GIT:-

Git was initially created by Linus Torvalds in 2005 for the Linux Kernel development, with assistance from other kernel developers.

The system Git is open source. Distributed version control system is what it is. It is employed to quickly and efficiently manage projects.

Github:-

The company GitHub, Inc. offers web hosting services for Git version control and software development. It provides its own features in addition to Git's Distributed version Control and source code management (SCM) capabilities.

Some important functions:

Git branch:-

In the realm of git, branches are crucial. Multiple developers can collaborate on the same project simultaneously by using branches. The git branch command can be used to create, list, and remove branches.

Commands :-
Create branch
"*Git branch <name>*"
List branch
"*Git branch*"
Deleting branch
"*Git branch -d <name>*"

Git checkout:-

One of the most used Git commands is this one. You must switch to a branch before you may begin working there. Git checkout is mostly used to transition between branches. Additionally, it can be used to check out files and commit changes.

Commands:-
“Git checkout <branchname>”

Additionally, we used -b to create and swap branches here.

Git push:

To upload content from a local repository to a remote repository, use the git push command. Commits are sent from your local repository to a remote repository via pushing. Pushing is the opposite of git fetch in that it exports commits to remote branches as opposed to fetching, which imports commits to local branches.

Commands:-
“Git puch <origin> <branch>”

Git pull:

Using the git pull command, you may quickly update your local repository with the content you just downloaded from a remote repository. In Git-based collaboration work processes, merging remote upstream changes into your local repository is a typical task.

Commands:
“Git pull <origin>”

Git Merge:-

The git merge command enables you to combine the several development lines produced by git branch into a single branch. The instructions listed below merge into the current branch, so take note of that.

Commands:
“Git merge <branch>”

Git clone:-

The command "git clone" is used to download current source code from a remote repository . Git clone, then, essentially creates an exact copy of the most recent version of a project in a repository and saves it to your computer.\

Commands:
“Git clone <url>”

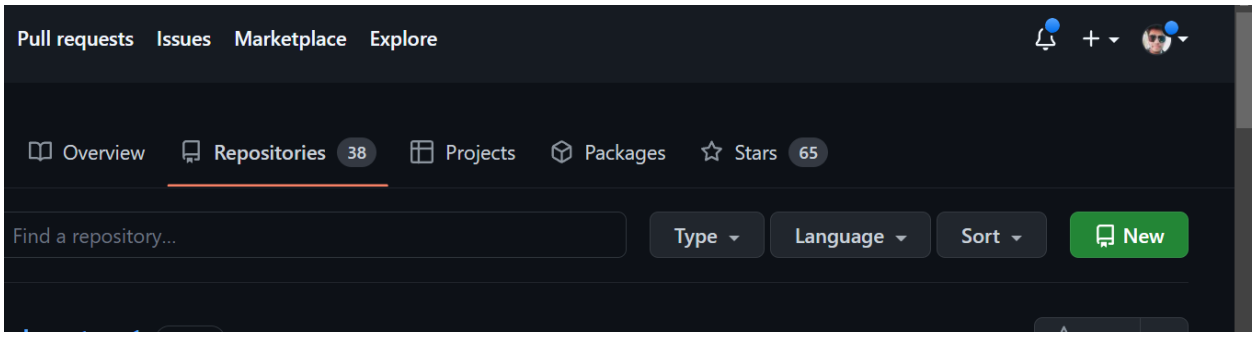
Git diff:-

Git uses the diff command to keep track of the differences between changes made to a file. Git is a version control system, thus keeping track of changes is essential. The diff command compares two inputs and displays their differences. These inputs don't have to be limited to files.

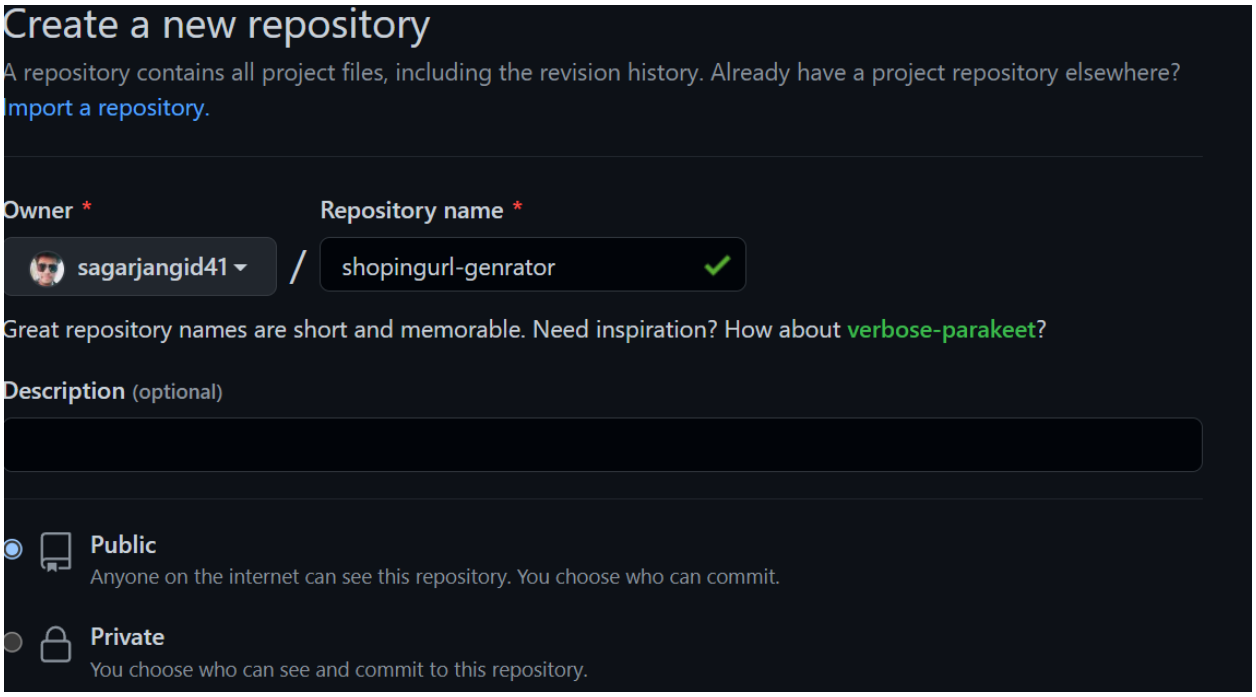
Commands:
“Git diff <id1> <id2>”

Creating a New Repository Using Git Gui :

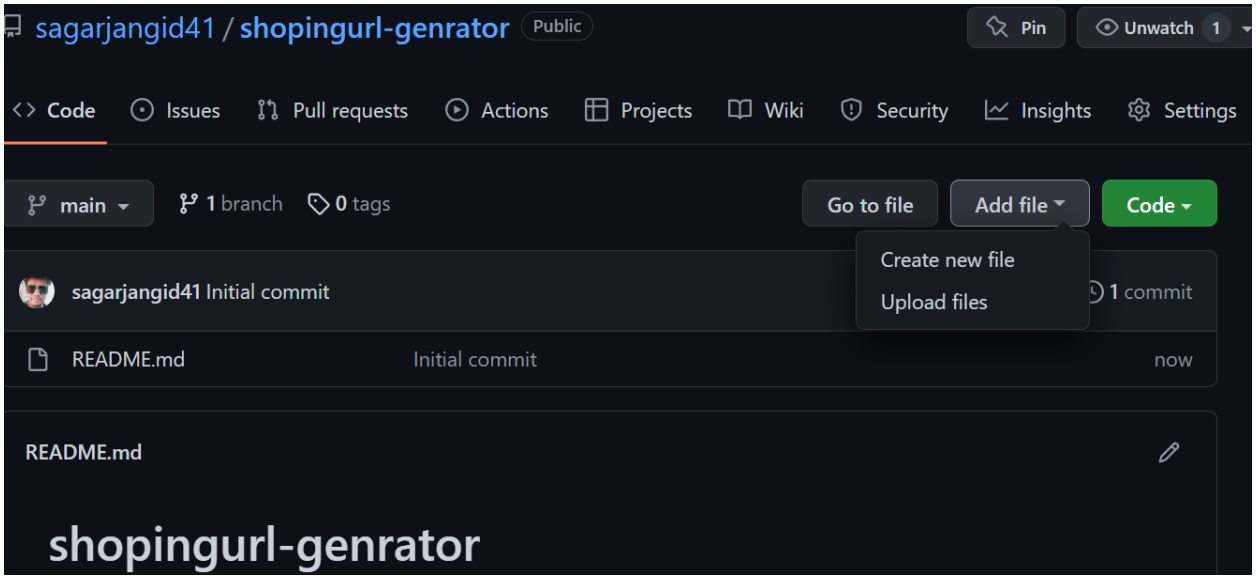
Click On New :-



Give Repository Name and Description:-



Create New File or Upload File :-



SWS Shopping Website Url Generator :-

Abstract :

A project called sws url generator that I created makes use of the factory design pattern.

For each page, we generate a url. This is useful for changing links from one to another.

Here, you may find sites for sports, clothing, electronics, and groceries. This software uses little memory thanks to the factory design approach.

Here, you must enter the item's name and category for the link to be generated correctly.

These classes we used are : sport.java, clothes.java, electronics.java, groceries.java

Create Url Interface:-

Ss.java

```
package linklist;
public interface ss {
   void notifyUser();
}
```

Classes for different category :-

Sport.java

```
package linklist;
public class sport implements ss {
        String name;
        sport(String name1){
         name=name1
   }
   public void notifyUser()
   {
      // TODO Auto-generated method stub
      System.out.println("https://sws/sport/"+name);
   }
}
```

Cloths.java

package linklist;

```
public class clothes implements ss {
        String name;
        clothes(String name1){
         name=name1
   }
   public void notifyUser()
   {
      // TODO Auto-generated method stub
      System.out.println("https://sws/clothes/"+name);
   }
}
```

Electronics.java

```
package linklist;
public class electronics implements ss {
        String name;
        electronics(String name1){
         name=name1 ;
   }
   public void notifyUser()
```

```
    {
        // TODO Auto-generated method stub
        System.out.println("https://sws/electronics/"+name);
    }
}
```

Grocery.java

```
package linklist;
public class grocery implements ss {

                String name;
                grocery(String name1){
                 name=name1 ;

            }
            public void notifyUser()
            {
                // TODO Auto-generated method stub
                System.out.println("https://sws/grocery/"+name);
            }
        }
```

Integrate category in factory class

Factory.java

```
package linklist;
public class factory {
    public  ss create(String type,String name)
    {
        if (type == null || type.isEmpty())
            return null;
        switch (type) {
        case "sport":
            return new sport(name);
        case "clothes":
            return new clothes(name);
        case "electronics":
            return new electronics(name);
        case "grocery":
            return new grocery(name);

        default:
            throw new IllegalArgumentException("Unknown selections "+name);
        }
```

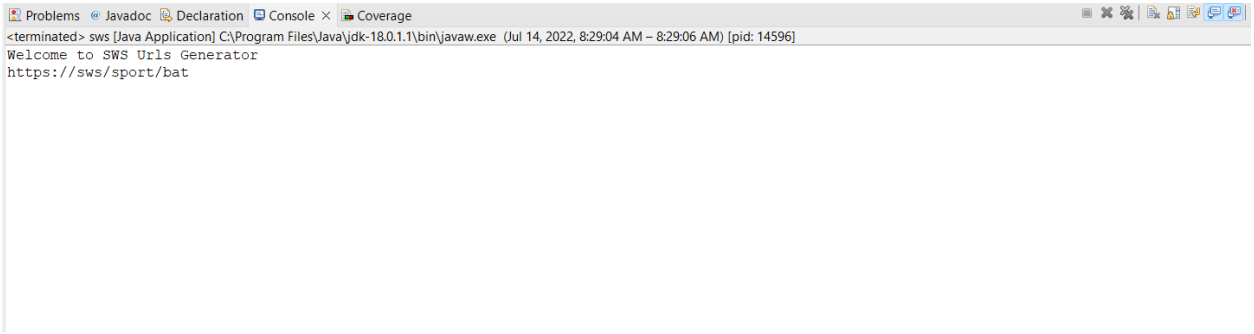
```
    }
}
```

Main SWS class:-

Sws.java

```
package linklist;
public class sws {
    public static void main(String[] args)
    {
        System.out.println("Welcome to SWS Urls Generator ");
        factory notificationFactory = new factory();
        ss notification = notificationFactory.create("sport","bat");
        notification.notifyUser();
    }
}
```

Output:-



Conclusion:-

We have made a project using factory Design pattern named SWS url Generator.

Where we generate shopping page urls.

Then using git GUI we created a repository shopping url-generator.

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.