

Comprehensive Assessment

Name- Priyansu Jena

MID- M1082626

User Stories

A) EPIC:

User Registration to a page for Discussion Forum

USER STORIES/TASKS: Assigned to Tushar

<As a new User>

<I want to register myself with my details including Email , password, First name and last name >

<So that the system can easily store my information for next time so i can login easily >

SUBTASKS:

1. user should give details along with password and username
2. Add the Register button

B) EPIC:

User Login to the page

USER STORIES/TASKS: Assigned to Upadhyaya

<As a User>

<I want to login to the page with my username and password only>

<So that i don't have to give all my details>

SUBTASKS:

1. Add Username box
2. Add password box
3. Add Login Button so that user can login to page by this two details

C) EPIC:

For ask question and post question.

USER STORIES/TASKS: Assigned to Lucky

<As a User>

<I want ask and post a question so the option like ask question and post question name to be select as per choice>

<so that i can easily select my question id and answer id instead of searching all of the Menu page>

SUBTASKS:

1. Add options SME answer and Reviewer description to the search bar
2. Add comments in it
3. Add your question in it so user can easily select

D) EPIC:

POST QUESTION

USER STORIES/TASKS: Assigned to Ranjan

<As a User>

<I want to select my prefer question with my preferable date and time>

<So that i am available at the time or date>

SUBTASKS:

1. enable the user to select his preferred question by giving select option on the screen name
2. Add post question so that they can ask there wanted date

E) EPIC:

DELETE THE QUESTION

USER STORIES/TASKS: Assigned to Saroj

<As a user>

< I Want to have a option like delete the question before see the SME answer >

<So that SME answer also Not display at a time>

SUBTASKS:

- 1.Reviewer can examine the question and able to give description.
- 2.As well as examine the answer and able to give description

Models

1. Userlogin.cs

```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace MyLearningPlatform.Models
4 {
5     public class Userlogin
6     {
7         [Key]
8         public int userid { get; set; }
9         public string username { get; set; }
10        public string password { get; set; }
11    }
12 }
13
14
```

```
using System.ComponentModel.DataAnnotations;
```

```
namespace MyLearningPlatform.Models
{
    public class Userlogin
    {
        [Key]
        public int userid { get; set; }
        public string username { get; set; }
        public string password { get; set; }
    }
}
```

2. Question.cs

```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace MyLearningPlatform.Models
4 {
5     public class Question
6     {
7         [Key]
8         public int QuestionId { get; set; }
9         public string PostQuestion { get; set; }
10        public string ReviewerDescription { get; set; }
11    }
12 }
13
14
```

```
using System.ComponentModel.DataAnnotations;
```

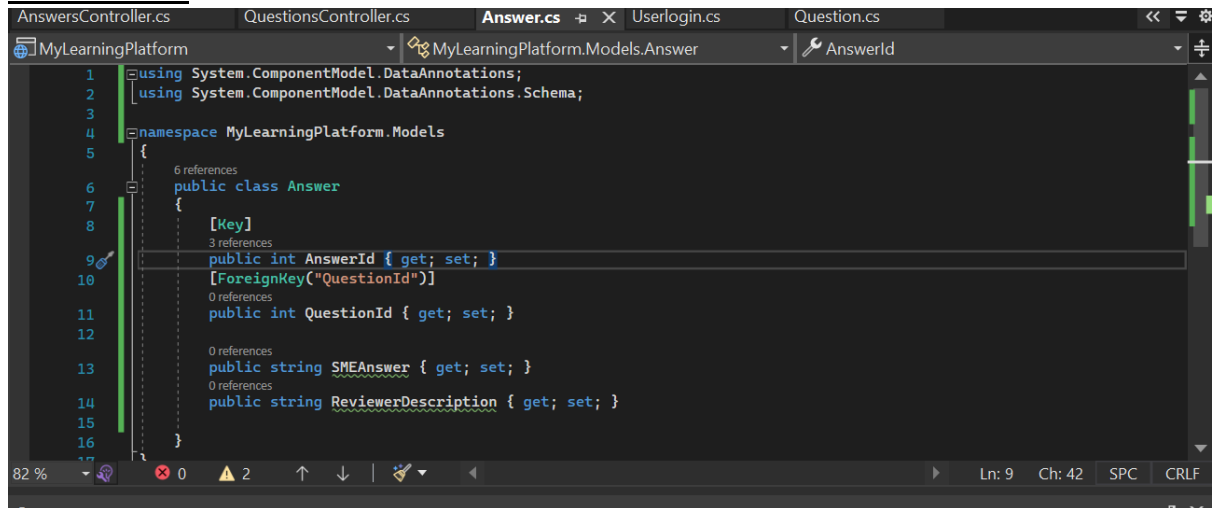
```
namespace MyLearningPlatform.Models
{
    public class Question
    {
        [Key]
        public int QuestionId { get; set; }
        public string PostQuestion { get; set; }
    }
}
```

```

        public string ReviewerDescription { get; set; }
    }
}

```

3. Answer.cs



```

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MyLearningPlatform.Models
{
    public class Answer
    {
        [Key]
        public int AnswerId { get; set; }
        [ForeignKey("QuestionId")]
        public int QuestionId { get; set; }

        public string SMEAnswer { get; set; }
        public string ReviewerDescription { get; set; }
    }
}

```

Controller

1. Userlogin Controller

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Http;
6 using Microsoft.AspNetCore.Mvc;
7 using Microsoft.EntityFrameworkCore;
8 using MyLearningPlatform.Data;
9 using MyLearningPlatform.Models;
10
11 namespace MyLearningPlatform.Controllers
12 {
13     [Route("api/[controller]")]
14     [ApiController]
15     public class UserLoginsController : ControllerBase
16     {
17         private readonly MyLearningPlatformContext _context;
18
19         public UserLoginsController(MyLearningPlatformContext context)
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MyLearningPlatform.Data;
using MyLearningPlatform.Models;
```

```
namespace MyLearningPlatform.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class UserLoginsController : ControllerBase
    {
        private readonly MyLearningPlatformContext _context;

        public UserLoginsController(MyLearningPlatformContext context)
        {
            _context = context;
        }

        // GET: api/UserLogins
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Userlogin>>> GetUserlogin()
        {
            if (_context.Registration == null)
            {
                return NotFound();
            }
            return await _context.Registration.ToListAsync();
        }

        // GET: api/UserLogins/5
        [HttpGet("{id}")]
        public async Task<ActionResult<Userlogin>> GetUserlogin(int id)
        {
            if (_context.Registration == null)
            {
                return NotFound();
            }
            var userlogin = await _context.Registration.FindAsync(id);
```

```

        if (userlogin == null)
        {
            return NotFound();
        }

        return userlogin;
    }

    // PUT: api/Userlogins/5
    // To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPut("{id}")]
    public async Task<ActionResult> PutUserlogin(int id, Userlogin userlogin)
    {
        if (id != userlogin.userid)
        {
            return BadRequest();
        }

        _context.Entry(userlogin).State = EntityState.Modified;

        try
        {
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!UserloginExists(id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }

        return NoContent();
    }

    // POST: api/Userlogins
    // To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
    [HttpPost]
    public async Task<ActionResult<Userlogin>> PostUserlogin(Userlogin userlogin)
    {
        if (_context.Registration == null)
        {
            return Problem("Entity set 'MyLearningPlatformContext.Registration' is null.");
        }
        _context.Registration.Add(userlogin);
        await _context.SaveChangesAsync();

        return CreatedAtAction("GetUserlogin", new { id = userlogin.userid }, userlogin);
    }

    // DELETE: api/Userlogins/5
    [HttpDelete("{id}")]
    public async Task<ActionResult> DeleteUserlogin(int id)
    {
        if (_context.Registration == null)

```

```

    {
        return NotFound();
    }
    var userlogin = await _context.Registration.FindAsync(id);
    if (userlogin == null)
    {
        return NotFound();
    }

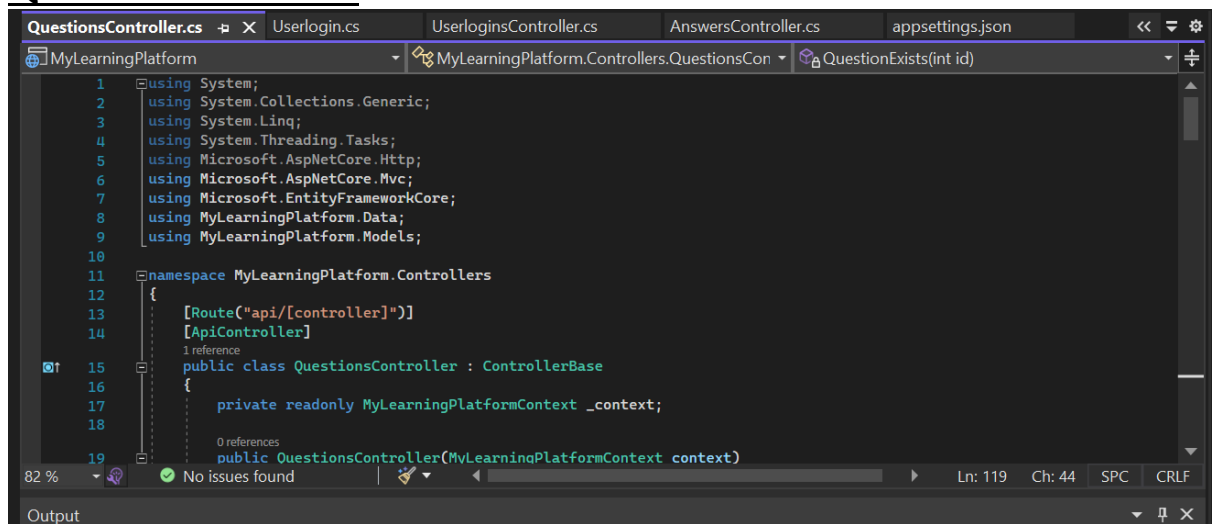
    _context.Registration.Remove(userlogin);
    await _context.SaveChangesAsync();

    return NoContent();
}

private bool UserloginExists(int id)
{
    return (_context.Registration?.Any(e => e.userid == id)).GetValueOrDefault();
}
}
}

```

2. Question Controller



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MyLearningPlatform.Data;
using MyLearningPlatform.Models;

namespace MyLearningPlatform.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class QuestionsController : ControllerBase
    {
        private readonly MyLearningPlatformContext _context;
    }
}

```

```

public QuestionsController(MyLearningPlatformContext context)
{
    _context = context;
}

// GET: api/Questions
[HttpGet]
public async Task<ActionResult<IEnumerable<Question>>> GetQuestion()
{
    if (_context.Question == null)
    {
        return NotFound();
    }
    return await _context.Question.ToListAsync();
}

// GET: api/Questions/5
[HttpGet("{id}")]
public async Task<ActionResult<Question>> GetQuestion(int id)
{
    if (_context.Question == null)
    {
        return NotFound();
    }
    var question = await _context.Question.FindAsync(id);

    if (question == null)
    {
        return NotFound();
    }

    return question;
}

// PUT: api/Questions/5
// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPut("{id}")]
public async Task<ActionResult> PutQuestion(int id, Question question)
{
    if (id != question.QuestionId)
    {
        return BadRequest();
    }

    _context.Entry(question).State = EntityState.Modified;

    try
    {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!QuestionExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }
}

```



```

    }
}

return NoContent();
}

// POST: api/Questions
// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPost]
public async Task<ActionResult<Question>> PostQuestion(Question question)
{
    if (_context.Question == null)
    {
        return Problem("Entity set 'MyLearningPlatformContext.Question' is null.");
    }
    _context.Question.Add(question);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetQuestion", new { id = question.QuestionId }, question);
}

// DELETE: api/Questions/5
[HttpDelete("{id}")]
public async Task<ActionResult> DeleteQuestion(int id)
{
    if (_context.Question == null)
    {
        return NotFound();
    }
    var question = await _context.Question.FindAsync(id);
    if (question == null)
    {
        return NotFound();
    }

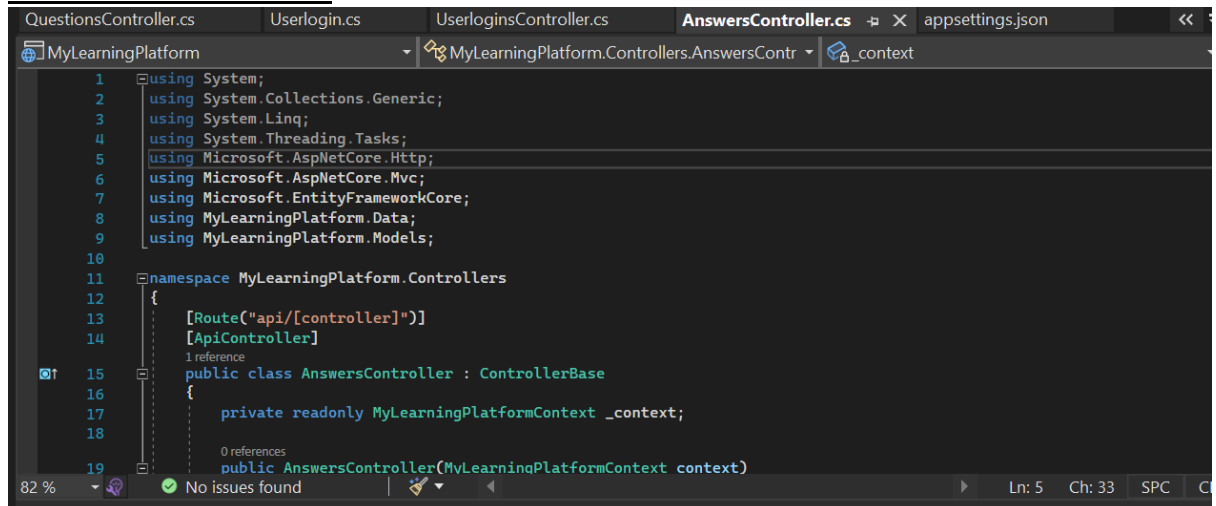
    _context.Question.Remove(question);
    await _context.SaveChangesAsync();

    return NoContent();
}

private bool QuestionExists(int id)
{
    return (_context.Question?.Any(e => e.QuestionId == id)).GetValueOrDefault();
}
}

```

3. Answer Controller



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Http;
6 using Microsoft.AspNetCore.Mvc;
7 using Microsoft.EntityFrameworkCore;
8 using MyLearningPlatform.Data;
9 using MyLearningPlatform.Models;
10
11 namespace MyLearningPlatform.Controllers
12 {
13     [Route("api/[controller]")]
14     [ApiController]
15     public class AnswersController : ControllerBase
16     {
17         private readonly MyLearningPlatformContext _context;
18
19         public AnswersController(MyLearningPlatformContext context)
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MyLearningPlatform.Data;
using MyLearningPlatform.Models;
```

```
namespace MyLearningPlatform.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class AnswersController : ControllerBase
    {
        private readonly MyLearningPlatformContext _context;

        public AnswersController(MyLearningPlatformContext context)
        {
            _context = context;
        }

        // GET: api/Answers
        [HttpGet]
        public async Task<ActionResult<IEnumerable<Answer>>> GetAnswer()
        {
            if (_context.Answer == null)
            {
                return NotFound();
            }
            return await _context.Answer.ToListAsync();
        }

        // GET: api/Answers/5
        [HttpGet("{id}")]
        public async Task<ActionResult<Answer>> GetAnswer(int id)
        {
            if (_context.Answer == null)
            {
                return NotFound();
            }
        }
    }
}
```

```

        return NotFound();
    }
    var answer = await _context.Answer.FindAsync(id);

    if (answer == null)
    {
        return NotFound();
    }

    return answer;
}

// PUT: api/Answers/5
// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPut("{id}")]
public async Task<ActionResult> PutAnswer(int id, Answer answer)
{
    if (id != answer.AnswerId)
    {
        return BadRequest();
    }

    _context.Entry(answer).State = EntityState.Modified;

    try
    {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!AnswerExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return NoContent();
}

// POST: api/Answers
// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
[HttpPost]
public async Task<ActionResult<Answer>> PostAnswer(Answer answer)
{
    if (_context.Answer == null)
    {
        return Problem("Entity set 'MyLearningPlatformContext.Answer' is null.");
    }
    _context.Answer.Add(answer);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetAnswer", new { id = answer.AnswerId }, answer);
}

// DELETE: api/Answers/5
[HttpDelete("{id}")]

```

```

public async Task<ActionResult> DeleteAnswer(int id)
{
    if (_context.Answer == null)
    {
        return NotFound();
    }
    var answer = await _context.Answer.FindAsync(id);
    if (answer == null)
    {
        return NotFound();
    }

    _context.Answer.Remove(answer);
    await _context.SaveChangesAsync();

    return NoContent();
}

private bool AnswerExists(int id)
{
    return (_context.Answer?.Any(e => e.AnswerId == id)).GetValueOrDefault();
}
}

```

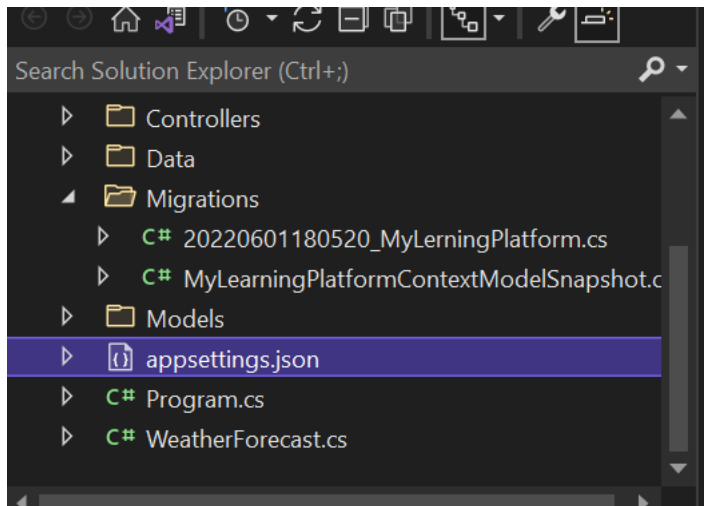
AppSettings.json

```

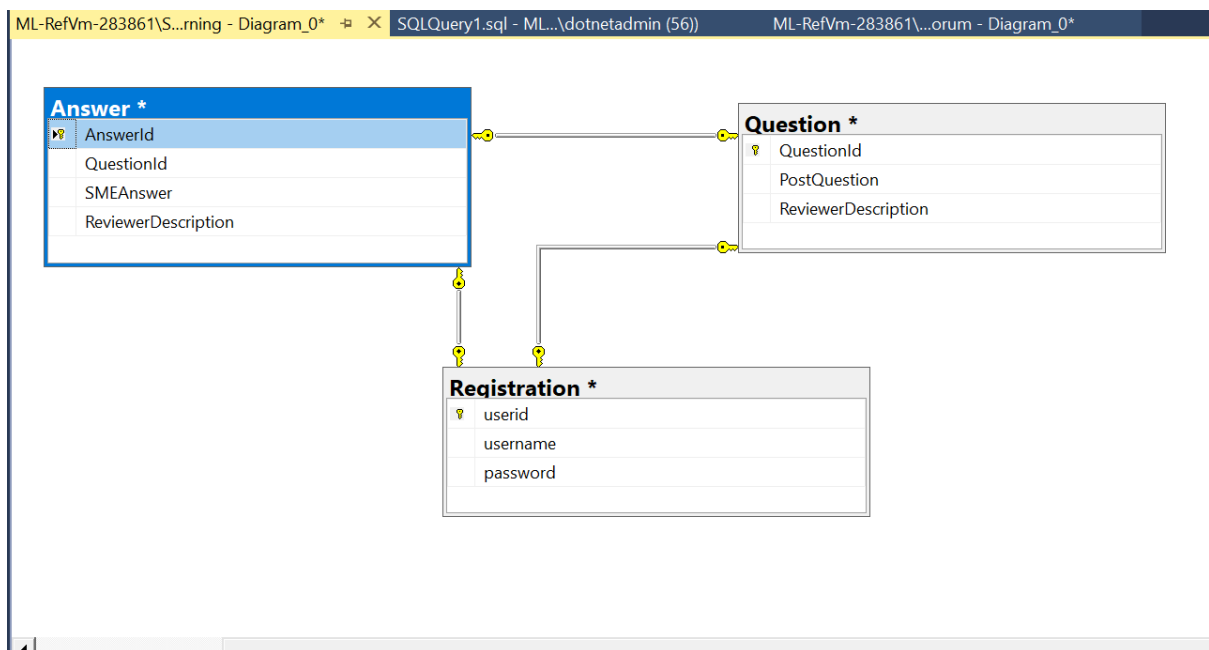
"AllowedHosts": "*",
"ConnectionStrings": {
    "MyLearningPlatformContext": "Server=ML-RefVm-
283861\\SQLEXPRESS;Database=MyLearningPlatform.Data;Trusted_Connection=True;MultipleActiveResultSets=true
"

```

Migration



Schematic Diagram



Swagger

Swagger UI

localhost:5072/swagger/index.html

Google Chrome isn't your default browser [Set as default](#)

Swagger
Supported by SMARTBEAR

Select a definition **MyLearningPlatform v1**

MyLearningPlatform ^{1.0} ^{OAS3}

<http://localhost:5072/swagger/v1/swagger.json>

Answers

- GET** /api/Answers
- POST** /api/Answers
- GET** /api/Answers/{id}
- PUT** /api/Answers/{id}
- DELETE** /api/Answers/{id}

Questions

- GET** /api/Questions
- POST** /api/Questions
- GET** /api/Questions/{id}
- PUT** /api/Questions/{id}
- DELETE** /api/Questions/{id}

Userlogins

- GET** /api/Userlogins
- POST** /api/Userlogins
- GET** /api/Userlogins/{id}
- PUT** /api/Userlogins/{id}
- DELETE** /api/Userlogins/{id}

Database

SQLQuery2.sql - ML...\dotnetadmin (57) SQLQuery1.sql - ML...\dotnetadmin (51) ML-RefVm-283861\S...Data - Diagram_1*

```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [QuestionId]  
                , [PostQuestion]  
                , [ReviewerDescription]  
FROM [MyLearningPlatform.Data].[dbo].[Question]
```

100 %

Results Messages

QuestionId	PostQuestion	ReviewerDescription
------------	--------------	---------------------

SQLQuery3.sql - ML...\dotnetadmin (52) SQLQuery2.sql - ML...\dotnetadmin (57) SQLQuery1.sql - ML...\dotnetadmin (51)

```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [userid]  
                , [username]  
                , [password]  
FROM [MyLearningPlatform.Data].[dbo].[Userlogin]
```

100 %

Results Messages

userid	username	password
--------	----------	----------

SQLQuery1.sql - ML...\dotnetadmin (51) ML-RefVm-283861\S...Data - Diagram_1*

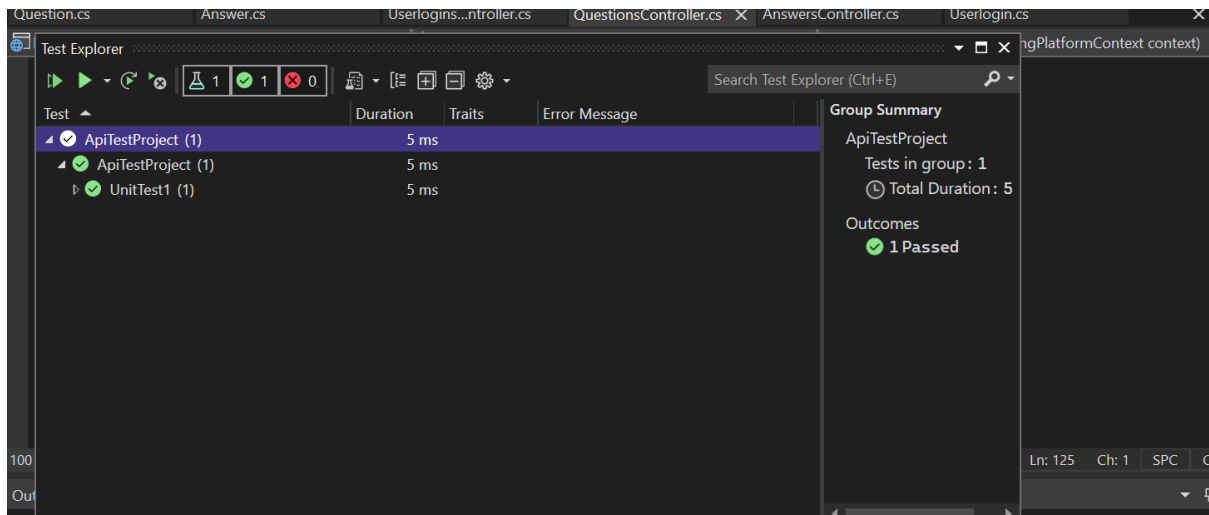
```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [AnswerId]  
                , [QuestionId]  
                , [SMEAnswer]  
                , [ReviewerDescription]  
FROM [MyLearningPlatform.Data].[dbo].[Answer]
```

100 %

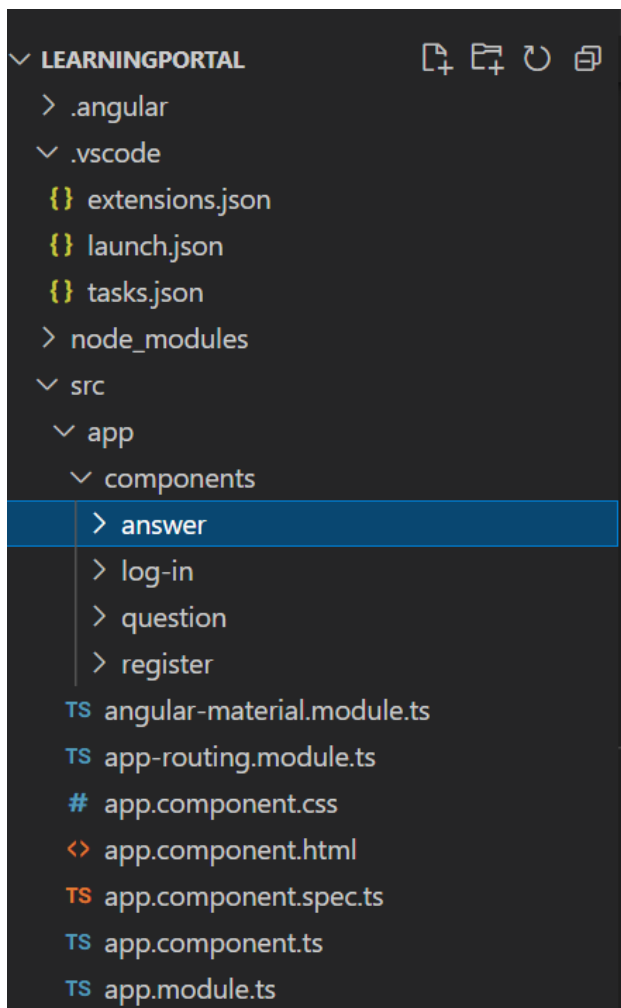
Results Messages

AnswerId	QuestionId	SMEAnswer	ReviewerDescription
----------	------------	-----------	---------------------

Test the project in MS Test



Angular



Appcomponent.html

```
<> app.component.html X <> index.html TS app.component.ts TS angular-material.module.ts TS app.module.ts
src > app > <> app.component.html > mat-toolbar.app-header > span.nav-tool-items > a
1 <!-- Toolbar -->
2 <mat-toolbar color="primary" class="app-header">
3   <div><a href="https://gsi.berkeley.edu/gsi-guide-contents/technology-intro/gsi-examples/online-di
4   <span class="nav-tool-items">
5     <a mat-button routerLink="login" routerLinkActive="active">Log in</a>
6     <a mat-button routerLink="Register" routerLinkActive="active">Register</a>
7     <a mat-button routerLink="question" routerLinkActive="active">Question</a>
8     <a mat-button routerLink="answer" routerLinkActive="active">Answer</a>
9   </span>
10 </div>
11 </mat-toolbar>
12 <router-outlet></router-outlet>
```

Appcomponent.Ts

```
<> app.component.html <> index.html TS app.component.ts X TS
src > app > TS app.component.ts > ...
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'LearningPortal';
10 }
11
```

Appmodule.Ts

```
import { BrowserModule } from '@angular/platform-browser';
/* Routing */
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
/* Angular Material */
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';
import { AngularMaterialModule } from './angular-material.module';
import { NgModule, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
/* FormsModule */
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
/* Angular Flex Layout */
import { FlexLayoutModule } from "@angular/flex-layout";
/* Components */
import { LogInComponent } from './components/log-in/log-in.component';
import { RegisterComponent } from './components/register/register.component';
import { QuestionComponent } from './components/question/question.component';
import { AnswerComponent } from './components/answer/answer.component';
import { HttpClientModule } from '@angular/common/http';
import { HttpClientInMemoryWebApiModule } from 'angular-in-memory-web-api';

@NgModule({
  declarations: [
    AppComponent,
    LogInComponent,
    LogInComponent,
    RegisterComponent,
    QuestionComponent,
    AnswerComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    AngularMaterialModule,
    ReactiveFormsModule,
    FormsModule,
    FlexLayoutModule,
    HttpClientModule,
    HttpClientInMemoryWebApiModule
  ],
  providers: [],
  bootstrap: [AppComponent],
})
```

```
    schemas: [CUSTOM_ELEMENTS_SCHEMA]
  })
  export class AppModule { }
```

App-routing.module.ts

```
TS app-routing.module.ts • <> question.component.html # question.component.css TS log-in.c
src > app > TS app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { AnswerComponent } from '../components/answer/answer.component';
4  import { LoginComponent } from '../components/log-in/log-in.component';
5  import { QuestionComponent } from '../components/question/question.component';
6  import { RegisterComponent } from '../components/register/register.component';
7
8  const routes: Routes = [
9    { path: '', pathMatch: 'full', redirectTo: 'login' },
10   { path: 'login', component: LoginComponent },
11   { path: 'register', component: RegisterComponent },
12   { path: 'Question', component: QuestionComponent },
13   { path: 'Answer', component: AnswerComponent }
14
15 ];
16 @NgModule({
17   imports: [RouterModule.forRoot(routes)],
18   exports: [RouterModule]
19 })
20 export class AppRoutingModule { }
21
```

answer.component.html

```

<div class="answer-wrapper" fxLayout="row" fxLayoutAlign="center center">
  <mat-card class="box">
    <mat-card-header>
      <mat-card-title>Answer</mat-card-title>
    </mat-card-header>
    <form class="example-form">
      <mat-card-content>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="SMEAnswer">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="ReviewerDescription">
        </mat-form-field>
      </mat-card-content>
      <button mat-stroked-button color="accent" class="btn-block">Log in</button>
    </form>
  </mat-card>
</div>

```

log-in.component.html

```

<div class="login-wrapper" fxLayout="row" fxLayoutAlign="center center">
  <mat-card class="box">
    <mat-card-header>
      <mat-card-title>Log in</mat-card-title>
    </mat-card-header>
    <form class="example-form">
      <mat-card-content>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="Username">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="Password">
        </mat-form-field>
      </mat-card-content>
      <button mat-stroked-button color="accent" class="btn-block">Log
in</button>
    </form>
  </mat-card>
</div>

```

question.component.html

```
<div class="question-wrapper" fxLayout="row" fxLayoutAlign="center center">
  <mat-card class="box">
    <mat-card-header>
      <mat-card-title>Question</mat-card-title>
    </mat-card-header>
    <form class="example-form">
      <mat-card-content>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="PostQuestion">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="ReviewerDescription">
        </mat-form-field>
      </mat-card-content>
      <button mat-stroked-button color="accent" class="btn-block">Log
in</button>
    </form>
  </mat-card>
</div>
```

register.component.html

```
<div class="login-wrapper" fxLayout="row" fxLayoutAlign="center center">
  <mat-card class="box">
    <mat-card-header>
      <mat-card-title>Register</mat-card-title>
    </mat-card-header>
    <form class="example-form">
      <mat-card-content>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="Username">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="Email">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <input matInput placeholder="Password">
        </mat-form-field>
        <mat-form-field class="example-full-width">
          <mat-label>Choose a role...</mat-label>
          <mat-select>
            <mat-option [value]="roles" *ngFor="let roles of
Roles">{{roles}}</mat-option>
          </mat-select>
        </mat-form-field>
      </mat-card-content>
    </form>
  </mat-card>
</div>
```

```

        </mat-option>
      </mat-select>
    </mat-form-field>
  </mat-card-content>
  <button mat-stroked-button color="accent" class="btn-
block">Register</button>
</form>
</mat-card>
</div>

```

Style.css

```

html,
body {
  height: 100%;
}
body {
  margin: 0;
  font-family: Roboto, "Helvetica Neue", sans-serif;
  min-height: 100vh;
  background: #e2e2e2;
}
.app-header {
  justify-content: space-between;
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  z-index: 2;
  box-shadow: 0 3px 5px -1px rgba(0, 0, 0, .2), 0 6px 10px 0 rgba(0, 0, 0,
.14), 0 1px 18px 0 rgba(0, 0, 0, .12);
}
.login-wrapper {
  height: 100%;
}
.DiscussionForum {
  text-decoration: none;
  color: lightslategray;
}
.box {
  position: relative;
  top: 0;

```

```
opacity: 1;
float: left;
padding: 60px 50px 40px 50px;
width: 100%;
background: #fff;
border-radius: 10px;
transform: scale(1);
-webkit-transform: scale(1);
-ms-transform: scale(1);
z-index: 5;
max-width: 330px;
}
.box.back {
  transform: scale(.95);
  -webkit-transform: scale(.95);
  -ms-transform: scale(.95);
  top: -20px;
  opacity: .8;
  z-index: -1;
}
.box:before {
  content: "";
  width: 100%;
  height: 30px;
  border-radius: 10px;
  position: absolute;
  top: -10px;
  background: rgba(255, 255, 255, .6);
  left: 0;
  transform: scale(.95);
  -webkit-transform: scale(.95);
  -ms-transform: scale(.95);
  z-index: -1;
}
.login-wrapper .example-form {
  min-width: 100%;
  max-width: 300px;
  width: 100%;
}
.login-wrapper .example-full-width,
.login-wrapper .btn-block {
  width: 100%;
}
.login-wrapper mat-card-header {
  text-align: center;
  width: 100%;
  display: block;
  font-weight: 700;
```

```

}
.login-wrapper mat-card-header mat-card-title {
  font-size: 30px;
  margin: 0;
}
.login-wrapper .mat-card {
  padding: 40px 70px 50px;
}
.login-wrapper .mat-stroked-button {
  border: 1px solid currentColor;
  line-height: 54px;
  background: lightskyblue;
}
.login-wrapper .mat-form-field-appearance-legacy .mat-form-field-infix {
  padding: 0.8375em 0;
}

```

Add connection

web-request.service.ts

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class WebRequestService {
  readonly ROOT_URL;

  constructor(private http: HttpClient) {
    this.ROOT_URL='http://localhost:5072'
  }
  get(url: string){
    return this.http.get(`${this.ROOT_URL}/${uri}`);
  }

}

```


task.service.ts

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class TaskService {

  constructor() { }
  createList(title :string){

  }

}
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  > node + v [] [X]

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\dotnetadmin\LearningPortal> ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor         | 5.37 MB |
polyfills.js        | polyfills      | 294.85 kB |
styles.css, styles.js | styles        | 252.98 kB |
main.js             | main           | 49.55 kB |
runtime.js          | runtime        | 6.53 kB |
                    | Initial Total  | 5.96 MB

Build at: 2022-06-01T15:33:30.410Z - Hash: e9a156ab3de46eda - Time: 43050ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
█
```

[Home](#) [Questions](#) [Forum](#) [Ask Question](#)

Email :

Password :

First Name :

Last Name :

Upload Photo :

Choose File

 No file chosen

Register

LearningPortal

localhost:4200/login

crome crome New Tab Search Gmail YouTube Maps

[DiscussionForum](#) [Log in](#) [Register](#) [Question](#) [Answer](#)

Log in

Username

Password

Log in

