Ex. No. 08	Web Application Development using Angular JS Framework
23.02.2024	

Aim:

To develop website using HTML, CSS, and Angular JS client-side Model View Controller Framework.

Description

AngularJS Architecture

AngularJS is a Single Page Application (SPA) meta-framework. Angular JS follows the MVC architecture. MVC stands for Model-View-Controller evolved as a way to separate data, logical units and presentation in web application development.

- The model is the driving force of the application. This is generally the data behind the application, usually fetched from the server.
- The view is the UI that the user sees and interacts with. It is dynamic, and generated based on the current model of the application.
- The controller is the business logic and presentation layer, which performs actions such as fetching data and makes decisions such as how to present the model, which parts of it to display,etc.

AngularJS Extends HTML

AngularJS extends HTML with ng-directives.

- **The ng-app**
  directive defines an AngularJS application
- **The ng-init**

directive used to create initial value(model) for the angular JS application.

- **The ng-model**
  directive binds the value of HTML controls (input, select, textarea) to application data.
- **The ng-bind**
  directive binds application data to the HTML view.

Source code:

```
<!DOCTYPE html>
<html lang="en" ng-app="budgetApp">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Budget Calculator</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
   <script src="ex8.js"></script>
</head>

<body ng-controller="budgetController">

   <h1>Budget Calculator</h1>

   <label for="totalAmount">Enter Total Amount:</label>
   <input type="number" id="totalAmount" ng-model="totalAmount" />

   <label for="roomRentPercentage">Room Rent (%):</label>
   <input type="number" id=" roomRentPercentage" ng-model="roomRentPercentage" />

   <label for="accessoriesPercentage">Accessories (%):</label>
   <input type="number" id="accessoriesPercentage" ng-model="accessoriesPercentage" />

   <label for="emergencyPercentage">Emergency (%):</label>
   <input type="number" id="emergencyPercentage" ng-model="emergencyPercentage" />

   <label for="savingPercentage">Saving (%):</label>
   <input type="number" id="savingPercentage" ng-model="savingPercentage" />
```

```
    <button ng-click="evaluateMoney()">Evaluate My Money</button>

    <div ng-if="errorMessage" style="color: red; margin-top: 10px;">{{ errorMessage
}}</div>

    <div ng-if="!errorMessage && evaluated">
        <p>Room Rent: {{ roomRent }}</p>
        <p>Accessories: {{ accessories }}</p>
        <p>Emergency: {{ emergency }}</p>
        <p>Saving: {{ saving }}</p>
    </div>

</body>

</html>
```

Angular code :

```
var app = angular.module('budgetApp', []);

app.controller('budgetController', function ($scope) {
    $scope.evaluateMoney = function () {
        $scope.evaluated = false;
        $scope.errorMessage = '';

        // Check if percentages sum up to 100
        if ($scope.roomRentPercentage + $scope.accessoriesPercentage +
$scope.emergencyPercentage + $scope.savingPercentage !== 100) {
            $scope.errorMessage = 'Percentage values must add up to 100.';
            return;
        }

        // Calculate amounts based on percentages
        $scope.roomRent = ($scope.totalAmount * $scope.roomRentPercentage) / 100;
        $scope.accessories = ($scope.totalAmount * $scope.accessoriesPercentage) / 100;
        $scope.emergency = ($scope.totalAmount * $scope.emergencyPercentage) / 100;
        $scope.saving = ($scope.totalAmount * $scope.savingPercentage) / 100;
```

```
        // Check if totalAmount is divisible by percentages
        if ($scope.totalAmount !== ($scope.roomRent + $scope.accessories + $scope.emergency
+ $scope.saving)) {
            $scope.errorMessage = 'Cannot divide total amount based on given percentages.';
            return;
        }

        $scope.evaluated = true;
    };
})
```

OUTPUT:

Result:

Thus, the program to develop website using HTML, CSS, and Angular JS client-side Model View Controller Framework.