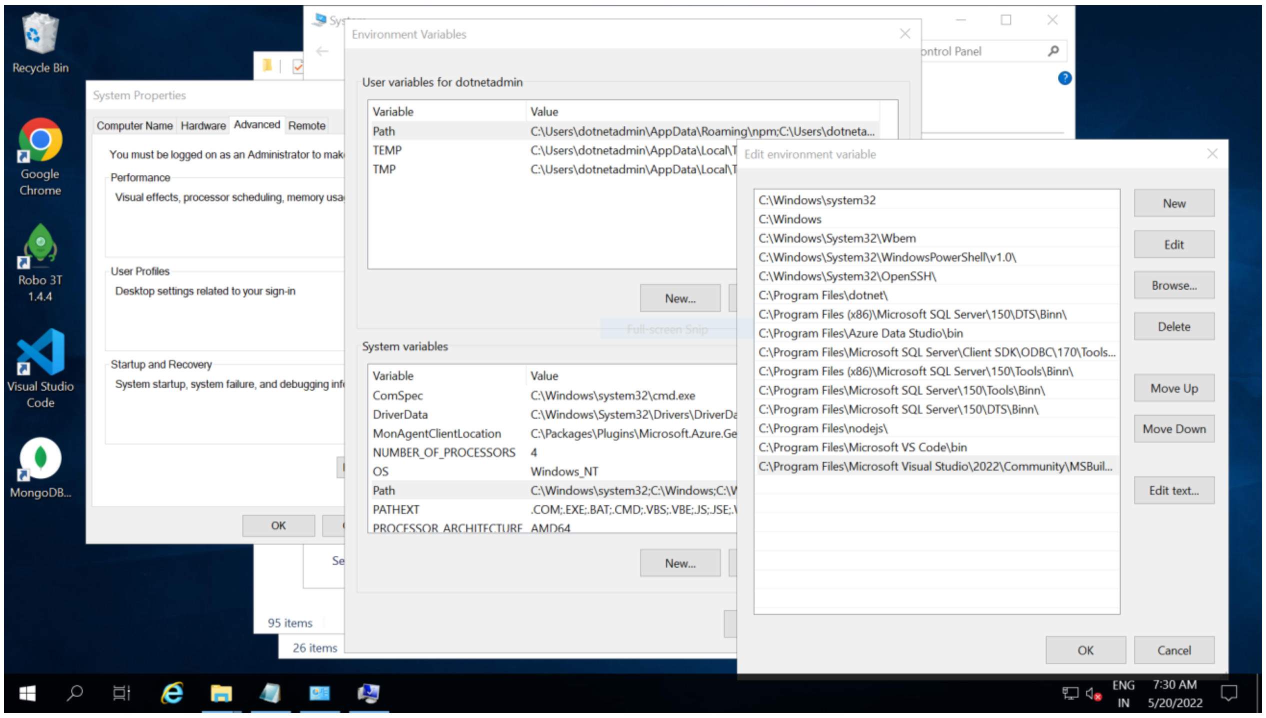
Milestone Assessment-6

Name: Priyansu Jena

Mid : M1082626

Path setting:



User variables for dotnetadmin

Variable	Value
Path	C:\Users\dotnetadmin\AppData\Roaming\npm;C:\Users\dotneta...
TEMP	C:\Users\dotnetadmin\AppData\Local\T
TMP	C:\Users\dotnetadmin\AppData\Local\T

System variables

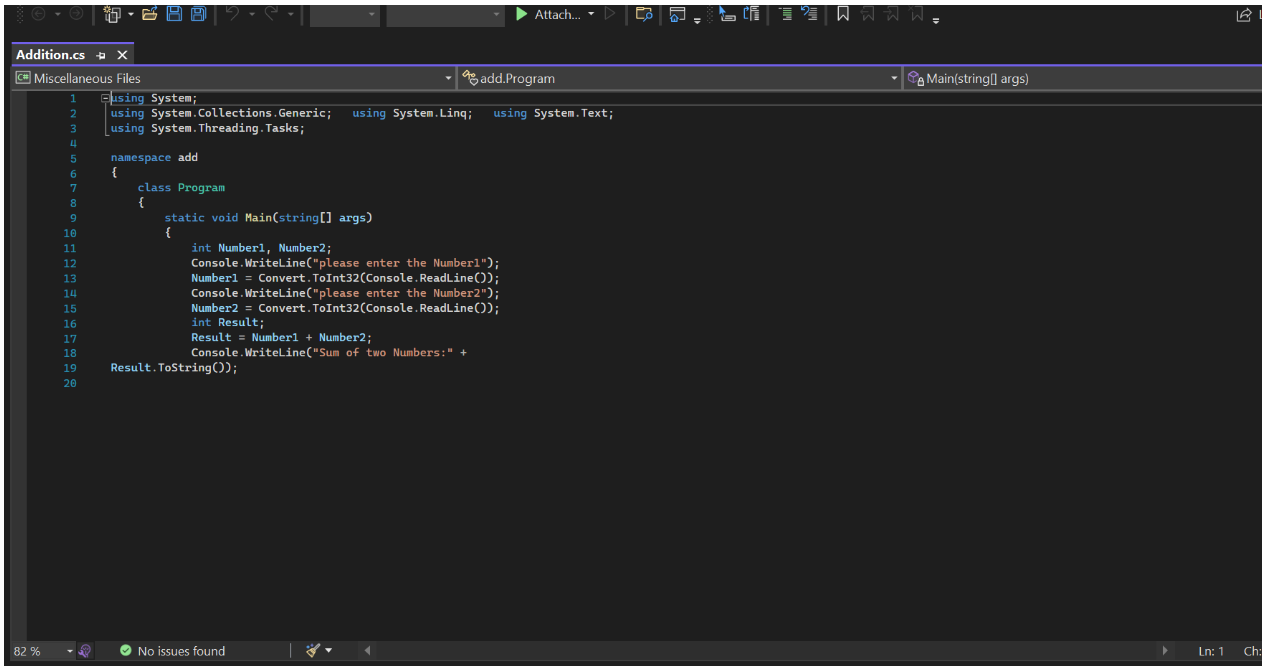
Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverDa
MonAgentClientLocation	C:\Packages\Plugins\Microsoft.Azure.Ge
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\W
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.\
PROCESSOR_ARCHITECTURE	AMD64

```
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files\dotnet\
C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files\Azure Data Studio\bin
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools...
C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files\nodejs\
C:\Program Files\Microsoft VS Code\bin
C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuil...
```

MyMSBuildSolution.csproj

```
<Project InitialTargets="MyMSBuildSolution" ToolsVersion="Current">
  <PropertyGroup>
    <Basepath>C:\MyCodeApp</Basepath>
    <Srcpath>$(Basepath)\Src</Srcpath>
    <Destpath>$(Basepath)\Out</Destpath>
    <FileUpgradeFlags>
    </FileUpgradeFlags>
    <UpgradeBackupLocation>
    </UpgradeBackupLocation>
    <OldToolsVersion>2.0</OldToolsVersion>
  </PropertyGroup>
  <ItemGroup>
```

```
    <Srcfiles Include="C:\MyCodeApp\Src\Addition.cs">
    </Srcfiles>
  </ItemGroup>
  <Target Name="EnvSetup">
    <MakeDir Directories="$(Srcpath)">
    </MakeDir>
    <MakeDir Directories="$(Destpath)">
    </MakeDir>
  </Target>
  <Target Name="MyMSBuildSolution" DependsOnTargets="EnvSetup">
    <Copy SourceFiles="@(srcfiles)" DestinationFolder="$(Srcpath)">
    </Copy>
    <Csc Sources="$(Srcpath)\*.cs"
OutputAssembly="$(Destpath)\MyMSBuildSolution.exe">
    </Csc>
  </Target>
</Project>
```

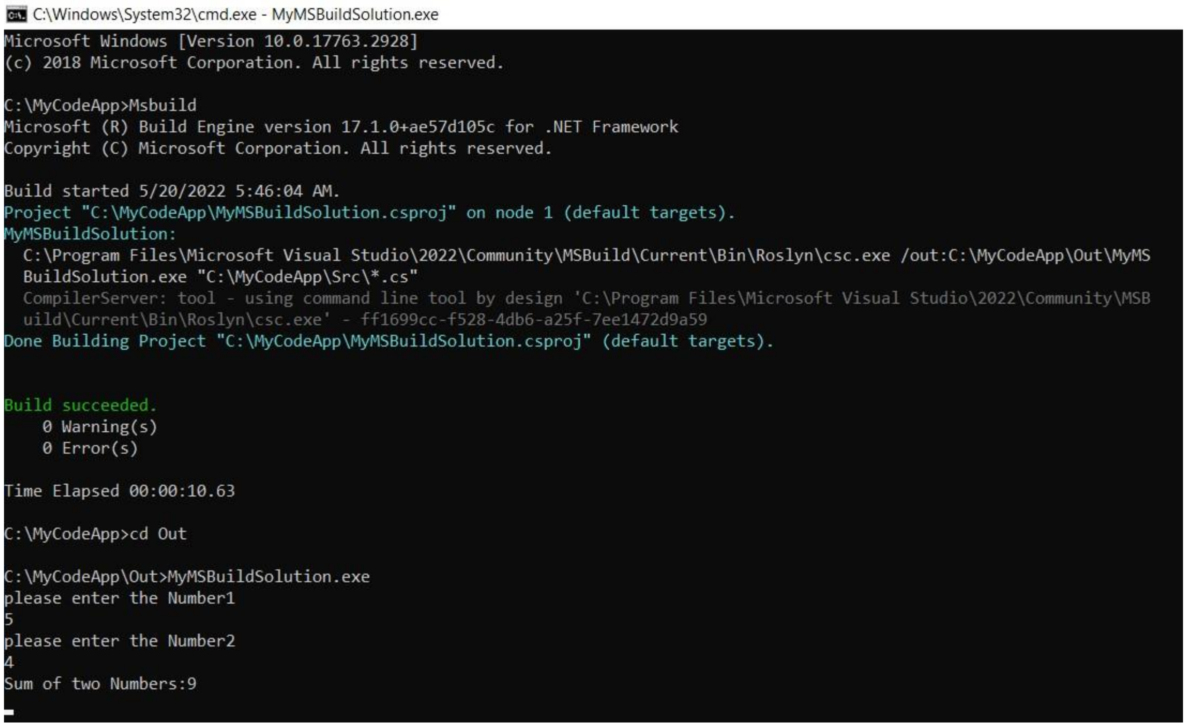


```
using System;
using System.Collections.Generic;   using System.Linq;   using System.Text;
using System.Threading.Tasks;

namespace add
{
    class Program
    {
        static void Main(string[] args)
        {
            int Number1, Number2;
            Console.WriteLine("please enter the Number1");
            Number1 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("please enter the Number2");
            Number2 = Convert.ToInt32(Console.ReadLine());
            int Result;
            Result = Number1 + Number2;
            Console.WriteLine("Sum of two Numbers:" +
Result.ToString());
```

Addition of two Numbers

```
using System;
using System.Collections.Generic;
using System.Linq;   using
System.Text;
using System.Threading.Tasks;

namespace add
{
    class Program
    {
        static void Main(string[] args)
        {
            int Number1, Number2;
            Console.WriteLine("please enter the Number1");
            Number1 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("please enter the Number2");
Number2 = Convert.ToInt32(Console.ReadLine());              int
Result;
            Result = Number1 + Number2;
            Console.WriteLine("Sum of two Numbers:" +
Result.ToString());
            Console.ReadLine();
        }
    }
}
```

Output



```
C:\Windows\System32\cmd.exe - MyMSBuildSolution.exe
Microsoft Windows [Version 10.0.17763.2928]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\MyCodeApp>Msbuild
Microsoft (R) Build Engine version 17.1.0+ae57d105c for .NET Framework
Copyright (C) Microsoft Corporation. All rights reserved.

Build started 5/20/2022 5:46:04 AM.
Project "C:\MyCodeApp\MyMSBuildSolution.csproj" on node 1 (default targets).
MyMSBuildSolution:
  C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Current\Bin\Roslyn\csc.exe /out:C:\MyCodeApp\Out\MyMS
  BuildSolution.exe "C:\MyCodeApp\Src\*.cs"
  CompilerServer: tool - using command line tool by design 'C:\Program Files\Microsoft Visual Studio\2022\Community\MSB
  uild\Current\Bin\Roslyn\csc.exe' - ff1699cc-f528-4db6-a25f-7ee1472d9a59
Done Building Project "C:\MyCodeApp\MyMSBuildSolution.csproj" (default targets).


Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:10.63

C:\MyCodeApp>cd Out

C:\MyCodeApp\Out>MyMSBuildSolution.exe
please enter the Number1
5
please enter the Number2
4
Sum of two Numbers:9
```

Case study 2

A) EPIC:
User Registration to a page for booking a ticket(or)Screen

USER STORIES/TASKS:Assigned to Rajesh

<As a new User>
<I want to register myself with my details including username and password >
<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
3. Add an option 'AlreadyanUser' so the user can directly login into page as the details were already stored.

B) EPIC:
User Login to the page

USER STORIES/TASKS:Assigned to Abhisek

<As a User>
<I want to login to the page with my username and password only>
<So that i dont 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:
Searching for the movie in the search box based on Location and Screeing name

USER STORIES/TASKS:Assigned to Bishnu
<As a User>
<I want to locate to search bar and have option like city and screen name to be select as per choice>
<so that i can easily select my prefered city and screenname instead of searching all of the Menu page>

SUBTASKS:
1. Add options City and ScreenName to the search bar
2. Add city names in it
3. Add Screen Names in it so user can easily select

D) EPIC:
BOOKING TICKET

USER STORIES/TASKS:Assigned to Babul
<As a User>
<I want to select my prefer movie with my preferable date and time>
<So that i am available at the time or date>

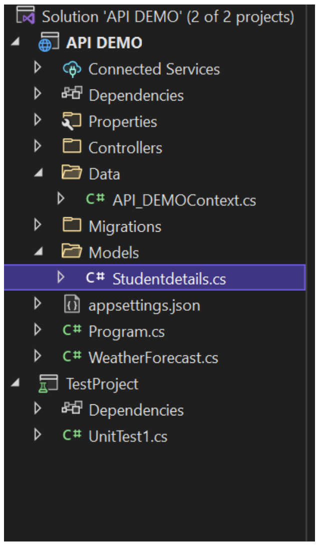
SUBTASKS:
1. enable the user to select his preferred movie by giving select option rightside of the screen name
2. Add Calender and TimeLine so that they can book there wanted date

E) EPIC:
CANCELLING THE TICKET

USER STORIES/TASKS:Assigned to Subrat
<As a user>
< I Want to have a option like cancelling the movie ticket before the booked date>
<So that i can cancel my show if i am not available>

SUBTASKS:
1.Add Cancel Booking in My showList
2.Make the date to be visible one day prior to the booked date to get cancelled

Case study 3

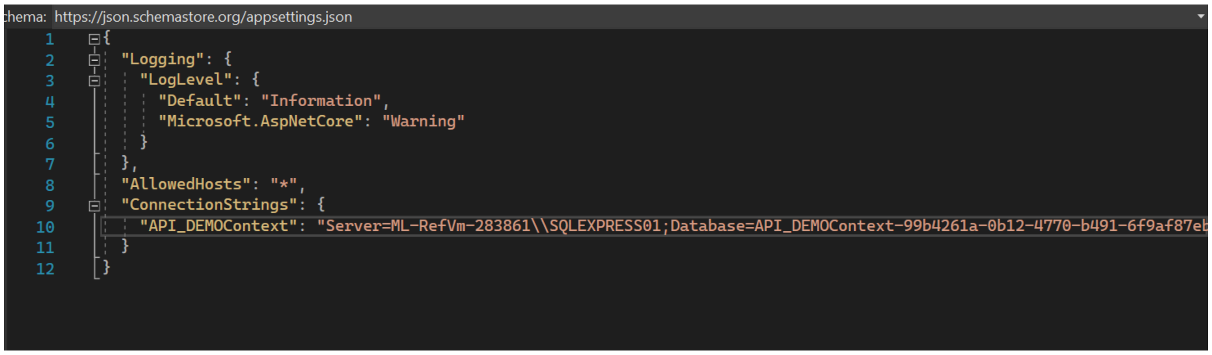


```
Solution 'API DEMO' (2 of 2 projects)
◢ API DEMO
    ▷ Connected Services
    ▷ Dependencies
    ▷ Properties
    ▷ Controllers
    ◢ Data
        ▷ API_DEMOContext.cs
    ▷ Migrations
    ◢ Models
        ▷ Studentdetails.cs
    ▷ appsettings.json
    ▷ Program.cs
    ▷ WeatherForecast.cs
◢ TestProject
    ▷ Dependencies
    ▷ UnitTest1.cs
```

StudentDetails.cs

```
using System.ComponentModel.DataAnnotations;
namespace API_DEMO.Models
{
    public class Studentdetails
    {
        [Key]
        public Guid id { get; set; }
        public string StudentName { get; set; }
        public string StudentDepartment { get; set; }
        public int DepartmentId { get; set; }

    }
}
```

Appsetting.json



```
chema:  https://json.schemastore.org/appsettings.json
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "API_DEMOContext": "Server=ML-RefVm-283861\\SQLEXPRESS01;Database=API_DEMOContext-99b4261a-0b12-4770-b491-6f9af87eb
  }
}
```

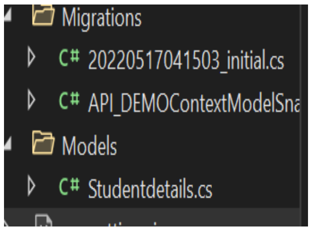
Controller

```
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 API_DEMO.Data;
using API_DEMO.Models;

namespace API_DEMO.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class PaymentdetailsController : ControllerBase
    {
        private readonly API_DEMOContext _context;

        public PaymentdetailsController(API_DEMOContext context)
        {
            _context = context;
```

```
}


[HttpGet]
public async Task<ActionResult<IEnumerable<Studentdetails>>> GetStudentdetails()
{
 if (_context.Studentdetails == null)
 {
    return NotFound();
 }
  return await _context.Studentdetails.ToListAsync();
}


[HttpGet("{id}")]
public async Task<ActionResult<Studentdetails>> GetStudentdetails(Guid id)
{
 if (_context.Studentdetails == null)
 {
    return NotFound();
 }
  var Studentdetails = await _context.Studentdetails.FindAsync(id);

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

  return Studentdetails;
}



[HttpPut("{id}")]
public async Task<IActionResult> Putstudentdetails(Guid id, Studentdetails studentdetails)
{
  if (id != Studentdetails.id)
  {
     return BadRequest();
  }

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

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

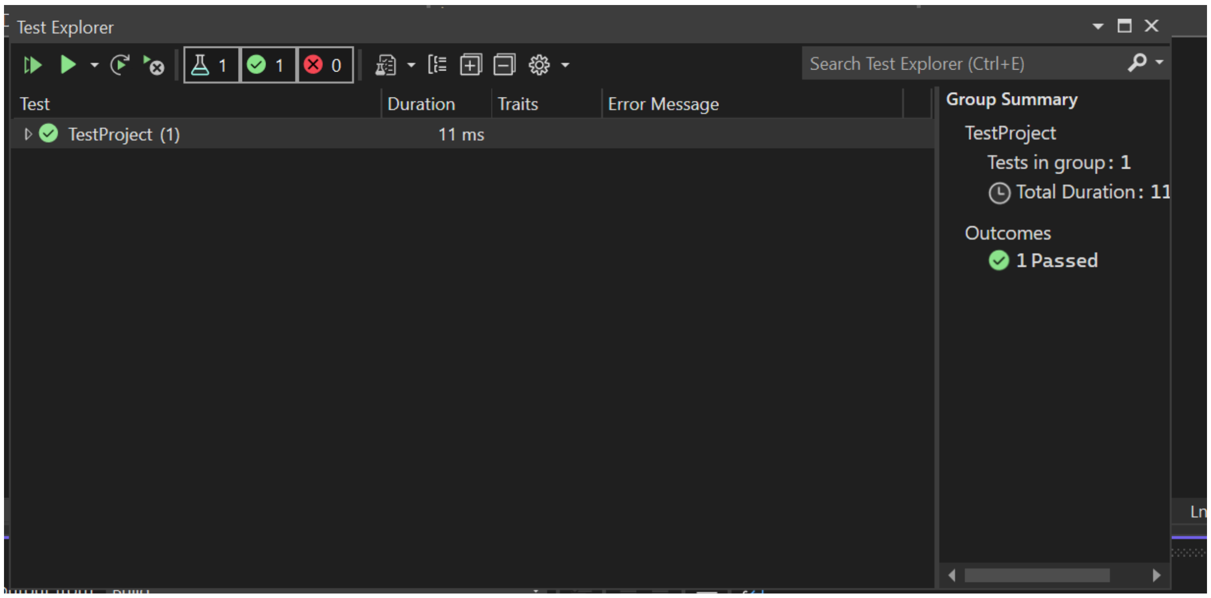
```
                throw;
            }
        }

        return NoContent();
    }


    [HttpPost]
    public async Task<ActionResult<Studentdetails>> PostStudentdetails(Studentdetails Studentdetails)
    {
     if (_context.Studentdetails == null)
     {
         return Problem("Entity set 'API_DEMOContext.Studentdetails'  is null.");
     }
       _context.Studentdetails.Add(Studentdetails);
       await _context.SaveChangesAsync();

       return CreatedAtAction("GetStudentdetails", new { id = Studentdetails.id }, Studentdetails);
    }


    [HttpDelete("{id}")]
    public async Task<IActionResult> DeleteStudentdetails(Guid id)
    {
        if (_context.Studentdetails == null)
        {
            return NotFound();
        }
        var Studentdetailss = await _context.Studentdetails.FindAsync(id);
        if (Studentdetails == null)
        {
            return NotFound();
        }

        _context.Studentdetails.Remove(Studentdetails);
        await _context.SaveChangesAsync();

        return NoContent();
    }

    private bool StudentdetailsExists(Guid id)
    {
        return (_context.Studentdetails?.Any(e => e.id == id)).GetValueOrDefault();
    }
  }
}
```

Migration



Migrations
C# 20220517041503_initial.cs
C# API_DEMOContextModelSna
Models
C# Studentdetails.cs

Ms test



Test Explorer

Test	Duration	Traits	Error Message
TestProject (1)	11 ms		

Group Summary
TestProject
Tests in group : 1
Total Duration : 11
Outcomes
1 Passed