INTIII

DATE:

INTRODUCTION TO DEVOPS

**AIM:**

To Explore DevOps.

**CASE STUDY:**

**DEVOPS:**

Azure DevOps provides developer services for allowing teams to plan work, collaborate on code development, and build and deploy applications. Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

**TOOLS AND SETTINGS:**

**BOARDS:**

Combine drag-and-drop sprint planning and flexible work item tracking with comprehensive traceability to have the perfect home for all your ideas–big and small.

**PIPELINES:**

Azure Pipelines automatically builds and tests code projects to make them available to others. It works with just about any language or project type. Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target.

**REPOS:**

Get unlimited private Git repository hosting and support for TFVC that scales from a hobby project to the world’s largest repository.

**TEST PLANS:**

Azure Test Plans, a service launched with [Azure DevOps](https://azure.microsoft.com/en-us/blog/introducing-azure-devops/) earlier this month, provides a browser-based test management solution for exploratory, planned manual, and user acceptance testing. Azure Test Plans also provides a browser extension for exploratory testing and gathering feedback from stakeholders.

**ARTIFACTS:**

Add fully integrated package management to your continuous integration/continuous delivery (CI/CD) pipelines with a single click. Create and share Maven, npm, NuGet and Python package feeds from public and private sources with teams of any size.

EX NO:

**AIM:**

To Install Git on your Local Device.

**CASE STUDY:**

1. To download the latest version of Git, click on the link below

[**https://git-scm.com/download/win/**](https://git-scm.com/download/win/)

1. Click 64-bit for Windows Setup**.**
2. After your download is complete, Run the .exe file in your system.
3. After you have pressed the **Run** button and agreed to the license, you will find a window prompt to select components to be installed.

After you have made selection of your desired components, click on **Next>**.

1. The next prompt window will let you choose the adjustment of your path environment. This is where you decide how do you want to use Git.

You can select any of the three options according to your needs. But for beginners, I recommend using **Use Git From Git Bash Only.**

1. The next step is to choose features for your Git. You get three options and you can choose any of them, all of them or none of them as per your needs. Let me tell you what these features are:

The first is the option to **Enable file system caching**.

The second option is to enable **Git Credential Manager**.

The third option is to **Enable symbolic links**.

1. Choose your terminal.

You can choose one from the options.

The default terminal of MYSYS2 which is a collection of GNU utilities like bash, make, gawk and grep to allow building of applications and programs which depend on traditionally UNIX tools to be present.

Or you can choose the window’s default console window (cmd.exe).

INSTALLING GIT

Date:

EX NO:

**AIM;**

**To Explore Revison Control Sytems.**

**CASE STUDY:**

1. Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.
2. Git was originally authored by **[Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds" \o "Linus Torvalds) in 2005** for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with other kernel developers contributing to its initial development. Since 2005, **Junio Hamano has** been the core maintainer. As with most other [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) systems, and unlike most [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) systems, every Git [directory](https://en.wikipedia.org/wiki/Directory_(computing)) on every [computer](https://en.wikipedia.org/wiki/Node_(networking)) is a full-fledged [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full version-tracking abilities, independent of network access or a central server. **Git is**[**free and open-source software**](https://en.wikipedia.org/wiki/Free_and_open-source_software) distributed under the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license
3. let’s explain a scenario **before Git:**

* Developers used to submit their codes to the central server without having copies of their own
* Any changes made to the source code were unknown to the other developers
* There was no communication between any of the developers

1. Now let’s look at the scenario **after Git:**

* Every developer has an entire copy of the code on their local systems
* Any changes made to the source code can be tracked by others
* There is regular communication between the developers

1. **Features of Git**

* Tracks history
* Free and open source
* Supports non-linear development
* Creates backups
* Scalable
* Supports collaboration
* Branching is easier and Distributed development

.

EXPLORING REVISION CONTROL SYTEMS

Date:

EX NO:

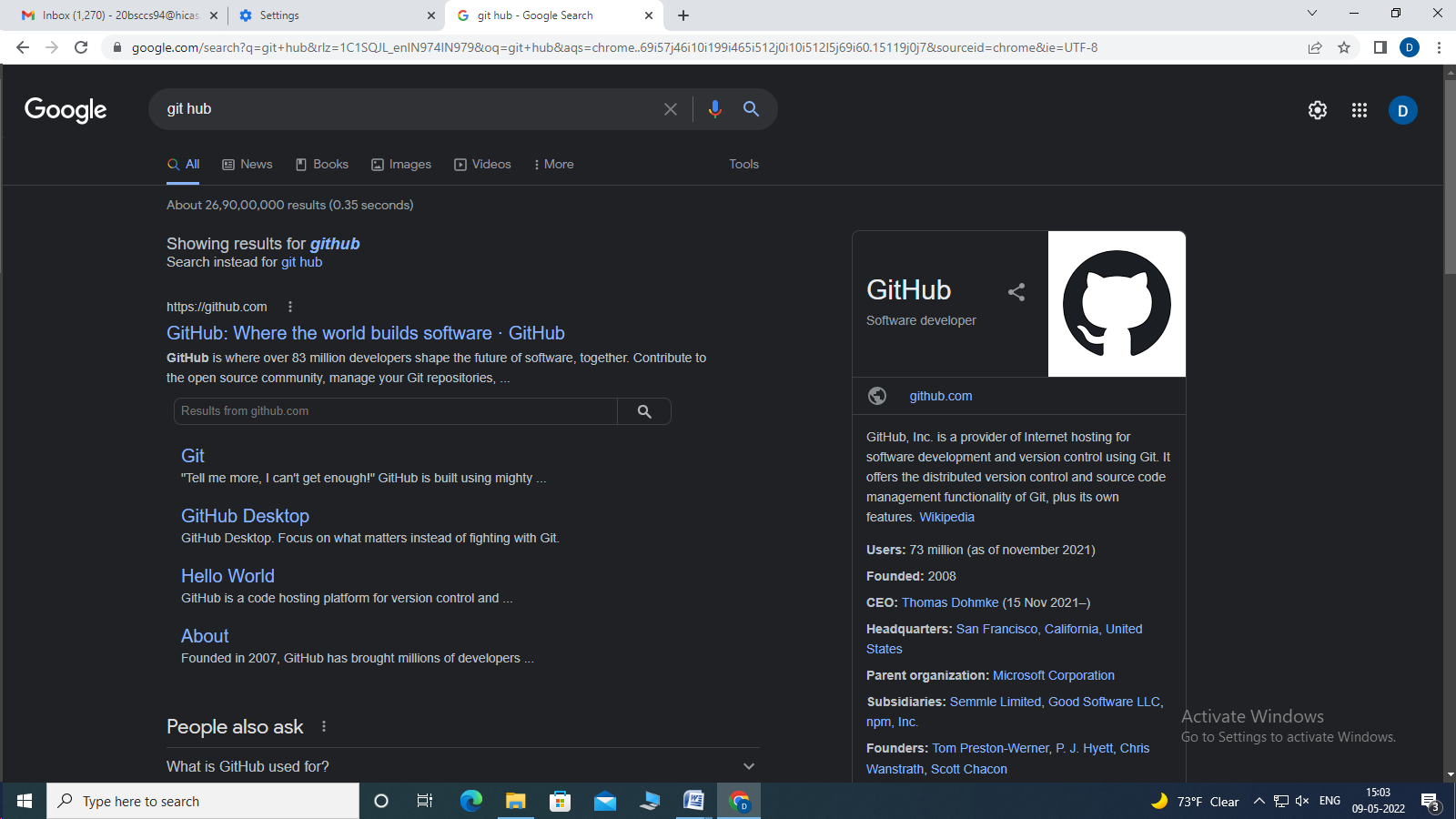
**AIM;**

To Work with Git Hub.

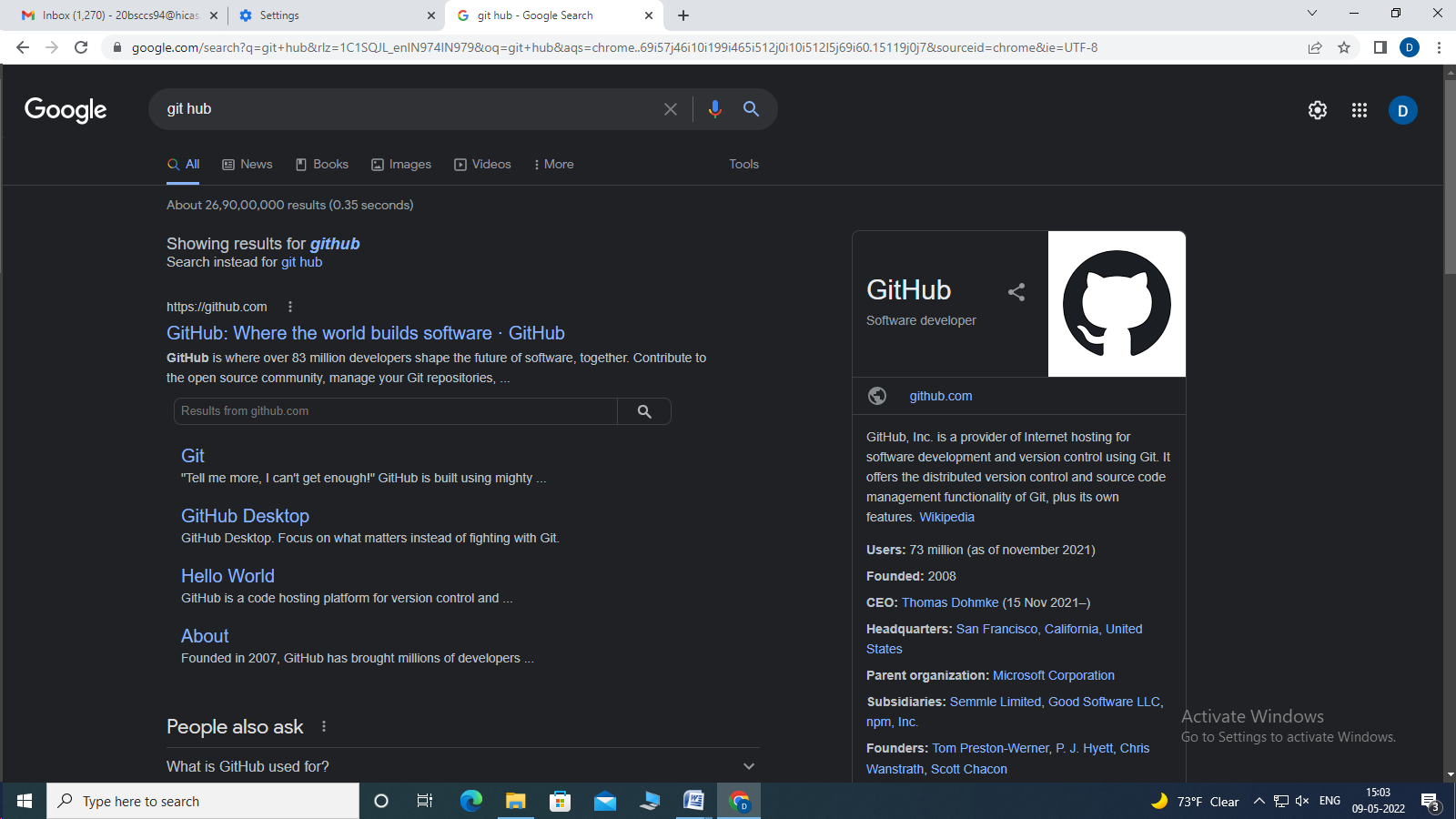
**ALGORITHM:**

**Step 1:** Open Google Chrome or any Browser in your Device.

**Step2 :** Search Github in the Browser and click it.



**Step3 :** The below screen will appear and click Sign in Option.

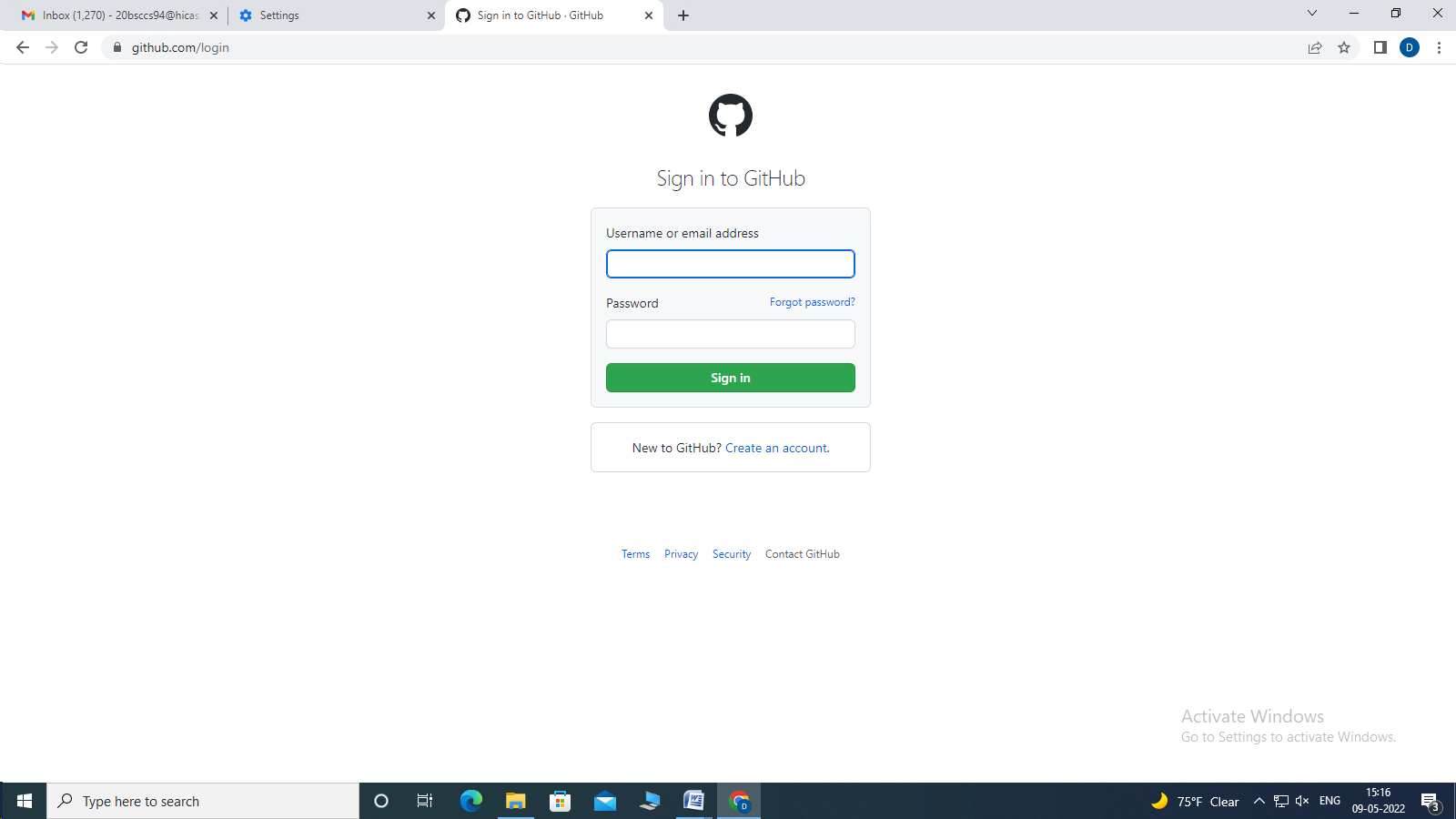


WORKING WITH GIT HUB

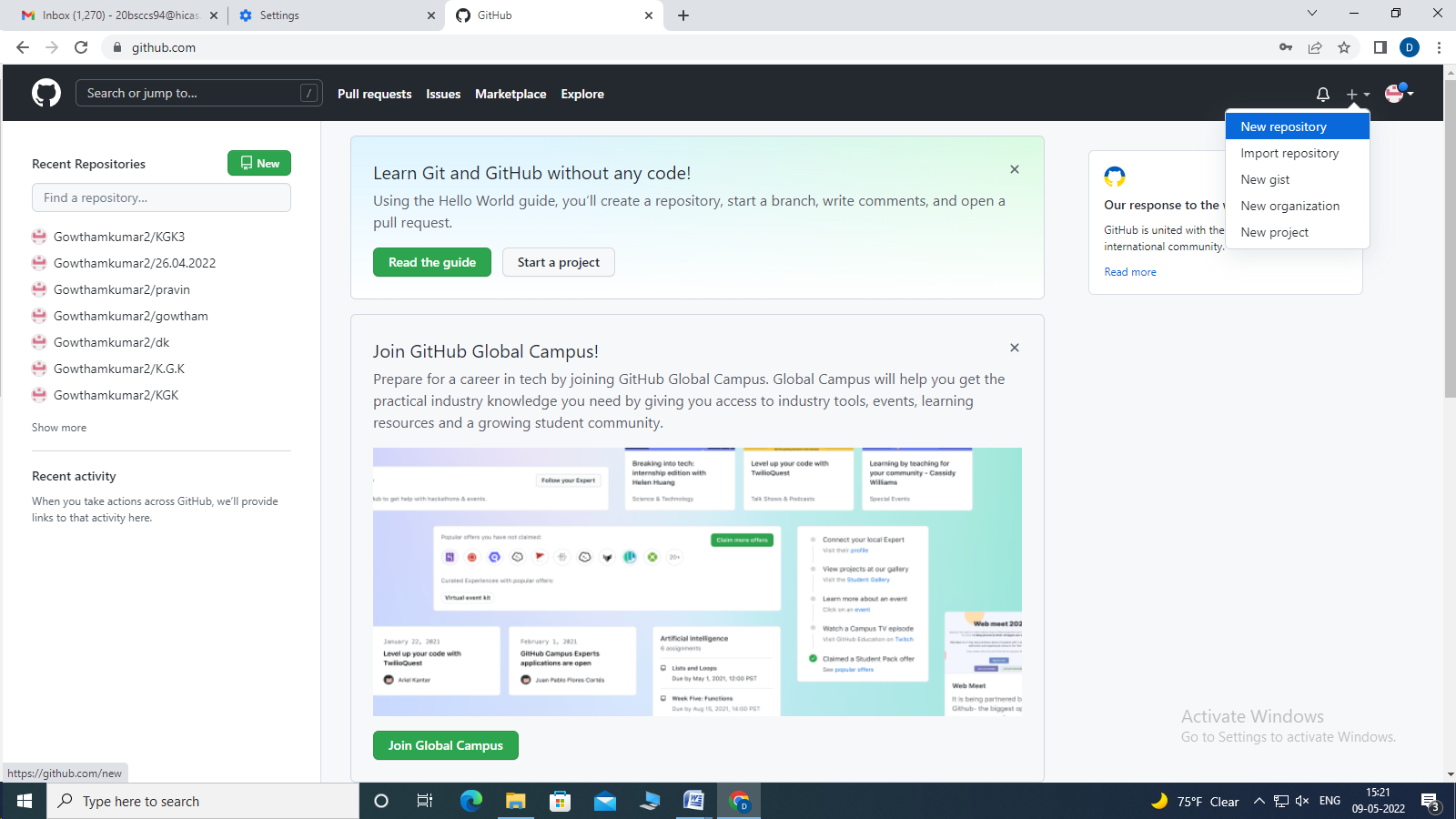
EX NO:

DATE:

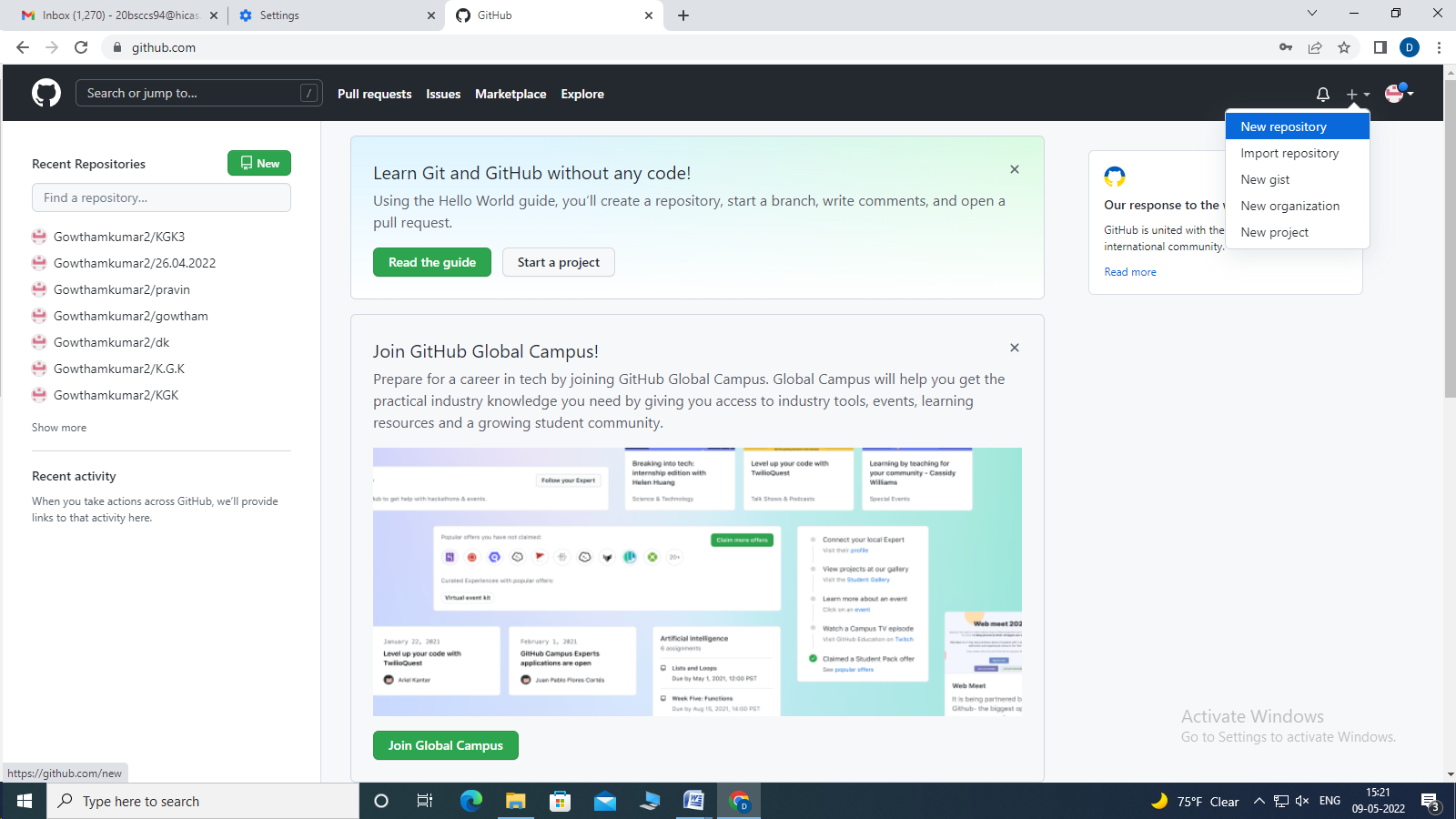
Step4: Logiin using your ID and Password.



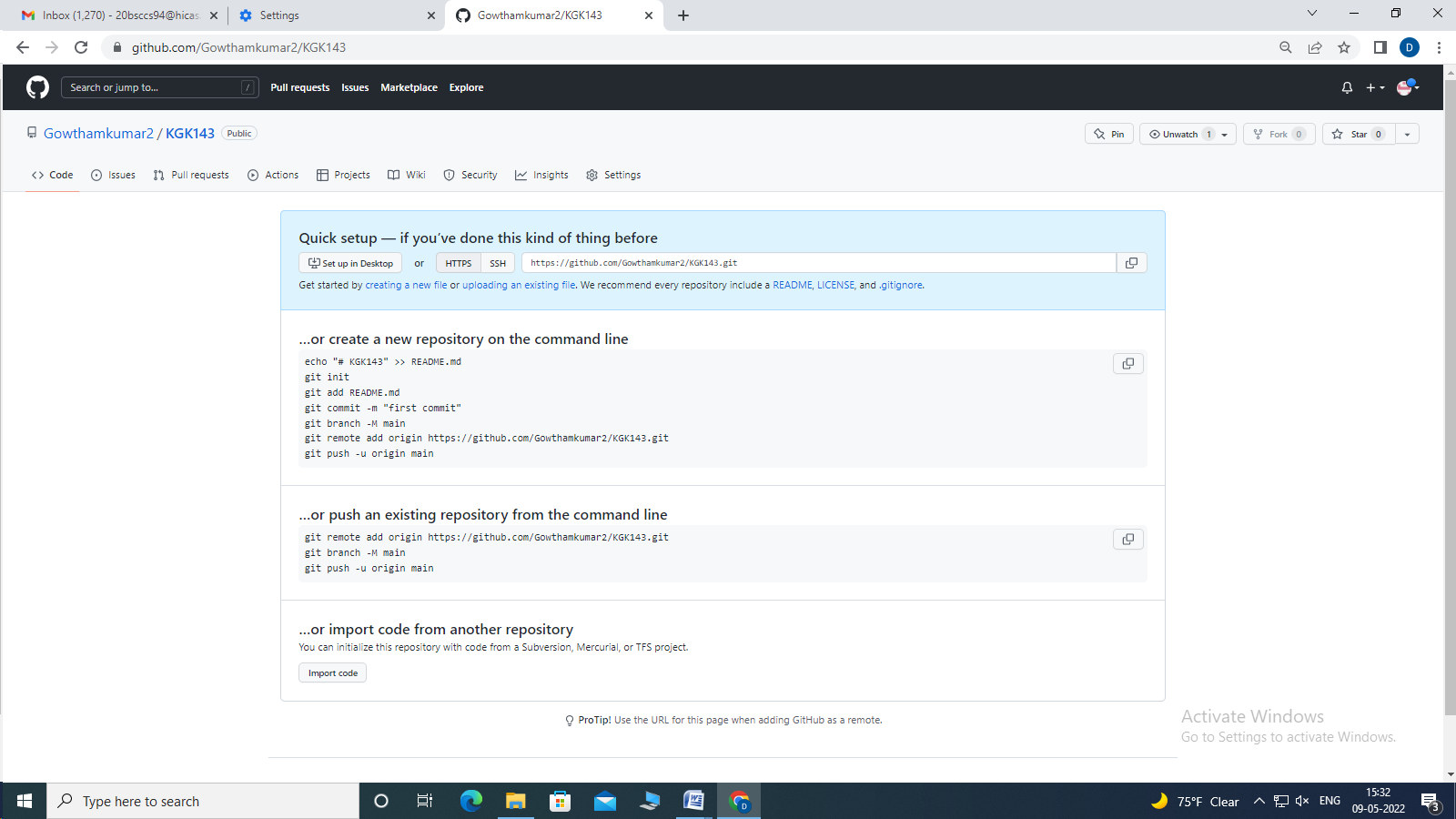
Step5 : Next, the below screen will appear then click plus(+) on the top right corner and select New Repository.



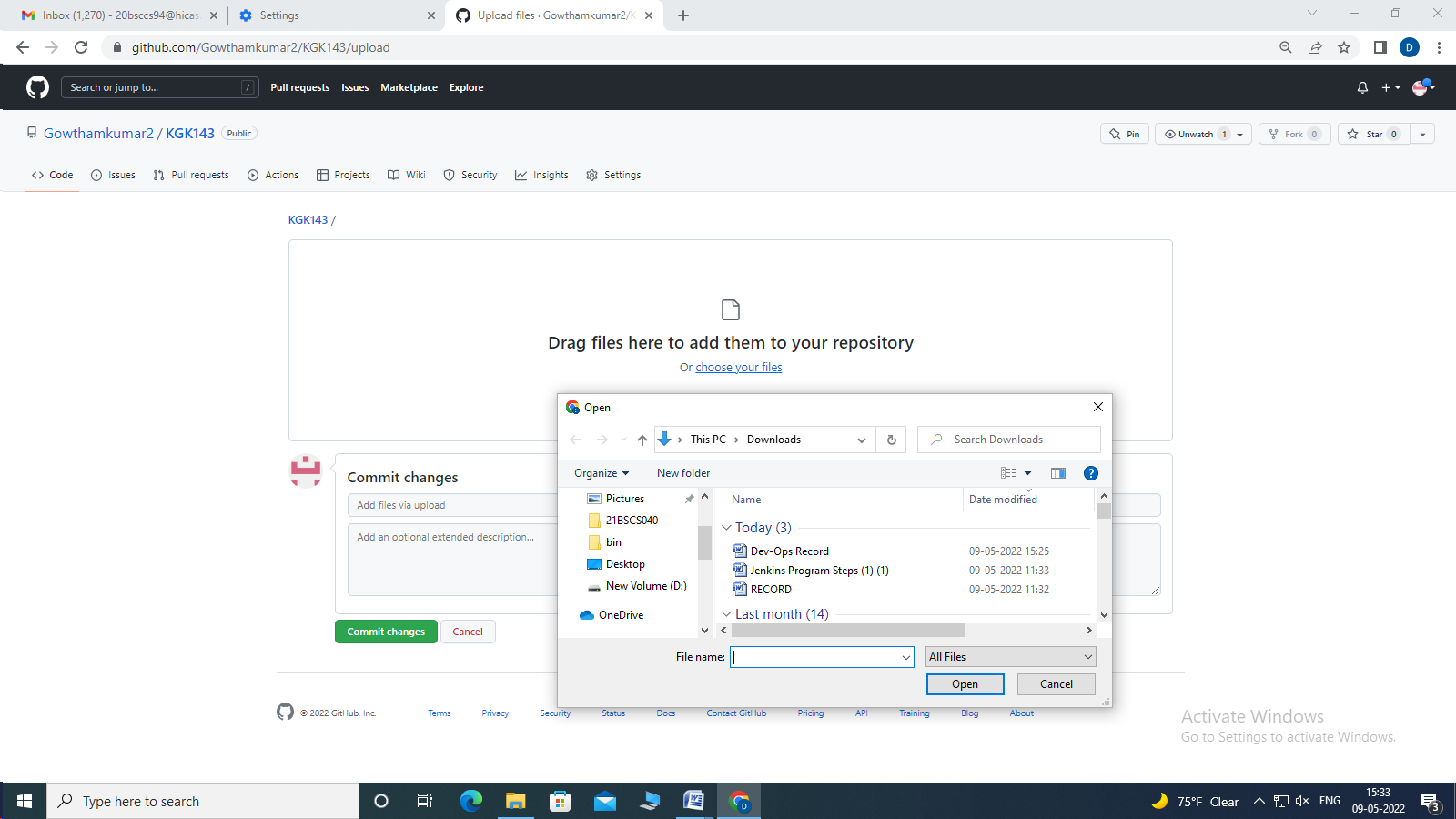
Step 6: Name your repository and give the required details below then click “Create repository”.



Step7 : The Below Screen will appear then we upload an exisiting file from our device by clicking “upload an exisiting file”.



Step8 : Choose your file and Upload it.



Step9 : Then Click “Commit Changes”

