Academic Year: 2024-25 **ter: V Class / Branch: TE IT**
Subject: DevOPs Lab (DL)
Subject Lab In-charge: Prof. Sujata Oak

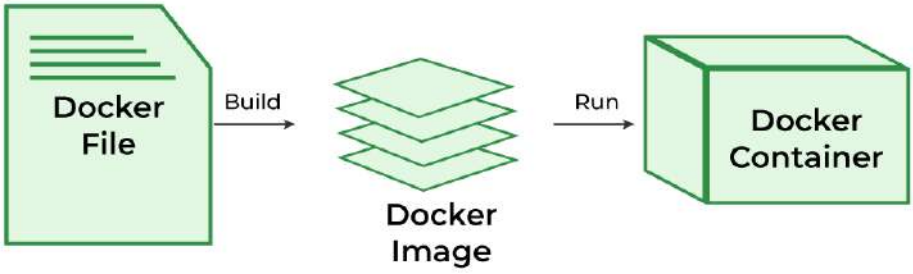
EXPERIMENT NO. 09

Aim: To build an image for a sample web application from a CLI and docker file using various docker file instructions

Theory: The Dockerfile uses DSL (Domain Specific Language) and contains instructions for generating a Docker image. Dockerfile will define the processes to quickly produce an image. While creating your application, you should create a Dockerfile in order since the Docker daemon runs all of the instructions from top to bottom.

An artifact with several layers and a lightweight, compact stand-alone executable package that contains all of the components required to run a piece of software, including the code, a runtime, libraries, environment variables, and configuration files is called a Docker image.

A container is a runtime instance of an image. Containers make development and deployment more efficient since they contain all the dependencies and parameters needed for the application it runs completely isolated from the host environment.



Dockerfile commands/Instructions

1. FROM

- Represents the base image(OS), which is the command that is executed first before any other commands.

Syntax

FROM <ImageName>

2. COPY

- *The copy command is used to copy the file/folders to the image while building the image.*

Syntax:

COPY <Source> <Destination>

3] RUN

- Scripts and commands are run with the RUN instruction. The execution of RUN commands or instructions will take place while you create an image on top of the prior layers (Image).

Syntax

RUN < Command + ARGS>

4] CMD

- *The main purpose of the CMD command is to start the process inside the container and it can be overridden.*

Syntax

CMD [command + args]

Stages of Creating Docker Image from Dockerfile
The following are the stages of creating docker image form Dockerfile:
1. Create a file named Dockerfile.
2. Add instructions in Dockerfile.
3. Build Dockerfile to create an image.
4. Run the image to create a container.

IMPLEMENTATION:

PART I: Containerize an application using docker CLI Commands:

Let's create an nginx webserver, it is a web server platform which helps to host your web applications.

STEP1: Download nginx official image and then containerized your web application in it.

#docker images

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
mysql        latest    245a6c909dc0   11 days ago    921MB
nginx        latest    2cd1d97f893f   2 weeks ago    192MB
ubuntu       latest    65ae7a6f3544   2 weeks ago    78.1MB
```

docker rmi mysql nginx ubuntu

#docker images

```
devasc@labvm:~/Desktop/sujata-docker$ sudo su
root@labvm:/home/devasc/Desktop/sujata-docker# docker images
REPOSITORY    TAG         IMAGE ID    CREATED    SIZE
```

#docker ps -a

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID    IMAGE      COMMAND    CREATED    STATUS     PORTS      NAMES
```

#docker pull nginx

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker pull nginx:latest
latest: Pulling from library/nginx
59e22667830b: Pull complete
140da4f89dcb: Pull complete
96e47e70491e: Pull complete
2ef442a3816e: Pull complete
4b1e45a9989f: Pull complete
1d9f51194194: Pull complete
f30ffbee4c54: Pull complete
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d568174487289 24a87ab32f12a72fb
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

#docker images

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker images
REPOSITORY    TAG         IMAGE ID        CREATED         SIZE
nginx         latest      2cd1d97f893f    2 weeks ago     192MB
```

STEP2: Run the container from nginx image

 # docker run --name webserver1 5ef

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker run --name webserver1 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform conf
iguration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/defaul
t.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/defa
ult.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/08/02 18:53:58 [notice] 1#1: using the "epoll" event method
2025/08/02 18:53:58 [notice] 1#1: nginx/1.29.0
2025/08/02 18:53:58 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/08/02 18:53:58 [notice] 1#1: OS: Linux 5.4.0-37-generic
2025/08/02 18:53:58 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/08/02 18:53:58 [notice] 1#1: start worker processes
2025/08/02 18:53:58 [notice] 1#1: start worker process 28
2025/08/02 18:53:58 [notice] 1#1: start worker process 29
```

In another terminal

#docker ps -a

```
devasc@labvm:~/Desktop/sujata-docker$ sudo su
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
   PORTS     NAMES
1007983ca28c   nginx     "/docker-entrypoint.…"   54 seconds ago   Up 54 seconds
   80/tcp    webserver1
```

In previous terminal: ctrl+C ie; exit from container

In another terminal

#docker ps -a

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS
                 PORTS     NAMES
1007983ca28c   nginx     "/docker-entrypoint.…"   About a minute ago   Exited (0
) 7 seconds ago            webserver1
```

Remove the container:

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker container rm 100
100
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

In terminal 1:

docker run -it -p 3031:80 --name server1 nginx:latest bash

```
root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it -p 3031:80 --name nse
rver1 nginx:latest bash
```

In Another Terminal:

#docker ps -a

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
        PORTS                                  NAMES
0b847b3b176c   nginx:latest   "/docker-entrypoint.…"   30 seconds ago   Up 30 se
conds   0.0.0.0:3031->80/tcp, :::3031->80/tcp   server1
```

Lets create a static website inside container. I need to go to the location where my index.html file is:

cd /usr/share/nginx/html/

```
root@0b847b3b176c:/# cd /usr/share/nginx/html/
```

root@0b847b3b176c:/usr/share/nginx/html#ls

```
root@0b847b3b176c:/usr/share/nginx/html# ls
50x.html  index.html
```

Rename the default index.html to index.html_backup

root@0b847b3b176c:/usr/share/nginx/html#

```
root@0b847b3b176c:/usr/share/nginx/html# mv index.html index.html_backup
```

#nano index.html

```
root@0b847b3b176c:/usr/share/nginx/html# nano index.html
```

Nano not found: Because the container that I am running inside the shell says that nano application is not available inside the container. So first install nano: apt install nano

```
root@0b847b3b176c:/usr/share/nginx/html# nano index.html
bash: nano: command not found
```

```
root@0b847b3b176c:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package nano is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'nano' has no installation candidate
```

root@0b847b3b176c:/usr/share/nginx/html# apt update

```
root@0b847b3b176c:/usr/share/nginx/html# apt update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8793 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [6916 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [272
kB]
Fetched 9327 kB in 2s (4129 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

#apt install nano

```
root@0b847b3b176c:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncursesw6
Suggested packages:
  gpm hunspell
```

```
root@0b847b3b176c:/usr/share/nginx/html# nano index.html
```

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
button {
       background-color: #4CAF50;
       width: 100%;
        color: orange;
        padding: 15px;
        margin: 10px 0px;
        border: none;
        cursor: pointer;
         }
 form {
        border: 3px solid #f1f1f1;
    }
 input[type=text], input[type=password] {
```

```
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
 button:hover {
        opacity: 0.7;
    }
  .cancelbtn {
        width: auto;
        padding: 10px 18px;
        margin: 10px 5px;
    }


 .container {
        padding: 25px;
        background-color: lightblue;
    }
</style>
</head>
<body>
    <center> <h1> <b>Student Login Form Designed by Sujata Oak</b> </h1> </center>
    <form>
        <div class="container">
            <label>Username : </label>
            <input type="text" placeholder="Enter Username" name="username" required>
            <label>Password : </label>
            <input type="password" placeholder="Enter Password" name="password" required>
            <button type="submit">Login</button>
            <input type="checkbox" checked="checked"> Remember me
            <button type="button" class="cancelbtn"> Cancel</button>
            Forgot <a href="#"> password? </a>
        </div>
    </form>
</body>
</html>
```

To check nginx service status:

```
root@0b847b3b176c:/usr/share/nginx/html# service nginx status
nginx is not running ... failed!
```

#service nginx start

```
root@0b847b3b176c:/usr/share/nginx/html# service nginx start
2025/08/02 19:21:08 [notice] 177#177: using the "epoll" event method
2025/08/02 19:21:08 [notice] 177#177: nginx/1.29.0
2025/08/02 19:21:08 [notice] 177#177: built by gcc 12.2.0 (Debian 12
2025/08/02 19:21:08 [notice] 177#177: OS: Linux 5.4.0-37-generic
2025/08/02 19:21:08 [notice] 177#177: getrlimit(RLIMIT_NOFILE): 1048
2025/08/02 19:21:08 [notice] 178#178: start worker processes
2025/08/02 19:21:08 [notice] 178#178: start worker process 179
2025/08/02 19:21:08 [notice] 178#178: start worker process 180
```

STEP 3:

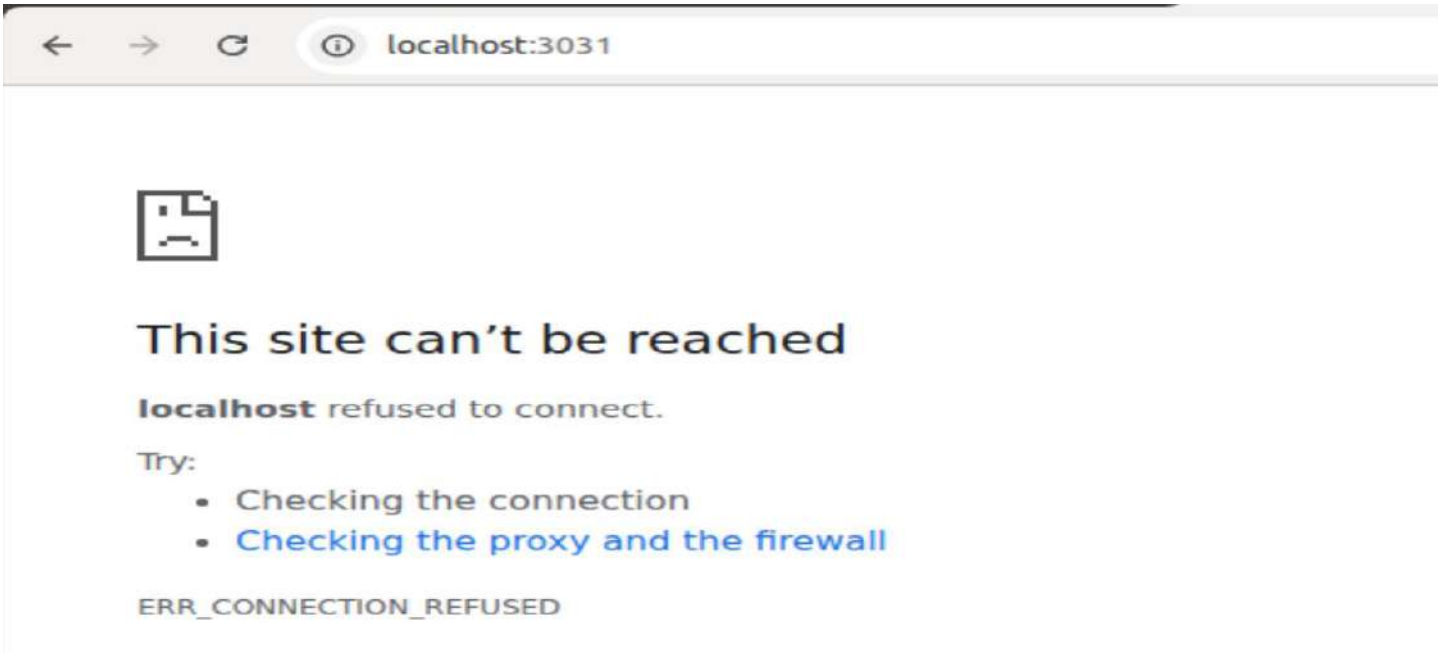
GOTO BROWSER: localhost:3031

To See the logs on first terminal:

```
root@0b847b3b176c:/usr/share/nginx/html# 172.17.0.1 - - [02/Aug/2025:19:21:44 +0000] "G
ET / HTTP/1.1" 200 1758 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/138.0.0.0 Safari/537.36" "-"
2025/08/02 19:21:44 [error] 180#180: *1 open() "/usr/share/nginx/html/favicon.ico" fail
ed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET
 /favicon.ico HTTP/1.1", host: "localhost:3031", referrer: "http://localhost:3031/"
172.17.0.1 - - [02/Aug/2025:19:21:44 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http:/
/localhost:3031/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/138.0.0.0 Safari/537.36" "-"
```

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker stop 0b8
0b8
```

Goto browser→ Refresh page . Your Container is stopped now

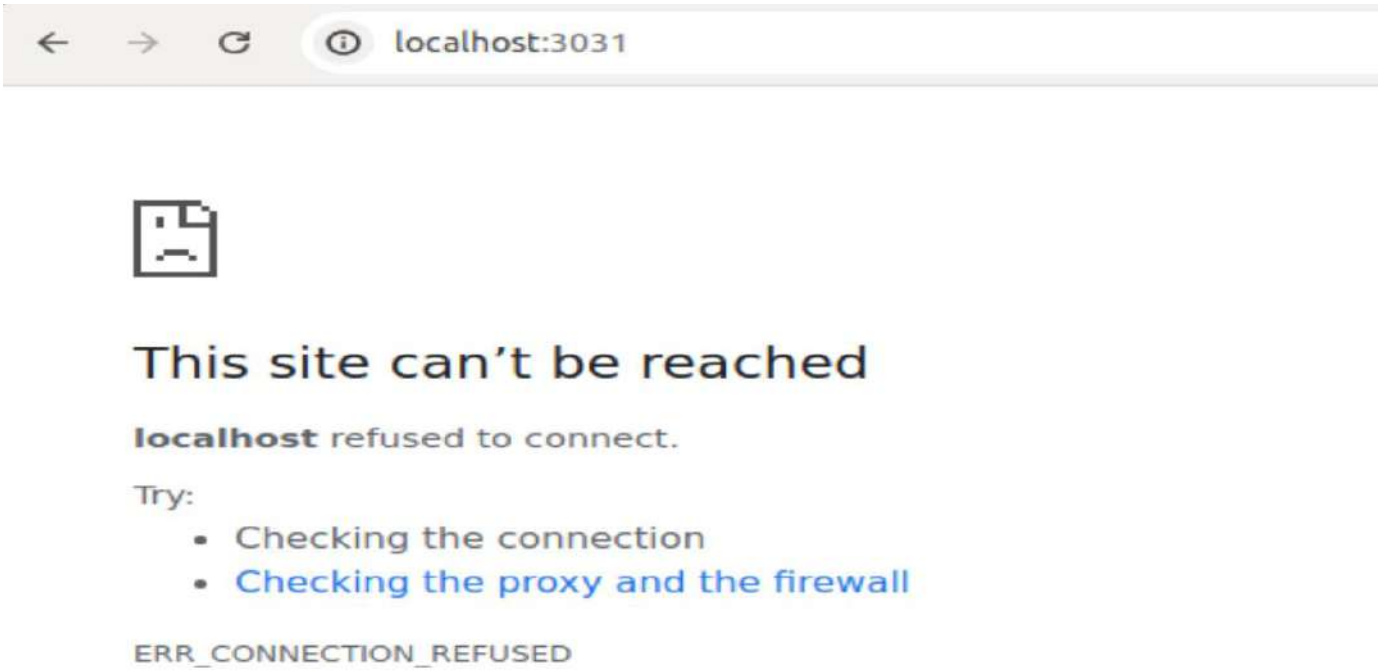



```
root@labvm:/home/devasc/Desktop/sujata-docker# docker start 0b8
0b8
```

Goto browser→ Refresh page . Your Container is not started

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker restart 0b8
0b8
```

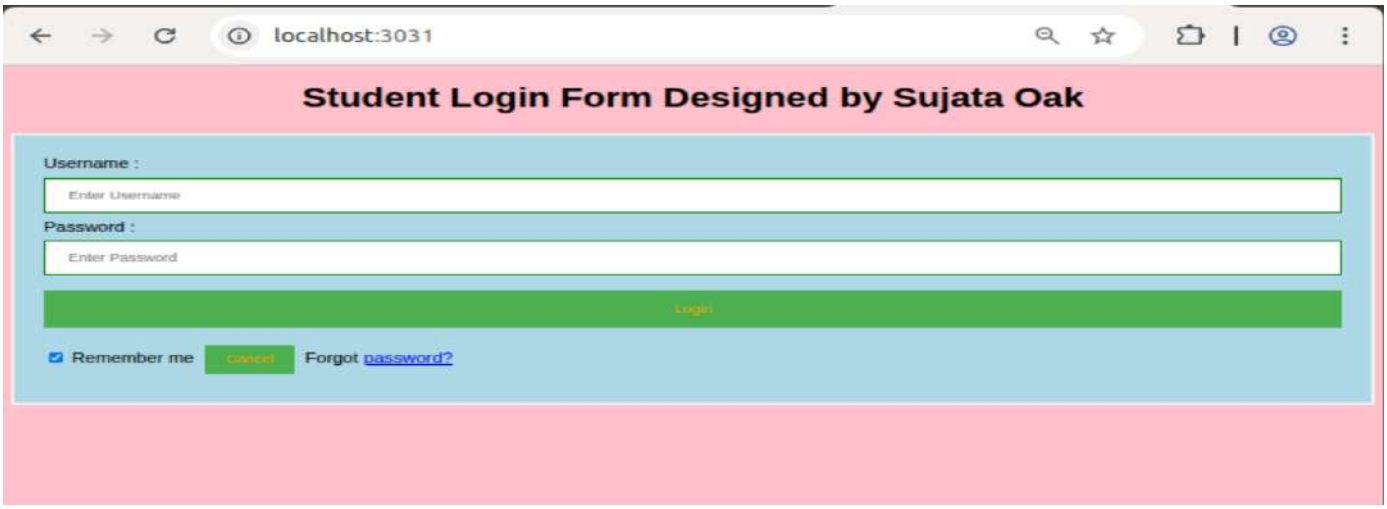
Goto browser→ Refresh page . Your Container is not restarted



docker exec 0b8 service nginx start

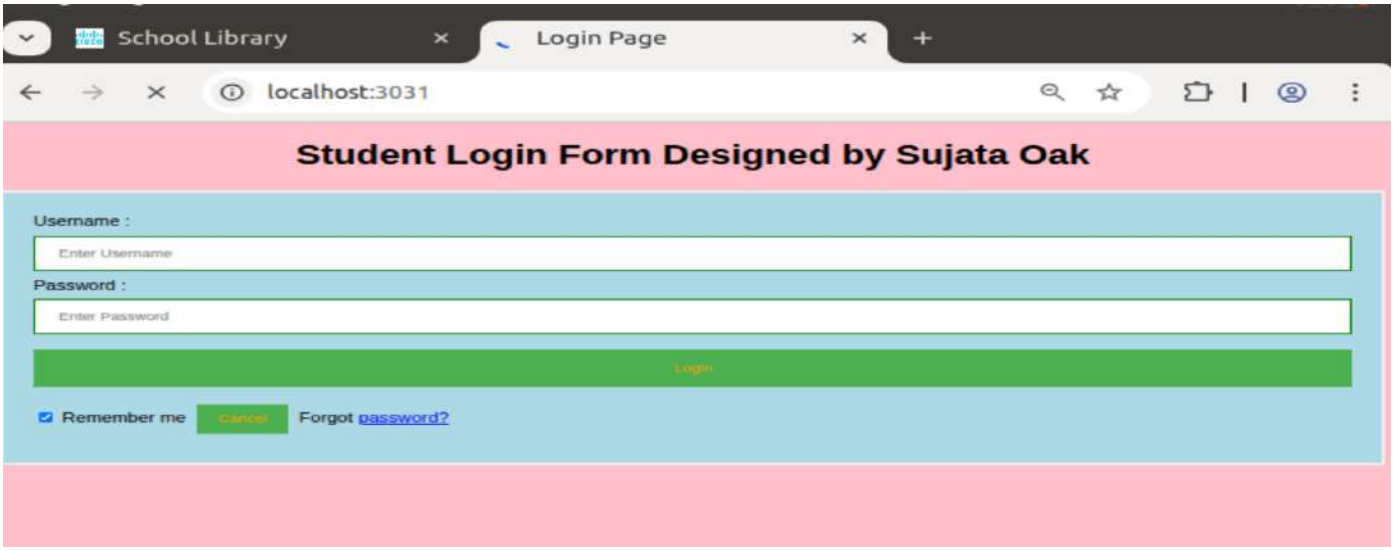
```
root@labvm:/home/devasc/Desktop/sujata-docker# docker exec 0b8 service nginx sta
rt
2025/08/02 19:29:29 [notice] 18#18: using the "epoll" event method
2025/08/02 19:29:29 [notice] 18#18: nginx/1.29.0
2025/08/02 19:29:29 [notice] 18#18: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u
1)
2025/08/02 19:29:29 [notice] 18#18: OS: Linux 5.4.0-37-generic
2025/08/02 19:29:29 [notice] 18#18: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/08/02 19:29:29 [notice] 19#19: start worker processes
2025/08/02 19:29:29 [notice] 19#19: start worker process 20
2025/08/02 19:29:29 [notice] 19#19: start worker process 21
```

Goto Browser and refresh it:



#docker pause 0b8

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker pause 0b8
0b8
```



#docker unpause 0b8

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker unpause 0b8
0b8
```

Firstly stop the container:

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker stop 0b8
0b8
```

Then, Remove the Container

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker container rm 0b8
0b8
```

To Verify container is removed or not:

```
root@labvm:/home/devasc/Desktop/sujata-docker# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

PART II: DOCKERFILE

Creating a Docker Image for your Application:

This is the recommended workflow for creating your own Docker image for your application:

1. Write a Dockerfile for your application.
2. Build the image with docker build command.
3. Host your Docker image on a registry.
4. Pull and run the image on the target machine.

Docker builds images automatically by reading the instructions from a Dockerfile. It is a text file that contains all commands needed to build a given image.

STEP 1: # git clone https://github.com/sujataoak799/nginx-dockerfile.git

```
root@labvm:/home/devasc/Desktop/sujata-docker# git clone https://github.com/sujataoak79
9/nginx-dockerfile.git
Cloning into 'nginx-dockerfile'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (8/8), 2.63 KiB | 674.00 KiB/s, done.
```

```
root@labvm:/home/devasc/Desktop/sujata-docker# ls
nginx-dockerfile
```

```
root@labvm:/home/devasc/Desktop/sujata-docker# cd nginx-dockerfile/
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# ls
Dockerfile  index.html  README.md  style.css
```

Step 2:

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# nano Dockerfile
```

```
  GNU nano 4.8
FROM ubuntu
LABEL author="Sujata Oak"
RUN apt-get update
RUN apt-get install nginx -y
COPY . /var/www/html/
EXPOSE 80
CMD ["nginx","-g","daemon off;"]
```

#docker build -t sujatadocker2025/websitetest25 .

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker build -t sujat
ocker2025/websitetest25 .
[+] Building 24.9s (10/10) FINISHED                                    docker:defaul
 => [internal] load build definition from Dockerfile                              0.0
 => => transferring dockerfile: 190B                                              0.0
 => [internal] load metadata for docker.io/library/ubuntu:latest                  4.2
 => [auth] library/ubuntu:pull token for registry-1.docker.io                     0.0
 => [internal] load .dockerignore                                                 0.1
 => => transferring context: 2B                                                   0.0
 => [1/4] FROM docker.io/library/ubuntu:latest@sha256:a08e551cb33850e4740772b382  6.1
 => => resolve docker.io/library/ubuntu:latest@sha256:a08e551cb33850e4740772b382   0.1
 => => sha256:a08e551cb33850e4740772b38217fc1796a66da2506d312abe 6.69kB / 6.69kB  0.0
 => => sha256:4f1db91d9560cf107b5832c0761364ec64f46777aa4ec637cca300 424B / 424B  0.0
 => => sha256:65ae7a6f3544bd2d2b6d19b13bfc64752d776bc92c510f8741 2.30kB / 2.30kB  0.0
 => => sha256:32f112e3802cadcab3543160f4d2aa607b3cc1c62140d57b 29.72MB / 29.72MB  3.0
 => => extracting sha256:32f112e3802cadcab3543160f4d2aa607b3cc1c62140d57b4f54413  2.1
 => [internal] load build context                                                 0.1
 => => transferring context: 28.65kB                                              0.0
 => [2/4] RUN apt-get update                                                      8.2
 => [3/4] RUN apt-get install nginx -y                                            5.0
```

 # docker images

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker images
REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
sujatadocker2025/websitetest25      latest    b4bdc0855a68   50 seconds ago   136MB
nginx                               latest    2cd1d97f893f   2 weeks ago      192MB
```

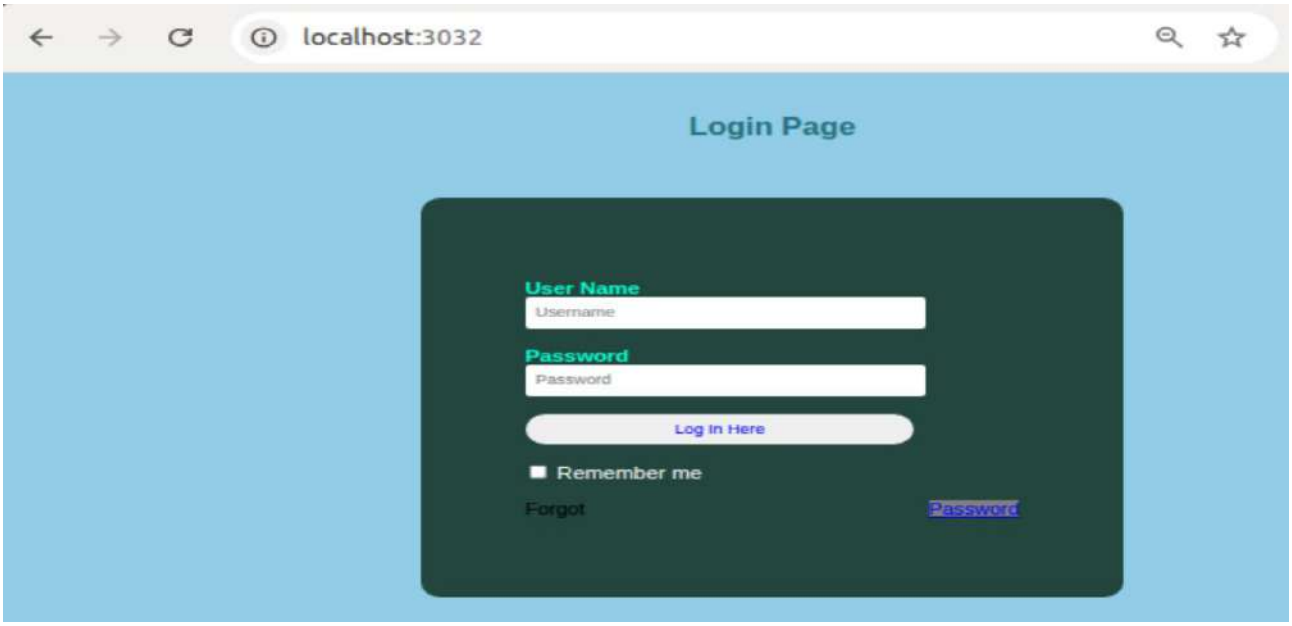
Step 3: Run the container now:

docker run -d -p 3032:80 --name sujata_webcontainer b4b

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker run -d -p 3032:8
0 --name sujata_webcontainer b4b
86b2ebd73cc97f6d98b55fc6eb0f0e4cfae086937903d517fb83137b9f51bcb2
```

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORT
S                                NAMES
86b2ebd73cc9   b4b       "nginx -g 'daemon of…"   57 seconds ago   Up 56 seconds   0.0.
0.0:3032->80/tcp, :::3032->80/tcp   sujata_webcontainer
```

Step 4: Goto Browser: localhost:3032



STEP 5: How to push this image to your dockerhub :

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker images
REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
sujatadocker2025/websitetest25      latest    b4bdc0855a68   8 minutes ago    136MB
nginx                               latest    2cd1d97f893f   2 weeks ago      192MB
```

 # docker push sujatadocker2025/websitetest25

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker push sujatadocke
r2025/websitetest25
Using default tag: latest
The push refers to repository [docker.io/sujatadocker2025/websitetest25]
3ec9475e5d94: Preparing
5b06bfd1f90d: Preparing
97d8e323fdac: Preparing
107cbdaeec04: Preparing
denied: requested access to the resource is denied
```

 # docker tag b4b 18061977/apsitsujatacontainer25:v1

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker tag b4b 18061977
/apsitsujatacontainer25:v1
```

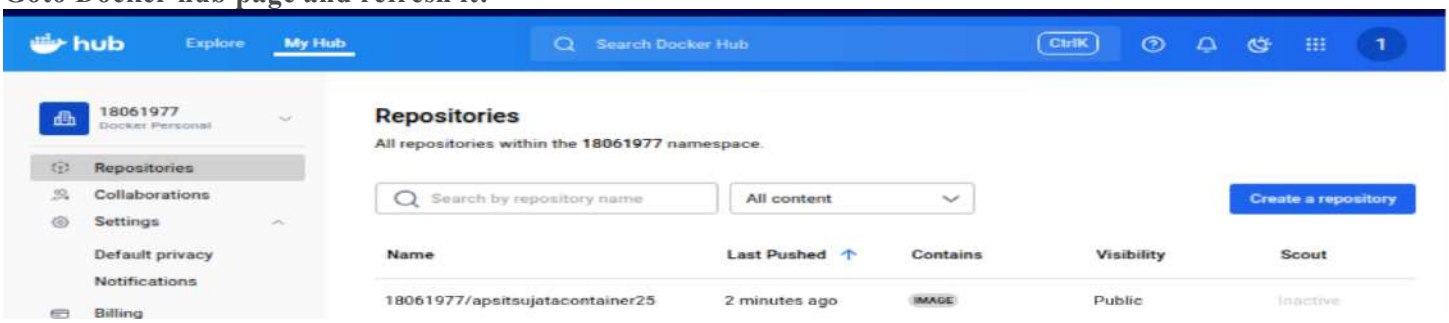
```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker images
REPOSITORY                                TAG       IMAGE ID       CREATED          SIZE
18061977/apsitsujatacontainer25           v1        b4bdc0855a68   14 minutes ago   136MB
sujatadocker2025/websitetest25            latest    b4bdc0855a68   14 minutes ago   136MB
nginx                                     latest    2cd1d97f893f   2 weeks ago      192MB
```

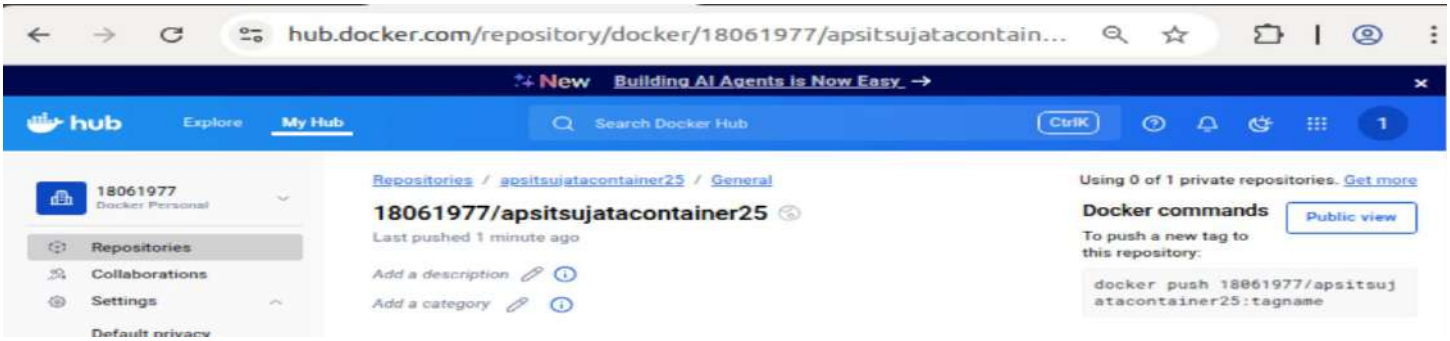
```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
```

 # docker push 18061977/apsitsujatacontainer25:v1

```
root@labvm:/home/devasc/Desktop/sujata-docker/nginx-dockerfile# docker push 18061977/ap
sitsujatacontainer25:v1
The push refers to repository [docker.io/18061977/apsitsujatacontainer25]
3ec9475e5d94: Pushed
5b06bfd1f90d: Pushed
97d8e323fdac: Pushed
107cbdaeec04: Pushed
v1: digest: sha256:763b31bf15297e1b8f1cf18f04f72a7903346228b5d16dea9d6e1c0aa04c11c5 siz
e: 1161
```

Goto Docker hub page and refresh it:





Conclusion: In the experiment, we used various docker commands to pull images that were already built, also we created our own images by using docker file instructions for a sample web application and atlast we have pushed the image to docker hub account for others to use the repository.