
		document.calculator.number1.value+=document.calculator.btn5.value;

	}
	function getdata6()
	{
		document.calculator.number1.value+=document.calculator.btn6.value;

	}
	function getdata7()
	{
		document.calculator.number1.value+=document.calculator.btn7.value;

	}
	function getdata8()
	{
		document.calculator.number1.value+=document.calculator.btn8.value;

	}
	function getdata9()
	{
		document.calculator.number1.value+=document.calculator.btn9.value;

	}
	function getdata0()
	{
		document.calculator.number1.value+=document.calculator.btn0.value;

	}
	function getdata00()
	{
		document.calculator.number1.value+=document.calculator.btn00.value;

	}
```

```
	function ans()
	{
		num2=Number(document.calculator.number1.value);

		if(opr==1)
				document.calculator.number1.value=(num1+num2);
		else if(opr==2)
			document.calculator.number1.value=(num1-num2);
		else if(opr==3)
			document.calculator.number1.value=(num1*num2);
		else if(opr==4)
			document.calculator.number1.value=(num1/num2);
		else if(opr==5)
			document.calculator.number1.value=(num1%num2);
	}
	function setclear()
	{
		document.calculator.number1.value="";
	}

 </script>

 </head>

 <body>
 <form name="calculator">
 Number 1: <input type="text" id = "number1" name="number1">  </br>
<input type="button" value="1" name="btn1" onclick="javascript:getdata1();">
 <input type="button" value="2" name="btn2" onclick="javascript:getdata2();">
 <input type="button" value="3" name="btn3" onclick="javascript:getdata3();"> </br>
 <input type="button" value="4" name="btn4" onclick="javascript:getdata4();">
 <input type="button" value="5" name="btn5" onclick="javascript:getdata5();">
 <input type="button" value="6" name="btn6" onclick="javascript:getdata6();"> </br>
 <input type="button" value="7" name="btn7" onclick="javascript:getdata7();">
 <input type="button" value="8" name="btn8" onclick="javascript:getdata8();">
 <input type="button" value="9" name="btn9" onclick="javascript:getdata9();"> </br>
 <input type="button" value="0" name="btn0" onclick="javascript:getdata0();">
 <input type="button" value="." name="btn00" onclick="javascript:getdata00();">
 <input type="button" value="ADD" onclick="javascript:addition();">
<input type="button" value="SUB" onclick="javascript:subtraction();"></br>
 <input type="button" value="MUL" onclick="javascript:multiply();">
 <input type="button" value="DIV" onclick="javascript:division();">
 <input type="button" value="MOD" onclick="javascript:modulus();">  </br>
 <input type="button" value="=" onclick="javascript:ans();">
 <input type="button" name ="can" id ="can" value="C" onclick="javascript:setclear();">
 </form>

 </body>
 </html>
```