Git:

Git was originally authored by Linus Torvalds in 2005 for development of the Linux Kernel, with other kernel developers contributing to its initial development.

- Git is a open source system.
- It is Distributed version Control System.
- It is used to handle Projects with efficiency and speedly.

GitHub:

GitHub, Inc. is a provider of Internet Hosting for software Development and version control using Git. It offers the Distributed version Control and source code management (SCM) functionality of Git, plus its own features.

- Provides Access control.
- Used for Bug tracking.
- Feature requests like task management.

GitBash:

Git Bash is an application for Microsoft Windows environments which provides an emulation layer for a Git command line experience.

Bash is an acronym for Bourne Again Shell. A shell is a terminal application used to interface with an operating system through written commands.

Bash is a popular default shell on Linux and macOS.

Important Commands of Git:

Git init:

Command : git init {Repository name}

We use this command to initialize a particular repository so, we can use our local folder.

Will create a .git directory.

Git add:

Command : git add {file name}

It will add the file in git repository in the staged area.

Stagged area can be tracked by Git.

To add all the files :

Command to be used is : **git add .**

Git Commit:

Command: git commit –m “message”

This command will record the file in version history.

All files in directory will be saved in Git File System.

Git Status:

Command : git status

It will show the addition status of the file.

Show all the files that have to be committed.

Git Remote:

Command: git remote add “remote name” “url”

HERE “url” we get from the new repository created.

After this we can operate te remote commands in our repository that we have created.

Git Push :

Command : git push “remote name” “Branch name”

By this we can push the file to the github . or the files can be synced with remote repository.

Git Pull :

Command: git pull “remote name” master

Here pull is our bydefault branch.

This first run “git fetch”. This will download the content from the remote repository.

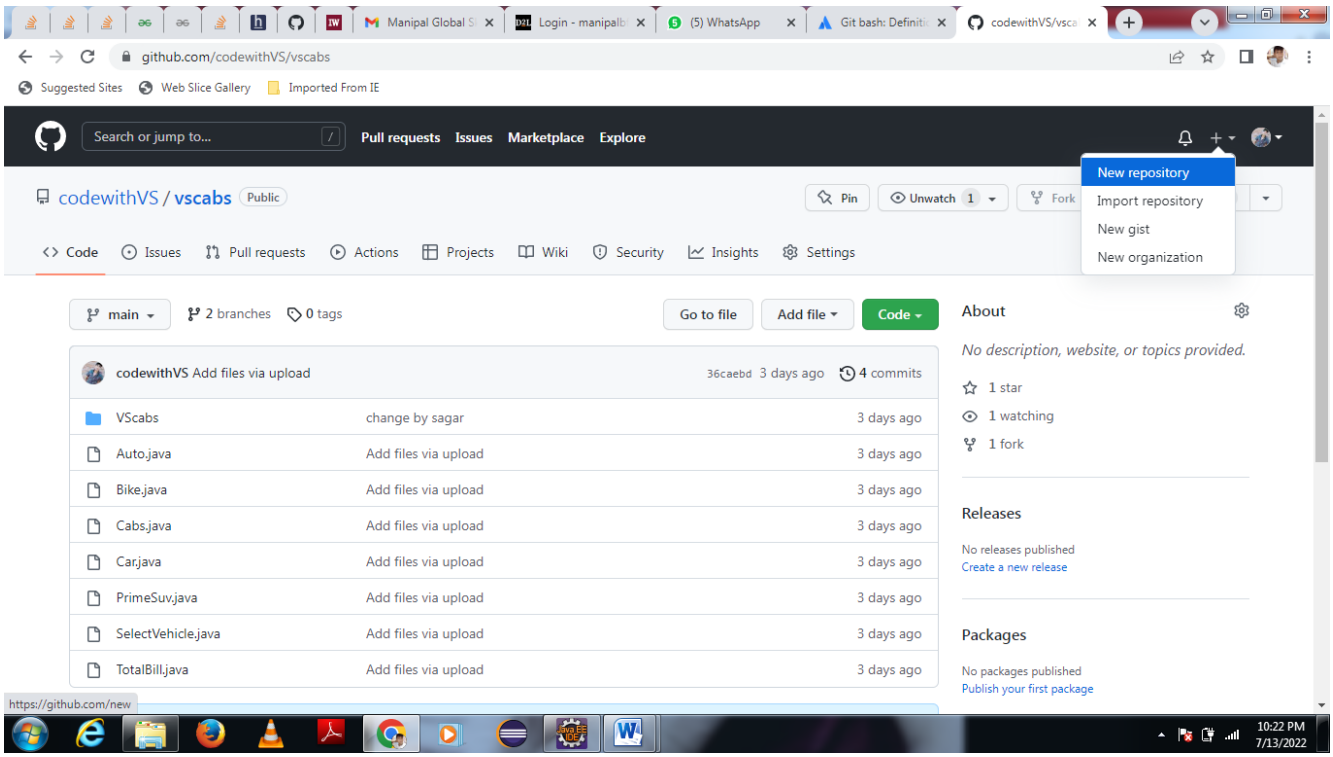
And then it update the local repository and

Thn it matches the content of the file.

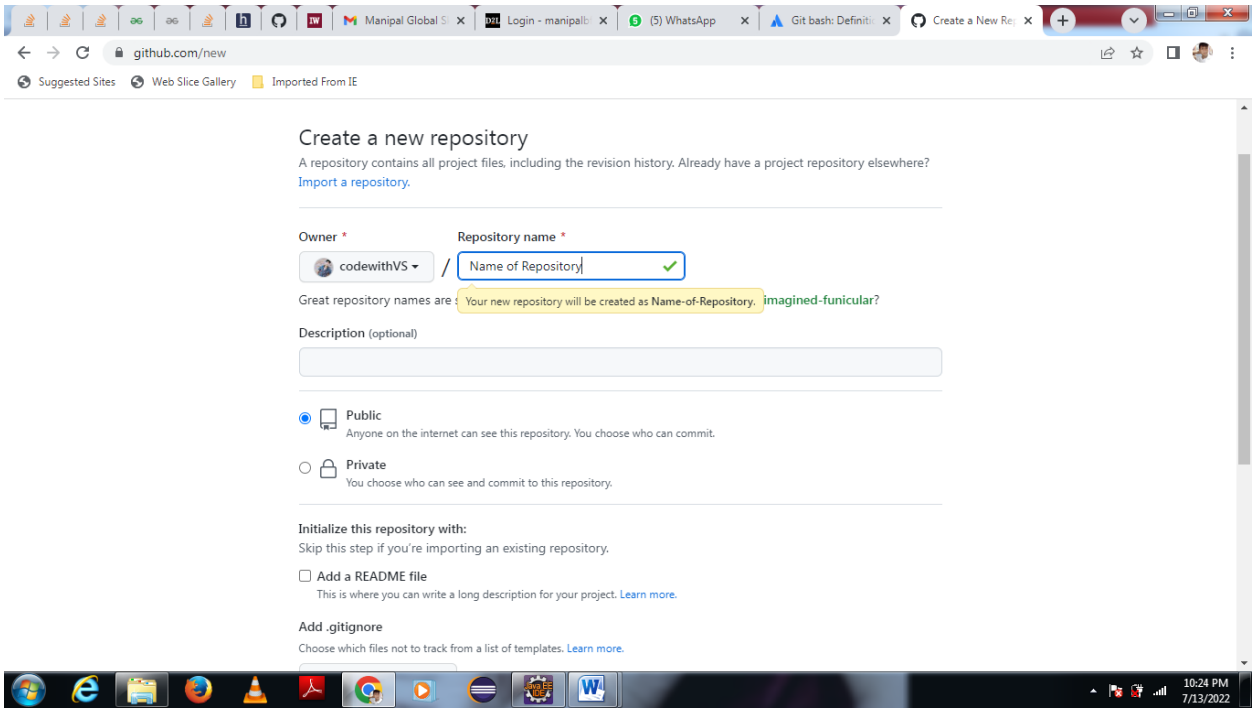
.

Creating a New Repository Using Git Gui :

Click on New repository :



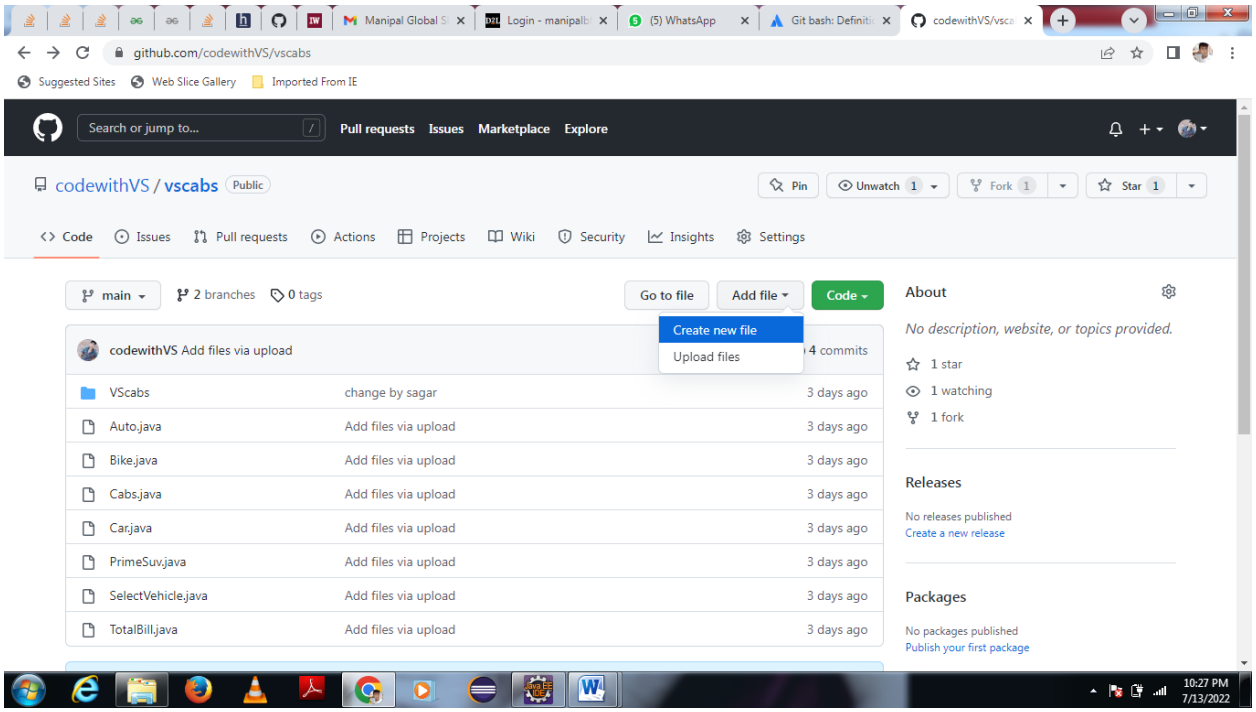
Give name to Repository :



Discription : In this the goal or the purpose behind that repo can be explained.

Click on Click Repository.

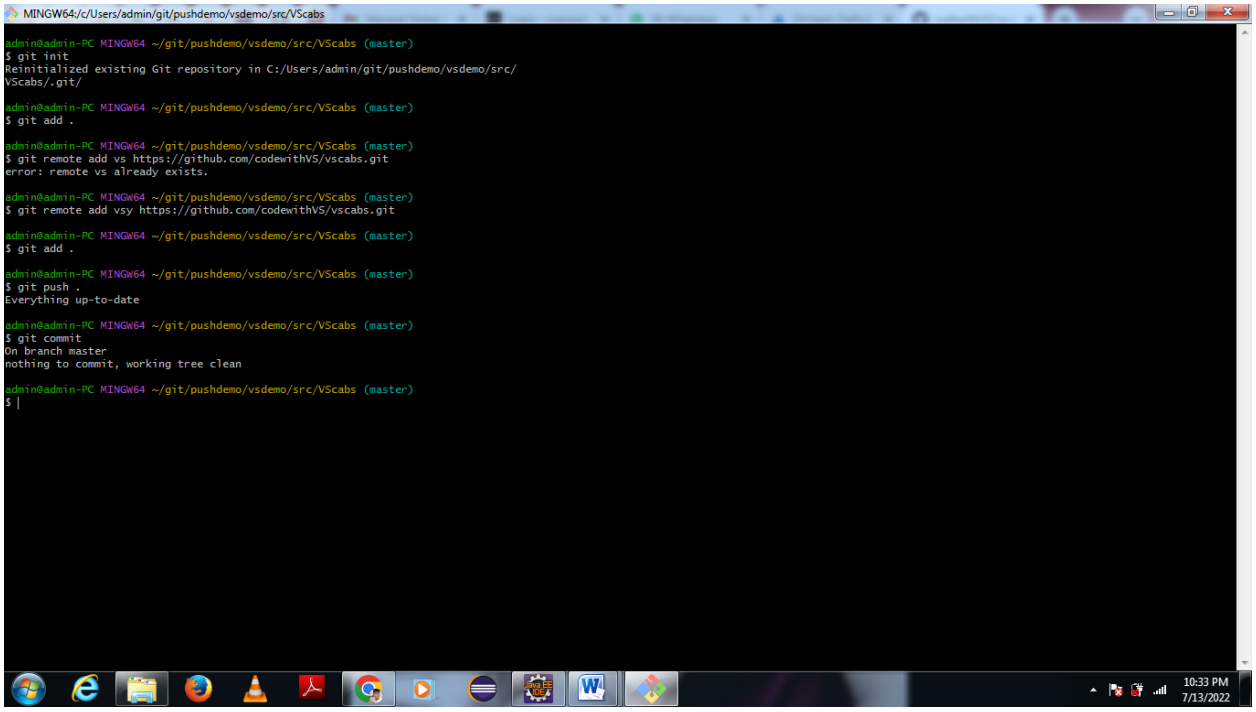
Create a File :



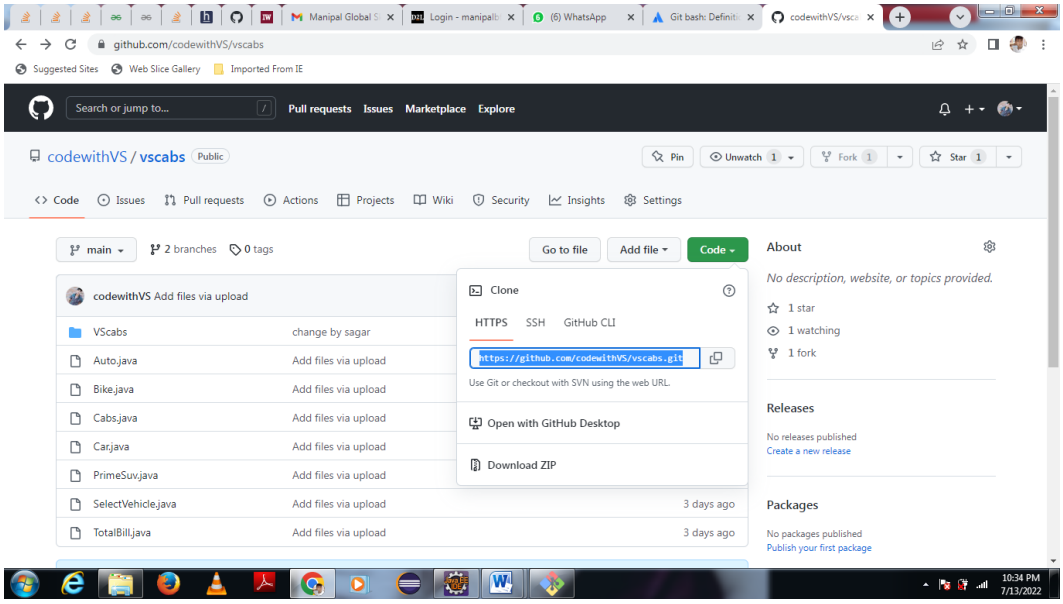
By using Create a file we can create a file or we can type the content of that particular file there.

Upload File : Using upload file we can upload a file on git from our local storage. Or we can browse a file and then upload on git.

USING Git Bash Commands :



The url is taken from git UI



VScabs :

Abstract :

I have used factory design pattern to create a project named VScabs.

In which we are making a cab system . and using the programme we are calculating the total cost of the journey.

In this we have provided a option to select a vehicle using class Vehicle.java . where we can Select the Vehicle.

An another class cabs.java which is the main class and where we are taking inputs from the user at run time . and finally calling the function to process Bill.

A class TotalBill.java in which we have calculated the total bill of the journey using the type of vehicle selected and getting rate using getRate().and according to the distance travelled.

Another vehicle classes in which we have defined the rate for that particular vehicle per km.

These classes we used are : Car,java , Bike.java , Auto.java , Primesuv.java

Class Cabs.java :

```
package VScabs;

import java.io.*;
public class Cabs {
	public static void main(String[] args) throws IOException {
		// TODO Auto-generated method stub
		SelectVehicle selfac = new SelectVehicle();
		BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
		System.out.println("Welcome to VScabs : ");
		System.out.println();
		System.out.println("Enter the Vehicle You want : ");
		String vtype = br.readLine();
		System.out.println("Enter the total distance km");
		int km = Integer.parseInt(br.readLine());
		TotalBill veh=selfac.getVeh(vtype);
		System.out.println("Bill amount for "+vtype+" to ride "+km
+ " km : ");
		veh.getRate();
		veh.processBill(km);
	}

}
```

Class SelectVehicle.java :

```
package VScabs;

public class SelectVehicle {
	public TotalBill getVeh(String PlanType) {
		if (PlanType==null) {
			return null;
		}
		if (PlanType .equalsIgnoreCase("Bike")) {
			return new Bike();
		}
		if (PlanType .equalsIgnoreCase("Auto")) {
			return new Auto();
		}
		if (PlanType .equalsIgnoreCase("car")) {
			return new Car();
		}
		if (PlanType .equalsIgnoreCase("PrimeSuv")) {
			return new PrimeSuv();
		}
		return null;
	}
```

Class TotalBill.java :

```
package VScabs;

abstract class TotalBill {
	protected double rate;
	abstract void getRate();
	public void processBill(int km) {
		System.out.println("Cost of total journey : "+(km*rate) + "
INR");
	}
}
```

Class Bike.java :

```
package VScabs;

public class Bike extends TotalBill{
	public void getRate() {
		rate=5;
	}
}
```

Class Car.java :

```
package VScabs;

public class Car extends TotalBill{
	public void getRate() {
		rate=15;
	}
}
```

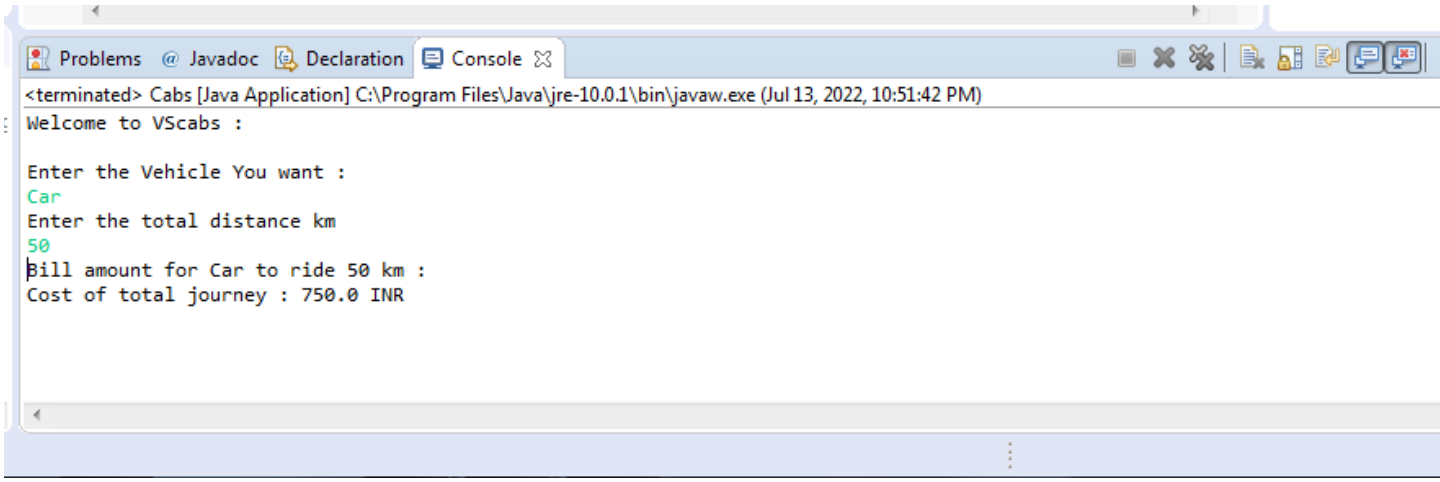
Class Auto.java :

```
package VScabs;

public class Auto extends TotalBill{
	public void getRate() {
		rate=8;
	}
}
```

Class PrimeSuv.java :

```
package VScabs;

public class PrimeSuv extends TotalBill{
	public void getRate() {
		rate=20;
	}
}
```

Output :


Problems
Javadoc
Declaration
Console
<terminated> Cabs [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe (Jul 13, 2022, 10:51:42 PM)
Welcome to VScabs :
Enter the Vehicle You want :
Car
Enter the total distance km
50
Bill amount for Car to ride 50 km :
Cost of total journey : 750.0 INR

Conclusion :

We have made a project using factory Design pattern named VScabs.

Where we calculated the total journey cost.

Then using git GUI we created a repository vscabs.

And using the multiple many git bash command we pushed the project on GitHub. That can be viewed by anyone and contribution can be done.

That project source code can be viewed at https://github.com/codewithVS/vscabs