

Microservice Architecture

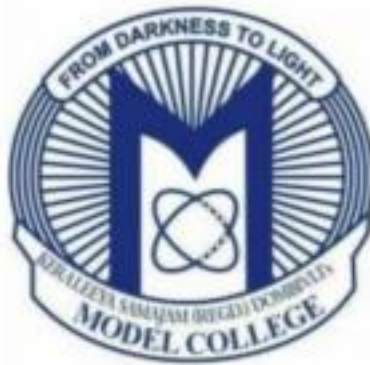
Certified Journal

**Submitted in partial fulfilment of the
Requirements for the award of the Degree of**

**MASTER OF SCIENCE
(INFORMATION TECHNOLOGY)**

By

Anjali Rameshwar Nimje



DEPARTMENT OF INFORMATION TECHNOLOGY

**KERALEEYA SAMAJAM (REGD.) DOMBIVLI'S
MODEL COLLEGE (AUTONOMOUS)**

Re-Accredited 'A' Grade by NAAC

(Affiliated to University of Mumbai)

FOR THE YEAR

(2022-23)



Keraleeya Samajam(Regd.) Dombivli's

MODEL COLLEGE

Re-Accredited Grade "A" by NAAC

Kanchan Goan Village, Khambalpada, Thakurli East – 421201
Contact No – 7045682157, 7045682158. www.model-college.edu.in

**DEPARTMENT OF INFORMATION TECHNOLOGY
AND COMPUTER SCIENCE**

CERTIFICATE

This is to certify that Mr. /Miss _____

Studying in Class _____ Seat No. _____

Has completed the prescribed practicals in the subject _____

During the academic year _____

Date : _____

External Examiner

Internal Examiner
M.Sc. Information Technology

INDEX

Practical No	Title	Date	Signature
1	Installation of .Net SDK and Building first console App	1 st April 2023	
2	Building Asp.net core MVC application	15 th April 2023	
3	Building asp .net core rest API	06 th May 2023	
4	Working with docker images and containers	15 th May 2023	
5	Working with docker volume	02 nd May 2023	

Practical No.1:

Installation of .Net SDK and Building first console App

Overview of .Net SDK

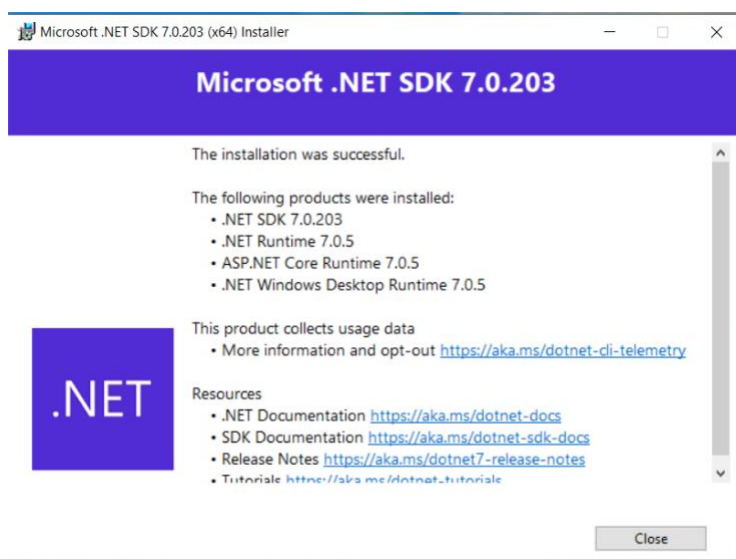
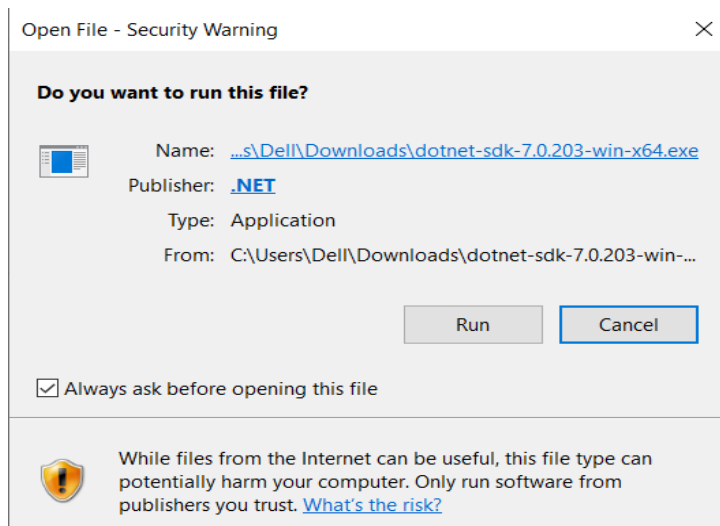
The .NET SDK(Software Development Kit)

1. The .NET SDK is a set of libraries and tools that allow developers to create .NET applications and libraries.
2. It contains the following components that are used to build and run applications.
 - The .NET CLI
 - .NET libraries and runtime
 - The dotnet driver

STEPS:

Step 1: Download and install

To start building .NET apps download and install the .NET SDK (SOFTWARE DEVELOPMENT KIT).

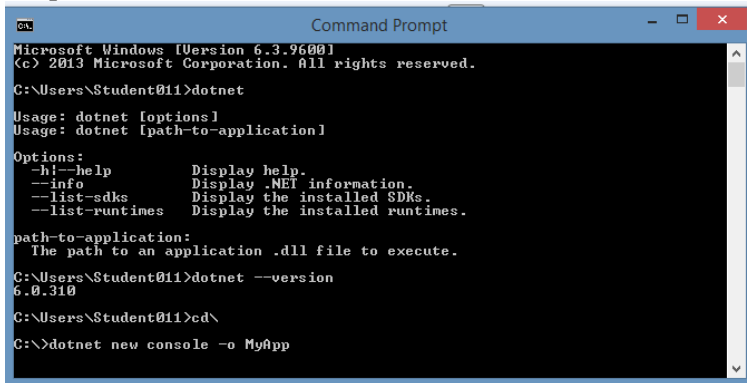


Step 2: Checking everything installed correctly

After installation is done open a new terminal and type the

Command : dotnet

Output



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student011>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.

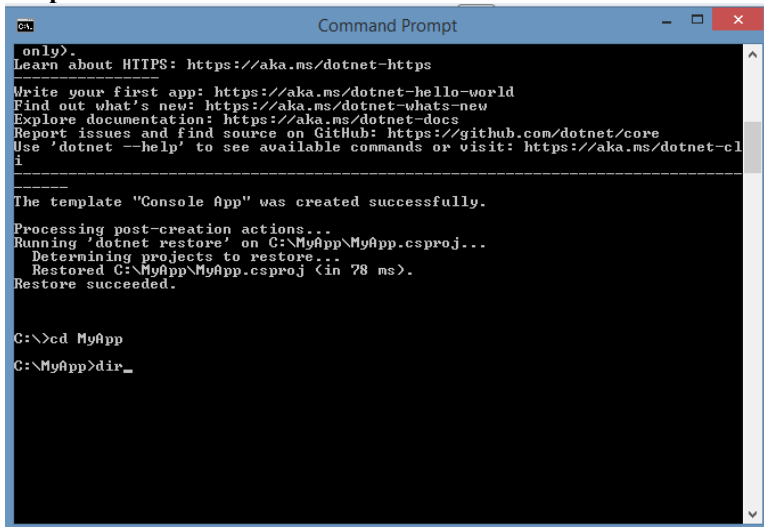
C:\Users\Student011>dotnet --version
6.0.310

C:\Users\Student011>cd\

C:\>dotnet new console -o MyApp
```

Step 3: Change the working directory and create the console app with name MyApp.

Output



```
only>.
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-cl
i

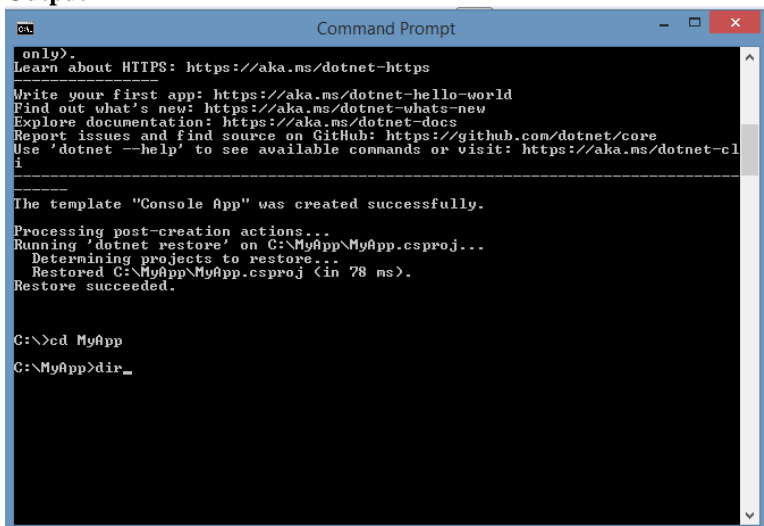
-----
The template "Console App" was created successfully.
Processing post-creation actions...
Running 'dotnet restore' on C:\MyApp\MyApp.csproj...
  Determining projects to restore...
  Restored C:\MyApp\MyApp.csproj (in 78 ms).
Restore succeeded.

C:\>cd MyApp
C:\MyApp>dir_
```

Step 4: Changing the directory to MyApp

Command : cd MyApp

Output



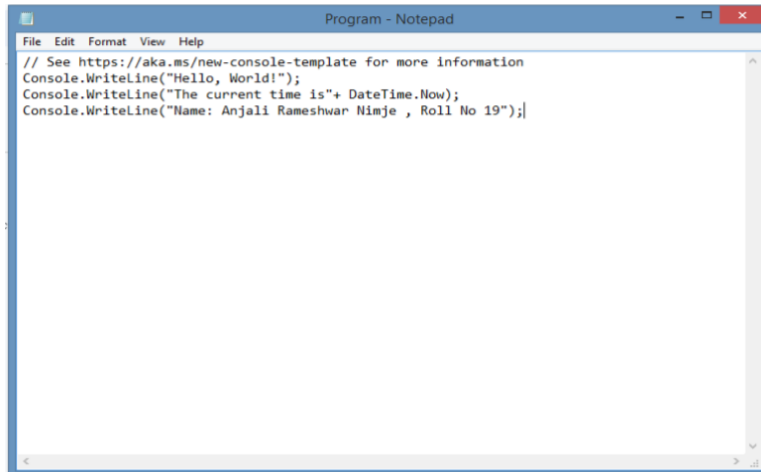
```
only>.
Learn about HTTPS: https://aka.ms/dotnet-https

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-cl
i

-----
The template "Console App" was created successfully.
Processing post-creation actions...
Running 'dotnet restore' on C:\MyApp\MyApp.csproj...
  Determining projects to restore...
  Restored C:\MyApp\MyApp.csproj (in 78 ms).
Restore succeeded.

C:\>cd MyApp
C:\MyApp>dir_
```

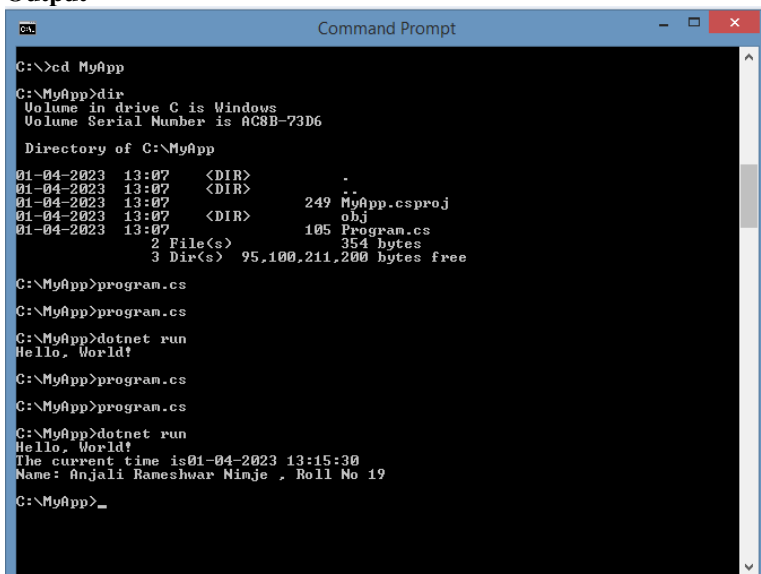
Step 5: Write code in program.cs file



Step 6 : Run the program by typing
Command : C:\MyApp>dotnet run

Step 7 : Output of the program.cs file

Output



Practical No.2: Building ASP.NET core MVC Application

Overview of MVC

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating MVC-based Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership- based authentication. The MVC framework is defined in the System.Web.Mvc namespace and is a fundamental, supported part of the System. Web namespace.

MVC is a standard design pattern that many developers are familiar with. Some types of Web applications will benefit from the MVC framework. Others will continue to use the traditional ASP.NET application pattern that is based on Web Forms and post backs. Other types of Web applications will combine the two approaches; neither approach excludes the other.

The MVC framework includes the following components:-

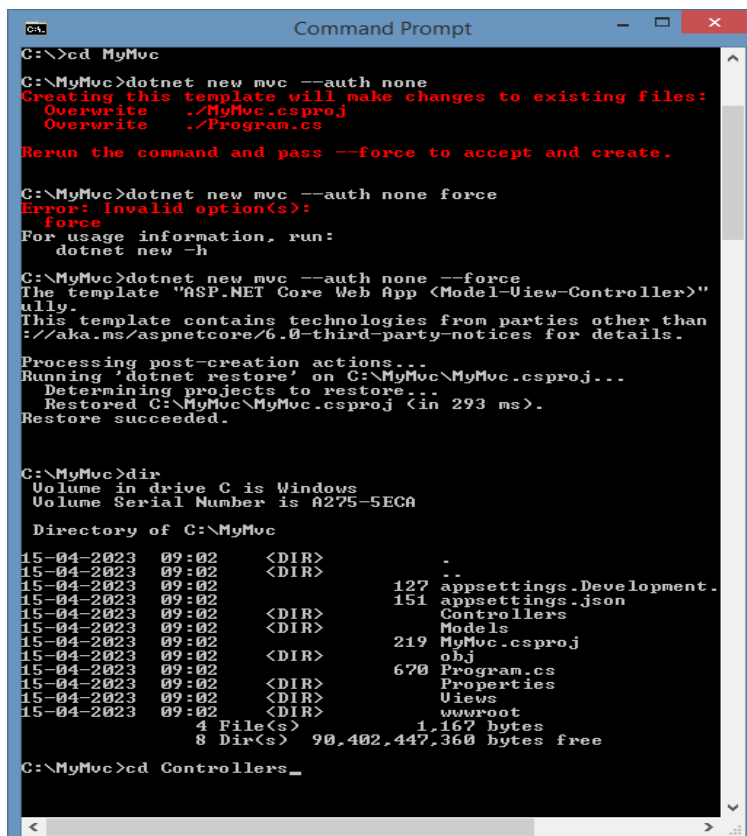
Models :- Model objects are the parts of the application that implement the logic for the applications data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in SQL Server.

Views :- Views are the components that display the applications user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Products object.

Controllers :- Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn queries the database by using the values.

- 1.Install.NetCoreSDK.
 - 2.Create Folder MyMVC folder in d:drive or any other drive.
 - 3.Open command prompt and perform the following operations. Command:-to create mvc project.
- Command :dotnet new mvc--auth none**

Output



```
C:\>cd MyMvc
C:\MyMvc>dotnet new mvc --auth none
Creating this template will make changes to existing files:
  Overwrite ./MyMvc.csproj
  Overwrite ./Program.cs
Rerun the command and pass --force to accept and create.

C:\MyMvc>dotnet new mvc --auth none force
Error: Invalid option(s):
  force
For usage information, run:
  dotnet new -h

C:\MyMvc>dotnet new mvc --auth none --force
The template "ASP.NET Core Web App (Model-View-Controller)"
successfully.
This template contains technologies from parties other than
Microsoft:
  - /aka.ms/aspnetcore/6.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on C:\MyMvc\MyMvc.csproj...
  Determining projects to restore...
  Restored C:\MyMvc\MyMvc.csproj (in 293 ms).
Restore succeeded.

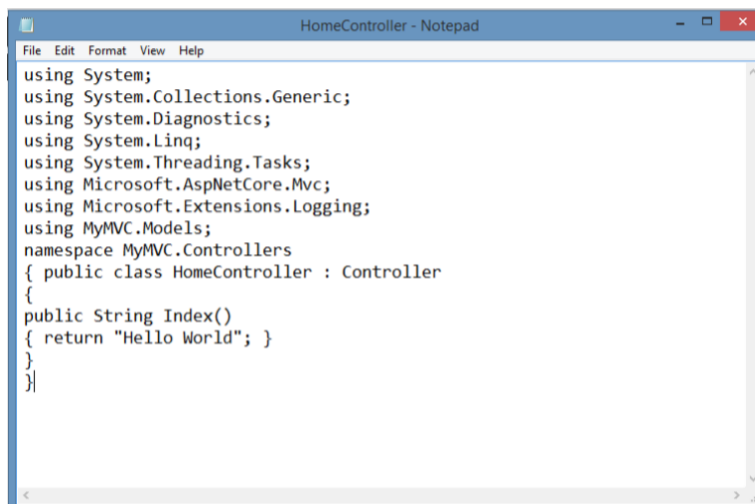
C:\MyMvc>dir
Volume in drive C is Windows
Volume Serial Number is A275-5ECA

Directory of C:\MyMvc

15-04-2023 09:02 <DIR>          .
15-04-2023 09:02 <DIR>          ..
15-04-2023 09:02          127 appsettings.Development.
15-04-2023 09:02          151 appsettings.json
15-04-2023 09:02 <DIR>          Controllers
15-04-2023 09:02 <DIR>          Models
15-04-2023 09:02          219 MyMvc.csproj
15-04-2023 09:02 <DIR>          obj
15-04-2023 09:02          670 Program.cs
15-04-2023 09:02 <DIR>          Properties
15-04-2023 09:02 <DIR>          Views
15-04-2023 09:02 <DIR>          wwwroot
               4 File(s)          1,167 bytes
               8 Dir(s)      90,402,447,360 bytes free

C:\MyMvc>cd Controllers_
```

4: go to controllers folder and modify HomeController.cs file to match following



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



Output




```
CA. Command Prompt - dotnet run
public HomeController(ILogger<HomeController> logger)
{
    _logger = logger;
}

public IActionResult Index()
{
    return View();
}

public IActionResult Privacy()
{
    return View();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLoc
= true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activit
pContext.TraceIdentifier });
}
}

C:\MyMvc\Controllers>HomeController.cs
C:\MyMvc\Controllers>cd..
C:\MyMvc>dotnet run
Building...
C:\MyMvc\Controllers\HomeController.cs(8,13): error CS0234:
ce name 'Models' does not exist in the namespace 'MyMVC' (ar
sembly reference?) [C:\MyMvc\MyMvc.csproj]

The build failed. Fix the build errors and run again.

C:\MyMvc>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
Now listening on: https://localhost:7256
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5231
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:\MyMvc\

```

```
CA. Command Prompt
C:\MyMvc\Controllers>edit HomeController.cs
'edit' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMvc\Controllers>type HomeController.cs
using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using MyMvc.Models;

namespace MyMvc.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

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

    public IActionResult Index()
    {
        return View();
    }

    public IActionResult Privacy()
    {
        return View();
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLoc
= true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activit
pContext.TraceIdentifier });
    }
}

C:\MyMvc\Controllers>HomeController.cs
C:\MyMvc\Controllers>cd..

```

```
Command Prompt
C:\MyMvc\Controllers>edit HomeController.cs
'edit' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMvc\Controllers>type HomeController.cs
@using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using MyMvc.Models;

namespace MyMvc.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

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

    public IActionResult Index()
    {
        return View();
    }

    public IActionResult Privacy()
    {
        return View();
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLoc
= true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activit
pContext.TraceIdentifier });
    }
}

C:\MyMvc\Controllers>HomeController.cs
C:\MyMvc\Controllers>cd..
```

```
Command Prompt
C:\MyMvc\Controllers>edit HomeController.cs
'edit' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMvc\Controllers>type HomeController.cs
@using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using MyMvc.Models;

namespace MyMvc.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

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

    public IActionResult Index()
    {
        return View();
    }

    public IActionResult Privacy()
    {
        return View();
    }

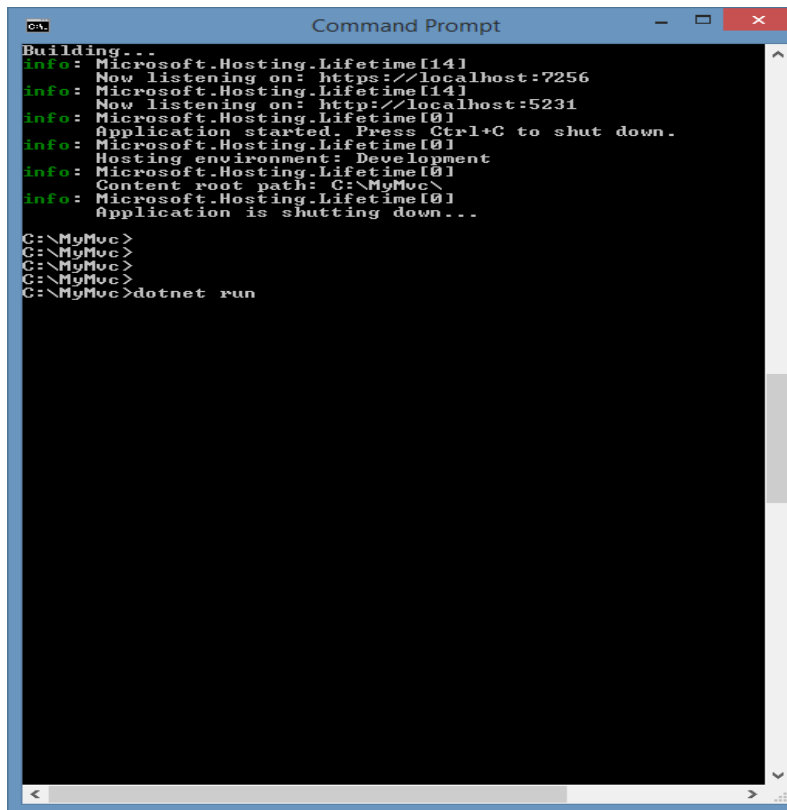
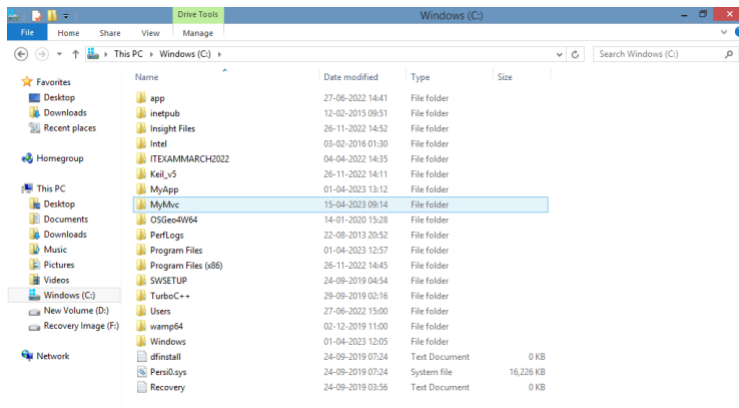
    [ResponseCache(Duration = 0, Location = ResponseCacheLoc
= true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activit
pContext.TraceIdentifier });
    }
}

C:\MyMvc\Controllers>HomeController.cs
C:\MyMvc\Controllers>cd..

C:\MyMvc>dotnet run
Building...
C:\MyMvc>dotnet run
C:\MyMvc\Controllers\HomeController.cs(8,13): error CS0234:
ce name 'Models' does not exist in the namespace 'MyMVC' (ar
sembly reference?) [C:\MyMvc\MyMvc.csproj]

The build failed. Fix the build errors and run again.

C:\MyMvc>dotnet run
```



Output

```

C:\. Command Prompt - dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7256
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5231
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:\MyMvc\
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...

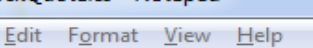
C:\MyMvc>
C:\MyMvc>
C:\MyMvc>
C:\MyMvc>dotnet run
Building...
C:\MyMvc>Controllers\HomeController.cs(14,35): error CS1012:
s in character literal [C:\MyMvc\MyMvc.csproj]

The build failed. Fix the build errors and run again.

C:\MyMvc>dotnet run
Building...
C:\MyMvc>Models\StockQuote.cs(5,17): warning CS8618: Non-nul
bol' must contain a non-null value when exiting constructor.
the property as nullable. [C:\MyMvc\MyMvc.csproj]
C:\MyMvc>Controllers\HomeController.cs(12,37): warning CS199
d lacks 'await' operators and will run synchronously. Consid
' operator to await non-blocking API calls, or 'await Task.R
bound work on a background thread. [C:\MyMvc\MyMvc.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7256
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5231
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:\MyMvc\

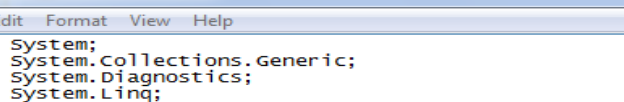
```

5. Create one notepad and save the file name as **StockQuote.cs** and make some changes. Save that file into **Models** folder. Make some changes in **Index.cshtml** file. This file is being located at- (“D:\MyMVC\Views\Home”



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

6. Again go to controllers folder and modify Home Controller.cd file to match following



```

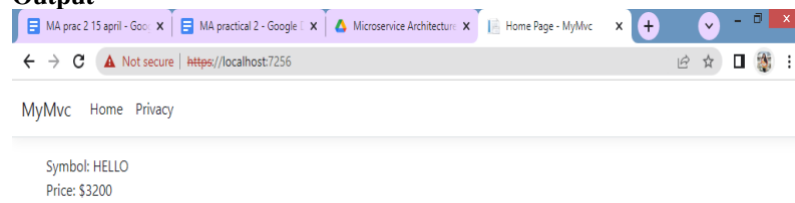
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;
namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public async Task <ActionResult> Index()
        {
            var model= new StockQuote{ Symbol="HELLO", Pice=3200};
            return View(model);
        }
    }
}

```

```
Index.cshtml - Notepad
File Edit Format View Help
@{
    ViewData["Title"] = "Home Page";
}

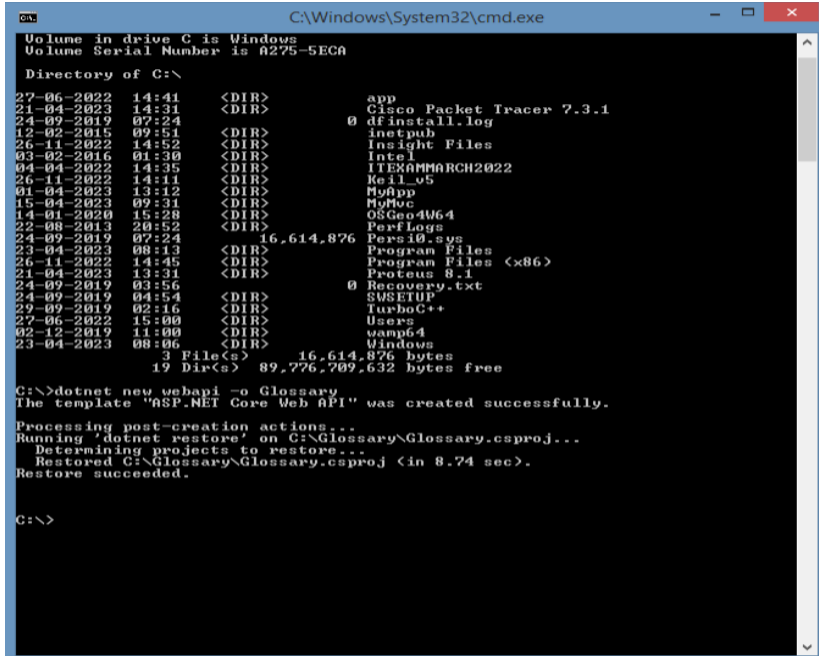
<div>
Symbol: @Model.Symbol <br/>
Price: $@Model.Price <br/>
</div>
```

Output



Practical No. 3: Building asp .net core rest API

Step1:(If Dot net is already installed) Open Command Prompt and run
Command: 'dotnet new web api -o Glossary'



```
C:\Windows\System32\cmd.exe
Volume in drive C is Windows
Volume Serial Number is A275-5ECA

Directory of C:\

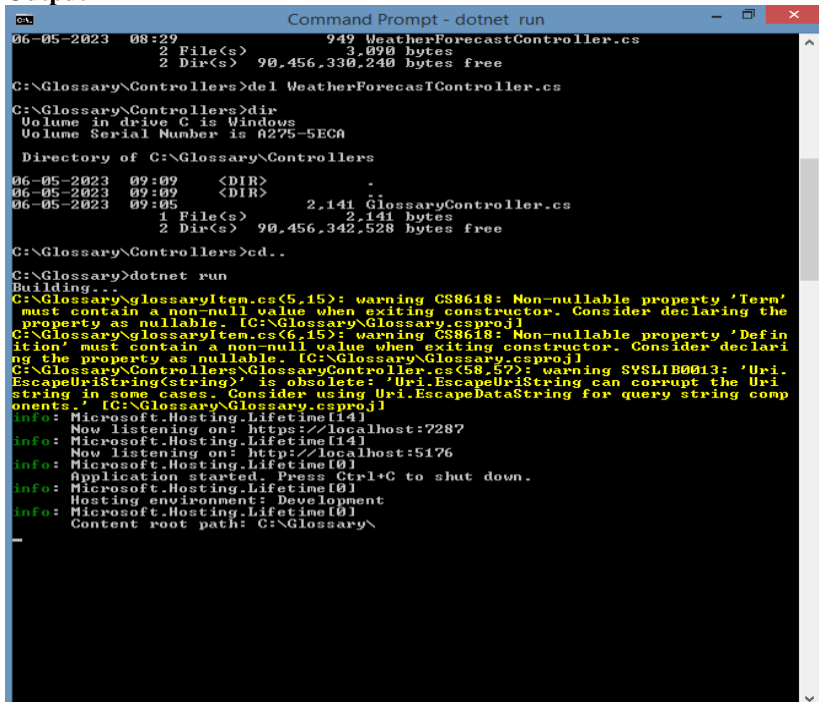
27-06-2022 14:41 <DIR>          app
21-04-2023 14:31 <DIR>          Cisco Packet Tracer 7.3.1
24-09-2019 07:24 <DIR>          0 dfinstall.log
12-02-2015 09:51 <DIR>          inetpub
26-11-2022 14:52 <DIR>          insight Files
03-02-2016 01:30 <DIR>          Intel
04-04-2022 14:35 <DIR>          ITEXAMMARCH2022
26-11-2022 14:11 <DIR>          Keil_v5
01-04-2023 13:12 <DIR>          MyApp
15-04-2023 09:31 <DIR>          MyMvc
14-01-2020 15:28 <DIR>          OSGeo4W64
22-08-2013 20:52 <DIR>          PerfLogs
24-09-2019 07:24 <DIR>          16.614.876 Perfid.sys
23-04-2023 08:13 <DIR>          Program Files
26-11-2022 14:45 <DIR>          Program Files (x86)
21-04-2023 13:31 <DIR>          Proteus 8.1
24-09-2019 03:56 <DIR>          0 Recovery.txt
24-09-2019 04:54 <DIR>          SWSETUP
29-09-2019 02:16 <DIR>          TurboC++
27-06-2022 15:00 <DIR>          Users
02-12-2019 11:00 <DIR>          wamp64
23-04-2023 08:06 <DIR>          Windows
3 File(s)          16.614.876 bytes
19 Dir(s)          89,776,709,632 bytes free

C:\>dotnet new webapi -o Glossary
The template "ASP.NET Core Web API" was created successfully.
Processing post-creation actions...
Running 'dotnet restore' on C:\Glossary\Glossary.csproj...
Determining projects to restore...
Restored C:\Glossary\Glossary.csproj (in 8.74 sec).
Restore succeeded.

C:\>
```

Step 2:open the glossary folder command: 'cd glossary' then run
Command:'dotnet run

Output



```
C:\Windows\System32\cmd.exe
06-05-2023 08:29 949 WeatherForecastController.cs
2 File(s) 3,090 bytes
2 Dir(s) 90,456,330,240 bytes free

C:\Glossary\Controllers>del WeatherForecastController.cs
C:\Glossary\Controllers>dir
Volume in drive C is Windows
Volume Serial Number is A275-5ECA

Directory of C:\Glossary\Controllers

06-05-2023 09:09 <DIR>          .
06-05-2023 09:09 <DIR>          ..
06-05-2023 09:05 2,141 GlossaryController.cs
1 File(s) 2,141 bytes
2 Dir(s) 90,456,342,528 bytes free

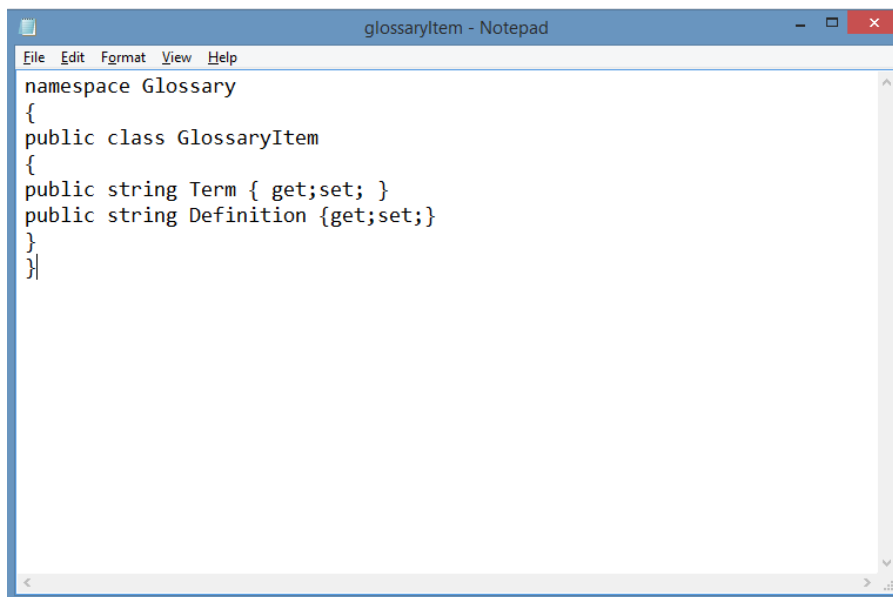
C:\Glossary\Controllers>cd..
C:\Glossary>dotnet run
Building...
C:\Glossary\glossaryItem.cs(5,15): warning CS8618: Non-nullable property 'Term' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Glossary\Glossary.csproj]
C:\Glossary\glossaryItem.cs(6,15): warning CS8618: Non-nullable property 'Definition' must contain a non-null value when exiting constructor. Consider declaring the property as nullable. [C:\Glossary\Glossary.csproj]
C:\Glossary\Controllers\GlossaryController.cs(58,57): 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.' [C:\Glossary\Glossary.csproj]
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7287
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5176
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:\Glossary\
-
```

Step3:test weatherForecast class

Step4: Delete WeatherForecast.cs and WeatherForecastController.cs file, create a new file glossaryitem.cs file and edit it.

Code:

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



Step5: Edit the Glossary Controller.cs file in 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();
}
}
}
}
}

```

Step 6: Now again run

Command: dotnet run in Glossary folder

Output

```

C:\Windows\System32\cmd.exe - dotnet run
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>dotnet new webapi -o Glossary
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\Glossary\Glossary.csproj...
  Determining projects to restore...
  Restored C:\Glossary\Glossary.csproj (in 243 ms).
Restore succeeded.

C:\>cd Glossary

C:\Glossary>dir
Volume in drive C is Windows
Volume Serial Number is A275-5ECA

Directory of C:\Glossary

06-05-2023  09:40    <DIR>          .
06-05-2023  09:40    <DIR>          ..
06-05-2023  09:40                127 appsettings.Development.json
06-05-2023  09:40                151 appsettings.json
06-05-2023  09:40    <DIR>          Controllers
06-05-2023  09:40                327 Glossary.csproj
06-05-2023  09:40    <DIR>          obj
06-05-2023  09:40                557 Program.cs
06-05-2023  09:40    <DIR>          Properties
06-05-2023  09:40                257 WeatherForecast.cs
               5 File(s)          1,419 bytes
               5 Dir(s)  90,443,149,312 bytes free

C:\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[141]
      Now listening on: https://localhost:7093
info: Microsoft.Hosting.Lifetime[141]
      Now listening on: http://localhost:5134
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:\Glossary\
-

```

Step 7 :

- 1)Getting a list of items:
- 2)Getting a single item
- 3)Creating an item
- 4)Update Item
- 5)Delete Item

Each new command is followed respectively

Output

```
CA C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>curl --insecure https://localhost:5001/api/glossary
curl: (7) Failed to connect to localhost port 5001 after 1203 ms: Couldn't connect to server

C:\>curl --insecure https://localhost:7240/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"},{"term":"MUC","definition":"Model View Controller"},{"term":"MFA","definition":"An authentication process."}]
C:\>curl --insecure https://localhost:7240/api/glossary/MUC
{"term":"MUC","definition":"Model View Controller"}
C:\>curl --insecure -X POST -d '{"term":"MFA","definition":"An authentication process."}' -H "Content-Type:application/json" https://localhost:7240/api/glossary
Cannot create the term because it already exists.
C:\>curl --insecure -X PUT -d '{"term":"MUC","definition":"Modified record of Model View Controller."}' -H "Content-Type:application/json" https://localhost:7240/api/glossary

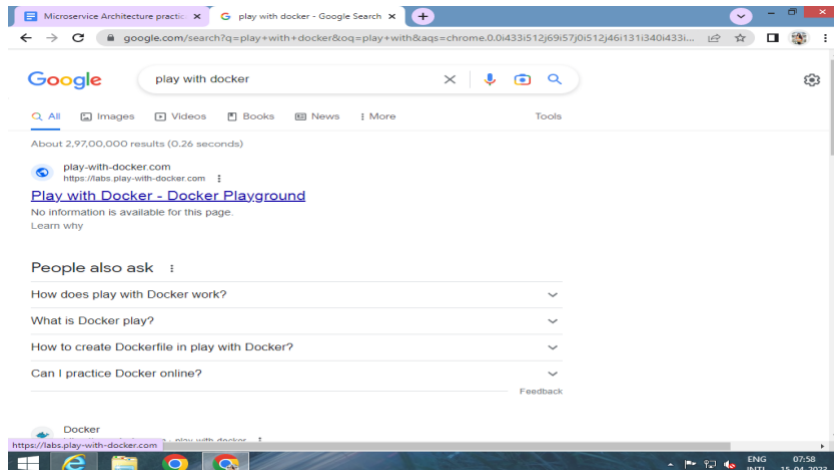
C:\>curl --insecure --request DELETE --url https://localhost:5001/api/glossary/oid
curl: (7) Failed to connect to localhost port 5001 after 1201 ms: Couldn't connect to server

C:\>curl --insecure --request DELETE --url https://localhost:7240/api/glossary/oid
{"type":"https://tools.ietf.org/html/rfc7231#section-6.5.4","title":"Not Found","status":404,"traceId":"00-2af8e9fcc152f5e9066db925df0e13-845dia29debf59cc-00"}
C:\>
```

Practical No.4: Working with Docker, Docker images and containers

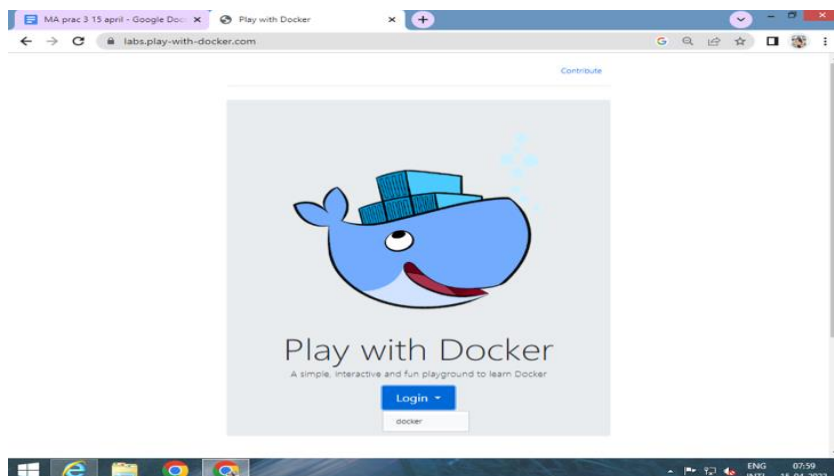
Overview of Docker

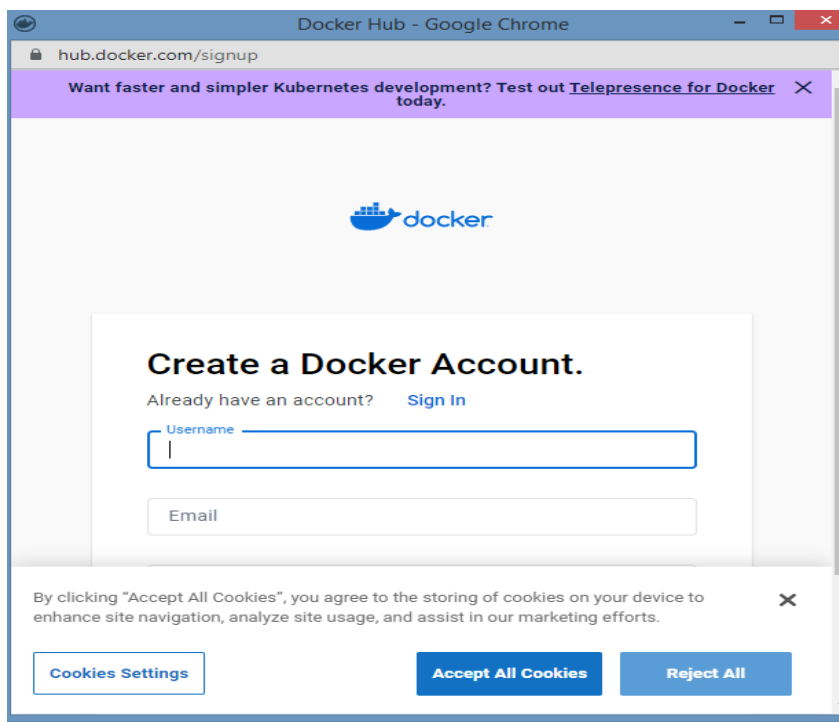
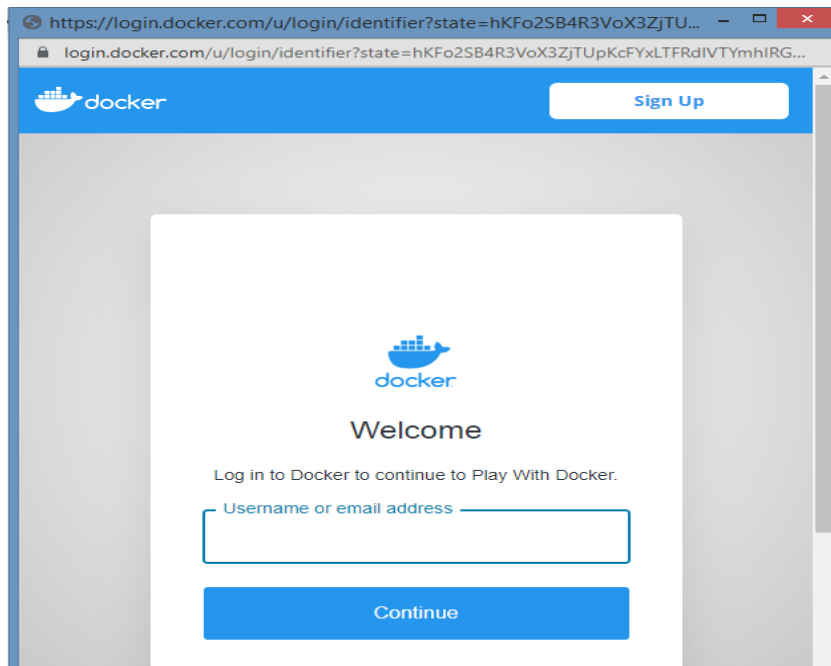
Docker is an open platform for developing ,shipping ,and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker , you can manage your infrastructure in the same ways you manage your applications .By taking advantage of Docker's methodologies for shipping , testing, and deploying code quickly ,you can significantly reduce the delay between writing code and running it in production



1.Create docker hub account (signup)

2.Login to <https://labs.play-with-docker.com/>





Docker Hub - Google Chrome

hub.docker.com/signup

Create a Docker Account.

Already have an account? [Sign In](#)

Username
anjali22001

Email
anjali22001@gmail.com

Password
.....

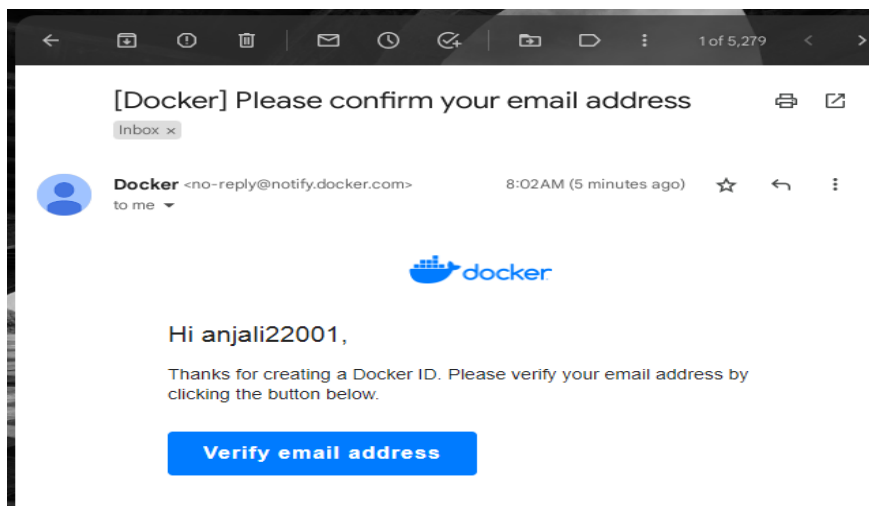
☒ Send me occasional product updates and announcements.

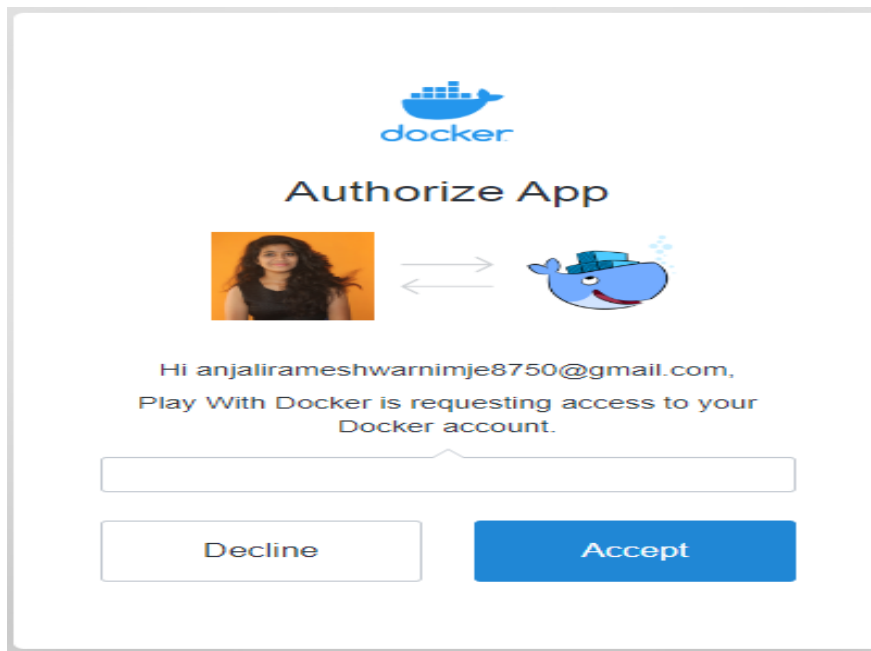
☒ I agree to the [Subscription Service Agreement](#), [Privacy Policy](#), and [Data Processing Terms](#).

☒ I'm not a robot

reCAPTCHA

Sign Up

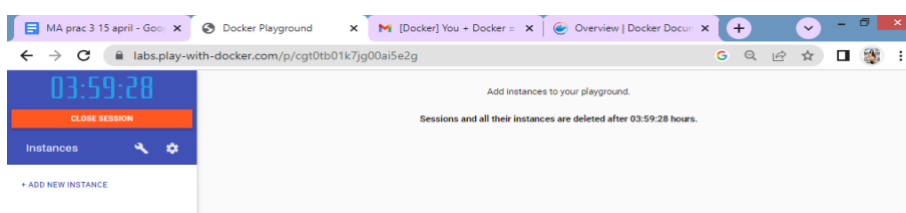


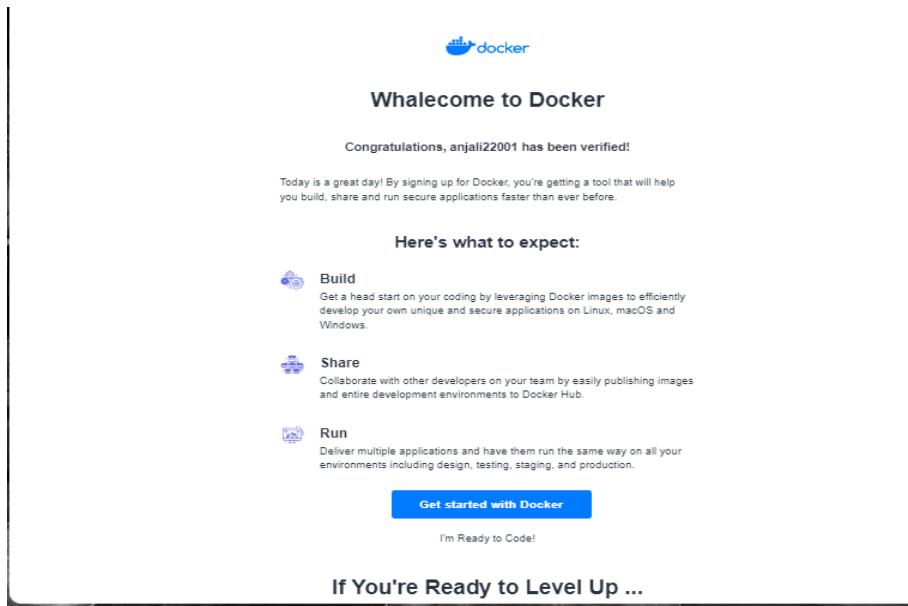


Click on start



3: Create new instance





4.Perform following

Method1 :To pull and push images using docker

Command: to check docker version **docker --version**

Command: to pull ready made image **docker pull rocker/verse**

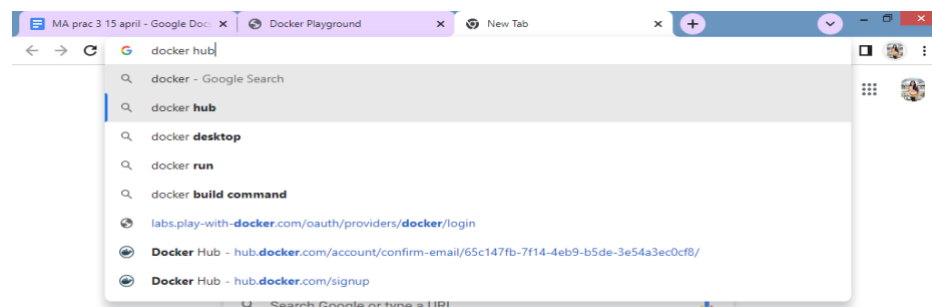
Command: to check images in docker **docker images**

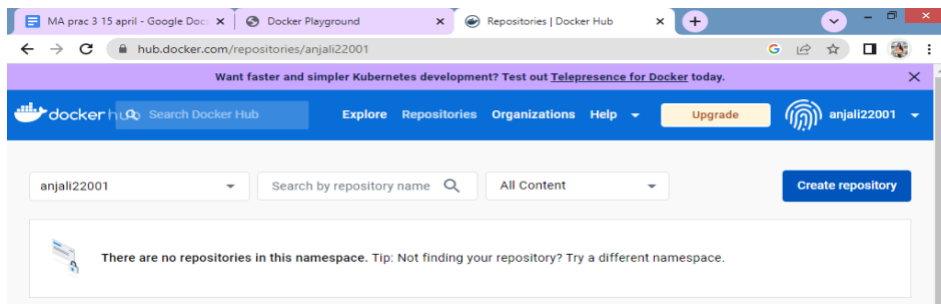
Output

```

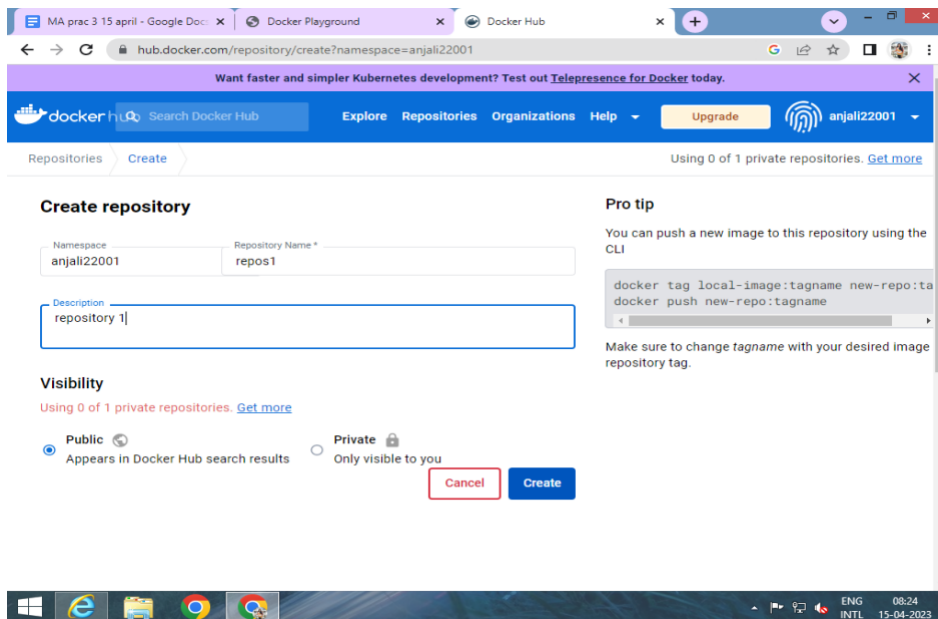
##### WARNING!!!! #####
# This is a sandbox environment. Using personal credentials #
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
# #
# The FWD team. #
#####

[macd1] (local) root@192.168.0.28 ~
$ docker --version
Docker version 20.10.17, build 100c701
[macd1] (local) root@192.168.0.28 ~
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
2ab09b027e7f: Pull complete
1875e7997572: Pull complete
453813be1883: Pull complete
5132964ecc30: Pull complete
f9ae60b737bd: Pull complete
cf514b297e6d: Pull complete
0da1c500d44b: Pull complete
16bff0b45131: Pull complete
719977ec8b94: Pull complete
558a8d6d50e1: Pull complete
Digest: sha256:a98aeca7e79700121b86ba4e7031d17561a209ee10ae48f84ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
[macd1] (local) root@192.168.0.28 ~
$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED        SIZE
rocker/verse        latest         7291950d643e   13 days ago   3.41GB
[macd1] (local) root@192.168.0.28 ~
$
  
```





Now Login to docker hub and create repository



Want faster and simpler Kubernetes development? Test out [Telepresence for Docker](#) today.

Search Docker Hub Explore Repositories Organizations Help Upgrade anjali22001

Repositories Create Using 0 of 1 private repositories. [Get more](#)

Create repository

Namespace: anjali22001 Repository Name: repos1

Description: repository 1

Visibility
Using 0 of 1 private repositories. [Get more](#)

☒ Public Appears in Docker Hub search results ☐ Private Only visible to you

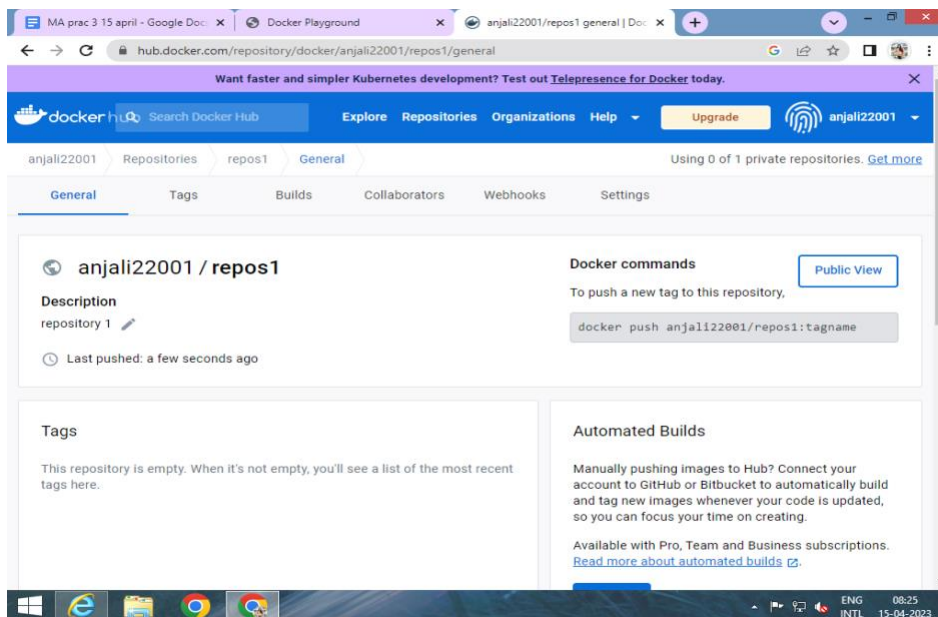
[Cancel](#) [Create](#)

Pro tip
You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

Click on Create button Now check repository created



Want faster and simpler Kubernetes development? Test out [Telepresence for Docker](#) today.

Search Docker Hub Explore Repositories Organizations Help Upgrade anjali22001

anjali22001 Repositories repos1 General Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Collaborators Webhooks Settings

anjali22001 / repos1

Description
repository 1

Last pushed: a few seconds ago

Docker commands [Public View](#)
To push a new tag to this repository.

```
docker push anjali22001/repos1:tagname
```

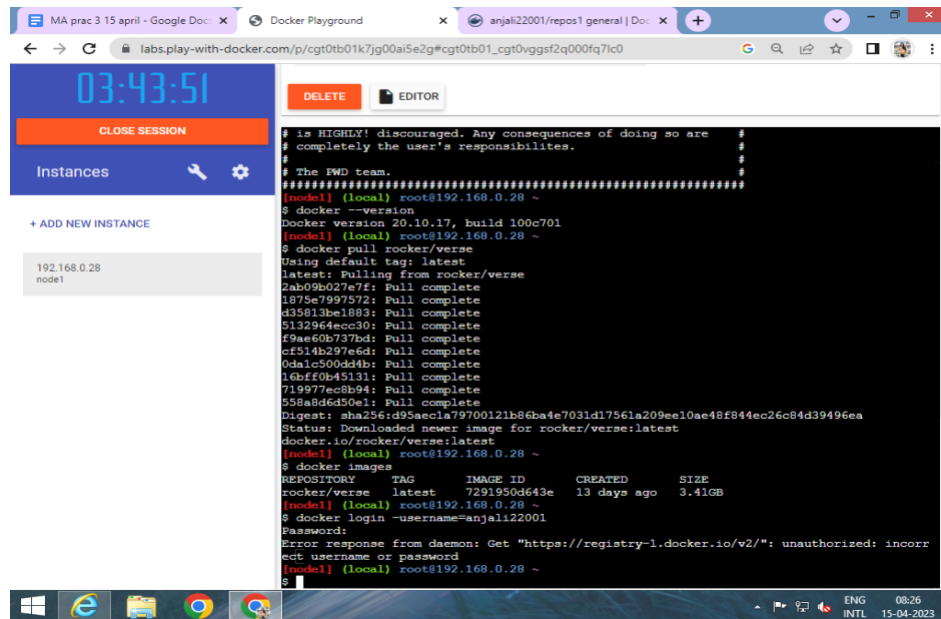
Tags
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

Command: to login to your docker account **docker login--username=anjali22001**
Password:

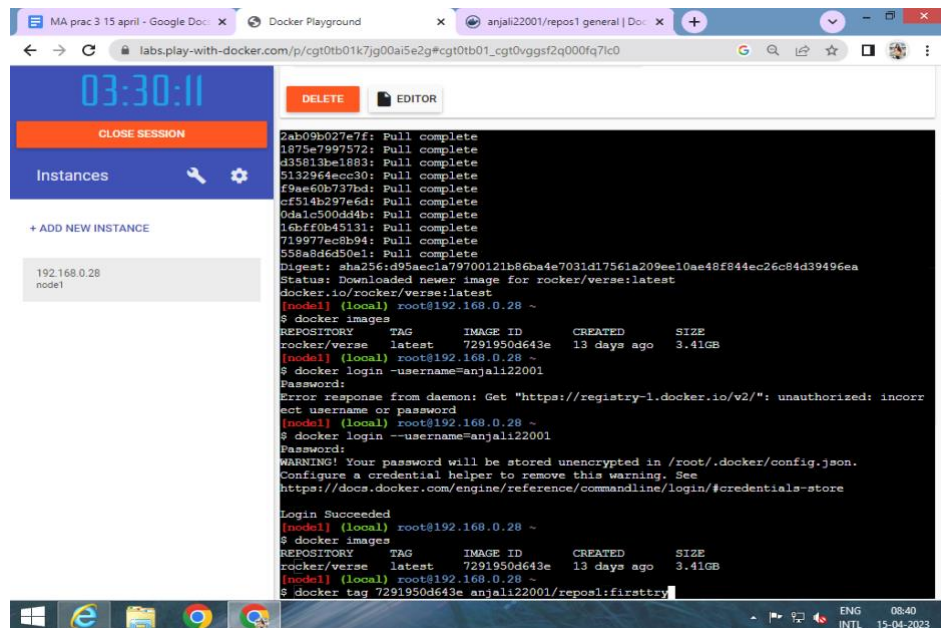
Output



The screenshot shows a Docker Playground interface with a terminal window. The terminal displays the following output:

```
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
#
# The FWD team. #
#####
(node1) (local) root@192.168.0.28 ~
$ docker --version
Docker version 20.10.17, build 100c701
(node1) (local) root@192.168.0.28 ~
$ docker pull rocker/verse
Using default tag: latest
latest: Pulling from rocker/verse
2ab09b027e7f: Pull complete
1875e7997572: Pull complete
d35813be1883: Pull complete
5132964ecc30: Pull complete
f9ae60b737bd: Pull complete
cf514b297e6d: Pull complete
0dalc500dd4b: Pull complete
16bfff0b45131: Pull complete
719977ec8b94: Pull complete
558a8d6d50e1: Pull complete
Digest: sha256:d95aeca1a79700121b86ba4e7031d17561a209ee10ae48f844ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID        CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago   3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker login -username=anjali22001
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorr
ect username or password
(node1) (local) root@192.168.0.28 ~
$
```

Command :to tag image `docker tag 7291950d643e anjali22001/repos1:firsttry`

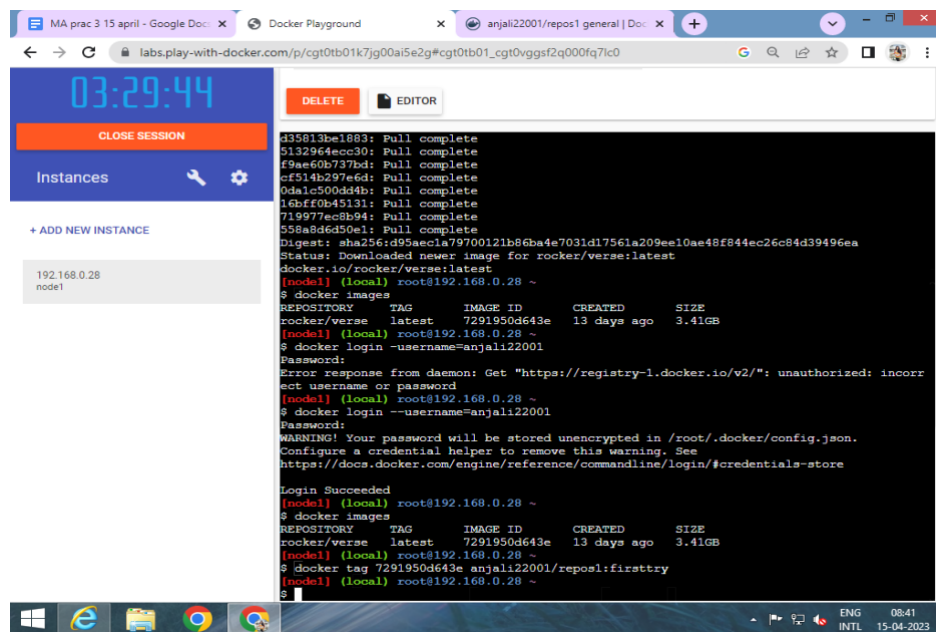


The screenshot shows a Docker Playground interface with a terminal window. The terminal displays the following output:

```
2ab09b027e7f: Pull complete
1875e7997572: Pull complete
d35813be1883: Pull complete
5132964ecc30: Pull complete
f9ae60b737bd: Pull complete
cf514b297e6d: Pull complete
0dalc500dd4b: Pull complete
16bfff0b45131: Pull complete
719977ec8b94: Pull complete
558a8d6d50e1: Pull complete
Digest: sha256:d95aeca1a79700121b86ba4e7031d17561a209ee10ae48f844ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID        CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago   3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker login -username=anjali22001
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorr
ect username or password
(node1) (local) root@192.168.0.28 ~
$ docker login --username=anjali22001
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID        CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago   3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker tag 7291950d643e anjali22001/repos1:firsttry
```

Output

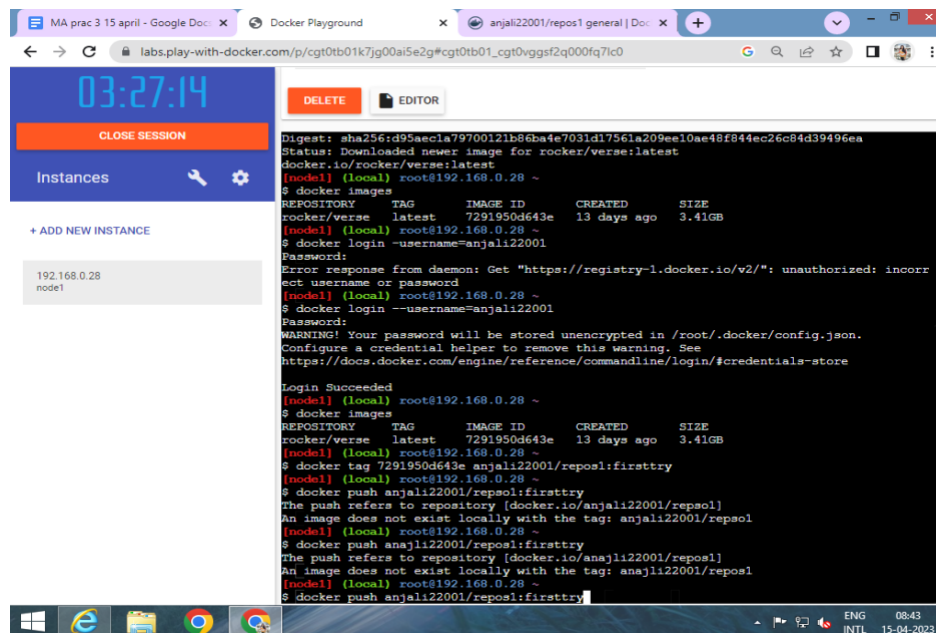


The screenshot shows a Docker Playground interface with a terminal window. The terminal displays the following commands and output:

```
435813be1883: Pull complete
5132964ecc30: Pull complete
f9ae60b737bd: Pull complete
cf514b297e6d: Pull complete
0da1c500d4b: Pull complete
12bf0845131: Pull complete
719977ec8b94: Pull complete
558a8d6d50e1: Pull complete
Digest: sha256:d95aecla79700121b86ba4e7031d17561a209ee10ae48f844ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago    3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker login -username=anjali22001
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
(node1) (local) root@192.168.0.28 ~
$ docker login --username=anjali22001
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago    3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker tag 7291950d643e anjali22001/repos1:firsttry
(node1) (local) root@192.168.0.28 ~
$
```

Command: to push image to docker hub account `docker push anjali22001/repos1:firsttry`



The screenshot shows a Docker Playground interface with a terminal window. The terminal displays the following commands and output:

```
Digest: sha256:d95aecla79700121b86ba4e7031d17561a209ee10ae48f844ec26c84d39496ea
Status: Downloaded newer image for rocker/verse:latest
docker.io/rocker/verse:latest
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago    3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker login -username=anjali22001
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
(node1) (local) root@192.168.0.28 ~
$ docker login --username=anjali22001
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
(node1) (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
rocker/verse   latest    7291950d643e   13 days ago    3.41GB
(node1) (local) root@192.168.0.28 ~
$ docker tag 7291950d643e anjali22001/repos1:firsttry
(node1) (local) root@192.168.0.28 ~
$ docker push anajli22001/repos1:firsttry
The push refers to repository [docker.io/anjali22001/repos1]
An image does not exist locally with the tag: anajli22001/repos1
(node1) (local) root@192.168.0.28 ~
$ docker push anajli22001/repos1:firsttry
The push refers to repository [docker.io/anjali22001/repos1]
An image does not exist locally with the tag: anajli22001/repos1
(node1) (local) root@192.168.0.28 ~
$ docker push anjali22001/repos1:firsttry
```

Output

The screenshot displays a web-based Docker Playground interface. The top navigation bar includes a clock showing 03:26:12, a 'CLOSE SESSION' button, and a list of instances. The main area is a terminal window with a black background and white text, showing a series of Docker commands and their outputs. The terminal session starts with a login success message, followed by listing local images, tagging a new image, and pushing it to a repository. The output shows the image being pushed successfully and the repository being updated. The terminal also displays a list of mounted volumes and their sources.

03:26:12

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.28
node1

DELETE EDITOR

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded
[root@192.168.0.28 ~]
\$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
rocker/verse latest 7291950d643e 13 days ago 3.41GB
[root@192.168.0.28 ~]
\$ docker tag 7291950d643e anjali22001/repool:fristtry
[root@192.168.0.28 ~]
\$ docker push anjali22001/repool:fristtry
The push refers to repository [docker.io/anjali22001/repool]
An image does not exist locally with the tag: anjali22001/repool
[root@192.168.0.28 ~]
\$ docker push anjali22001/repool:fristtry
The push refers to repository [docker.io/anjali22001/repool]
An image does not exist locally with the tag: anjali22001/repool
[root@192.168.0.28 ~]
\$ docker push anjali22001/repool:fristtry
The push refers to repository [docker.io/anjali22001/repool]
9d1382211021: Mounted from rocker/verse
e4144a267522: Mounted from rocker/verse
7ee25b0c682a: Mounted from rocker/verse
elececf7a947: Mounted from rocker/verse
829d4370cc55: Mounted from rocker/verse
a295f65c9f46: Mounted from rocker/verse
14d748851378: Mounted from rocker/verse
96f921c0089d: Mounted from rocker/verse
98a6c148ed82: Mounted from rocker/verse
b93c1bd012ab: Mounted from rocker/verse
fristtry: digest: sha256:3e11e7dc9187d49a9fe36a349af8901a9901123783c14d049b31417386f72b4
c size: 2428
[root@192.168.0.28 ~]
\$

ENG 08:44
INTL 15-04-2023

Practical No. 5: Working with docker volume & networks

Working with docker Volumes and Networks.
Perform Following Inside Play-With-Docker

Steps:

1) Pull nginx image into docker

Command: `docker pull nginx`

Output

```
[node1] (local) root@192.168.0.28 ~
$ docker pull nginx
bash: docker: command not found
[node1] (local) root@192.168.0.28 ~
$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
26c5c85e47da: Pull complete
4f3256bdf66b: Pull complete
2019c71d5655: Pull complete
8c767bdbc9ae: Pull complete
78e14bb05fd3: Pull complete
75576236abf5: Pull complete
Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
Status: Downloaded newer image for nginx:latest
```

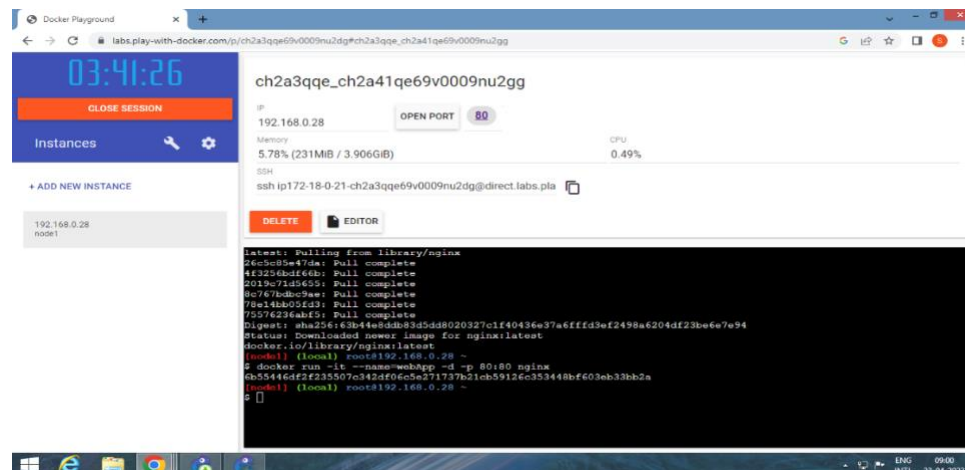
2) Now run the pulled image in Container named “webApp”

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

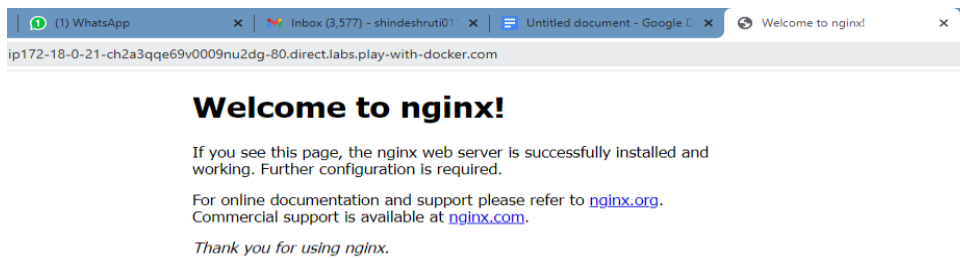
Output

```
[node1] (local) root@192.168.0.28 ~
$ docker run -it --name=webApp -d -p 80:80 nginx
6b55446df2f235507c342df06c5e271737b21cb59126c353448bf603eb33bb2a
```

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



Output



4) We make changes into "index.html" file inside /usr/share/nginx/html folder

Commands:

docker exec -it webApp bash //this command to execute bash shell
cd /usr/share/nginx/html //to go inside html folder
echo "Hello KB">index.html //to change content of index.html file

```
[node1] (local) root@192.168.0.28 ~
$ docker exec -it webApp bash
root@6b55446df2f2:/# cd /usr/share/nginx/html
bash: cd /usr/share/nginx/html: No such file or directory
root@6b55446df2f2:/# cd /usr/share/nginx/html
root@6b55446df2f2:/usr/share/nginx/html# echo "Hello KB">index.html
root@6b55446df2f2:/usr/share/nginx/html# exit
exit
```

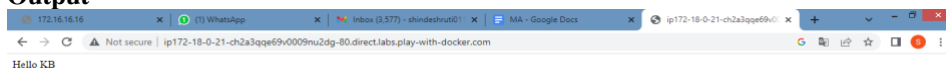
5) Type exit to return to docker prompt and check process status using ps option

Commands :Exit Docker ps

```
[node1] (local) root@192.168.0.28 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
6b55446df2f2   nginx    "/docker-entrypoint. ..." 15 minutes ago Up 15 minutes 0.0.0.0:80->80/tcp      webApp
[node1] (local) root@192.168.0.28 ~
```

6) Now refresh on port 80 output (you should get modified output)

Output



7) Now stop running container named "webApp".

Command: docker stop webApp

```
[node1] (local) root@192.168.0.28 ~
$ docker stop webApp
webApp
```

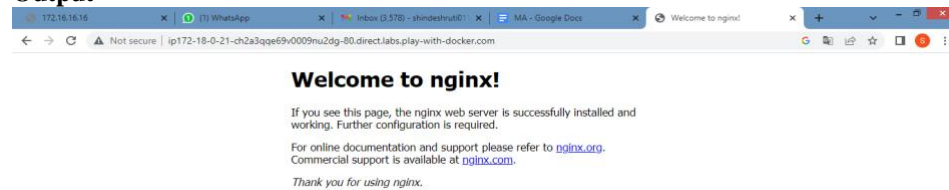
8) Start nginx in new container named "webApp1".

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

```
[node1] (local) root@192.168.0.28 ~
$ docker run -it --name=webApp1 -d -p 80:80 nginx
d83631275f4624e9fe19791b2b4b7eb6f5852e1219f54937cb177b254807838c
```

9)Now Click on port 80 (you will see the welcome page again)

Output



10)To solve this issue we create new volume.

Command: docker volume create MyVolume

```
[node1] (local) root@192.168.0.28 ~
$ docker volume create MyVolume
MyVolume
```

11)Check volume is created

Command: docker volume ls

```
[node1] (local) root@192.168.0.28 ~
$ docker volume ls
DRIVER      VOLUME NAME
local       MyVolume
[node1] (local) root@192.168.0.28 ~
$
```

12)Check details of volume

Command: docker volume inspect MyVolume

```
[node1] (local) root@192.168.0.28 ~
$ docker volume inspect MyVolume
[
  {
    "CreatedAt": "2023-04-23T03:40:19Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/MyVolume/_data",
    "Name": "MyVolume",
    "Options": {},
    "Scope": "local"
  }
]
```

13)Mount this volume to nginx new container named “webApp99”

Command: docker run -d --name=webApp99 --mount source=MyVolume,destination=/usr/share/nginx/html -p 81:80 nginx

Output

```
[node1] (local) root@192.168.0.28 ~
$ docker run -d --name=webApp99 --mount source=MyVolume,destination=/usr/share/nginx/html -p 81:80 nginx
4797324086ceb6b10aa1a3607f3f2db50a2e917c99dab342537e77a2ede96c91
```


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

Commands : ls/ cd/var/lib/docker

ls

Output

```
[node1] (local) root@192.168.0.28 ~
$ ls/
bash: ls/: No such file or directory
[node1] (local) root@192.168.0.28 ~
$ ls /
bin          docker.log  lib         mnt         root        srv         usr
certs        etc         lib64       opt         run         sys         var
dev          home        media       proc        sbin        tmp
[node1] (local) root@192.168.0.28 ~
$ cd /var/lib/docker
[node1] (local) root@192.168.0.28 /var/lib/docker
$ ls
buildkit     containers  network    plugins     swarm       trust
containerd  image      overlay2   runtimes    tmp          volumes
[node1] (local) root@192.168.0.28 /var/lib/docker
```

Commands: cd volumes

ls

cd MyVolume

ls

cd _data

ls

Output

```
[node1] (local) root@192.168.0.28 /var/lib/docker
$ cd volumes
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes
$ ls
MyVolume      backingFsBlockDev  metadata.db
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes
$ cd MyVolume
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume
$ ls
_data
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume
$ cd _data
bash: cd _data: command not found
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume
$ cd _data
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ _data
$ ls
50x.html      index.html
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ _data
```

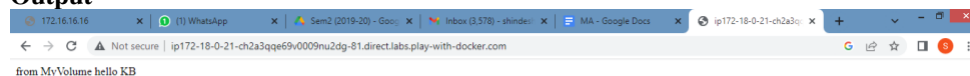
15) Modify contents of index.html file with “from My Volume hello KB”

Command: echo “from My Volume hello KB”> index.html

```
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ _data
$ echo "from MyVolume hello KB"> index.html
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ _data
```

16) Now refresh port 80(to get modified output)

Output



The screenshot shows a web browser window with the address bar displaying 'ip172-18-0-21-ch2a3q...'. The page content shows 'from MyVolume hello KB'.

17) Now stop this running container named “webApp4”

Command: docker stop webApp99

Output

```
$ docker stop webApp99
webApp99
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ data
$
```

18) Now run nginx in new container named “webApp10”

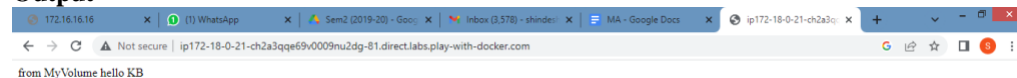
Command: `docker run -d --name=webApp10 --mount source=MyVolume,destination=/usr/share/nginx/html -p 81:80 nginx`

Output

```
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ data
$ docker run -d --name=webApp10 --mount source=MyVolume,destination=/usr/share/nginx/html -p 81:80 nginx
79278dc81df0fd5f53eb10e22157d6be45c787c5705334c0571be437db1f1f
[node1] (local) root@192.168.0.28 /var/lib/docker/volumes/MyVolume/ data
$
```

19) 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.

Output



The screenshot shows a web browser window with multiple tabs. The active tab is titled "ip172-18-0-21-ch2a3c0571be437db1f1f". The address bar shows "ip172-18-0-21-ch2a3c0571be437db1f1f.direct.labs.play-with-docker.com". The page content displays "from MyVolume hello KB".