

DEVOPS

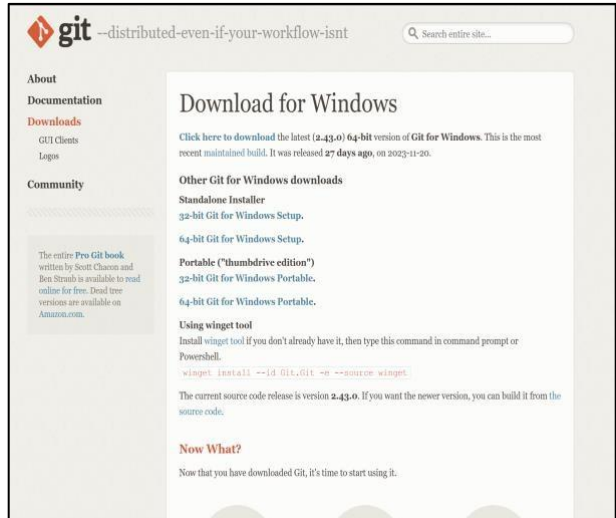
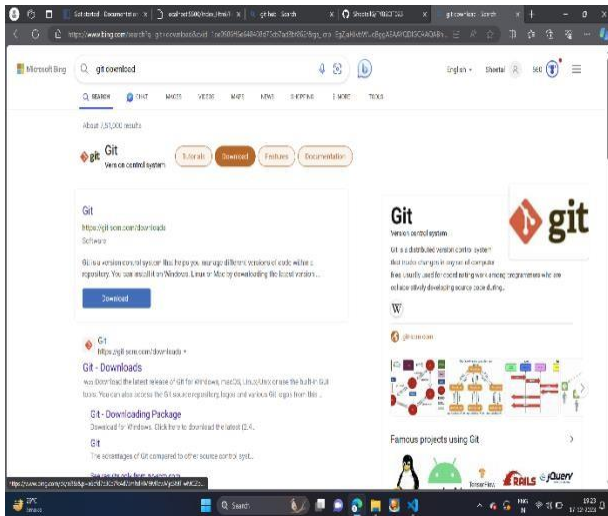
PRACTICALS

Name: Burhanuddin Khairulla

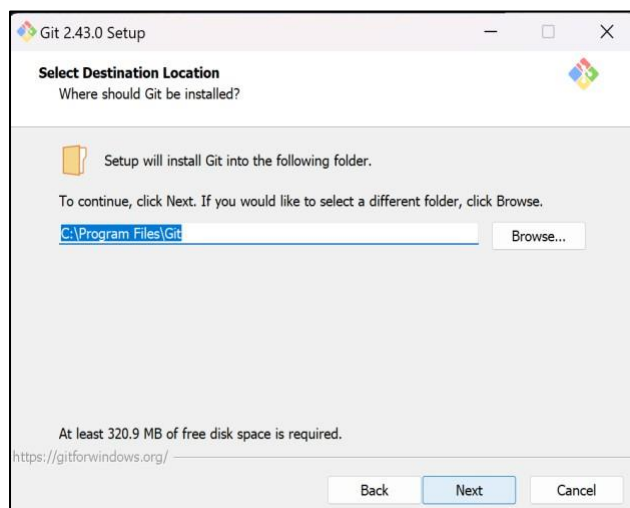
Roll No: 036

Practical No 1: Installation of Git

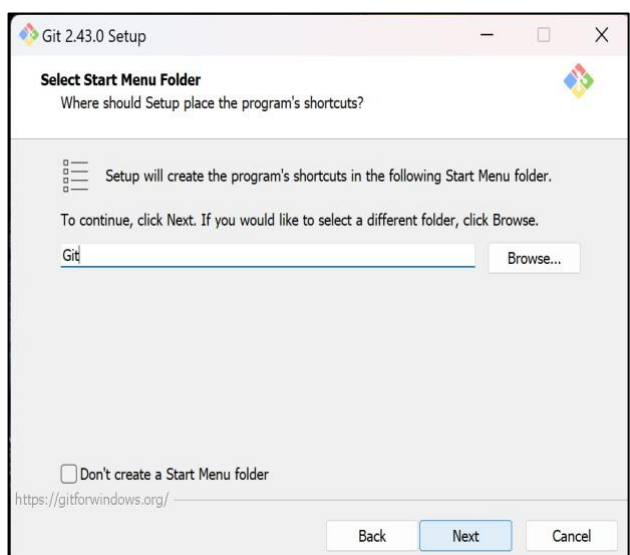
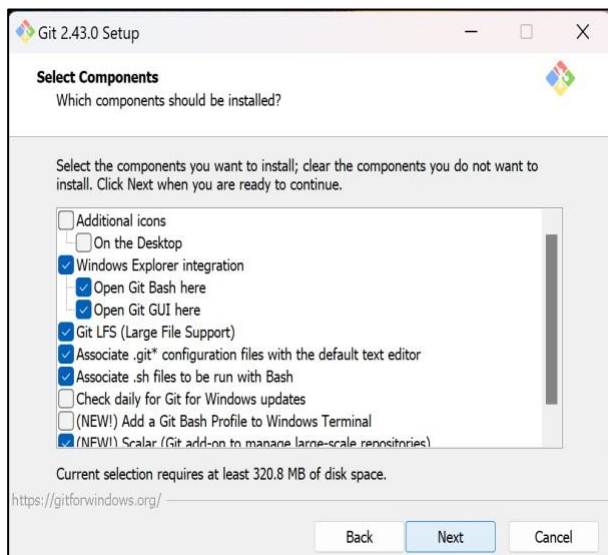
Step 1: Download the latest version of GIT and choose either 64/32 bit.



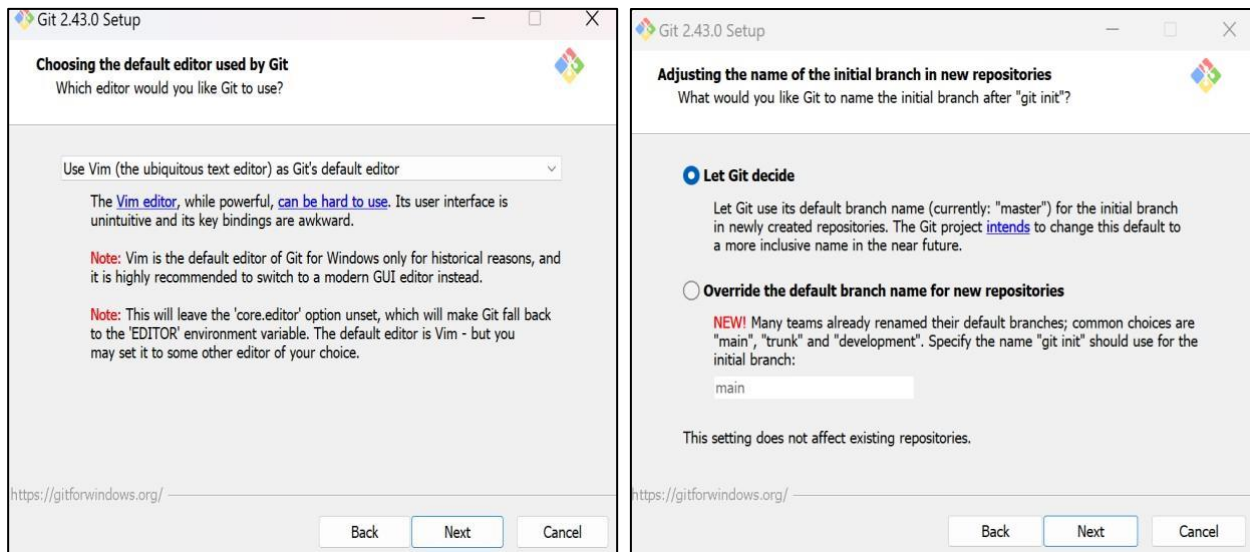
Step 2: After the file is downloaded, install it in the system, once install double click or extract and launch the installer. Go to GNU (General Public License). Click Next. Choose the installation location of your choice and click Next.



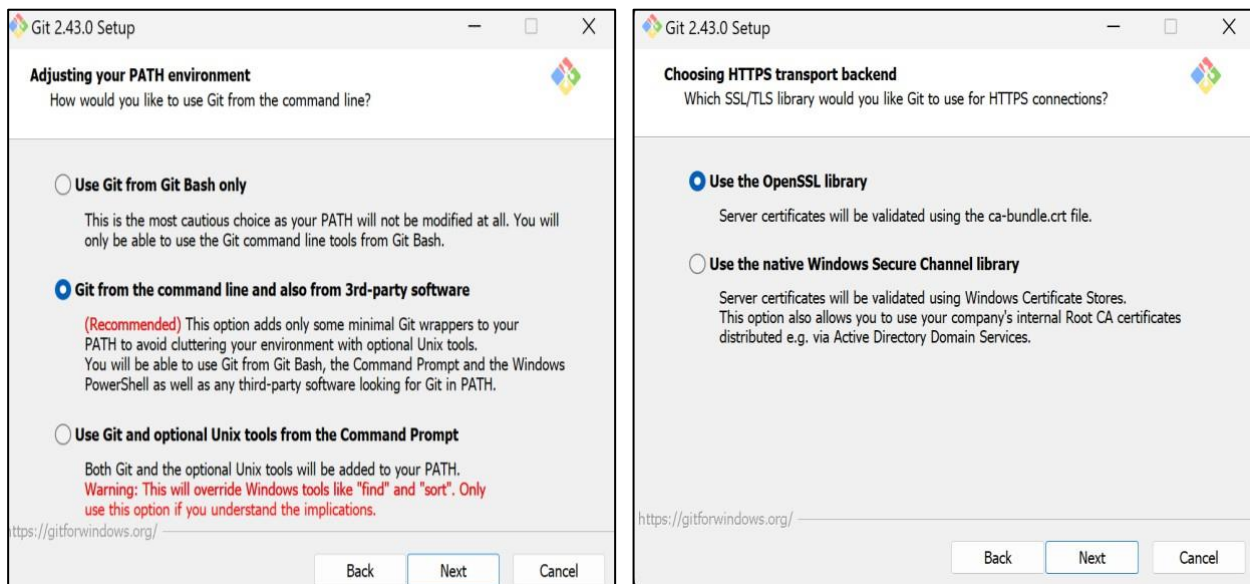
Step 3: Leave the defaulted selected component and click next. Choose the start menu folder location and click on next.



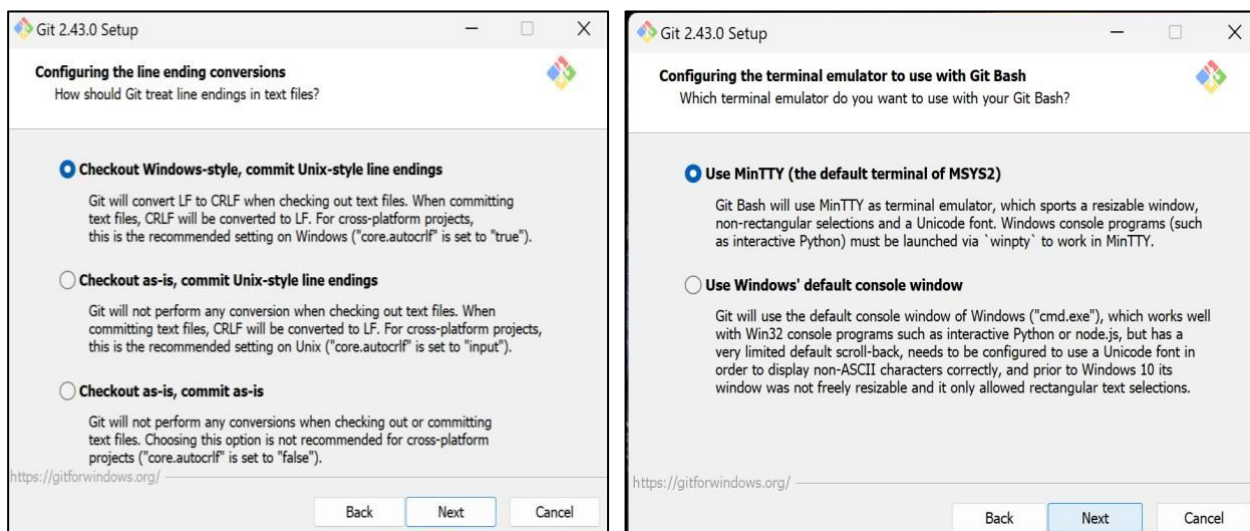
Step 4: Choose VIM/Text editor you would like to use with Git and click next.



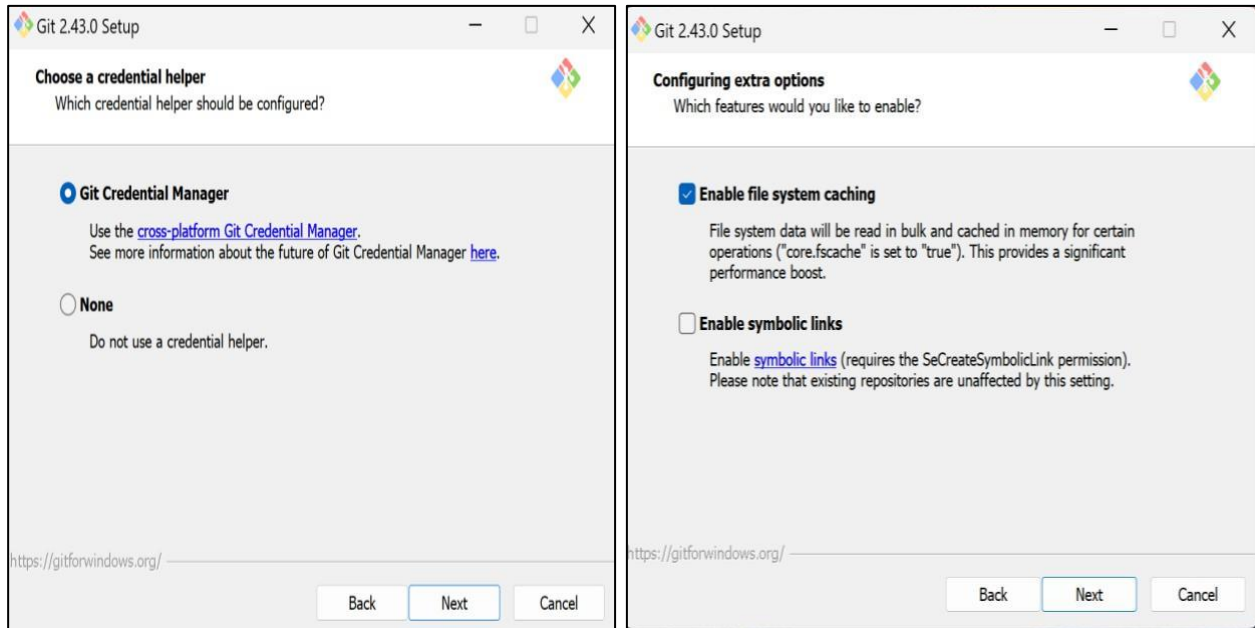
Step 5: Select Git from command line and also from third party software and click on next. Select the server certificate of your choice the default one is used to open SSL library and click next.



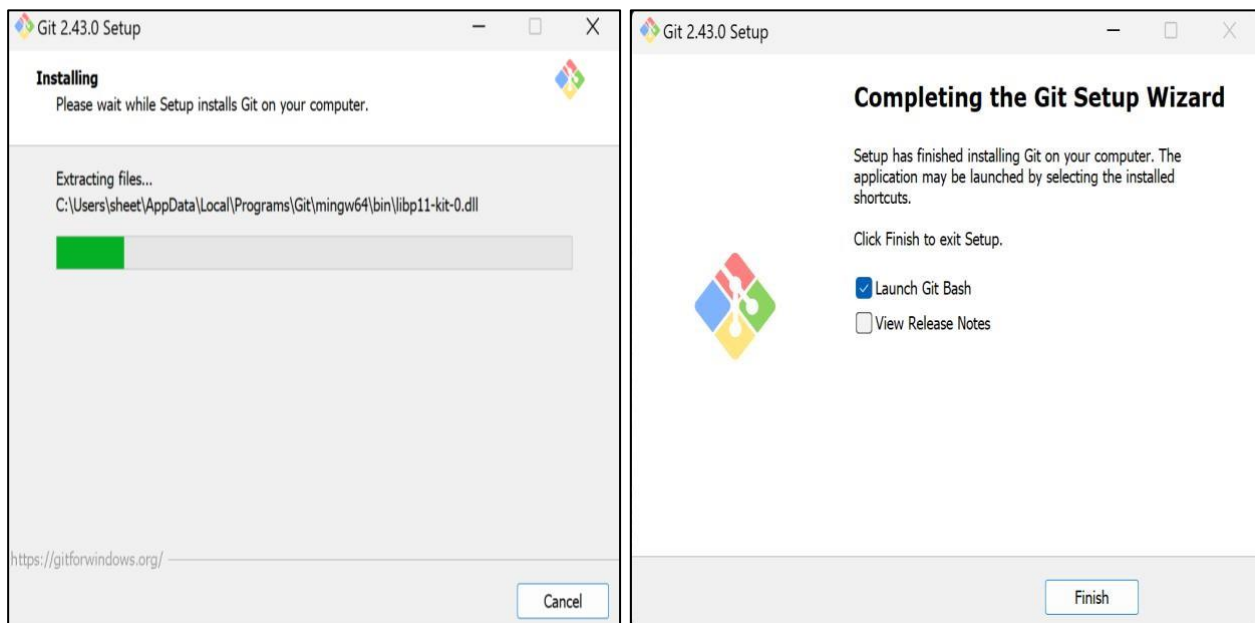
Step 6: Choose the checkout window style, commit unix-style line endings and click next. Select the default MINTTY and click next.



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

New repository

Search Type to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner * burhankhair

Repository name * TYBSCIT036

TYBSCIT036 is available.

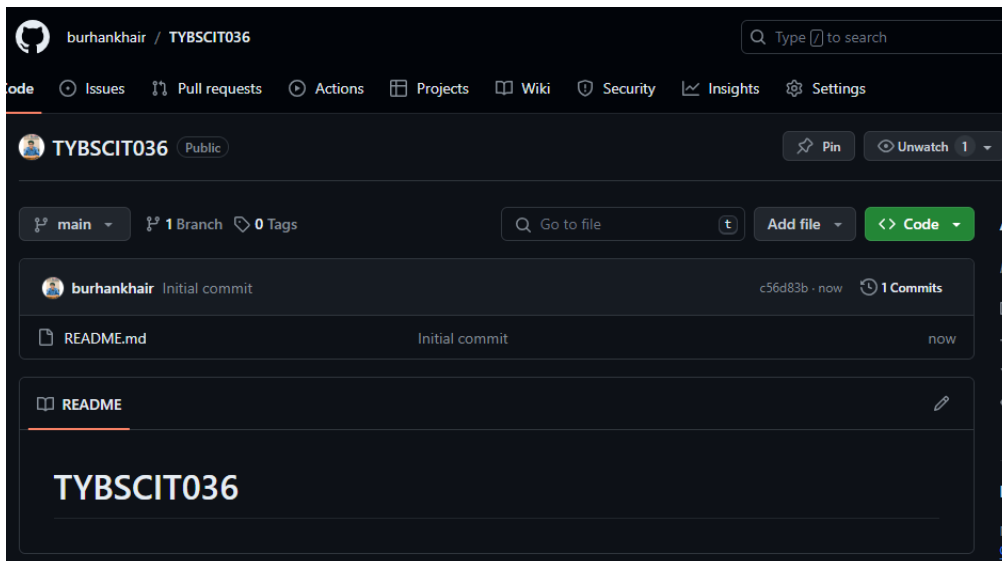
Great repository names are short and memorable. Need inspiration? How about [fuzzy-engine](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private



Step 2: Open Git Bash -> Create a directory

```
admin@COL-MSCIT-PC4 MINGW32 /d
$ mkdir burhan

admin@COL-MSCIT-PC4 MINGW32 /d
$ cd burhan

admin@COL-MSCIT-PC4 MINGW32 /d/burhan
$ |
```

HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

Step 3: Git Commands.

- a- git config --global user.name "burhankhair"
- b- git init

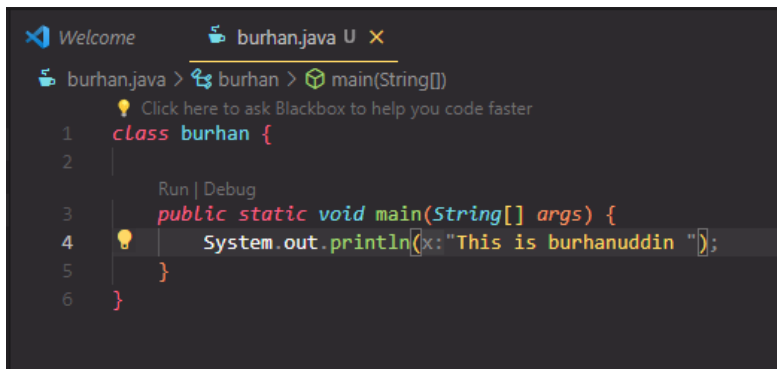
```
admin@COL-MSCIT-PC4 MINGW32 /d/burhan
$ git config --global user.name "burhankhair"

admin@COL-MSCIT-PC4 MINGW32 /d/burhan
$ git config --global user.email "k.burhankhair@gmail.com"

admin@COL-MSCIT-PC4 MINGW32 /d/burhan
$ git init
Initialized empty Git repository in D:/burhan/.git/

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

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



```
Welcome  burhan.java U x
burhan.java > burhan > main(String[])
Click here to ask Blackbox to help you code faster
1  class burhan {
2
3      Run | Debug
4      public static void main(String[] args) {
5          System.out.println("This is burhanuddin ");
6      }
}
```

Step 5: Git commands

Git status

```
$ git status
On branch master

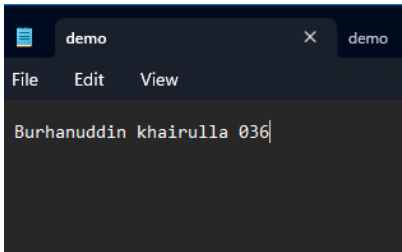
No commits yet

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

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

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

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

nothing added to commit but untracked files present (use "git add" to track)
admin@COL-MSCIT-PC4 MINGW32 /d/burhan (master)
$
```

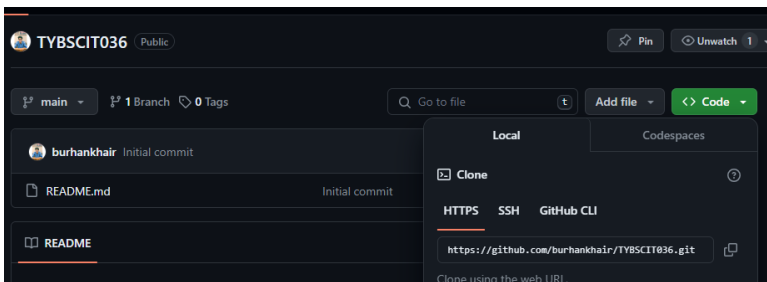
Git commit

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

 tybscit036-b Public


 Pin

 Unwatch 1

 master had recent pushes 3 seconds ago

[Compare & pull request](#)

 main  1 Branch  0 Tags

 Go to file




Add file

 Code

 burhankhair Initial commit

031773b · 1 minute ago

 1 Commits

 README.md

Initial commit

1 minute ago

HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

Practical 2B : Perform 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
```

HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

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 repository

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

Practical 3 – Docker Installation

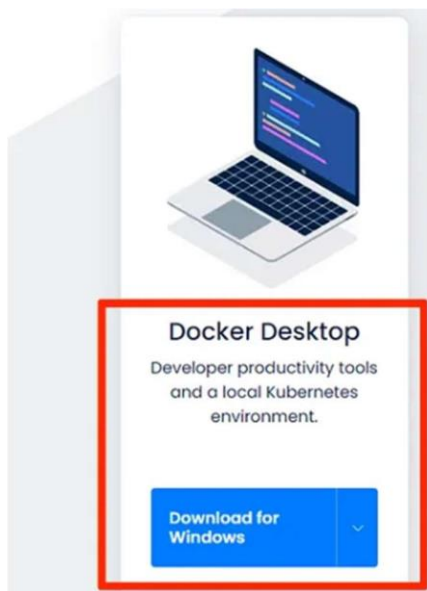
Install Docker on Windows Step 1:

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.



HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

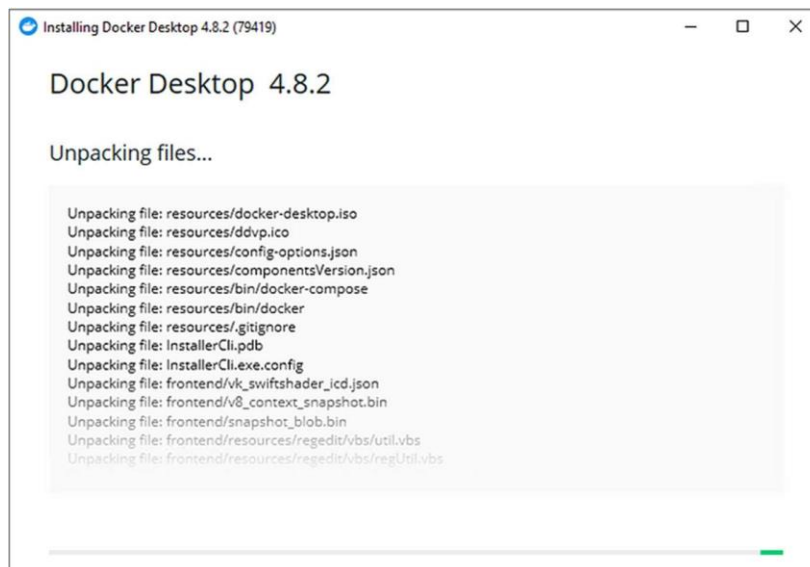
Step 2: Configuration

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



Step 3: Running the installation

Click Ok, and wait a bit...



HSNC University , Mumbai

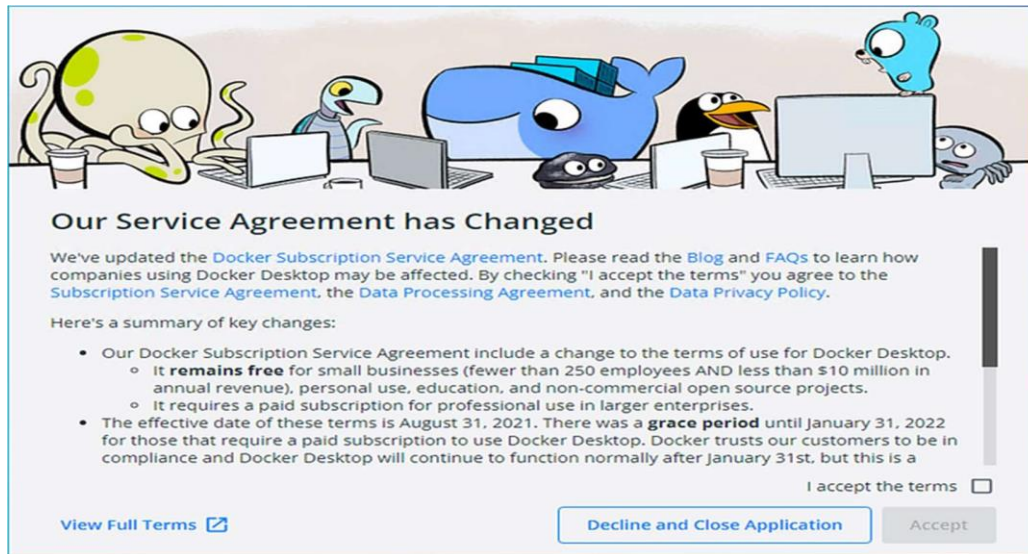
Kishinchand Chellaram College, Mumbai – 20.

Step 4: Restart

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

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 MySQL... | 9 | | |
| google/mysql | MySQL server for Google Compute Engine | 25 | | [OK] |
| rapidfort/mysql-official | RapidFort optimized, hardened image for MySQL... | 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] |

```
C:\Users\kburh>
```

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: sha256:9b448de897d211c9e0ec635a485650aed6e28d4ecalefbc34940560a480b3f1f
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 |

```
C:\Users\kburh>
```

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

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

Docker rm

```
C:\Users\kburh>docker rm 5ceaa85f1bbf  
5ceaa85f1bbf
```

Practical 5 (Docker with Java)

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

Step 1 Create Java File:

```

J prac5a.java X Dockerfile

J prac5a.java > prac5a

1  /**
2   * prac5a
3   */
4  public class prac5a {
5
6      public static void main(String[] args) {
7          System.out.println(x:"Hello Burhan");
8      }
9  }

```

Step 2 Docker File:

```

J prac5a.java Dockerfile X

Dockerfile

1  FROM openjdk:latest
2
3  WORKDIR /app
4
5  COPY . /app
6
7  RUN javac prac5a.java
8
9  CMD ["java","prac5a"]

```

Step 3 Docker build

```

C:\BSCIT\TV\sem6\devOps\docker\java>docker build -t burhanapp .
[+] Building 61.5s (9/9) FINISHED
=> [internal] load .dockerignore 0.1s
=> [internal] load build definition from Dockerfile 0.1s
=> [internal] load metadata for docker.io/library/openjdk:latest 0.0s
=> [internal] load build context 0.0s
=> [internal] load metadata for docker.io/library/openjdk:latest 0.0s
=> [1/4] FROM docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4ecafebc34940560a48 54.5s
=> resolve docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4ecafebc34940560a48 0.0s
=> sha256:9b448de897d211c9e0ec635a485650aed6e28d4ecafebc34940560a480b3f1f 1.04kB / 1.04kB 0.0s
=> sha256:fe95457a5e9b9403f9e72eeba507ae8a4237d2d3f219fa52c6b12b482a9ee 95kB / 95kB 0.0s
=> sha256:71268f256d19f4a5c762601e5301418d2516ca591103b1376f063be0b7ba856 4.46kB / 4.46kB 0.0s
=> sha256:197c1adcd755131915cd019bdd58658d4445b3638f65449932c18ee39b6047c 44.56MB / 44.56MB 22.1s
=> sha256:57b698b7af4b18900b53c768746b1dfb603dfb9aee1ee328fdac86d37001e2a 12.26MB / 12.26MB 18.1s
=> sha256:95a27d0e0150755fca4304b4afdb0d7d6dd6a40ede6fdb38da8568e9e8cdf23a9 188.74MB / 188.74MB 47.1s
=> extracting sha256:197c1adcd755131915cd019bdd58658d4445b3638f65449932c18ee39b6047c 3.9s
=> extracting sha256:57b698b7af4b18900b53c768746b1dfb603dfb9aee1ee328fdac86d37001e2a 0.8s
=> extracting sha256:95a27d0e0150755fca4304b4afdb0d7d6dd6a40ede6fdb38da8568e9e8cdf23a9 0.9s
=> [2/4] WORKDIR /app 0.2s
=> [3/4] COPY . /app 0.1s
=> [4/4] RUN javac prac5a.java 1.8s
=> exporting to image 0.2s
=> exporting layers 0.1s
=> writing image sha256:8861d93a828bfebfdf84c13227215bad859e7247f400c9040ed57584c5d6282 0.8s
=> naming to docker.io/library/burhanapp 0.8s

What's Next?

```

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:

```
J prac5a.java  J prac5b.java X  Dockerfile
J prac5b.java >  J prac5b > calculateFactorial(int)
Click here to ask Blackbox to help you code faster
1 public class prac5b {
2     // Burhanuddin Khairulla TYBSCIT 036
3     Run | Debug
4     public static void main(String[] args) {
5         int number = 5;
6         long factorial = calculateFactorial(number);
7         System.out.println("Factorial of " + number + " is: " + factorial);
8     }
9
10    public static long calculateFactorial(int n) {
11        long result = 1;
12        for (int i = 1; i <= n; i++) {
13            result *= i;
14        }
15        return result;
16    }
17 }
18 }
```

Step 2 Create Docker File:

```
J prac5a.java X  J prac5b.java  Dockerfile X
Dockerfile
Click here to ask Blackbox to help you code faster
1 FROM openjdk:latest
2
3 WORKDIR /app
4
5 COPY . /app
6
7 RUN javac prac5b.java
8
9 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

```
prac5c.java > n1
Click here to ask Blackbox to help you code faster
1 public class prac5c {
2     // Burhanuddin Khairulla 036
3     static int n1 = 0, n2 = 1;
4     void printFibonacci(int n) {
5         if (n == 0)
6             return;
7         System.out.print(" " + n1);
8         printFibonacci(n - 1);
9         int sum = n1 + n2;
10        n1 = n2;
11        n2 = sum;
12    }
13
14    Run | Debug
15    public static void main(String[] args) {
16        prac5c p = new prac5c();
17        System.out.println(x:"Fibonacci Series: ");
18        p.printFibonacci(n:9);
19    }
20 }
```

2 Create Docker file:

```
prac5a.java | prac5b.java | Dockerfile X | prac5c.java
Dockerfile
Click here to ask Blackbox to help you code faster
1 FROM openjdk:latest
2
3 WORKDIR /app
4
5 COPY . /app
6
7 RUN javac prac5c.java
8
9 CMD ["java", "prac5c"]
```

Step 3 Docker build

```
C:\BSCIT\TY\sem6\devOps\docker\java>docker build -t burhanapp5c .
[+] Building 1.3s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 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>
```

Practical 6

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

Step 1 - Create Nodejs file

```
JS app.js > ...
Click here to ask Blackbox to help you code faster
1 const http = require('http');
2 const hostname = '0.0.0.0';
3 const port = 3000;
4 const server = http.createServer((req, res) => {
5   res.statusCode = 200;
6   res.setHeader('Content-Type', 'text/plain');
7   res.end('Hello, world!\n Burhanuddin 036');
8 });
9 server.listen(port, hostname, () => {
10   console.log(`Server running at http://${hostname}:${port}/`);
11 });
12
```

Step 2 – Create Docker file

```
Dockerfile
Click here to ask Blackbox to help you code faster
1
2 FROM node:14
3 WORKDIR /app
4 COPY package*.json .
5 RUN npm install
6 COPY . .
7 EXPOSE 3000
8 CMD ["node", "app.js"]
9
```

Step 3 - Docker build

```
PS C:\BSCIT\TY\sem6\devOps\docker\nodejs> docker build -t burhan-node-js-app .
[*] Building 1.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 178B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:14
=> [internal] load build context
=> => transferring context: 58B
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa
=> => transferring context: 2B
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY package*.json ./
=> CACHED [4/5] RUN npm install
=> CACHED [5/5] COPY . .
=> exporting to image
=> => writing image sha256:c154ba7c699edbf6a1e7749a49051b6c4cbb191a2084c412745f48b4c289282
=> => naming to docker.io/library/burhan-node-js-app
```

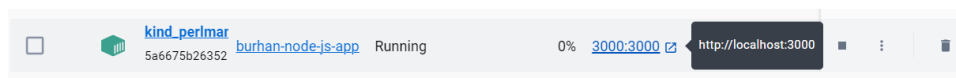
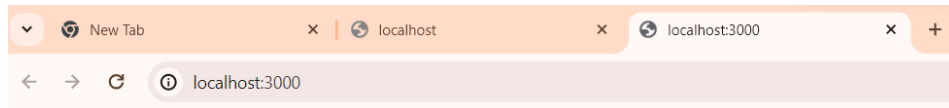
HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

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/
```

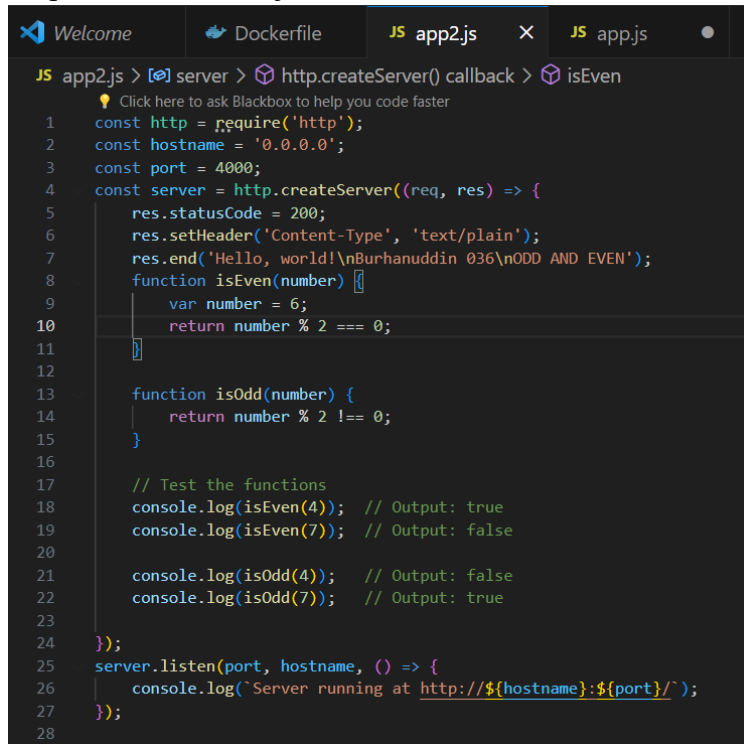
Server running



HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

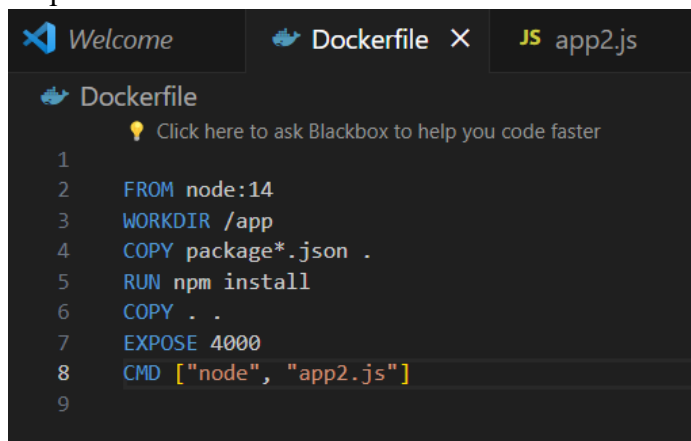
B) Odd and Even

Step 1 - Create Nodejs file



```
JS app2.js > [?] server > http.createServer() callback > isEven
Click here to ask Blackbox to help you code faster
1  const http = require('http');
2  const hostname = '0.0.0.0';
3  const port = 4000;
4  const server = http.createServer((req, res) => {
5      res.statusCode = 200;
6      res.setHeader('Content-Type', 'text/plain');
7      res.end('Hello, world!\nBurhanuddin 036\nODD AND EVEN');
8      function isEven(number) {
9          var number = 6;
10         return number % 2 === 0;
11     }
12
13     function isOdd(number) {
14         return number % 2 !== 0;
15     }
16
17     // Test the functions
18     console.log(isEven(4)); // Output: true
19     console.log(isEven(7)); // Output: false
20
21     console.log(isOdd(4)); // Output: false
22     console.log(isOdd(7)); // Output: true
23
24 });
25 server.listen(port, hostname, () => {
26     console.log(`Server running at http://${hostname}:${port}/`);
27 });
28
```

Step 2 - Create Docker file



```
VS Welcome Dockerfile X JS app2.js
Dockerfile
Click here to ask Blackbox to help you code faster
1
2 FROM node:14
3 WORKDIR /app
4 COPY package*.json .
5 RUN npm install
6 COPY . .
7 EXPOSE 4000
8 CMD ["node", "app2.js"]
9
```


HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

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
=> [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




 localhost:4000

Hello, world!
Burhanuddin 036
ODD AND EVEN

 **pensive_rhodes**
ec980723bddb

[burhan-node-js-app2](#) Running

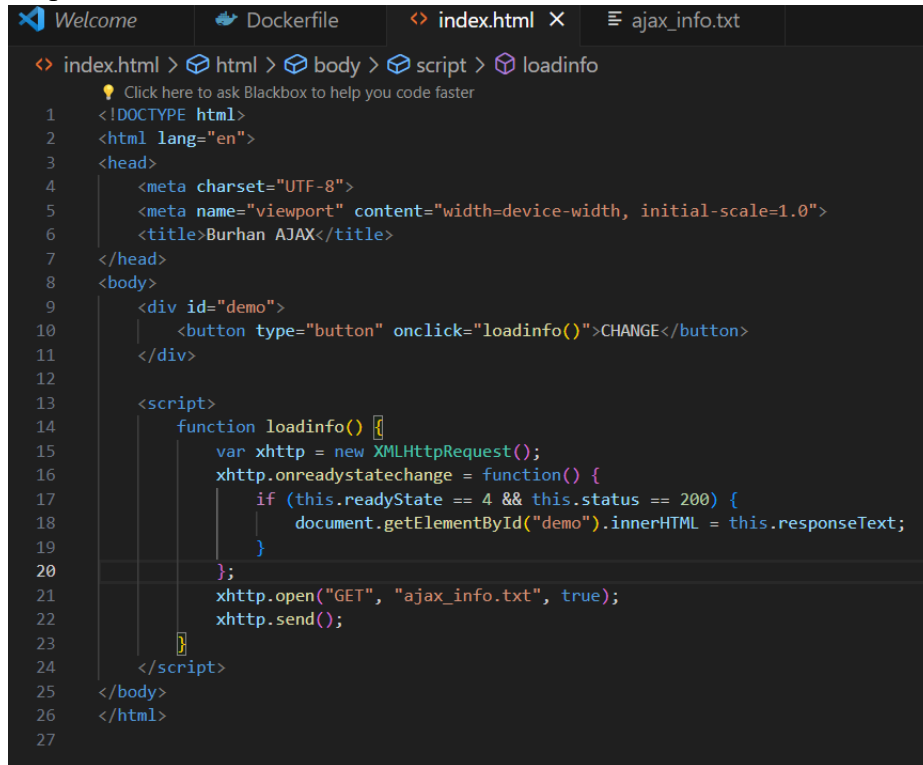
0% [4000:4000](#) 24 minutes ago



Showing 0 items

Practical 7 – Working with ajax

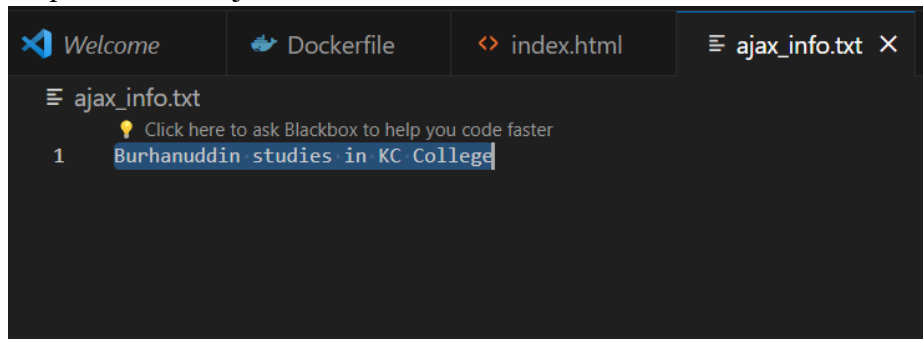
Step 1 - Create Index.html



The screenshot shows the VS Code editor with the 'index.html' file open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Burhan AJAX</title>
7 </head>
8 <body>
9   <div id="demo">
10    <button type="button" onclick="loadinfo()">CHANGE</button>
11  </div>
12
13  <script>
14    function loadinfo() {
15      var xhttp = new XMLHttpRequest();
16      xhttp.onreadystatechange = function() {
17        if (this.readyState == 4 && this.status == 200) {
18          document.getElementById("demo").innerHTML = this.responseText;
19        }
20      };
21      xhttp.open("GET", "ajax_info.txt", true);
22      xhttp.send();
23    }
24  </script>
25 </body>
26 </html>
27
```

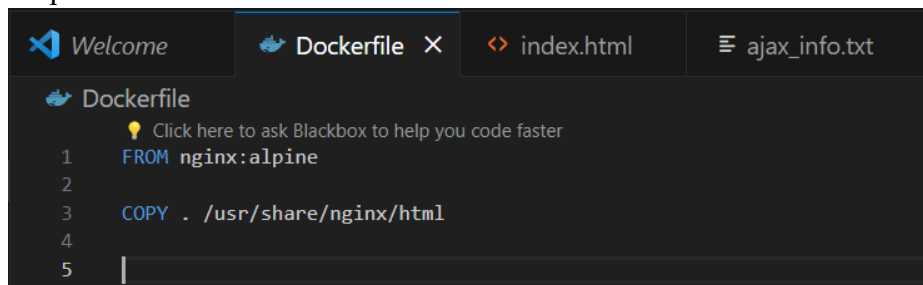
Step 2 – Create Ajax info text file



The screenshot shows the VS Code editor with the 'ajax_info.txt' file open. The content is as follows:

```
1 Burhanuddin studies in KC College
```

Step 3 – Create Docker file



The screenshot shows the VS Code editor with the 'Dockerfile' file open. The content is as follows:

```
1 FROM nginx:alpine
2
3 COPY . /usr/share/nginx/html
4
5
```

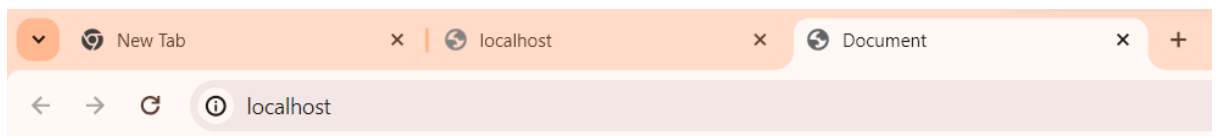
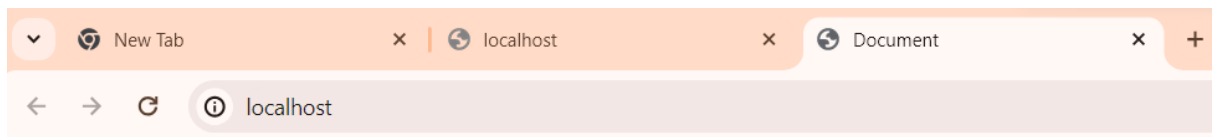
Step 4 - Docker build

```
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
```

Step 5 - Docker run

```
PS C:\BSCIT\TY\sem6\devOps\docker\ajax> docker run -d -p 80:80 burhanajax
a2f7d1712c5f7662da4bddd2650d76993b15b5e28d3e261cf95f8b907127e46
PS C:\BSCIT\TY\sem6\devOps\docker\ajax> █
```

Server



Burhanuddin studies in KC College

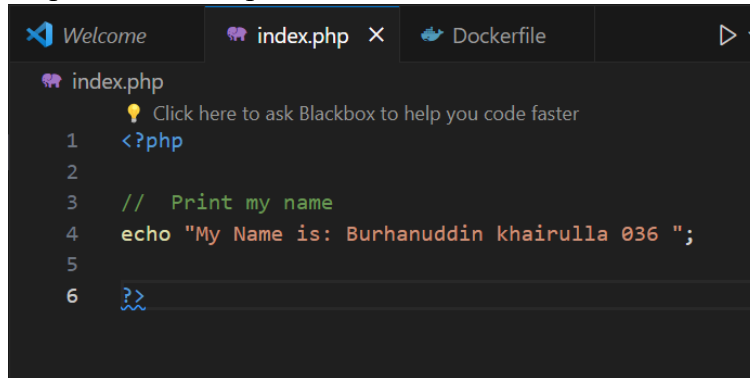
| | | | | | | | | |
|--|---|----------------------------|---------|----|-----------------------|----------------|--|--|
| | great_rubin a2f7d1712c5f | burhanajax | Running | 0% | 80:80 | 11 minutes ago | | |
|--|---|----------------------------|---------|----|-----------------------|----------------|--|--|

Showing 9 items

HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

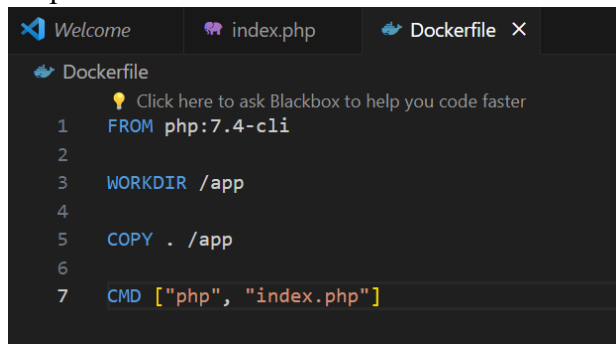
Practical 8 – Working with PHP

Step 1 – Create Php file



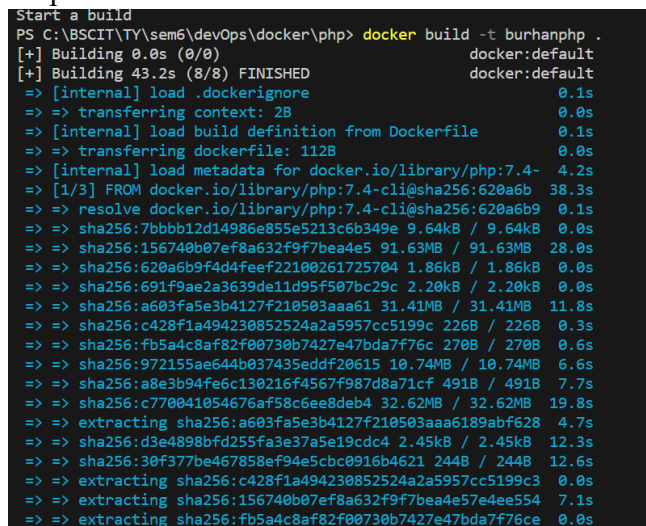
```
index.php
1  <?php
2
3  // Print my name
4  echo "My Name is: Burhanuddin khairulla 036 ";
5
6  ?>
```

Step 2 – Create Docker file



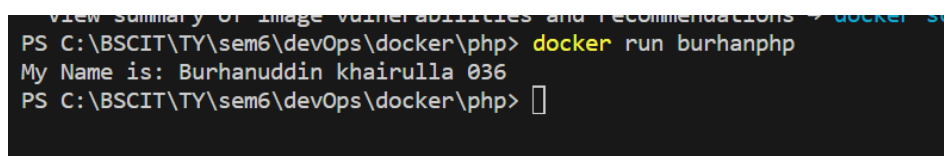
```
Dockerfile
1  FROM php:7.4-cli
2
3  WORKDIR /app
4
5  COPY . /app
6
7  CMD ["php", "index.php"]
```

Step 3 – Run Docker build



```
Start a build
PS C:\BSCIT\TY\sem6\devOps\docker\php> docker build -t burhanphp .
[+] Building 0.0s (0/0)                                docker:default
[+] Building 43.2s (8/8) FINISHED                      docker:default
=> [internal] load .dockerignore                        0.1s
=> => transferring context: 2B                          0.0s
=> [internal] load build definition from Dockerfile     0.1s
=> => transferring dockerfile: 112B                     0.0s
=> [internal] load metadata for docker.io/library/php:7.4- 4.2s
=> [1/3] FROM docker.io/library/php:7.4-cli@sha256:620a6b 38.3s
=> => resolve docker.io/library/php:7.4-cli@sha256:620a6b9 0.1s
=> => sha256:7bbb12d14986e855e5213c6b349e 9.64kB / 9.64kB 0.0s
=> => sha256:156740b07ef8a632f9f7bea4e5 91.63MB / 91.63MB 28.0s
=> => sha256:620a6b9f4d4feef22100261725704 1.86kB / 1.86kB 0.0s
=> => sha256:691f9ae2a3639de11d95f507bc29c 2.20kB / 2.20kB 0.0s
=> => sha256:a603fa5e3b4127f210503aaa61 31.41MB / 31.41MB 11.8s
=> => sha256:c428f1a494230852524a2a5957cc5199c 226B / 226B 0.3s
=> => sha256:fb5a4c8af82f00730b7427e47bda7f76c 270B / 270B 0.6s
=> => sha256:972155ae644b037435eddf20615 10.74MB / 10.74MB 6.6s
=> => sha256:a8e3b94fe6c130216f4567f987d8a71cf 491B / 491B 7.7s
=> => sha256:c770041054676af58c6ee8deb4 32.62MB / 32.62MB 19.8s
=> => extracting sha256:a603fa5e3b4127f210503aaa6189abf628 4.7s
=> => sha256:d3e4898bfd255fa3e37a5e19cdc4 2.45kB / 2.45kB 12.3s
=> => sha256:30f377be467858ef94e5cbc0916b4621 244B / 244B 12.6s
=> => extracting sha256:c428f1a494230852524a2a5957cc5199c3 0.0s
=> => extracting sha256:156740b07ef8a632f9f7bea4e57e4ee554 7.1s
=> => extracting sha256:fb5a4c8af82f00730b7427e47bda7f76ce 0.0s
```

Step 4 – Docker run



```
view summary of image vulnerabilities and recommendations > docker sc
PS C:\BSCIT\TY\sem6\devOps\docker\php> docker run burhanphp
My Name is: Burhanuddin khairulla 036
PS C:\BSCIT\TY\sem6\devOps\docker\php> □
```

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! kubectll is now configured to use "minikube" c
PS C:\Users\kburh> kubectll 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> kubectll 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> kubectll 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 'kubectll cluster-info dump'.
```

Kubectl create pod

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

Kubectl get pods

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

Kubectl delete pod

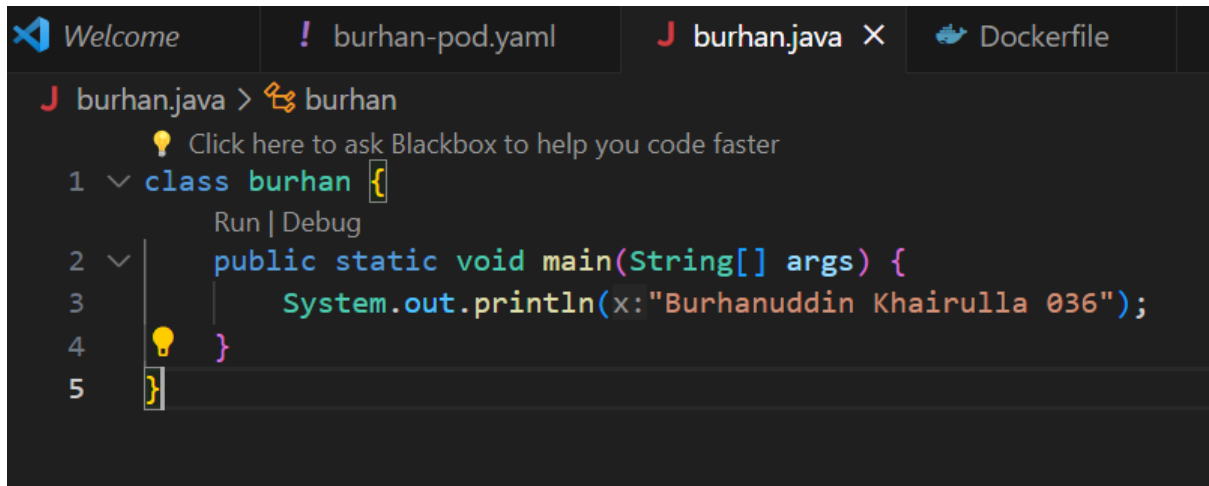
```
C:\BSCIT\TY\sem6\devOps\practice\kubernetes>kubectll delete pod burhan-pod
pod "burhan-pod" deleted
```

Kubectl logs pod

```
C:\BSCIT\TY\sem6\devOps\practice\kubernetes>kubectll 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



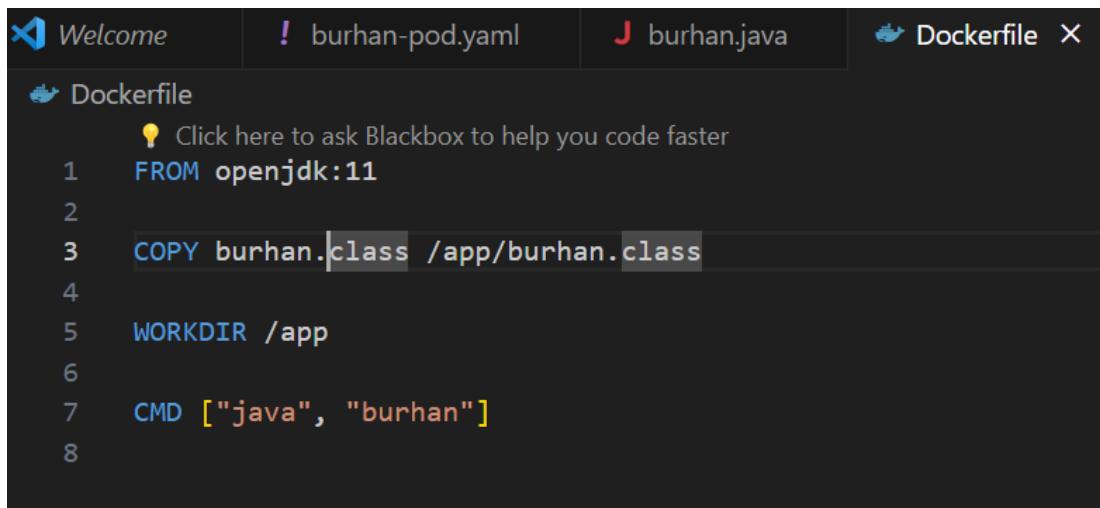
The screenshot shows the Visual Studio Code editor with the file `burhan.java` open. The code defines a class `burhan` with a `main` method that prints a string. The code is as follows:

```
1 class burhan {  
2     public static void main(String[] args) {  
3         System.out.println(x:"Burhanuddin Khairulla 036");  
4     }  
5 }
```

Step 2 – Compile the java application

```
C:\BSCIT\TY\sem6\devOps\practice\kubenetes>javac burhan.java
```

Step 3 – Create a dockerfile



The screenshot shows the Visual Studio Code editor with the `Dockerfile` open. The Dockerfile contains the following instructions:

```
1 FROM openjdk:11  
2  
3 COPY burhan.class /app/burhan.class  
4  
5 WORKDIR /app  
6  
7 CMD ["java", "burhan"]  
8
```


Step 4 – Build docker image

```
C:\BSCIT\TY\sem6\devOps\practice\kubernetes>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@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9c9cd7c21ab
=> => resolve docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9c9cd7c21ab
=> => sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9c9cd7c21ab 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
=> => sha256:47a932d998b743b9b0bce55aa8ede77de94a6a183c8a67dec9d5e3b8ce0faa7 6.26kB / 6.26kB
=> => sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbcb3a264c218c2ec7bca716e6 54.58MB / 54.58MB
=> => sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a1049710ebc4692de3dd5 5.42MB / 5.42MB
=> => sha256:66223a710990a0ae7162aead80417d30303afa3f24aaf57aa30348725e2230b 213B / 213B
=> => sha256:db38d58ec8ab4111b072f6700f978a51985acd252aabce3be377f25162e68301 202.07MB / 202.07MB
=> => extracting sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452
=> => extracting sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165
=> => extracting sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a
=> => extracting sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbcb3a264c218c2ec7bca716e6
=> => extracting sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a1049710ebc4692de3dd5
=> => extracting sha256:66223a710990a0ae7162aead80417d30303afa3f24aaf57aa30348725e2230b
=> => extracting sha256:db38d58ec8ab4111b072f6700f978a51985acd252aabce3be377f25162e68301
=> [internal] load build context
```

Step 5 – Run the docker container

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

Step 6 – Create a deployment file:

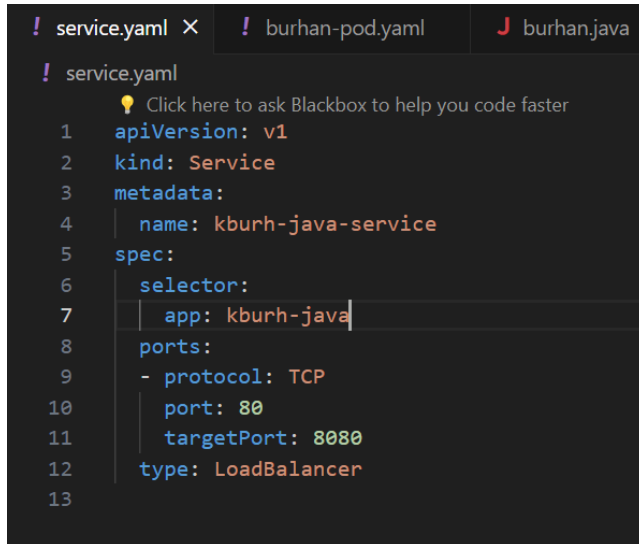
```
! burhan-pod.yaml  J burhan.java  Dockerfile  ! deploy-burhan.yaml X
! deploy-burhan.yaml
  Click here to ask Blackbox to help you code faster
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: kburh-java-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: kburh-java
10   template:
11     metadata:
12       labels:
13         app: kburh-java
14   spec:
15     containers:
16       - name: kburh-java-container
17         image: burhankhair/kburh-java
18
```

HSNC University , Mumbai
Kishinchand Chellaram College, Mumbai – 20.

Step 7 – Apply the Deployment to kubertenes

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

Step 8 – Expose the Deployment as a Service:



```
! service.yaml X ! burhan-pod.yaml J burhan.java
! service.yaml
  Click here to ask Blackbox to help you code faster
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: kburh-java-service
5  spec:
6    selector:
7      app: kburh-java
8    ports:
9      - protocol: TCP
10        port: 80
11        targetPort: 8080
12    type: LoadBalancer
13
```

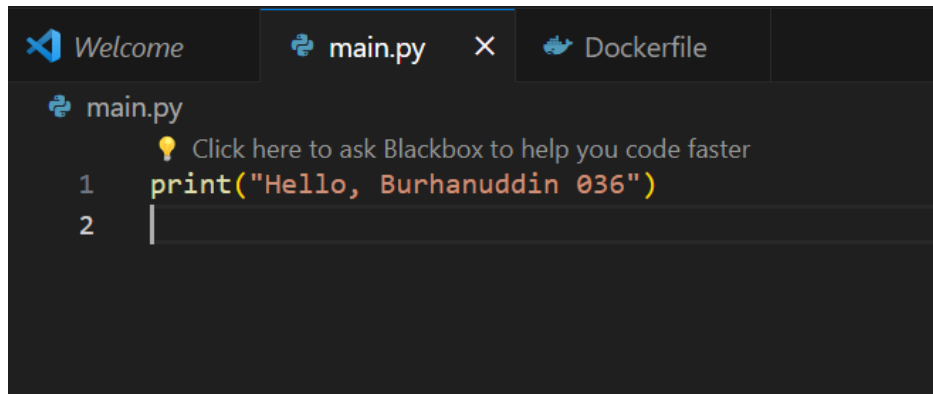
```
C:\BSCIT\TY\sem6\devOps\practice\kubernetes>kubectl apply -f service.yaml
service/kburh-java-service created
```

Step 9 – kubectl get svc kburh-java-service

```
C:\BSCIT\TY\sem6\devOps\practice\kubernetes>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
```

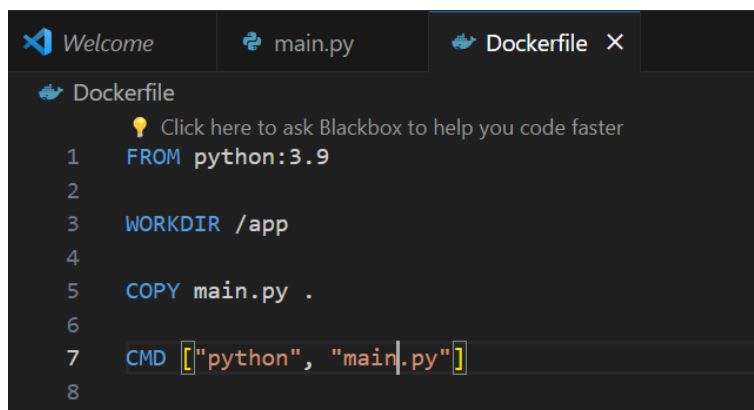
Practical 11 – Working with python

Step 1 – Create Python file



The screenshot shows the Visual Studio Code editor interface. At the top, there are three tabs: 'Welcome', 'main.py', and 'Dockerfile'. The 'main.py' tab is active. Below the tabs, the editor displays the content of 'main.py'. It starts with a lightbulb icon and the text 'Click here to ask Blackbox to help you code faster'. Below this, there are two lines of code: line 1 is `print("Hello, Burhanuddin 036")` and line 2 is empty. The cursor is positioned at the end of line 2.

Step 2 Create Docker file



The screenshot shows the Visual Studio Code editor interface. At the top, there are three tabs: 'Welcome', 'main.py', and 'Dockerfile'. The 'Dockerfile' tab is active. Below the tabs, the editor displays the content of 'Dockerfile'. It starts with a lightbulb icon and the text 'Click here to ask Blackbox to help you code faster'. Below this, there are eight lines of code: line 1 is `FROM python:3.9`, line 2 is empty, line 3 is `WORKDIR /app`, line 4 is empty, line 5 is `COPY main.py .`, line 6 is empty, line 7 is `CMD ["python", "main.py"]`, and line 8 is empty. The cursor is positioned at the end of line 7.

HSNC University , Mumbai

Kishinchand Chellaram College, Mumbai – 20.

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
=> => sha256:530d4ba717be787c0e2d011aa107edac6d721f8c06fe6d44708d4aa5e9bc5ec9 2.01kB / 2.01kB
=> => sha256:e301b6ca47814a93ddd0420cfffbe960bc5363a43d3e8644e53d547ffce16a7a5 7.33kB / 7.33kB
=> => sha256:7bb465c2914923b08ae03b7fc67b92a1ef9b09c4c1eb9d6711b22ee6bbb46a00 49.55MB / 49.55MB
=> => sha256:2b9b41aaa3c52ab268b47da303015b94ced04a1eb02e58860e58b283404974f4 24.05MB / 24.05MB
=> => sha256:49b40be4436eff6fe463f6977159dc727df37cabe65ade75c75c1caa3cb0a234 64.14MB / 64.14MB
=> => sha256:c558fac597f8ecbb7a66712e14912ce1d83b23a92ca8b6ff14eef209ab01aff2 211.12MB / 211.12MB
=> => extracting sha256:7bb465c2914923b08ae03b7fc67b92a1ef9b09c4c1eb9d6711b22ee6bbb46a00 6.7s
=> => sha256:11402150a57e537c64dc69a28bba37f13acd5d50d8788894398a7b774786e7d 6.39MB / 6.39MB
=> => sha256:d662c27d7e5762cd7807a257f6ec30067c714fdd9a9390836991830c0fbbb47 15.82MB / 15.82MB
=> => extracting sha256:2b9b41aaa3c52ab268b47da303015b94ced04a1eb02e58860e58b283404974f4 1.7s
=> => sha256:26ff257bfcf12b2ad8a2527c7a42179ff6febdddde7064dec546ec0869a4627 242B / 242B
=> => sha256:5a8f6c8c0bd20abb177c033e0a9dc88085bb9ae78d41671925372d005c7ca048 2.85MB / 2.85MB
=> => extracting sha256:49b40be4436eff6fe463f6977159dc727df37cabe65ade75c75c1caa3cb0a234 7.6s
=> => extracting sha256:c558fac597f8ecbb7a66712e14912ce1d83b23a92ca8b6ff14eef209ab01aff2 23.5s
=> => extracting sha256:11402150a57e537c64dc69a28bba37f13acd5d50d8788894398a7b774786e7d 1.1s
=> => extracting sha256:d662c27d7e5762cd7807a257f6ec30067c714fdd9a9390836991830c0fbbb47 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 quick
PS C:\BSCIT\TY\sem6\devOps\docker\python> docker run burhan-python-app
Hello, Burhanuddin 036
PS C:\BSCIT\TY\sem6\devOps\docker\python> []
```