HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

DEVOPS

PRACTICALS

Name: Burhanuddin Khairulla

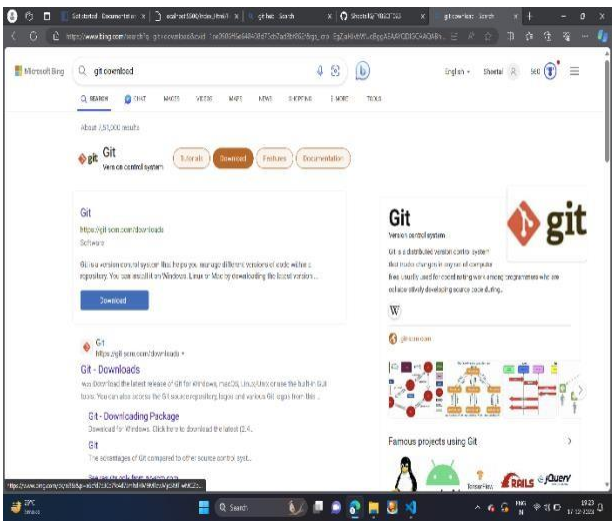
Roll No: 036

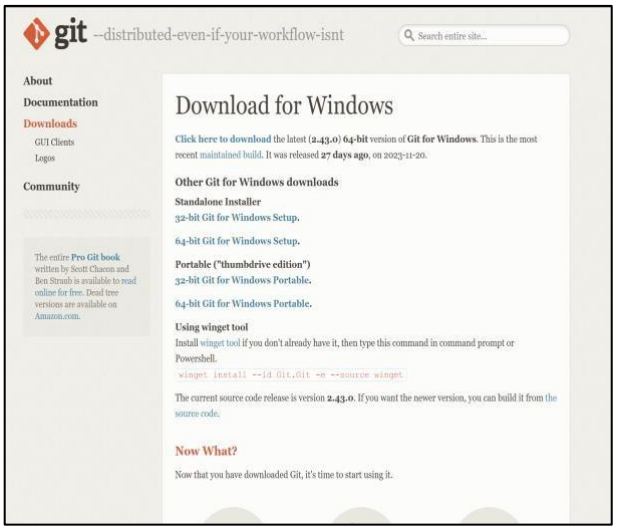
HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

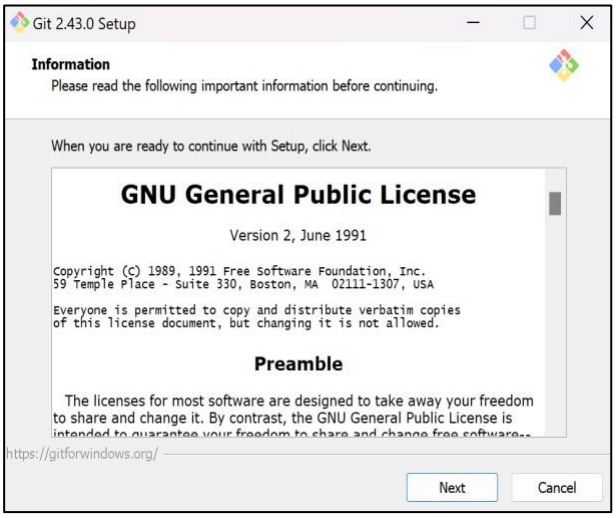
Practical No 1: Installation of Git

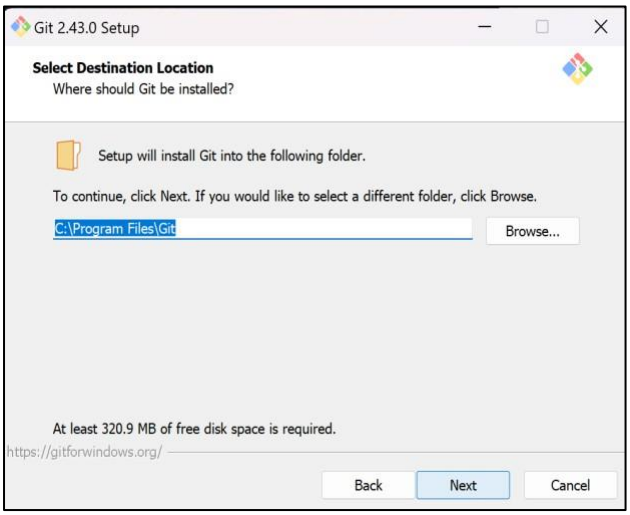
Step 1: Download the latest version of GIT and choose either 64/32 bit.



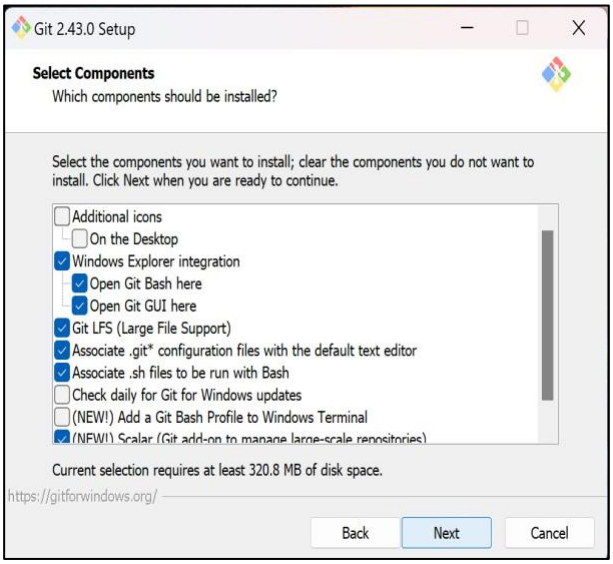


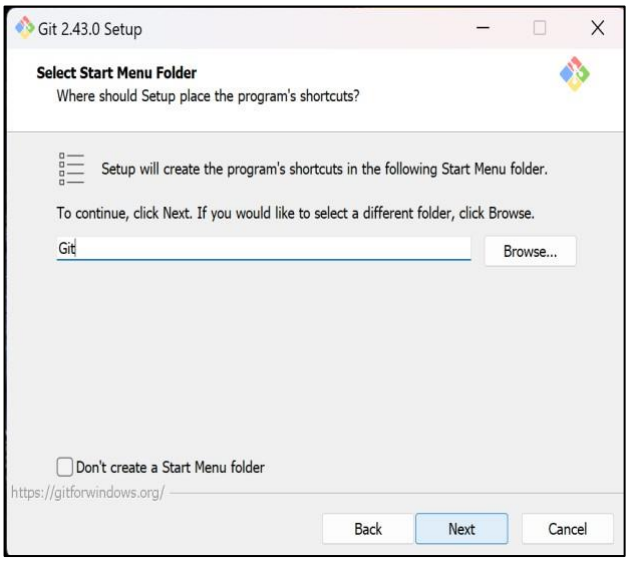
Step 2: After the file is downloaded, install it in the system, once install double click or extract and launch the installer. Go to GNU (General Public License). Click Next. Choose the installation location of your choice and click Next.



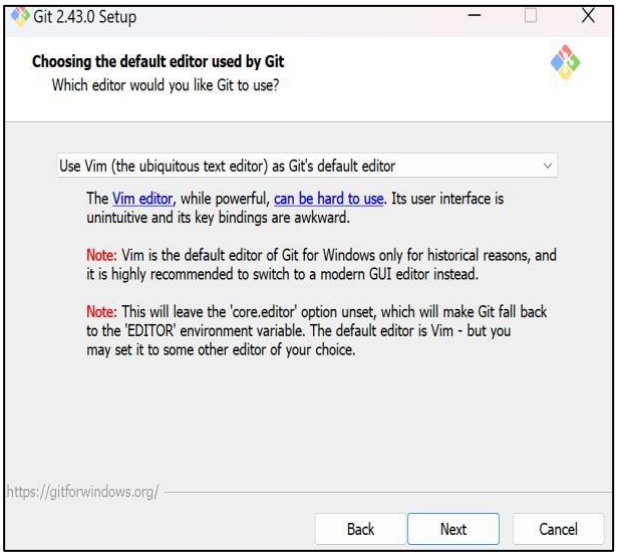


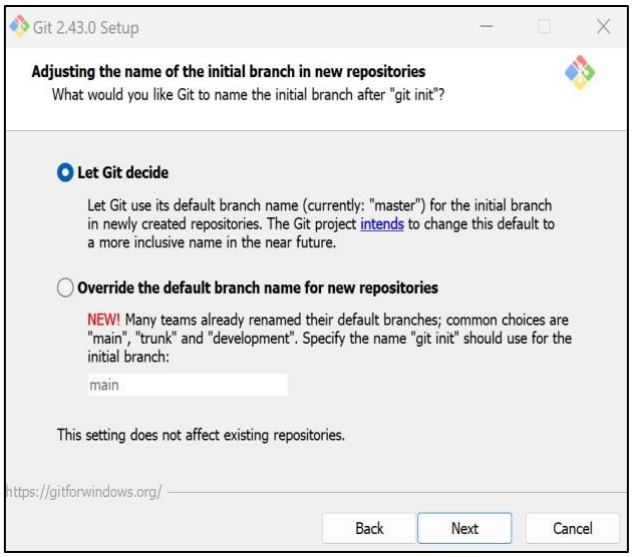
Step 3: Leave the defaulted selected component and click next. Choose the start menu folder location and click on next.



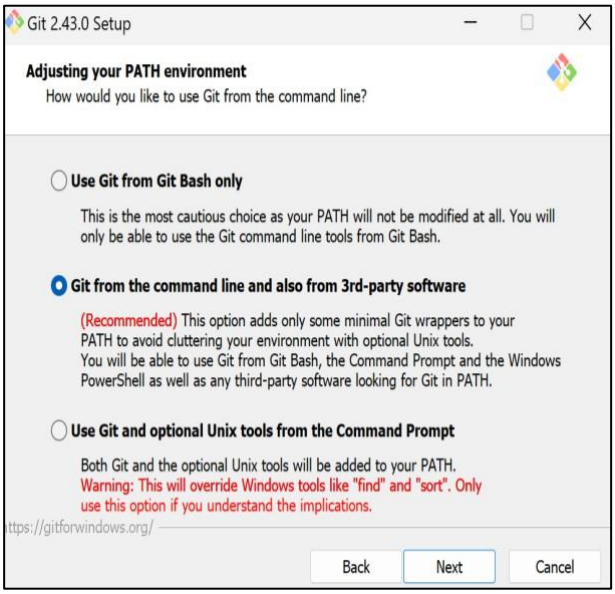


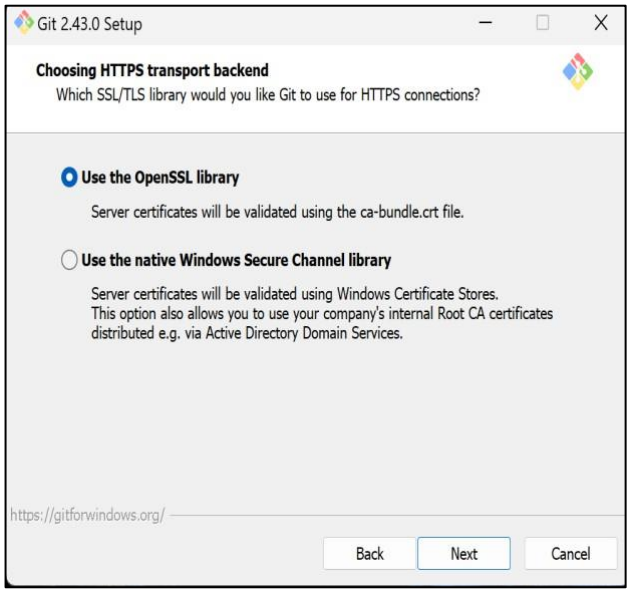
Step 4: Choose VIM/Text editor you would like to use with Git and click next.



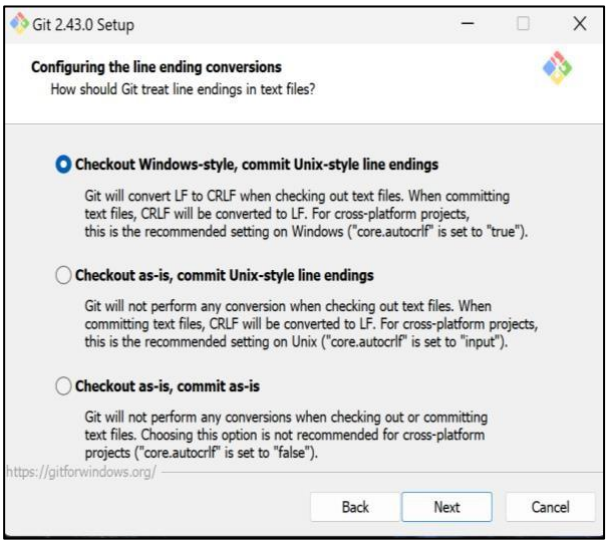


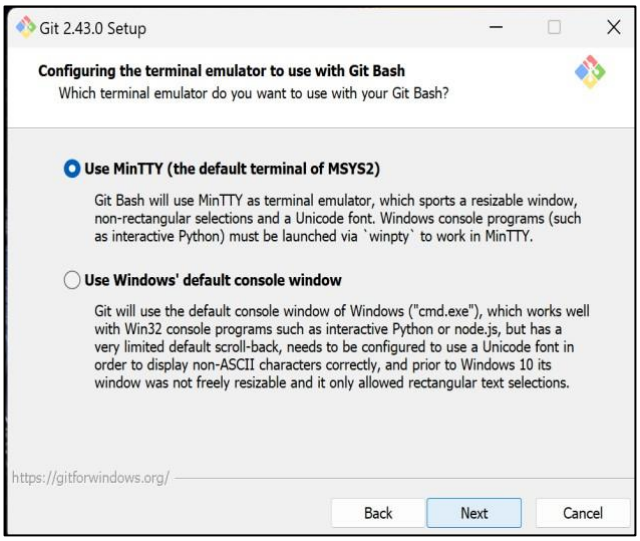
Step 5: Select Git from command line and also from third party software and click on next. Select the server certificate of your choice the default one is used to open SSL library and click next.



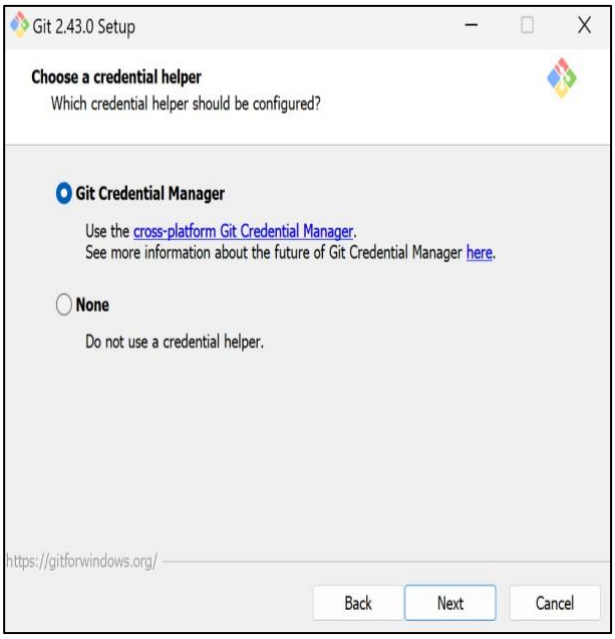


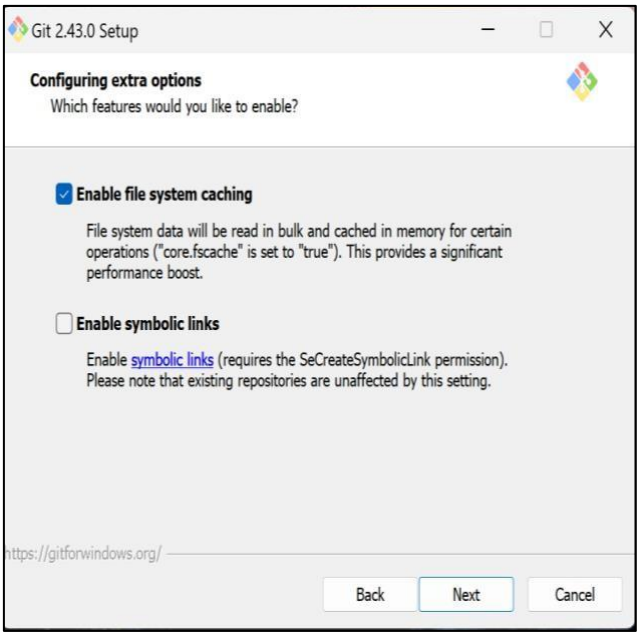
Step 6: Choose the checkout window style, commit unix-style line endings and click next. Select the default MINTTY and click next.



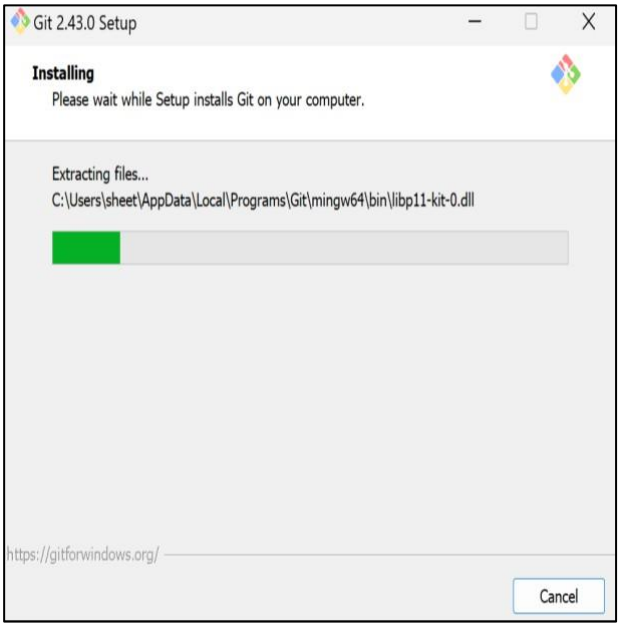


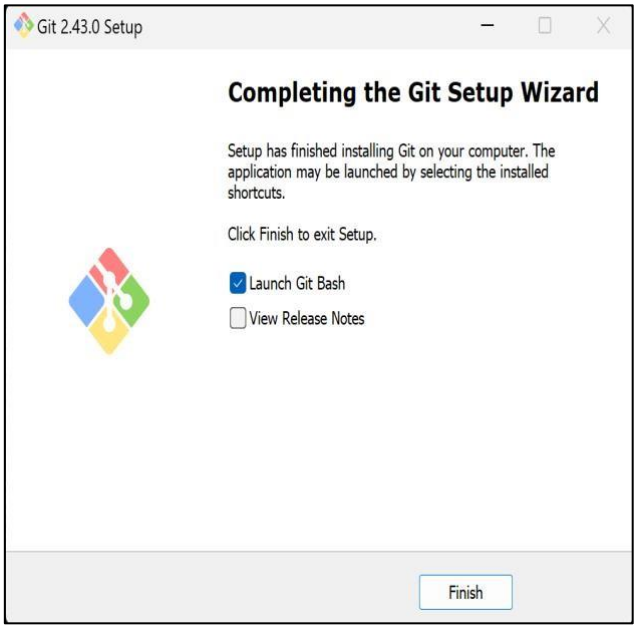
Step 7: Select the Git Credential manager and click next. In the configuring extra options select Enable file system caching and click next.





Step 8: After this Git will install.



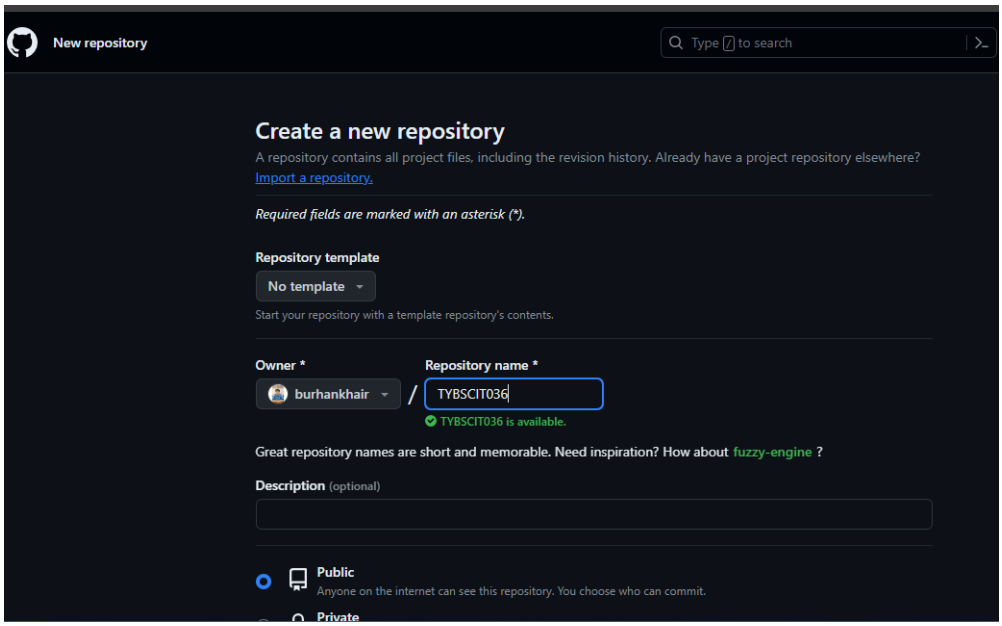


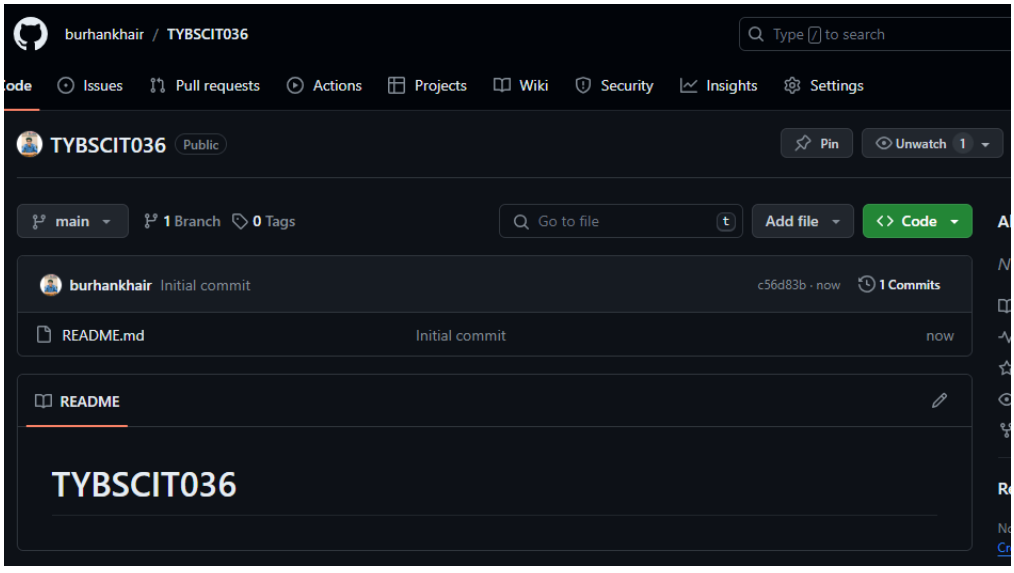
HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

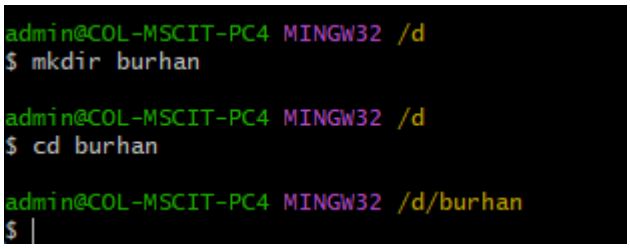
Practical 2A: Git Commands working with local repository

Step 1: Login in Git hub and create a repository, give repository name and then click on create repository



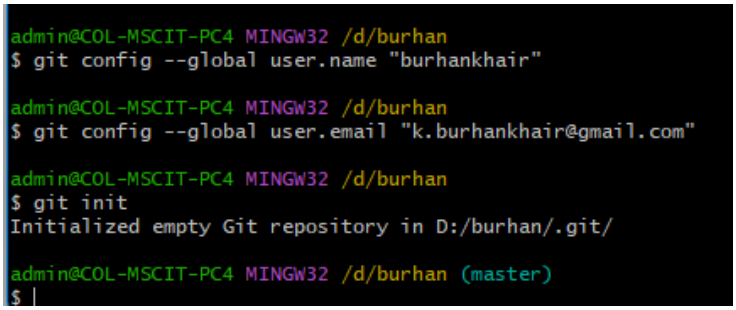


Step 2: Open Git Bash -> Create a directory

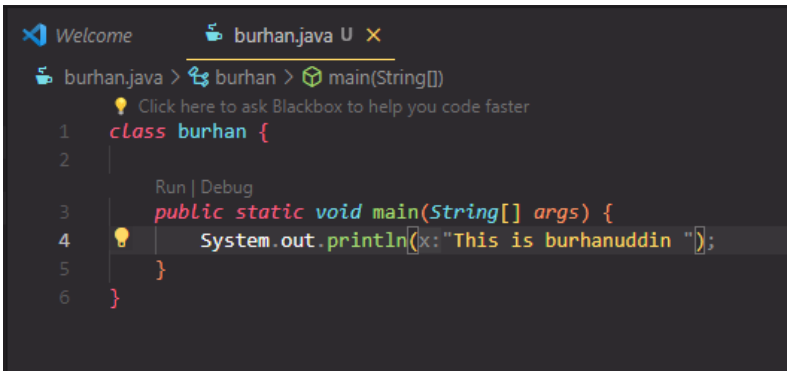


Step 3: Git Commands.

a- git config –global user.name “burhankhair”
b- git init

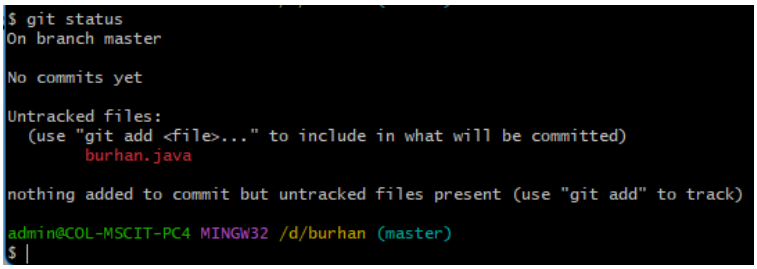


Step 4: Create a java file and save it in local directory

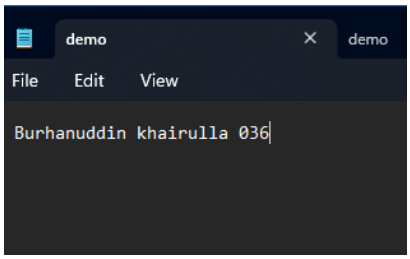


Step 5: Git commands

Git status



Step 6: Create another file in the directory




```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        burhan.java
        demo.txt

nothing added to commit but untracked files present (use "git add" to track)

admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$
```

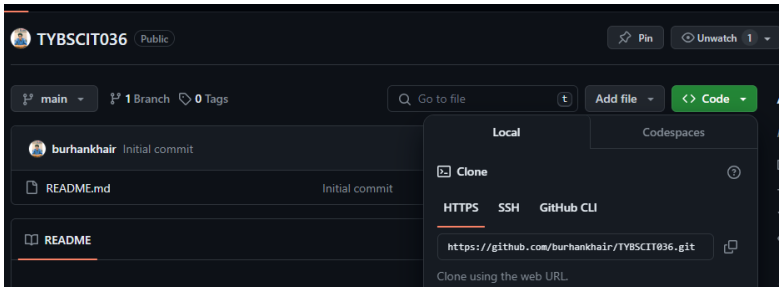
Git add

```
$ git add burhan.java

admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   burhan.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        demo.txt
```

Git commit

```
$ git commit -m "Burhanuddin"
[master (root-commit) 9ebeccf] Burhanuddin
 1 file changed, 6 insertions(+)
 create mode 100644 burhan.java

admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        demo.txt

nothing added to commit but untracked files present (use "git add" to track)

admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$
```

Git clone url



HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

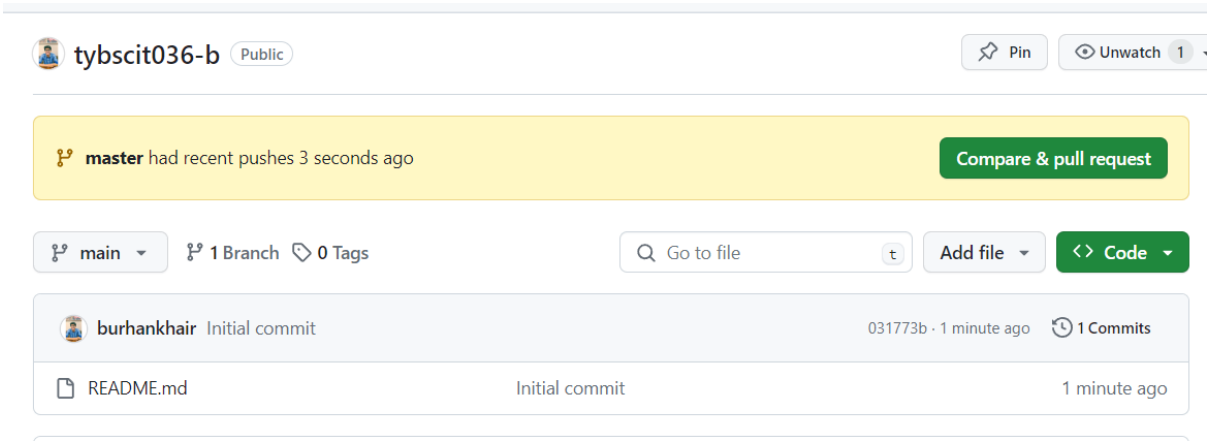
```
admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$ git clone https://github.com/burhankhair/TYBSCIT036.git
Cloning into 'TYBSCIT036'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$
```

Git remote origin

```
admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$ git remote add origin https://github.com/burhankhair/TYBSCIT036.git
```

Git push

```
admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$ git push -u origin master
```



HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

Practical 2B : Perfom VCS with github – (Push, Pull,Fetch, Merge,Branch)

Step 1 – Creating a branch

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git checkout -b burhan-branch
Switched to a new branch 'burhan-branch'
```

Step 2: Adding a new file (to make changes)

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>touch burhan.java
Touching burhan.java
```

Step 3 – Adding it

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git add burhan.java
```

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git commit -m "ADDED burhan.java"
[burhan-branch d5a00df] ADDED burhan.java
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 burhan.java
```

Step 4 : Push your changes to the remote repository on GitHub:

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git push origin burhan-branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'burhan-branch' on GitHub by visiting:
remote:      https://github.com/burhankhair/tybscit036-b/pull/new/burhan-branch
remote:
To https://github.com/burhankhair/tybscit036-b.git
 * [new branch]      burhan-branch -> burhan-branch
```

Step 5 – Fetch Changes

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 195 bytes | 17.00 KiB/s, done.
From https://github.com/burhankhair/tybscit036-b
 * [new branch]      master     -> origin/master
```

Step 6 – Merge Changes

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git checkout burhan-branch
Already on 'burhan-branch'

C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git merge origin
Already up to date.
```

Step 7 – Push the merged changes to repositary

```
C:\BSCIT\TY\sem6\devOps\practice\tybscit036-b>git push origin burhan-branch
Everything up-to-date
```

HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

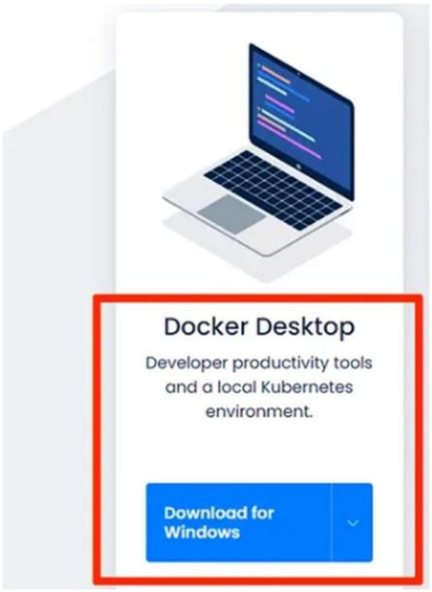
Practical 3 – Docker Installation

Install Docker on Windows Step 1:

Downloading Docker

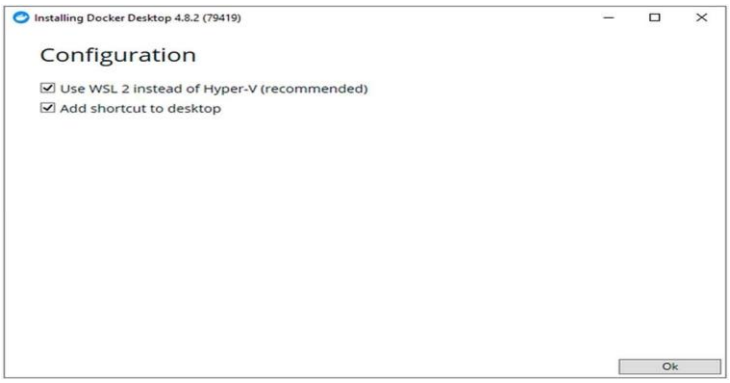
The first place to start is the official Docker website from where we can download DockerDesktop.

Please note that Docker Desktop is intended only for Windows 10/11 and not for Windows Server.



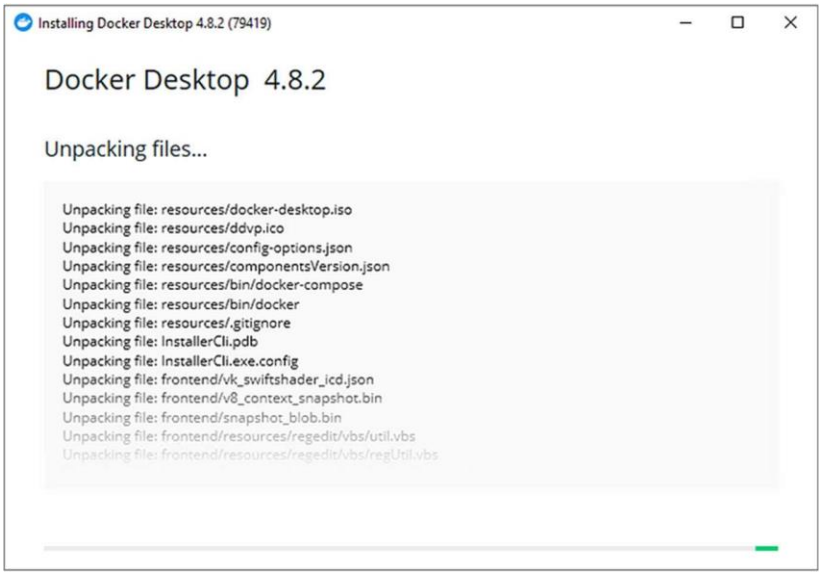
Step 2: Configuration

To run Linux on Windows, Docker requires a virtualization engine. Docker recommendsusing WSL 2.



Step 3: Running the installation

Click Ok, and wait a bit…



Step 4: Restart

For Docker to be able to properly register with Windows, a restart is required at thispoint.

Step 5: License agreement

After the restart, Docker will start automatically and you should see the window below:

HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

Practical 4 – Docker Commands

docker search mysql

```
C:\Users\kburh>docker search mysql
NAME                            DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
mysql                           MySQL is a widely used, open-source relation…   14859     [OK]
mariadb                         MariaDB Server is a high performing open sou…   5675      [OK]
percona                         Percona Server is a fork of the MySQL relati…   624       [OK]
phpmyadmin                      phpMyAdmin - A web interface for MySQL and M…   949       [OK]
bitnami/mysql                   Bitnami MySQL Docker Image                      107                  [OK]
bitnami/mysqld-exporter                                                         6
cimg/mysql                                                                      3
ubuntu/mysql                    MySQL open source fast, stable, multi-thread…   59
rapidfort/mysql                 RapidFort optimized, hardened image for MySQL   25
rapidfort/mysql8-ib             RapidFort optimized, hardened image for MySQ…   9
google/mysql                    MySQL server for Google Compute Engine          25                   [OK]
rapidfort/mysql-official        RapidFort optimized, hardened image for MySQ…   9
elestio/mysql                   Mysql, verified and packaged by Elestio         0
hashicorp/mysql-portworx-demo                                                   0
bitnamicharts/mysql                                                             0
databack/mysql-backup           Back up mysql databases to... anywhere!         109
linuxserver/mysql               A Mysql container, brought to you by LinuxSe…   41
mirantis/mysql                                                                  0
docksal/mysql                   MySQL service images for Docksal - https://d…   0
linuxserver/mysql-workbench                                                     55
vitess/mysqlctld                vitess/mysqlctld                                1                    [OK]
eclipse/mysql                   Mysql 5.7, curl, rsync                          1                    [OK]
drupalci/mysql-5.5              https://www.drupal.org/project/drupalci         3                    [OK]
drupalci/mysql-5.7              https://www.drupal.org/project/drupalci         0
datajoint/mysql                 MySQL image pre-configured to work smoothly …   2                    [OK]
```

Docker pull openjdk

```
C:\Users\kburh>docker pull openjdk
Using default tag: latest
latest: Pulling from library/openjdk
197c1adcd755: Already exists
57b698b7af4b: Already exists
95a27dbe0150: Already exists
Digest: sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f
Status: Downloaded newer image for openjdk:latest
docker.io/library/openjdk:latest
```

Docker ps

```
C:\Users\kburh>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                    NAMES
5ceaa85f1bbf   nodejsburhan   "docker-entrypoint.s…"   24 seconds ago   Up 21 seconds   0.0.0.0:3000->3000/tcp   cool_robinson
```

Docker images

```
C:\Users\kburh>docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
nodejsburhan            latest    af5087fdb7b2   38 seconds ago   912MB
burhan-nodejs           latest    a3f69b7be4cd   4 minutes ago    912MB
burhan-python-app       latest    42e858dc1609   28 hours ago     997MB
burhanphp               latest    61b2edd008d7   28 hours ago     474MB
burhanajax              latest    a318a3d82b54   29 hours ago     42.6MB
<none>                  <none>    21a681208b95   29 hours ago     42.6MB
<none>                  <none>    e484797c8c05   29 hours ago     42.6MB
burhan-node-js-app2     latest    712bd09ab9d6   29 hours ago     912MB
<none>                  <none>    9c4b948e786c   29 hours ago     912MB
burhan-node-js-app      latest    c154ba7c699e   30 hours ago     912MB
node-js-burhan          latest    c440b42f9682   30 hours ago     912MB
burhannodejsapp         latest    319eecfb9ad6   30 hours ago     912MB
burhan-node-js          latest    2c83a6d95faf   30 hours ago     912MB
burhanappnodejs         latest    2c83a6d95faf   30 hours ago     912MB
my-nodejs-app           latest    2c83a6d95faf   30 hours ago     912MB
burhanapp-java          latest    ae9e6c4ec4cd   38 hours ago     470MB
burhanapp5c             latest    ae9e6c4ec4cd   38 hours ago     470MB
burhanapp               latest    ae9e6c4ec4cd   38 hours ago     470MB
<none>                  <none>    98adddf619fd   38 hours ago     470MB
<none>                  <none>    094985e7c006   38 hours ago     470MB
<none>                  <none>    9452224764eb   38 hours ago     470MB
<none>                  <none>    8861d93a820b   38 hours ago     470MB
openjdk                 latest    71260f256d19   12 months ago    470MB
```

Docker stop

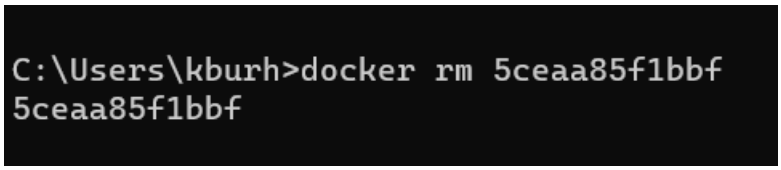
```
openjdk                 latest    71260f256d19

C:\Users\kburh>docker stop 5ceaa85f1bbf
5ceaa85f1bbf

C:\Users\kburh>
```

Docker restart

```
5ceaa85f1bbf

C:\Users\kburh>docker restart 5ceaa85f1bbf
5ceaa85f1bbf

C:\Users\kburh>
```

docker run

Docker rm



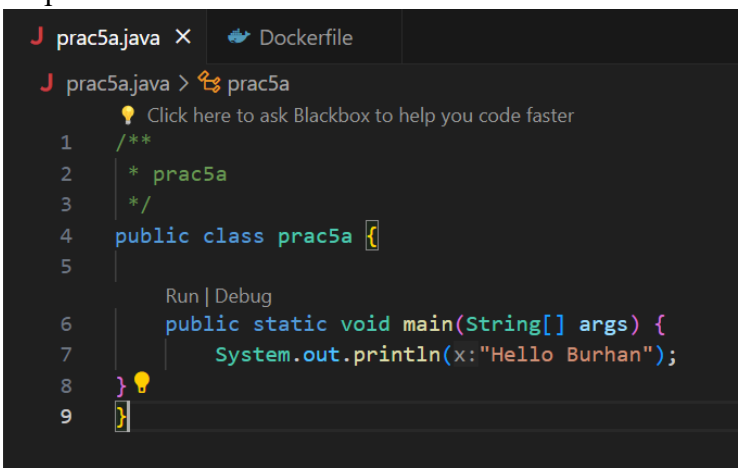
HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

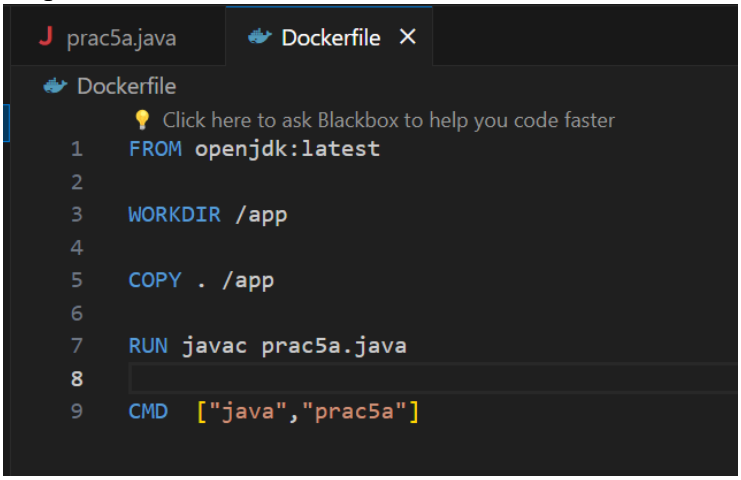
Practical 5 (Docker with Java)

A) Working with java – (“hello Burhan”)

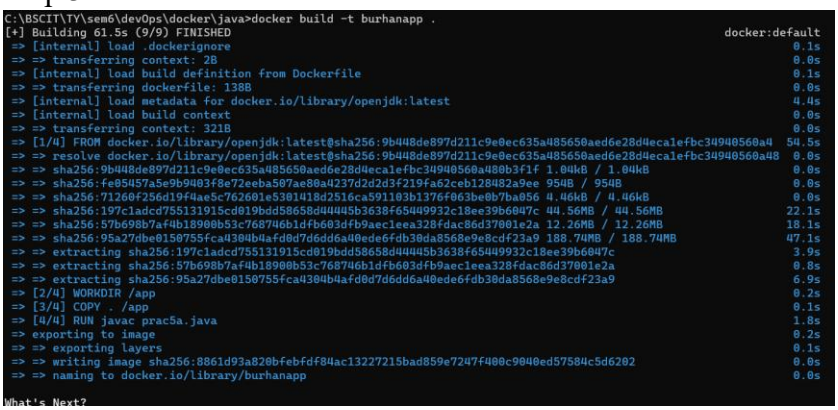
Step 1 Create Java File:



Step 2 Docker File:



Step 3 Docker build



Step 4 Docker run

```
C:\BSCIT\TY\sem6\devOps\docker\java>docker run burhanapp
Hello Burhan

C:\BSCIT\TY\sem6\devOps\docker\java>
```

B) Factorial

Step 1 Create Java Code:

prac5a.java | prac5b.java | Dockerfile

prac5b.java > prac5b > calculateFactorial(int)

Click here to ask Blackbox to help you code faster

```java
public class prac5b {
    // Burhanuddin Khairulla TYBSCIT 036
    Run | Debug
    public static void main(String[] args) {
        int number = 5;
        long factorial = calculateFactorial(number);
        System.out.println("Factorial of " + number + " is: " + factorial);
    }

    public static long calculateFactorial(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result *= i;
        }
        return result;
    }

}
```

Step 2 Create Docker File:

prac5a.java | prac5b.java | Dockerfile

Dockerfile

Click here to ask Blackbox to help you code faster

```
FROM openjdk:latest

WORKDIR /app

COPY . /app

RUN javac prac5b.java

CMD ["java","prac5b"]
```

Step 3 Docker build

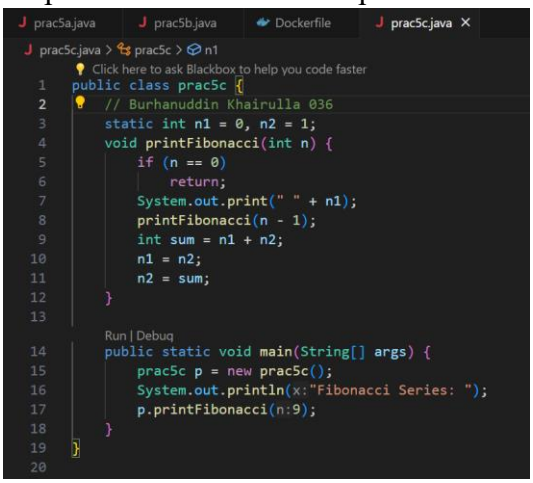
```
PS C:\BSCIT\TY\sem6\devOps\docker\java> docker build -t burhanapp .
[+] Building 4.5s (9/9) FINISHED                                                docker:default
 => [internal] load build definition from Dockerfile                                      0.0s
 => => transferring dockerfile: 138B                                                      0.0s
 => [internal] load .dockerignore                                                         0.0s
 => => transferring context: 2B                                                           0.0s
 => [internal] load metadata for docker.io/library/openjdk:latest                         1.8s
 => [1/4] FROM docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f   0.0s
 => [internal] load build context                                                         0.0s
 => => transferring context: 772B                                                         0.0s
 => CACHED [2/4] WORKDIR /app                                                             0.0s
 => [3/4] COPY . /app                                                                     0.0s
 => [4/4] RUN javac prac5b.java                                                           2.3s
 => exporting to image                                                                    0.1s
 => => exporting layers                                                                   0.1s
 => => writing image sha256:9452224764eb41b0d8cfe29b40cb3ca678e2235a7b222965dfbbac606dc853fb   0.0s
 => => naming to docker.io/library/burhanapp                                              0.0s
```

Step 4 Docker run

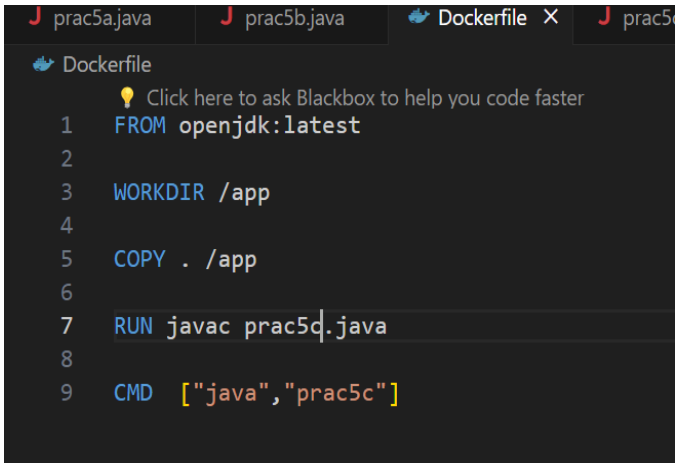
```
PS C:\BSCIT\TY\sem6\devOps\docker\java> docker run burhanapp
Factorial of 5 is: 120
```

C) Fibonacci

Step 1 Create Java code Step



2 Create Docker file:



Step 3 Docker build

```
C:\BSCIT\TY\sem6\devOps\docker\java>docker build -t burhanapp5c .
[+] Building 1.3s (9/9) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                        0.1s
 => => transferring dockerfile: 138B                                                        0.0s
 => [internal] load .dockerignore                                                           0.0s
 => => transferring context: 2B                                                             0.0s
 => [internal] load metadata for docker.io/library/openjdk:latest                           1.0s
 => [1/4] FROM docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a48  0.0s
 => [internal] load build context                                                           0.0s
 => => transferring context: 124B                                                           0.0s
 => CACHED [2/4] WORKDIR /app                                                               0.0s
 => CACHED [3/4] COPY . /app                                                                0.0s
 => CACHED [4/4] RUN javac prac5c.java                                                      0.0s
 => exporting to image                                                                      0.0s
 => => exporting layers                                                                     0.0s
 => => writing image sha256:094985e7c00637b9afe07177f448bbd529118edc16df56692a1d1ea45d2f7e7d  0.0s
 => => naming to docker.io/library/burhanapp5c                                              0.0s
```

Step 4 Docker run

```
C:\BSCIT\TY\sem6\devOps\docker\java>docker run burhanapp5c
0 1 1 2 3 5 8 13 21 34
C:\BSCIT\TY\sem6\devOps\docker\java>
```

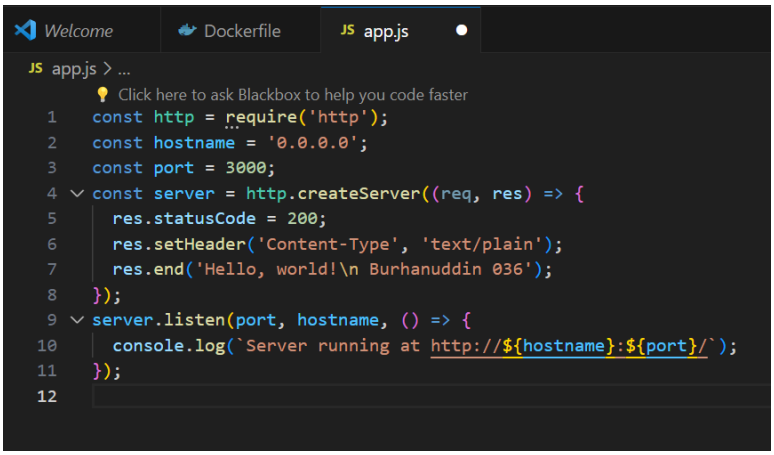
HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

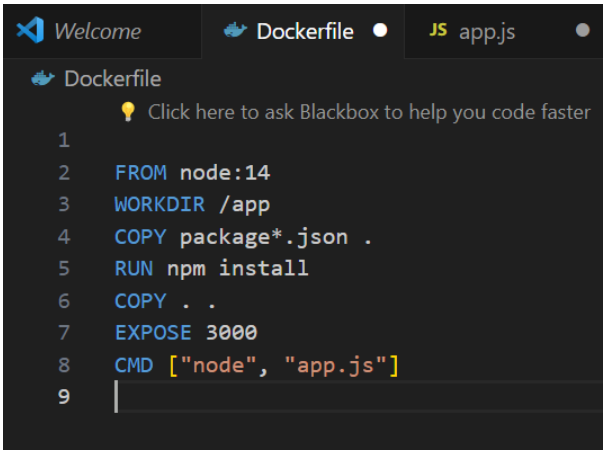
Practical 6

A) Working out docker with nodejs (Hello world!)

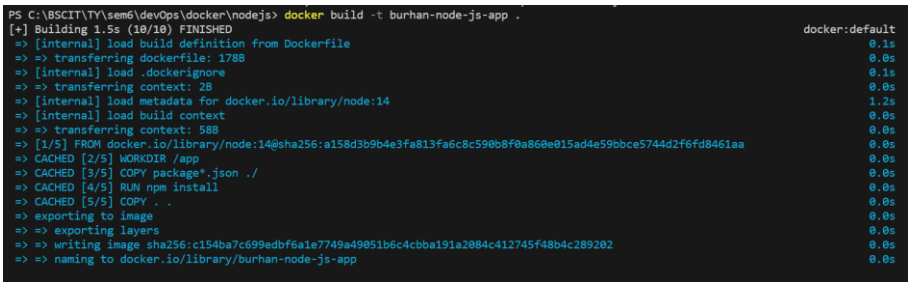
Step 1 - Create Nodejs file



Step 2 – Create Docker file



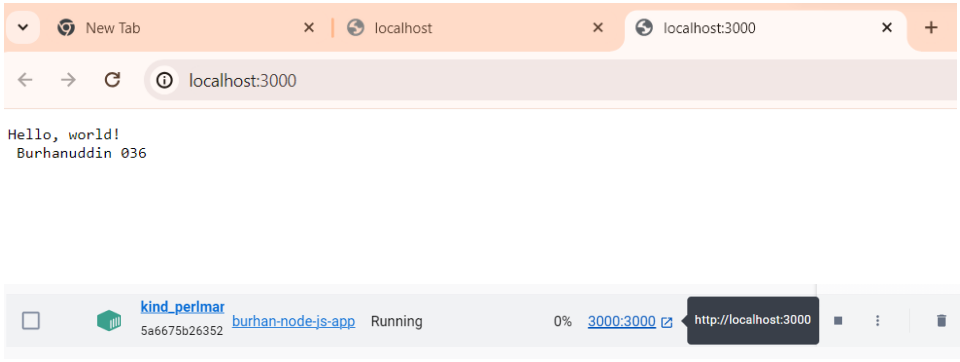
Step 3 - Docker build



Step 4 - Docker run

```
PS C:\BSCIT\TY\sem6\devOps\docker\nodejs> docker run -p 3000:3000 burhan-node-js-app
Server running at http://0.0.0.0:3000/
```

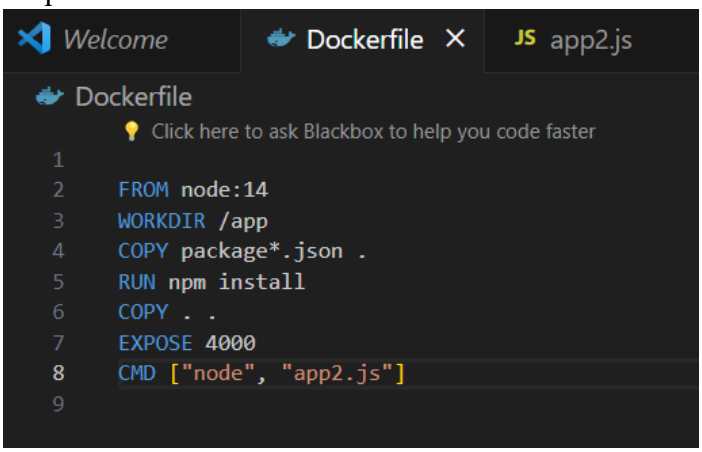
Server running



B) Odd and Even

Step 1 - Create Nodejs file

Step 2 - Create Docker file



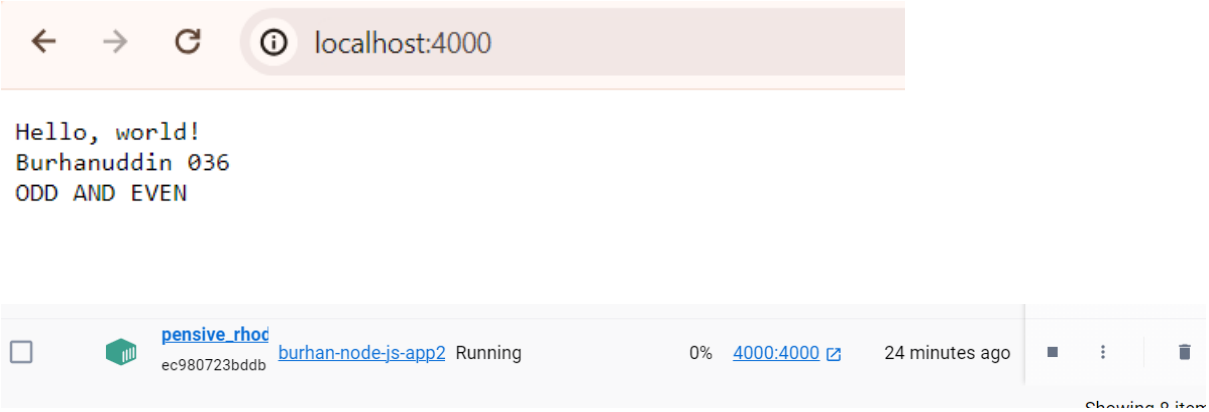
Step 3 - Docker build

```
PS C:\BSCIT\TY\sem6\devOps\docker\nodejs> docker build -t burhan-node-js-app2 .
[+] Building 1.6s (10/10) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                         0.1s
 => => transferring dockerfile: 154B                                                                                         0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [internal] load metadata for docker.io/library/node:14                                                                   1.3s
 => [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa             0.0s
 => [internal] load build context                                                                                            0.0s
 => => transferring context: 207B                                                                                            0.0s
 => CACHED [2/5] WORKDIR /app                                                                                                0.0s
 => CACHED [3/5] COPY package*.json .                                                                                        0.0s
 => CACHED [4/5] RUN npm install                                                                                             0.0s
 => [5/5] COPY . .                                                                                                           0.0s
 => exporting to image                                                                                                       0.1s
 => => exporting layers                                                                                                      0.1s
 => => writing image sha256:712bd09ab9d66a23a1b3ee7153d59d20d641c57def20403dc049a8ea38da573d                                 0.0s
 => => naming to docker.io/library/burhan-node-js-app2                                                                       0.0s

What's Next?
```

Step 4 - Docker run

```
PS C:\BSCIT\TY\sem6\devOps\docker\nodejs> docker run -p 4000:4000 burhan-node-js-app2
Server running at http://0.0.0.0:4000/
true
false
false
true
true
false
false
true
```

Server

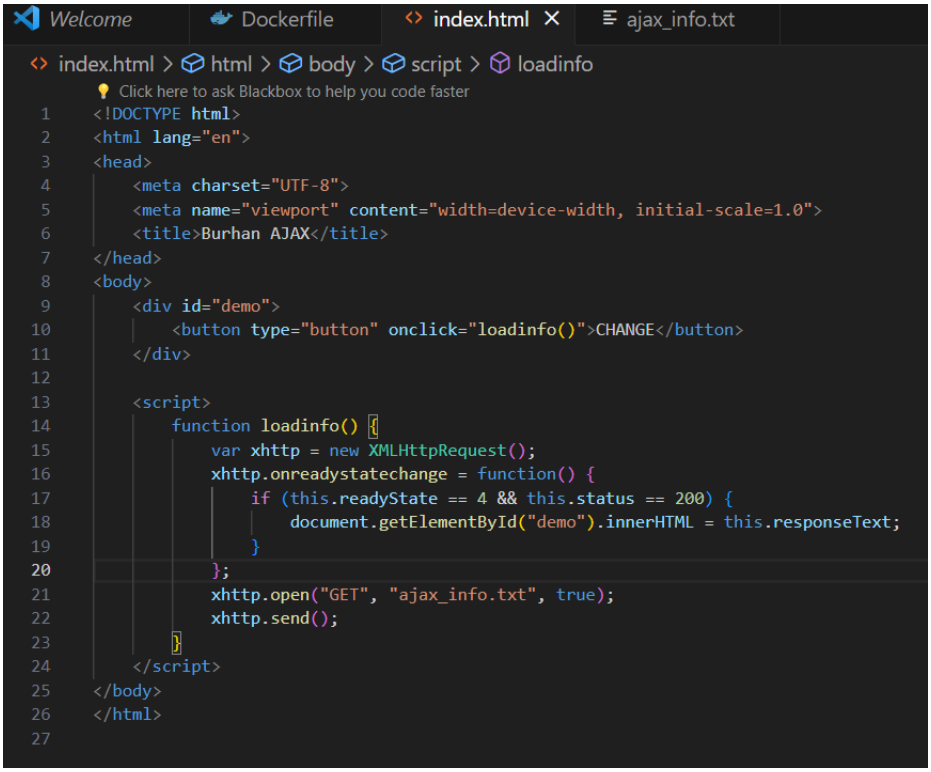


HSNC University , Mumbai

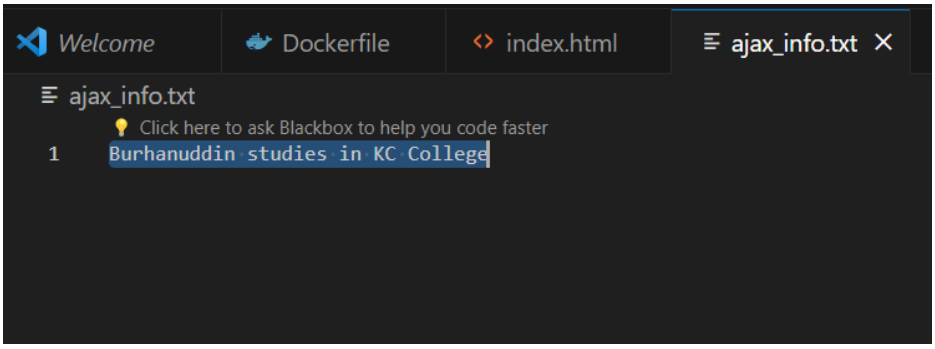
Kishinchand Chellaram College, Mumbai – 20.

Practical 7 – Working with ajax

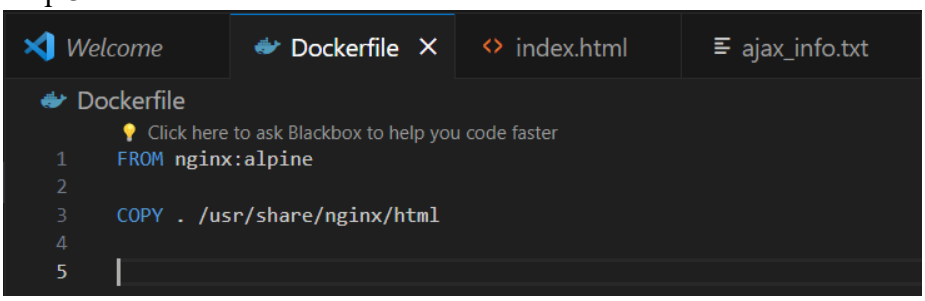
Step 1 - Create Index.html



Step 2 – Create Ajax info text file



Step 3 – Create Docker file



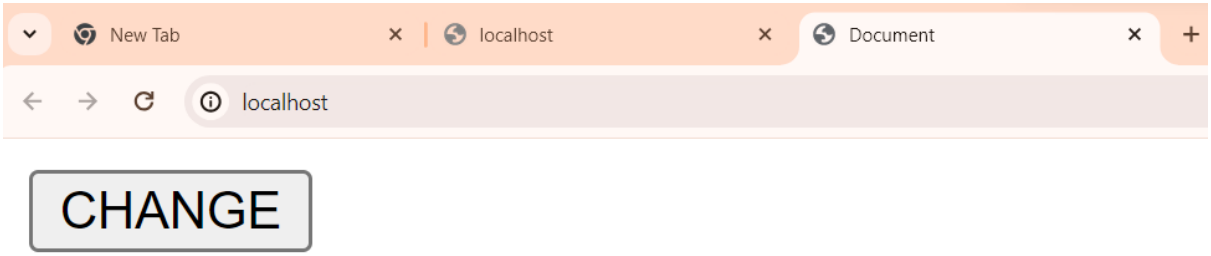
Step 4 - Docker build

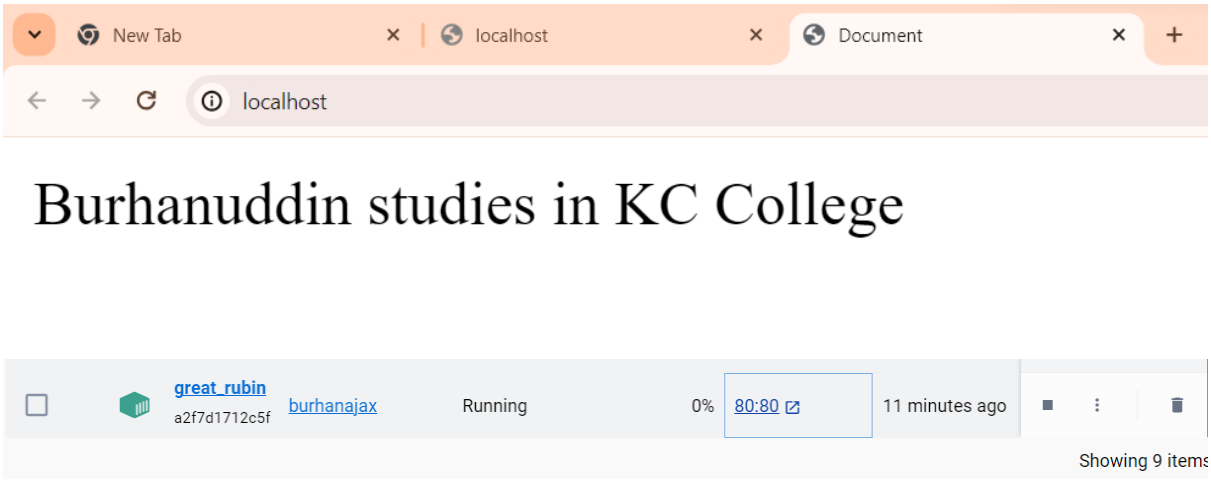
```
PS C:\BSCIT\TY\sem6\devOps\docker\ajax> docker build -t burhanajax .
[+] Building 1.0s (7/7) FINISHED                        docker:default
 => [internal] load build definition from Dockerfile             0.0s
 => => transferring dockerfile: 90B                              0.0s
 => [internal] load .dockerignore                                0.0s
 => => transferring context: 2B                                  0.0s
 => [internal] load metadata for docker.io/library/nginx:al      0.8s
 => [internal] load build context                                0.1s
 => => transferring context: 93B                                 0.1s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:6a2f8b      0.0s
 => CACHED [2/2] COPY . /usr/share/nginx/html                    0.0s
 => exporting to image                                           0.0s
 => => exporting layers                                          0.0s
 => => writing image sha256:a318a3d82b540f48262f194454471f3      0.0s
 => => naming to docker.io/library/burhanajax                    0.0s

What's Next?
```

Step 5 - Docker run

```
PS C:\BSCIT\TY\sem6\devOps\docker\ajax> docker run -d -p 80:80 burhanajax
a2f7d1712c5f7662da4bdddf2650d76993b15b5e28d3e261cf95f8b907127e46
PS C:\BSCIT\TY\sem6\devOps\docker\ajax>
```

Server



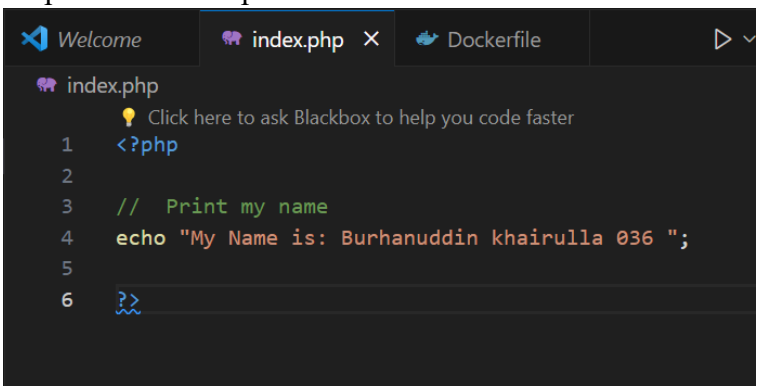


HSNC University , Mumbai

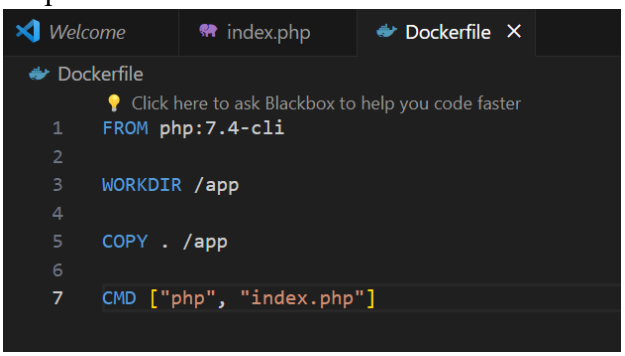
Kishinchand Chellaram College, Mumbai – 20.

Practical 8 – Working with PHP

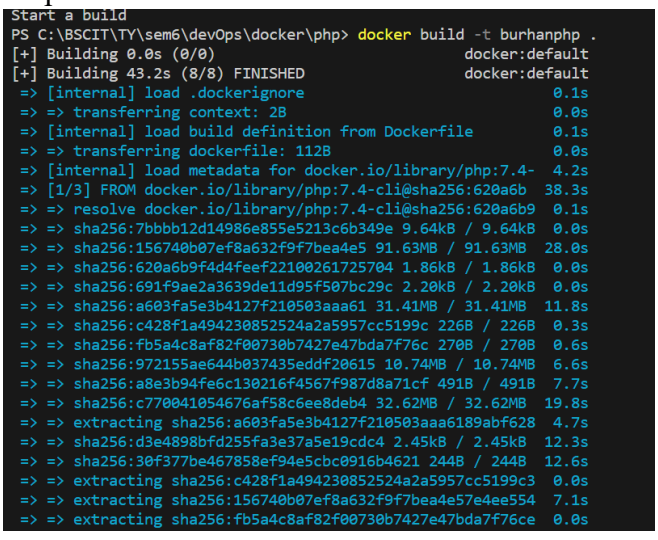
Step 1 – Create Php file



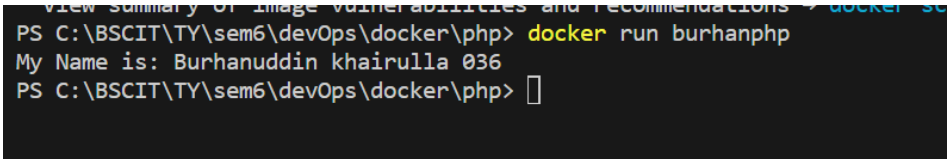
Step 2 – Create Docker file



Step 3 – Run Docker build



Step 4 – Docker run



HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

Practical 9 – Working docker kubertus Command

Minikube start

```
</body></html>
PS C:\Users\kburh> minikube start
W0220 20:44:58.469710    4800 main.go:291] Unable to resolve the current Docker CLI context "default": co
burh\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: Th
* minikube v1.32.0 on Microsoft Windows 11 Home Single Language 10.0.22621.3155 Build 22621.3155
* Automatically selected the docker driver
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
    > preloaded-images-k8s-v18-v1...:  403.35 MiB / 403.35 MiB  100.00% 2.96 Mi
    > gcr.io/k8s-minikube/kicbase...:  453.90 MiB / 453.90 MiB  100.00% 3.01 Mi
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\kburh>
```

Minikube stop

```
PS C:\Users\kburh> minikube stop
W0220 20:52:43.238931    2388 main.go:291] Unable to resolve the current Docker C
burh\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f413
* Stopping node "minikube"  ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\Users\kburh>
```

Minikube delete

```
PS C:\Users\kburh> minikube delete
W0220 20:55:06.320765     840 main.go:291] Unable to resolve the
burh\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2
* Deleting "minikube" in docker ...
* Deleting container "minikube" ...
* Removing C:\Users\kburh\.minikube\machines\minikube ...
* Removed all traces of the "minikube" cluster.
PS C:\Users\kburh>
```

Minikube status

```
PS C:\Users\kburh> minikube status
W0220 20:57:22.292897    5004 main.go:291] Unable to resolve th
burh\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164
* Profile "minikube" not found. Run "minikube profile list" to
  To start a cluster, run: "minikube start"
PS C:\Users\kburh>
```

Kubectl get nodes

```
* Done! kubectl is now configured to use "minikube" c
PS C:\Users\kburh> kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   56s   v1.28.3
PS C:\Users\kburh>
```

Kubectl service

```
PS C:\Users\kburh> kubectl get service -n default
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   7m59s
```

Kubectl cluster info

```
PS C:\Users\kburh> kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:51214
CoreDNS is running at https://127.0.0.1:51214/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Kubectl create pod

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl create -f burhan-pod.yaml
pod/burhan-pod created
```

Kubectl get pods

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl get pods
NAME         READY   STATUS             RESTARTS   AGE
burhan-pod   0/1     ImagePullBackOff   0          9s
```

Kubectl delete pod

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl delete pod burhan-pod
pod "burhan-pod" deleted
```

Kubectl logs pod

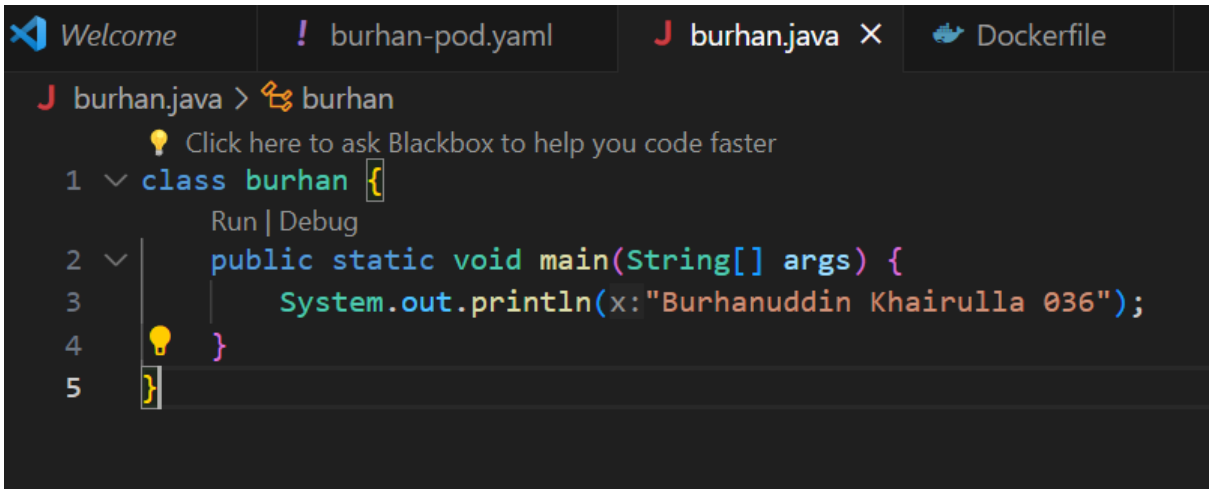
```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl logs burhan-pod
Error from server (BadRequest): container "my-container" in pod "burhan-pod" is waiting to start: trying and failing to
pull image
```

HSNC University , Mumbai

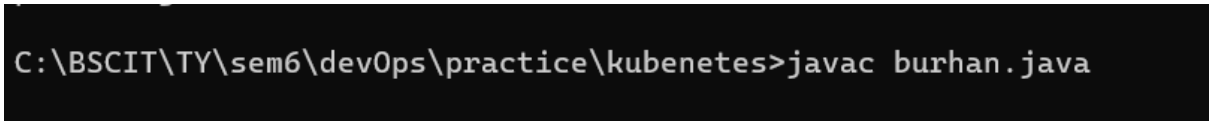
Kishinchand Chellaram College, Mumbai – 20.

Practical 10 – Working with Kube

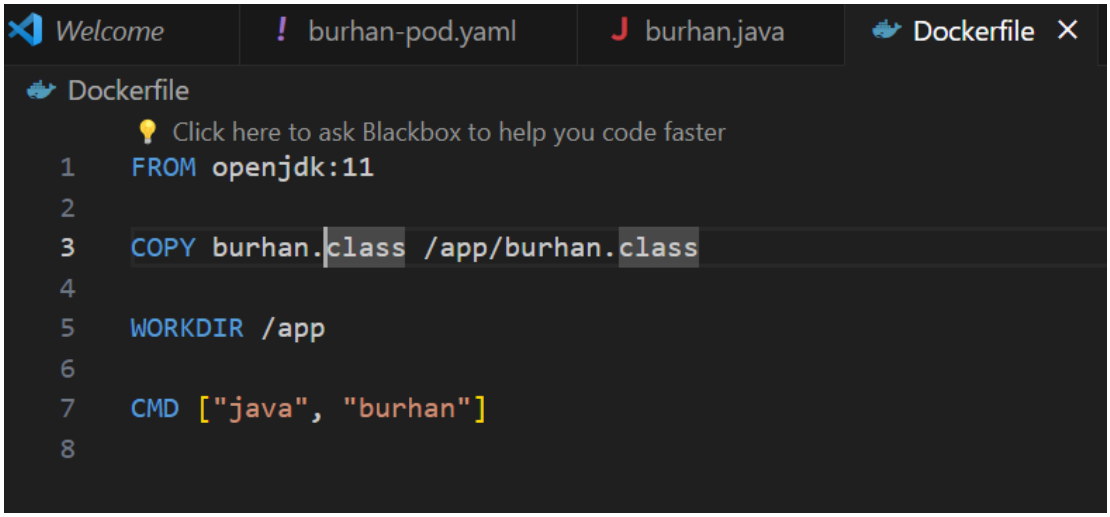
Step 1 - Create java file



Step 2 – Compile the java application



Step 3 – Create a dockerfile



Step 4 – Build docker image

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>docker build -t burhan-java .
[+] Building 66.6s (8/8) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 136B
 => [internal] load metadata for docker.io/library/openjdk:11
 => [1/3] FROM docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc
 => => resolve docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc
 => => sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9cdb7c21ab 1.04kB / 1.04kB
 => => sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a 10.88MB / 10.88MB
 => => sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452 55.00MB / 55.00MB
 => => sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165 5.16MB / 5.16MB
 => => sha256:e81b7f317654b0f26d3993e014b04bcb29250339b11b9de41e130feecd4cd43c 1.79kB / 1.79kB
 => => sha256:47a932d998b743b9b0bcce55aa8ede77de94a6a183c8a67dec9d5e3b8ce0faa7 6.26kB / 6.26kB
 => => sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbcb3a264c218c2ec7bca716e6 54.58MB / 54.58MB
 => => sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a1049710ebc4692de3dd5 5.42MB / 5.42MB
 => => sha256:66223a710990a0ae7162aeed80417d30303afa3f24aafa57aa30348725e2230b 213B / 213B
 => => sha256:db38d58ec8ab4111b072f6700f978a51985acd252aabce3be377f25162e68301 202.07MB / 202.07MB
 => => extracting sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452
 => => extracting sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165
 => => extracting sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a
 => => extracting sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbcb3a264c218c2ec7bca716e6
 => => extracting sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a1049710ebc4692de3dd5
 => => extracting sha256:66223a710990a0ae7162aeed80417d30303afa3f24aafa57aa30348725e2230b
 => => extracting sha256:db38d58ec8ab4111b072f6700f978a51985acd252aabce3be377f25162e68301
 => [internal] load build context
```

Step 5 – Run the docker container

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>docker run burhanjava
Burhanuddin Khairulla 036
```

Step 6 – Create a deployment file:

deploy-burhan.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kburh-java-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kburh-java
  template:
    metadata:
      labels:
        app: kburh-java
    spec:
      containers:
        - name: kburh-java-container
          image: burhankhair/kburh-java
```

Step 7 – Apply the Deployment to kubertenes

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl apply -f deploy-burhan.yaml
deployment.apps/kburh-java-deployment created
```

Step 8 – Expose the Deployment as a Service:

service.yaml | burhan-pod.yaml | burhan.java

service.yaml

Click here to ask Blackbox to help you code faster

```yaml
apiVersion: v1
kind: Service
metadata:
  name: kburh-java-service
spec:
  selector:
    app: kburh-java
  ports:
  - protocol: TCP
    port: 80
    targetPort: 8080
  type: LoadBalancer

```

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl apply -f service.yaml
service/kburh-java-service created
```

Step 9 – kubectl get svc kburh-java-service

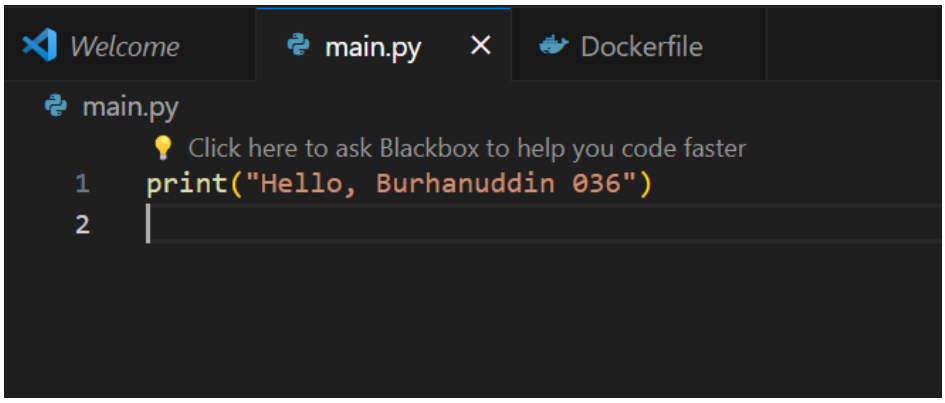
```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>kubectl get svc kburh-java-service
NAME                TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kburh-java-service  LoadBalancer   10.97.144.17   <pending>     80:32173/TCP   26s
```

HSNC University , Mumbai

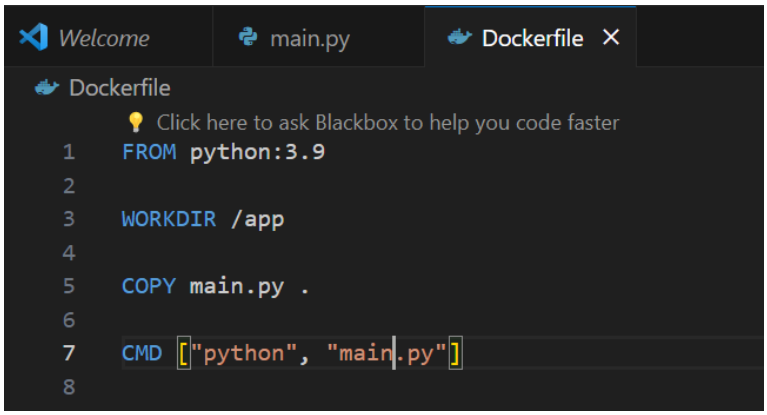
Kishinchand Chellaram College, Mumbai – 20.

Practical 11 – Working with python

Step 1 – Create Python file



Step 2 Create Docker file



Step 3 - Docker build

```
PS C:\BSCIT\TY\sem6\devOps\docker\python> docker build -t burhan-python-app .
[+] Building 91.3s (8/8) FINISHED                                                                                    docker:default
 => [internal] load .dockerignore                                                                                              0.0s
 => => transferring context: 2B                                                                                                0.0s
 => [internal] load build definition from Dockerfile                                                                           0.0s
 => => transferring dockerfile: 117B                                                                                           0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                                  0.5s
 => [1/3] FROM docker.io/library/python:3.9@sha256:383d072c4b840507f25453c710969aa1e1d13e47731f294a8a8890e53f834bdf           89.8s
 => => resolve docker.io/library/python:3.9@sha256:383d072c4b840507f25453c710969aa1e1d13e47731f294a8a8890e53f834bdf            0.0s
 => => sha256:383d072c4b840507f25453c710969aa1e1d13e47731f294a8a8890e53f834bdf 1.86kB / 1.86kB                                  0.0s
 => => sha256:530d4ba717be787c0e2d011aa107edac6d721f8c06fe6d44708d4aa5e9bc5ec9 2.01kB / 2.01kB                                  0.0s
 => => sha256:e301b6ca47814a93ddd0420cffbe960bc5363a43d3e8644e53d547ffce16a7a5 7.33kB / 7.33kB                                  0.0s
 => => sha256:7bb465c2914923b08ae03b7fc67b92a1ef9b09c4c1eb9d6711b22ee6bbb46a00 49.55MB / 49.55MB                               17.0s
 => => sha256:2b9b41aaa3c52ab268b47da303015b94ced04a1eb02e58860e58b283404974f4 24.05MB / 24.05MB                               17.9s
 => => sha256:49b40be4436eff6fe463f6977159dc727df37cabe65ade75c75c1caa3cb0a234 64.14MB / 64.14MB                               35.4s
 => => sha256:c558fac597f8ecbb7a66712e14912ce1d83b23a92ca8b6ff14eef209ab01aff2 211.12MB / 211.12MB                             61.9s
 => => extracting sha256:7bb465c2914923b08ae03b7fc67b92a1ef9b09c4c1eb9d6711b22ee6bbb46a00                                      6.7s
 => => sha256:11402150a57e537c64dc69a28bba37f13acdedd50d8788894398a7b774786e7d 6.39MB / 6.39MB                                 21.0s
 => => sha256:d662c27d7e5762cd7807a257f6ec30067c714fdd9a9390836991830c0fbbbb47 15.82MB / 15.82MB                               28.0s
 => => extracting sha256:2b9b41aaa3c52ab268b47da303015b94ced04a1eb02e58860e58b283404974f4                                      1.7s
 => => sha256:26ff257bfcf12b2ad8a2527c7a42179ff6febdddde7064dec546ec0869a4627 242B / 242B                                     28.8s
 => => sha256:5a8f6c8c0bd20abb177c033e0a9dc88085bb9ae78d41671925372d005c7ca048 2.85MB / 2.85MB                                 30.5s
 => => extracting sha256:49b40be4436eff6fe463f6977159dc727df37cabe65ade75c75c1caa3cb0a234                                      7.6s
 => => extracting sha256:c558fac597f8ecbb7a66712e14912ce1d83b23a92ca8b6ff14eef209ab01aff2                                     23.5s
 => => extracting sha256:11402150a57e537c64dc69a28bba37f13acdedd50d8788894398a7b774786e7d                                      1.1s
 => => extracting sha256:d662c27d7e5762cd7807a257f6ec30067c714fdd9a9390836991830c0fbbbb47                                      1.8s
 => => extracting sha256:26ff257bfcf12b2ad8a2527c7a42179ff6febdddde7064dec546ec0869a4627                                      0.0s
 => => extracting sha256:5a8f6c8c0bd20abb177c033e0a9dc88085bb9ae78d41671925372d005c7ca048                                      0.6s
 => [internal] load build context                                                                                              0.0s
 => => transferring context: 67B                                                                                               0.0s
 => [2/3] WORKDIR /app                                                                                                         0.5s
 => [3/3] COPY main.py .                                                                                                       0.1s
 => exporting to image                                                                                                         0.1s
 => => exporting layers                                                                                                        0.0s
```

Step 4 - Docker run

```
  View summary of image vulnerabilities and recommendations → docker scout quic
PS C:\BSCIT\TY\sem6\devOps\docker\python> docker run burhan-python-app
Hello, Burhanuddin 036
PS C:\BSCIT\TY\sem6\devOps\docker\python>
```