# Final Question Solution

Fall 2018

1.a)

MVC stands for Model–view–controller. It is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. In MVC pattern when a client requests a server it transfers request to the controller. After that controllers take decision which file will be served for that request. If any database request or query is needed then controller handles it and finally after generating result it shows in browser in html format.
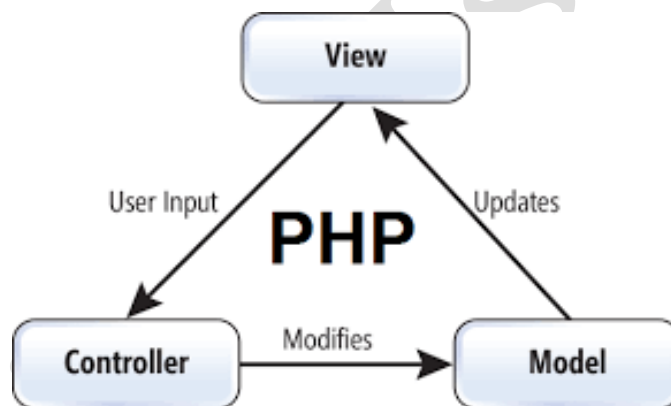
Pros of MVC

1) Faster development process: MVC supports rapid and parallel development. With MVC, one programmer can work on the view while other can work on the controller to create business logic of the web application. The application developed using MVC can be three times faster than application developed using other development patterns.

2) Ability to provide multiple views: In the MVC Model, you can create multiple views for a model. Code duplication is very limited in MVC because it separates data and business logic from the display.

3) Support for asynchronous technique: MVC also supports asynchronous technique, which helps developers to develop an application that loads very fast.

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

4) Modification does not affect the entire model: Modification does not affect the entire model because model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture.

5) MVC model returns the data without formatting: MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface.

6) SEO friendly Development platform: Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a specific application.

Cons of MVC

1) Increased complexity

2) Inefficiency of data access in view

3) Difficulty of using MVC with modern user interface

4) Need multiple programmers

5) Knowledge on multiple technologies is required

6) Developer should have knowledge of both client-side code and html code

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

2.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <form action="">
        <p>Width <input type="text" id="width">
            height <input type="text" id="height">
            Area <input type="text" id="area"><br><br>
            <input type="button" onclick="calculateArea()"
value="Calculate">
        </p>
    </form>
    <script>
        function calculateArea() {
        var width  = parseInt(document.getElementById('width').value);
        var height= parseInt(document.getElementById('height').value);
        var area   = width * height;
        document.getElementById('area').value = area;

        }
    </script>
</body>
</html>
```

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

3.

Intro.html

```html
<html lang="en">
<head>
      <!DOCTYPE html>
<meta charset="UTF-8">
</head>
<body>
      <form action="welcome.php" method="post">
            Name   <input type="text" name="name"><br><br>
            Email   <input type="email" name="email"><br><br>
            <input type="submit">
      </form>
</body>
</html>
```

Welcome.php

```php
<?php
      if (isset($_POST['name'])) {
            $name    = $_POST['name'];
            $email   = $_POST['email'];
            $data    = "Welcome ".$name;
            $data   .= "<br>Your email address is: ".$email;
            echo $data;
      }
?>
```

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

4).

$_GET: $_GET is an array of variables passed to the current script via the URL parameters. Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.

$_POST: $_POST is an array of variables passed to the current script via the HTTP POST method. Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send. Moreover, POST supports advanced functionality such as support for multi-part binary input while uploading files to server.

setCookie(): A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

```php
<?php
    $cookie_name = "user";
    $cookie_value = "John Doe";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
```

Include vs Require: The difference between include and require arises when the file being included cannot be found: include will emit a warning (E_WARNING) and the script will continue, whereas require will emit a fatal error (E_COMPILE_ERROR) and halt the script. If the file being included is critical to the rest of the script running correctly then you need to use require.

Foreach Loop: The foreach loop is mainly used for looping through the values of an array. It loops over the array, and each value for the current array element is assigned to $value, and the array pointer is advanced by one to go the next element in the array.

Ariful Islam
ID: 161-15-946
Daffodil International University
https://github.com/arif98741

```php
<?php
foreach (array as $value){
  //code to be executed;
}
?>
<?php
$salary[]=2000;
$salary[]=3000;
$salary[]=5000;

foreach($salary as $value){
  echo "Salary: $value<br>";
}
?>
Salary: 2000
Salary: 3000
Salary: 5000
```