

SIES COLLEGE OF ARTS, SCIENCE & COMMERCE
(AUTONOMOUS) SION(W), MUMBAI-22

DEPARTMENT OF INFORMATION TECHNOLOGY

MSc (IT), SEMESTER II

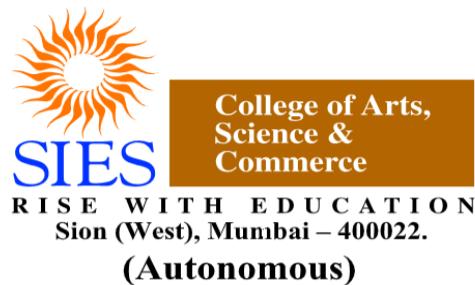
Practical Journal
for
the subject

Microservices Architecture

Submitted by
Mayank Mansukh Khetiya

FMSC2324226

for the academic year
2023-2024



SIES College of Arts, Science and Commerce (Autonomous)
Sion (W), Mumbai – 400 022.

Department of Information Technology

CERTIFICATE

This is to certify that Mr./MS. Mayank Khetiya of MSc [Information Technology],
Part I, Seat No FMSC2324226 has successfully completed the practical's and submitted it
online in Microsoft Teams for the subject of Microservices Architecture as a partial
fulfilment of the degree MSc (IT) during the academic year 2023-2024.

Faculty-in-charge

Iqra Shaikh

Internal Examiner

External Examiner

Date:15/04/2024

College Seal

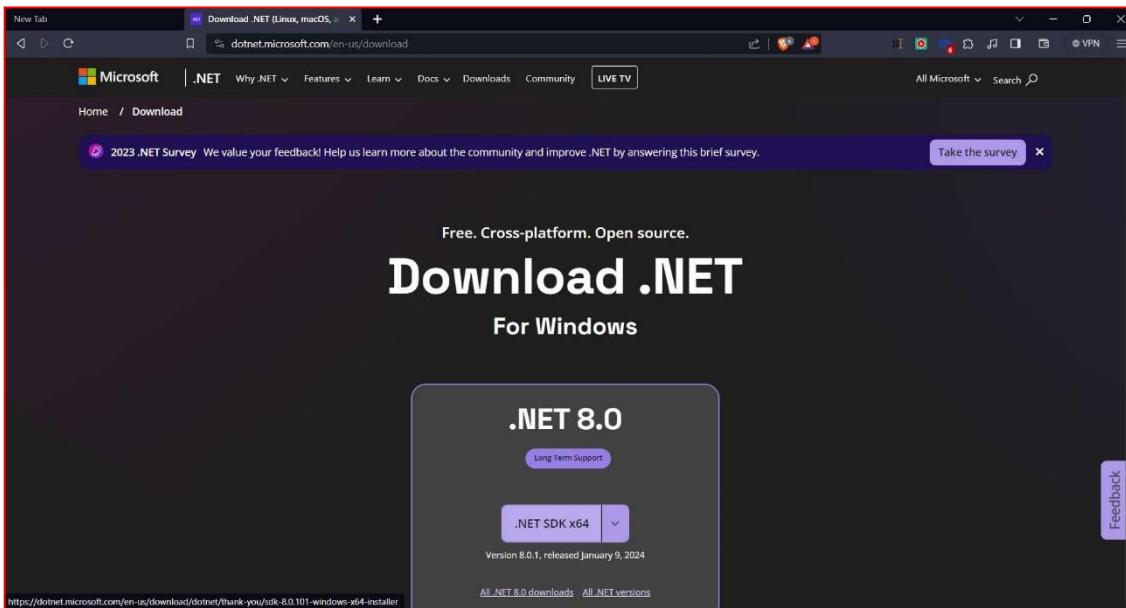
SR. NO.	PRACTICAL TITLE	SIGNATURE
1	Building .Net Core MVC (BookStore)	
2(A)	Building ASP.Net Core REST API (using command prompt)	
2(B)	Building ASP.Net Core REST API (using Visual Studio)	
3	Building .Net Core MVC (Hello World & StockQuote)	
4	Working with Docker Commands, Images and Container	
5	Working with Docker Volumes and Networks (Play With Docker)	
6	Working with Docker Swarm	
7	Working with CircleCI	
8	Working with Kubernetes	

Practical 1

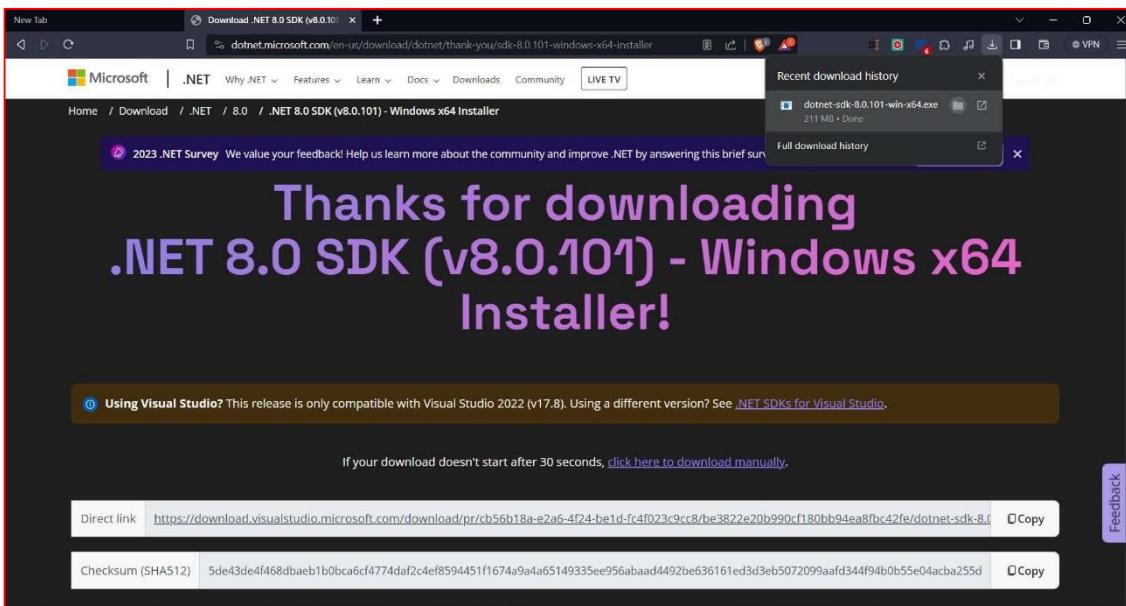
AIM:- Building ASP.NET Core MVC (BOOKSTORE)

Step 1: Install .Net Core Sdk from official website

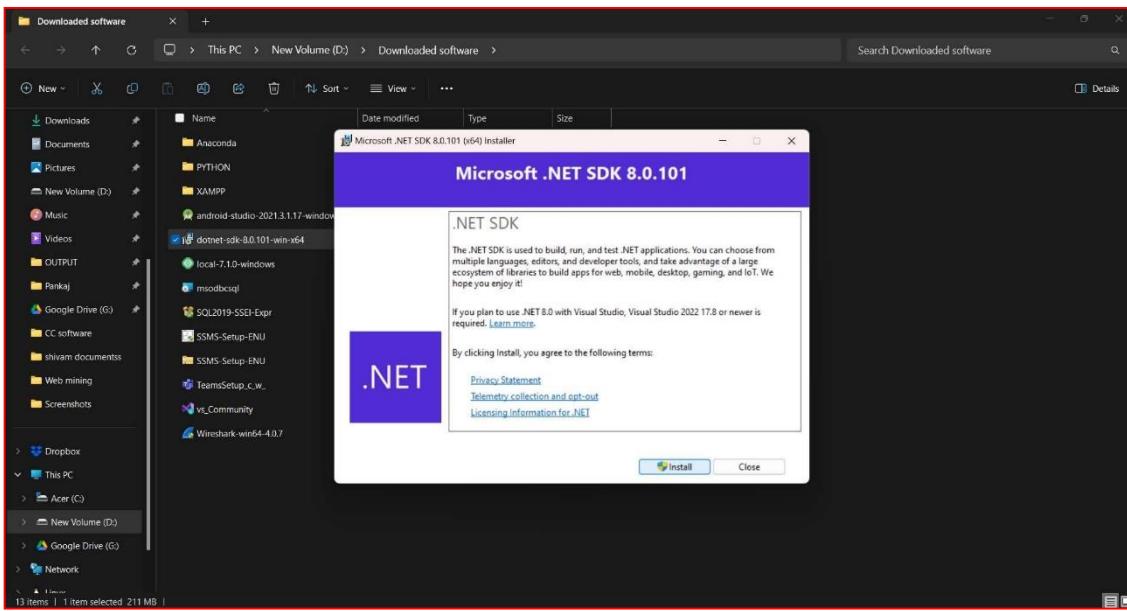
(Link :- <https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/sdk-8.0.101-windows-x64-installer> or <https://dotnet.microsoft.com/en-us/download>)



Open the location of downloaded file and install it.



Install the .NetCore properly and then follow the next step



Step 2: Create a folder named BookStore in D: drive or any other drive using the command prompt

Step 3: Open command prompt and perform following operations in order to create the above folder and performing the further steps

Commands:

(First change the directory to your desired drive here the used directory is D: drive)

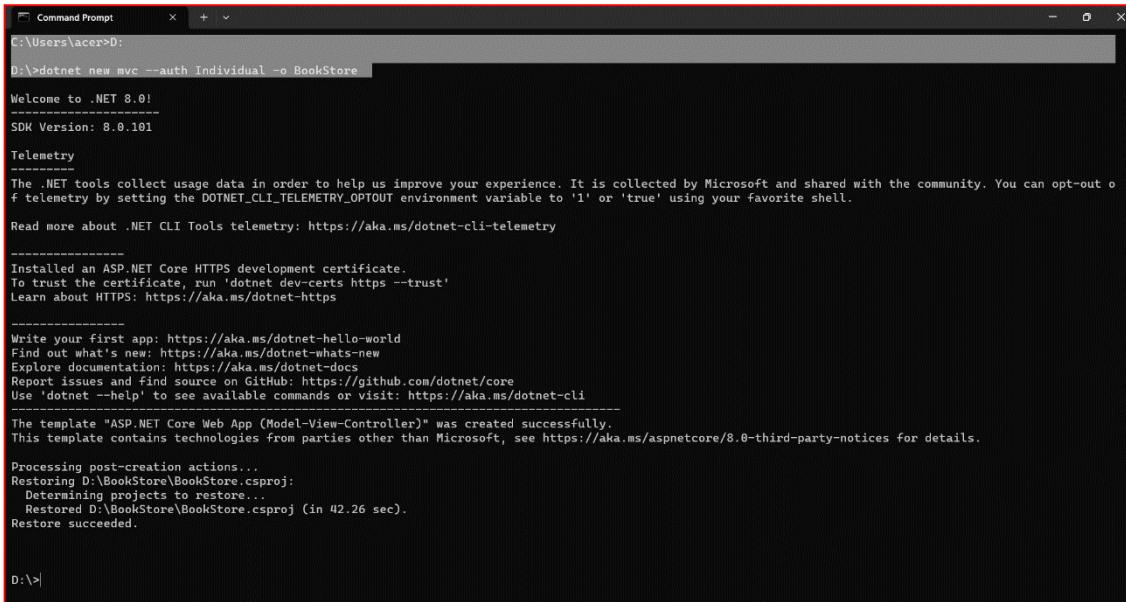
Command 1: dotnet new mvc --auth Individual -o BookStore

Command 2: cd BookStore

Command 3: dotnet run

Output (For command 1):-

Below it shows that the BookStore folder is been created successfully



```
C:\Users\acer>dotnet new mvc --auth Individual -o BookStore
Welcome to .NET 8.0!
-----
SDK Version: 8.0.101
Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate, run 'dotnet dev-certs https --trust'
Learn about HTTPS: https://aka.ms/dotnet-https

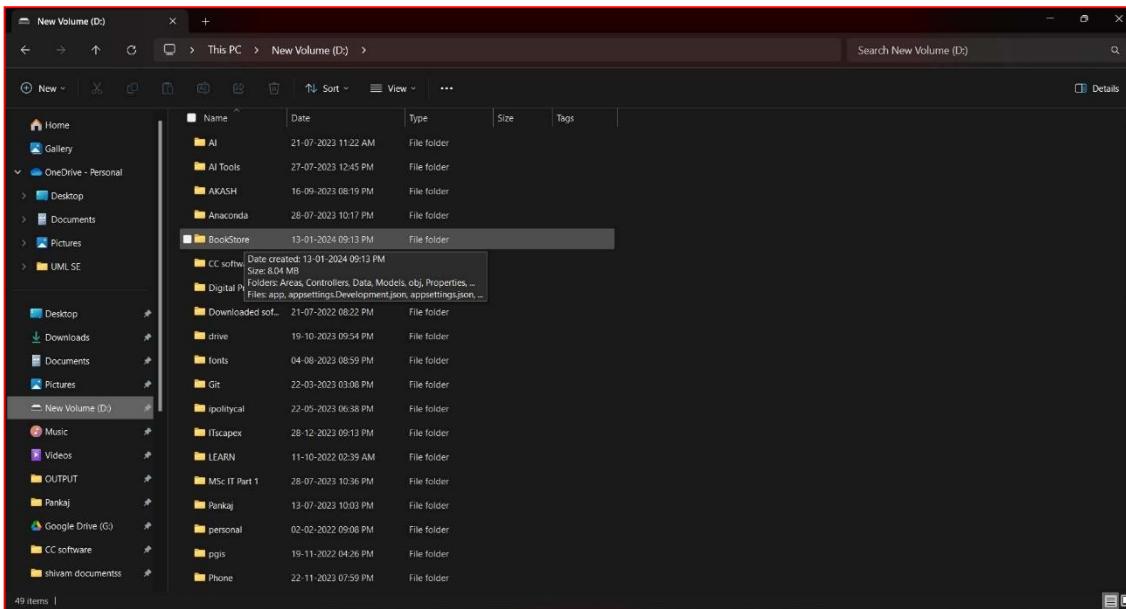
-----
Write your first app: https://aka.ms/dotnet-hello-world
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli

The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/8.0-third-party-notices for details.

Processing post-creation actions...
Restoring D:\BookStore\BookStore.csproj...
  Determining projects to restore...
  Restored D:\BookStore\BookStore.csproj (in 42.26 sec).
Restore succeeded.

D:\>|
```

The BookStore Folder is visible inside the D: drive

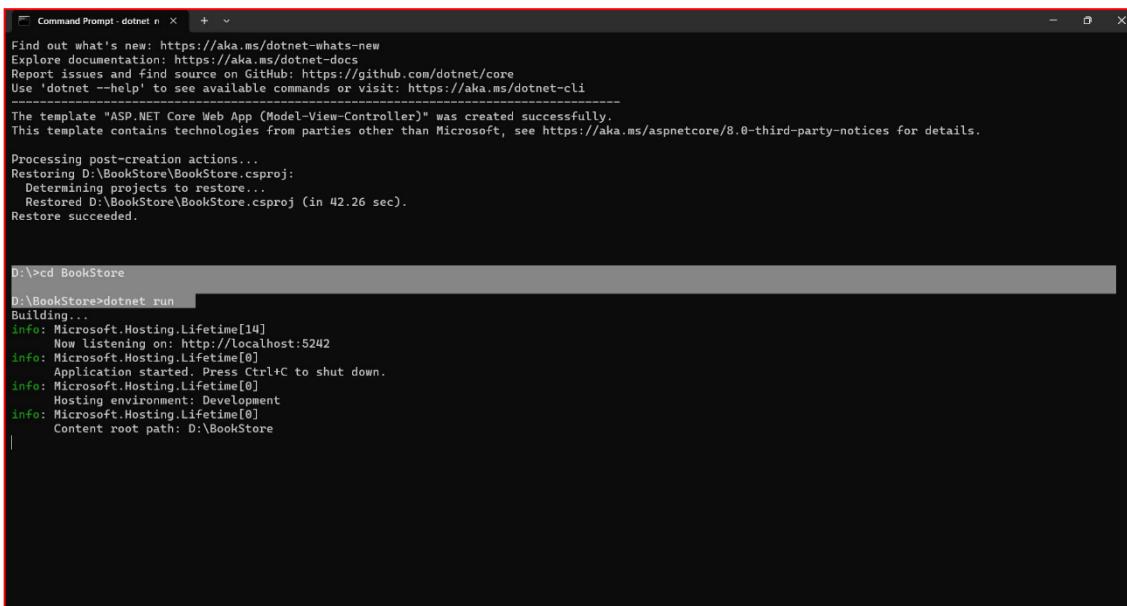


Output (For command 2 and command 3):-

Here our BookStore folder is been hosted on localhost from where we can access our bookstore.
Here you need to note the http link for further step.

i.e. <http://localhost:5242>

(Note that the http link will be different for different systems)



```

Command Prompt - dotnet run
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli

The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/8.0-third-party-notices for details.

Processing post-creation actions...
Restoring D:\BookStore\BookStore.csproj:
  Determining projects to restore...
  Restored D:\BookStore\BookStore.csproj (in 42.26 sec).
Restore succeeded.

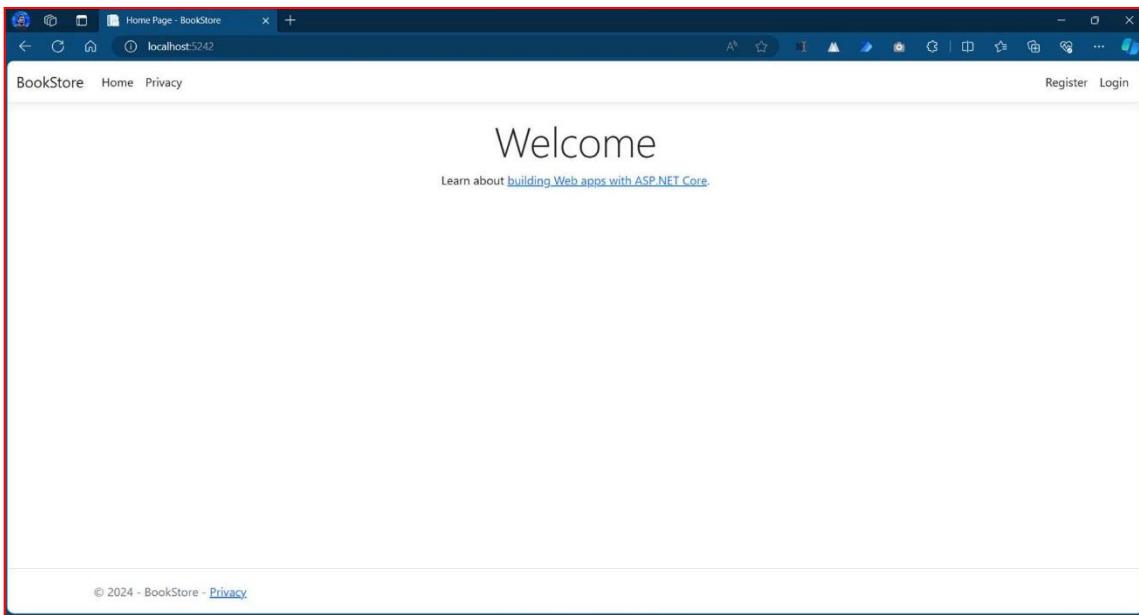
D:>cd BookStore
D:\BookStore>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5242
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\BookStore
|

```

Now you need to copy the noted http link or url i.e. <http://localhost:5242> in your browser



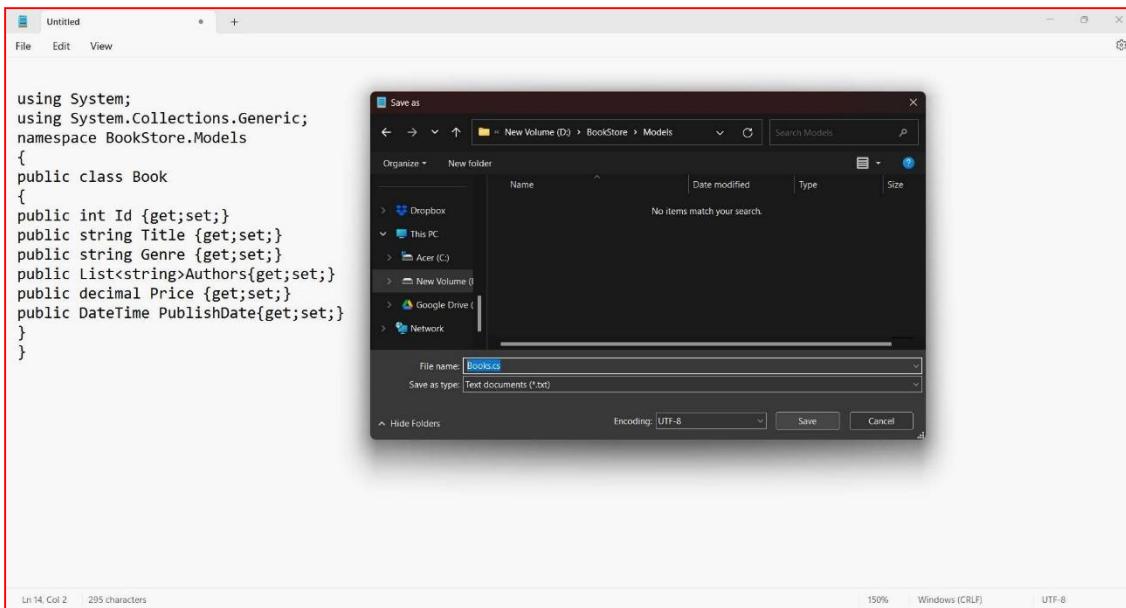
You will see the Home page or welcome page of your created bookstore



Step 4: Write the below code and add the following by saving it as ‘Books.cs’ file in the ‘Models’ folder which is inside the BookStore Folder

Code:-

```
using System;
using System.Collections.Generic;
namespace BookStore.Models
{
    public class Book
    {
        public int Id {get;set;}
        public string Title {get;set;}
        public string Genre {get;set;}
        public List<string> Authors{get;set;}
        public decimal Price {get;set;}
        public DateTime PublishDate{get;set;}
    }
}
```



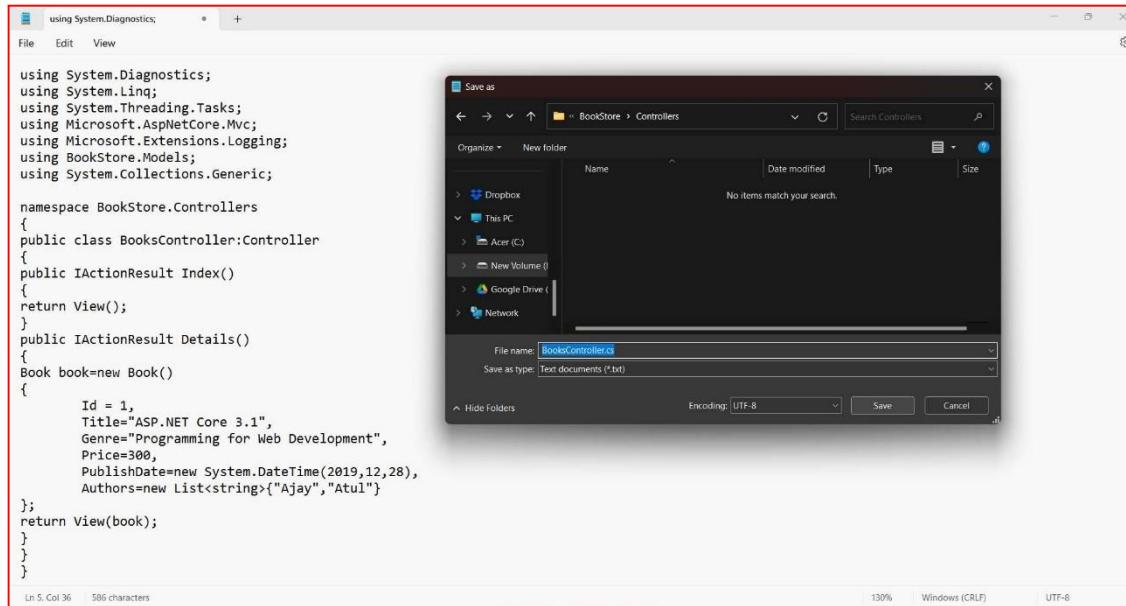
Step 5: Write the below code and add the following by saving it as '**BooksController.cs**' file in the '**Controllers**' folder which is inside the BookStore Folder

Code:-

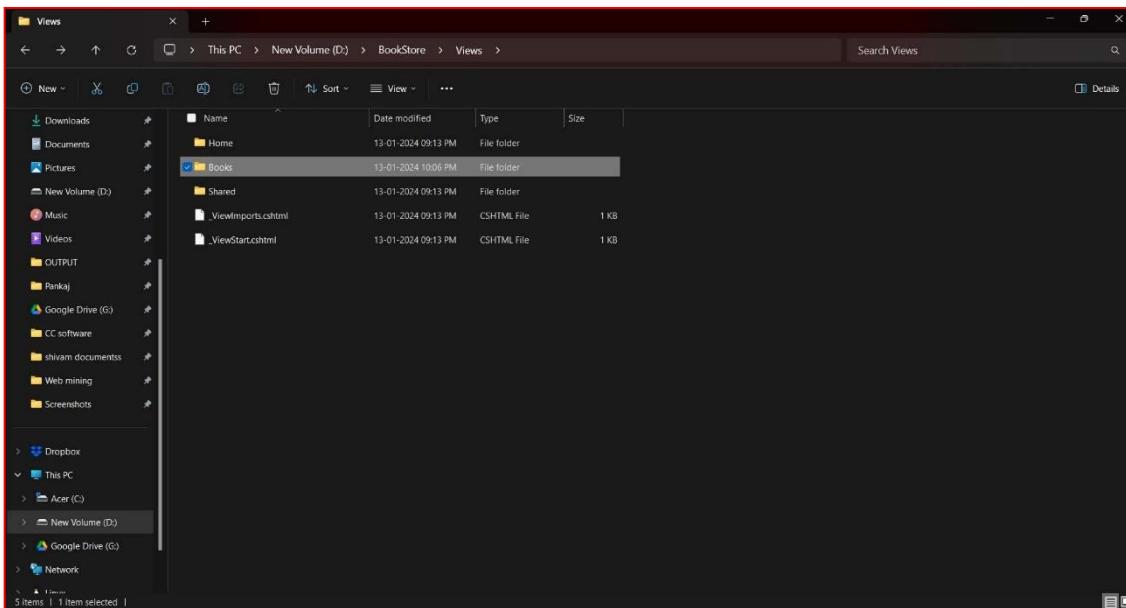
```
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using BookStore.Models;
using System.Collections.Generic;

namespace BookStore.Controllers
{
    public class BooksController:Controller
    {
        public IActionResult Index()
        {
            return View();
        }
        public IActionResult Details()
```

```
{  
    Book book=new Book()  
    {  
        Id = 1,  
        Title="ASP.NET Core 3.1",  
        Genre="Programming for Web Development",  
        Price=300,  
        PublishDate=new System.DateTime(2019,12,28),  
        Authors=new List<string>{"Ajay","Atul"}  
    };  
    return View(book);  
}  
}  
}  
}
```



Step 6: Go to the ‘Views’ Folder inside BookStore and add a new folder named ‘Books’.

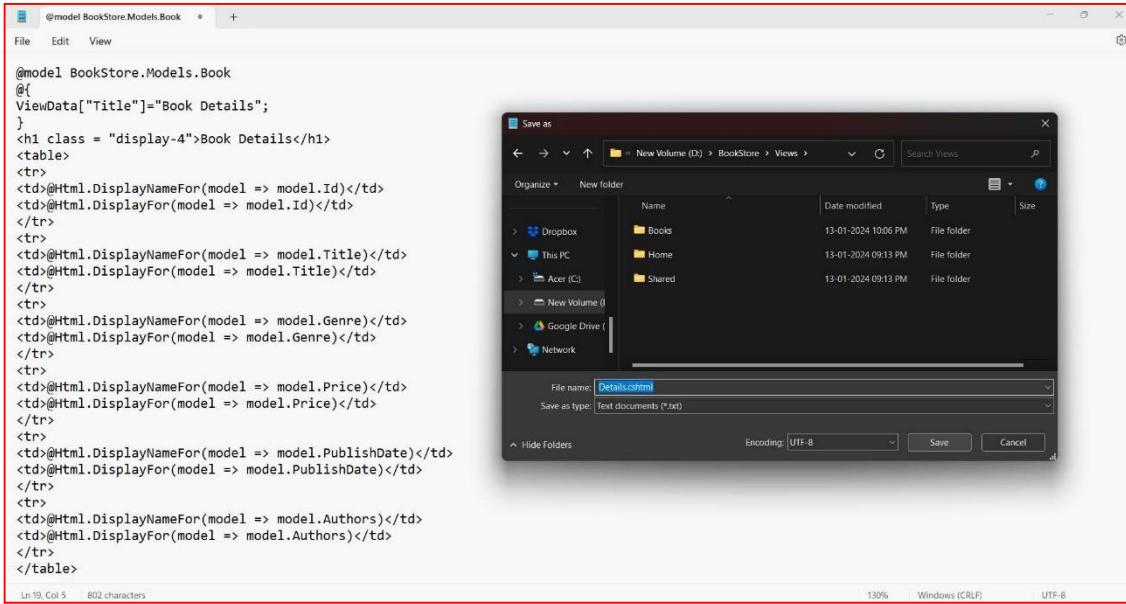


Add the file in Folders. Now write the below code and add the following by saving it as ‘Details.cshtml’ file in the ‘Views’ folder which is inside the BookStore Folder.

Code:-

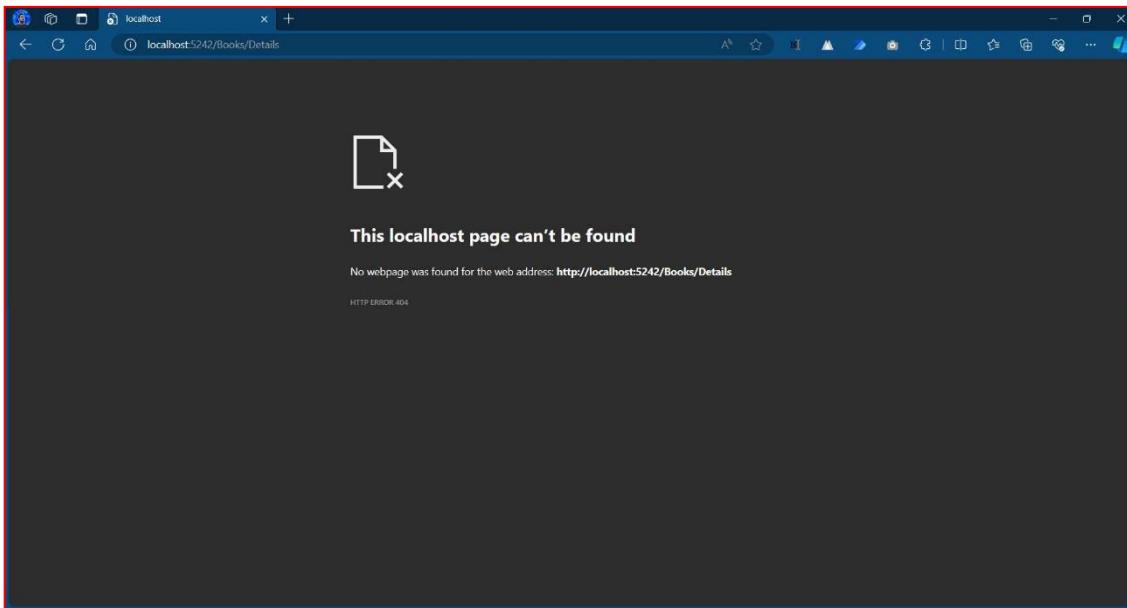
```
@model BookStore.Models.Book
@{
    ViewData["Title"] = "Book Details";
}
<h1 class = "display-4">Book Details</h1>
<table>
<tr>
<td>@Html.DisplayNameFor(model => model.Id)</td>
<td>@Html.DisplayFor(model => model.Id)</td>
</tr>
<tr>
<td>@Html.DisplayNameFor(model => model.Title)</td>
<td>@Html.DisplayFor(model => model.Title)</td>
</tr>
<tr>
<td>@Html.DisplayNameFor(model => model.Genre)</td>
```

```
<td>@Html.DisplayFor(model => model.Genre)</td>
</tr>
<tr>
<td>@Html.DisplayNameFor(model => model.Price)</td>
<td>@Html.DisplayFor(model => model.Price)</td>
</tr>
<tr>
<td>@Html.DisplayNameFor(model => model.PublishDate)</td>
<td>@Html.DisplayFor(model => model.PublishDate)</td>
</tr>
<tr>
<td>@Html.DisplayNameFor(model => model.Authors)</td>
<td>@Html.DisplayFor(model => model.Authors)</td>
</tr>
</table>
```

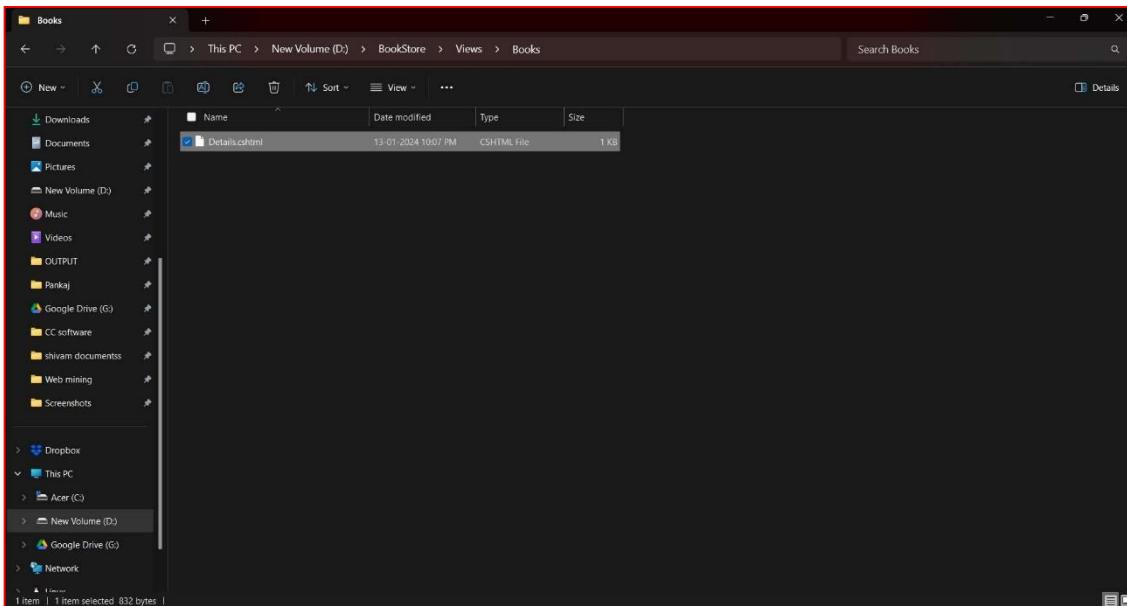


Final Step:

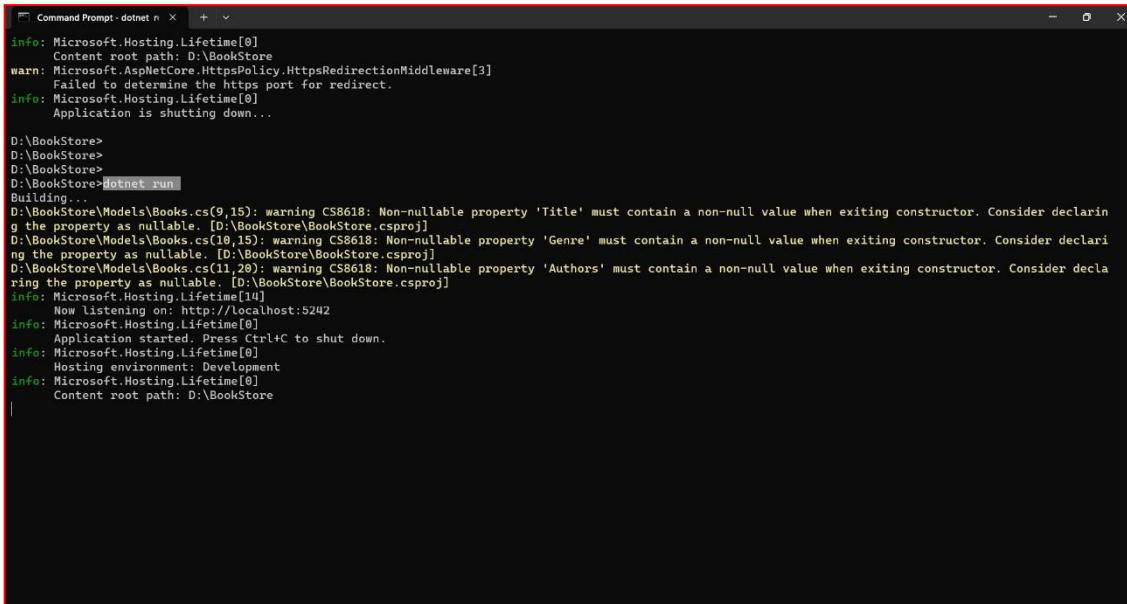
Now go back to your browser and add /Books/Details after the previously pasted http link of your bookstore i.e. <http://localhost:5242/Books/Details>. It will give you an error such as page does not found.



So, to resolve these issue you need to cut & paste the '**Details.cshtml**' file from 'Views' folder to the '**Books**' folder.



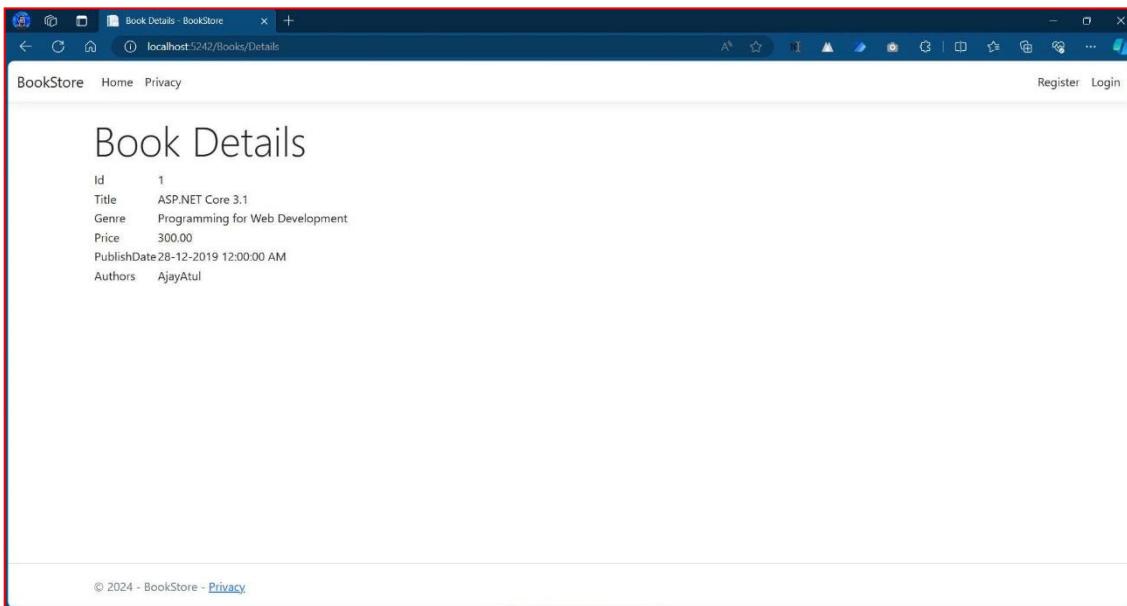
Before proceeding to your browser open the command prompt and click on “**CTRL+C**” and write the command ‘**dotnet run**’. If it shows the previous output without any errors then your output is correct and the desired page can be visited in the browser.



```
Command Prompt - dotnet run + ×
info: Microsoft.Hosting.Lifetime[0]
Content root path: D:\BookStore
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
Failed to determine the https port for redirect.
info: Microsoft.Hosting.Lifetime[0]
Application is shutting down...

D:\BookStore>
D:\BookStore>
D:\BookStore>
D:\BookStore>dotnet run
Building...
D:\BookStore\Models\Books.cs(9,15): warning CS8618: Non-nullable property 'Title' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [D:\BookStore\BookStore.csproj]
D:\BookStore\Models\Books.cs(18,15): warning CS8618: Non-nullable property 'Genre' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [D:\BookStore\BookStore.csproj]
D:\BookStore\Models\Books.cs(11,28): warning CS8618: Non-nullable property 'Authors' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [D:\BookStore\BookStore.csproj]
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5242
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: D:\BookStore
|
```

Now again go to your browser and refresh the page (<http://localhost:5242/Books/Details>)



Practical 2A

AIM:- BUILDING ASP .NET CORE REST API

Step 1:

1) Download and Install

To start building .NET apps you just need to download and install the .NET SDK (Software Development Kit version 3.0 above).

Link: <https://dotnet.microsoft.com/en-us/download/visual-studio-sdks>

2) Download the curl (Link: <https://curl.se/windows/>)

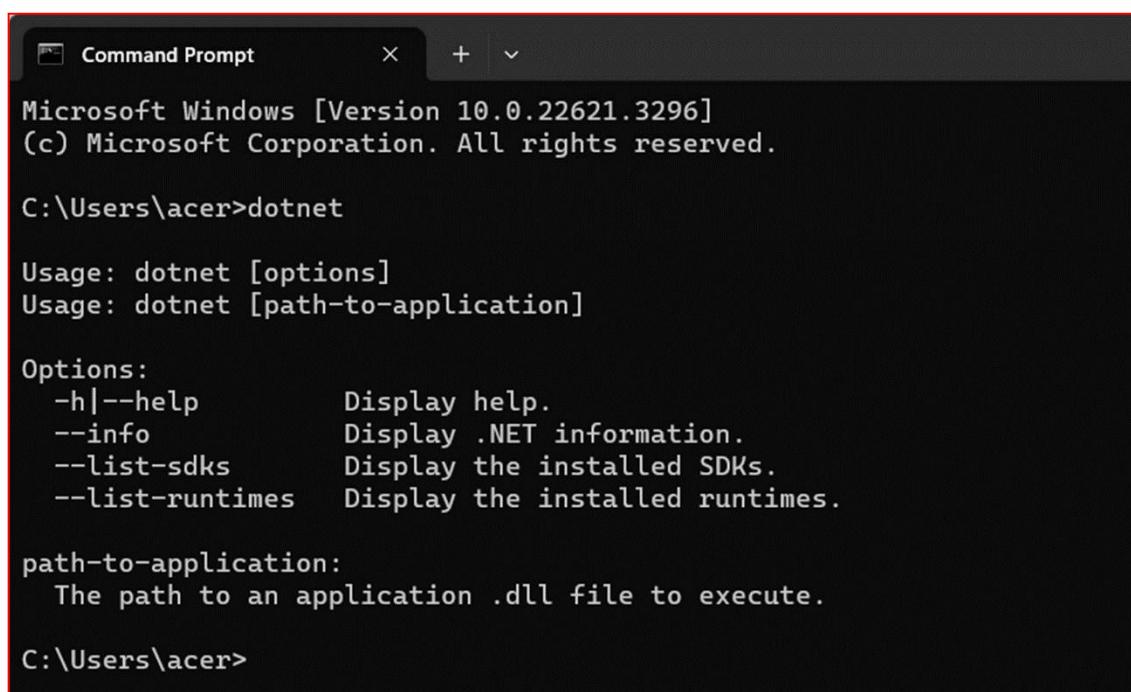
3) Check everything installed correctly Once you've installed, open a new command prompt and run the following command:

Commands:

(Command to check whether .net sdk have been downloaded properly)

Command 1: dotnet

Output:-



```

Command Prompt      X + ▾
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\acer>

```

Step 2: Create your web API

1) Open two command prompts (Command prompt 1 & 2)

Command prompt 1:

Command 1: dotnet new webapi -o Glossary

Command 2: cd Glossary

Command 3: dotnet run

Output:-

```
Command Prompt - dotnet n x + v
C:\Users\acer>dotnet new webapi -o Glossary
Welcome to .NET 8.0!
-----
SDK Version: 8.0.103
Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate, run 'dotnet dev-certs https --trust'
Learn about HTTPS: https://aka.ms/dotnet-https

-----
Write your first app: https://aka.ms/dotnet-hello-world
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli
-----
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring C:\Users\acer\Glossary\Glossary.csproj:
```

```
C:\Users\acer>cd Glossary
C:\Users\acer\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\acer\Glossary
|
```

Command prompt 2:

Command 1: curl

Command 2: curl --insecure http://localhost:5001/weatherforecast

Note: Update the localhost port according to your dotnet run output.**Output:-**

```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

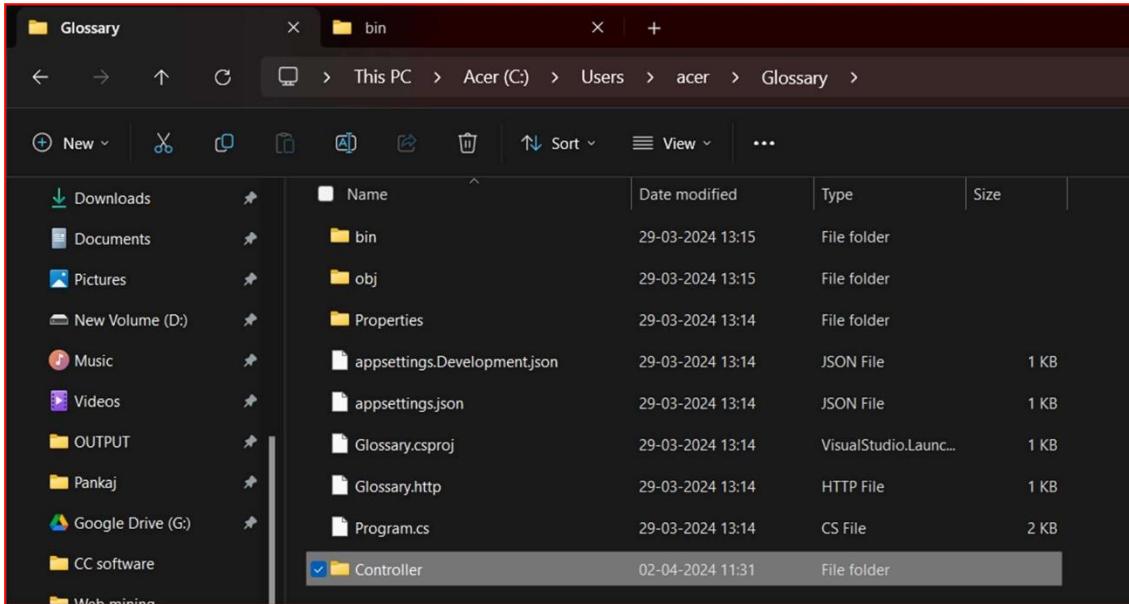
C:\Users\acer>curl
curl: try 'curl --help' for more information

C:\Users\acer>curl --insecure http://localhost:5118/weatherforecast
curl: (7) Failed to connect to localhost port 5118 after 2257 ms: Couldn't connect to server

C:\Users\acer>curl --insecure http://localhost:5001/weatherforecast
[{"date": "2024-04-03", "temperatureC": 10, "summary": "Warm", "temperatureF": 49}, {"date": "2024-04-04", "temperatureC": 6, "summary": "Mild", "temperatureF": 42}, {"date": "2024-04-05", "temperatureC": 31, "summary": "Warm", "temperatureF": 87}, {"date": "2024-04-06", "temperatureC": -12, "summary": "Mild", "temperatureF": 11}, {"date": "2024-04-07", "temperatureC": -11, "summary": "Warm", "temperatureF": 13}]
```

2) Now Change the content:

To get started, create a **Controller** Folder in Glossary Folder



3) Open notepad and paste the below code in it and save the file as “**GlossaryController.cs**” inside Controller folder.

Code:-

```
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.IO;
namespace Glossary.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class GlossaryController: ControllerBase
    {
        private static List <GlossaryItem> Glossary = new List<GlossaryItem>
        {
            new GlossaryItem
            {
                Term= "HTML",
                Definition = "Hypertext Markup Language"
            },
        }
    }
}
```

```
new GlossaryItem
{
    Term= "MVC",
    Definition = "Model View Controller"
},
new GlossaryItem
{
    Term= "OpenID",
    Definition = "An open standard for authentication"
}
};

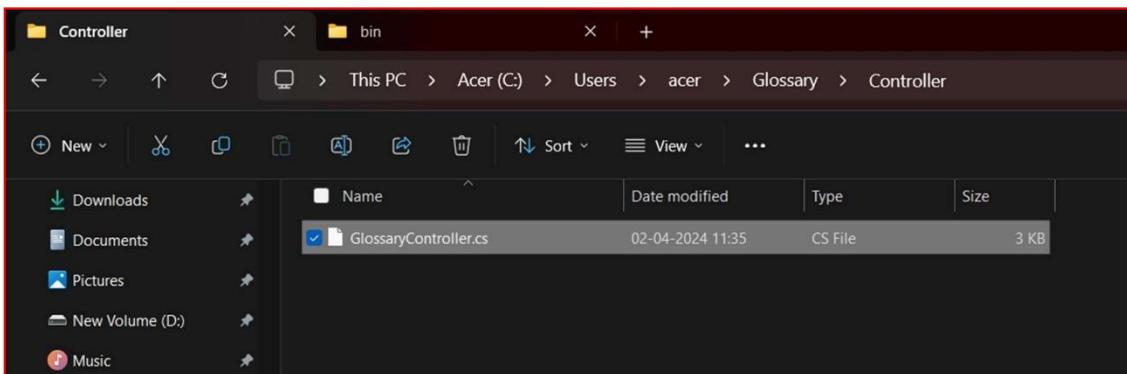
[HttpGet]
public ActionResult<List<GlossaryItem>> Get()
{
    return Ok(Glossary);
}

[HttpGet]
[Route("{term}")]
public ActionResult<GlossaryItem> Get(string term)
{
    var glossaryItem = Glossary.Find(item => item.Term.Equals(term,
        StringComparison.InvariantCultureIgnoreCase));
    if (glossaryItem == null)
    {
        return NotFound();
    } else
    {
        return Ok(glossaryItem);
    }
}

[HttpPost]
public ActionResult Post(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
```

```
item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));  
if (existingGlossaryItem != null)  
{  
    return Conflict("Cannot create the term because it already exists.");  
}  
else  
{ Glossary.Add(glossaryItem);  
var resourceUrl = Path.Combine(Request.Path.ToString(),  
Uri.EscapeUriString(glossaryItem.Term));  
return Created(resourceUrl, glossaryItem);  
}  
}  
}  
[HttpPut]  
public ActionResult Put(GlossaryItem glossaryItem)  
{  
    var existingGlossaryItem = Glossary.Find(item =>  
item.Term.Equals(glossaryItem.Term,  
StringComparison.InvariantCultureIgnoreCase));  
if (existingGlossaryItem == null)  
{  
    return BadRequest("Cannot update a nont existing term.");  
}  
else  
{existingGlossaryItem.Definition = glossaryItem.Definition; return Ok();  
}  
}  
}  
[HttpDelete]  
[Route("{term}")]  
public ActionResult Delete(string term)  
{  
    var glossaryItem = Glossary.Find(item => item.Term.Equals(term,  
StringComparison.InvariantCultureIgnoreCase));  
if (glossaryItem == null)
```

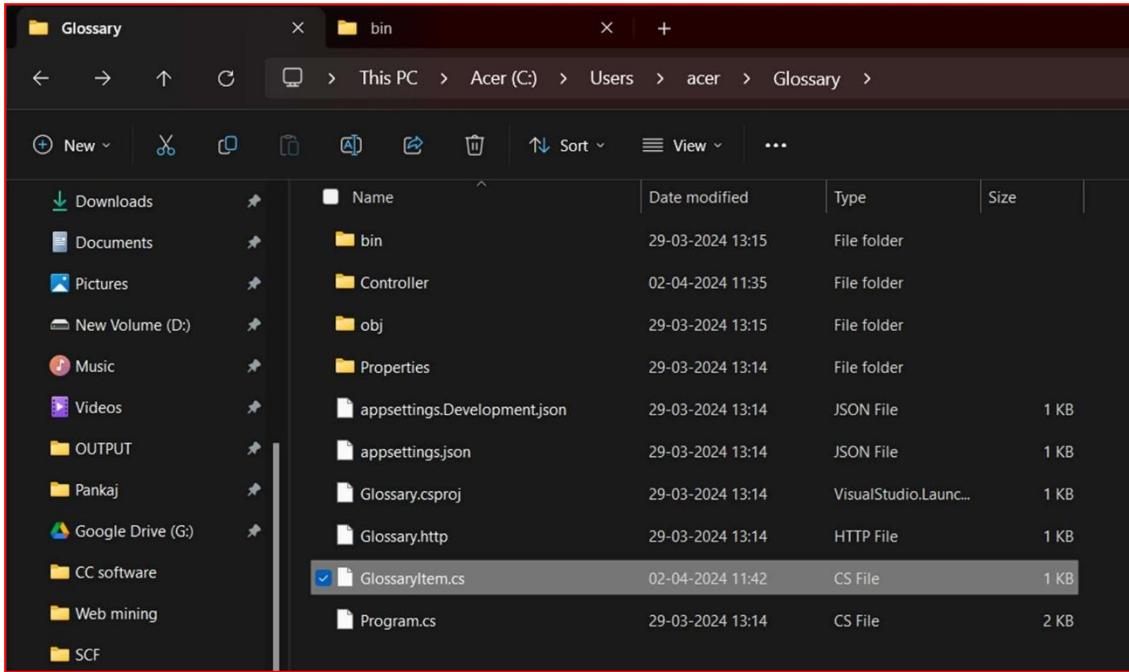
```
{ return NotFound();  
}  
else  
{ Glossary.Remove(glossaryItem);  
return NoContent();  
}  
}  
}  
}  
}
```



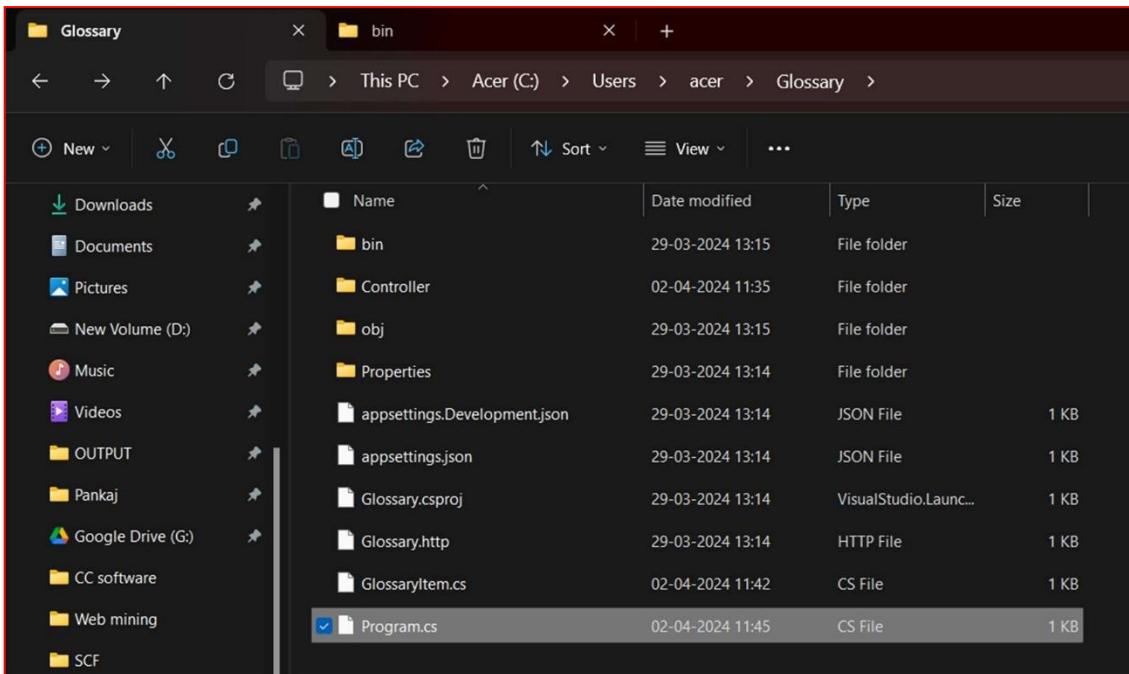
4) In Glossary Folder create a file “**GlossaryItem.cs**” and paste the below code.

Code:-

```
namespace Glossary  
{  
public class GlossaryItem  
{  
public string Term {get;set;}  
public string Definition{get;set;}  
}  
}
```



5) Now open “**Program.cs**” file and update the code by pasting the below code.



Code:-

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllers();

// Learn more about configuring Swagger/OpenAPI at
// https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

- 6) Go to **command prompt 1** and stop running previous dotnet run code using Ctrl+C and run it again.

Command: dotnet run

Output:-

```
Command Prompt - dotnet run + v
Now listening on: http://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\acer\Glossary
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
Failed to determine the https port for redirect.
info: Microsoft.Hosting.Lifetime[0]
Application is shutting down...

C:\Users\acer\Glossary>dotnet run
Building...
C:\Users\acer\Glossary\GlossaryItem.cs(5,15): warning CS8618: Non-nullable property 'Term' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\acer\Glossary\Glossary.csproj]
C:\Users\acer\Glossary\GlossaryItem.cs(6,15): warning CS8618: Non-nullable property 'Definition' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Users\acer\Glossary\Glossary.csproj]
C:\Users\acer\Glossary\Controller\GlossaryController.cs(57,57): warning SYSLIB0013: 'Uri.EscapeUriString(string)' is obsolete: 'Uri.EscapeUriString can corrupt the Uri string in some cases. Consider using Uri.EscapeDataString for query string components instead.' (https://aka.ms/dotnet-warnings/SYSLIB0013) [C:\Users\acer\Glossary\Glossary.csproj]
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\acer\Glossary
```

7) Now extract the **curl** in downloads folder or wherever it is downloaded.

Open Command Prompt 2 and execute the following:**1) Getting a list of Item**

Command: curl --insecure http://localhost:5001/api/Glossary

Output:-

```
C:\Users\acer>curl --insecure http://localhost:5001/api/Glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}]
C:\Users\acer>
```

2) Getting a single Item

Command: curl --insecure http://localhost:5001/api/Glossary/MVC

Output:-

```
C:\Users\acer>curl --insecure http://localhost:5001/api/Glossary/MVC
{"term": "MVC", "definition": "Model View Controller"}
C:\Users\acer>
```

3) Creating an Item**Command:**

curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" http://localhost:5001/api/Glossary

Output:-

```
C:\Users\acer>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" http://localhost:5001/api/Glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\acer>
```

4) Update Item

Command:

Command1: curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\":\"Modified record of Model View Controller.\"}" -H "Content-Type:application/json"
 http://localhost:5001/api/Glossary

Command 2: curl --insecure http://localhost:5001/api/Glossary

Output:-

```
C:\Users\acer>curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\":\"Modified record of Model View Controller.\"}" -H "Content-Type:application/json"
http://localhost:5001/api/Glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Modified record of Model View Controller."}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\acer>
```

5) Delete Item

Command:

Command 1: curl --insecure --request DELETE -url http://localhost:5001/api/Glossary

Command 2: curl --insecure http://localhost:5001/api/Glossary

Output:-

```
C:\Users\acer>curl --insecure --request DELETE -url http://localhost:5001/api/Glossary
curl: (6) Could not resolve host: xn--url-1n8a
C:\Users\acer>curl --insecure http://localhost:5001/api/Glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Modified record of Model View Controller."}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\acer>
```

Practical 2B

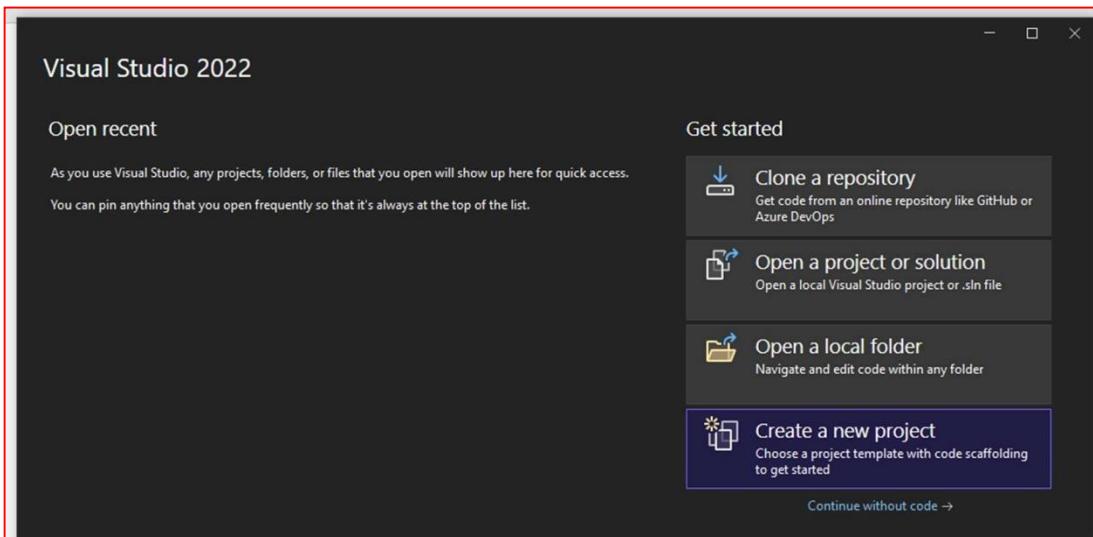
AIM:- BUILDING ASP .NET CORE REST API USING VISUAL STUDIO

Step 1:

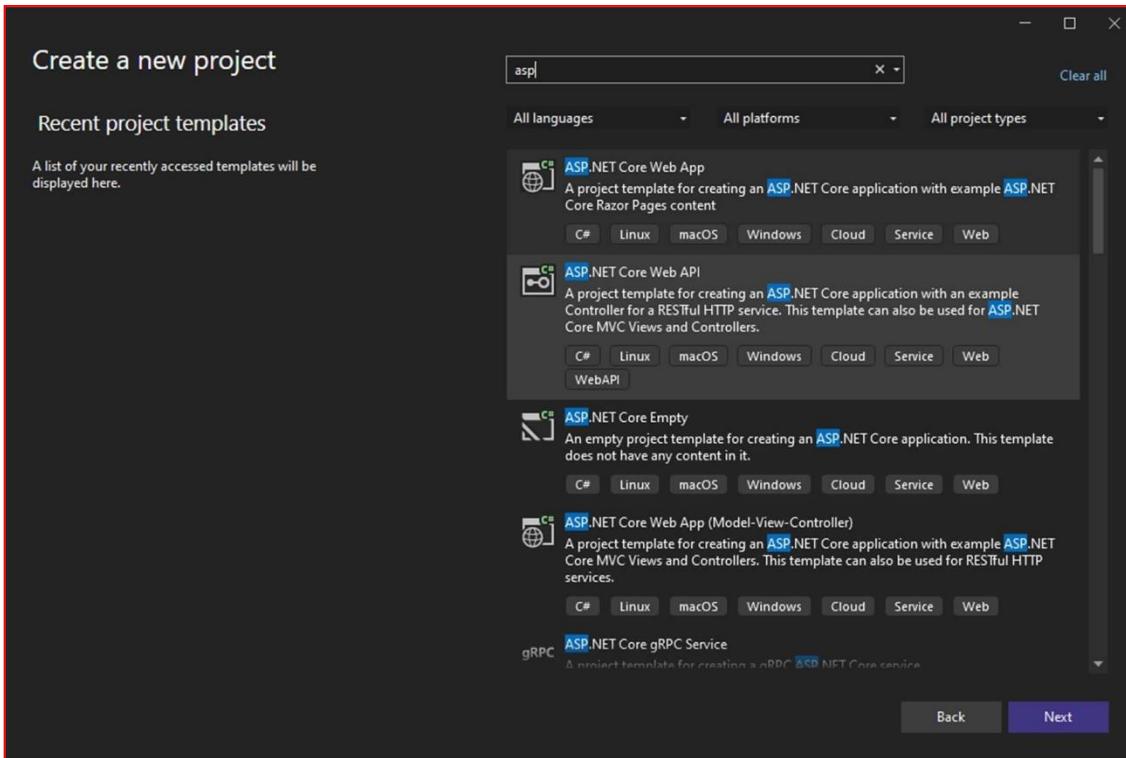
- 1) Download and install Microsoft visual studio 2022 and open it



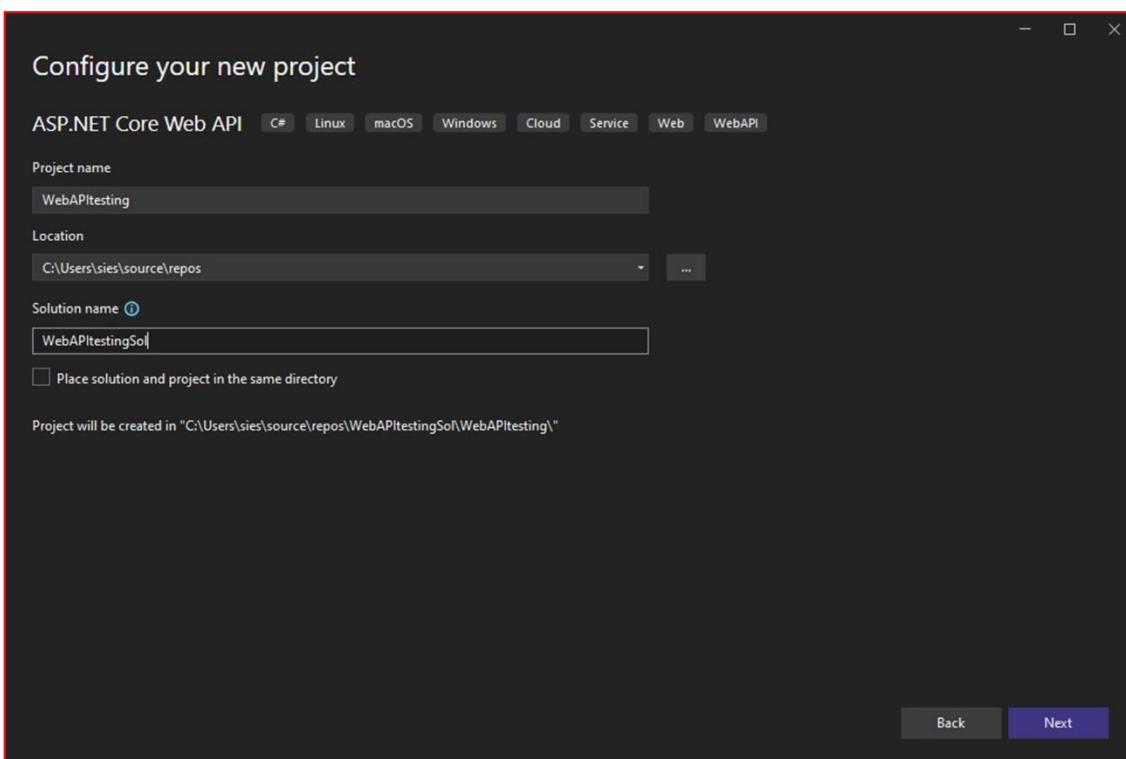
Create a new project:-



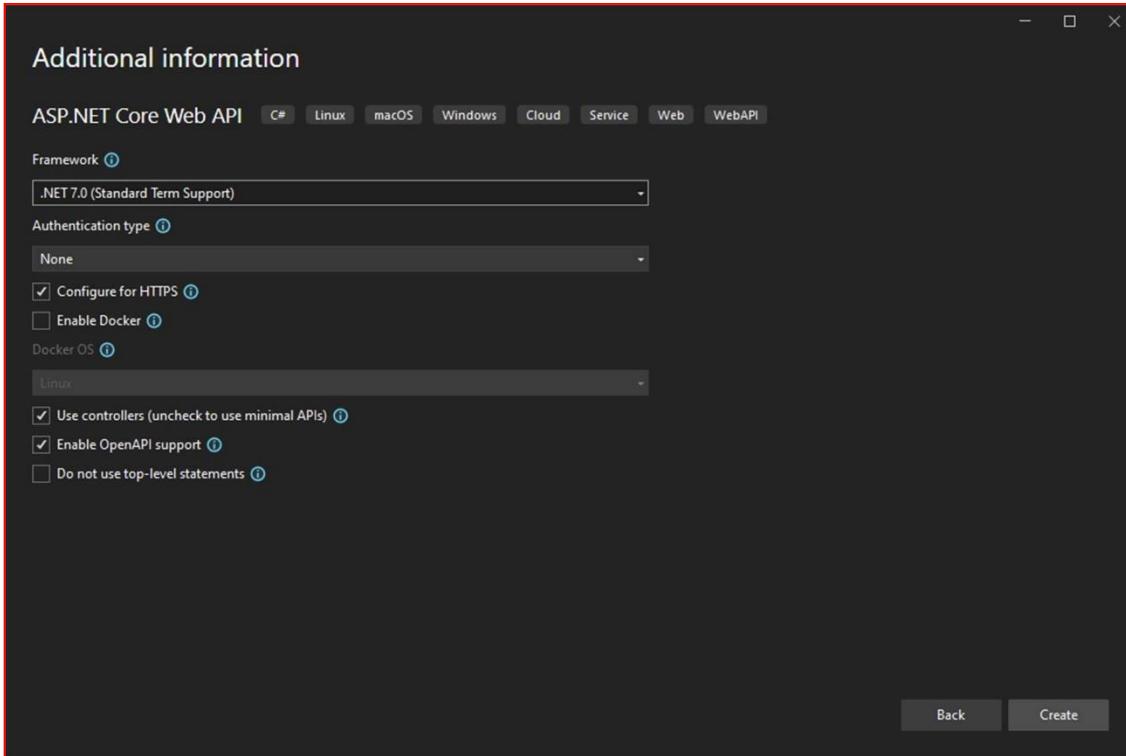
Select the project type (ASP.NET Core Web Api)



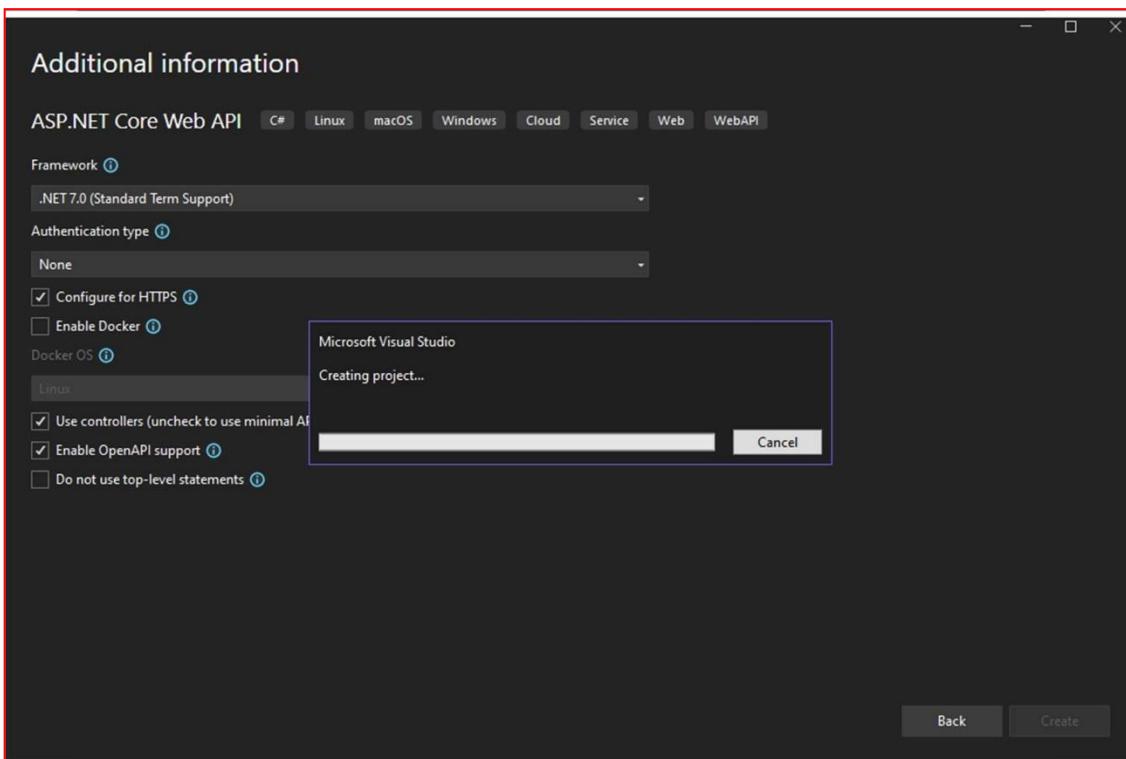
Project name will be WebAPITesting and Solution name will be WebAPITestingSol



Select the Target Framework. (.NET 7.0) and click on create



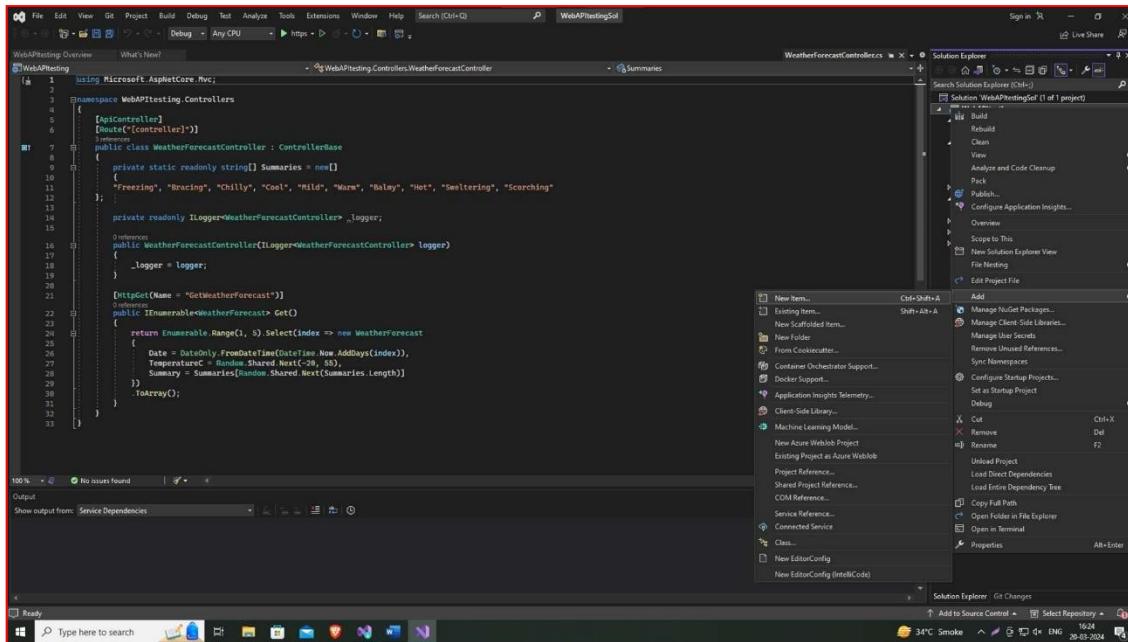
Creating project.....



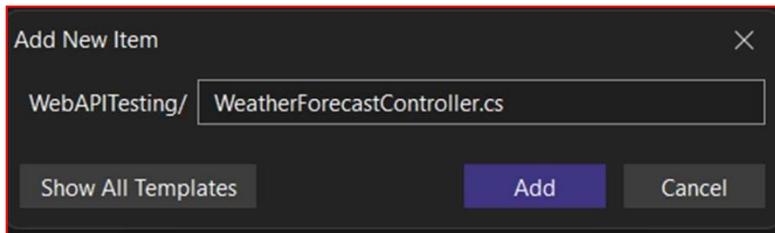
Step 2:-

This will create the standard API project with the Weather Forecast controller. Next, add a new http file as below:

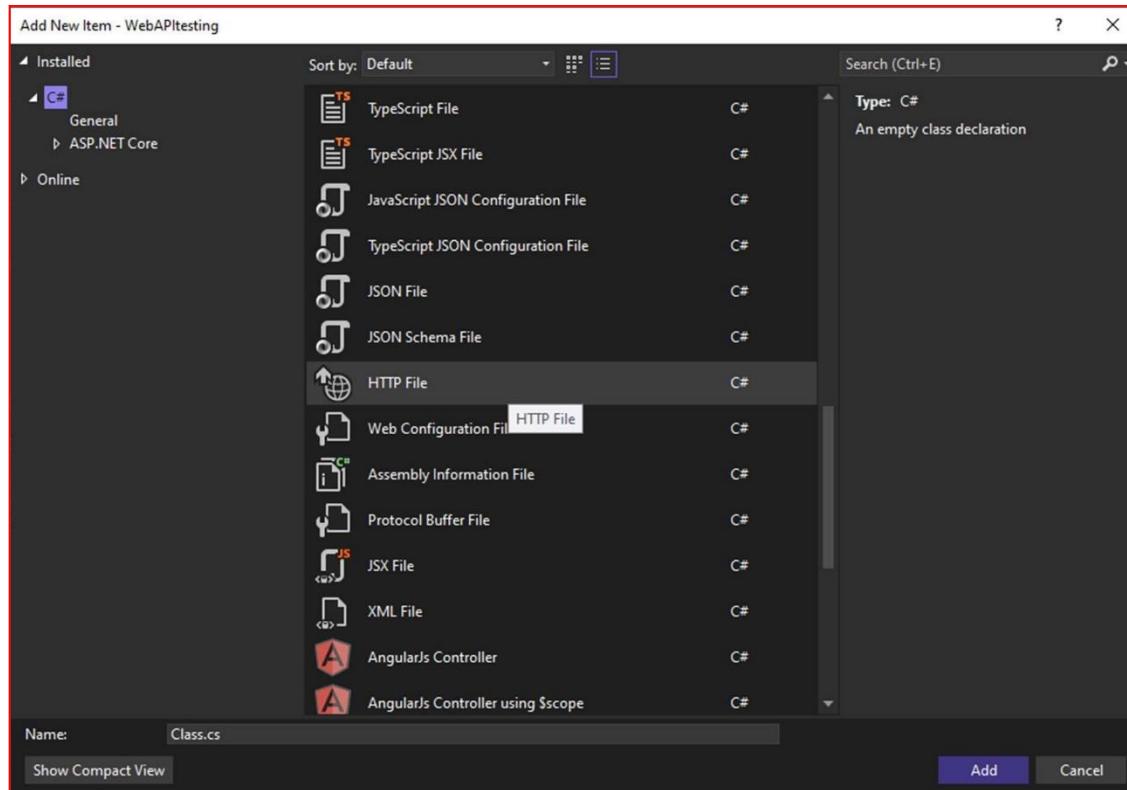
Right Click on WebAPITesting >> Add >> New Item



Give name “**WeatherForecastController.cs**” Show all templates and select “**HTTP File**”



Select HTTP File and click on add.



Go to Controller and update “**WeatherForecastController.cs**” file using below code

```

using Microsoft.AspNetCore.Mvc;
namespace WebAPITesting.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Mild", "Warm", "Hot", "Sweltering", "Scorching"
        };

        private readonly ILogger<WeatherForecastController> _logger;

        public WeatherForecastController	ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }

        [HttpGet(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
            var rng = Random.Shared.Next(20, 85);
            return Enumerable.Range(1, 6).Select(index => new WeatherForecast
            {
                Date = DateTimeOffset.UtcNow.AddHours(index),
                TemperatureC = Random.Shared.Next(-20, 85),
                Summary = Summaries[Random.Shared.Next(Summaries.Length)]
            });
        }
    }
}

```

Code:-

```

using Microsoft.AspNetCore.Mvc;

namespace WebAPITesting.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot",
            "Sweltering", "Scorching"
        };

        private readonly ILogger<WeatherForecastController> _logger;

        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }

        [HttpGet(Name = "GetWeatherForecast")]
        public IEnumerable<WeatherForecast> Get()
        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
                TemperatureC = Random.Shared.Next(-20, 55),
                Summary = Summaries[Random.Shared.Next(Summaries.Length)]
            })
            .ToArray();
        }

        [HttpPost("{id}", Name = "GetLocationDetails")]

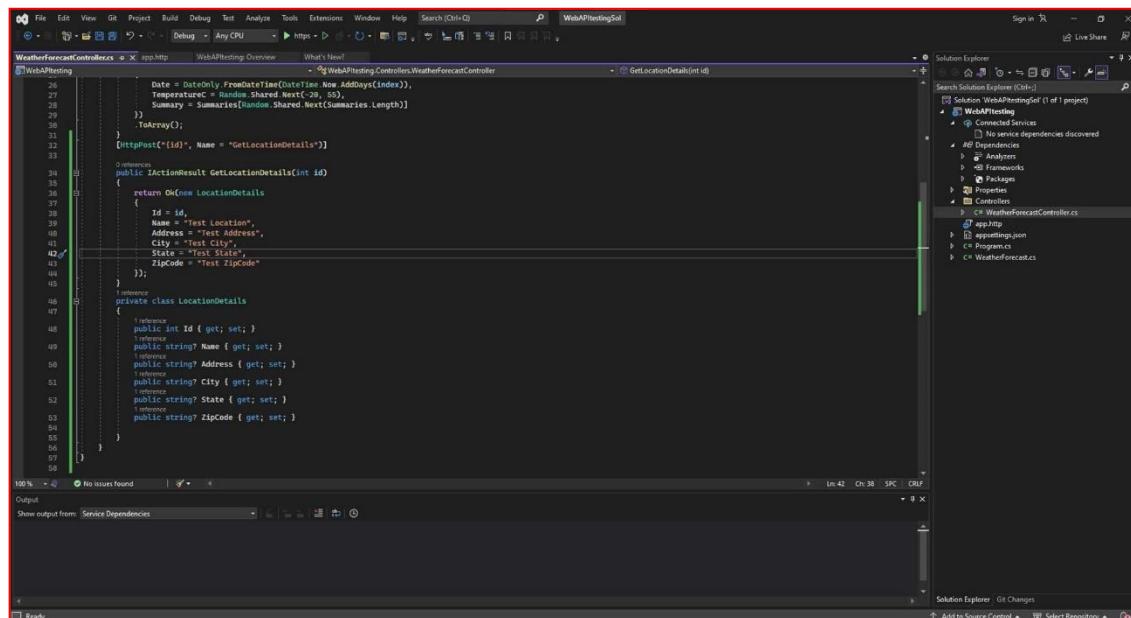
        public IActionResult GetLocationDetails(int id)
        {
            return Ok(new LocationDetails
            {
                Id = id,
                Name = "Test Location",
                Address = "Test Address",
                City = "Test City",
                State = "Test State",
                ZipCode = "Test ZipCode"
            });
        }

        private class LocationDetails
        {
            public int Id { get; set; }
            public string? Name { get; set; }
            public string? Address { get; set; }
            public string? City { get; set; }
        }
    }
}

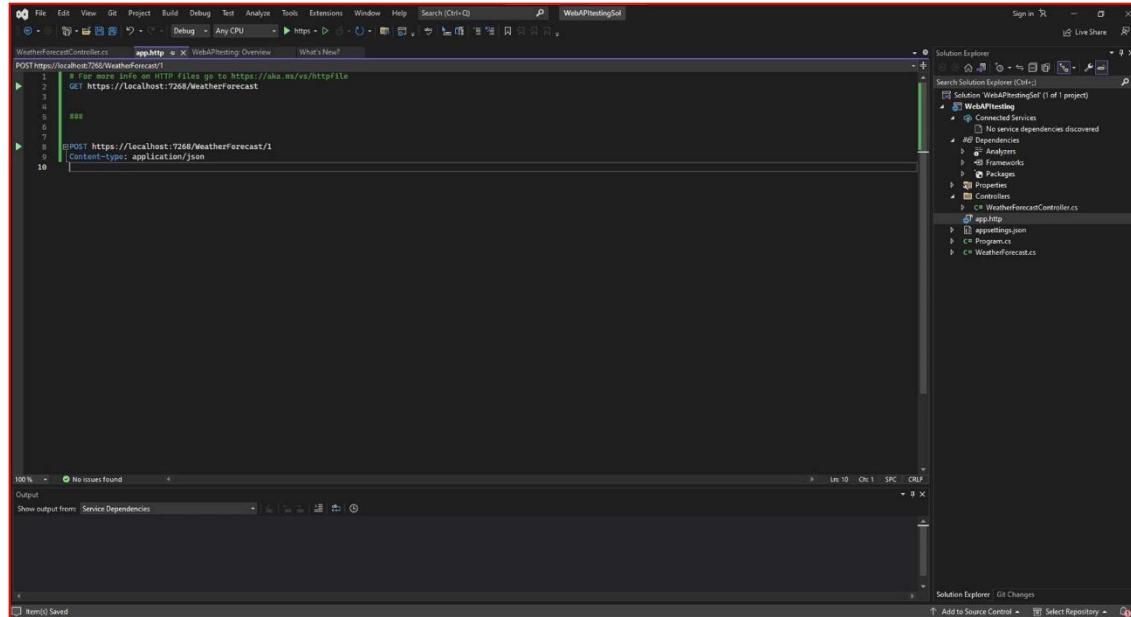
```

```
public string? State { get; set; }
public string? ZipCode { get; set; }
```

```
}
```



Finally, update the code in the “app.http” file as below



Code:-

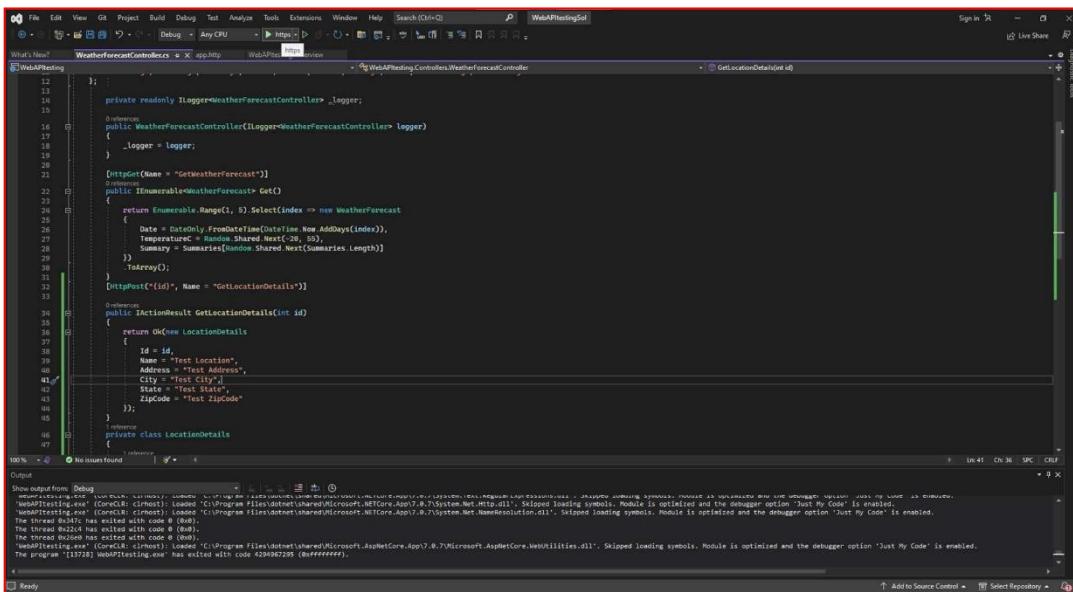
```
# For more info on HTTP files go to https://aka.ms/vs/httpfile
GET https://localhost:7268/WeatherForecast
```

```
###
```

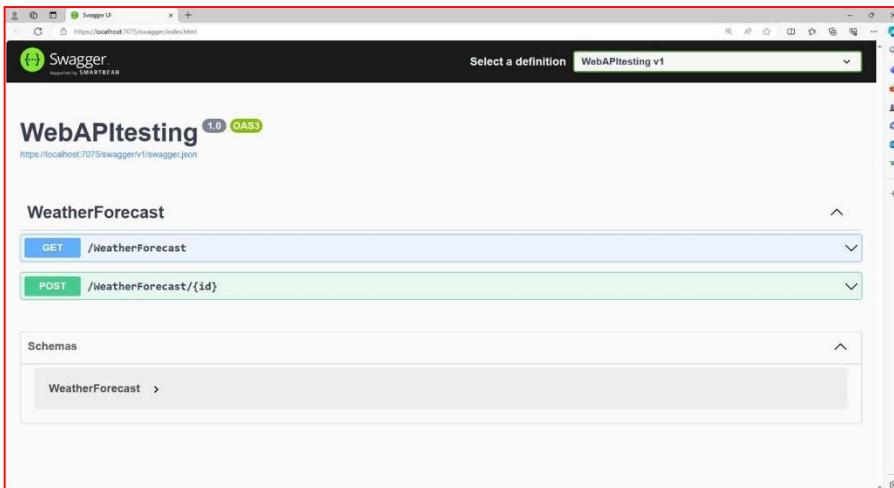
POST https://localhost:7268/WeatherForecast/1
Content-type: application/json

Step 3:-

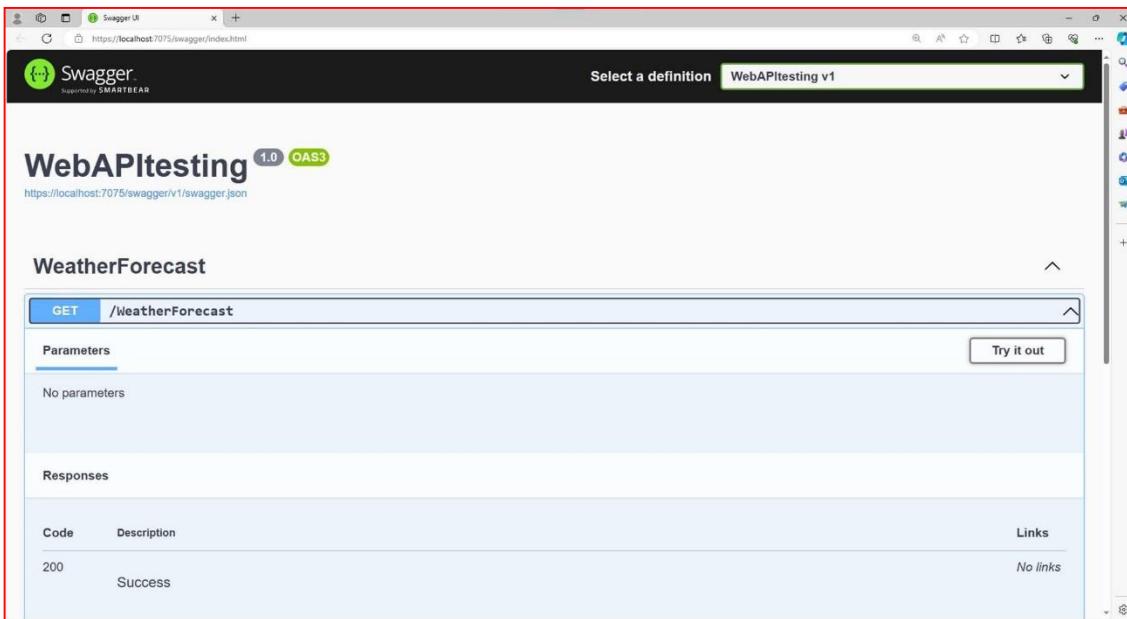
Run the code using the https button



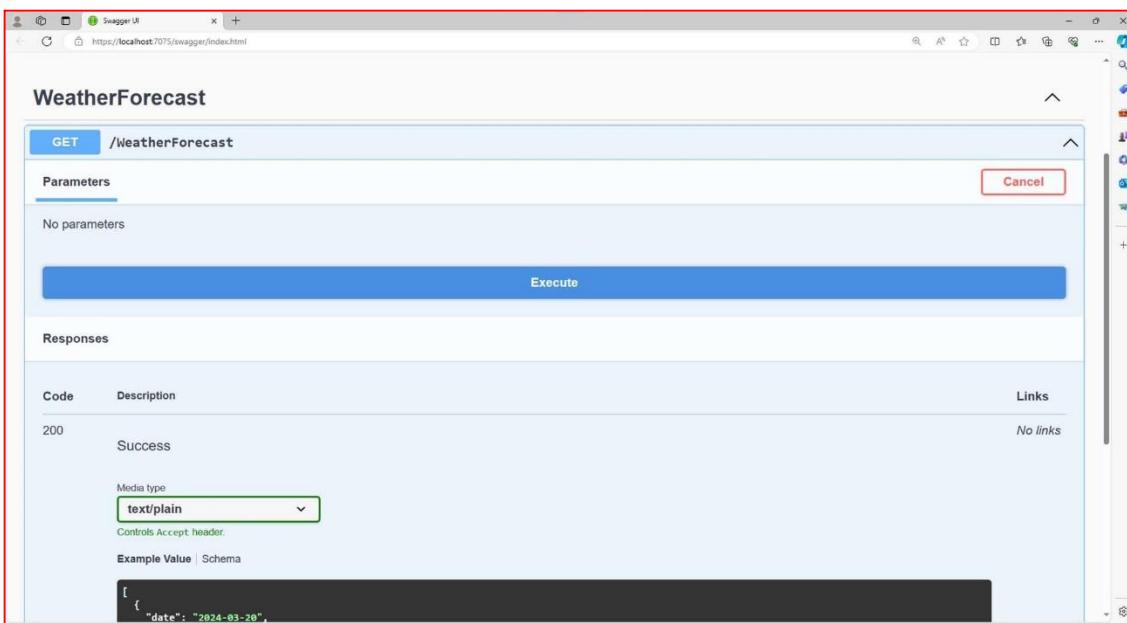
The output will be seen in browser



Click on GET >> Try it out >> Execute

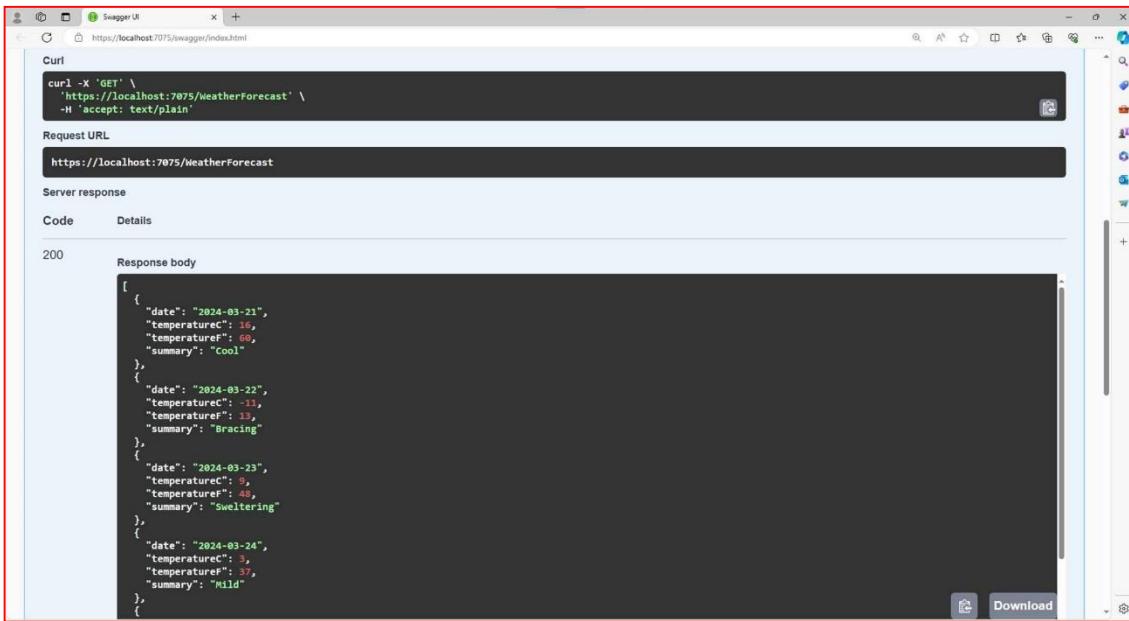


The screenshot shows the Swagger UI interface for the 'WebAPITesting' API. At the top, there's a navigation bar with the Swagger logo and the text 'Select a definition' followed by 'WebAPITesting v1'. Below this, the main title 'WebAPITesting' is displayed with a version '1.0' and a 'CORS' badge. The URL 'https://localhost:7075/swagger/v1/swagger.json' is shown below the title. The main content area is titled 'WeatherForecast'. It features a 'GET /WeatherForecast' button, a 'Parameters' section (which is currently empty), and a 'Responses' section. The 'Responses' section includes a table with two rows: one for code 200 with description 'Success' and another row for 'Links' which says 'No links'. A 'Try it out' button is located in the top right corner of the main content area.



This screenshot is similar to the previous one, showing the 'WeatherForecast' endpoint in the Swagger UI. However, the 'Execute' button has been clicked, as indicated by the red border around the 'Execute' button in the screenshot. The 'Responses' section now includes a dropdown menu for 'Media type' set to 'text/plain', with a note below stating 'Controls Accept header.' and 'Example Value | Schema'. An example value is shown as a JSON array: '[{"date": "2024-03-26"}]'.

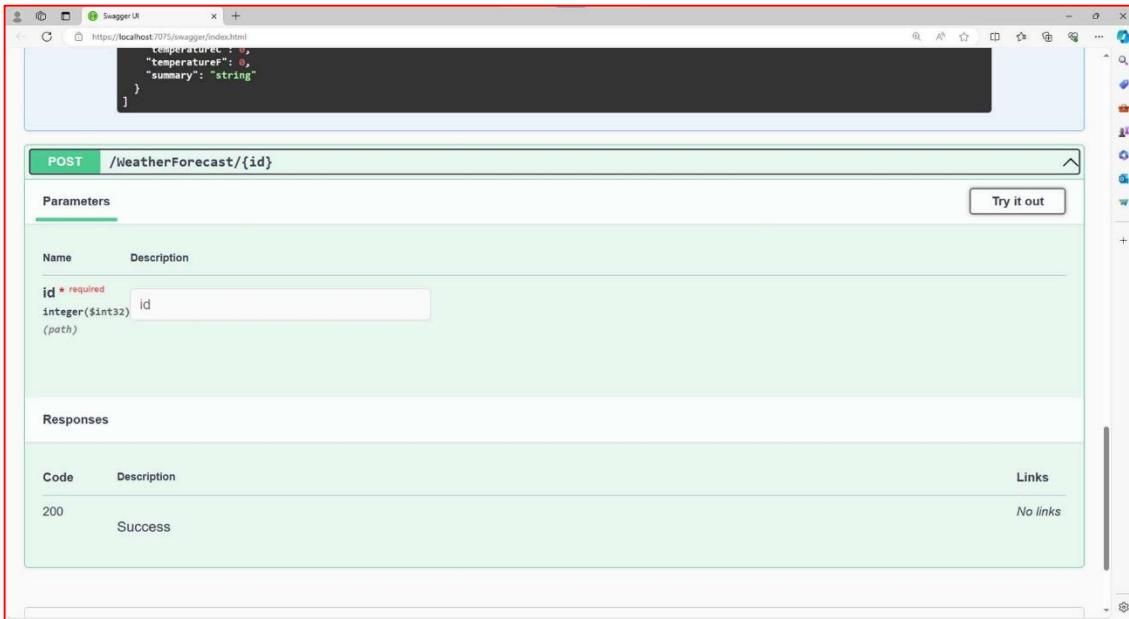
Output:- GET



The screenshot shows the Swagger UI interface for a GET request to the endpoint `/WeatherForecast`. The response body is a JSON array containing four weather forecast entries:

```
[{"date": "2024-03-21", "temperatureC": 16, "temperatureF": 60, "summary": "Cool"}, {"date": "2024-03-22", "temperatureC": 11, "temperatureF": 50, "summary": "Bracing"}, {"date": "2024-03-23", "temperatureC": 9, "temperatureF": 48, "summary": "Sweltering"}, {"date": "2024-03-24", "temperatureC": 3, "temperatureF": 30, "summary": "Mild"}]
```

Click on POST >> Try it out >> Enter id as 1 >> Execute



The screenshot shows the Swagger UI interface for a POST request to the endpoint `/WeatherForecast/{id}`. The parameters section shows a required parameter `id` of type integer(\$int32) with a value of 1 entered. The responses section shows a 200 status code with a description of "Success".

The screenshot shows the Swagger UI interface for a POST request to the endpoint `/WeatherForecast/{id}`. The request body is defined as:

```
temperature: 0,  
temperatureF: 0,  
summary: "string"  
]
```

The parameters section shows a required parameter `id` with a value of `1`. The responses section shows a successful response (200) with the description `Success`.

Output:- POST

The screenshot shows the results of a successful POST request. The curl command used is:

```
curl -X 'POST' \  
      'https://localhost:7075/WeatherForecast/1' \  
      -H 'accept: */*' \  
      -d ''
```

The Request URL is `https://localhost:7075/WeatherForecast/1`. The Server response includes a code of 200 and a response body:

```
{  
    "id": 1,  
    "name": "Test Location",  
    "address": "Test Address",  
    "city": "Test City",  
    "state": "Test State",  
    "zipcode": "Test Zipcode"  
}
```

The Response headers include:

```
content-type: application/json; charset=utf-8  
date: Wed, 20 Mar 2024 11:17:41 GMT  
server: Kestrel
```

The Responses section shows a successful response (200) with the description `Success`.

The screenshot shows a browser window with the title "Swagger UI". The URL in the address bar is <https://localhost:7075/swagger/index.html>. The main content area displays an API endpoint for a weather forecast. The endpoint path is `/weatherForecast/{id}`, where `{id}` is a required integer parameter with a value of 1. Below the path, there are "Execute" and "Clear" buttons. The "Responses" section shows a curl command and a request URL (`https://localhost:7075/weatherForecast/1`). For a 200 response, the "Response body" is displayed as a JSON object:

```
{
  "id": 1,
  "name": "Tokaiwell",
  "address": "Tokaiwell",
  "city": "Tokaiwell",
  "state": "Tokaiwell",
  "zipCode": "7000231"
}
```

The "Response headers" section shows the following values:

```
content-type: application/json; charset=utf-8
date: Wed, 20 Mar 2024 11:15:43 GMT
server: Kestrel
```

The "Responses" section also includes tabs for "Code" (selected) and "Details". At the bottom right, there is a "Links" section with the message "No links".

Practical 3

AIM:- BUILDING .NET CORE MVC (Hello World & StockQuote)

Step 1: Install .Net Core Sdk from official website

(Link :- <https://dotnet.microsoft.com/en-us/download/dotnet/thank you/sdk-8.0.101-windows-x64-installer> or <https://dotnet.microsoft.com/en-us/download>)

(For checking if .Net Core is installed in your system or not, open cmd prompt and type dotnet and enter)

```
Microsoft Windows [Version 10.0.22621.3085]
(c) Microsoft Corporation. All rights reserved.

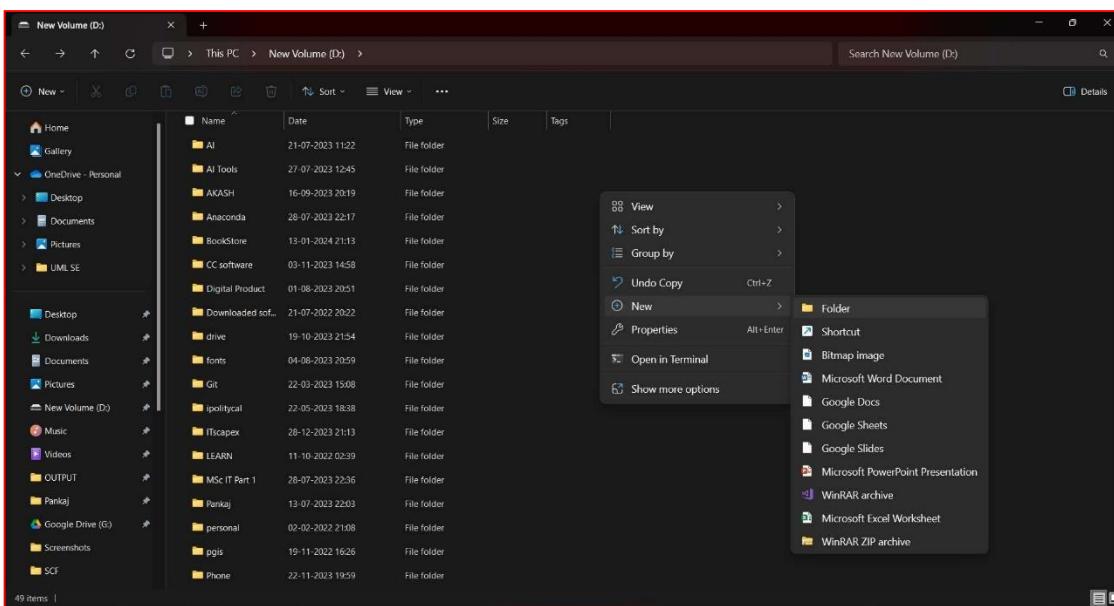
C:\Users\acer>dotnet
Usage: dotnet [options]
Usage: dotnet [path-to-application]

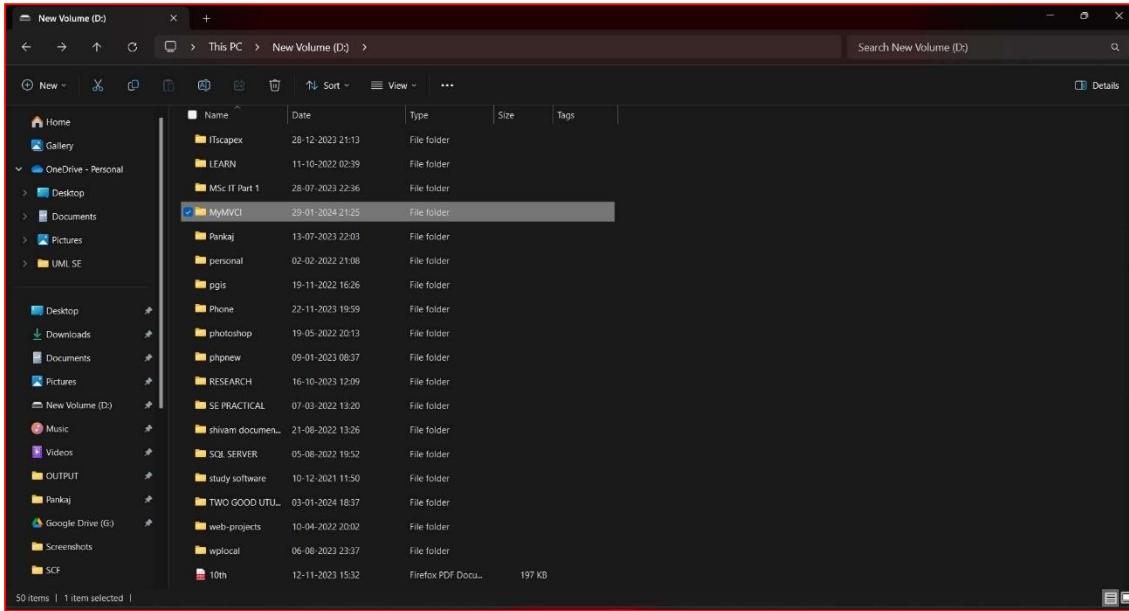
Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\acer>
```

Step 2: Create a folder named MyMVC1 in D: drive directly in your file explorer.





Step 3: Open command prompt and perform following operations in order to change the directory and access your MyMVC1 folder and performing the further steps:-

Commands:

(First change the directory to your desired drive here the used directory is D: drive)

Command 1: cd MyMVC1

Command 2: dotnet new mvc -- auth none

Output (For command 1):-

```
Command Prompt
Microsoft Windows [Version 10.0.22621.3085]
(c) Microsoft Corporation. All rights reserved.

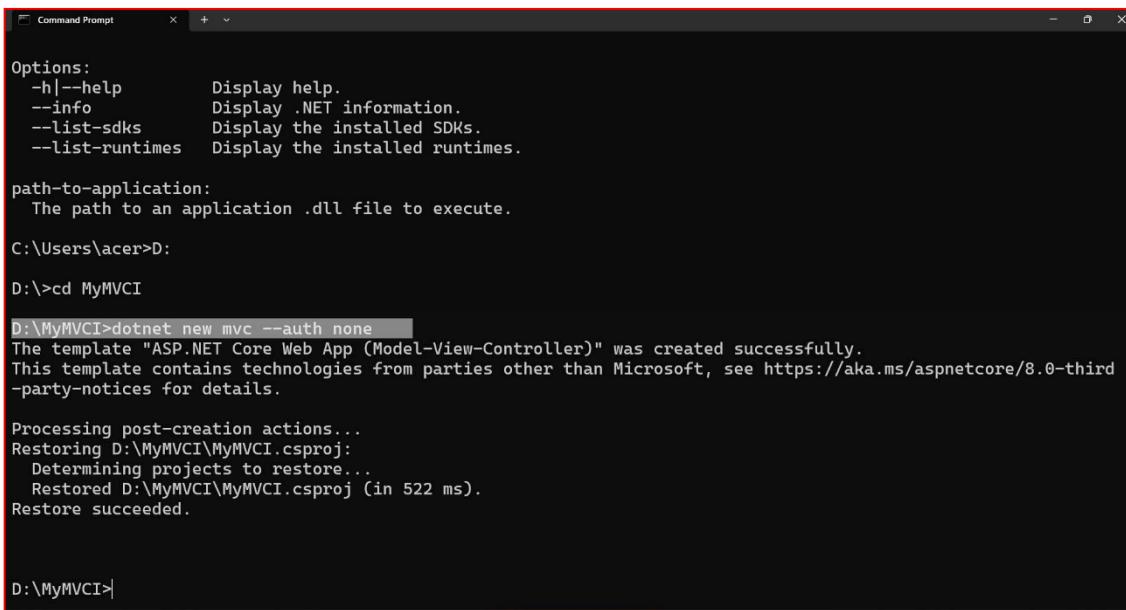
C:\Users\acer>dotnet
Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help      Display help.
  --info         Display .NET information.
  --list-sdks    Display the installed SDKs.
  --list-runtimes Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\acer>D:
D:\>cd MyMVC1
D:\MyMVC1>
```

Output (For command 2):-



```

Command Prompt

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes     Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\acer>D:

D:>cd MyMVCI

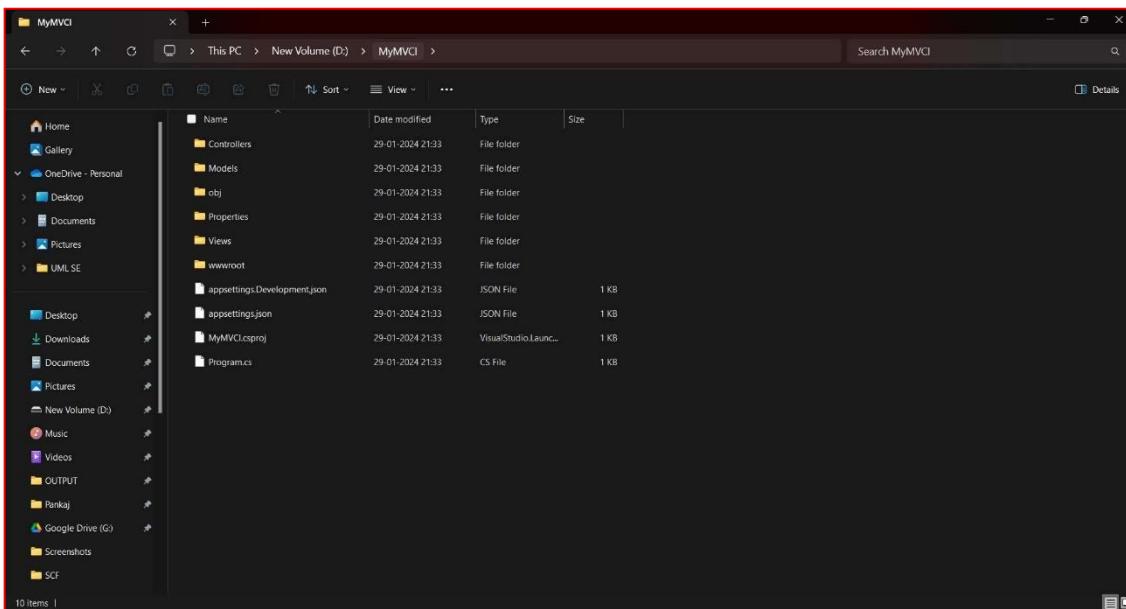
D:\MyMVCI>dotnet new mvc --auth none
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/8.0-third-
-party-notices for details.

Processing post-creation actions...
Restoring D:\MyMVCI\MyMVCI.csproj:
  Determining projects to restore...
    Restored D:\MyMVCI\MyMVCI.csproj (in 522 ms).
Restore succeeded.

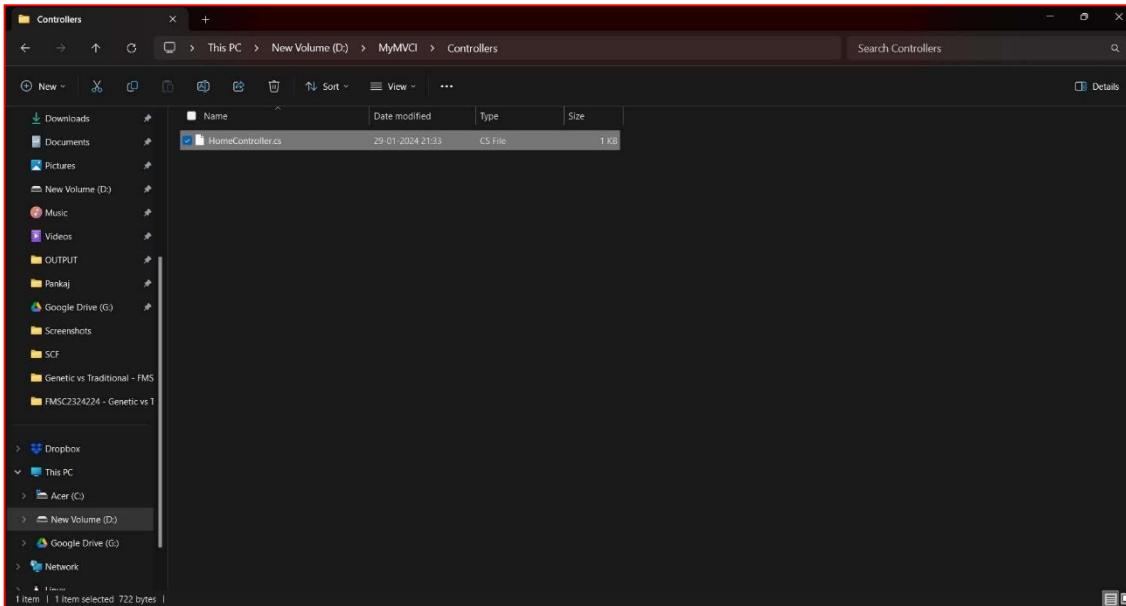
D:\MyMVCI>

```

Now check your created folder MyMVCI, it will consist of several other folders and files.



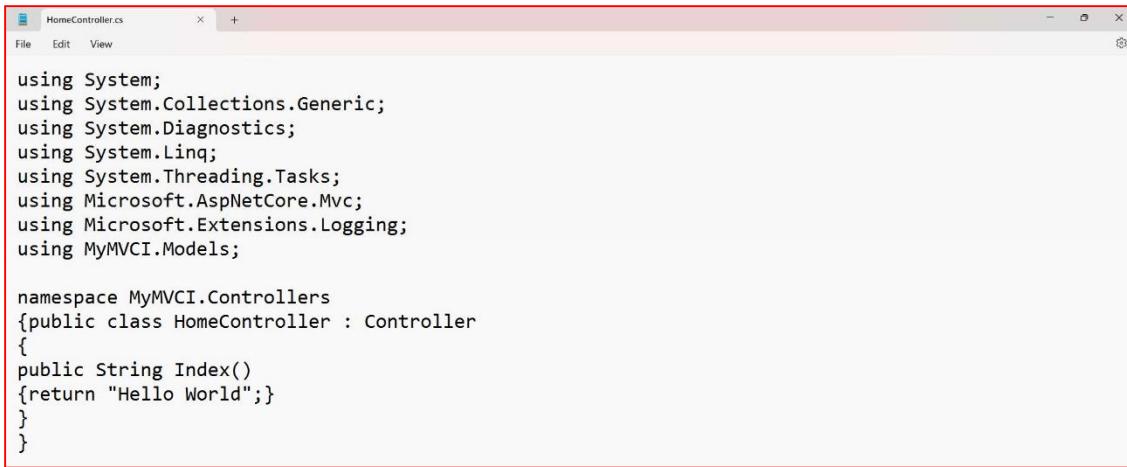
Step 4: Write the below code in the existing file named ‘**HomeController.cs**’ file in the ‘**Controllers**’ folder which is inside the MyMVC1 folder and save it.



Code:-

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC1.Models;

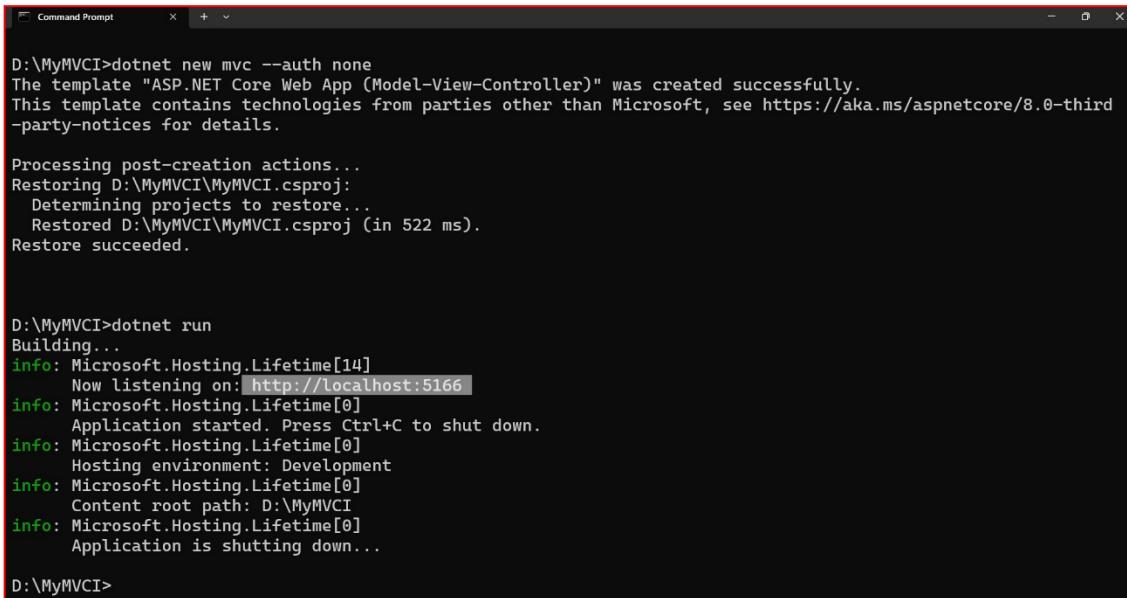
namespace MyMVC1.Controllers
{public class HomeController : Controller
{
    public String Index()
    {return "Hello World";}
}}
```



```
HomeController.cs
File Edit View
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC1.Models;

namespace MyMVC1.Controllers
{public class HomeController : Controller
{
public String Index()
{return "Hello World";}
}
}
```

Now go to the command prompt and run the **dotnet run** command from the output copy the url i.e. for e.g. here the url is **http://localhost:5166**



```
D:\MyMVC1>dotnet new mvc --auth none
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/8.0-third-party-notices for details.

Processing post-creation actions...
Restoring D:\MyMVC1\MyMVC1.csproj:
  Determining projects to restore...
  Restored D:\MyMVC1\MyMVC1.csproj (in 522 ms).
Restore succeeded.

D:\MyMVC1>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5166
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\MyMVC1
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...

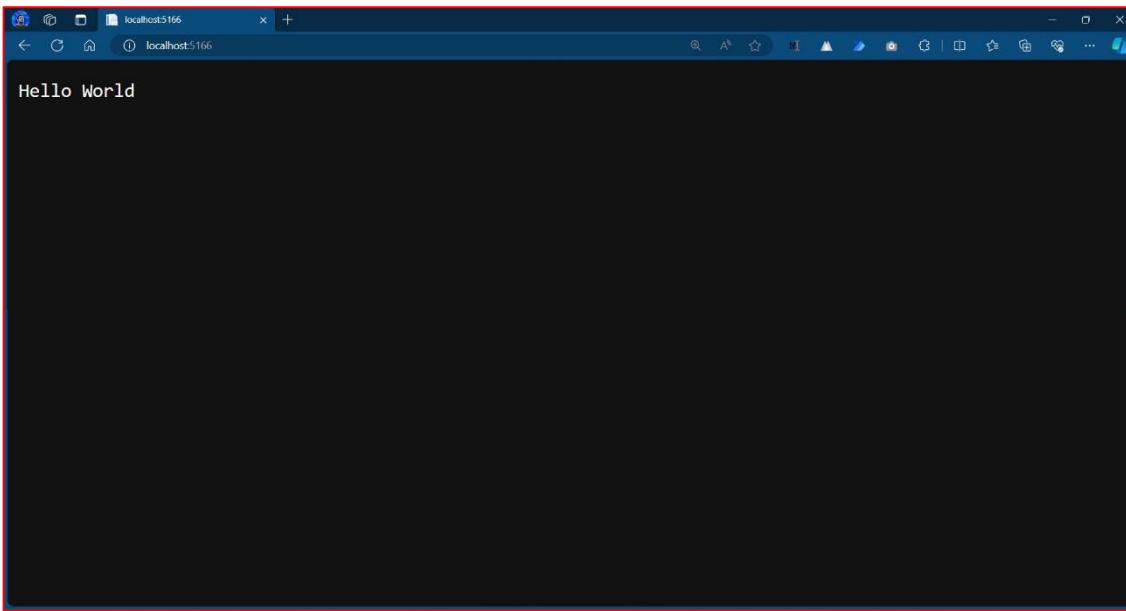
D:\MyMVC1>
```

Now paste the copied url in your browser and press enter



Output (For step 4):-

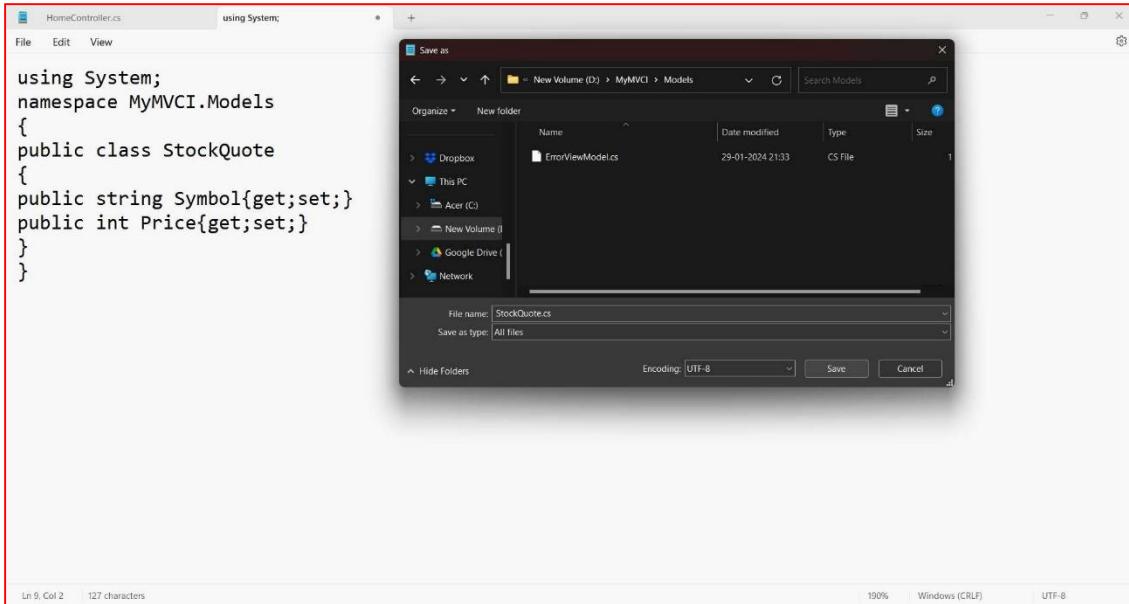
You will see the output as **Hello World**.



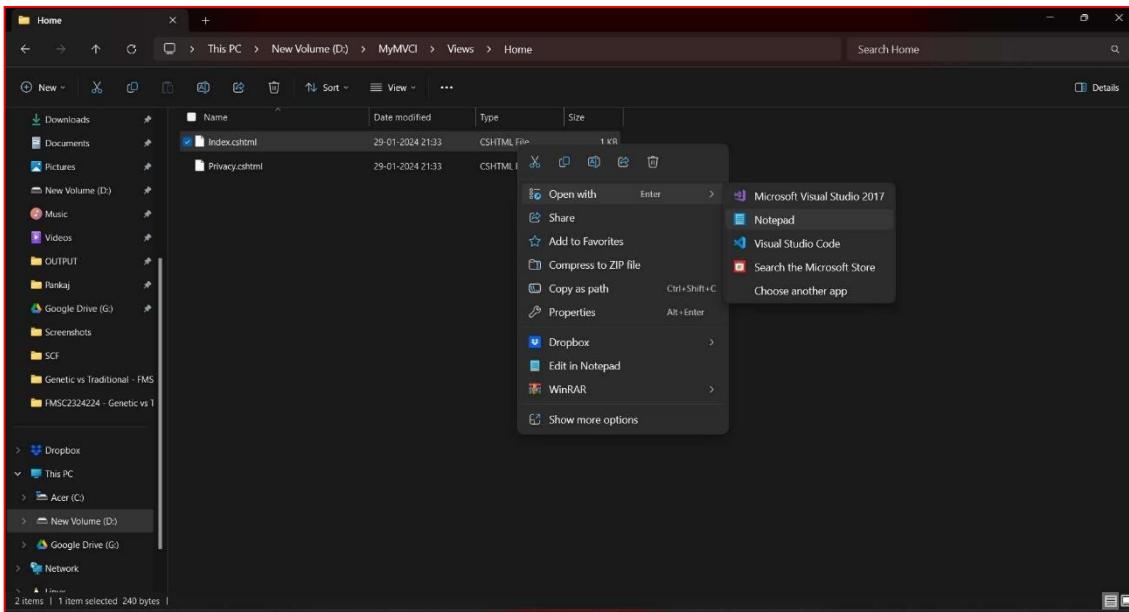
Step 5: Write the below code in a new notepad the file and save it as ‘StockQuote.cs’ in the ‘Models’ folder which is inside the MyMVCI folder.

Code:-

```
using System;
namespace MyMVCI.Models
{
    public class StockQuote
    {
        public string Symbol{get;set;}
        public int Price{get;set;}
    }
}
```



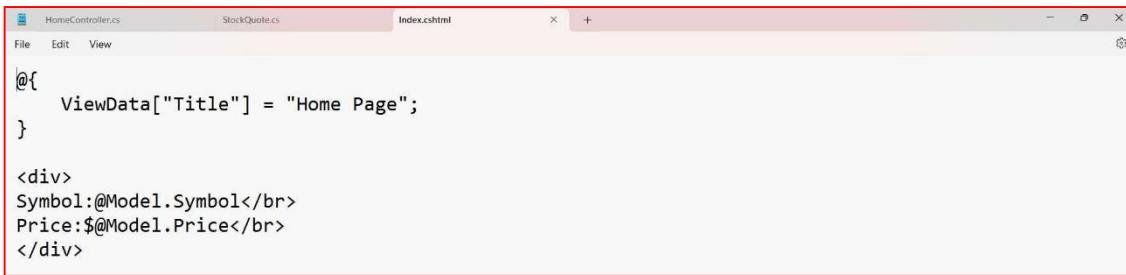
Step 6: Write the below code in the existing file named ‘index.cshtml’ file in the ‘Home’ folder which is inside the ‘Views’ folder & it’s inside the MyMVC1 folder and save it.



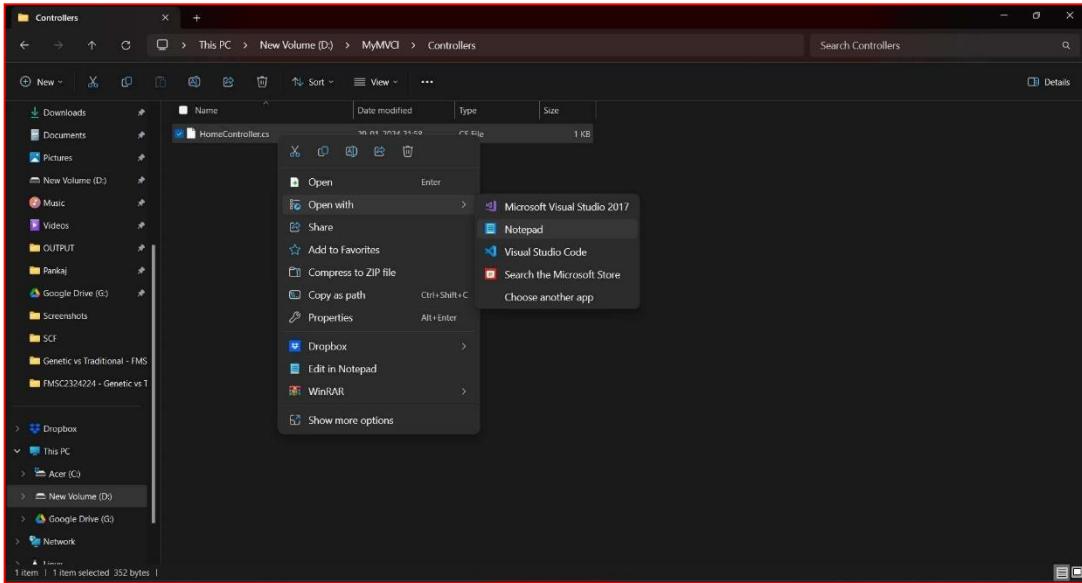
Code:-

```
@{
    ViewData["Title"] = "Home Page";
}

<div>
Symbol:@Model.Symbol</br>
Price:$@Model.Price</br>
</div>
```



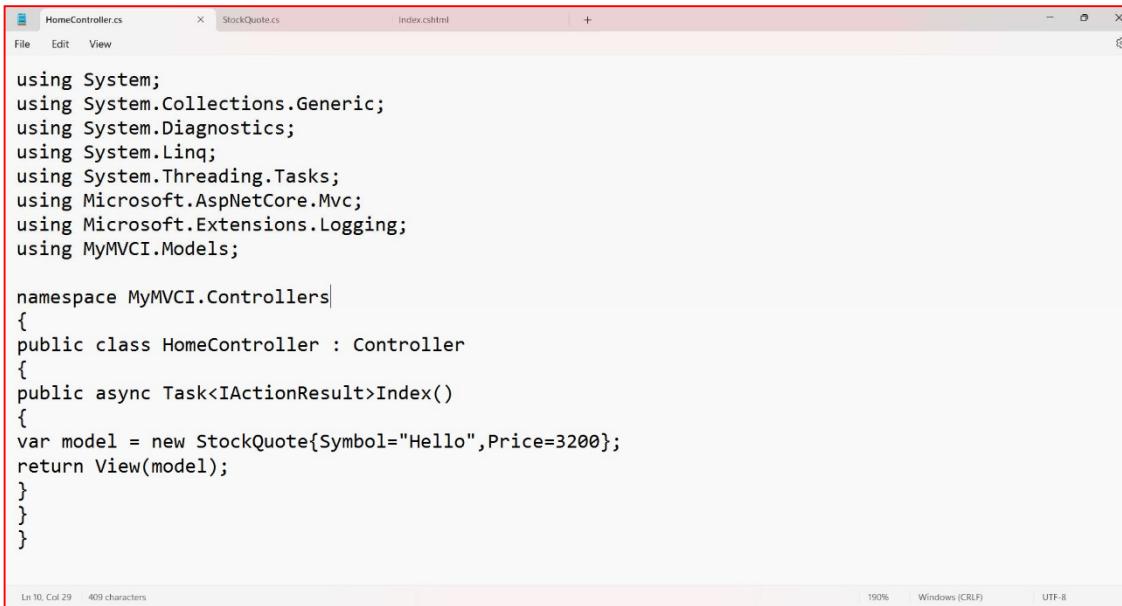
Step 7: Now return open the earlier used file named ‘HomeController.cs’ file in the ‘Controllers’ folder which is inside the MyMVCI folder, modify the code using the below code and save it.



Code:-

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC1.Models;

namespace MyMVC1.Controllers
{
    public class HomeController : Controller
    {
        public async Task<IActionResult> Index()
        {
            var model = new StockQuote{Symbol="Hello",Price=3200};
            return View(model);
        }
    }
}
```



```
HomeController.cs      StockQuote.cs      Index.cshtml
File Edit View

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC1.Models;

namespace MyMVC1.Controllers
{
    public class HomeController : Controller
    {
        public async Task<IActionResult> Index()
        {
            var model = new StockQuote{Symbol="Hello",Price=3200};
            return View(model);
        }
    }
}

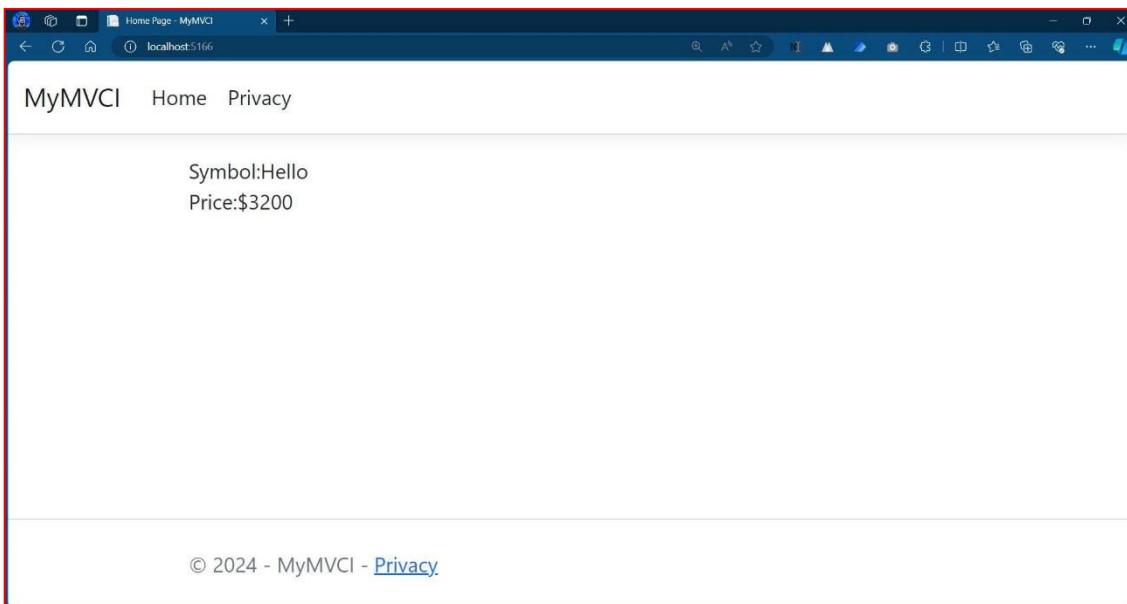
Ln 10, Col 29  409 characters  190%  Windows (CRLF)  UTF-8
```

Now go to the command prompt and run the **dotnet run** command from the output copy the url i.e. for e.g. here the url is **http://localhost:5166** & paste the copied url in your browser and press enter

```
Command Prompt - dotnet n
Now listening on: http://localhost:5166
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\MyMVCI
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...

D:\MyMVCI>dotnet run
Building...
D:\MyMVCI\Controllers\HomeController.cs(14,33): warning CS1998: This async method lacks 'await' operators and
will run synchronously. Consider using the 'await' operator to await non-blocking API calls, or 'await Task.
Run(...)' to do CPU-bound work on a background thread. [D:\MyMVCI\MyMVCI.csproj]
D:\MyMVCI\Models\StockQuote.cs(6,15): warning CS8618: Non-nullable property 'Symbol' must contain a non-null
value when exiting constructor. Consider declaring the property as nullable. [D:\MyMVCI\MyMVCI.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5166
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\MyMVCI
|
```

Output (For step 7):-



Practical 4

AIM:- WORKING WITH DOCKER COMMANDS, IMAGES AND CONTAINER

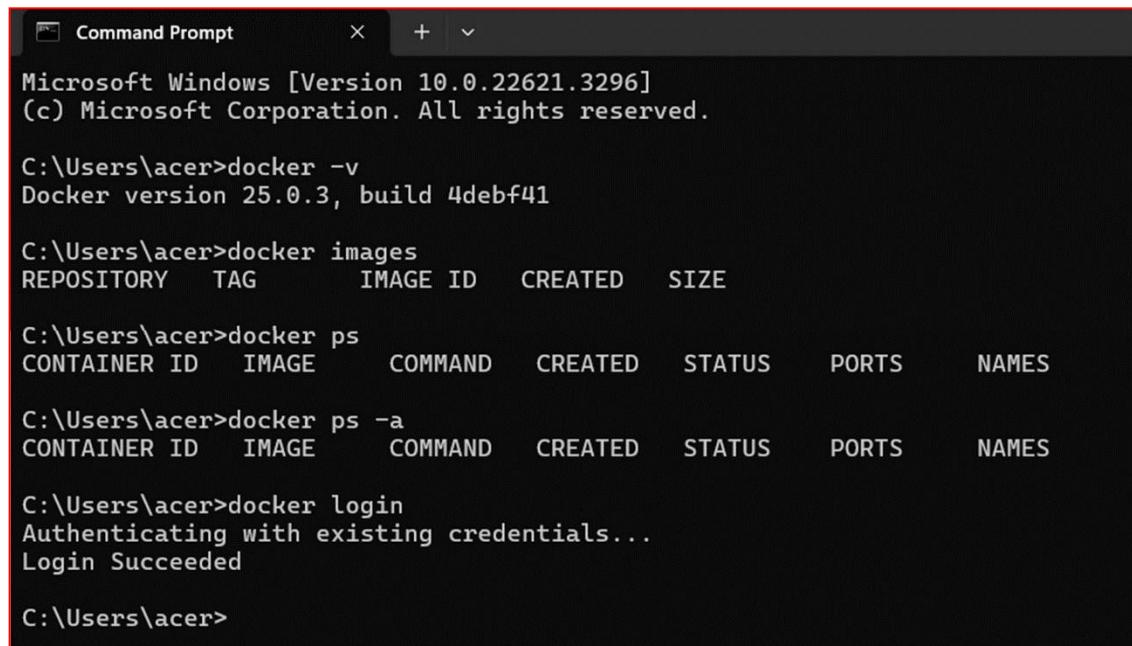
Step 1:-

Download docker desktop and install it. After installation open the command prompt and execute the below code. Remember to sign in with your github account in order to execute the last docker login command successfully.

Command:

```
> docker -v  
> docker images  
> docker ps  
> docker ps -a  
> docker login
```

Output:-



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The output of the commands is as follows:

```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>docker -v
Docker version 25.0.3, build 4debf41

C:\Users\acer>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

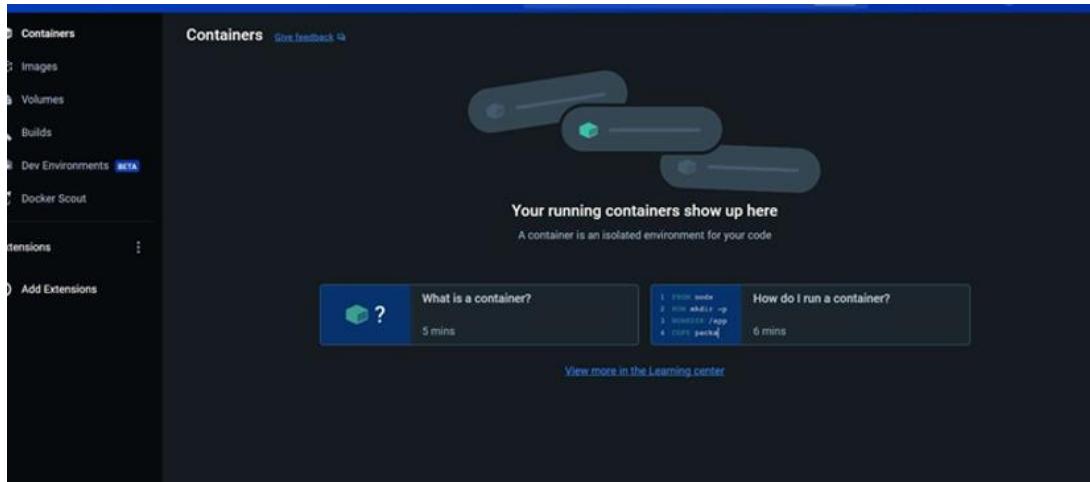
C:\Users\acer>docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES

C:\Users\acer>docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES

C:\Users\acer>docker login
Authenticating with existing credentials...
Login Succeeded

C:\Users\acer>
```

Note: In the below code neeraj24kumar refers to your account name



Command:

- > docker pull hello-world
- > docker tag hello-world neeraj24kumar/hello-world
- > docker tag hello-world neeraj24kumar/hello-world:v1

Output:-

```
C:\Users\acer>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:53641cd209a4fecfc68e21a99871ce8c6920b2e7502df0a20671c6fccc73a7c6
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview hello-world

C:\Users\acer>docker tag hello-world neeraj24kumar/hello-world
C:\Users\acer>docker tag hello-world neeraj24kumar/hello-world:v1
```

Command:

- > docker push neeraj24kumar/hello-world
- > docker push neeraj24kumar/hello-world:v1

Output:-

```
C:\Users\acer>docker push neeraj24kumar/hello-world
Using default tag: latest
The push refers to repository [docker.io/neeraj24kumar/hello-world]
ac28800ec8bb: Mounted from library/hello-world
latest: digest: sha256:d37ada95d47ad12224c205a938129df7a3e52345828b4fa27b03a98825d1e2e7 size: 524

C:\Users\acer>docker push neeraj24kumar/hello-world:v1
The push refers to repository [docker.io/neeraj24kumar/hello-world]
ac28800ec8bb: Layer already exists
v1: digest: sha256:d37ada95d47ad12224c205a938129df7a3e52345828b4fa27b03a98825d1e2e7 size: 524
```

Go to Docker hub through your defined browser, login and see the repository

The screenshot shows a browser window with multiple tabs open. The active tab is 'https://hub.docker.com/repository/docker/neeraj24kumar/hello-world/general'. The page title is 'neeraj24kumar / neeraj24kumar/hello-world / General'. It includes sections for 'General' (with a note to add a short description), 'Tags' (listing 'v1' and 'latest'), and 'Automated Builds' (with a note about connecting to GitHub or Bitbucket). There are also 'Docker commands' and a 'Public View' button.

Command:

- > docker images
- > docker ps
- > docker ps -a

Output:-

```
C:\Users\acer>docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
neeraj24kumar/hello-world    latest   d2c94e258dcb  11 months ago  13.3kB
neeraj24kumar/hello-world    v1      d2c94e258dcb  11 months ago  13.3kB
hello-world          latest   d2c94e258dcb  11 months ago  13.3kB

C:\Users\acer>docker ps
CONTAINER ID        IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES

C:\Users\acer>docker ps -a
CONTAINER ID        IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
```

Command:

- > docker rmi -f d2c94e258dcb (Image ID)

Output:-

```
C:\Users\acer>docker rmi -f d2c94e258dcb
Untagged: hello-world:latest
Untagged: hello-world@sha256:53641cd209a4fecfc68e21a99871ce8c6920b2e7502df0a20671c6fccc73a7c6
Untagged: neeraj24kumar/hello-world:latest
Untagged: neeraj24kumar/hello-world:v1
Untagged: neeraj24kumar/hello-world@sha256:d37ada95d47ad12224c205a938129df7a3e52345828b4fa27b03a98825d1e2e7
Deleted: sha256:d2c94e258dc3c5ac2798d32e1249e42ef01cba4841c2234249495f87264ac5a
Deleted: sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa81e
```

Command:

```
> docker pull neeraj24kumar/hello-world
> docker run -it --name phpcontainer php:7.2-cli /bin/bash
> #exit
```

Output:-

```
C:\Users\acer>docker pull neeraj24kumar/hello-world
Using default tag: latest
latest: Pulling from neeraj24kumar/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d37ada95d47ad12224c205a938129df7a3e52345828b4fa27b03a98825d1e2e7
Status: Downloaded newer image for neeraj24kumar/hello-world:latest
docker.io/neeraj24kumar/hello-world:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview neeraj24kumar/hello-world

C:\Users\acer>docker run -it --name phpcontainer php:7.2-cli /bin/bash
Unable to find image 'php:7.2-cli' locally
7.2-cli: Pulling from library/php
6ec7b7d162b2: Pull complete
db606474d60c: Pull complete
afb30f0cd8e0: Pull complete
3bb2e8051594: Pull complete
c4b2cf9245a3: Pull complete
4058dc7d24d3: Pull complete
430dfa3bb1e0: Pull complete
64a6b2d5dbc7: Pull complete
646bcd25dd59: Pull complete
3656e1195ad5: Pull complete
Digest: sha256:42ffbc0798e4449bbd1e14fc4dcba87774aa1ad1900a09ef6a965bc0880aa2161
Status: Downloaded newer image for php:7.2-cli
root@32d11a2a18d3:/# exit
exit

C:\Users\acer>
```

Command:

```
> docker ps -a
> docker system prune -a
> docker images
> docker ps -a
> docker volume ls
```

Output:-

```
C:\Users\acer>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED     STATUS      PORTS     NAMES
32d11a2a18d3   php:7.2-cli "docker-php-entrypoi..."  2 minutes ago   Exited (0) 2 minutes ago   phpcontainer

C:\Users\acer>docker system prune -a
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all images without at least one container associated to them
- all build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
32d11a2a18d3ca9e169ec339386b7a4226724b51a4d7f0f3601e7960cb71d184

Deleted Images:
untagged: neeraj24kumar/hello-world:latest
untagged: neeraj24kumar/hello-world@sha256:d37ada95d47ad12224c205a938129df7a3e52345828b4fa27b03a98825d1e2e7
deleted: sha256:d2c94e258dcbb3c5ac2798d32e1249e42ef01cba4841c2234249495f87264ac5a
deleted: sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa81e
untagged: php:7.2-cli
untagged: php@sha256:42ffbc0798e449bbd1e14fc4dcb87774aa1ad1900a09ef6a965bc0880aa2161
deleted: sha256:9f290ac43498b3f88a0c8bbc8c326bcfb92e9eae31a41b76c62db5414e3b40c
deleted: sha256:b095b573bbe4a11032718bc7ee3a89beed39644e2d9ac07c122b44fd2e1246e
deleted: sha256:99b22386397afe0bf9ce85c2581f271a492558b8d954fd5e412a2990690b6a29
deleted: sha256:9623a6aede0a8a4788ea355cf216438d1ca2055857823ded972b780d1ea769d
deleted: sha256:221e31a6b72ab86db44222c2bb7594ea8fc20ffd4ac236605d67462a4de6ef2
deleted: sha256:7051e2fdd9d146dee1ebf8c338be03c725d470f7f9049eb80ca42c44aea419
deleted: sha256:3ea87e284ca869f56b5af68cf07d961df87b8a911972e7d1ece60ee80ddb01
deleted: sha256:a953d04decb53b836e21dfc15a2b8f9bbecaad1085ef606b0d9db21b69078337
deleted: sha256:8a8f49ca28d2e5ca5aa9455c92331b26ae131d2c78a6be0e062429cf3812552
deleted: sha256:db10680b18837fbef39e1e6452ff5448803e1e7b0c13eda5d53b19743e09d947
deleted: sha256:87c8a1d8f54f3aa4e05569e8919397b65056aa71cdf48b7f061432c98475eee9

Total reclaimed space: 397.9MB
```

```
C:\Users\acer>docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE

C:\Users\acer>docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES

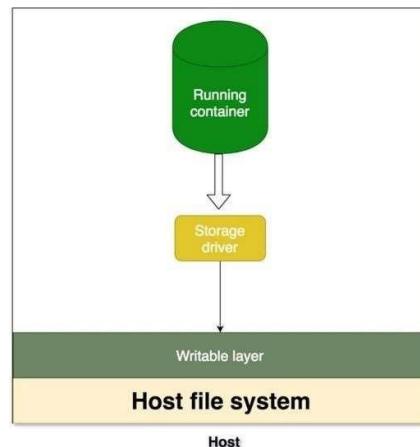
C:\Users\acer>docker volume ls
DRIVER      VOLUME NAME

C:\Users\acer>
```

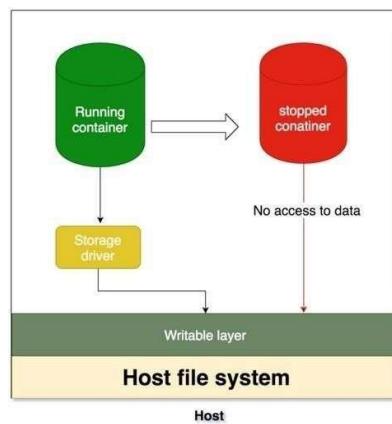
Practical 5

AIM:- WORKING WITH DOCKER VOLUMES AND NETWORKS (Play with Docker)

Before going deep into volumes, Let's understand how containers persist data in the host filesystem

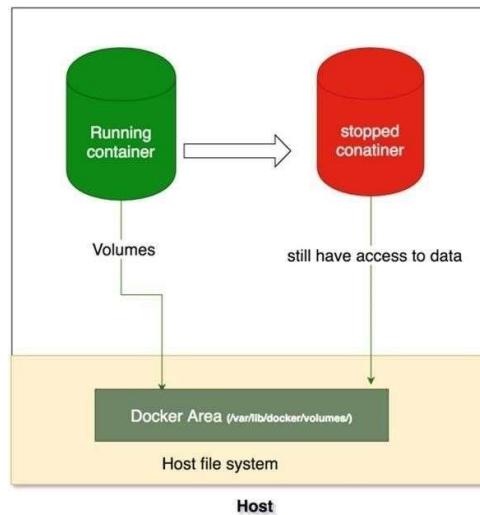


- If we look at the above diagram, whenever running container wants to persist data, it actually put that data into the writable layer through storage driver.
- What are the problems:
- Data is no longer persisted and difficult to access if container stops as shown in the following diagram
- As we can see writable layer is tightly coupled with host filesystem and difficult to move the data.
- We have an extra layer of abstraction with a storage driver which reduces the performance.



- How Volumes can solve above issues
- Volumes are saved in the host filesystem (/var/lib/docker/volumes/) which is owned and maintained by docker.
- Any other nondocker process can't access it.

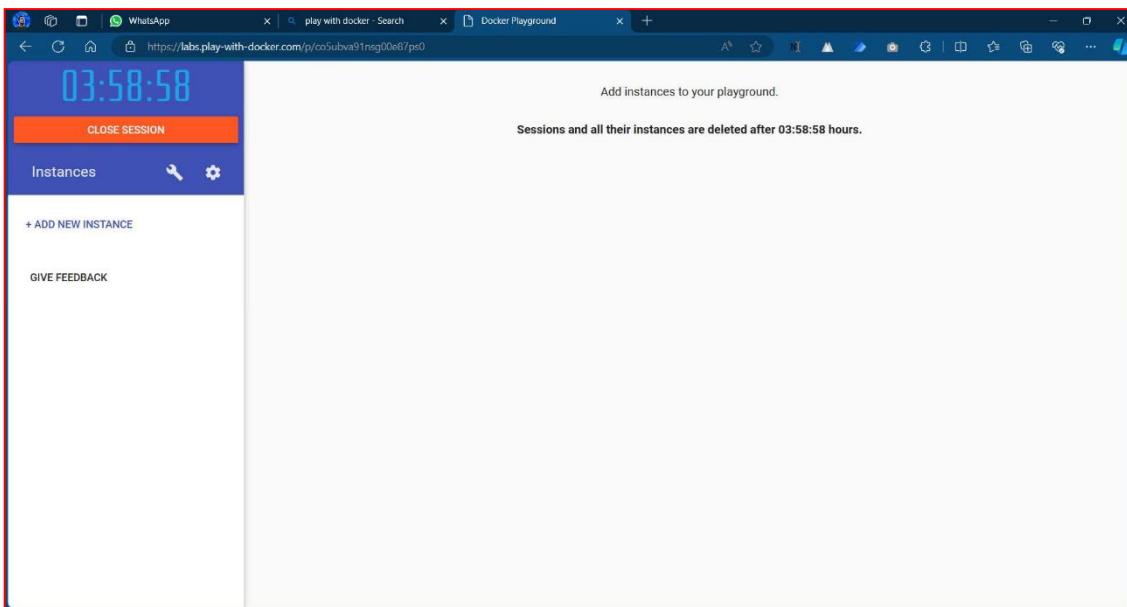
- But, As depicted in the below other docker processes/containers can still access the data even container is stopped since it is isolated from the container file system.



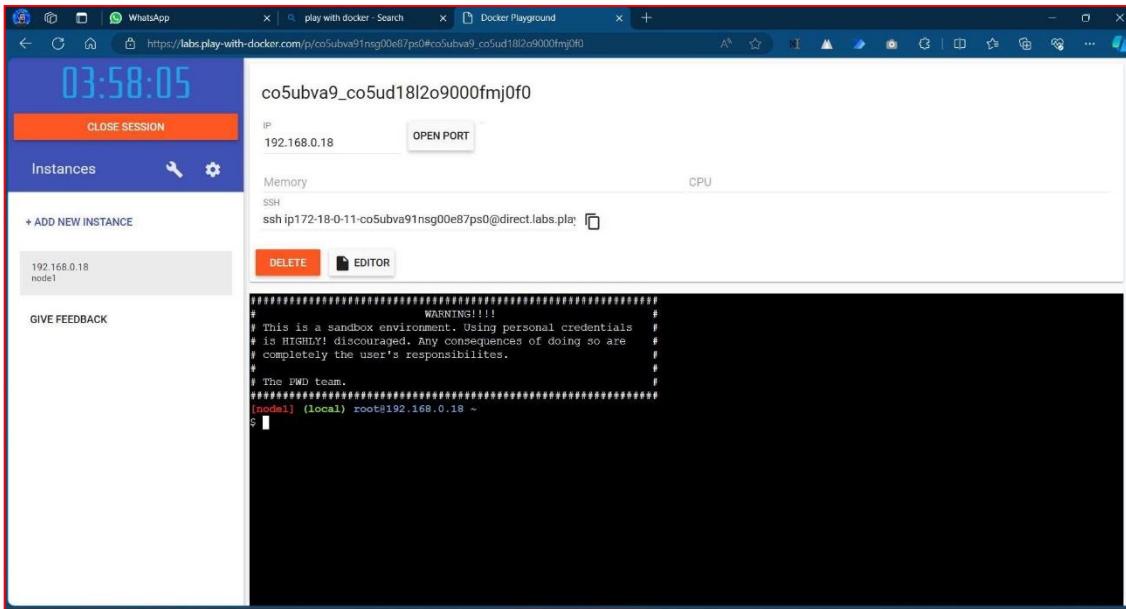
Steps:

- 1) Go to website Play with Docker and login, login with your existing docker account as soon as you have logged in successfully the session will start.

(Link: Play with Docker (play-with-docker.com))



2) Add new instance (after clicking on add new instance new window appear)



3) Perform Following Inside Play-With-Docker

4) Pull nginx image into docker

Command:

docker pull nginx

Output:

```

#####
##### WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node1] (local) root@192.168.0.18 ~
$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
3a1e25ce7c4f: Pull complete
e78b137be355: Pull complete
39fc875bd2b2: Pull complete
035788421403: Pull complete
87c3fb37cbf2: Pull complete
c5cdd1ce752d: Pull complete
33952c599532: Pull complete
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febfb0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[node1] (local) root@192.168.0.18 ~
$ 

```

5) Now run the pulled image in Container named “webApp”

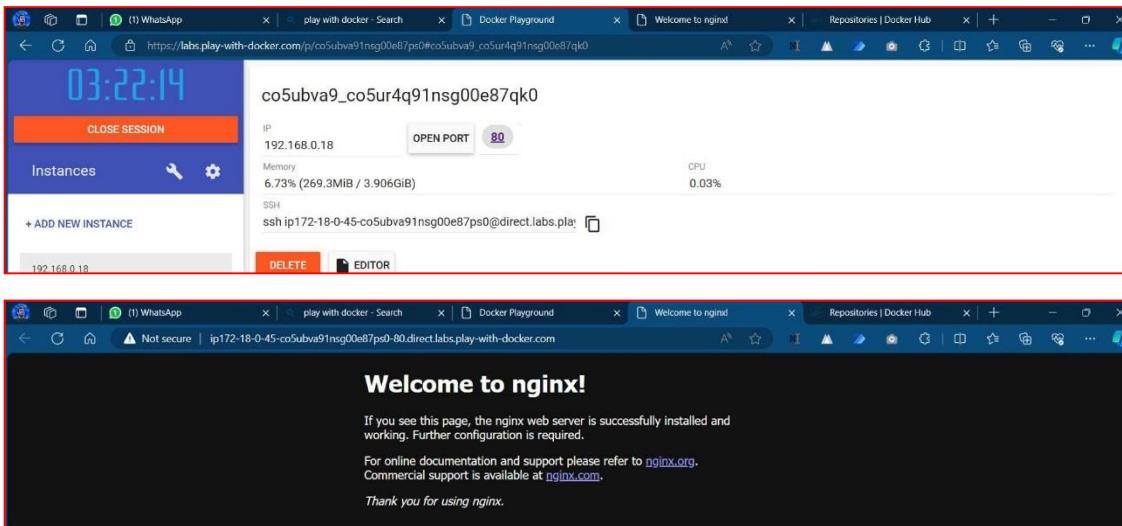
Command:

```
docker run -it --name=webApp -d -p 80:80 nginx
```

Output:

```
$ docker run -it --name=webApp -d -p 80:80 nginx
9240661bbb9a5d9eb810ec16e2f3caf1d7e922b418691aacd9d8e40799b753f2
[node1] (local) root@192.168.0.18 ~
```

6) Click on port 80 to check output (it shows welcome page)



7) We make changes into “index.html” file inside /usr/share/nginx/html folder

Command:

```
docker exec -it webApp bash
cd /usr/share/nginx/html
echo "Hello KB" > index.html
exit
```

Output:

```
$ docker exec -it webApp bash
root@9240661bbb9a:/# cd /usr/share/nginx/html
root@9240661bbb9a:/usr/share/nginx/html# echo "Hello KB" > index.html
root@9240661bbb9a:/usr/share/nginx/html# exit
exit
[node1] (local) root@192.168.0.18 ~
```

8) Type docker prompt and check process status using ps option

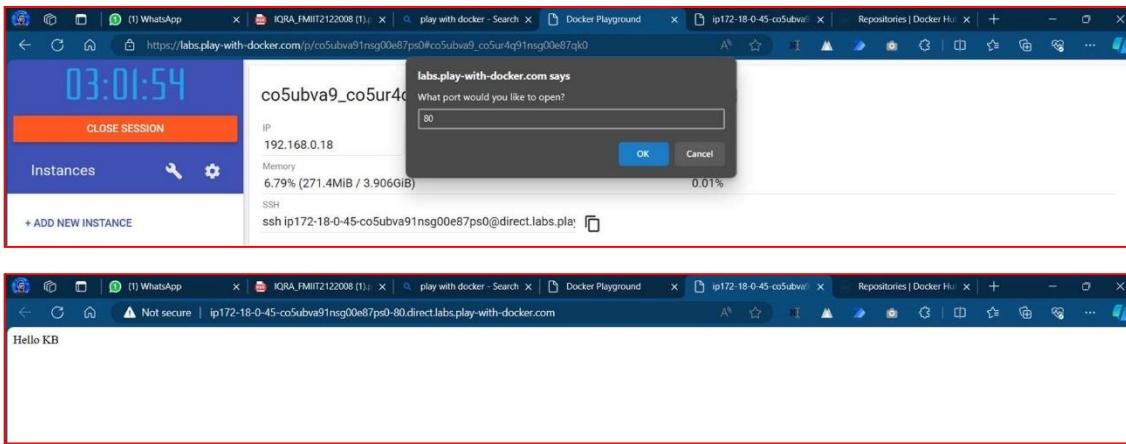
Command:

docker ps

Output:

```
$ docker ps
CONTAINER ID   IMAGE     COMMAND          CREATED      STATUS      PORTS          NAMES
9240661bbb9a   nginx    "/docker-entrypoint...."  6 minutes ago   Up 6 minutes   0.0.0.0:80->80/tcp   webApp
[node1] (local) root@192.168.0.18 ~
```

9) Click on open port, enter port as 80 then click OK and after the page is open refresh the page to see the desired output



10) Now stop running container named “webApp”.

Command:

docker stop webApp

Output:

```
[node1] (local) root@192.168.0.18 ~
$ docker stop webApp
webApp
[node1] (local) root@192.168.0.18 ~
```

11) Create new instance and Start nginx in new container named as “webApp1”.

Command:

docker run -it --name=webApp1 -d -p 80:80 nginx

Output:

```

#                               WARNING!!!!
# This is a sandbox environment. Using personal credentials #
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilites. #
#
# The PWD team. #
#####
[node2] (local) root@192.168.0.17 ~
$ docker run -it --name=webApp1 -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8a1e25ce7c4f: Pull complete
e78b137be355: Pull complete
39fc875bd2b2: Pull complete
035788421403: Pull complete
87c3fb37cbf2: Pull complete
c5cdd1ce752d: Pull complete
33952c599532: Pull complete
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
1ded4112ad83d0a793908c2d04630c686e45d73ec5eddf0177bb3f35edfcbccb
[node2] (local) root@192.168.0.17 ~

```

- 12) Click on port 80 to check output (it shows welcome page)



- 13) Go back to node1 and create new volume

Command:

```

docker volume create MyVolume
docker volume ls
docker volume inspect MyVolume

```

Output:

```
[node1] (local) root@192.168.0.18 ~
$ docker volume create MyVolume
MyVolume
[node1] (local) root@192.168.0.18 ~
$ docker volume ls
DRIVER      VOLUME NAME
local      MyVolume
[node1] (local) root@192.168.0.18 ~
$ docker volume inspect MyVolume
[
  {
    "CreatedAt": "2024-04-02T12:08:02Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/MyVolume/_data",
    "Name": "MyVolume",
    "Options": null,
    "Scope": "local"
  }
]
[node1] (local) root@192.168.0.18 ~
```

- 14) Mount this volume to nginx new container named “webApp5”

Command:

```
docker run -d --name=webApp5 --mount source=MyVolume,destination=/usr/share/nginx/html -p 80:80 nginx
```

Output:

```
[node1] (local) root@192.168.0.18 ~
$ docker run -d --name=webApp5 --mount source=MyVolume,destination=/usr/share/nginx/html/ -p 80:80 nginx
45f3327e6d72de9868d33eb8b30f5d0ce9d8565042383d66c7d86ce352587494
[node1] (local) root@192.168.0.18 ~
```

- 15) Now keep on doing “ls” and “cd “ to go inside _data folder of our volume “MyVolume”

Command:

```
ls /
cd /var/lib/docker
ls
```

Output:

```
[node1] (local) root@192.168.0.18 ~
$ ls /
bin      dev      etc      lib      mnt      proc      run      srv      tmp      var
certs   docker.log home    media    opt      root      sbin     sys      usr
[node1] (local) root@192.168.0.18 ~
$ cd /var/lib/docker
[node1] (local) root@192.168.0.18 /var/lib/docker
$ ls
buildkit  containers  image      overlay2  runtimes  tmp
containerd engine-id  network   plugins    swarm     volumes
[node1] (local) root@192.168.0.18 /var/lib/docker
```

Command:

```
cd volumes
ls
cd MyVolume
ls
```

Output:

```
[node1] (local) root@192.168.0.18 /var/lib/docker
$ cd volumes
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes
$ ls
MyVolume          backingFsBlockDev  metadata.db
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes
$ cd MyVolume
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume
$ ls
_data
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume
```

Command:

```
cd _data
ls
```

Output:

```
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume
$ cd _data
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
$ ls
50x.html  index.html
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
```

16) Modify contents of index.html file with “from MyVolume hello KB”

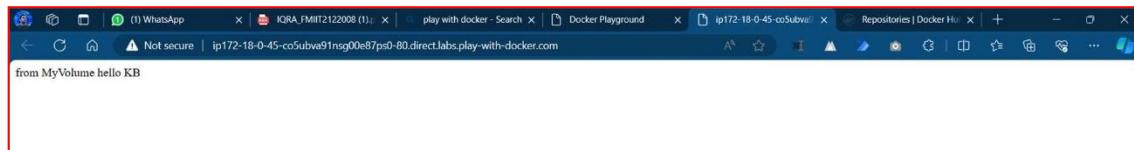
Command:

```
echo "from MyVolume hello KB" > index.html
```

Output:

```
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
$ echo "from MyVolume hello KB" > index.html
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
```

17) Open the port 80 (to get the modified output)



18) Now stop this running container named “webApp5”

Command:

```
docker stop webApp5
```

Output:

```
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
$ docker stop webApp5
webApp5
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
```

19) Now run nginx in new container named “webApp7”

Command:

```
docker run -d --name=webApp7 --mount source=MyVolume,destination=
/usr/share/nginx/html -p 80:80 nginx
```

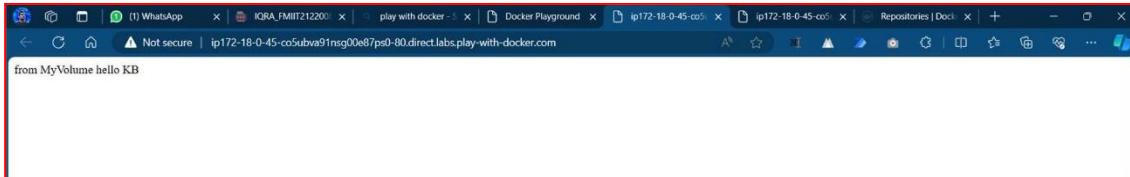
Output:

```
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
$ docker run -d --name=webApp7 --mount source=MyVolume,destination=/usr/share/nginx/html/ -p 80:80 nginx
16a3775505df9fb335e6cef533ec3b2a3493b849ffdb5c4a122cd52e5dd4e8c0
[node1] (local) root@192.168.0.18 /var/lib/docker/volumes/MyVolume/_data
```

20) Click on port 80 and refresh the page you should get edited file as output.

We can load the page again localhost:80 and still see the html file that we edited in the volume.

So, with the help of volumes, we can easily access the data even we stop the container and it's very easy to access data and import the data to anywhere.



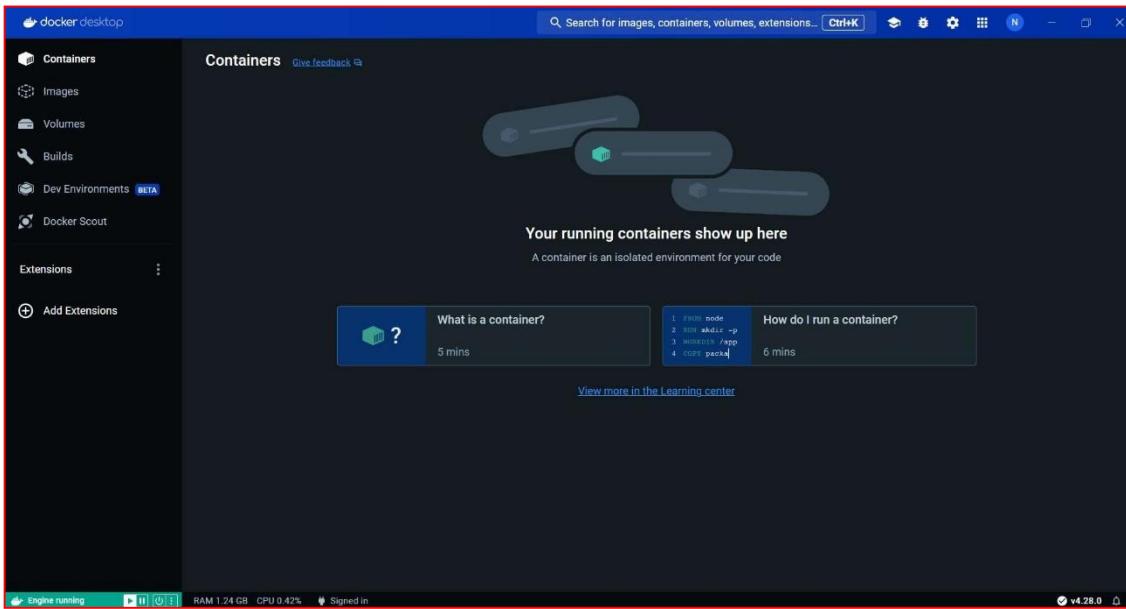
Practical 6

AIM:- WORKING WITH DOCKER SWARM

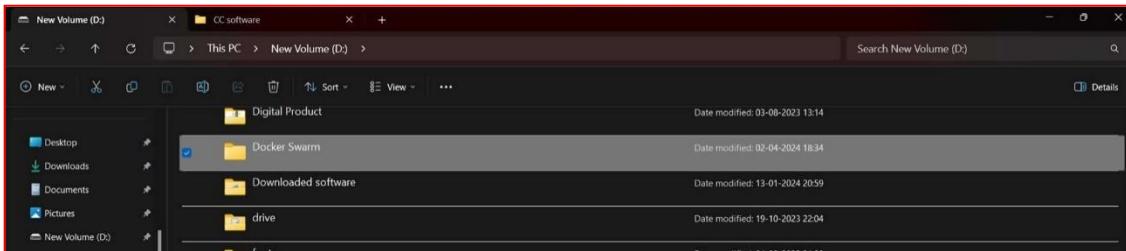
(P.S. This practical is implemented on a single node which acts both as manager and worker)

Steps:

- 1) Login the Docker with github



- 2) Create a Folder in local drive with name “DockerSwarm”. Here it is created in D: drive



- 3) Open the command prompt and enter the commands

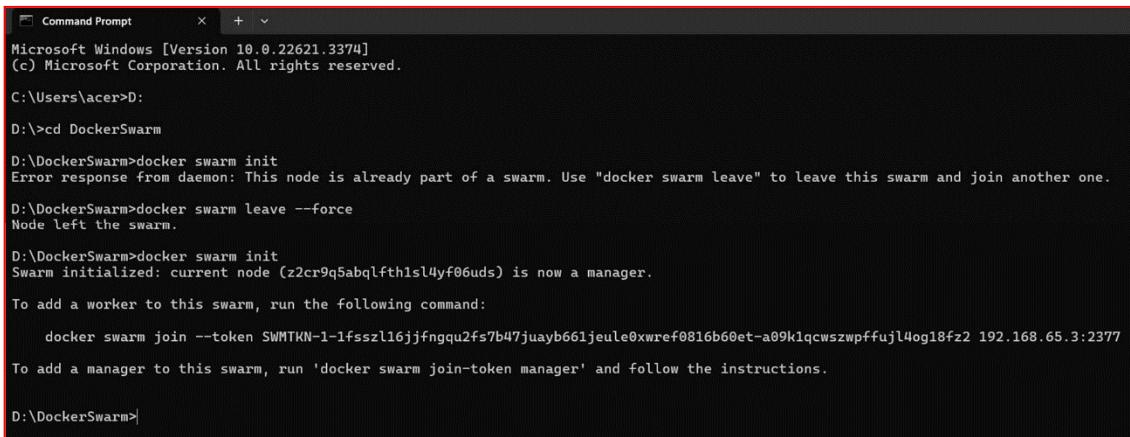
Commands:

D:

cd DockerSwarm

docker swarm init

Output:-



```

Command Prompt      + 
Microsoft Windows [Version 10.0.22621.3374]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer>D:
D:\>cd DockerSwarm
D:\DockerSwarm>docker swarm init
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm and join another one.

D:\DockerSwarm>docker swarm leave --force
Node left the swarm.

D:\DockerSwarm>docker swarm init
Swarm initialized: current node (z2cr9q5abqlfth1sl4yf06uds) is now a manager.

To add a worker to this swarm, run the following command:

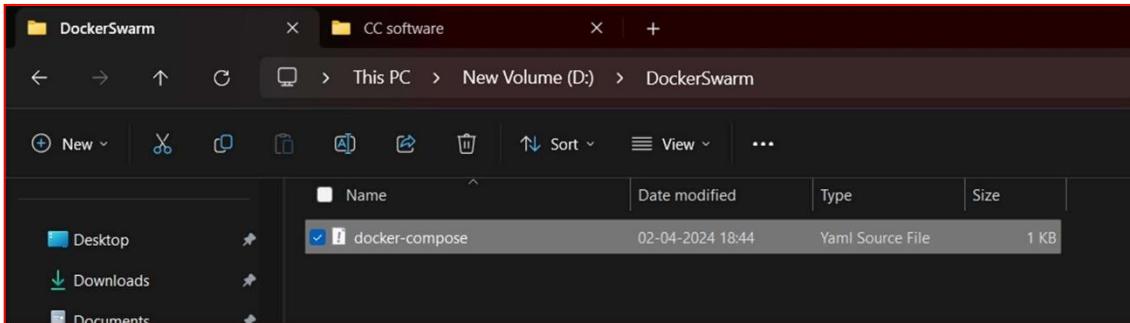
    docker swarm join --token SWMTKN-1-1fsszl16jjfnngqu2fs7b47juayb661jeule0xwref0816b60et-a09k1qcwszwppffujl4og18fz2 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

D:\DockerSwarm>

```

4) Open the folder Docker Swarm in D: drive add “**docker-compose.yml**” file in it and add the following contents in it



Code:-

version : '3'

services :

database:

image : redis

deploy :

 replicas : 2

web:

 image : nginx

 deploy :

 replicas : 2

 ports :

 - 8500:80

```

visualizer:
  image : dockersamples/visualizer
  deploy:
    placement:
      constraints: [node.role == manager]
  ports :
    - 8080:8080
  volumes :
    - /var/run/docker.sock:/var/run/docker.sock

```

Command:

docker-compose config

Output:-

```

D:\DockerSwarm>docker-compose config
services:
  database:
    deploy:
      replicas: 2
    image: redis
    networks:
      default: null
  visualizer:
    deploy:
      placement:
        constraints:
          - node.role == manager
    image: dockersamples/visualizer
    networks:
      default: null
    ports:
      - mode: ingress
        target: 8080
        published: "8080"
        protocol: tcp
    volumes:
      - type: bind
        source: /var/run/docker.sock
        target: /var/run/docker.sock
        bind:
          create_host_path: true
  web:
    deploy:
      replicas: 2
    image: nginx
    networks:
      default: null
    ports:
      - mode: ingress
        target: 80
        published: "8500"
        protocol: tcp
networks:
  default:
    name: dockerswarm_default
D:\DockerSwarm>

```

Command:

docker stack deploy nodeapp -c docker-compose.yml

Output:-

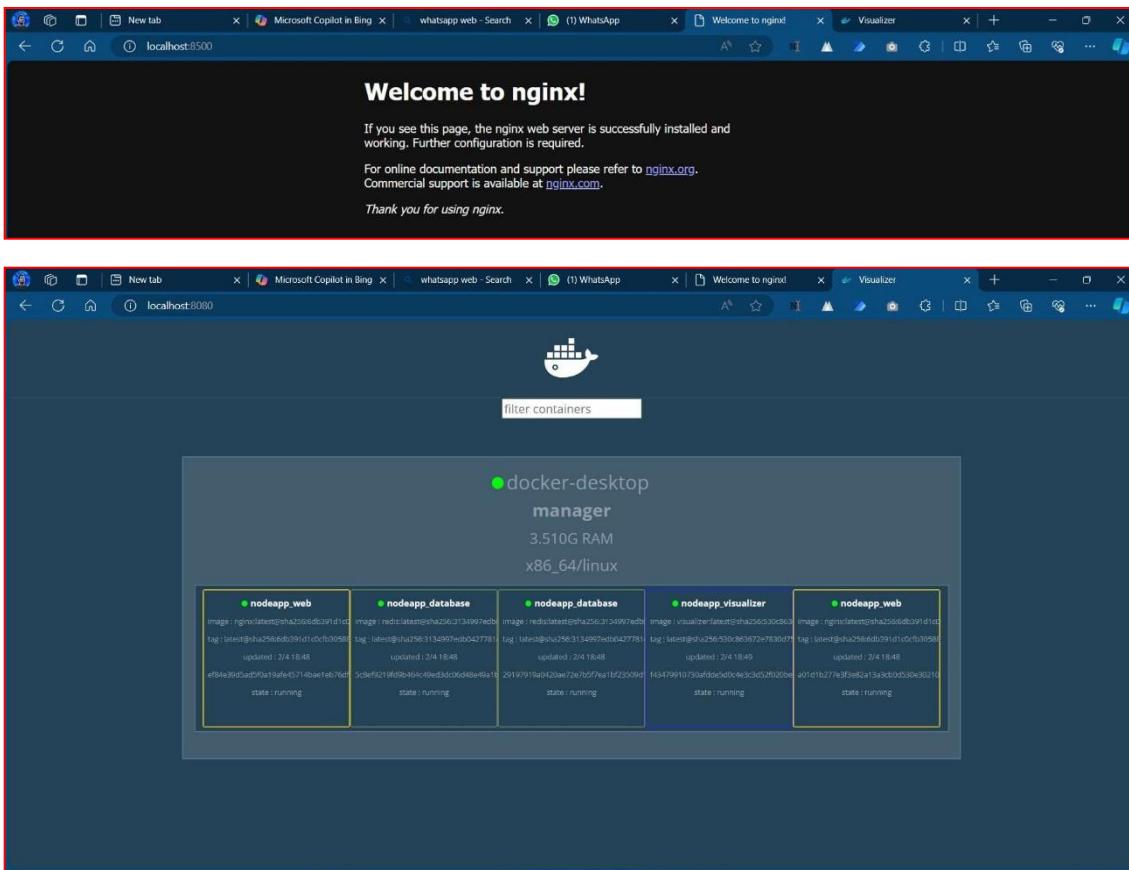
```

D:\DockerSwarm>docker stack deploy nodeapp -c docker-compose.yml
Creating network nodeapp_default
Creating service nodeapp_web
Creating service nodeapp_visualizer
Creating service nodeapp_database

D:\DockerSwarm>docker stack deploy nodeapp -c docker-compose.yml
Updating service nodeapp_database (id: aahnlv94yf8yuit0x7lm6qdm5)
Updating service nodeapp_web (id: hwcanv2i4urgxus1sb9n4voxj)
Updating service nodeapp_visualizer (id: n3df9e8cvogztrdxgkszryw4t)

```

5) Go to browser and check the “localhost:8500” and “localhost:8080”



Command:

```
docker stack ls
docker stack services nodeapp
docker stack ps nodeapp
docker service scale nodeapp_web=4
```

Output:-

```
D:\DockerSwarm>docker stack ls
NAME      SERVICES
nodeapp   3

D:\DockerSwarm>docker stack services nodeapp
ID          NAME      MODE      REPLICAS  IMAGE
aahnlv94yf8y  nodeapp_database  replicated  2/2    redis:latest
n3df9e8cvgzg  nodeapp_visualizer  replicated 1/1    dockersamples/visualizer:latest  *:8080->8080/tcp
hwcanv2i4urg  nodeapp_web       replicated  2/2    nginx:latest           *:8500->80/tcp

D:\DockerSwarm>docker stack ps nodeapp
ID          NAME      IMAGE      NODE      DESIRED STATE     CURRENT STATE      ERROR      PORTS
mldtpjz1cw8p  nodeapp_database.1  redis:latest  docker-desktop  Running  4 minutes ago
flftp8w4lq9z  nodeapp_database.2  redis:latest  docker-desktop  Running  4 minutes ago
pfzobeyatv72  nodeapp_visualizer.1  dockersamples/visualizer:latest  docker-desktop  Running  3 minutes ago
lw3llytipw5y  nodeapp_web.1      nginx:latest  docker-desktop  Running  4 minutes ago
zwzespe34jdf  nodeapp_web.2      nginx:latest  docker-desktop  Running  4 minutes ago

D:\DockerSwarm>docker service scale nodeapp_web=4
nodeapp_web scaled to 4
overall progress: 4 out of 4 tasks
1/4: running  [=====]
2/4: running  [=====]
3/4: running  [=====]
4/4: running  [=====]
verify: Service converged

D:\DockerSwarm>
```

Command:

```
docker stack ls
docker service ls
docker service ps nodeapp_web
docker stack rm nodeapp
docker swarm leave --force
```

Output:-

```
D:\DockerSwarm>docker stack ls
NAME      SERVICES
nodeapp   3

D:\DockerSwarm>docker service ls
ID        NAME          MODE      REPLICAS  IMAGE
aahnlv94yf8y  nodeapp_database  replicated  2/2      redis:latest
n3df9e8cvogz  nodeapp_visualizer  replicated  1/1      dockersamples/visualizer:latest  *:8080->8080/tcp
hwcanv2i4urg  nodeapp_web       replicated  4/4      nginx:latest
                                         *:8500->80/tcp

D:\DockerSwarm>docker service ps nodeapp_web
ID        NAME          IMAGE          NODE          DESIRED STATE     CURRENT STATE          ERROR          PORTS
1w3lvtipw6y  nodeapp_web.1  nginx:latest  docker-desktop  Running   Running 6 minutes ago
zwzespe34jdf  nodeapp_web.2  nginx:latest  docker-desktop  Running   Running 6 minutes ago
mse5mv55084g  nodeapp_web.3  nginx:latest  docker-desktop  Running   Running about a minute ago
4m2hk2viukml  nodeapp_web.4  nginx:latest  docker-desktop  Running   Running about a minute ago

D:\DockerSwarm>docker stack rm nodeapp
Removing service nodeapp_database
Removing service nodeapp_visualizer
Removing service nodeapp_web
Removing network nodeapp_default

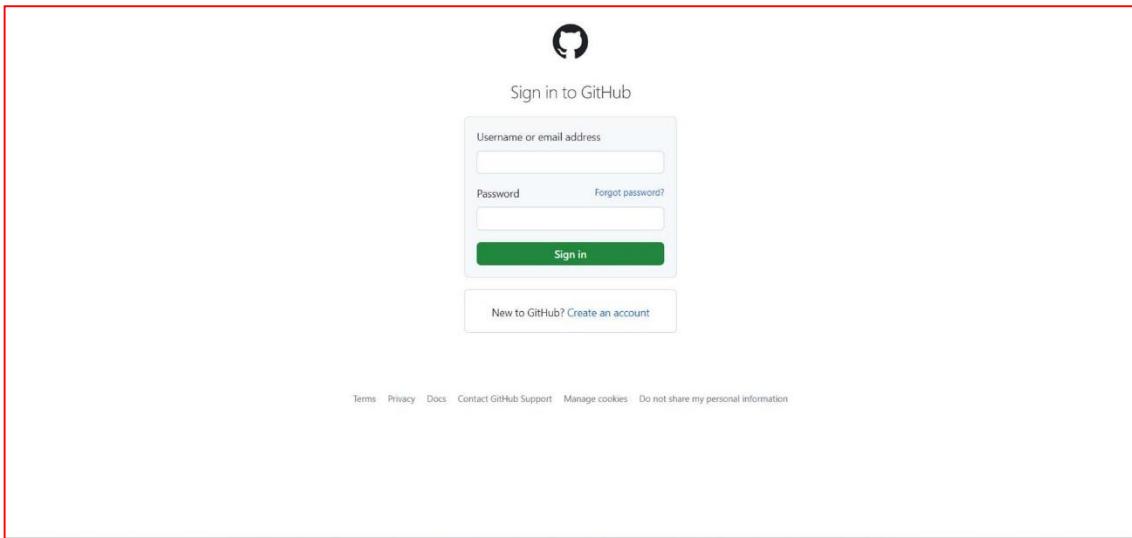
D:\DockerSwarm>docker swarm leave --force
Node left the swarm.
```

Practical 7

AIM:- WORKING WITH CIRCLECI

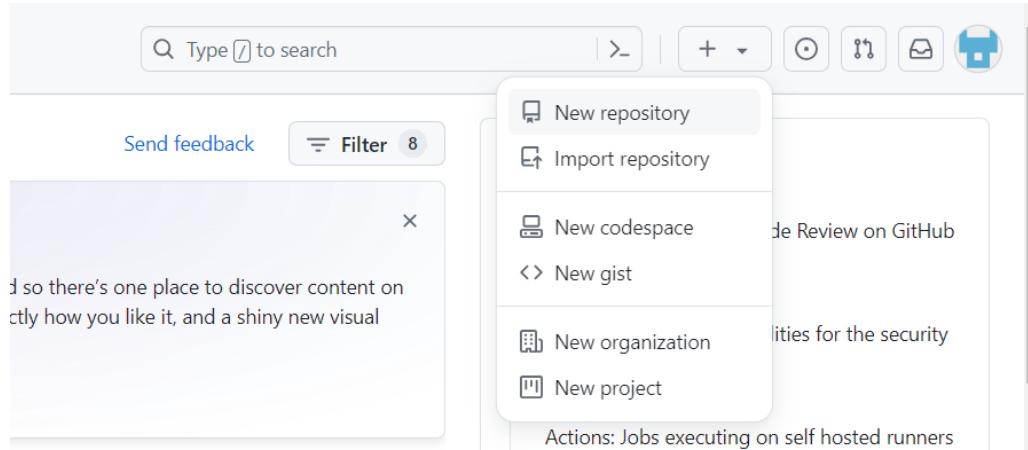
Steps:-

- 1) Create an account in Github or Login in with existing account



The screenshot shows the GitHub dashboard for the user 'mayank21k'. The left sidebar includes sections for "Top Repositories" (with links to 'mscpart1', 'CircleCI-Python', and 'github'), "Recent activity", and a search bar. The main content area has sections for "Updates to your homepage feed", "Start writing code", "Start a new repository for mayank21k" (with a "Create a new repository" button), and "Introduce yourself with a profile README" (containing a sample README file). The right sidebar contains a user profile picture, the username 'mayank21k', and a list of quick links such as "Set status", "Your profile", "Add account", "Your repositories", and "Your organizations". Other links include "Upgrade", "Try Enterprise", "Feature preview", "Settings", "GitHub Support", and "GitHub Community".

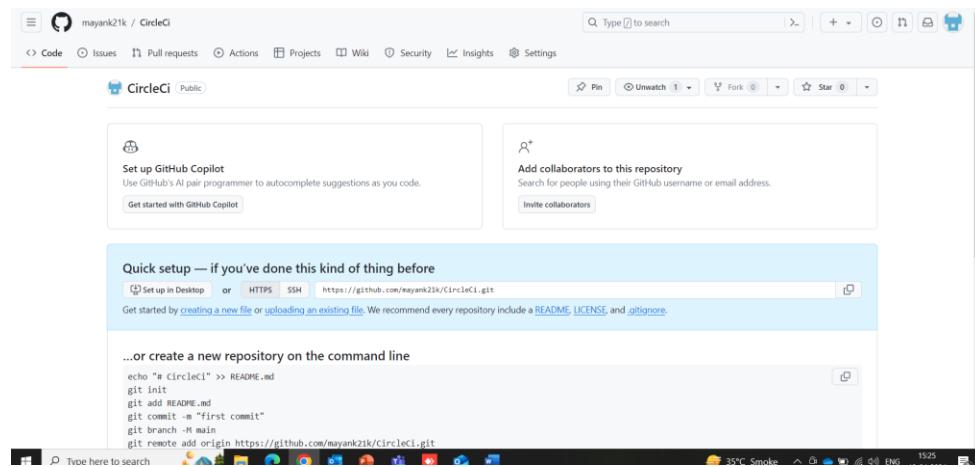
2) Navigate to GitHub and create a new repository. Input the name of your repository and finally, click on new repository



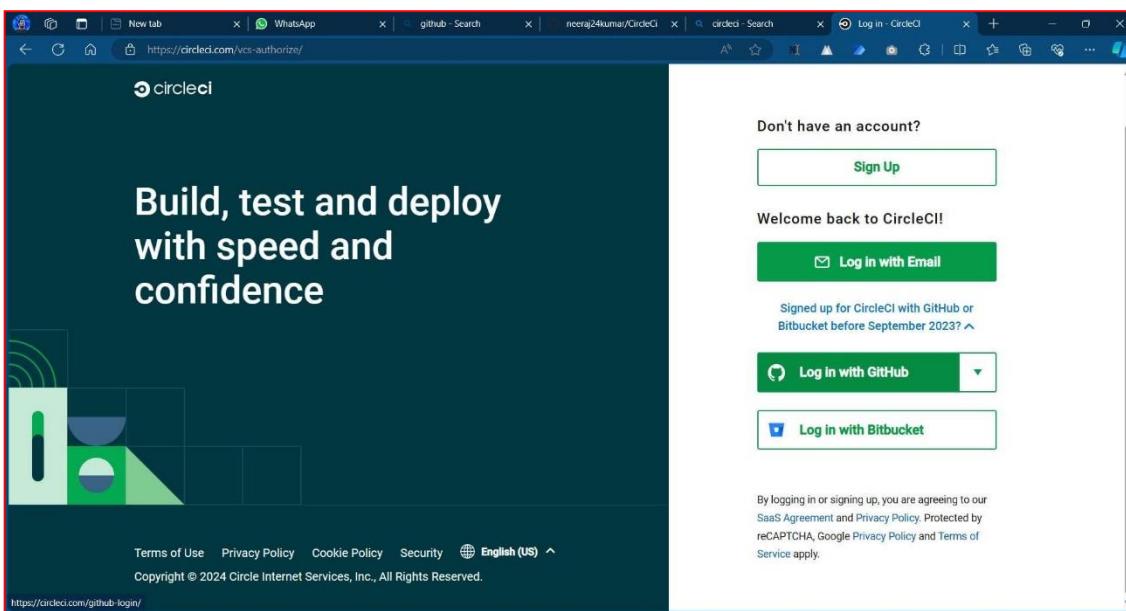
Repository Name:- CircleCi

The screenshot shows the 'Create a new repository' page. At the top, there's a header with a 'New repository' button and a search bar. Below the header, the title 'Create a new repository' is displayed, followed by a note about what a repository is. There are fields for 'Owner' (set to 'mayank21k') and 'Repository name' (set to 'CircleCi'). A note says 'CircleCi is available'. Below these, there's a text area for 'Description (optional)'. Under the 'Repository name' field, there are two radio buttons: 'Public' (selected) and 'Private'. A note for 'Public' says 'Anyone on the internet can see this repository. You choose who can commit.' A note for 'Private' says 'You choose who can see and commit to this repository.' At the bottom, there are sections for initializing the repository ('Add a README file', 'Add .gitignore') and a note about the README file.

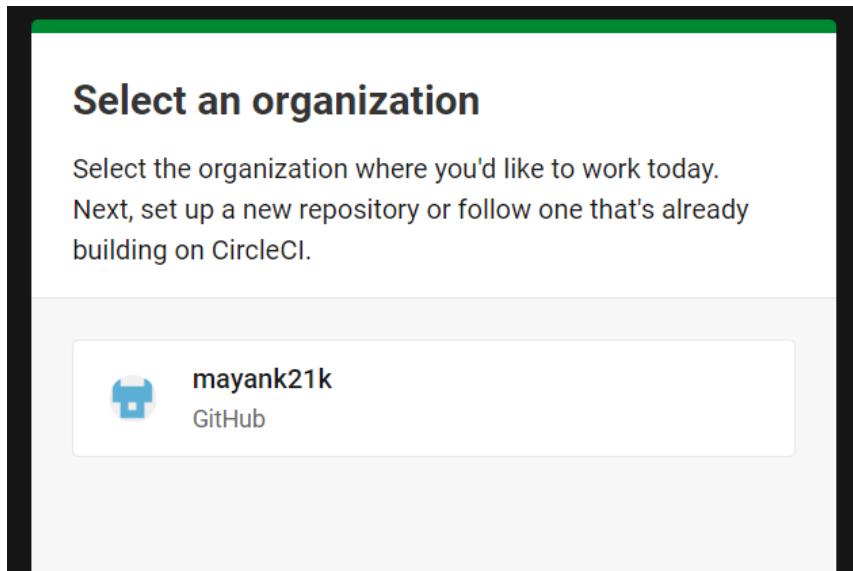
Repository created successfully



3) Create an account on CircleCI by navigating to the sing up page or sign in with GitHub

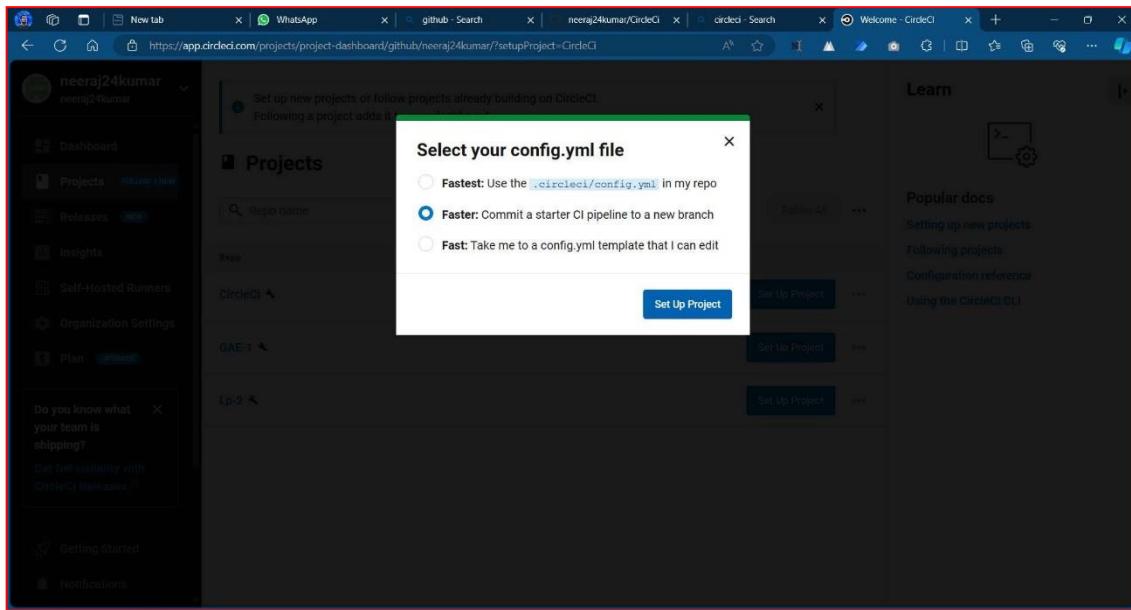


4) Skip setup and then Authorize GitHub with CircleCI.



5) Click on set up project of “CircleCI”

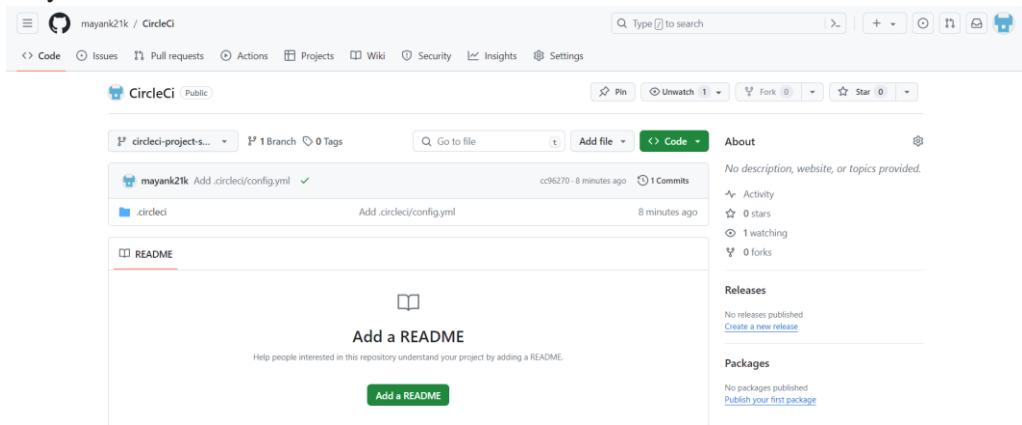
A screenshot of the CircleCI dashboard for the organization 'mayank21k'. The left sidebar shows 'Dashboard', 'Projects' (with a 'FOLLOW 1 NEW' button), 'Releases', 'Insights', 'Self-Hosted Runners', 'Organization Settings', and 'Plan'. A modal window titled 'Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard.' is open. The main area is titled 'Projects' and lists four projects: 'CircleCI' (with a 'Set Up Project' button), 'mscitetpart1', 'CircleCI-Python', and 'github'. Each project has 'Unfollow Project' and a three-dot menu button. To the right, there's a 'Learn' sidebar with a 'Popular docs' section containing links to 'Setting up new projects', 'Following projects', 'Configuration reference', and 'Using the CircleCI CLI'.



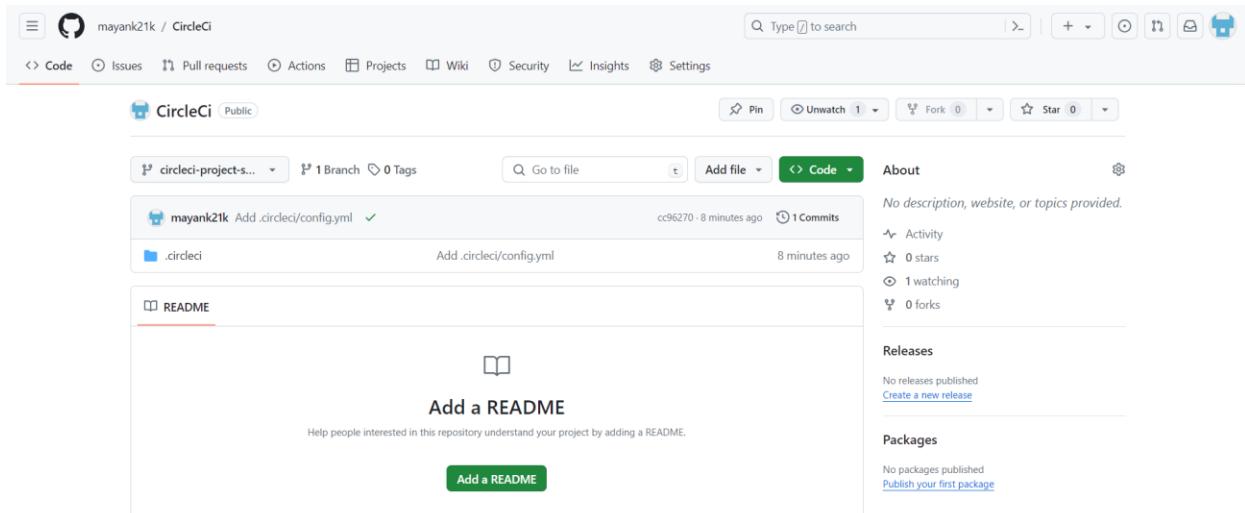
A screenshot of the CircleCI pipeline dashboard for the project "CircleCI". The sidebar includes options for Dashboard, Projects, Releases (marked as NEW), Insights, Self-Hosted Runners, Organization Settings, and Plan (UPGRADE). The main area shows two pipelines: "CircleCI 2" and "CircleCI 1". Both pipelines show a status of "Success" with the "say-hello-workflow". The "Trigger Event" for both was "circleci-project-setup cc96270 Add .circleci/config.yml". The "Start" time was 2m ago, and the "Duration" was 4s. There are "Edit Config", "Trigger Pipeline", and "Project Settings" buttons above the table. A "Filters" section allows selecting "Everyone's Pipelines", "CircleCI", "Select a Branch", and "All days". An "Auto-expand" toggle is also present.

Pipeline	Status	Workflow	Trigger Event	Start	Duration	Actions
CircleCI 2	Success	say-hello-workflow	circleci-project-setup cc96270 Add .circleci/config.yml	2m ago	4s	
CircleCI 1	Success	say-hello-workflow	circleci-project-setup cc96270	2m ago	4s	

6) Now go back to GitHub and click on the repositories and refresh it, you will see “.circleci” is already created.



7) Add two python files in “CircleCi” repository



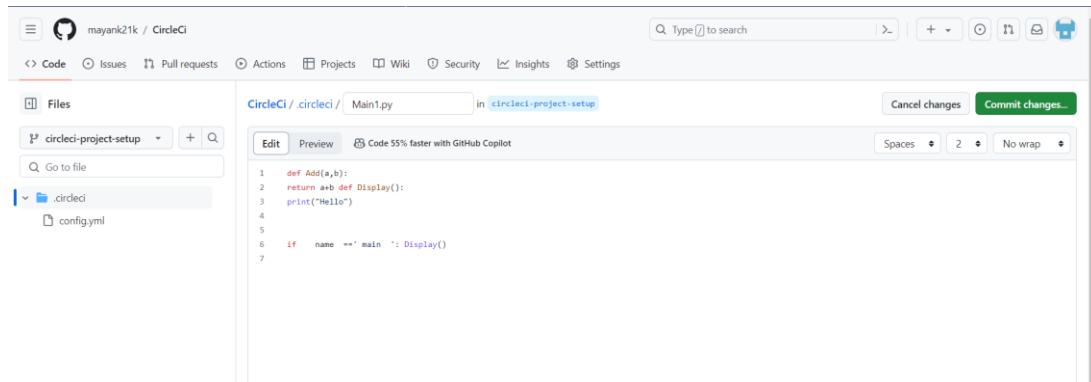
“Main1.py” – filename

Code:-

```
def Add(a,b):
    return a+b

def Display():
    print("Hello")

if __name__=='__main__':
    Display()
```



“Test.py” - filename

Code:-

```
from Main1 import Add

def TestAdd():
    assert Add(3,4)==9
    print("Add Function works correctly")

if __name__=='__main__':
    TestAdd()
```

```

1 from Main1 import Add
2
3
4 def TestAdd():
5     assert Add(3,4)==9
6     print("Add Function works correctly")
7
8 if __name__ == '__main__':
9     main()

```

8) Upload the both file in “CircleCi” repositories (Main1.py and Test.py) and commit changes.

Name	Last commit message	Last commit date
Main1.py	Create Main1.py	1 minute ago
Test.py	Create Test.py	1 minute ago
config.yml	Add .circleci/config.yml	21 minutes ago

9) Write the code as shown below in the existing **config.yml** file in the GitHub repositories in CircleCI.

Code:-

version: 2.1

#Define the jobs we want to run for this project

jobs:

build:

docker:

- image: python

steps:

- checkout

```

    - run: python Main1.py

test:
  docker:
    - image: python

steps:
  - checkout
  - run: python Test.py

# Orchestrate our job run sequence

workflows:
  build_and_test:
    jobs:
      - build
      - test

```

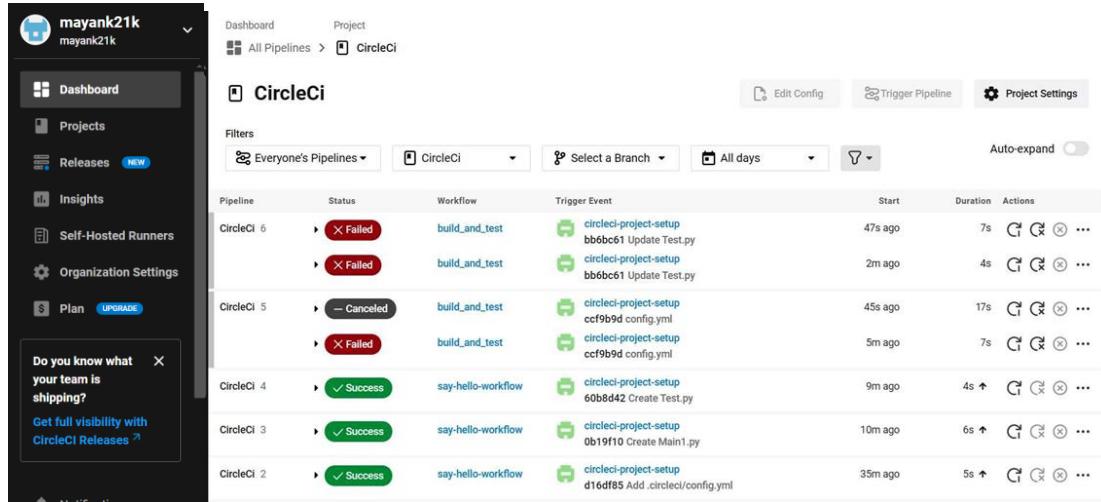
Pipeline	Status	Workflow	Trigger Event	Start	Duration	Actions
CircleCi 5	✗ Failed	build_and_test	circleci-project-setup cc9b9d config.yml	1m ago	7s	↻ ⟳ ✖ ...
CircleCi 4	✓ Success	say-hello-workflow	circleci-project-setup 60b8d42 Create Test.py	5m ago	4s ↑	↻ ⟳ ✖ ...
CircleCi 3	✓ Success	say-hello-workflow	circleci-project-setup 0b19f10 Create Main1.py	6m ago	6s ↑	↻ ⟳ ✖ ...
CircleCi 2	✓ Success	say-hello-workflow	circleci-project-setup d16df85 Add .circleci/config.yml	31m ago	5s ↑	↻ ⟳ ✖ ...
CircleCi 1	✓ Success	say-hello-workflow	circleci-project-setup d16df85	31m ago	4s	↻ ⟳ ✖ ...

10) After making all changes. Check the CircleCi website

11) This time we have done (3,4)==9 It must show failed.

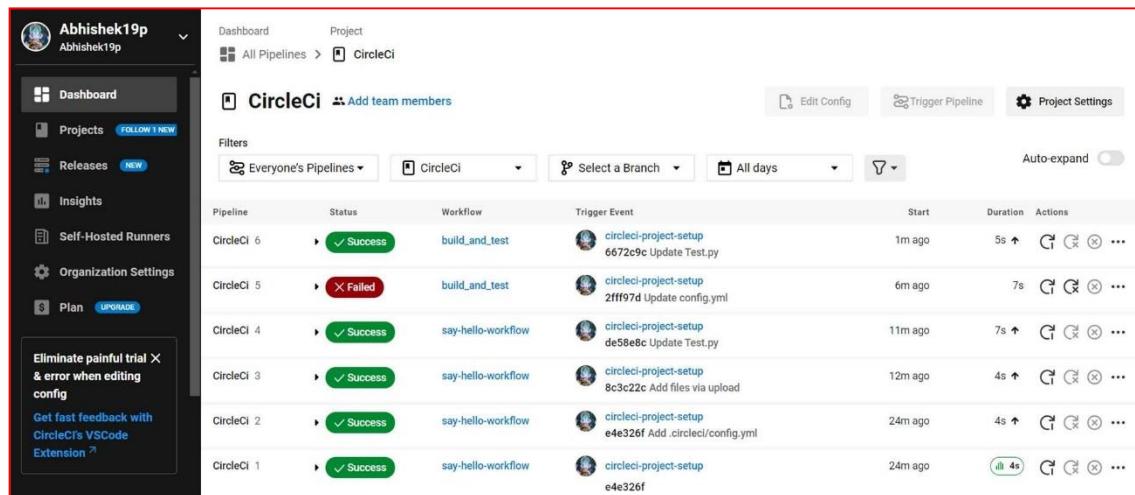
12) Change the code in “Test.py” (3,4)==7 and commit the changes.

13) Go to CircleCi Website and refresh it and it will show successfully done.



The screenshot shows the CircleCI dashboard for the project 'mayank21k'. The sidebar on the left includes 'Dashboard', 'Projects', 'Releases', 'Insights', 'Self-Hosted Runners', 'Organization Settings', and 'Plan' (with an 'UPGRADE' button). A modal window on the left says 'Do you know what your team is shipping? Get full visibility with CircleCI Releases'. The main area displays a table of pipelines:

Pipeline	Status	Workflow	Trigger Event	Start	Duration	Actions
CircleCI 6	✗ Failed	build_and_test	circleci-project-setup bb6bc61 Update Test.py	47s ago	7s	↻ ⟳ ✖ ...
CircleCI 6	✗ Failed	build_and_test	circleci-project-setup bb6bc61 Update Test.py	2m ago	4s	↻ ⟳ ✖ ...
CircleCI 5	✗ Canceled	build_and_test	circleci-project-setup ccf9b9d config.yml	45s ago	17s	↻ ⟳ ✖ ...
CircleCI 5	✗ Failed	build_and_test	circleci-project-setup ccf9b9d config.yml	5m ago	7s	↻ ⟳ ✖ ...
CircleCI 4	✓ Success	say-hello-workflow	circleci-project-setup 60bb842 Create Test.py	9m ago	4s ↑	↻ ⟳ ✖ ...
CircleCI 3	✓ Success	say-hello-workflow	circleci-project-setup 0b19f10 Create Main1.py	10m ago	6s ↑	↻ ⟳ ✖ ...
CircleCI 2	✓ Success	say-hello-workflow	circleci-project-setup d16df85 Add .circleci/config.yml	25m ago	5s ↑	↻ ⟳ ✖ ...



The screenshot shows the CircleCI dashboard for the project 'Abhishek19p'. The sidebar on the left includes 'Dashboard', 'Projects', 'Releases', 'Insights', 'Self-Hosted Runners', 'Organization Settings', and 'Plan' (with an 'UPGRADE' button). A modal window on the left says 'Eliminate painful trial & error when editing config' and 'Get fast feedback with CircleCI's VSCode Extension'. The main area displays a table of pipelines:

Pipeline	Status	Workflow	Trigger Event	Start	Duration	Actions
CircleCI 6	✓ Success	build_and_test	circleci-project-setup 6672c9c Update Test.py	1m ago	5s ↑	↻ ⟳ ✖ ...
CircleCI 5	✗ Failed	build_and_test	circleci-project-setup 2fff97d Update config.yml	6m ago	7s	↻ ⟳ ✖ ...
CircleCI 4	✓ Success	say-hello-workflow	circleci-project-setup de58e8c Update Test.py	11m ago	7s ↑	↻ ⟳ ✖ ...
CircleCI 3	✓ Success	say-hello-workflow	circleci-project-setup 8c3c22c Add files via upload	12m ago	4s ↑	↻ ⟳ ✖ ...
CircleCI 2	✓ Success	say-hello-workflow	circleci-project-setup e4e326f Add .circleci/config.yml	24m ago	4s ↑	↻ ⟳ ✖ ...
CircleCI 1	✓ Success	say-hello-workflow	circleci-project-setup e4e326f	24m ago	4s ↑	↻ ⟳ ✖ ...

Practical 8

AIM:- WORKING WITH KUBERNETS

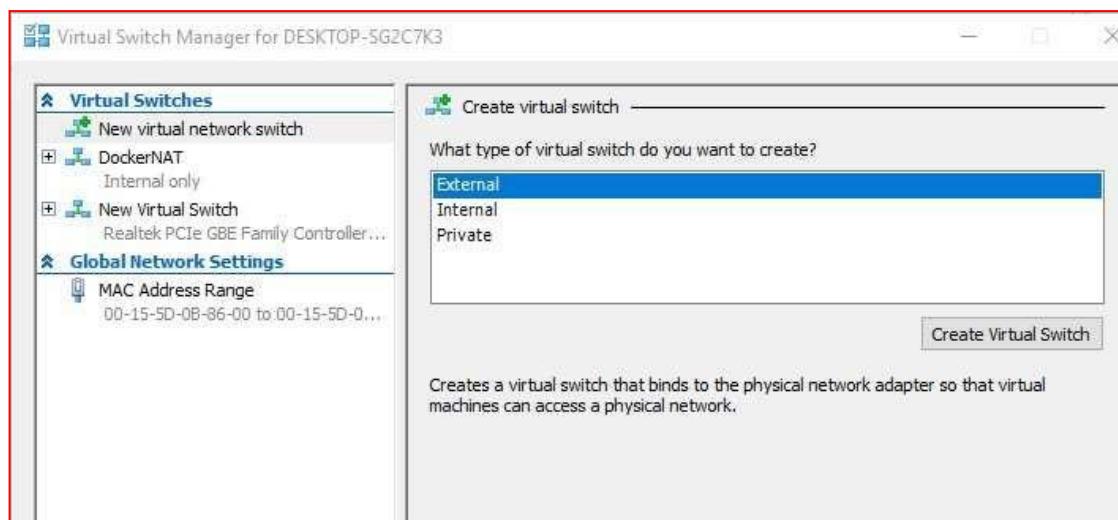
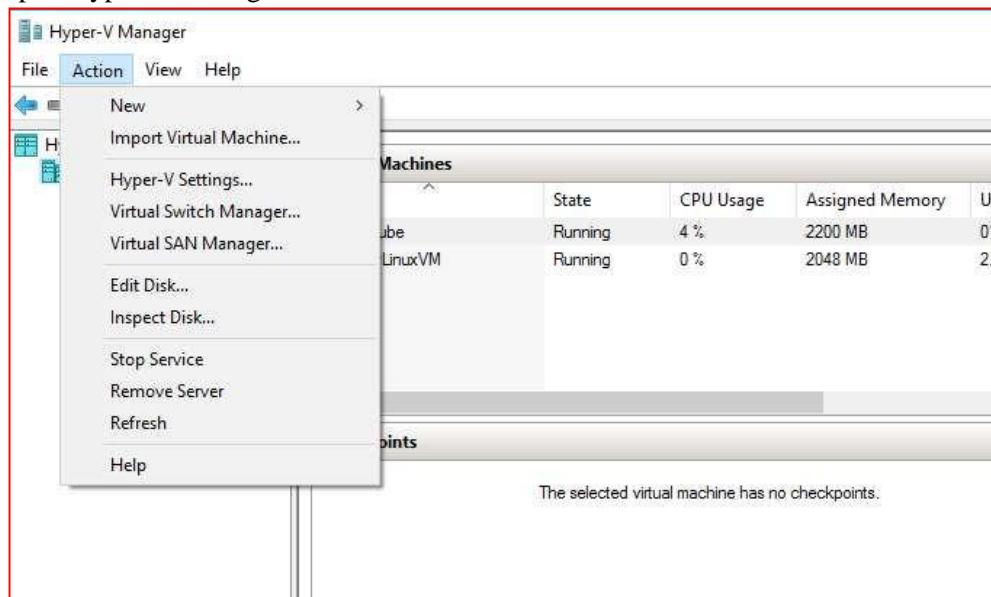
Steps:-

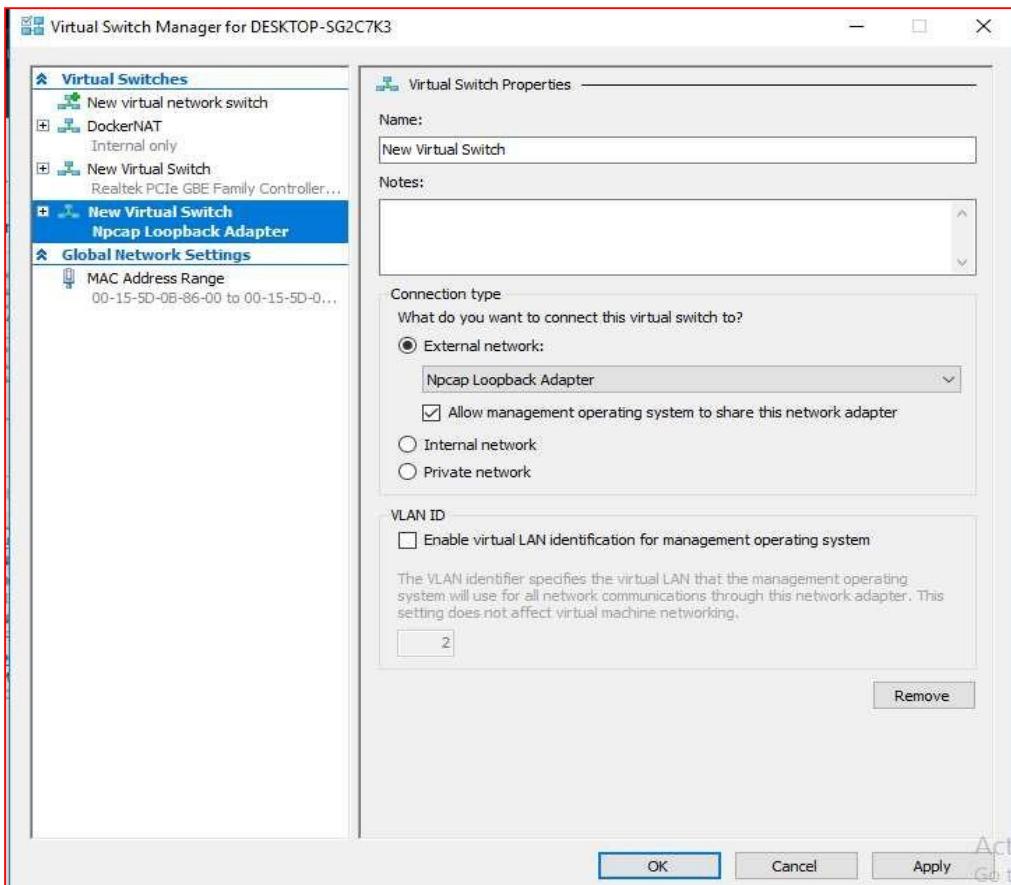
Download kubectl and minikube

Refer the following link for the complete installation process:- <https://iteritory.com/install-minikube-in-windows-10-laptop-step-by-step-tutorial/>

Enter the path in environment variable

Open Hyper-V Manager





5) Make the changes in kubectl-edit-iofgz.yaml
//Nginx.yaml file

```

apiVersion:
  apps/v1 kind:
Deployment
metadata :

  name : nginx-
  deployment labels :

    app :
    nginx spec :
    replicas : 1
    selector :

      matchLabels :

        app : nginx
        template :
        metadata :
        labels :
        app :

```

nginx spec :

containers :

- name : nginx
- image :
- nginx:1.16
- ports :
- containerPort : 80

Open Powershell run it as administrator and type the following commands.

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> minikube start --driver=hyperv
* minikube v1.25.2 on Microsoft Windows 10 Pro 10.0.14393 Build 14393
* Using the hyperv driver based on user configuration
* Starting control plane node minikube in cluster minikube
* Creating hyperv VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
* Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  - kubelet.housekeeping-interval=5m
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

PS C:\Windows\system32> minikube stop
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\Windows\system32> minikube start
* minikube v1.25.2 on Microsoft Windows 10 Pro 10.0.14393 Build 14393
* Using the hyperv driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Restarting existing hyperv VM for "minikube" ...
* Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  - kubelet.housekeeping-interval=5m
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32> kubectl get nodes
NAME     STATUS   ROLES      AGE     VERSION
minikube Ready    control-plane,master  5m58s   v1.23.3
PS C:\Windows\system32> kubectl get pod
No resources found in default namespace.
PS C:\Windows\system32> kubectl get services
NAME        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes  ClusterIP  10.96.0.1   <none>        443/TCP   6m16s
PS C:\Windows\system32> kubectl create deployment nginxdep --image=nginx
deployment.apps/nginxdep created
PS C:\Windows\system32> kubectl create deployment
E0419 22:39:23.138635   3508 run.go:120] "command failed" err="required flag(s) \"image\" not
PS C:\Windows\system32> kubectl get deployment
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginxdep   0/1     1           0           23s
```

```

PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginxdep-5467dbf6d9-6swt5  1/1     Running   0          29s
PS C:\Windows\system32> kubectl get deployment
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
nginxdep        1/1     1           1           37s
PS C:\Windows\system32> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
nginxdep-5467dbf6d9  1         1         1         47s
PS C:\Windows\system32> kubectl edit deployment nginxdep
Error from server (NotFound): deployments.apps "nginxdep" not found
PS C:\Windows\system32> kubectl edit deployment nginxdep
deployment.apps/nginxdep edited
PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginxdep-568789d95-9v895  1/1     Running   0          28s
PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginxdep-568789d95-9v895  1/1     Running   0          39s
PS C:\Windows\system32> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
nginxdep-5467dbf6d9  0         0         0         4m20s
nginxdep-568789d95  1         1         1         50s
PS C:\Windows\system32> kubectl log nginxdep-568789d95-t9pbf
E0419 22:44:26.949348  940 run.go:120] "command failed" err="unknown command \"log\" for \"k
this?\n\ttop\n\tlogs\n\""
PS C:\Windows\system32> kubectl log nginxdep-568789d95
E0419 22:44:36.771320  8216 run.go:120] "command failed" err="unknown command \"log\" for \"k
this?\n\ttop\n\tlogs\n\""
PS C:\Windows\system32> kubectl logs nginxdep-568789d95-t9pbf
Error from server (NotFound): pods "nginxdep-568789d95-t9pbf" not found
PS C:\Windows\system32> kubectl logs nginxdep-568789d95
Error from server (NotFound): pods "nginxdep-568789d95" not found
PS C:\Windows\system32> kubectl logs nginxdep-568789d95-9v895
PS C:\Windows\system32> kubectl create deployment mongodep --image=mongo
deployment.apps/mongodep created
PS C:\Windows\system32> kubectl get deployment
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
mongodep        0/1     1           0           11s
nginxdep        1/1     1           1           6m42s
PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS      RESTARTS   AGE
mongodep-9ccfdfff7-4n8j4  0/1     ContainerCreating   0          19s
nginxdep-568789d95-9v895  1/1     Running     0          3m20s
PS C:\Windows\system32> kubectl logs mongodep-9ccfdfff7-4n8j4
Error from server (BadRequest): container "mongo" in pod "mongodep-9ccfdfff7-4n8j4" is waiting t
g
PS C:\Windows\system32> kubectl get deployment
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
mongodep        0/1     1           0           61s
nginxdep        1/1     1           1           7m32s
PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
mongodep-9ccfdfff7-4n8j4  1/1     Running   0          66s
nginxdep-568789d95-9v895  1/1     Running   0          4m7s

```

```
PS C:\Windows\system32> kubectl logs mongodep-9ccfdfff7-4n8j4
{"t":{"$date":"2022-04-20T05:46:57.283+00:00","s":"I", "c":"NETWORK", "id":4915701, "ctx":"d wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":1}, "outgoing":{"minWireVersion":0,"maxWireVersion":13}}}
 {"t":{"$date":"2022-04-20T05:46:57.291+00:00","s":"I", "c":"CONTROL", "id":23285, "ctx":"I'm disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
 {"t":{"$date":"2022-04-20T05:46:57.328+00:00","s":"W", "c":"ASIO", "id":22601, "ctx":"rtLayer configured during NetworkInterface startup"}
 {"t":{"$date":"2022-04-20T05:46:57.337+00:00","s":"I", "c":"NETWORK", "id":4648601, "ctx":"TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient ."}
 {"t":{"$date":"2022-04-20T05:46:57.415+00:00","s":"W", "c":"ASIO", "id":22601, "ctx":"rtLayer configured during NetworkInterface startup"}
 {"t":{"$date":"2022-04-20T05:46:57.415+00:00","s":"I", "c":"REPL", "id":5123008, "ctx":"I've registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService", "ns":"config"}}
 {"t":{"$date":"2022-04-20T05:46:57.415+00:00","s":"I", "c":"REPL", "id":5123008, "ctx":"I've registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService", "ns":"contents"}}
 {"t":{"$date":"2022-04-20T05:46:57.415+00:00","s":"I", "c":"CONTROL", "id":5945603, "ctx":"ading initialized"}
 {"t":{"$date":"2022-04-20T05:46:57.416+00:00","s":"I", "c":"CONTROL", "id":4615611, "ctx":"oDB starting", "attr":{"pid":1, "port":27017, "dbPath":"/data/db", "architecture":"64-bit", "host":}}
 {"t":{"$date":"2022-04-20T05:46:57.416+00:00","s":"I", "c":"CONTROL", "id":23403, "ctx":"d Info", "attr":{"buildInfo":{"version":"5.0.7", "gitVersion":"b977129dc70eed766cbee7e412d901ee2", "OpenSSL 1.1.1f 31 Mar 2020", "modules":[], "allocator":"tcmalloc", "environment":{"distmod":"ubuntu64", "target_arch":"x86_64"}}}}
 {"t":{"$date":"2022-04-20T05:46:57.416+00:00","s":"I", "c":"CONTROL", "id":51765, "ctx":"ating System", "attr":{"os":{"name":"Ubuntu", "version":"20.04"}}}
 {"t":{"$date":"2022-04-20T05:46:57.416+00:00","s":"I", "c":"CONTROL", "id":21951, "ctx":"ons set by command line", "attr":{"options":{"net":{"bindIp":"*"}}}}
 {"t":{"$date":"2022-04-20T05:46:57.425+00:00","s":"I", "c":"STORAGE", "id":22297, "ctx":"g the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/storage-xfs", "tags":["startupWarnings"]}
 {"t":{"$date":"2022-04-20T05:46:57.425+00:00","s":"I", "c":"STORAGE", "id":22315, "ctx":"ing WiredTiger", "attr":{"config":{"create_cache_size=550M, session_max=33000, eviction=(threads_min=base=false, statistics=(fast), log=(enabled=true, archive=true, path=journal, compressor=snappy), build=(compression_level=6)), file_manager=(close_idle_time=600, close_scan_interval=10, close_handoff=(wait=0), verbose=[recovery_progress, checkpoint_progress, compact_progress],)"}}}
 {"t":{"$date":"2022-04-20T05:46:58.418+00:00","s":"I", "c":"STORAGE", "id":22430, "ctx":"dTiger message", "attr":{"message":"[1650433618:418318][1:0x7f8475f10c80], txn-recover: [WT_VERB recovery timestamp: (0, 0)]"}}
 {"t":{"$date":"2022-04-20T05:46:58.418+00:00","s":"I", "c":"STORAGE", "id":22430, "ctx":}}
 PS C:\Windows\system32> kubectl describe pod mongodep-9ccfdfff7-4n8j4
Name: mongodep-9ccfdfff7-4n8j4
Namespace: default
Priority: 0
Node: minikube/192.168.11.126
Start Time: Tue, 19 Apr 2022 22:45:51 -0700
Labels: app=mongodep
        pod-template-hash=9ccfdfff7
Annotations: <none>
Status: Running
IP: 172.17.0.3
IPs:
  IP: 172.17.0.3
Controlled By: ReplicaSet/mongodep-9ccfdfff7
Containers:
  mongo:
    Container ID: docker://3cc54a29b110a5bbb9d51381842670e24d212aeddea73ce71cba57291ace9ed3
    Image: mongo
    Image ID: docker-pullable://mongo@sha256:1e72fdd16fc769e5200dad77eff5b2316730d42473c281d8192872698e
    Port: <none>
    Host Port: <none>
    State: Running
      Started: Tue, 19 Apr 2022 22:46:53 -0700
    Ready: True
    Restart Count: 0
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6fj9x (ro)
Conditions:
  Type Status
  Initialized True
  Ready True

```

Activate Windows
Go to Settings to activate Windows

```

PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
mongodep-9ccfdfff7-4n8j4  1/1     Running   0          2m36s
nginxdep-568789d95-9v895  1/1     Running   0          5m37s
PS C:\Windows\system32> kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@mongodep-9ccfdfff7-4n8j4:/# ls
bin  dev  home  lib32  media  proc  sbin  tmp
boot  docker-entrypoint-initdb.d  js-yaml.js  lib64  mnt  root  srv  usr
data  etc  lib  libx32  opt  run  sys  var
root@mongodep-9ccfdfff7-4n8j4:/# df -h
Filesystem      Size  Used  Avail Use% Mounted on
overlay        17G   2.7G  14G  17% /
tmpfs          64M     0  64M  0% /dev
tmpfs          1.1G    0  1.1G  0% /sys/fs/cgroup
/dev/sda1       17G   2.7G  14G  17% /data/db
shm             64M     0  64M  0% /dev/shm
tmpfs          2.1G   12K  2.1G  1% /run/secrets/kubernetes.io/serviceaccount
tmpfs          1.1G    0  1.1G  0% /proc/acpi
tmpfs          1.1G    0  1.1G  0% /proc/scsi
tmpfs          1.1G    0  1.1G  0% /sys/firmware
root@mongodep-9ccfdfff7-4n8j4:/# exit
exit
PS C:\Windows\system32> kubectl delete deployment mongodep
deployment.apps "mongodep" deleted
PS C:\Windows\system32> kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginxdep      1/1     1           1           10m
PS C:\Windows\system32> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginxdep-568789d95-9v895  1/1     Running   0          7m32s
PS C:\Windows\system32> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
nginxdep-5467dbf6d9  0         0         0         11m
nginxdep-568789d95  1         1         1         7m47s
PS C:\Windows\system32> kubectl delete deployment nginxdep
deployment.apps "nginxdep" deleted
PS C:\Windows\system32> kubectl get deployment
No resources found in default namespace.
PS C:\Windows\system32> kubectl get pod
No resources found in default namespace.
PS C:\Windows\system32> kubectl get replicaset
No resources found in default namespace.
PS C:\Windows\system32> D:
PS D:> kubectl
kubectl : The term 'kubectl' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ kubectl
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (kubectl:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS D:> minikube
minikube provisions and manages local Kubernetes clusters optimized for
development workflows.

Basic Commands:
  start      Starts a local Kubernetes cluster
  status     Gets the status of a local Kubernetes cluster
  stop       Stops a running local Kubernetes cluster
  delete     Deletes a local Kubernetes cluster
  dashboard  Access the Kubernetes dashboard running within the minikube
  cluster    pause      pause Kubernetes
PS D:> kubectl apply -f nginx_depl_config.yml
error: error parsing nginx_depl_config.yml: error converting YAML to JSON: yaml: line 4: found character that can't be part of any token
PS D:> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment created
PS D:> kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/1     1           1           10s
PS D:> kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/1     1           1           3m54s
PS D:> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment unchanged
PS D:> kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 1/1     1           1           4m12s
PS D:> kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-7956bd8bb9-882vs  1/1     Running   0          4m24s
PS D:> kubectl get replicaset
NAME           DESIRED   CURRENT   READY   AGE
nginx-deployment-7956bd8bb9  1         1         1         5m24s
PS D:> kubectl delete -f nginx_depl_config.yml
deployment.apps "nginx-deployment" deleted
PS D:> kubectl get deployment
No resources found in default namespace.
PS D:> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment created
PS D:> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment unchanged
PS D:> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment unchanged

```

```

PS D:\> kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  1/1     1           1          5s
PS D:\> kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  1/1     1           1          6s
PS D:\> kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  1/1     1           1          7s
PS D:\> kubectl get pod
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-7956bd8bb9-qz9hn  1/1     Running   0          12s
PS D:\> kubectl get pod
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-7956bd8bb9-qz9hn  1/1     Running   0          15s
PS D:\> kubectl get replicaset
NAME          DESIRED  CURRENT  READY   AGE
nginx-deployment-7956bd8bb9  1        1        1      22s
PS D:\> kubectl get replicaset
NAME          DESIRED  CURRENT  READY   AGE
nginx-deployment-7956bd8bb9  1        1        1      24s
PS D:\> kubectl get replicaset
NAME          DESIRED  CURRENT  READY   AGE
nginx-deployment-7956bd8bb9  1        1        1      26s
PS D:\> kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7956bd8bb9-qz9hn  1/1     Running   0          38s
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.96.0.1    <none>       443/TCP   42m
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment  1/1     1           1          38s

PS D:\> kubectl delete deployment nginxdep
Error from server (NotFound): deployments.apps "nginxdep" not found
PS D:\> kubectl delete deployment nginxdep
Error from server (NotFound): deployments.apps "nginxdep" not found
PS D:\> kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7956bd8bb9-qz9hn  1/1     Running   0          108s
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.96.0.1    <none>       443/TCP   43m
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment  1/1     1           1          108s
NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/nginx-deployment-7956bd8bb9  1        1        1      108s
PS D:\> kubectl delete -f nginx_depl_config.yml
deployment.apps "nginx-deployment" deleted
PS D:\> kubectl get all
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.96.0.1    <none>       443/TCP   43m
PS D:\> kubectl apply -f nginx_depl_config.yml
deployment.apps/nginx-deployment created
PS D:\> kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment  1/1     1           1          6m12s
PS D:\> kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
PS D:\> kubectl get all
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.96.0.1    <none>       443/TCP   50m

```

