

DEPARTMENT OF COMPUTER SCIENCE
INSTITUTE OF MANAGEMENT AND RESEARCH, JALGAON

Name Kawale Roshani Vasant
Expt. Title Set up a GitHub account & send an email message to friend telling to your friend about the account
Class EXMCA Batch B2 Performed on _____
Roll No. 48 Expt. No. 48 1 Submitted on _____
Remarks _____ Returned on _____

* What is GitHub?

GitHub is a Git repository hosting service. GitHub is an American company. GitHub also facilitates with many of its features, such as access control and collaboration. It provides a web-based graphical interface.

It hosts source code of your project the form of different programming languages and keeps track of the various changes made by programmers. It offers both distributed version control and source code management (scm) functionality of Git.

* Features of GitHub:-

GitHub is a place where programmers and designers work together. They collaborate, contribute and fix bugs together.

Some of its significant features are as follows:

- 1) Collaboration
- 2) Integrated issue and bug tracking.
- 3) Graphical representation of branches.
- 4) Git repositories hosting.
- 5) Project Management
- 6) Team Management
- 7) Code hosting
- 8) Track and design tasks.
- 9) conversations
- 10) Wikisc

* Benefits of GitHub:-

The key benefits of GitHub are as follows:-

- 1) It is easy to contribute to open-source projects via GitHub.
 - 2) It helps to create an excellent document.
 - 3) You can attract recruiter by showing of your work.
- If you have a profile on GitHub, you will have a

Incomplete for :

- 1) Algorithm
- 2) Flow Chart
- 3) Programme Listing
- 4) Results
- 5) Comments

higher chance of being recruited.

- 4) It allows your work to get out there in front of the public.
- 5) You can track changes in your code across versions.

* Advantages of GitHub:-

- 1) Similar to other online report services, GitHub comes with a big benefit to have the content inputted in such a format without having to know another system.
- 2) Flavoured markdown is another benefit of using GitHub.
- 3) Collaboration is another benefit of using GitHub. For people who do not remain in the same physical location.
- 4) You do not need to connect with your company's VPN as it is easier to dump with the collaboration feature of GitHub.

* Disadvantages of GitHub:-

- 1) GitHub does not come with private repositories. This is one of the disadvantages.
- 2) Some employers and clients can only allow code on their private secure internal git as far as the matter of policy is concerned.
- 3) Pricing is another potential drawback of GitHub.
- 4) Some of the features and online repositories happen to be locked behind a SaaS paywall.

Roll No:-48

Name:- Kawale Roshani Vasant

Practical NO:-1

Set up a GitHub account and send an email message to friend telling to your friend about the account.

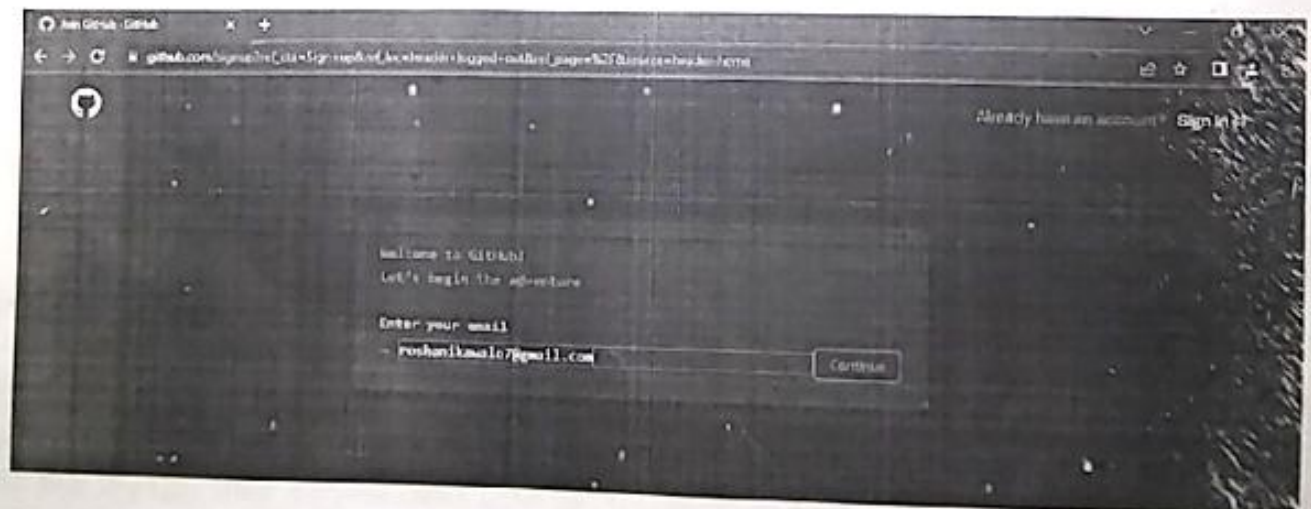
Create a GitHub account

If you don't already have a GitHub account, here's how to create one.

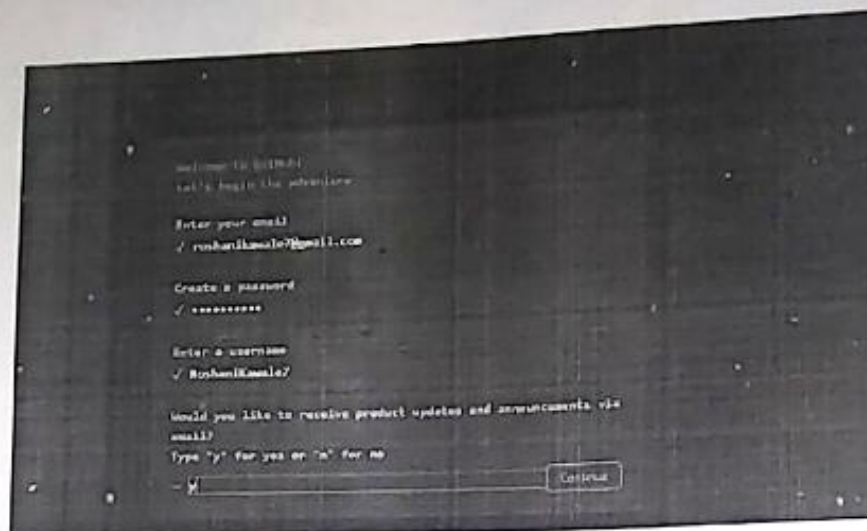
1. Open <https://github.com> in a web browser, and then select **Sign up**.



2. Enter your email address.



3. Create a password for your new GitHub account, and Enter a username, too. Next, choose whether you want to receive updates and announcements via email, and then select **Continue**.

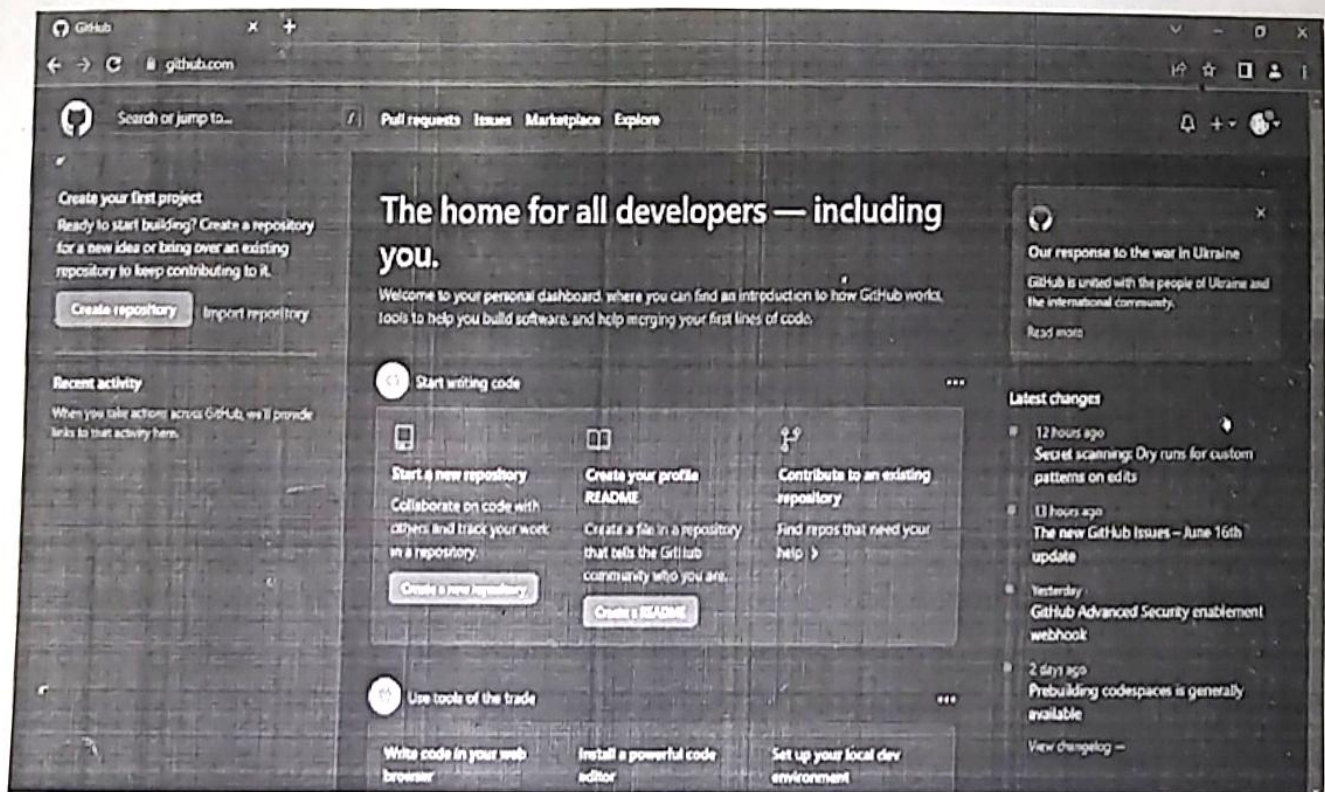


4. **Verify your account** by solving a puzzle. Select the **Start Puzzle** button to do so, and then follow the prompts.
5. After you verify your account, select the **Create account** button.
6. Next, GitHub sends a launch code to your email address. Type that launch code in the **Enter code** dialog, and then press **Enter**.



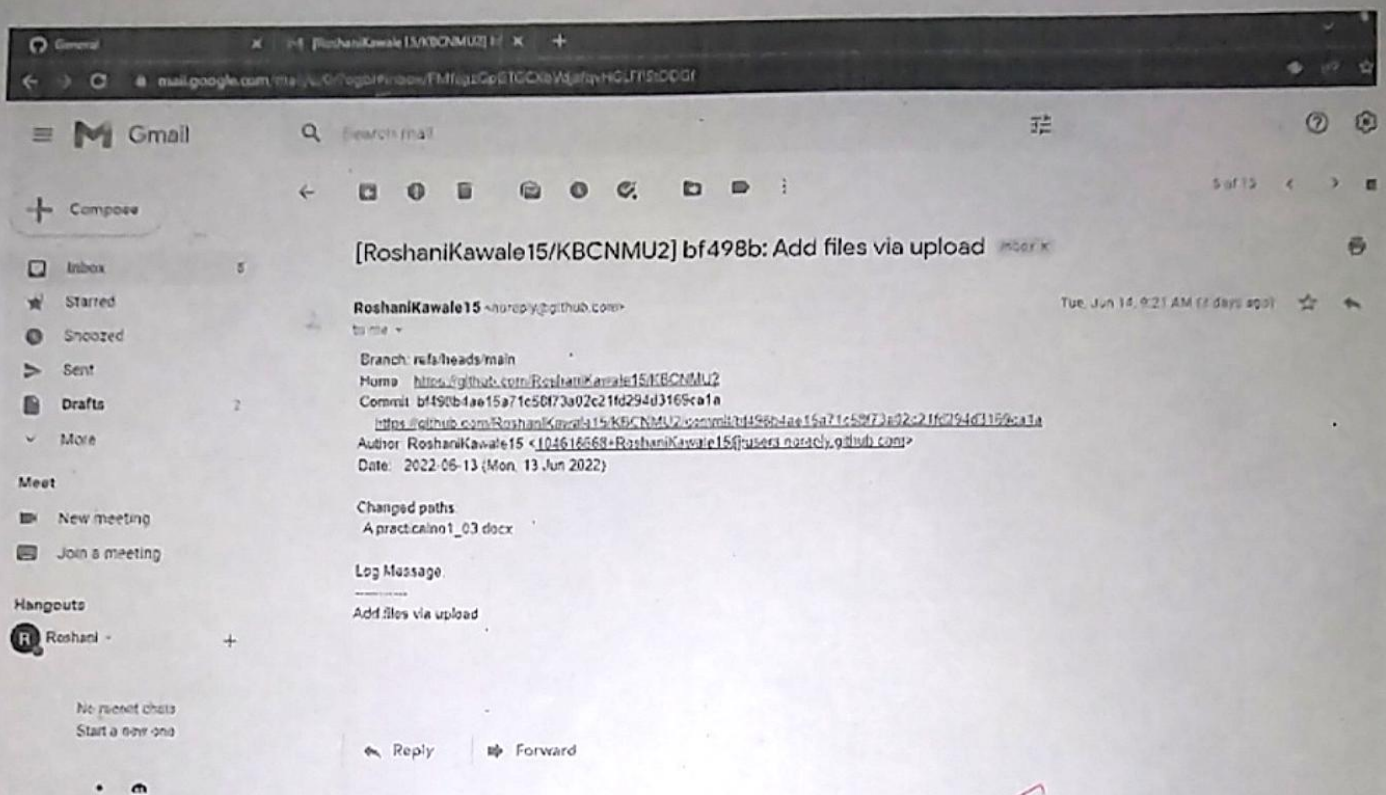
7. GitHub asks you some questions to help tailor your experience. Choose the answers that apply to you in the following dialogs:
 - **How many team members will be working with you?**
 - **What specific features are you interested in using?**
8. On the **Where teams collaborate and ship** screen, you can choose whether you want to use the Free account or the Team account. To choose the **Free** account, select the **Skip personalization** button.

GitHub opens a personalized page in your browser.



Congratulations! You've successfully created your GitHub account.

Paste the screen shots of email which you had send to your friend through Github



**DEPARTMENT OF COMPUTER SCIENCE
INSTITUTE OF MANAGEMENT AND RESEARCH, JALGAON**

Name Kawale Roshani Vasant

Expt. Title Clone the repository at <https://github.com/KBCNMU>.

Class EXMCA Batch B2 Performed on _____

Roll No. 48 Expt. No. 2 Submitted on _____

Remarks _____ Returned on _____

★ Introduction of Git :-

Git is an open-source distributed version control system. It is designed to handle minor to major projects with high speed and efficiency. It is developed to coordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

Git is foundation of many services like GitHub and GitLab, but we can use git without any other Git services. Git can be used privately and publically.

★ Features of Git :-

1) Open Source :-

Git is an open-source tool. It is released under the GPL (General Public License) license.

2) Scalable

3) Distributed

4) Security

5) Speed

6) Supports non-linear development

7) Branching and Merging

8) Data Assurance

9) Staging Area

10) Maintain the clean history.

★ Benefits of Git :-

1) Saves Time :-

Git is lightning fast technology. Each commands takes only a few seconds to execute so we can save a lot of time as compared to login to a GitHub account and find out its features.

or:

e Listing

2) Offline Working:-

One of the most important benefits of Git is that it supports offline working. If we are facing internet connectivity issues, it will not affect our work.

3) Undo Mistakes:-

One additional benefit of Git is we can undo mistakes. Sometimes the undo can be a savior option for us.

★ What does Git do?

- Manage projects with Repositories
- clone a project to work on a local copy
- control and track changes with staging and committing.
- Pull the latest version of a project to a local copy.
- Push local updates to the main project.

★ Commands :-

1) Version :- This command is used to list the version history for the current branch.

Syntax:- \$git --version.

2) Help :- This command is used to display all the information about git.

syntax:- \$git --help.

3) clone :- Git allows making a copy of only particular branch from a repository. You can make a directory for the individual branch by using the git clone command. It creates a copy of an existing Git repository. cloning is the most common way for developers to obtain a working copy of a central repository.

syntax:-

\$git clone <Repository URI>

Roll No: - 48

Practical No:-2

Practical Name:- Clone the repository at <https://github.com/KBCNMU>.

```
comp@comp MINGW32 ~
$ git --version
git version 2.36.0.windows.1
```

```
comp@comp MINGW32 ~
$ git --help
usage: git [--version] [--help] [-c <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

- clone Clone a repository into a new directory
- init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

- add Add file contents to the index
- mv Move or rename a file, a directory, or a symlink
- restore Restore working tree files
- rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

- bisect Use binary search to find the commit that introduced a bug
- diff Show changes between commits, commit and working tree, etc
- grep Print lines matching a pattern
- log Show commit logs
- show Show various types of objects
- status Show the working tree status

grow, mark and tweak your common history

- branch List, create, or delete branches
- commit Record changes to the repository
- merge Join two or more development histories together
- rebase Reapply commits on top of another base tip
- reset Reset current HEAD to the specified state
- switch Switch branches
- tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

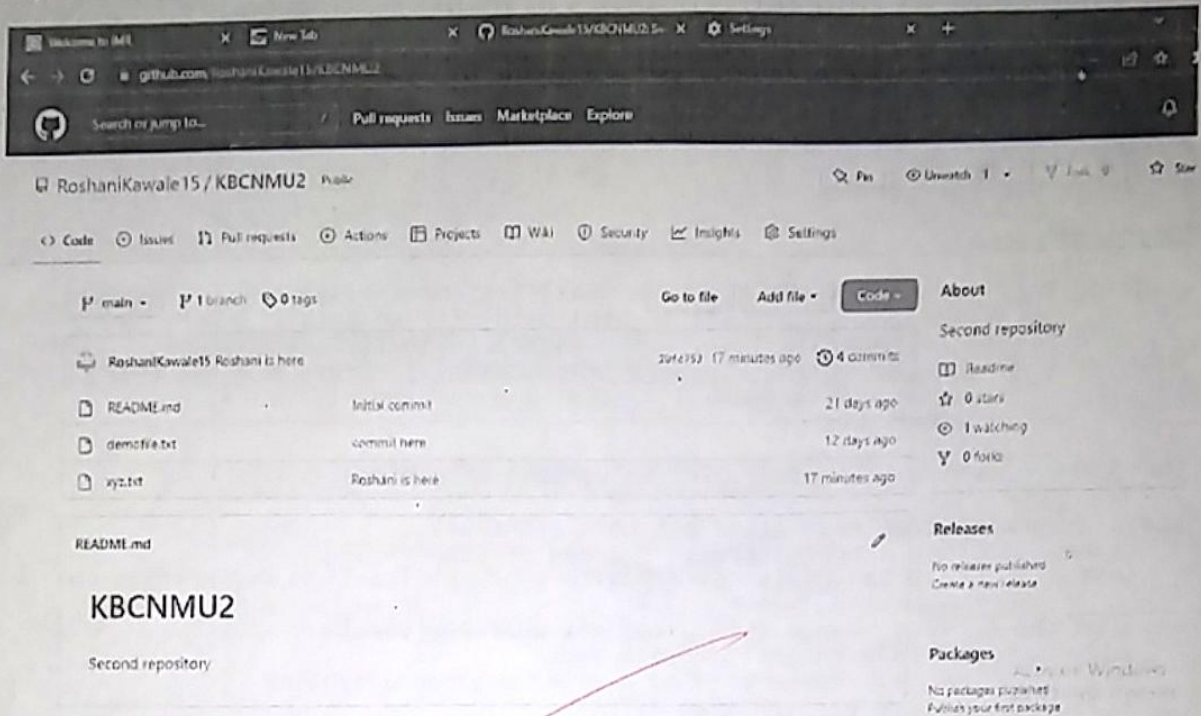
- fetch Download objects and refs from another repository
- pull Fetch from and integrate with another repository or a local branch
- push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

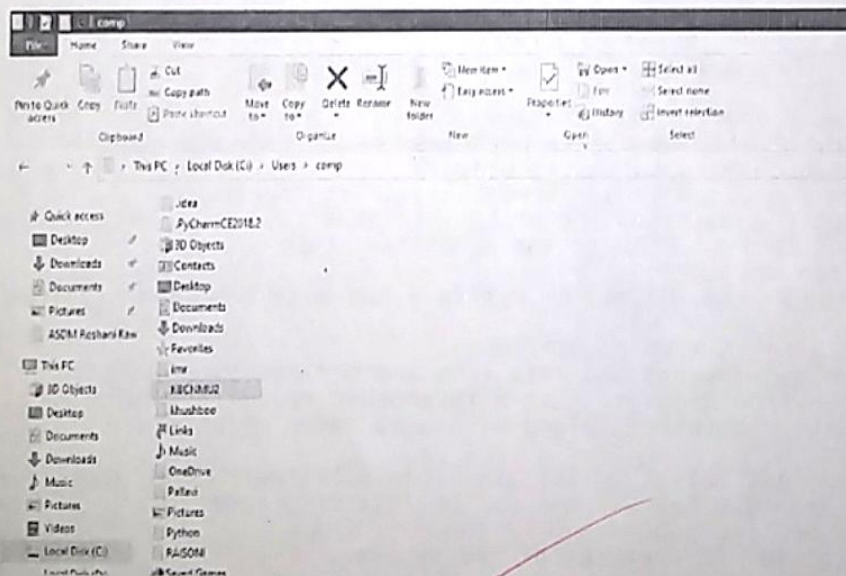
```
comp@comp MINGW32 ~
$ git clone https://github.com/RoshaniKawale15/KBCNMU2.git
Cloning into 'KBCNMU2'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
```

OUTPUT:-

Repository on GitHub:



Repository on Local PC:



©
Roshani