<html>

<body>

<h1> Exercise-3 Index file <h1>

<h2> Hello Charmy</h2>

</body>

</html>

**Exercise 1: Create your first Commit.**

**Task:**

Initialize your Repository: git init project Exercise-1

Create a file with name index.html

Write code in newly created file

Stage index.html

Commit with message ‘Initial Message.’

**Code**

mkdir Exercise-1

git init Exercise-1

cd Exercise-1

touch index.html

nano index.html

git status

git add index.html

git status

git commit -m "Initial Massage"

**Exercise 2: Commit only one file.**

**Task:**

Initialize your Repository: git init Exercise-2

Create a file with names index.html and about.html

Add code in both files.

Stage only one file.

Commit with message: ‘File x added.’

**Code**

mkdir Exercise-2

git init Exercise-2

cd Exercise-2

touch index.html about.html

nano index.html

nano About.html

git status

git add index.html

git status

git commit -m “Index file added”

**Exercise 3: Modify last Commit.**

**Task:**

Initialize your Repository: git init project Exercise-3

Create a file with name index.html

Write code in newly created file

Stage index.html

Commit with message ‘adding index.html.’

Add new about.html file with some code.

Now modify message of previous commit as ‘Adding index.html and

about.html ’and add about.html file in previous commit.

**Code**

mkdir Exercise-3

git init Exercise-3

cd Exercise-3

touch index.html

nano index.html

git status

git add index.html

git status

git commit -m “adding index.html"

touch about.html

nano about.html

git status

git commit --amend -m "Adding index.html and about.html"

git add about.html

git commit --amend --no-edit

**Exercise 4: Create git bignore file.**

**Task:**

Initialize your Repository: git init Exercise-4

Create a gitignore file that will ignore

all files with ‘exe’ extension

all files with ‘o’ extension

all files with ‘jar’ extension

the whole libraries directory

**Code**

Mkdir Exercise-4

Git init Exercide-4

Cd Exercise-4

Touch excel.exe file.O filejar.jar

Mkdirk Libreries

touch .gitignore

Git status

nano .gitignore

cat .gitignore

\*.exe

\*.jar

\*.o

Libreries/

Git status

**Exercise 5: Save your work with stash.**

**Task:**

Initialize your Repository: git init Exercise-5

Add some random files with code

Now use git stash command to save required files from above files.

(For this exercise don’t send repository just submit git commands and output of

git stash list)

**code**

Mkdir Exercise-5

Git init Exercide-5

Cd Exercise-5

touch text.txt web.html text1.txt

nano text.txt

nano text1.txt

nano web.html

git add text.txt

git commit -m “text file commited”

git add text1.txt

git commit -m “text1 file commited”

git add web.html

git commit -m “wen.html file commited”

nano text.txt (code was change)

nano web.html (code was change)

git stash -m “web.html edit”

git stash list

cat web.html

git stash apply stash@{0}

cat web.html

Text

Description automatically generated

**Exercise 6: Reset Previous Commit Keeping changes. (Soft Reset)**

**Task:**

Initialize your Repository: git init Exercise-6.

Do some series of commits.

Now use git reset command to reset one of the commit and keep

changes in staging area.

In answer also add output of git status command.

**Code**

Mkdir Exercise-6

Git init Exercide-6

Cd Exercise-6

Touch demo6.txt new\_demo6.txt

Nano demo6.txt

Git add demo6.txt

Git commit -m “one commit”

Nano demo6.txt

Git add demo6.txt

Git commit -m “second commit”

Nano demo6.txt

Git add demo6.txt

Git commit -m “third commit”

Nano demo6.txt

Git add demo6.txt

Git commit -m “fourth commit”

Git log

Cat demo6.txt

git reset --soft HEAD~2

cat demo6.txt

git log

**Text

Description automatically generated**

**Text

Description automatically generated**

**Exercise 7: Reset Previous Commit (Hard Reset).**

**Task:**

Initialize your Repository: git init Exercise-7.

Do some series of commits.

Use git reset command to reset one of the commit without keeping

changes.

**Code**

Mkdir Exercise-7

Git init Exercide-7

Cd Exercise-7

Touch demo\_test.txt new\_demo.txt

Nano demo\_test.txt

Git add demo\_test.txt

Git commit -m “1st commit”

Nano demo\_test.txt

Git add demo\_test.txt

Git commit -m “2nd commit”

Nano demo\_test.txt

Git add demo\_test.txt

Git commit -m “3rd commit”

Git log

git reset --hard 160e839020cdd4a0f8f3b71c256b4eb3684eeb49

car demo\_test.txt

git log

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Exercise 8: Merge branch (No Conflicts).**

**Task:**

Initialize your Repository: git init Exercise-8

Create index.html file add some code.

Make Commit.

Create new branch from master branch with name my-feature.

Add about-us.html file with code and commit changes in my-feature

branch.

Now checkout master branch.

Merge my-feature branch into master.

**Code**

Mkdir Exercise-8

Git init Exercide-8

Cd Exercise-8

Touch index.html

Nano index.html

Git add .

Git commit -m “index file added”

Git branch my-feature

Git checkout my-feature

Touch about-us.html

Nano about-us.html

Git add .

Git commit -m ”about-us file committed”

Git push –set-upstream origin my-feature

Git checkout main

Git merge my-feature

**Exercise 9: Merge branch (Conflicts).**

**Task:**

Initialize your Repository: git init Exercise-9

Create index.html file add some code.

Make Commit.

Create new branch from master branch with name my-feature.

Add about-us.html file with code

Make changes in index.html

Commit changes in my-feature branch.

Now checkout master branch.

Do some code changes in index.html

Commit changes.

Now merge my-feature branch into master.

You will get conflicts for index.html file. Resolve conflict by selecting both

changes for the same index.html file.

**Code**

Mkdir Exercise-9

Git init Exercide-9

Cd Exercise-9

Touch index.html

Nano index.html

Git add .

Git commit -m “index file added”

Git branch my-feature

Git checkout my-feature

Touch about-us.html

Nano about-us.html

Nano index.html

Git add .

Git commit -m ”index & about-us file committed”

Git push –set-upstream origin my-feature

Git checkout main

Nano index.html

Git add .

Git commit -m “index file updated”

Git merge my-feature

Nano index.html