**Source Code :-**

**Aspnet.json:-**

{

    "id": 1,

    "name": "Asp.Net Quiz",

    "description": "Asp.Net Quiz (contains webform, mvc, web API, etc.)",

    "questions": [

        {

            "id": 1010,

            "name": "ASP.NET webform separates the HTML output from program logic using a feature named as",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Exception",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Code-behind",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Code-front",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1011,

            "name": "The feature in ASP.NET 2.0 that is used to fire a normal postback to a different page in the application is called",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Theme",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Code-front",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Cross Page Posting",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1012,

            "name": "What class does the ASP.NET Web Form class inherit from by default?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "System.Web.UI.Page",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "System.Web.UI.Form",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "System.Web.GUI.Page",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "System.Web.Form",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1013,

            "name": "What does MVC stand for?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Model View Controller",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Maximum Virtual Control",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Microsoft Visual Core",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1014,

            "name": "Which of the following does NOT require type casting?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Session",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "TempData",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "ViewData",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "ViewBag",

                    "isAnswer": true

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1015,

            "name": "Which is the correct order of Page life-cycle in asp.net webform?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Init, PreRender, Load",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Load, PreRender, Init",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Init, Load, PreRender",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "None of the above",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1016,

            "name": "Which of these data source controls do not implement caching?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "ObjectDataSource",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "LinqDataSource",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "SqlDataSource",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "XmlDataSource",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1017,

            "name": "Which tag asp:Label control by default renders to?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "div",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "span",

                    "isAnswer": true

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "body",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "label",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1018,

            "name": "Which method do you use to explicitly kill a user's session?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "Session.Terminate()",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Session.TimeOut()",

                    "isAnswer": false

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Session.Abondon()",

                    "isAnswer": true

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "Session.Kill()",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        },

        {

            "id": 1019,

            "name": "Which of the following object is ideal for keeping data alive for a single request?",

            "questionTypeId": 1,

            "options": [

                {

                    "id": 1055,

                    "questionId": 1010,

                    "name": "HttpContext",

                    "isAnswer": true

                },

                {

                    "id": 1056,

                    "questionId": 1010,

                    "name": "Session",

                    "isAnswer": false

                },

                {

                    "id": 1057,

                    "questionId": 1010,

                    "name": "Cookies",

                    "isAnswer": false

                },

                {

                    "id": 1058,

                    "questionId": 1010,

                    "name": "SqlServer",

                    "isAnswer": false

                }

            ],

            "questionType": {

                "id": 1,

                "name": "Multiple Choice",

                "isActive": true

            }

        }

    ]

}

**Quiz.component.Html:-**

<div class="row">

  <div class="col-6">

    <h3>Quiz Application</h3>

  </div>

  <div class="col-6 text-right">

    Select Quiz:

    <select [(ngModel)]="quizName" (change)="loadQuiz(quizName)">

      <option \*ngFor="let quiz of quizes" [value]="quiz.id">{{quiz.name}}</option>

    </select>

  </div>

</div>

<div id="quiz">

  <h2 class="text-center font-weight-normal">{{quiz.name}}</h2>

  <hr />

  <div \*ngIf="mode=='quiz' && quiz">

    <div \*ngFor="let question of filteredQuestions;">

      <div class="badge badge-info">Question {{pager.index + 1}} of {{pager.count}}.</div>

      <div \*ngIf="config.duration" class="badge badge-info float-right">Time: {{ellapsedTime}} / {{duration}}</div>

      <h3 class="font-weight-normal">{{pager.index + 1}}.

        <span [innerHTML]="question.name"></span>

      </h3>

      <div class="row text-left options">

        <div class="col-6" \*ngFor="let option of question.options">

          <div class="option">

            <label class="font-weight-normal" [attr.for]="option.id">

              <input id="{{option.id}}" type="checkbox" [(ngModel)]="option.selected" (change)="onSelect(question, option);" /> {{option.name}}

            </label>

          </div>

        </div>

      </div>

    </div>

    <hr />

    <div class="text-center">

      <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(0);">First</button>

      <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(pager.index - 1);">Prev</button>

      <button class="btn btn-primary" (click)="goTo(pager.index + 1);">Next</button>

      <button class="btn btn-default" \*ngIf="config.allowBack" (click)="goTo(pager.count - 1);">Last</button>

      <!--<pagination \*ngIf="config.showPager" direction-links="false" total-items="totalItems" items-per-page="itemsPerPage" ng-model="currentPage" ng-change="pageChanged()"></pagination>-->

    </div>

  </div>

  <div class="row text-center" \*ngIf="mode=='review'">

    <div class="col-4 cursor-pointer" \*ngFor="let question of quiz.questions; let index = index;">

      <div (click)="goTo(index)" class="p-3 mb-2 {{ isAnswered(question) == 'Answered'? 'bg-info': 'bg-warning' }}">{{index + 1}}. {{ isAnswered(question) }}</div>

    </div>

  </div>

  <div class="result" \*ngIf="mode=='result'">

    <h2>Quiz Result</h2>

    <div \*ngFor="let question of quiz.questions; let index = index">

      <div class="result-question">

        <h5>{{index + 1}}. {{question.name}}</h5>

        <div class="row">

          <div class="col-6" \*ngFor="let Option of question.options">

            <input id="{{Option.id}}" type="checkbox" disabled="disabled" [(ngModel)]="Option.selected" /> {{Option.name}}

          </div>

        </div>

        <div class="p-1 m-2 alert {{ isCorrect(question) == 'correct'? 'alert-success': 'alert-danger'}}">Your answer is {{isCorrect(question)}}.</div>

      </div>

    </div>

    <h4 class="alert alert-info text-center">You may close this window now.</h4>

  </div>

  <hr />

  <div \*ngIf="mode!='result'">

    <button class="btn btn-warning" (click)="mode = 'quiz'">Quiz</button>

    <button class="btn btn-info" (click)="mode = 'review'">Review</button>

    <button class="btn btn-primary" (click)="onSubmit();">Submit Quiz</button>

  </div>

</div>

**Index.Html:-**

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Angular Quiz</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>