

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

By

MOHITE SAYALI SHYAM

Seat Number: (4133164)

SUBJECT NAME:

MICREOSERVICE ARCHITECTURE AND IMAGE PROCESSING



DEPARTMENT OF INFORMATION TECHNOLOGY

N. G. ACHARYA & D. K. MARATHE COLLEGE

(Affiliated to University of Mumbai)

MUMBAI , 400071

MAHARASHTRA

2022-23

DEPARTMENT OF INFORMATION TECHNOLOGY
N. G. ACHARYA & D. K. MARATHE COLLEGE
(Affiliated to University of Mumbai)
MUMBAI – MAHARASHTRA - 400071

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that Mohite Sayali Shyam bearing Seat No: 4133164 submitted journal of **Microservices Architecture** techquie in partial fulfillment of the requirements for the award of Degree of **MASTER OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai.

Internal Guide

Coordinator

External Examiner

Date:

College Seal

MICROSERVICES ARCHITECTURE

INDEX

Sr. No.	Date	Title	Signature
1.	19/03/2022	Create a console based ASP.net core application	
2.	24/03/2022	Create a MVC Project in ASP.net core	
3.	31/03/2022	Usage of Docker Desktop	
4.	07/04/2022	Working with Docker	
5.	16/04/2022	Building ASP.Net core REST API	
6.	21/04/2022	Working with Circle CI for continuous integration	
7.	22/04/2022	Working with Team Service	

Practical No 1

Aim : Create a console based ASP.net core application.

Source Code :

Step 1 :

- Download the asp.net core sdk from
<https://dotnet.microsoft.com/learn/dotnet/hello-worldtutorial/install>
- Install the asp.net core sdk.
- To check whether the asp.net sdk is successful install, open command prompt and type command: **dotnet**

```
C:\Users\Shraddha Shah>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\Shraddha Shah>
```

- To check the version of the dotnet

```
C:\Users\Shraddha Shah>dotnet --version
6.0.202
```

Step 2 :

- Go to the drive where you want to create the console application. Create a folder in the drive and go to that folder. Type the following command in the command prompt to create the application.

```
D:\MSA Pracs\prac1>cd..
D:\MSA Pracs>md HelloWorld
D:\MSA Pracs>cd Hell*
D:\MSA Pracs\HelloWorld>dotnet new console
The template "Console App" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on D:\MSA Pracs\HelloWorld\HelloWorld.csproj...
  Determining projects to restore...
    Restored D:\MSA Pracs\HelloWorld\HelloWorld.csproj (in 63 ms).
Restore succeeded.
```

- Restore the project and run the application

```
D:\MSA Pracs>cd hell*
D:\MSA Pracs\HelloWorld>dotnet restore
  Determining projects to restore...
    Restored D:\MSA Pracs\HelloWorld\HelloWorld.csproj (in 25.91 sec).

D:\MSA Pracs\HelloWorld>dotnet run
Hello, World!
```

Step 3 :

- Now open **HelloWorld.csproj** file, edit the code

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.Mvc"
      Version="1.1.1"/>
    <PackageReference Include="Microsoft.AspNetCore.Server.Kestrel"
      Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Logging"
      Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Logging.Console"
      Version="1.1.1"/>
    <PackageReference Include="Microsoft.Extensions.Logging.Debug"
      Version="1.1.1"/>
    <PackageReference
      Include="Microsoft.Extensions.Configuration.CommandLine"
      Version="1.1.1"/>
  </ItemGroup>
</Project>
```

- Open Program.cs file and edit the code

```
using System;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Configuration;
namespace HelloWorld // Note: actual namespace depends on the
project name.
{
  internal class Program
  {
```

```

        static void Main(string[] args)
    {
        var config = new ConfigurationBuilder()
            .AddCommandLine(args)
        .Build();
        var host = new WebHostBuilder()
            .UseKestrel()
            .UseStartup<Startup>()
            .UseConfiguration(config)
        .Build();
        host.Run();
    }
}

public class Startup
{
    public Startup(IHostingEnvironment env) { }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)
    {
        app.Run(async (context) => { await
        context.Response.WriteAsync("Hello, world!");});
    }
}

```

Step 4 :

Restore the project.

```

D:\MSA Pracs\HelloWorld>dotnet restore
Determining projects to restore...
All projects are up-to-date for restore.

```

Output :

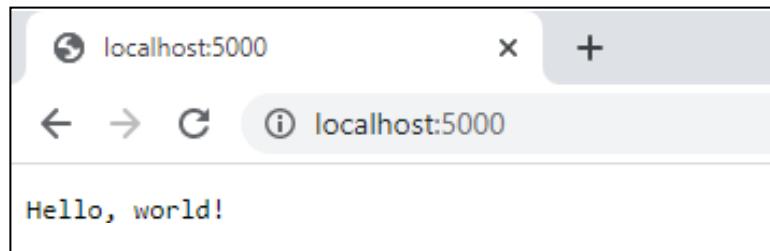
Run the application

```

D:\MSA Pracs\HelloWorld>dotnet run
Hosting environment: Production
Content root path: D:\MSA Pracs\HelloWorld\bin\Debug\net6.0\
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
Application is shutting down...

```

Now open the browser open the url: <http://localhost:5000>



```
C:\Users\Sameer Dhotre>curl http://localhost:5000
Hello, world!
C:\Users\Sameer Dhotre>curl localhost:5000/will/any/url/work?
Hello, world!
C:\Users\Sameer Dhotre>
```

A screenshot of a Windows command prompt window titled "C:\Users\Sameer Dhotre\Appl". It shows two curl commands being run. The first command sends a GET request to "http://localhost:5000" and receives the response "Hello, world!". The second command sends a GET request to "localhost:5000/will/any/url/work?" and also receives the response "Hello, world!". The prompt ends with a cursor at "C:\Users\Sameer Dhotre>".

Practical No 2

Aim : Create a MVC Project in ASP.net core

Source Code :

Step 1 :

Create a mvc project

```
dotnet new mvc --auth none
```

```
D:\Microservices Architecture\Practice Practical\Practs\pracs2>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/6.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on D:\Microservices Architecture\Practice Practical\Practs\pracs2\pracs2.csproj...
  Determining projects to restore...
    Restored D:\Microservices Architecture\Practice Practical\Practs\pracs2\pracs2.csproj (in 278 ms).
Restore succeeded.

D:\Microservices Architecture\Practice Practical\Practs\pracs2>
```

Step 2 :

Restore, build and run the program.

Use the first url of the command prompt in the browser and see the output

```
D:\Microservices Architecture\Practice Practical\Practs\pracs2>dotnet build
Microsoft (R) Build Engine version 17.1.1+ad2f73656 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

  Determining projects to restore...
  All projects are up-to-date for restore.
  pracs2 -> D:\Microservices Architecture\Practice Practical\Practs\pracs2\bin\Debug\net6.0\pracs2.dll

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

Time Elapsed 00:00:04.72

D:\Microservices Architecture\Practice Practical\Practs\pracs2>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7091
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5103
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:\Microservices Architecture\Practice Practical\Practs\pracs2\
```

prac21 Home Privacy

Welcome

Learn about [building Web apps with ASP.NET Core](#).

Step 3 :

Go to Models Folder and create StockQuote.cs file in it.

```
using System;
```

```
namespace pracs.Models
```

```
{
```

```
  public class StockQuote
```

```

    {
        public string Symbol {get;set;}
        public int Price{get;set;}
    }
}

```

Step 4 :

Now go to views folder and then in home folder. Edit the index.cshtml file

```

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

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    Symbol: @Model.Symbol <br/>
    Price: $@Model.Price <br/>
</div>

```

Step 5 :

Now go to controller folder and edit HomeController.cs

```

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 pracs2.Models;

```

```

namespace pracs2.Controllers;
```

```

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

```

}

Step 6 :

```
D:\Microservices Architecture\Practice Practical\Practs\pracs2>dotnet build
Microsoft (R) Build Engine version 17.1.1+02f73656 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
All projects are up-to-date for restore.
pracs2 -> D:\Microservices Architecture\Practice Practical\Practs\pracs2\bin\Debug\net6.0\pracs2.dll

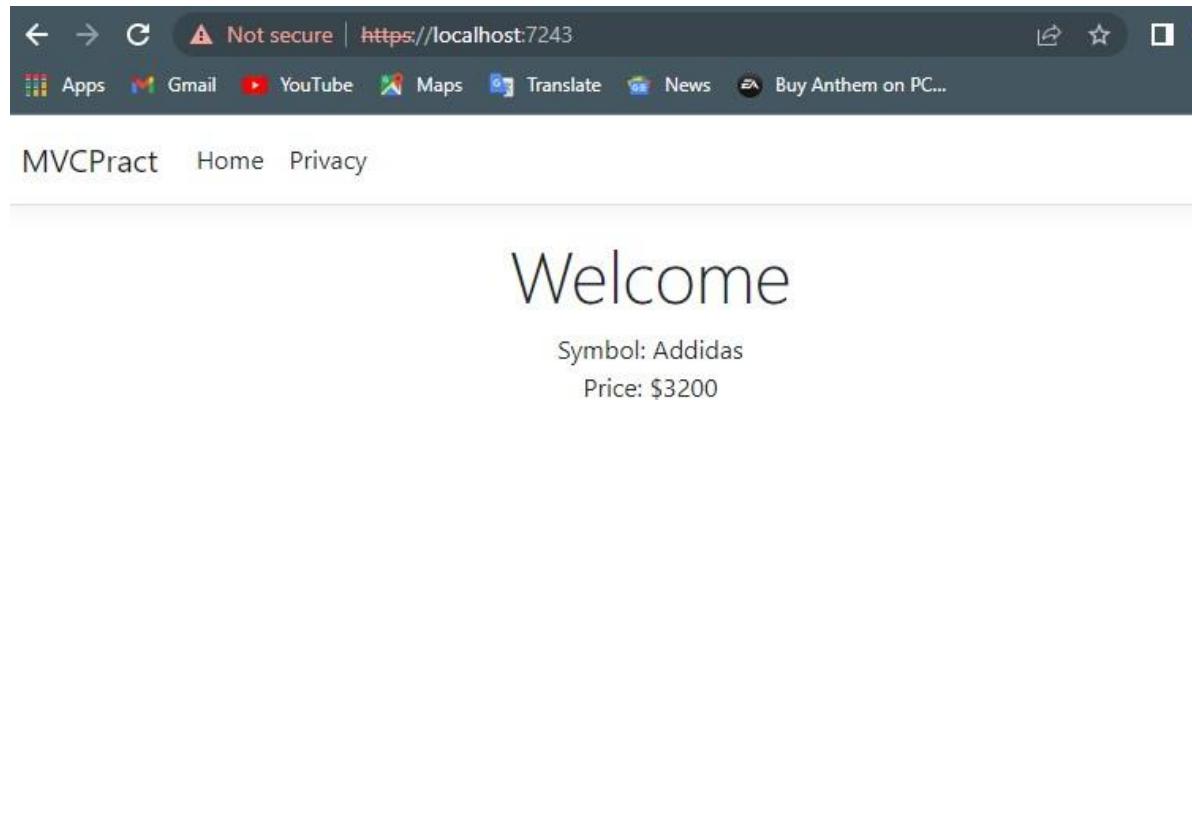
Build succeeded.
  0 Warning(s)
  0 Error(s)

Time Elapsed 00:00:04.31

D:\Microservices Architecture\Practice Practical\Practs\pracs2>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7091
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5103
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:\Microservices Architecture\Practice Practical\Practs\pracs2\
```

Output :

Open the first url in the browser and see the output



Practical No 3

Aim : Usage of Docker Desktop

Commands & its output :

Open command prompt

- To check whether docker is installed properly
\$ docker

```
D:\msa>docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                       "C:\\\\Users\\\\Admin\\\\.docker")
  -c, --context string Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string  Trust certs signed only by this CA (default
                       "C:\\\\Users\\\\Admin\\\\.docker\\\\ca.pem")
  --tlscert string    Path to TLS certificate file (default
                       "C:\\\\Users\\\\Admin\\\\.docker\\\\cert.pem")
  --tlskey string     Path to TLS key file (default
                       "C:\\\\Users\\\\Admin\\\\.docker\\\\key.pem")
  --tlsverify         Use TLS and verify the remote
  -v, --version        Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*      Docker Buildx (Docker Inc., v0.8.2)
  compose*     Docker Compose (Docker Inc., v2.4.1)
  config       Manage Docker configs
  container    Manage containers
  context      Manage contexts
  image        Manage images
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
  node         Manage Swarm nodes
  plugin      Manage plugins
  sbom*        View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scan*        Docker Scan (Docker Inc., v0.17.0)
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  trust        Manage trust on Docker images
  volume      Manage volumes
```

- To see the version of the docker

\$ docker -v

```
D:\msa>docker -v
Docker version 20.10.14, build a224086

D:\msa>-
```

- To run hello-world image

```
$ docker run -p 8080:8080 dotnetcoreservices/hello-world
```

```
D:\msa>docker run -p 8080:8080 dotnetcoreservices/hello-world
Hosting environment: Production
Content root path: /pipeline/source/app/publish
Now listening on: http://0.0.0.0:8080
Application started. Press Ctrl+C to shut down.
```

- Run localhost in the browser

```
http://localhost:8080
```



- To see the output in the command prompt

```
$ curl http://localhost:8080/will/itblend?
```

```
C:\ Command Prompt
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>curl http://localhost:8080/will/itblend?
Hello, world!

C:\Users\Admin>
```

- To see the images in the docker

```
$ docker ps
```

- To terminate the image in the docker.
note the container id of the docker that you want to terminal and
replace the <Containerid> in the below command

\$ docker kill <containerid>

```
C:\Users\Admin>docker kill 35c840e18b74  
35c840e18b74
```

To check whether the docker is terminated or not
\$ docker ps

Practical No 4

Aim : Working with Docker

Commands and its output :

Step 1:

- Create a account in the docker hub. Remember the username and password of the account

Step 2 :

- Now to go <https://labs.play-with-docker.com/> and click on **Start** button.
- Click on **Add New Instance**. You will see the editor open in the right pane. Give the commands in the editor

Step 3 :

- To check the version of the docker

```
$ docker –version
```

```
[node1] (local) root@192.168.0.18 ~  
$ docker --version  
Docker version 20.10.0, build 7287ab3  
[node1] (local) root@192.168.0.18 ~  
c
```

- To pull the readymade image

```
$ docker pull hello-world
```

```
$ docker pull hello-world  
Using default tag: latest  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:80f31da1ac7b312ba29d65080fddff97dd76acfb870e677f390d5acba9741b17  
Status: Downloaded newer image for hello-world:latest  
docker.io/library/hello-world:latest  
[node1] (local) root@192.168.0.18 ~  
c
```

- To check the images in docker

```
$ docker images
```

```
$ docker images  
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE  
hello-world     latest       feb5d9fea6a5   7 months ago   13.3kB  
[node1] (local) root@192.168.0.18 ~
```

Part 1: To pull and Push images in docker

Step 4 :

- Open the new tab in the browser and login to hub.docker.com
- Click on **Repositories** and then click on **Create Repositories**

- Give the name of the repository as “**repo1**” and in description add “**My first repository**”
- Make visibility as **Private**
- And now click on **Create** button and check whether the repository is created or not.

Step 5 :

- Now come to the <https://labs.play-with-docker.com/> and give the following command
- Login into docker account
\$ docker login -username= your_user_name
password:

```
[node1] (local) root@192.168.0.18 ~
$ docker login --username=vishwakarma1919
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
```

Note: Give your username and password that you have used to login to hub.docker.com

- To tag an image in docker
\$ docker tag <image id> <username>/repo1:firsttry

```
[node1] (local) root@192.168.0.18 ~
$ docker tag feb5d9fea6a5 vishwakarma1919/repo1:firsttry
[node1] (local) root@192.168.0.18 ~
$
```

- To push the image to docker account
\$ docker push <username>/repo1:firsttry

```
[node1] (local) root@192.168.0.18 ~
$ docker push vishwakarma1919/repo1:firsttry
The push refers to repository [docker.io/vishwakarma1919/repo1]
e07ee1baac5f: Mounted from library/hello-world
firsttry: digest: sha256:f54a58bc1aac5eala25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
[node1] (local) root@192.168.0.18 ~
$
```

Note: firsttry is tag name created above.

- Check it in hub.docker.com now in tags tab

Tags and Scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
firsttry		---	4 minutes ago

[See all](#)

Part 2 : Build and image and then push and run in the docker0

Step 6 :

- In <https://labs.play-with-docker.com/> give the following command


```
cat > Dockerfile <<EOF
FROM busybox
CMD echo "Hello world! This is my first Docker image."
EOF
```

```
[node1] (local) root@192.168.0.18 ~
$ cat> Dockerfile <<EOF
> FROM busybox
> CMD echo "Hello World! This Is My First Docker Image."
> EOF
```

- To build the image from docker file

```
$ docker build -t <username>/repo2 .
```

```
[node1] (local) root@192.168.0.18 ~
$ docker build -t vishwakarma1919/repo2 .
Sending build context to Docker daemon    47MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
50e8d59317eb: Pull complete
Digest: sha256:d2b53584f580310186df7a2055ce3ff83cc0df6caacf1e3489bff8cf5d0af5d8
Status: Downloaded newer image for busybox:latest
--> 1a80408de790
Step 2/2 : CMD echo "Hello World! This Is My First Docker Image."
--> Running in 523badc76755
Removing intermediate container 523badc76755
--> 58a88ef19a6a
Successfully built 58a88ef19a6a
Successfully tagged vishwakarma1919/repo2:latest
[node1] (local) root@192.168.0.18 ~
```

- Check images in docker

```
$ docker images
```

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
vishwakarma1919/repo2    latest   58a88ef19a6a  26 seconds ago  1.24MB
busybox              latest   1a80408de790  4 weeks ago   1.24MB
hello-world          latest   feb5d9fea6a5  7 months ago   13.3kB
vishwakarma1919/repo1    firsttry  feb5d9fea6a5  7 months ago   13.3kB
[node1] (local) root@192.168.0.18 ~
```

- To push the image on the docker hub

```
$ docker push <username>/repo2.
```

```
[node2] (local) root@192.168.0.8 ~
$ docker push vishwakarma1919/repo2
Using default tag: latest
The push refers to repository [docker.io/vishwakarma1919/repo2]
eb6b01329ebe: Mounted from library/busybox
latest: digest: sha256:4452bb83a562a0ce6a5e1fa11159957b8ad3cc62dff6ad14b60dd4e5dd29bf3 size: 527
```

- Check it in hub.docker.com now in tags tab



- Come back to the <https://labs.play-with-docker.com/> and give the below command to run the docker image

```
$ docker run <username>/repo2
```

```
[node2] (local) root@192.168.0.8 ~
$ docker run vishwakarma1919/repo2
Hello world! This is My First Docker Image
[node2] (local) root@192.168.0.8 ~
$
```

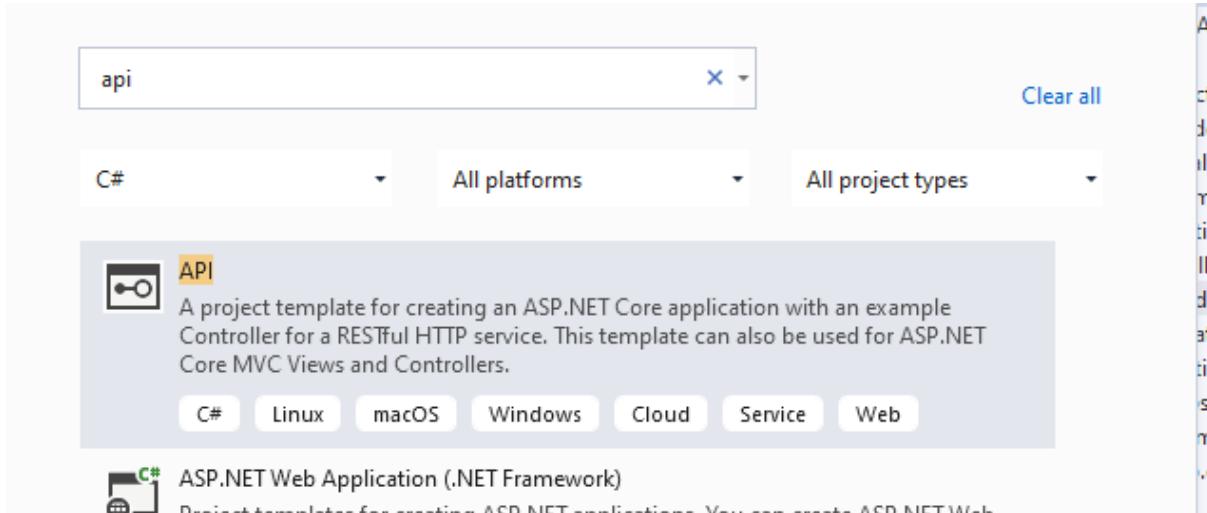
- Close the session

PRACTICAL NO 5

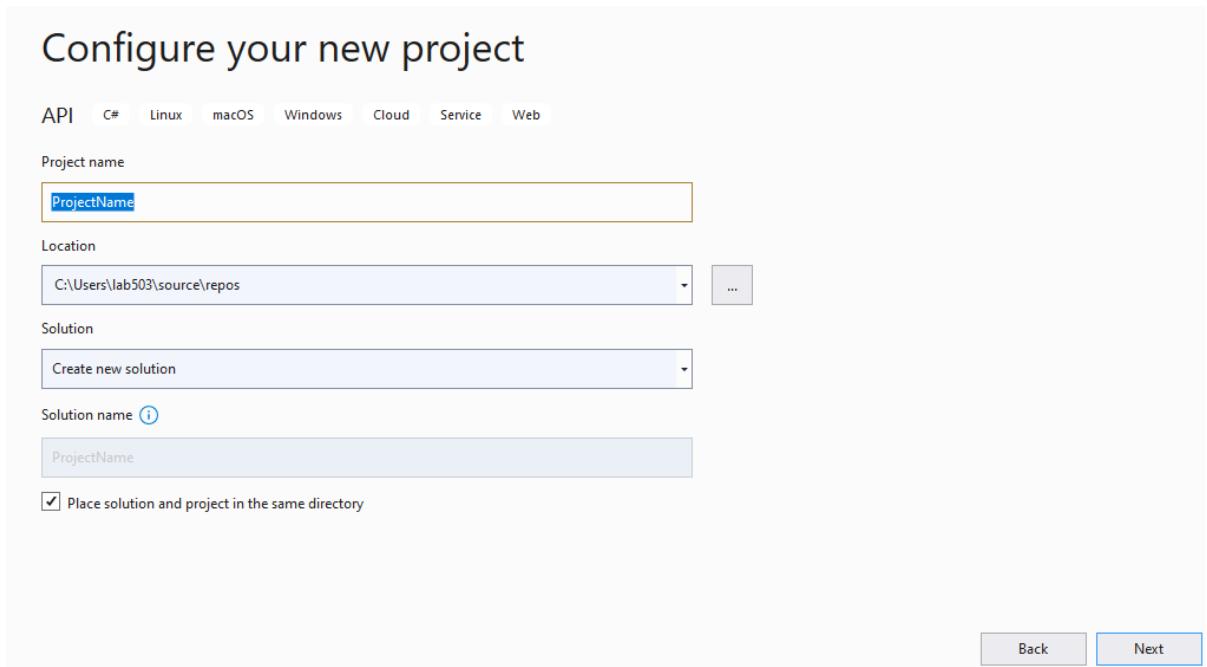
Building ASP.Net core REST API

Step 1:- create new project in visual studio 2019

Step 2:- Select the Template API as mentioned below



Step 3:- Create the project with name and configures



Additional information

API C# Linux macOS Windows Cloud Service Web

Target Framework [\(i\)](#)

.NET Core 3.1 (Out of support)

Authentication Type [\(i\)](#)

None

Configure for HTTPS [\(i\)](#)

Enable Docker [\(i\)](#)

Docker OS [\(i\)](#)

Linux

Step 4:- You will Get all the pre-configured Code under the *Program.cs* and *Startup.cs*

Below are the Code of Startup.cs

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace ProjectName
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }
    }
}
```

```

public IConfiguration Configuration { get; }

    // This method gets called by the runtime. Use this method to add services to the
    container.
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddControllers();
    }
    [Route ("api/[controller]")]

    public class GlossaryController : ControllerBase

        // This method gets called by the runtime. Use this method to configure the HTTP
        request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }

            app.UseHttpsRedirection();

            app.UseRouting();

            app.UseAuthorization();

            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllers();
            });
        }
    }
}

```

Below are the Code of Program.cs

```

using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging; using
System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace ProjectName
{
    public class Program
    {

```

```

public static void Main(string[] args)
{
    CreateHostBuilder(args).Build().Run();
}

public static IHostBuilder CreateHostBuilder(string[] args)
=>Host.CreateDefaultBuilder(args)
    .ConfigureWebHostDefaults(webBuilder =>
    {
        webBuilder.UseStartup<Startup>();
    });
}
}

```

Step 5: Create Student pojo class to store the student infoBelow

Student.cs pojo class

```

namespace RestAPI.Controllers
{
    public class Students
    {
        public string Name { get; set; }
        public string Rollno { get; set; }

    }
}

```

Below Student Controller class file

```

using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.IO;

namespace RestAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class Student : ControllerBase
    {
        //this is local database
        private static readonly List<Students> Studentlist = new List<Students>
        {
            new Students
            {
                Name = "Abdulsalam",
                Rollno = "195"
            },
            new Students
        }
    }
}

```

```

{
    Name = "Maria",
    Rollno = "179"
},
new Students
{
    Name = "imran",
    Rollno = "201"
}

};

//You will get all student from here post
method[HttpGet]
public ActionResult<List<Students>> Get()
{
    return Ok(Studentlist);
}

//You will get perticular student from here post method
[HttpGet]
[Route("{Name}")]
public ActionResult<List<Students>> Get(string Name)
{
    var student = Studentlist.Find(student =>

        student.Name.Equals(Name,
        StringComparison.InvariantCultureIgnoreCase));if (student == null)
    {
        return NotFound();
    }
    else
    {
        return Ok(student);
    }
}

//You will post perticular student from here post method
// post
[HttpPost]
public ActionResult Post(Students sentstudents)
{
    var sendstudent = Studentlist.Find(item =>
        item.Name.Equals(sentstudents.Name,
        StringComparison.InvariantCultureIgnoreCase));
    if (sendstudent != null)
    {
        return Conflict("Cannot create the student because it already exists.");
    }
    else
}

```

```

    {
        Studentlist.Add(sentstudents);
        var resourceUrl =
            Path.Combine(Request.Path.ToString(),
            Uri.EscapeUriString(sentstudents.Name));
        return Created(resourceUrl, sentstudents);
    }
}

//post
//You will add perticular student from here put method
//put
// [HttpPost]
public ActionResult Put(Students sentstudents)
{
    var sendstudent = Studentlist.Find(item =>
    item.Name.Equals(sentstudents.Name,
    StringComparison.InvariantCultureIgnoreCase));
    if (sendstudent == null)
    {
        return BadRequest("Cannot update a nont existing term.");
    }
    else
    {
        sendstudent.Rollno = sentstudents.Rollno;
        return Ok();
    }
}
//
//delete
/// //You will Delete perticular student from here Delete
method[HttpDelete]
[Route("{Name}")]
public ActionResult Delete(string Name)
{
    var Studentnames = Studentlist.Find(item =>
    item.Name.Equals(Name,
    StringComparison.InvariantCultureIgnoreCase));
    if (Studentnames == null)
    {
        return NotFound();
    }
    else
    {
        Studentlist.Remove(Studentnames);
        return NoContent();
    }
}
}

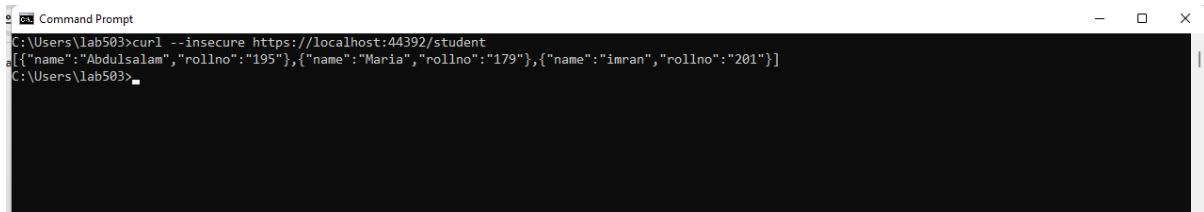
```

```
//  
}
```

All other configurations and file will remain same

Output

Get method

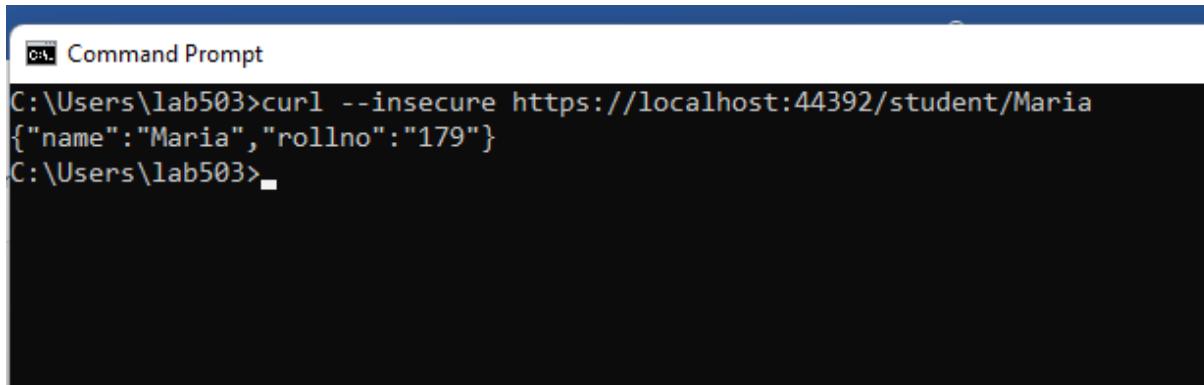


```
Command Prompt  
C:\Users\lab503>curl --insecure https://localhost:44392/student  
[{"name": "Abdulsalam", "rollno": "195"}, {"name": "Maria", "rollno": "179"}, {"name": "imran", "rollno": "201"}]  
C:\Users\lab503>
```

Get method Particular

Student

In this case student name *Maria*



```
Command Prompt  
C:\Users\lab503>curl --insecure https://localhost:44392/student/Maria  
{"name": "Maria", "rollno": "179"}  
C:\Users\lab503>
```

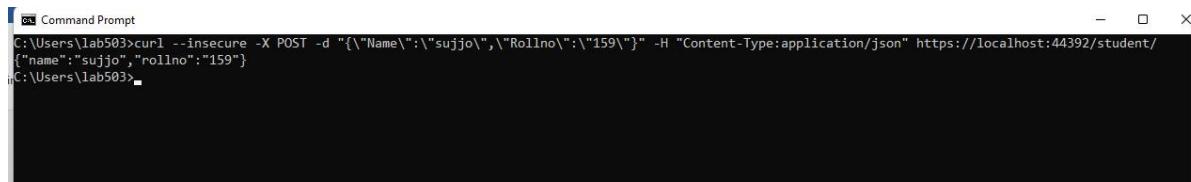
Post method Particular

Student

In this case student name *Sujjo*

```
curl --insecure -X POST -d "{\"Name\":\"sujjo\",\"Rollno\":\"159\"}" -H "Content-
```

Type:application/json" <https://localhost:44392/student/>



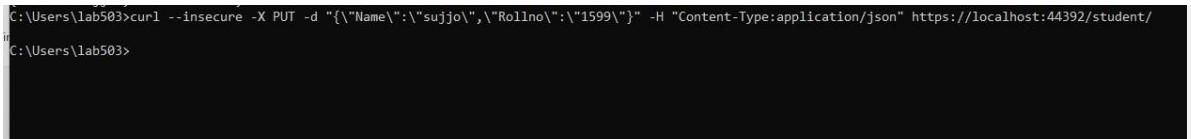
```
C:\Users\lab503>curl --insecure -X POST -d "{\"Name\":\"sujjo\",\"Rollno\":\"159\"}" -H "Content-Type:application/json" https://localhost:44392/student/
C:\Users\lab503>
```

Put method Particular

Student

In this case student name *Sujjo*

```
curl --insecure -X PUT -d "{\"Name\":\"sujjo\",\"Rollno\":\"1599\"}" -H "Content-Type:application/json" https://localhost:44392/student/
```



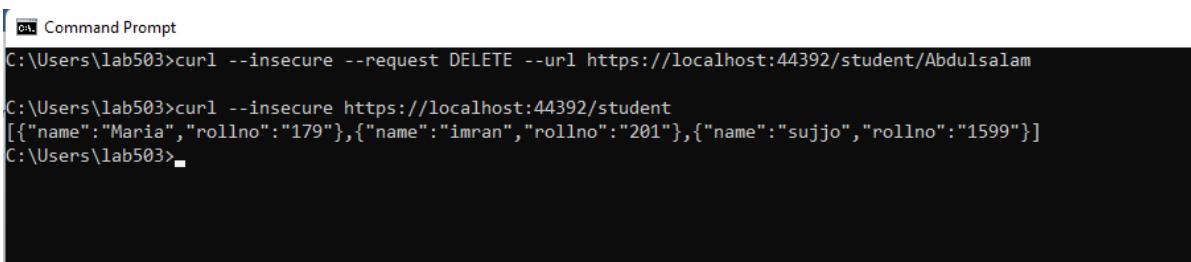
```
C:\Users\lab503>curl --insecure -X PUT -d "{\"Name\":\"sujjo\",\"Rollno\":\"1599\"}" -H "Content-Type:application/json" https://localhost:44392/student/
C:\Users\lab503>
```

Delete

Particular Student

In this case student name *Abdulsalam*

```
curl --insecure --request DELETE --url https://localhost:44392/student/Abdulsalam
```



```
C:\Users\lab503>curl --insecure --request DELETE --url https://localhost:44392/student/Abdulsalam
C:\Users\lab503>curl --insecure https://localhost:44392/student
[{"name": "Maria", "rollno": "179"}, {"name": "imran", "rollno": "201"}, {"name": "sujjo", "rollno": "1599"}]
C:\Users\lab503>
```

```

        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();
    }
}

```

```

        }

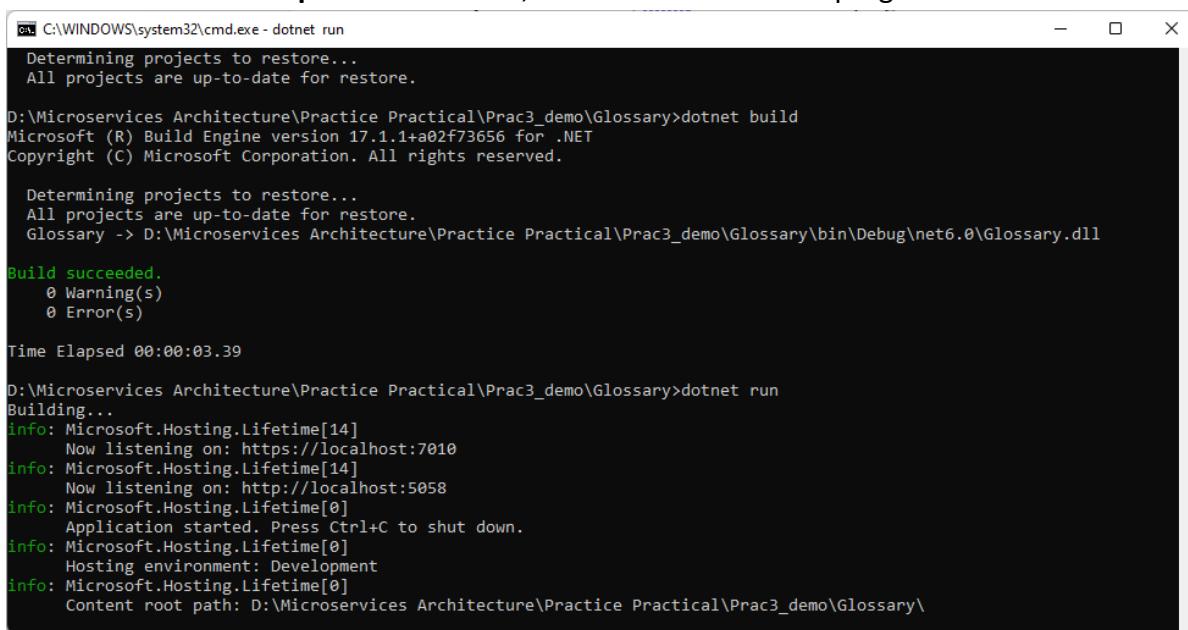
    }

    [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 : To stop the application running on command prompt do Ctrl+c

Step 7 : Now restore, build and then run the program



```

C:\WINDOWS\system32\cmd.exe - dotnet run
Determining projects to restore...
All projects are up-to-date for restore.

D:\Microservices Architecture\Practice Practical\Prac3_demo\Glossary>dotnet build
Microsoft (R) Build Engine version 17.1.1+a02f73656 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

Determining projects to restore...
All projects are up-to-date for restore.
Glossary -> D:\Microservices Architecture\Practice Practical\Prac3_demo\Glossary\bin\Debug\net6.0\Glossary.dll

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

Time Elapsed 00:00:03.39

D:\Microservices Architecture\Practice Practical\Prac3_demo\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:7010
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5058
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:\Microservices Architecture\Practice Practical\Prac3_demo\Glossary\

```

Output :

Open the other command prompt and give the following command.

Kindly note the port number that you will get in the previous command prompt and change the port number in the curl

1. Getting the List of Items

curl --insecure <https://localhost:7010/api/glossary>

2. Getting Single Item

a. curl --insecure <https://localhost:7010/api/glossary/MVC>

b. curl --insecure <https://localhost:7010/api/glossary/HTML>

c. curl --insecure <https://localhost:7010/api/glossary/OpenID>

3. Creating an item

curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json"
<https://localhost:7010/api/glossary>

4. Updating an Item

curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json"
<https://localhost:7010/api/glossary>

5. Delete an Item

curl --insecure --request DELETE --url
<https://localhost:7010/api/glossary/openid>

Output:-

Open the other command prompt and give the following command.

Kindly note the port number that you will get in the previous command prompt and change the port number in the curl

1. Getting the List of Items

```
curl --insecure https://localhost:7010/api/glossary
```

```
D:\>curl --insecure https://localhost:7136/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MVC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}]
D:\>
```

2. Getting Single Item

a. curl --insecure <https://localhost:7010/api/glossary/MVC>

```
D:\>curl --insecure https://localhost:7136/api/glossary/MVC
{"term": "MVC", "definition": "Model View Controller"}
```

b. curl --insecure <https://localhost:7010/api/glossary/HTML>

```
D:\>curl --insecure https://localhost:7136/api/glossary/HTML
[{"term": "HTML", "definition": "Hypertext Markup Language"}]
D:\>
```

c. curl --insecure <https://localhost:7010/api/glossary/OpenID>

```
D:\>curl --insecure https://localhost:7136/api/glossary/OpenID
{"term": "OpenID", "definition": "An open standard for authentication"}
D:\>
```

3. Creating an item

```
curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:7010/api/glossary
```

```
D:\>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:7136/api/glossary
>{"term": "MFA", "definition": "An authentication process."}
D:\>
```

4. Updating an Item

```
curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json"
https://localhost:7010/api/glossary
```

5.

Delete an Item

```
curl --insecure --request DELETE --url https://localhost:7010/api/glossary/openid
```

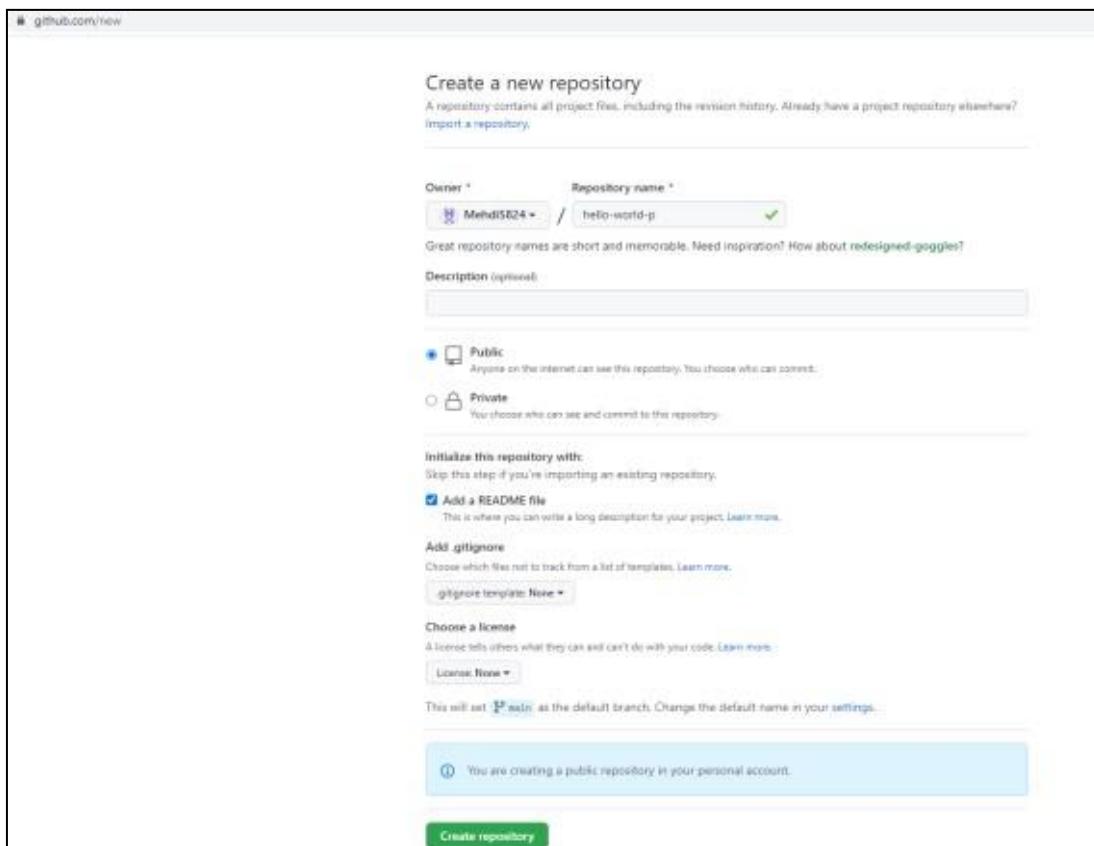
Practical No 6

Aim : Working with Circle CI for continuous integration

Steps and its output :

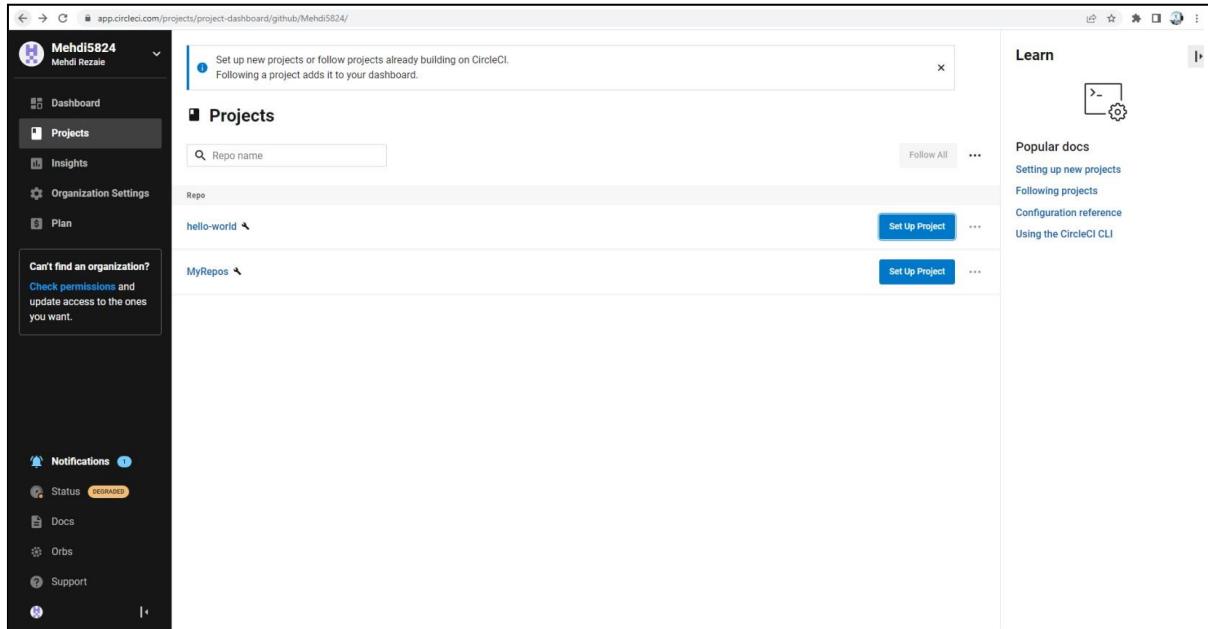
Step 1 : Create a repository

1. Log in to GitHub and begin the process to create a new repository.
2. Enter a name for your repository (for example, hello-world).
3. Select the option to initialize the repository with a README file.
4. Finally, click Create repository.
5. There is no need to add any source code for now.



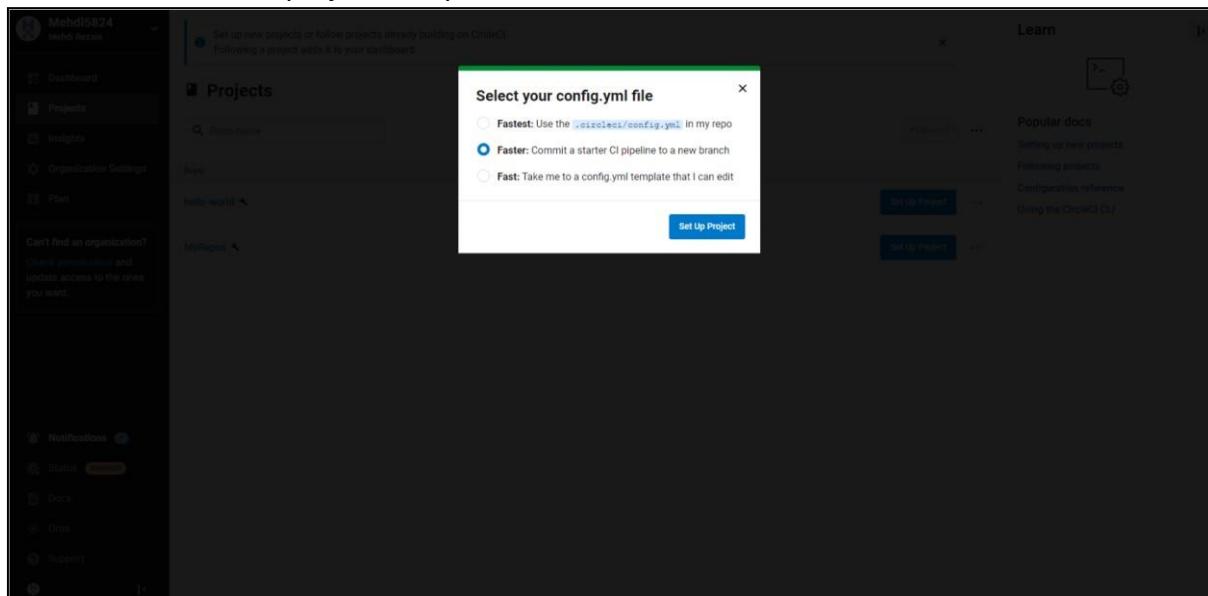
Step 2 : Set up CircleCI

- Login to Circle CI <https://app.circleci.com/> using GitHub Login
- Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.



The screenshot shows the CircleCI Projects dashboard. On the left, there's a sidebar with options like Dashboard, Projects (which is selected), Insights, Organization Settings, Plan, Notifications (with 1 notification), Status (DEGRADED), Docs, Orbs, and Support. The main area shows a search bar for 'Repo name' and two repositories: 'hello-world' and 'MyRepos'. A modal window at the top says 'Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard.' Below the repos, there are 'Set Up Project' and '...' buttons. To the right, there's a 'Learn' sidebar with links to Popular docs (Setting up new projects, Following projects, Configuration reference, Using the CircleCI CLI).

- You will be taken to the Projects dashboard. On the dashboard, select the project you want to set up (hello-world).
- Select the option to commit a starter CI pipeline to a new branch, and click Set Up Project. This will create a file **.circleci/config.yml** at the root of your repository on a new branch called circleci-project-setup.



The screenshot shows a 'Select your config.yml file' dialog box over the CircleCI Projects dashboard. The dialog has three options:

- Fastest: Use the `.circleci/config.yml` in my repo
- Faster: Commit a starter CI pipeline to a new branch
- Fast: Take me to a config.yml template that I can edit

A 'Set Up Project' button is at the bottom of the dialog. The background shows the same dashboard elements as the previous screenshot, including the 'hello-world' repository and the 'Set Up Project' button.

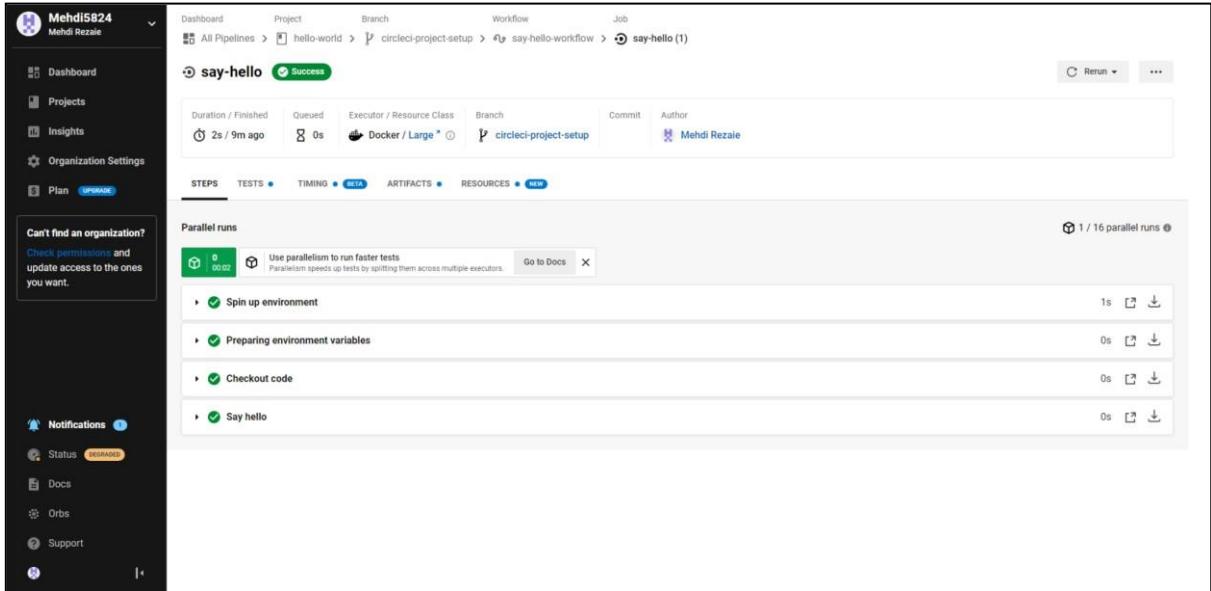
Step 3 : Your first pipeline

- On your project's pipeline page, click the green Success button, which brings you to the workflow that ran (say-helloworld).
- Within this workflow, the pipeline ran one job, called say-hello. Click say-hello to see the steps in this job:
 - Spin up environment
 - Preparing environment variables
 - Checkout code
 - Say hello
- Now select the “say-hello-workflow”

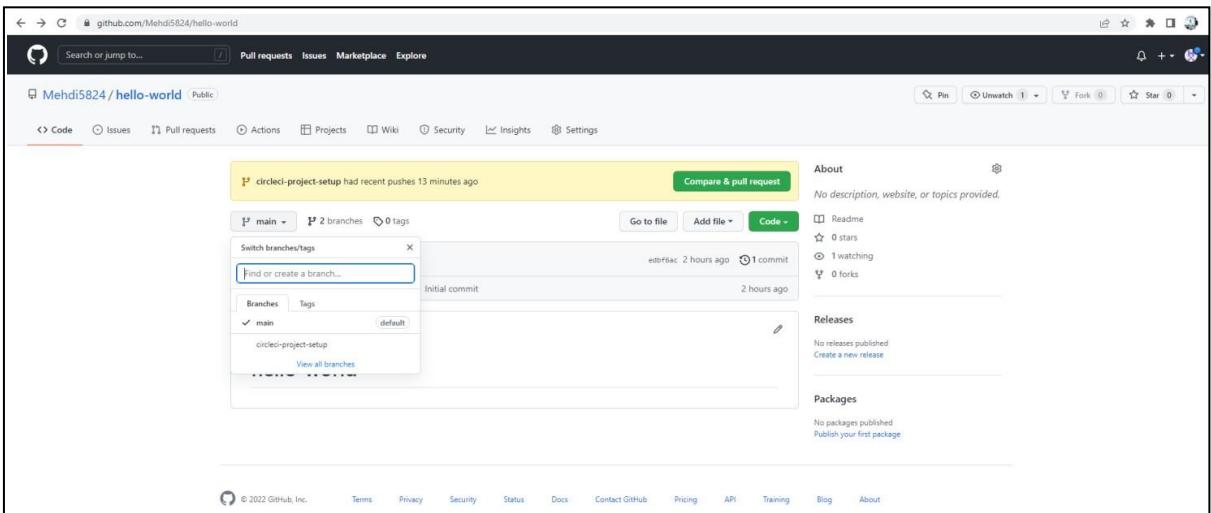
The screenshot shows the CircleCI dashboard for the 'hello-world' project. The pipeline 'hello-world' is listed with one job, 'say-hello-workflow', which has a status of 'Success'. The job was triggered by a commit from the 'master' branch with hash '6b308f0'. The pipeline was started 3m ago and took 3s. There are buttons for 'Edit Config', 'Trigger Pipeline', and 'Project Settings'.

- Select “say-hello” Job with a green tick

The screenshot shows the CircleCI job details for the 'say-hello' job in the 'say-hello-workflow'. The job status is 'Success'. The job was run 5s / 8m ago on the 'circleci-project-setup' branch, commit '6b308f0' by author 'Mehdi5824'. The job step 'say-hello' has a green success tick. The pipeline summary at the bottom indicates 20 pipelines/day, 70 min to recovery, 5 pipelines/day, and 142 min.

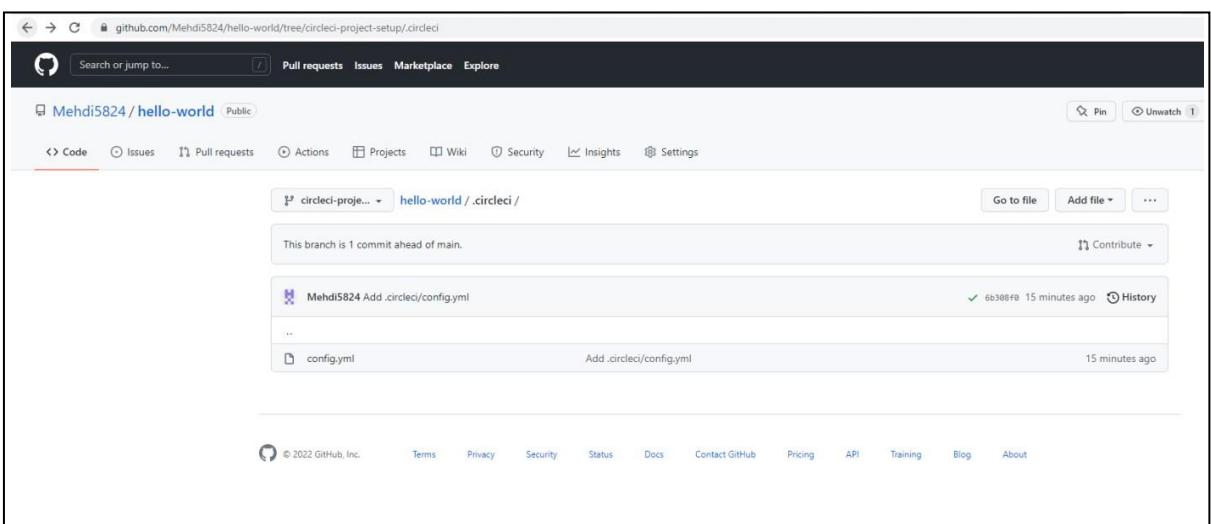
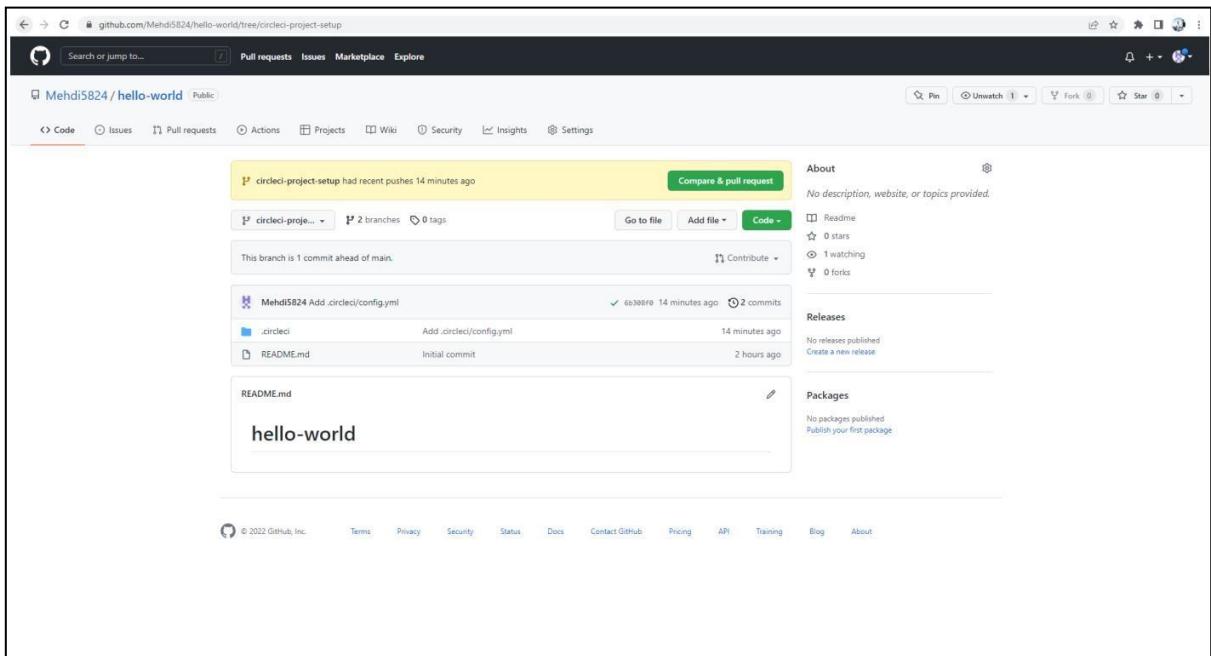


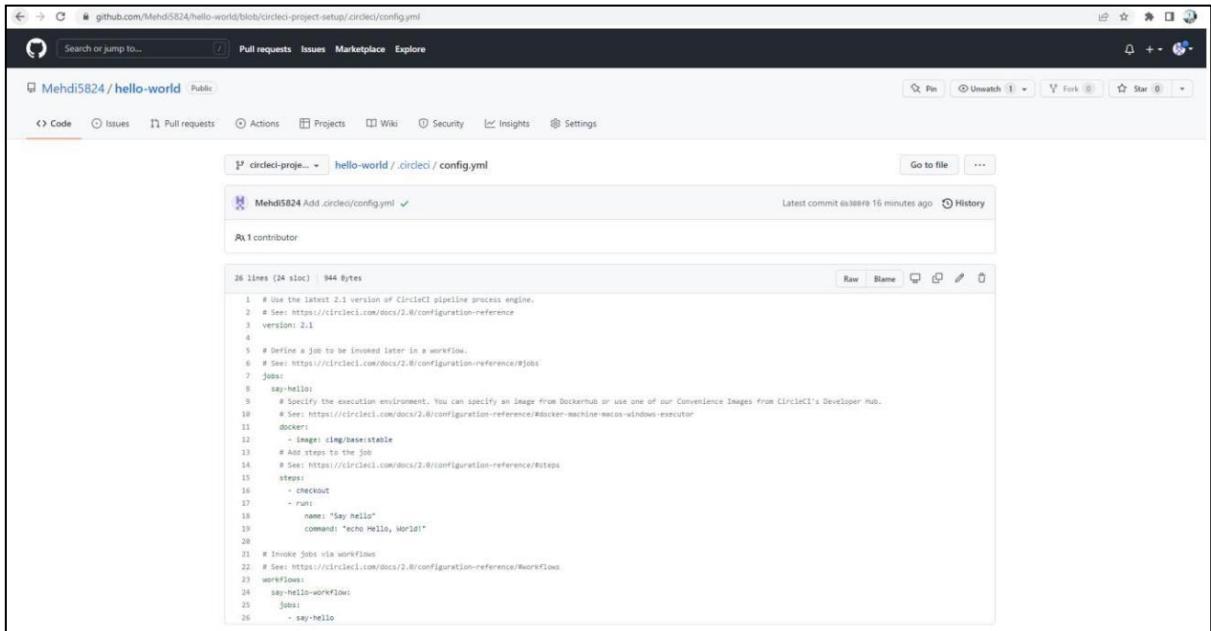
- Select Branch and option circleci-project-setup



Step 4 : Break your build

- In this section, you will edit the .circleci/config.yml file and see what happens if a build does not complete successfully.
- It is possible to edit files directly on GitHub.





The screenshot shows a GitHub repository page for 'Mehd15824/hello-world'. The 'Code' tab is selected, displaying the file 'config.yml'. The code content is as follows:

```
1 # Use the latest 2.1 version of CircleCI pipeline process engine.
2 # See: https://circleci.com/docs/2.0/configuration-reference
3 version: 2.1
4
5 # Define a job to be invoked later in a workflow.
6 # See: https://circleci.com/docs/2.0/configuration-reference/#jobs
7 jobs:
8   say-hello:
9     # Set the execution environment. You can specify an image from DockerHub or use one of our Convenience Images from CircleCI's Developer Hub.
10    # See: https://circleci.com/docs/2.0/configuration-reference/#docker-machine-macos-windows-executor
11    docker:
12      # image: circleci/node:stable
13      # Add steps to the job
14      # See: https://circleci.com/docs/2.0/configuration-reference/#steps
15      steps:
16        - checkout
17        - run:
18          name: "Say hello"
19          command: "echo Hello, World!"
20
21 # Invoke jobs via workflows
22 # See: https://circleci.com/docs/2.0/configuration-reference/#workflows
23 workflows:
24   say-hello-workflow:
25     jobs:
26       - say-hello
```

Let's use the **Node orb**. Replace the existing config by pasting the following code:

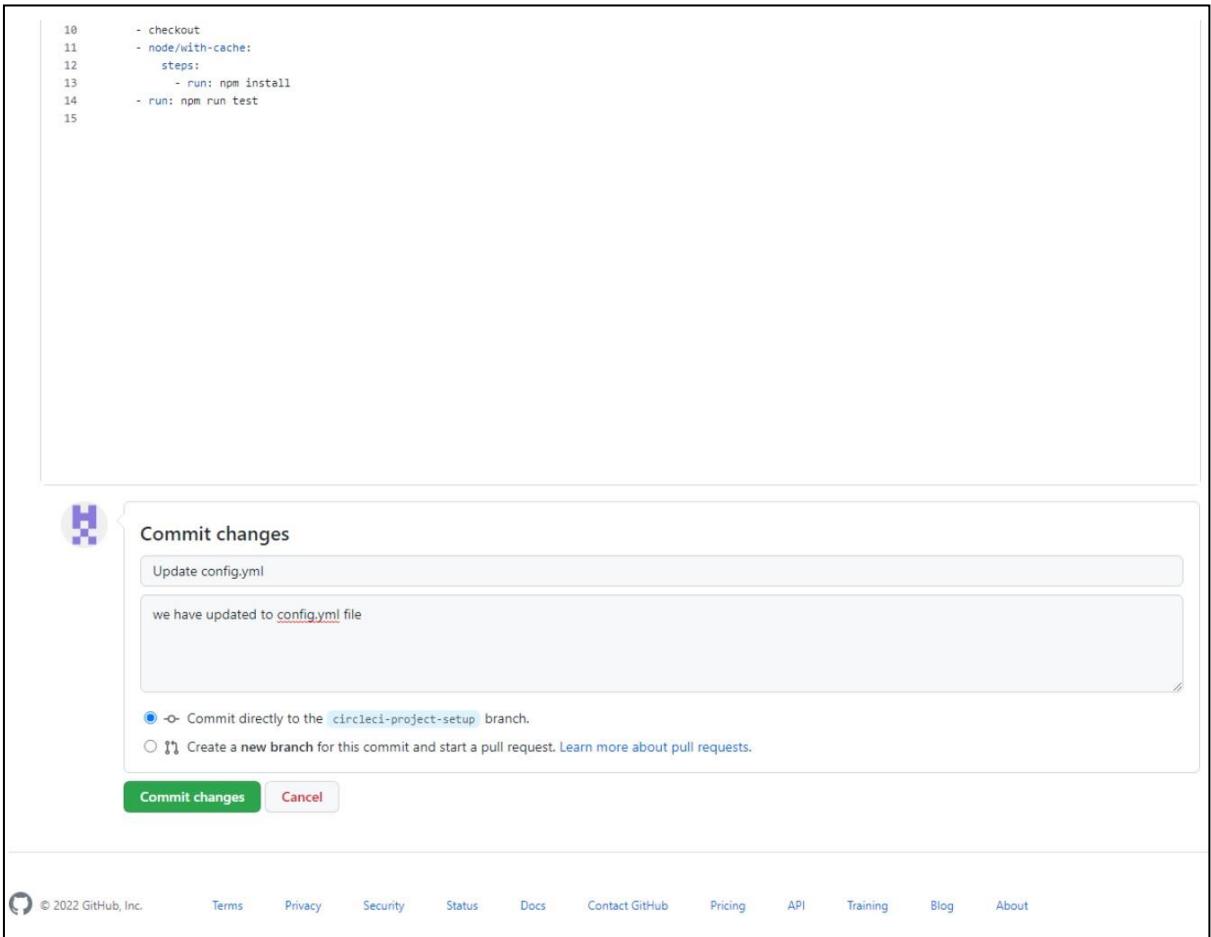
```
1  version: 2.1
2  orbs:
3    node: circleci/node@4.7.0
4  jobs:
5    build:
6      executor:
7        name: node/default
8        tag: '10.4'
9      steps:
10        - checkout
11        - node/with-cache:
12          steps:
13            - run: npm install
14            - run: npm run test
```

The GitHub file editor should look like this

A screenshot of a GitHub repository page for 'Mehdi5824/hello-world'. The user is viewing the '.circleci/config.yml' file. The code editor shows the following configuration:

```
1 version: 2.1
2 orbs:
3   node: circleci/node@4.7.0
4 jobs:
5   build:
6     executor:
7       name: node/default
8       tag: '10.4'
9     steps:
10      - checkout
11      - node/with-cache:
12        steps:
13          - run: npm install
14          - run: npm run test
15
```

Scroll down and Commit your changes on GitHub



- After committing your changes, then return to the Projects page in CircleCI. You should see a new pipeline running... and it will fail! The Node orb runs some common Node

tasks. Because you are working with an empty repository, running `npm run test`, a Node script, causes the configuration to fail. To fix this, you need to set up a Node project in your repository.

The screenshot shows the CircleCI interface. On the left, there's a sidebar with user information (MehdiRezaie), navigation links (Dashboard, Projects, Insights, Organization Settings, Plan, UPLOAD), and a message about organization access. The main area is titled "All Pipelines" and displays two entries:

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 2	Failed	Error calling workflow: 'workflow'Error calling job: 'build'Cannot find a definition for command named node/with-cache	circleci-project-setup ca5ab74 Update config.yml	3m ago	6s	
hello-world 1	Success	say-hello-workflow	circleci-project-setup 6b308f0	22m ago		

No more pipelines to load

Step 5 : Use Workflows

You do not have to use orbs to use CircleCI. The following example details how to create a custom configuration that also uses the workflow feature of CircleCI.

- Take a moment and read the comments in the code block below. Then, to see workflows in action, edit your `.circleci/config.yml` file and copy and paste the following text into it.

```

1  version: 2
2  jobs: # we now have TWO jobs, so that a workflow can coordinate them!
3    one: # This is our first job.
4      docker: # it uses the docker executor
5        - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
6        auth:
7          username: mydockerhub-user
8          password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
9        # Steps are a list of commands to run inside the docker container above.
10      steps:
11        - checkout # this pulls code down from GitHub
12        - run: echo "A first hello" # This prints "A first hello" to stdout.
13        - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
14    two: # This is our second job.
15      docker: # it runs inside a docker image, the same as above.
16        - image: cimg/ruby:3.0.2
17        auth:
18          username: mydockerhub-user
19          password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
20      steps:
21        - checkout
22        - run: echo "A more familiar hi" # We run a similar echo command to above.
23        - run: sleep 15 # and then sleep for 15 seconds.
24  # Under the workflows: map, we can coordinate our two jobs, defined above.
25  workflows:
26    version: 2
27    one_and_two: # this is the name of our workflow
28      jobs: # and here we list the jobs we are going to run.
29        - one
30        - two

```

You don't need to write the comments which are the text after #

- Commit these changes to your repository and navigate back to the CircleCI Pipelines page. You should see your pipeline running.

```

version: 2
jobs:
  one: # This is our first job.
    docker: # it uses the docker executor
      - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
      auth:
        username: mydockerhub-user
        password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
    steps:
      - checkout # this pulls code down from GitHub
      - run: echo "A first hello" # This prints "A first hello" to stdout.
      - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
  two: # This is our second job.
    docker: # it runs inside a docker image, the same as above.
      - image: cimg/ruby:3.0.2
      auth:
        username: mydockerhub-user
        password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A more familiar hi" # We run a similar echo command to above.
      - run: sleep 15 # and then sleep for 15 seconds.
workflows:
  version: 2
  one_and_two: # this is the name of our workflow
    jobs: # and here we list the jobs we are going to run.
      - one
      - two

```

- Click on the running pipeline to view the workflow you have created. You should see that two jobs ran (or are currently running!) concurrently.

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 3	Running	one_and_two	circleci-project-setup e11ad95 Update config.yml	14s ago	13s	View Logs Re-run ...
hello-world 2	Failed	Build Error	circleci-project-setup ca5ab74 Update config.yml	9m ago	6s	View Logs Re-run ...
hello-world 1	Success	say-hello-workflow	circleci-project-setup 6b308f0	29m ago	41s	View Logs Re-run ...

Duration / Finished	Branch	Commit	Author & Message
34s / 3s ago	circleci-project-setup	e11ad95	Update config.yml

Job Status	Duration
two	19s
one	31s

Step 6 : Add some changes to use workspaces

- Each workflow has an associated workspace which can be used to transfer files to downstream jobs as the workflow progresses. You can use workspaces to pass along data that is unique to this run and which is needed for downstream jobs. Try updating config.yml to the following:

```
1  version: 2
2  jobs:
3    one:
4      docker:
5        - image: cimg/ruby:3.0.2
6        auth:
7          username: mydockerhub-user
8          password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
9      steps:
10     - checkout
11     - run: echo "A first hello"
12     - run: mkdir -p my_workspace
13     - run: echo "Trying out workspaces" > my_workspace/echo-output
14     - persist_to_workspace:
15       # Must be an absolute path, or relative path from working_directory
16       root: my_workspace
17       # Must be relative path from root
18     paths:
19       - echo-output
20 two:
21   docker:
22     - image: cimg/ruby:3.0.2
23     auth:
24       username: mydockerhub-user
25       password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
26   steps:
27     - checkout
28     - run: echo "A more familiar hi"
29     - attach_workspace:
30       # Must be absolute path or relative path from working_directory
31       at: my_workspace
32
33     - run: |
34       if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
35         echo "It worked!";
36       else
37         echo "Nope!"; exit 1
38       fi
39 workflows:
40   version: 2
41   one_and_two:
42     jobs:
43       - one
44       - two:
45         requires:
46           - one
```

- Updated config.yml in GitHub file editor should be updated like this

hello-world/.circleci/ config.yml in `circleci-project-setup`

```

1 version: 2
2 jobs:
3   one:
4     docker:
5       - image: cimg/ruby:3.0.2
6         auth:
7           username: mydockerhub-user
8           password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
9       steps:
10      - checkout
11      - run: echo "A first hello"
12      - run: mkdir -p my_workspace
13      - run: echo "Trying out workspaces" > my_workspace/echo-output
14      - persist_to_workspace:
15        # Must be an absolute path, or relative path from working_directory
16        root: my_workspace
17        # Must be relative path from root
18        paths:
19          - echo-output
20   two:
21     docker:
22       - image: cimg/ruby:3.0.2
23         auth:
24           username: mydockerhub-user
25           password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
26       steps:
27      - checkout
28      - run: echo "A more familiar hi"
29      - attach_workspace:
30        # Must be absolute path or relative path from working_directory
31        at: my_workspace
32
33      - run: |
34        if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
35          echo "It worked!";

```

Commit changes

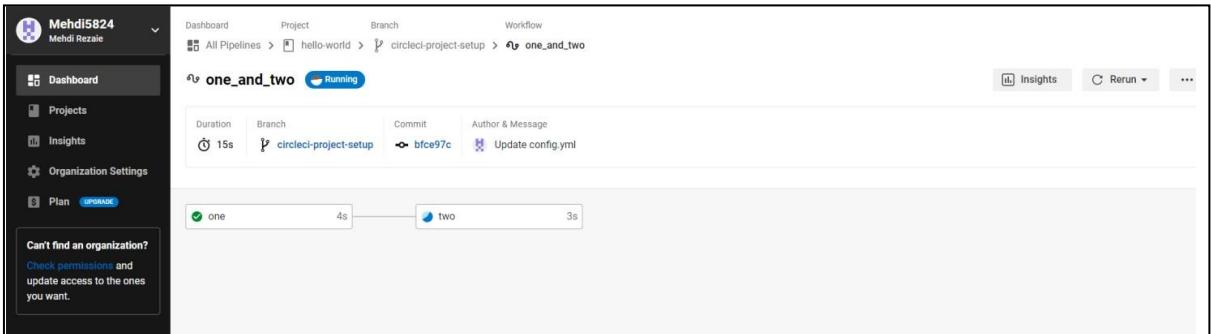
Update config.yml

3rd Update

Commit directly to the `circleci-project-setup` branch.

Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

- Finally your workflow with the jobs running should look like this



Practical No 7

Aim : Working with TeamService

Source Code :

Step 1 :

- Open command prompt and create a web api

```
C:\Windows\System32\cmd.exe

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

Processing post-creation actions...
Running 'dotnet restore' on TeamService\TeamService.csproj...
  Restore completed in 5.9 sec for D:\TeamService\TeamService.csproj.

Restore succeeded.
```

- Remove existing weatherforecast files both model and controller files.

Step 2 :

- Add new files as follows:
- Add Member.cs to “D:\TeamService\Models” folder

```
using System;
namespace TeamService.Models
{
    public class Member
    {
        public Guid ID { get; set; }

        public string FirstName { get; set; }
        public string LastName { get; set; }
        public Member() { }
        public Member(Guid id) : this()
        {
            this.ID = id;
        }
        public Member(string firstName, string lastName, Guid id) : this(id)
        {
            this.FirstName = firstName;
```

```

        this.LastName = lastName;
    }
    public override string ToString()
    {
        return this.LastName;
    }
}
}



- Add Team.cs to “D:\TeamService\Models” folder



```

using System;
using System.Collections.Generic;
namespace TeamService.Models
{
 public class Team
 {
 public string Name { get; set; }
 public Guid ID { get; set; }
 public ICollection<Member> Members { get; set; }
 public Team()
 {
 this.Members = new List<Member>();
 }
 public Team(string name) : this()
 {
 this.Name = name;
 }
 public Team(string name, Guid id) : this(name)
 {
 this.ID = id;
 }
 public override string ToString()
 {
 return this.Name;
 }
 }
}

```


```

- add TeamsController.cs file to “D:\TeamService\Controllers” folder
- ```

using System;
using Microsoft.AspNetCore.Hosting;

```

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using System.Linq;
using TeamService.Models;
using System.Threading.Tasks;
using TeamService.Persistence;
namespace TeamService
{
 [Route("[controller]")]
 public class TeamsController : Controller
 {
 ITeamRepository repository;
 public TeamsController(ITeamRepository repo)
 {
 repository = repo;
 }
 [HttpGet]
 public virtual IActionResult GetAllTeams()
 {
 return this.Ok(repository.List());
 }
 [HttpGet("{id}")]
 public IActionResult GetTeam(Guid id)
 {
 Team team = repository.Get(id);
 if (team != null)
 {
 return this.Ok(team);
 }
 else
 {
 return this.NotFound();
 }
 }
 [HttpPost]
 public virtual IActionResult CreateTeam([FromBody]Team newTeam)
 {
 repository.Add(newTeam);
 return this.Created($"/{newTeam.ID}", newTeam);
 }
 }
}
```

```

 }
 [HttpPut("{id}")]
 public virtual IActionResult UpdateTeam([FromBody]Team team,
 Guid id)
 {
 team.ID = id;
 if(repository.Update(team) == null)
 {
 return this.NotFound();
 }
 else
 {
 return this.Ok(team);
 }
 }
 [HttpDelete("{id}")]
 public virtual IActionResult DeleteTeam(Guid id)
 {
 Team team = repository.Delete(id);
 if (team == null)
 {
 return this.NotFound();
 }
 else
 {
 return this.Ok(team.ID);
 }
 }
 }
}

```

- Add MembersController.cs file to “D:\TeamService\Controllers” folder using System;  
using Microsoft.AspNetCore.Hosting;  
using Microsoft.AspNetCore.Builder;  
using Microsoft.AspNetCore.Mvc;  
using System.Collections.Generic;  
using System.Linq;  
using TeamService.Models;  
using System.Threading.Tasks;  
using TeamService.Persistence;namespace TeamService

```

{
[Route("/teams/{teamId}/{controller}")]
 public class MembersController : Controller
 {
 ITeamRepository repository;
 public MembersController(ITeamRepository repo)
 {
 repository = repo;
 }
 [HttpGet]
 public virtual IActionResult GetMembers(Guid teamID)
 {
 Team team = repository.Get(teamID);
 if(team == null)
 {
 return this.NotFound();
 }
 else
 {
 return this.Ok(team.Members);
 }
 }
 [HttpGet]
 [Route("/teams/{teamId}/{controller}/{memberId}")]
 public virtual IActionResult GetMember(Guid teamID, Guid
memberId)
 {
 Team team = repository.Get(teamID);
 if(team == null)

 {
 return this.NotFound();
 }
 else
 {
 var q = team.Members.Where(m => m.ID == memberId);
 if(q.Count() < 1)
 {
 return this.NotFound();
 }
 else

```

```

 {
 return this.Ok(q.First());
 }
}
}

[HttpPut]
[Route("/teams/{teamId}/{controller}/{memberId}")]
public virtual IActionResult UpdateMember([FromBody]Member
updatedMember, Guid teamID, Guid memberId)
{
 Team team = repository.Get(teamID);
 if(team == null)
 {
 return this.NotFound();
 }
 else
 {
 var q = team.Members.Where(m => m.ID == memberId);
 if(q.Count() < 1)
 {
 return this.NotFound();
 }
 else
 {
 team.Members.Remove(q.First());
 team.Members.Add(updatedMember);
 return this.Ok();
 }
 }
}

[HttpPost]
public virtual IActionResult CreateMember([FromBody]Member
newMember, Guid teamID)
{
 Team team = repository.Get(teamID);
 if(team == null)
 {
 return this.NotFound();
 }
 else

```

```

 {
 team.Members.Add(newMember);
 var teamMember = new {TeamID = team.ID, MemberID =
newMember.ID};
 return
this.Created($"/teams/{teamMember.TeamID}/[controller]/{teamMember.
MemberID}", teamMember);
 }
}
[HttpGet]
[Route("/members/{memberId}/team")]
public IActionResult GetTeamForMember(Guid memberId)
{
 var teamId = GetTeamIdForMember(memberId);
 if (teamId != Guid.Empty)
 {
 return this.Ok(new {TeamID = teamId });
 }
 else
 {
 return this.NotFound();
 }
}
private Guid GetTeamIdForMember(Guid memberId)
{
 foreach (var team in repository.List())
 {
 var member = team.Members.FirstOrDefault(m => m.ID ==
memberId);
 if (member != null)
 {
 return team.ID;
 }
 }
 return Guid.Empty;
}
}

```

**Step 3 :**

- Create folder “D:\TeamService\Persistence”
- Add file ITeamReposiroty.cs in “D:\TeamService\Persistence” folder
 

```
using System;
using System.Collections.Generic;
using TeamService.Models;
namespace TeamService.Persistence
{
 public interface ITeamRepository
 {
 IEnumerable<Team> List();
 Team Get(Guid id);
 Team Add(Team team);
 Team Update(Team team);
 Team Delete(Guid id);
 }
}
```
- Add MemoryTeamRepository.cs in “D:\TeamService\Persistence” folder
 

```
using System;
using System.Collections.Generic;
using System.Linq;
using TeamService;
using TeamService.Models;
namespace TeamService.Persistence
{
 public class MemoryTeamRepository : ITeamRepository
 {
 protected static ICollection<Team> teams;
 public MemoryTeamRepository()
 {
 if(teams == null)
 {
 teams = new List<Team>();
 }
 }
 public MemoryTeamRepository(ICollection<Team> teams)
 {
 MemoryTeamRepository.teams = teams;
 }
 public IEnumerable<Team> List()
```

```

 {
 return teams;
 }
 public Team Get(Guid id)
 {
 return teams.FirstOrDefault(t => t.ID == id);
 }
 public Team Update(Team t)
 {
 Team team = this.Delete(t.ID);
 if(team != null)
 {
 team = this.Add(t);
 }
 return team;
 }
 public Team Add(Team team)
 {
 teams.Add(team);
 return team;
 }
 public Team Delete(Guid id)
 {
 var q = teams.Where(t => t.ID == id);
 Team team = null;
 if (q.Count() > 0)
 {
 team = q.First();
 teams.Remove(team);
 }
 return team;
 }
}

```

#### **Step 4 :**

- Add following line to Startup.cs in public void ConfigureServices(IServiceCollection services) method  
services.AddScoped<ITeamRepository, MemoryTeamRepository>();

#### **Output:**

- Open two command prompt
- Command Prompt 1: go inside folder teamservice first

```
c:\ Command Prompt - dotnet run

D:\TeamService>dotnet run
info: Microsoft.Hosting.Lifetime[0]
 Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
 Now listening on: http://localhost:5000
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:\TeamService
```

- On Command Prompt 2:

#### To get all teams

curl --insecure <https://localhost:5001/teams>

```
D:>curl --insecure https://localhost:5001/teams
[]
```

#### To create new team

curl --insecure -H "Content-Type:application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" <https://localhost:5001/teams>

```
D:>curl --insecure -H "Content-Type:application/json" -X POST -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC\"}" https://localhost:5001/teams
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
D:>
```

#### To create one more new team

curl --insecure -H "Content-Type:application/json" -X POST -d "{\"id\":\"e12baa63-d511-417e-9e54-7aab04286281\", \"name\":\"MSC Part1\"}" <https://localhost:5001/teams>

```
D:>curl --insecure -H "Content-Type:application/json" -X POST -d "{\"id\":\"e12baa63-d511-417e-9e54-7aab04286281\", \"name\":\"MSC Part1\"}" https://localhost:5001/teams
{"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "members": []}
D:>
```

#### To get all teams

curl --insecure <https://localhost:5001/teams>

```
D:\>curl --insecure https://localhost:5001/teams
[{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}, {"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "members": []}]
D:\>
```

#### To get single team with team-id as parameter

```
curl --insecure https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

```
D:\> curl --insecure https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name": "KC", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
D:\>
```

#### To update team details (change name of first team from “KC” to “KC IT DEPT”)

```
curl --insecure -H "Content-Type:application/json" -X PUT -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC IT DEPT\"}"
https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

```
D:\>curl --insecure -H "Content-Type:application/json" -X PUT -d "{\"id\":\"e52baa63-d511-417e-9e54-7aab04286281\", \"name\":\"KC IT DEPT\"}" https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
{"name": "KC IT DEPT", "id": "e52baa63-d511-417e-9e54-7aab04286281", "members": []}
D:\>
```

#### To delete team

```
curl --insecure -H "Content-Type:application/json" -X DELETE
```

```
https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
```

```
Command Prompt
D:\>curl --insecure -H "Content-Type:application/json" -X DELETE https://localhost:5001/teams/e52baa63-d511-417e-9e54-7aab04286281
"e52baa63-d511-417e-9e54-7aab04286281"
D:\>
```

#### Confirm: with get all teams now it shows only one team (first one is deleted)

```
curl --insecure https://localhost:5001/teams
```

```
Command Prompt
D:\>curl --insecure https://localhost:5001/teams
[{"name": "MSC Part1", "id": "e12baa63-d511-417e-9e54-7aab04286281", "members": []}]
D:\>
```

# **Image Processing**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**N. G. ACHARYA & D. K. MARATHE COLLEGE**

*(Affiliated to University of Mumbai)*

**MUMBAI – MAHARASHTRA - 400071**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that Mohite Sayali Shyam bearing Seat No: 4133164 submitted journal of **Image Processing** techquie in partial fulfillment of the requirements for the award of Degree of **MASTER OF SCIENCE** in **INFORMATION TECHNOLOGY** from University of Mumbai.

**Internal Guide**

**Coordinator**

**External Examiner**

**Date:**

**College Seal**

## INDEX

| Sr. no | Table of Content                                        | Date | Sign |
|--------|---------------------------------------------------------|------|------|
| 1      | Image sampling and quantization                         |      |      |
| 2      | Analysis of special and intensity resolution of images. |      |      |
| 3      | Intensity transformation of images                      |      |      |
| 4      | DFT analysis of images                                  |      |      |
| 5a     | Walsh Transform                                         |      |      |
| 5b     | Hadamard Transform                                      |      |      |
| 5c     | DCT Transform                                           |      |      |
| 5d     | Harr Transform                                          |      |      |
| 6      | Histogram processing                                    |      |      |
| 7      | Image Enhancement-Special filtering                     |      |      |
| 8      | Image Enhancement-Filtering in frequency domain         |      |      |
| 9a     | Edge detection                                          |      |      |
| 9b     | Line detection                                          |      |      |
| 9c     | Point detection                                         |      |      |
| 10     | Basic Morphological operation                           |      |      |

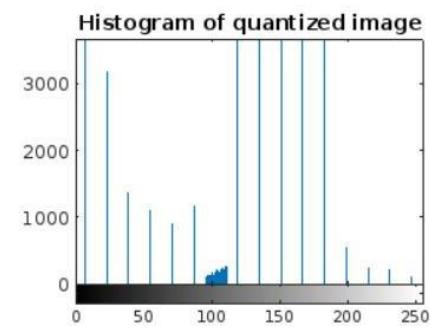
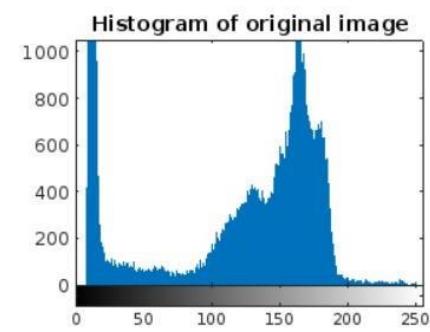
## Practical 1

Aim : Image sampling and quantization

Code :

```
a=imread('cameraman.tif');
subplot(2,2,1)
imshow(a);
title('Original image');
subplot(2,2,2);
imhist(a);
title('Histogram of original image');
[m n]=size(a);
for i=1:1:m
 for j=1:1:n
 if a(i,j)<16 a(i,j)=7;
 elseif a(i,j)>=16 && a(i,j)<32 a(i,j)=23;
 elseif a(i,j)>=32 && a(i,j)<48 a(i,j)=39;
 elseif a(i,j)>=48 && a(i,j)<64 a(i,j)=55;
 elseif a(i,j)>=64 && a(i,j)<80 a(i,j)=71;
 elseif a(i,j)>=80 && a(i,j)<96 a(i,j)=87;
 elseif a(i,j)>=96 && a(i,j)<96 a(i,j)=103;
 elseif a(i,j)>=112 && a(i,j)<128 a(i,j)=119;
 elseif a(i,j)>=128 && a(i,j)<144 a(i,j)=135;
 elseif a(i,j)>=144 && a(i,j)<160 a(i,j)=151;
 elseif a(i,j)>=160 && a(i,j)<176 a(i,j)=167;
 elseif a(i,j)>=176 && a(i,j)<192 a(i,j)=183;
 elseif a(i,j)>=192 && a(i,j)<208 a(i,j)=199;
 elseif a(i,j)>=208 && a(i,j)<224 a(i,j)=215;
 elseif a(i,j)>=224 && a(i,j)<240 a(i,j)=231;
 elseif a(i,j)>=240 && a(i,j)<256 a(i,j)=247;
 end
end
end
subplot(2,2,3)
imshow(a);
title('Quantised image')
subplot(2,2,4)
imhist(a);
title('Histogram of quantized image')
```

**Output :**



## Practical 2

**Aim : Analysis of special and intensity of resolution**

**Spacial resolution :**

**Code :**

```
z=imread('cameraman.tif');
z=imresize(z,[1024,1024]);
[r c]=size(z);
l=1;
for i=1:r
k=1;
for j=1:c
a(l,k)=z(i,j);
k=k+1;
end
l=l+1;
end
l=1;
for i=1:4:r
k=1;
for j=1:4:c
b(l,k)=z(i,j);
k=k+1;
end
l=l+1;
end
l=1;
for i=1:8:r
k=1;
for j=1:8:c
c(l,k)=z(i,j);
k=k+1;
end
l=l+1;
end
l=1;
for i=1:16:r
k=1;
for j=1:16:c
d(l,k)=z(i,j);
k=k+1;
end
l=l+1;
end
subplot(2,2,1),imshow(a)
```

```
subplot(2,2,2),imshow(b)
subplot(2,2,3),imshow(e)
subplot(2,2,4),imshow(d)
```

**Output :**



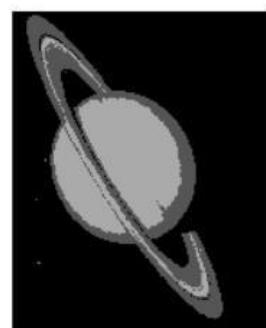
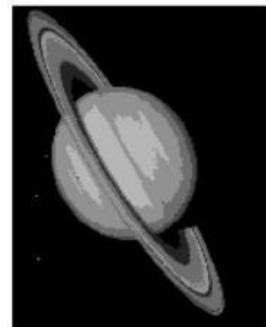
### Intensity resolution

**Code :**

```
I=imread('saturn.png');
I=rgb2gray(I);
[I256,map256]=gray2ind(I,256);
subplot(2,2,1);
imshow(I256,map256);
[I128,map128]=gray2ind(I,128);
subplot(2,2,2);
imshow(I128,map128);
[I64,map64]=gray2ind(I,64)
subplot(2,2,3);
imshow(I64,map64);
[I32,map32]=gray2ind(I,32);
subplot(2,2,4);
imshow(I32,map32);
```

```
[I16,map16]=gray2ind(I,16);
figure,
subplot(2,2,1);
imshow(I16,map16);
[I8,map8]=gray2ind(I,8);
subplot(2,2,2);
imshow(I8,map8);
[I4,map4]=gray2ind(I,4);
subplot(2,2,3);
imshow(I4,map4);
[I2,map2]=gray2ind(I,2);
subplot(2,2,4);
imshow(I2,map2);
```

**Output :**



### Practical 3

**Aim : Information transformation of images**

#### 1. photographic negative

**Code :**

```
I=imread('cameraman.tif');
imshow(I)
J=imcomplement(I);
figure, imshow(J)
```

**Output :**



#### 2. Gamma transformation

**Code :**

```
I=imread('tire.tif');
subplot(2,2,1);
imshow(I)
J=imadjust(I,[],[],1);
J2=imadjust(I,[],[],3);
J3=imadjust(I,[],[],0.4);
subplot(2,2,2);
imshow(J);
subplot(2,2,3);
imshow(J2);
subplot(2,2,4);
imshow(J3);
```

**Output :**



### **3. Logarithmic transformation**

**Code :**

```
tire = imread("tire.tif");
d = im2double(tire);
figure, imshow(d);
f = d;
c = 1/log(1+1);
j1 = c*log(1+f);
figure, imshow(j1);
f = d*255;
c = 1/log(1+255);
j2 = c*log(1+f);
figure, imshow(j2);
f = d*2^16;
c = 1/log(1+2^16);
j3 = c*log(1+f);
```

```
figure, imshow(j3);
```

**Output :**



#### 4. Contrast stretching with changing E

**Code :**

```
I=imread('tire.tif');
I2=im2double(I);
m=mean2(I2)
contrast1=1./(1+(m./I2+eps).^4);
contrast2=1./(1+(m./I2+eps).^5);
contrast3=1./(1+(m./I2+eps).^10);
imshow(I2)
figure,imshow(contrast1)
figure,imshow(contrast2)
figure,imshow(contrast3)
```

**Output :**



### 5. Contrast stretching with changing m

**Code :**

```
I = imread('tire.tif');
I2 = im2double(I);
contrast1 = 1./ (1 + (0.2 ./ (I2 + eps)).^4);
contrast2 = 1./ (1 + (0.5 ./ (I2 + eps)).^4);
contrast3 = 1./ (1 + (0.7 ./ (I2 + eps)).^4);
imshow(I2)
figure, imshow(contrast1)
figure, imshow(contrast2)
figure, imshow(contrast3)
```

**Output :**

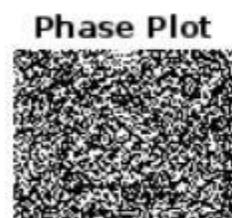
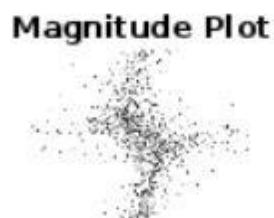
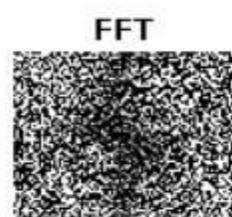


## Practical 4

**Aim : DFT analysis of image**  
**Code :**

```
a=imread('coins.png');
subplot(2,3,1);
imshow(a);
title('Original');
b=im2double(a);
c=fft2(b);
subplot(2,3,2);
imshow(c);
title('FFT');
d=ifft2(c);
subplot(2,3,3);
imshow(d);
title('IFFT');
mag=abs(c);
subplot(2,3,4);
imshow(mag);
title('Magnitude Plot');
ang=angle(c);
subplot(2,3,5);
imshow(ang);
title('Phase Plot');
```

**Output :**

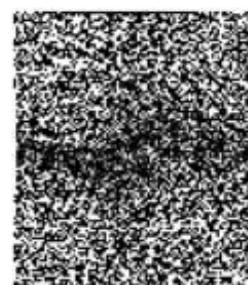
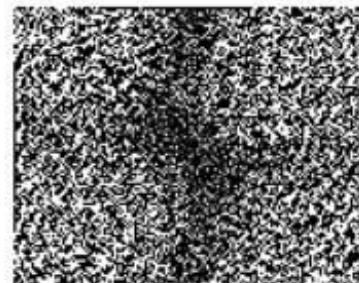


## 2. Rotation property

Code :

```
a=imread('coins.png');
subplot(2,2,1);
imshow(a);
a1=im2double(a);
b=fft2(a1);
subplot(2,2,2);
imshow(b);
c=imrotate(a1,90);
subplot(2,2,3);
imshow(c);
d=fft2(c);
subplot(2,2,4);
imshow(d);
```

Output :



## Practical 5

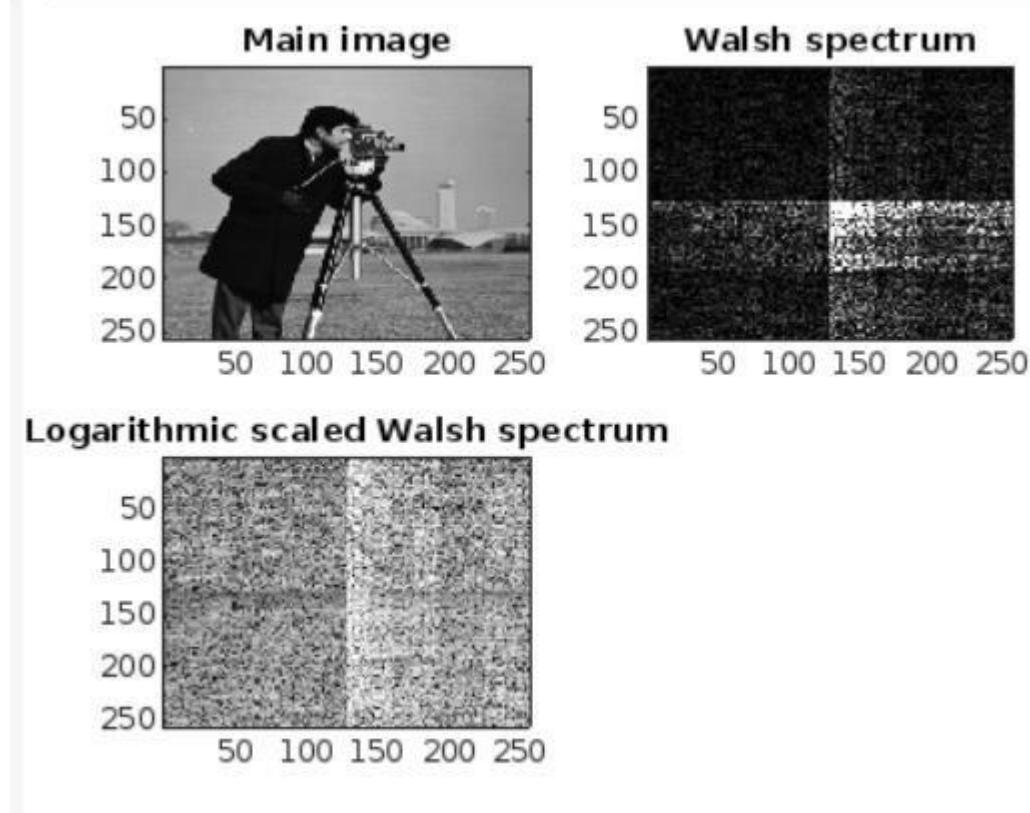
### a. walsh transformation

Code :

```
a=imread('cameraman.tif');
N=length(a);
n=log2(N);
n=1+fix(n);
f=ones(N,N);
for x=1:N;
for u=1:N
p=dec2bin(x-1,n);
q=dec2bin(u-1,n);
for i=1:n;
f(x,u)=f(x,u)*((-1)^(p(n+1-i)*q(i)));
end;
end;
end;
F=(1/N)*f*double(a)*f;
for i=1:N/2; for j=1:N/2
G(i+N/2,j+N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=1:N/2
G(i-N/2,j+N/2)=F(i,j);
end;
end
for i=1:N/2;
for j=N/2+1:N
G(i+N/2,j-N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=N/2+1:N;
G(i-N/2,j-N/2)=F(i,j);
end;
end;
H=log(1+abs(G));
for i=1:N
H(i,:)=H(i,:)*255/abs(max(H(i,:)));
end
colormap(gray(255));
subplot(2,2,1),image(a),title('Main image');
subplot(2,2,2),image(abs(G)),title('Walsh spectrum');
```

```
subplot(2,2,3),image(H),title('Logarithmic scaled Walsh spectrum');
```

**Output :**



### b. Hadamard transformation

**Code :**

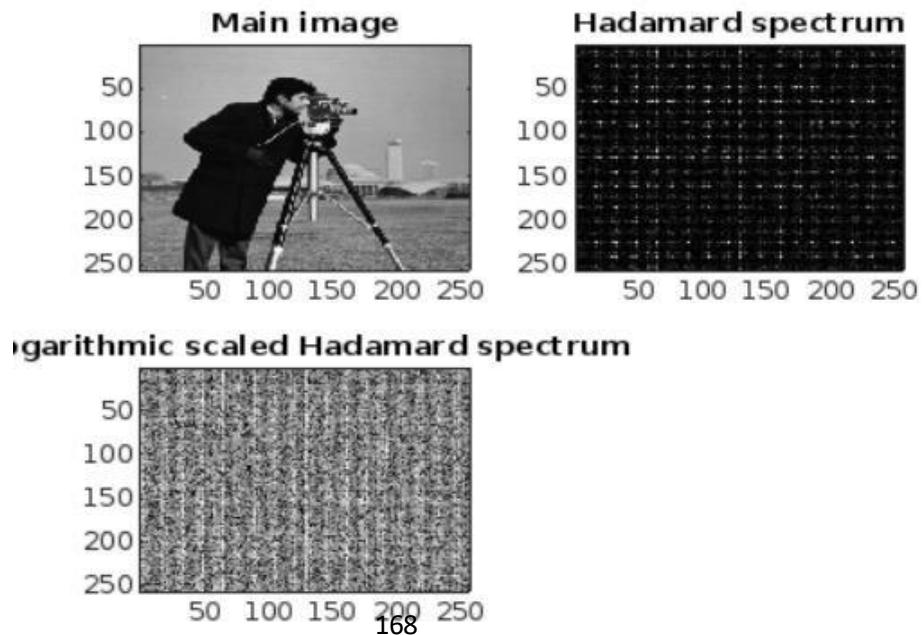
```
a=imread('cameraman.tif');
N=length(a);
n=log2(N);
n=1+fix(n);
f=ones(N,N);
for x=1:N;
for u=1:N
p=dec2bin(x-1,n);
q=dec2bin(u-1,n);
for i=1:n;
f(x,u)=f(x,u)*((-1)^(p(n+1-i)*q(n+1-i)));
end;
end;
F=(1/N)*f*double(a)*f;
for i=1:N/2;
```

```

for j=1:N/2
G(i+N/2,j+N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=1:N/2
G(i-N/2,j+N/2)=F(i,j);
end;
end
for i=1:N/2;
for j=N/2+1:N
G(i+N/2,j-N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=N/2+1:N;
G(i-N/2,j-N/2)=F(i,j);
end;
end;
H=log(1+abs(G));
for i=1:N
H(i,:)=H(i,:)*255/abs(max(H(i,:)));
end
colormap(gray(255));
subplot(2,2,1),image(a),title('Main image');
subplot(2,2,2),image(abs(G)),title('Hadamard spectrum');
subplot(2,2,3),image(H),title('Logarithmic scaled Hadamard spectrum');

```

**Output :**



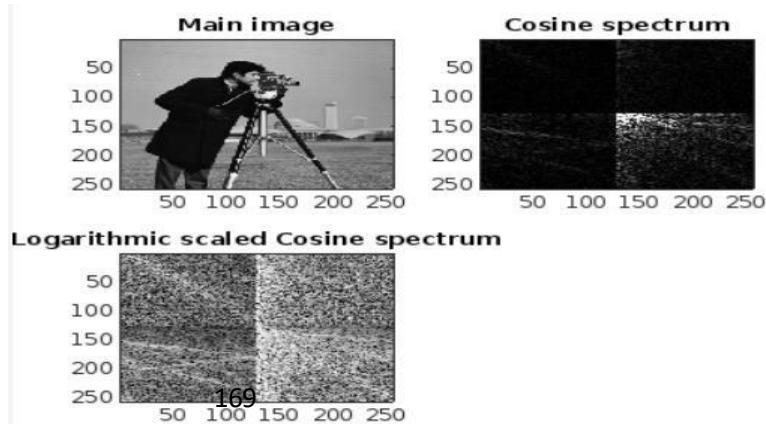
### c. Descrete cosine transformation

Code :

```
a=imread('cameraman.tif');
N=length(a);
F=dct2(double(a));
for i=1:N/2;
for j=1:N/2
G(i+N/2,j+N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=1:N/2
G(i-N/2,j+N/2)=F(i,j);
end;
end
for i=1:N/2;
for j=N/2+1:N
G(i+N/2,j-N/2)=F(i,j);
end;
end
for i=N/2+1:N;
for j=N/2+1:N
G(i-N/2,j-N/2)=F(i,j);
end;
end;
H=log(1+abs(G));
for i=1:N
H(i,:)=H(i,:)*255/abs(max(H(i,:)));
end
colormap(gray(255));
subplot(2,2,1),image(a),title('Main image');
subplot(2,2,2),image(abs(G)),title('Cosine spectrum');
subplot(2,2,3),image(H),title('Logarithmic scaled Cosine spectrum');
```

Output :

### d. Farr transformatio

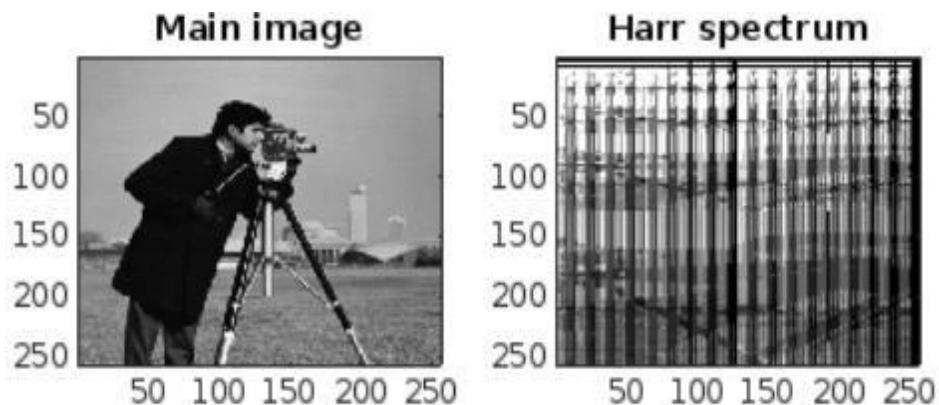


**n**

**Code :**

```
a=imread('cameraman.tif');
N=length(a);
for i=1:N;
p=fix(log2(i));
q=i-(2^p);
for j=1:N
z=(j-1)/N;
if(z>=(q-1)/(2^p))&&(z<(q-1/2)/2^p)
f(i,j)=(1/(sqrt(N)))*(2^(p/2));
elseif(z>=(q-1)/(2^p))&&(z<(q/2)/2^p)
f(i,j)=(1/(sqrt(N)))*(-2^(p/2));
else f(i,j)=0;
end;
end;
F=f*double(a)*f
colormap(gray(255));
subplot(2,2,1),image(a),title('Main image');
subplot(2,2,2),image(abs(F)),title('Harr spectrum');
```

**Output :**



## Practical 6

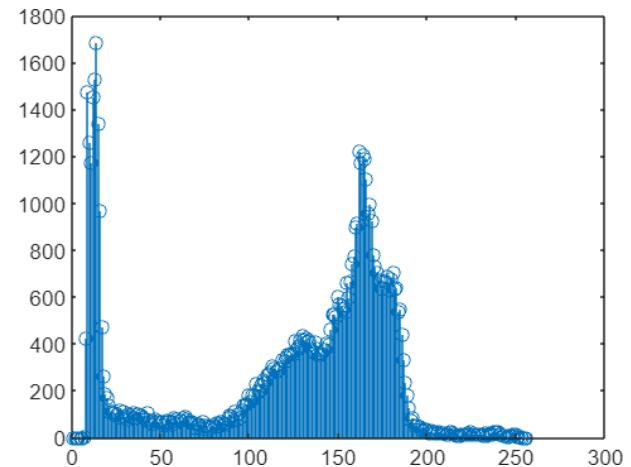
**Aim : To study the histogram and histogram equalization**

**Histogram without inbuilt function**

**Code :**

```
histo=zeros(1,256);
I=imread('cameraman.tif');
imshow(I);
si=size(I);
for i=1:si(1)
 for j=1:si(2)
 for g=1:256
 if I(i,j)==g
 histo(g)=histo(g)+1;
 end
 end
 end
end
figure,stem(histo)
```

**Output :**

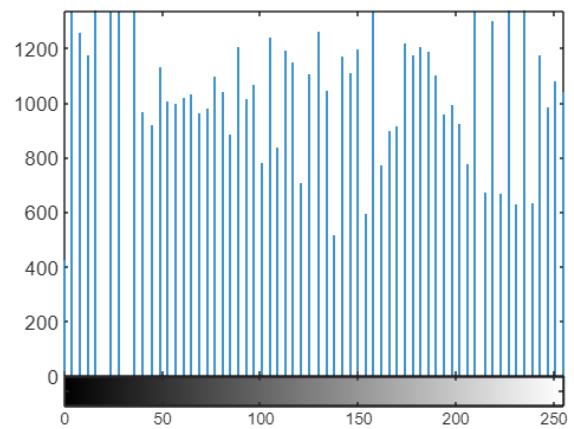


### 2. histogram equalization

**Code :**

```
I=imread('cameraman.tif');
a=histeq(I);
imshow(a);
figure,imhist(a)
```

**Output :**



## Practical 7

Aim : To perform image enhancement by spacial filtering

a. Average

Code :

```
i=imread('cameraman.tif');
imshow(i);
w=fspecial('average',[3 3]);
g=imfilter(i,w,'symmetric');
figure,imshow(g,[])
```

Output :



b. Guassian

code :

```
i=imread('cameraman.tif');
w=fspecial('gaussian',[3 3],0.5);
g=imfilter(i,w,'symmetric');
imshow(g,[])
```

Output :

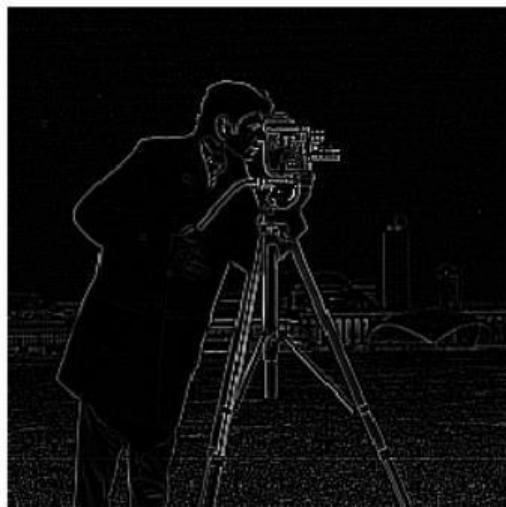


c. Laplacian

**Code :**

```
i=imread('cameraman.tif');
w=fspecial('laplacian', 0.5);
g=imfilter(i,w,'symmetric');
imshow(g,[])
```

**Output :**



**d. sobel**

**Code :**

```
i=imread('cameraman.tif');
w=fspecial('sobel');
g=imfilter(i,w,'symmetric');
imshow(g,[])
```

**Output :**



**e. non linear order static filter**

**Code :**

```
i=imread('cameraman.tif');
h=ordfilt2(i,1,ones(3,3));
h1=ordfilt2(i,3*3,ones(3,3));
h2=ordfilt2(i,median(1:3*3),ones(3,3));
subplot(2,2,1)
imshow(i);
subplot(2,2,2)
imshow(h,[]);
subplot(2,2,3)
imshow(h1,[]);
subplot(2,2,4)
imshow(h2,[]);
```

**Output :**



**f. Median filter**

**Code :**

```
g=imread('cameraman.tif');
m=medfilt2(g,[3 3]);
imshow(m,[]);
```

**Output :**



## Practical 8

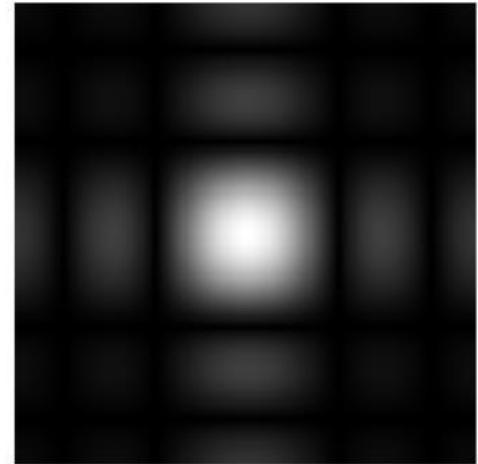
Aim : To obtain frequency domain filters from spacial domain

a. average

Code :

```
f=imread('cameraman.tif');
h=fspecial('average',[5 5]);
Fs=size(f);
F=fft2(f);
H=freqz2(h,Fs(1),Fs(2));
G=F.*H;
g=ifft2(G);
imshow(real(g),[]);
figure,imshow(abs(H));
```

Output :

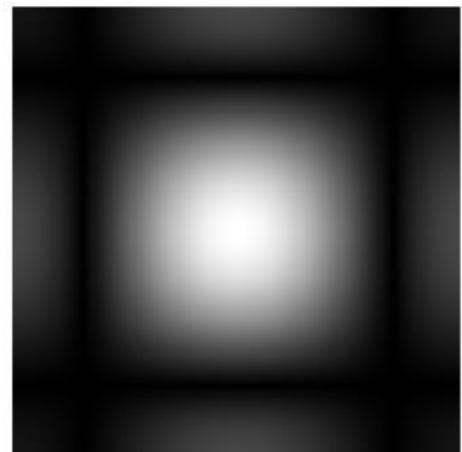


b. Guassian

Code :

```
f=imread('cameraman.tif');
h=fspecial('gaussian',[3 3],2);
Fs=size(f);
F=fft2(f);
H=freqz2(h,Fs(1),Fs(2));
G=F.*H;
g=ifft2(G);
imshow(real(g),[]);
figure,imshow(abs(H));
```

Output :

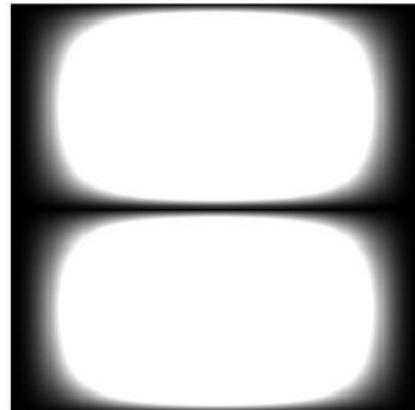


### c. Sobel

**Code :**

```
f=imread('cameraman.tif');
h=fspecial('sobel');
Fs=size(f);
F=fft2(f);
H=freqz2(h,Fs(1),Fs(2));
G=F.*H;
g=ifft2(G);
imshow(real(g),[]);
figure,imshow(abs(H));
```

**Output :**



### 4b. To generate filters directly in the frequency domain'

**a. Butterworth LowPass filter**

**Code :**

```
clear;
clc;
```

```

img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
for j=1:Y
d(i,j)=sqrt((i-x).^2+(j-y).^2);
h(i,j)=1/(1+((d(i,j))/rad).^(2*N));
end
end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After LPF');
subplot(2,2,3);
surf(h);
title('LPF in 3D');
subplot(2,2,4);
imshow(h);
title('LPF as Image');

```

**b. Butterworth high pass :**

**Code :**

```

clear;
clc;
img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
for j=1:Y
d(i,j)=sqrt((i-x).^2+(j-y).^2);
h(i,j)=1/(1+(rad/d(i,j)).^(2*N));
end

```

```

end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After HPF');
subplot(2,2,3);
surf(h);
title('HPF in 3D');
subplot(2,2,4);
imshow(h);
title('HPF as Image');

```

**c. Guassian low pass :**

**Code :**

```

clear;
clc;
img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
 for j=1:Y
 d(i,j)=sqrt((i-x).^2+(j-y).^2);
 h(i,j)=exp(-(d(i,j).^2)/(2*((rad).^2)));
 end
end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After Gaussian LPF');

```

```
subplot(2,2,3);
surf(h);
title('Gaussian LPF in 3D');
subplot(2,2,4);
imshow(h);
title('Gaussian LPF as Image');
```

**d. Gussian high pass filter :**

**code :**

```
clear;
clc;
img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
 for j=1:Y
 d(i,j)=sqrt((i-x).^2+(j-y).^2);
 h(i,j)=1-exp(-(d(i,j).^2)/(2*((rad).^2)));
 end
end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After Gaussian HPF');
subplot(2,2,3);
surf(h);
title('Gaussian HPF in 3D');
subplot(2,2,4);
imshow(h);
title('Gaussian HPF as Image');
```

**e. Ideal Low Pass filter :**

**Code :**

```
clear;
clc;
```

```

img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
for j=1:Y
d(i,j)=sqrt((i-x).^2+(j-y).^2);
h(i,j)=double(d(i,j)<=rad);
end
end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After LPF');
subplot(2,2,3);
surf(h);
title('LPF in 3D');
subplot(2,2,4);
imshow(h);
title('LPF as Image');

```

#### **f. Ideal high pass filter :**

**Code :**

```

clear;
clc;
img=imread('Coins.png');
[X,Y]=size(img);
N=input('Order of Filter=');
x=ceil(X/2);
y=ceil(Y/2);
rad=26;
for i=1:X
for j=1:Y
d(i,j)=sqrt((i-x).^2+(j-y).^2);
h(i,j)=double(d(i,j)>rad);
end
end

```

```
end
fft1=fftshift(fft2(img));
fil=h.*fft1;
fin=ifft2(fil);
fin1=uint8(fin);
subplot(2,2,1);
imshow(img);
title('Original');
subplot(2,2,2);
imshow(fin1);
title('After HPF');
subplot(2,2,3);
surf(h);
title('HPF in 3D');
subplot(2,2,4);
imshow(h);
title('HPF as Image');
```

## Practical 9a

**Aim : To detect edges in image**

**Code :**

```
a=imread('office_2.jpg');
imshow(a);
f=rgb2gray(a);
figure,imshow(f);
[g,t]=edge(f,'sobel','vertical');
figure,subplot(3,1,1);
imshow(g);
[g,t]=edge(f,'sobel','horizontal');
subplot(3,1,2);
imshow(g);
[g,t]=edge(f,'sobel','both');
subplot(3,1,3);
imshow(g);
[g,t]=edge(f,'prewitt','vertical');
figure,subplot(3,1,1);
imshow(g);
[g,t]=edge(f,'prewitt','horizontal');
subplot(3,1,2);
imshow(g);
[g,t]=edge(f,'prewitt','both');
subplot(3,1,3);
imshow(g);
[g,t]=edge(f,'roberts','vertical');
figure,subplot(3,1,1);
imshow(g);
[g,t]=edge(f,'roberts','horizontal');
subplot(3,1,2);
imshow(g);
[g,t]=edge(f,'roberts','both');
subplot(3,1,3);
imshow(g);
[g,t]=edge(f,'canny','vertical');
figure,subplot(3,1,1);
imshow(g);
[g,t]=edge(f,'canny','horizontal');
subplot(3,1,2);
imshow(g);
[g,t]=edge(f,'canny','both');
subplot(3,1,3);
imshow(g);
```

**Output :**

Original image :



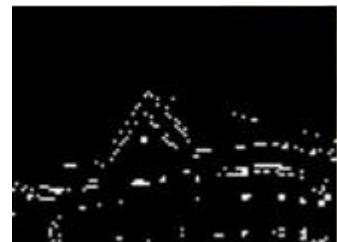
Sobel-vertical



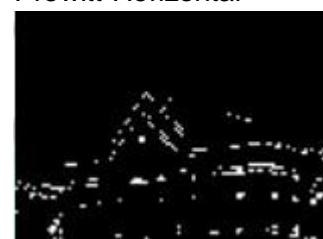
Prewitt- vertical



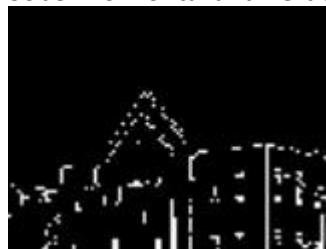
Sobel-Horizontal



Prewitt-Horizontal



Sobel-Horizontal and Vertical



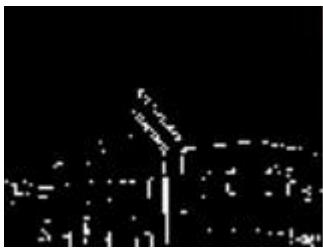
Prewitt-Horizontal and vertical



Robert-vertical



Robert-Horizontal



Robert-Horizontal and vertical



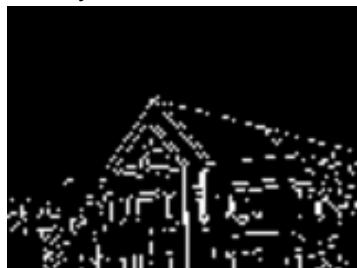
Canny- vertical



Canny-Horizontal



Canny-Horizontal and vertical



## Practical 9b

**Aim : To detect lines in the image using matlab**

**Horizontal lines**

**Code :**

```
a=imread('line.jpg');
f=rgb2gray(a);
imshow(f);
w=[-1,-1,-1;2,2,2;-1,-1,-1]
g=abs(imfilter(double(f),w));
T=300;
g=g>=T;
figure,imshow(g);
```

**Vertical lines :**

```
a=imread('line.jpg');
f=rgb2gray(a);
imshow(f);
w=[-1,2,-1;-1,2,-1;-1,2,-1]
g=abs(imfilter(double(f),w));
T=300;
g=g>=T;
figure,imshow(g);
```

**45 degree lines :**

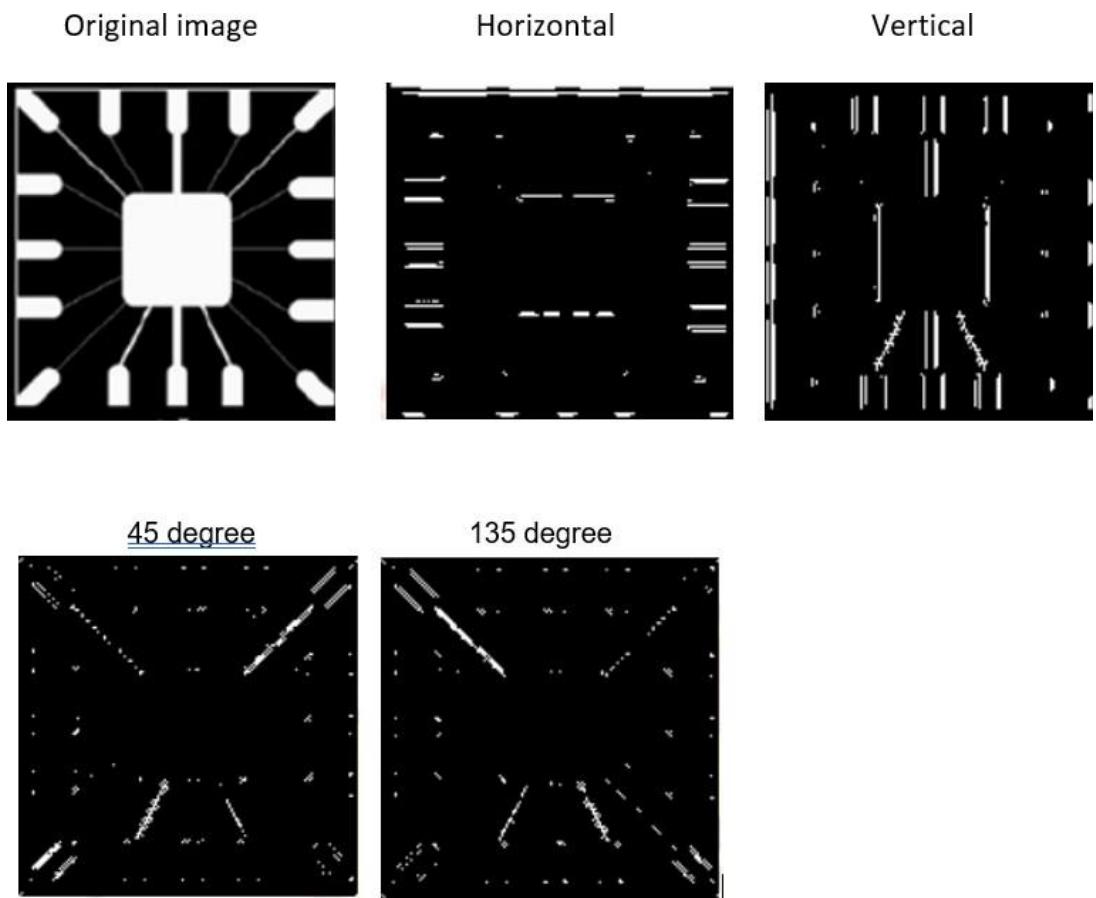
```
a=imread('line.jpg');
f=rgb2gray(a);
imshow(f);
w=[-1,-1,2;-1,2,-1;2,-1,-1]
g=abs(imfilter(double(f),w));
T=300;
g=g>=T;
figure,imshow(g);
```

**135 degree**

```
a=imread('line.jpg');
f=rgb2gray(a);
imshow(f);
w=[2,-1,-1;-1,2,-1;-1,-1,2]
g=abs(imfilter(double(f),w));
T=300;
g=g>=T;
```

```
figure,imshow(g);
```

**Output :**



## Practical 9c

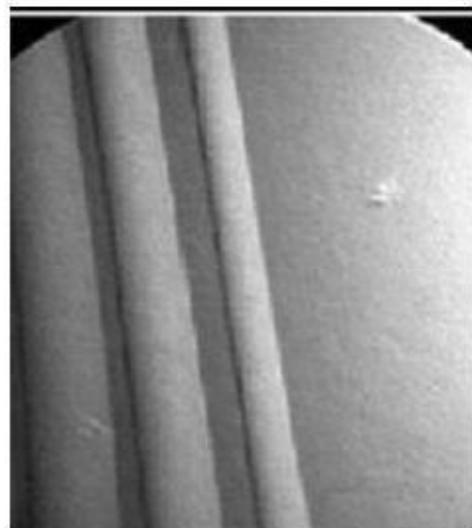
**Aim : To detect points in an image**

**Code :**

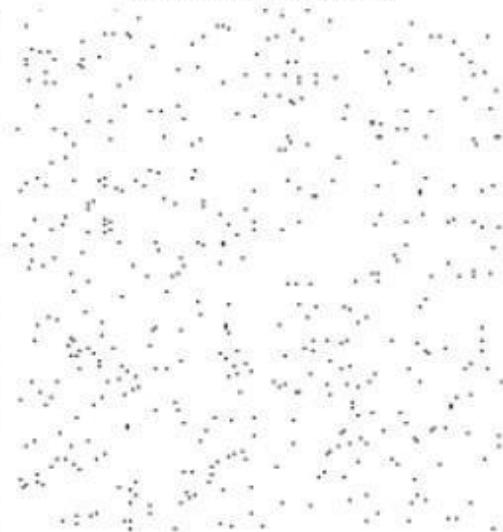
```
a=imread('point.jpg');
f=rgb2gray(a);
imshow(f);
w=[-1,-1,-1;-1,8,-1;-1,-1,-1];
g=abs(imfilter(double(f),w));
imshow(g);
```

**Output :**

Input image



Point detection



## Practical 10

Aim : To perform morphological operations on image using matlab.

Code :

```
imread('coins.png');
b=strel('disk',10);
c=imdilate(a,b);
figure;
subplot(2,2,1);
imshow(c);
title('dilation using disk stereo element');
b=strel('square',10);
c=imdilate(a,b);
subplot(2,2,2);
imshow(c);
title('dilation using square stereo element');
b=strel('disk',10);
c=imerode(a,b);
subplot(2,2,3);
imshow(c);
title('erosion using disk stereo element');
b=strel('square',10);
c=imerode(a,b);
subplot(2,2,4);
imshow(c);
title('erosion using square stereo element');
bw=im2bw(a);
figure;
subplot(2,2,1);
imshow(bw);
title('original image');
b=strel('disk',5);
c=imopen(a,b);
subplot(2,2,2);
imshow(c);
title('image after opening');
b=strel('disk',5);
c=imclose(a,b);
subplot(2,2,3);
imshow(c);
title('image after closing');
a=imread('rice.png');
b=im2bw(a);
c=bwmorph(b,'remove');
figure;
```

```

subplot(2,3,1);
imshow(c);
title('image remove');
c=bwmorph(b,'clean');
subplot(2,3,2);
imshow(c);
title('image clean');
c=bwmorph(b,'shrink');
subplot(2,3,3);
imshow(c);
title('image shrink');
c=bwmorph(b,'fill');
subplot(2,3,4);
imshow(c);
title('image fill');
c=bwmorph(b,'thin');
subplot(2,3,5);
imshow(c);
title('image thin');
c=bwmorph(b,'thick');
subplot(2,3,6);
imshow(c);
title('image thick');

```

**Output :**

Dilation using disk stereo element



Original image

Dilation using square stereo element

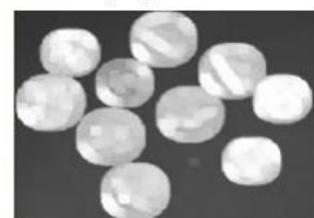
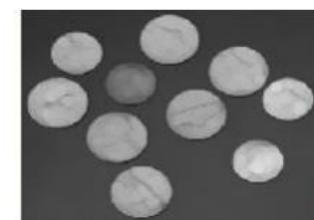
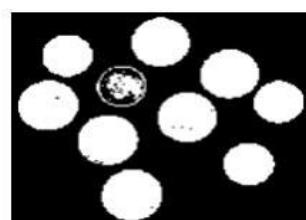
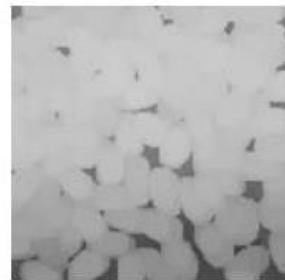


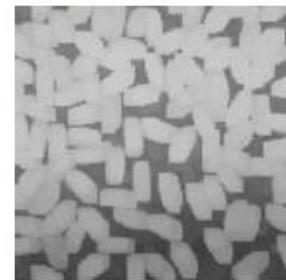
image after opening



dilation using disk stereo element



dilation using square stereo element



original image



image after opening

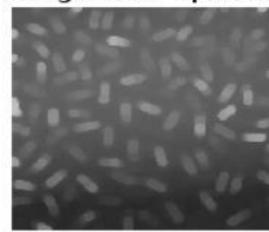


image after closing

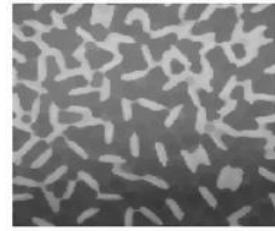


image remove

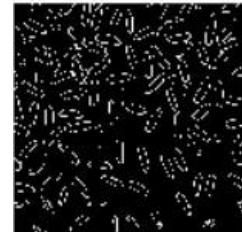


image clean



image shrink



image fill



image thin



image thick

