39110373

HARIHARAN BP

CAE-2 CIDD

(6)

```
< form method =" post" action =" <? php echo
                    htmlspecialchars ($_SERVER
                                      ["PHP_SELF"]
~~Name: <input type ="t~~                    ); ?>">
Name: <input type="text" name="name" >Required >
Email: < input type="email" name =" email "> Required>
Website: < input type=" text" name=" website ">
                                      Required>
Comment: < textarea name=" comment" rows=" 5 "
                              cols =" 40"></textarea>

Gender:
<input type="radio" name =" gender" value=" female">
                                          ~~Female~~
   Female
<input type="radio" name=" gender" value=" male">
                                          ~~Male~~
   Male
<input type="radio" name=" gender" value=" other">

   Other

<~~/form~~>
<input type=" submit" class=" btn btn-primary"
                   value =" submit" >
```

39110373

```
<?php
 // define variable and set to empty values
 $nameErr = $emailErr = $genderErr = $~~Web~~websiteErr = "";
 $name = $email = $gender = $website = $comment = "";
 if ($_SERVER["REQUEST_METHOD"] = "POST")
 {
      if( empty($_POST["name"]))
      {
          $nameErr = "Name is required";
      }
      else
      {
          $name = test_input($_POST["name"]);
      }
      if(empty($_POST["email"]))
      {
          $emailErr = " Email is required";
      }
      else
      {
          $email = test_input($_POST["email"]);
      }
      if (empty($_POST["website"]))
      {
          $website~~Err~~ = "";
      }
      else
{
   $ewebsite= text_input( $POST ["website"]);
}
if (empty $_POST ["comment"]))
{
      $comment="";
  }
  else
  {
     $comment = text_input( $POST ["comment"]);
  }
  if (empty ( $_POST ["gender"]))
  {
     $genderErr= "gender is required";
  }
  else
  {
     ~~$~~ $gender= text_input ( $_POST ["gender"]);
  }
 }
?>
```

(8) The following are the data binding methods

(i) String Interpolation :-

→ It is a type of one-way data binding ~~where~~ where text is ted b/w a set of curly braces often uses the name of a component property.

→ Angular replaces that name with the string value.

Eg. Syntax:

{{ code }}

(ii) Property Binding :-

→ Allows us to bind the view of the template expression

→ In simple term defined as updating the ~~value~~ value of a certain variable in component and presentation layer.

Syntax, Eg:-

```
<h1> Prop Binding </h1>
<img [src]="imagePath" class="image-~~st~~adjustment"/>
```

(iii) Event Binding:

→ Defined as the updating/ sending of the value/ information of a certain variable from the presentation layer to the Component

→ E.g. Clicking Button

Syntax:-

```
<h1> Event Binding</h1>
<h1> {{ title }} </h1>
<button (click)=" changeMyTitle() "> Title is
       changed on clicking" </button>
```

(IV) 2-Way Binding:-

→ It is a combination of both property and Event Binding.

→ A ~~Continuos~~ Continuous synchronization of a ~~dat~~ data from view to components and from component to view.

Syntax:

```
<h1> 2 Way Binding </h1>
<input type= "text" [(ngModel)] = "userName" />
   <br>
```

39110373

<h4> Welcome {{ userName }} </h4>

## PART-A

1. It is a funcn that converts special characters to HTML entities. This means that it will replace HTML characters like < and > with &lt; and &gt;

2. → Browsers can refuse to accept cookies.

→ It adds network overhead to send lots of info.

→ limits the amount of information that can be sent

3. PHP filters are ~~used~~ used to validate and sanitize external input. It has many exten~~sion~~ functions as which is needed for checking user input & is designed to make data validation easier

4.

39110373 (4)

| (4) | JQuery | Angular |
|---|---|---|
| (i) | JavaScript based liberary | Typescript based front-end framework |
| (ii) | Donot have ~~Has~~ Rest ful API | D~~onot~~ have Rest ful API |
| (iii) | ~~Suppo~~ Doesn't Support mvc pattern | Supports mvc pattern |
| (iv) | supports deep linking routing | Does not support deep linking routing |

| (5) | prop() | attr() |
|---|---|---|
| (i) | Returns the default value ( Original state) of a property | Returns the current value ( Current State) of a property |
| (ii) | changes properties for that HTML tag as per the DOM tree | Changes the attributes for that HTML tag |
| (iii) | Explicitly retrieve property values | Retrieves attributes ~~Explicitly~~ Explicitly. |