SE LAB TASK 1

27-11-2024

*Sai Ganesh Eswaraprasad*

*HU22CSEN0100287*

*What is GitHub?*

GitHub is a website where people can store and share their computer code. It's like an online folder where you can put your projects, work on them with other people, and keep track of changes over time.

Think of it as a tool that helps developers collaborate. You can:

1. Store your code: Upload your programming files so they are safe and accessible from anywhere.
2. Collaborate: Other people can work on your project, suggest changes, or fix problems, and you can review and approve those changes.
3. **Track changes**: GitHub helps keep a history of all the changes made to the code, so you can see who changed what and why.

It's based on a tool called **Git**, which keeps track of different versions of files, so you can always go back to an earlier version if needed.

Many people use GitHub for software development, but it's also used for other types of projects, like writing documentation or managing data.

*Introduction to Github*

GitHub is a platform used by developers and other project teams to store, share, and manage code and projects. It’s not just a place to save your files; it’s a powerful tool that helps teams collaborate, track progress, automate tasks, and organize work more efficiently.

Here’s a simple breakdown of GitHub’s features and how it works:

*Key Features of GitHub*

1. Repositories (Repos)

* A repository is like a project folder where all your files (code, documents, etc.) are stored on GitHub.
* You can make repositories public (anyone can see and contribute) or private (only selected people can access).

2. Issues and Project Management

* Issues are used to track tasks, bugs, or improvements. They are like to-do items for the project.
* You can assign issues to people, label them (e.g., bug, enhancement), and track their progress.
* Project Boards help organize tasks visually (like Trello) with columns like "To Do," "In Progress," and "Done."

3. Branches

* Branches allow you to work on different parts of the project without affecting the main code (usually the main or master branch).
* You can create a branch for new features, bug fixes, or experiments, and later merge them back into the main project.

4. Pull Requests (PRs)

* A pull request is a way to propose changes to a project. You can request that your changes (from a branch) be reviewed and merged into the main code.
* It’s also where you discuss changes with your team and ensure everything is working correctly before merging.

5. GitHub Actions and Automation

* GitHub Actions are automated workflows that run tasks like testing, building, or deploying your project.
* For example, you can set up an action to automatically run tests every time someone adds new code to the project.
* Actions save time by automating repetitive tasks and making the development process smoother.

6. Version Control (Git)

* GitHub uses Git, which keeps track of all changes made to the code.
* You can always go back to an earlier version of the code if something goes wrong.
* Git also allows multiple people to work on the same project without conflicting changes.

7. Wiki and Documentation

* GitHub allows you to create a Wiki for your project, which is a place to document how your project works, how to set it up, or how others can contribute.
* It helps new contributors understand the project quickly.

8. Security and Access Control

* Branch Protection: You can protect the main code branches, ensuring changes are reviewed before being merged.
* Dependabot: GitHub automatically checks for security vulnerabilities in the libraries your project depends on and can suggest updates to fix them.

9. Community and Open Source

* GitHub is home to millions of open-source projects, allowing anyone to contribute.
* You can fork (copy) a project, make changes, and then submit a pull request to contribute your changes back to the original project.

10. Discussions

* Discussions are a space for project teams and communities to have conversations, ask questions, and share ideas, separate from code issues.
* It's like a forum where you can discuss features, ask for help, or share general thoughts.

11. Collaboration Tools

* GitHub allows teams to collaborate by assigning tasks, reviewing changes, and discussing updates through issues, pull requests, and discussions.
* You can easily share work, review code, and contribute to other projects.

12. Integration with Other Tools

* GitHub can integrate with other services, like Slack, Trello, or CI/CD tools, to streamline communication and development.
* You can link GitHub with tools for automatic deployments, continuous integration, and monitoring.

13. Free and Paid Plans

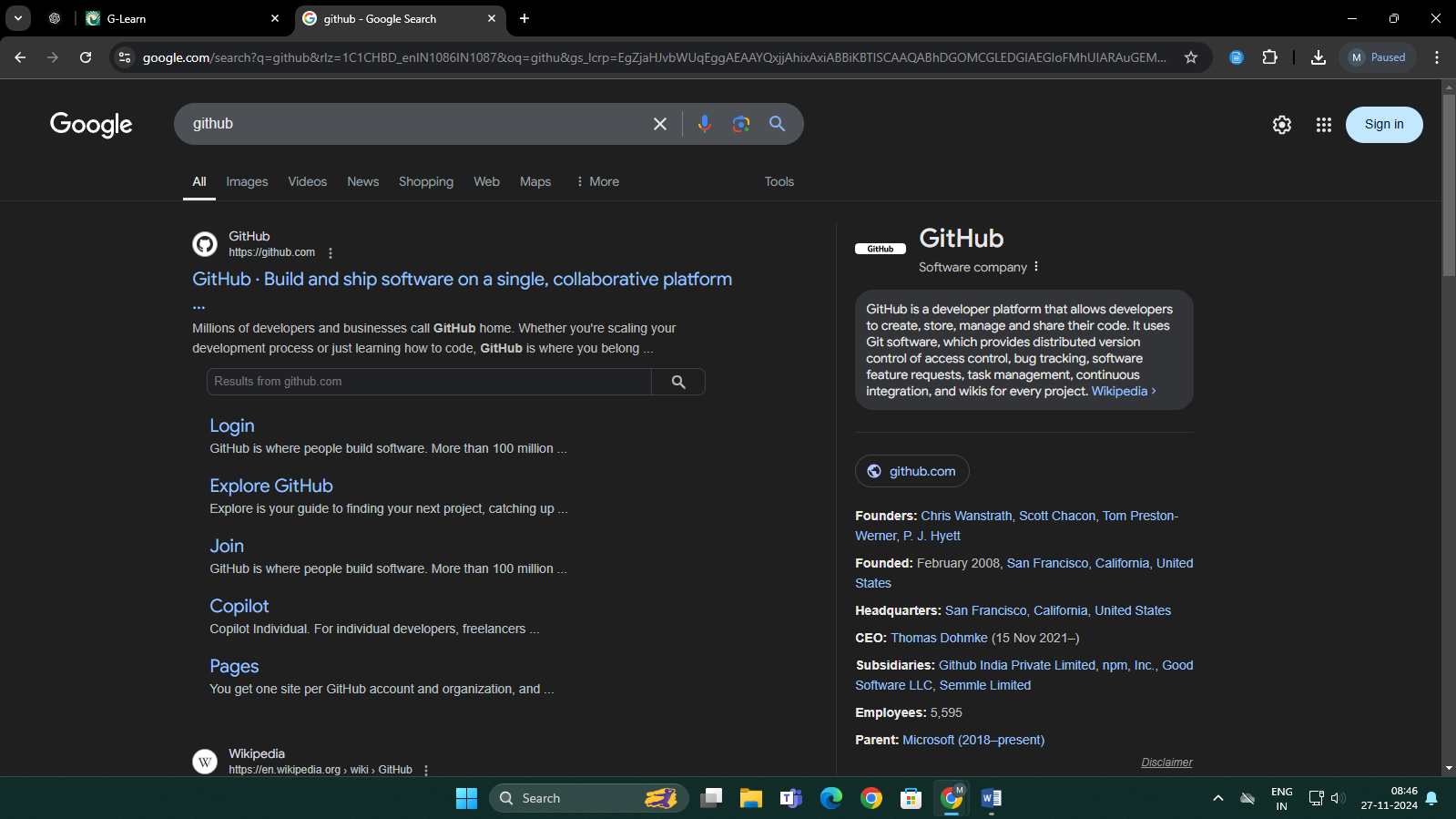
* GitHub is free for public repositories and small teams.
* There are paid plans for private repositories or teams with advanced features like better security controls.

*Difference Between Git And GitHub*

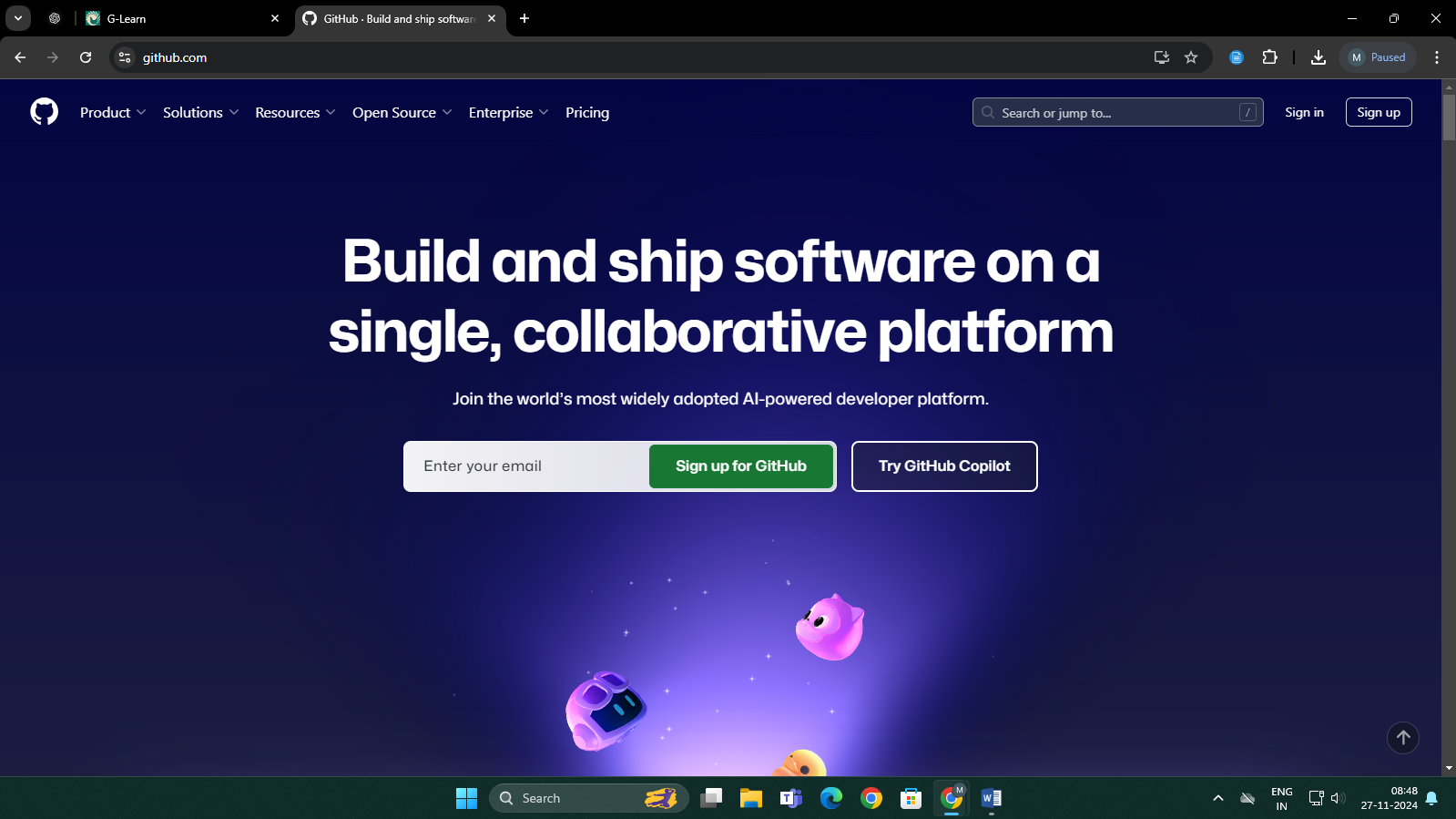
| Feature | Git | GitHub |
| --- | --- | --- |
| Definition | Version control system | Web platform for hosting Git repositories |
| Purpose | Track changes locally on your machine | Host code remotely, enable collaboration |
| Functionality | Manages versions, commits, branches | Stores code, offers collaboration tools (PRs, issues) |
| Hosting | Works locally (on your computer) | Cloud-based, stores code online |
| Usage | Used to manage local code versions | Used to share, collaborate, and manage code remotely |
| Collaboration | No built-in tools for collaboration | Pull requests, issues, discussions, team collaboration |
| Repositories | Local repositories | Remote repositories on GitHub servers |
| Version Control | Tracks changes in local files | Displays version history and commits online |
| Access Control | Local file access only | Public/private repos, team access management |
| Backup | No built-in backup (requires remote push) | Auto backups, cloud storage |
| Security | No built-in security features | Two-factor authentication, security alerts |
| Integration | Works with local tools (e.g., IDEs) | Integrates with CI/CD tools, project management, and more |
| Community | No community features | Supports open-source projects, discussions, and contributions |
| Cost | Free | Free for public repos, paid for private repos and teams |
| Performance | Fast and lightweight (works offline) | Relies on internet connection for access |
| Workflow | Command-line tool for version control | Web interface, GitHub Desktop, and Git CLI |
| Forking | Doesn’t support forking (local only) | Allows users to fork and contribute to other repositories |
| Code Reviews | No built-in code review features | Pull requests with comments and reviews |
| Collaboration History | No tracking of collaboration details | Tracks contributors, pull requests, and comments |
| Open-Source Projects | N/A (Git itself is not a project) | Hosts millions of open-source projects for public collaboration |
| Issue Tracking | No issue tracking | Built-in issue tracking for bugs, tasks, and feature requests |
| Documentation | No built-in docs support | Wiki feature for project documentation, README support |
| Continuous Integration | No built-in CI/CD support | GitHub Actions for automating workflows like testing and deployment |
| Branching | Local branches for feature development | Remote branches on GitHub for team collaboration |
| Community Features | No social aspects | Stars, forks, discussions, and followers for community engagement |

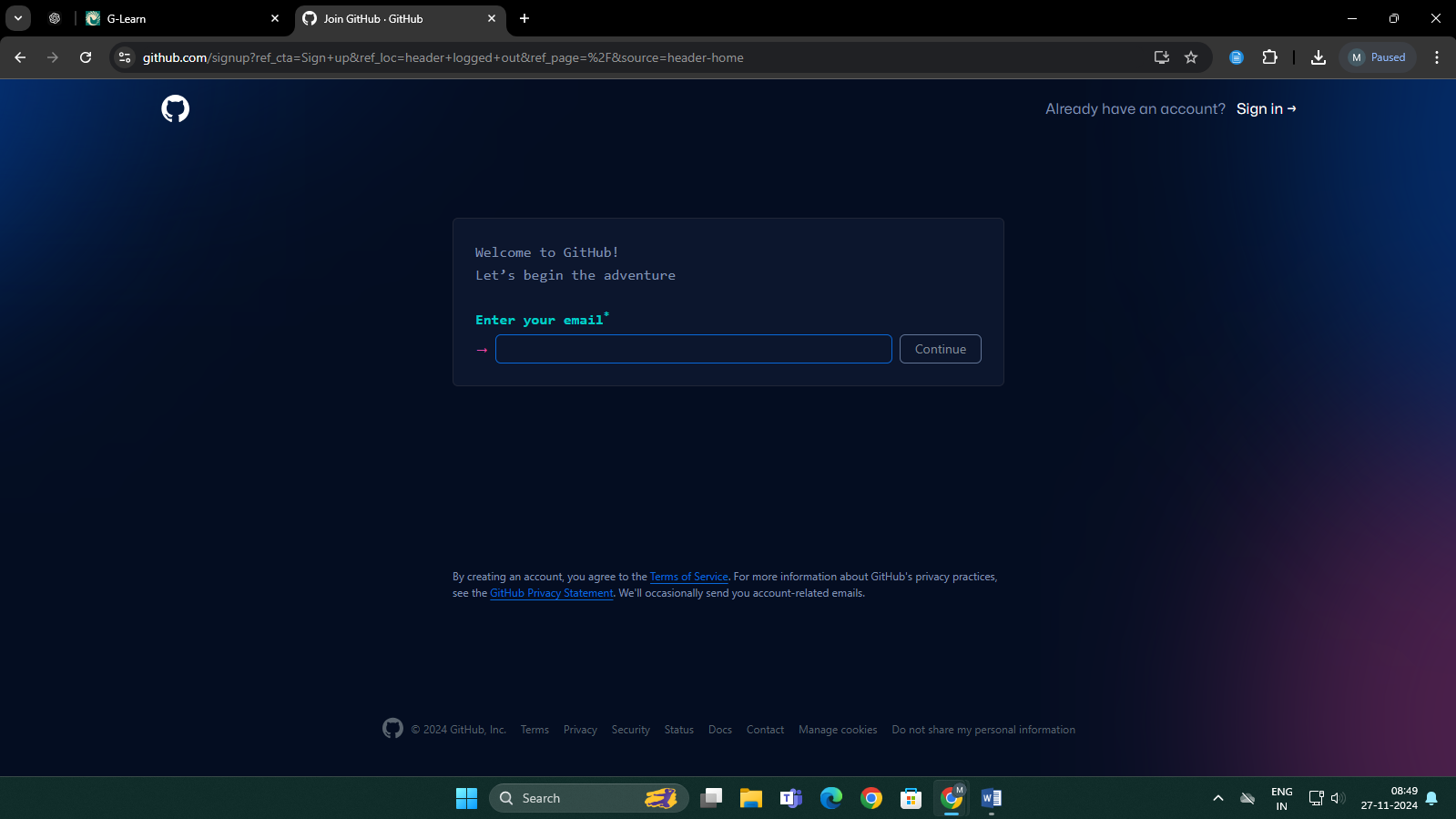
*Getting Started with GitHub*

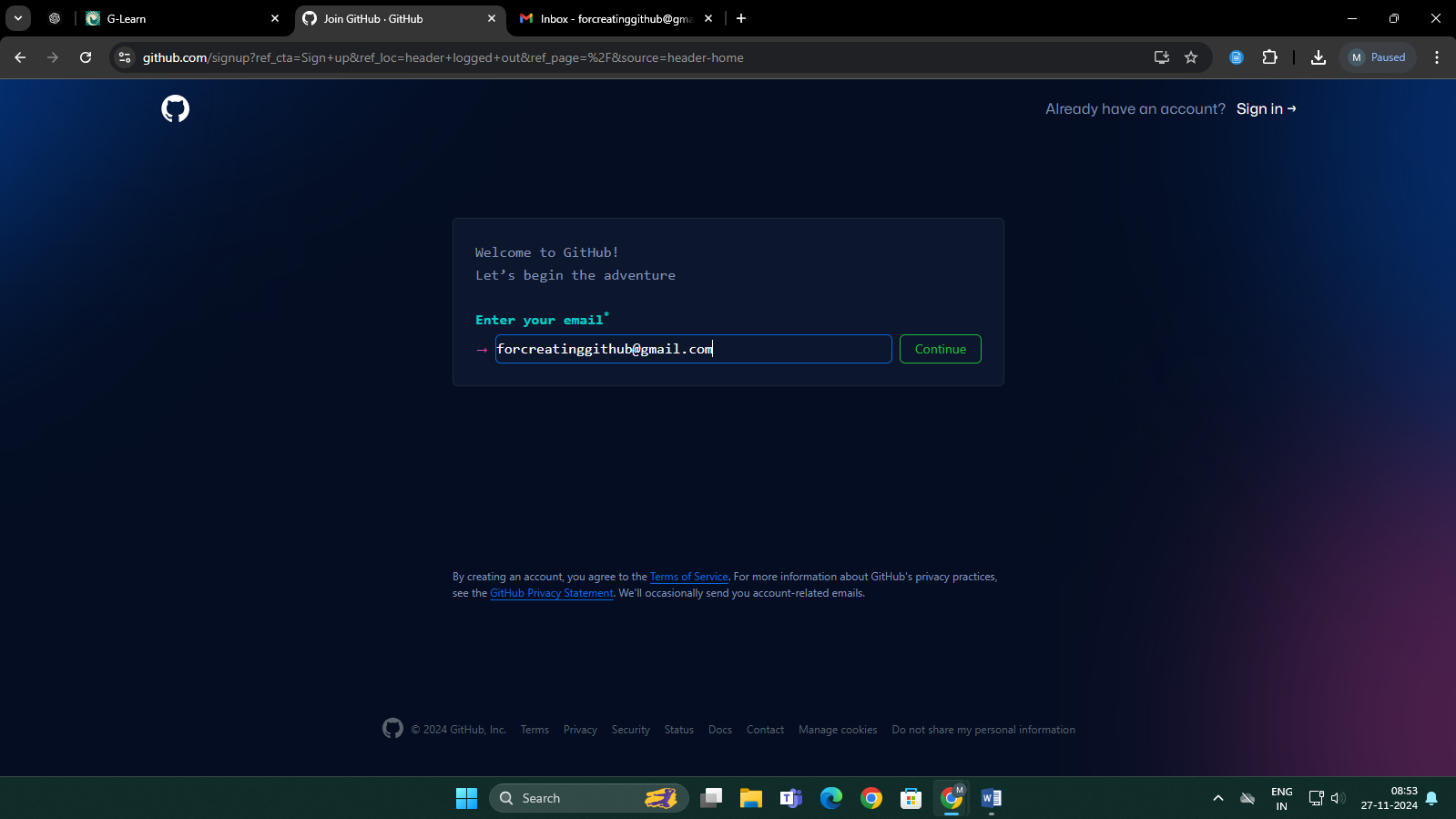
1. Type “Github” in Google searchbar and search for it



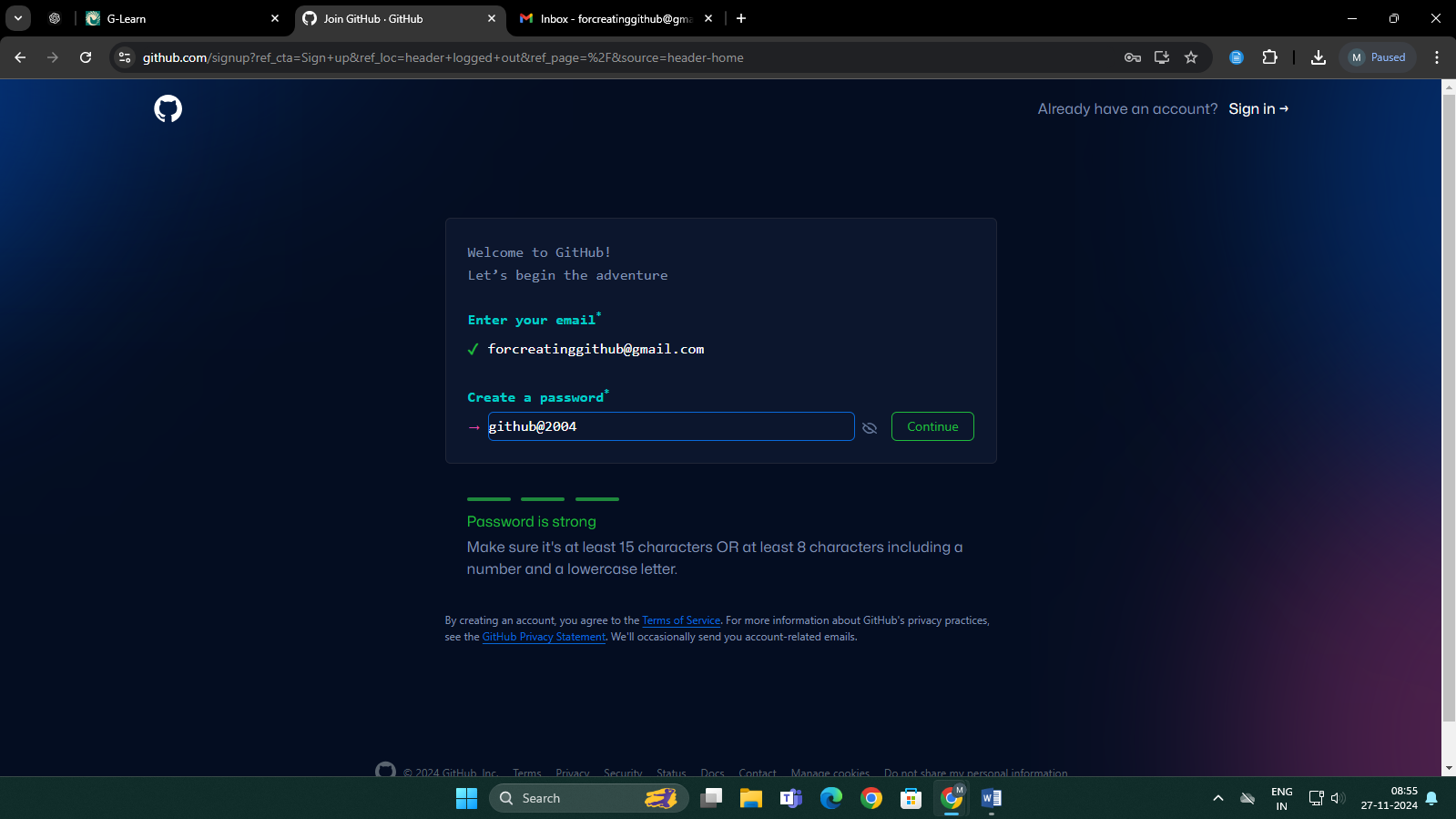
1. Click on the Github Website link which directs to the GitHub Website



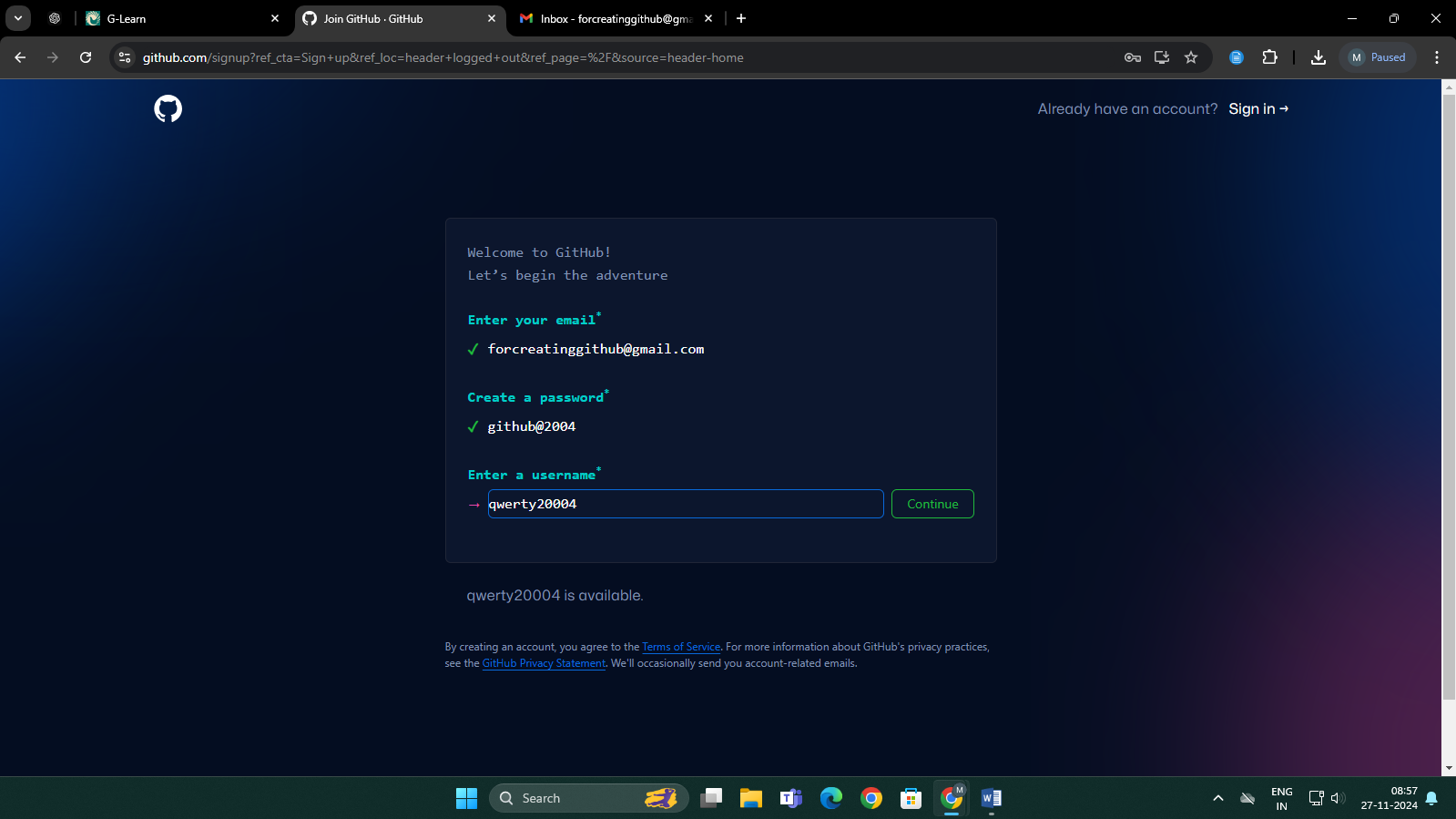
1. Click on “Sign Up” present on the top right
2. Enter email credentials

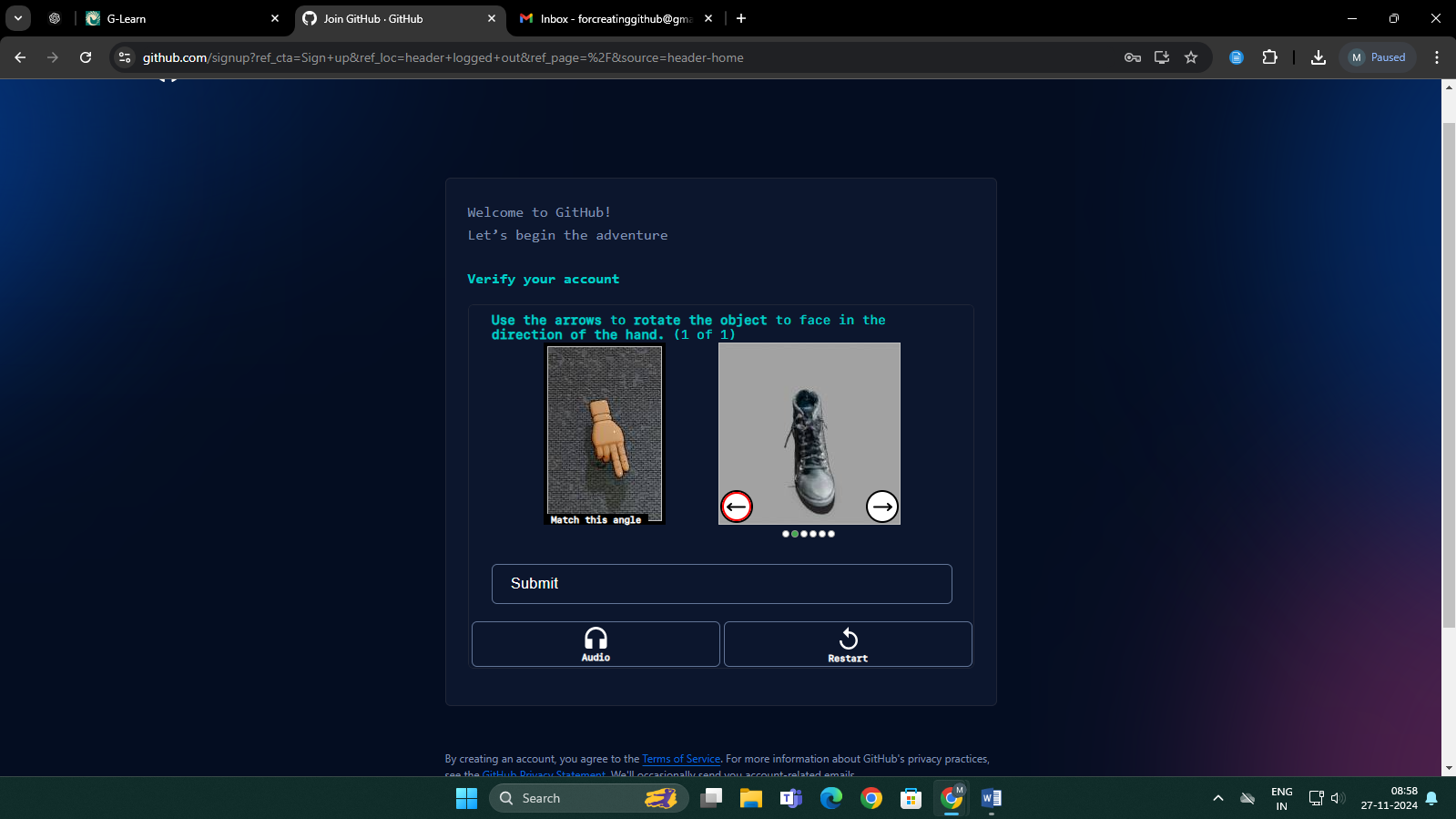


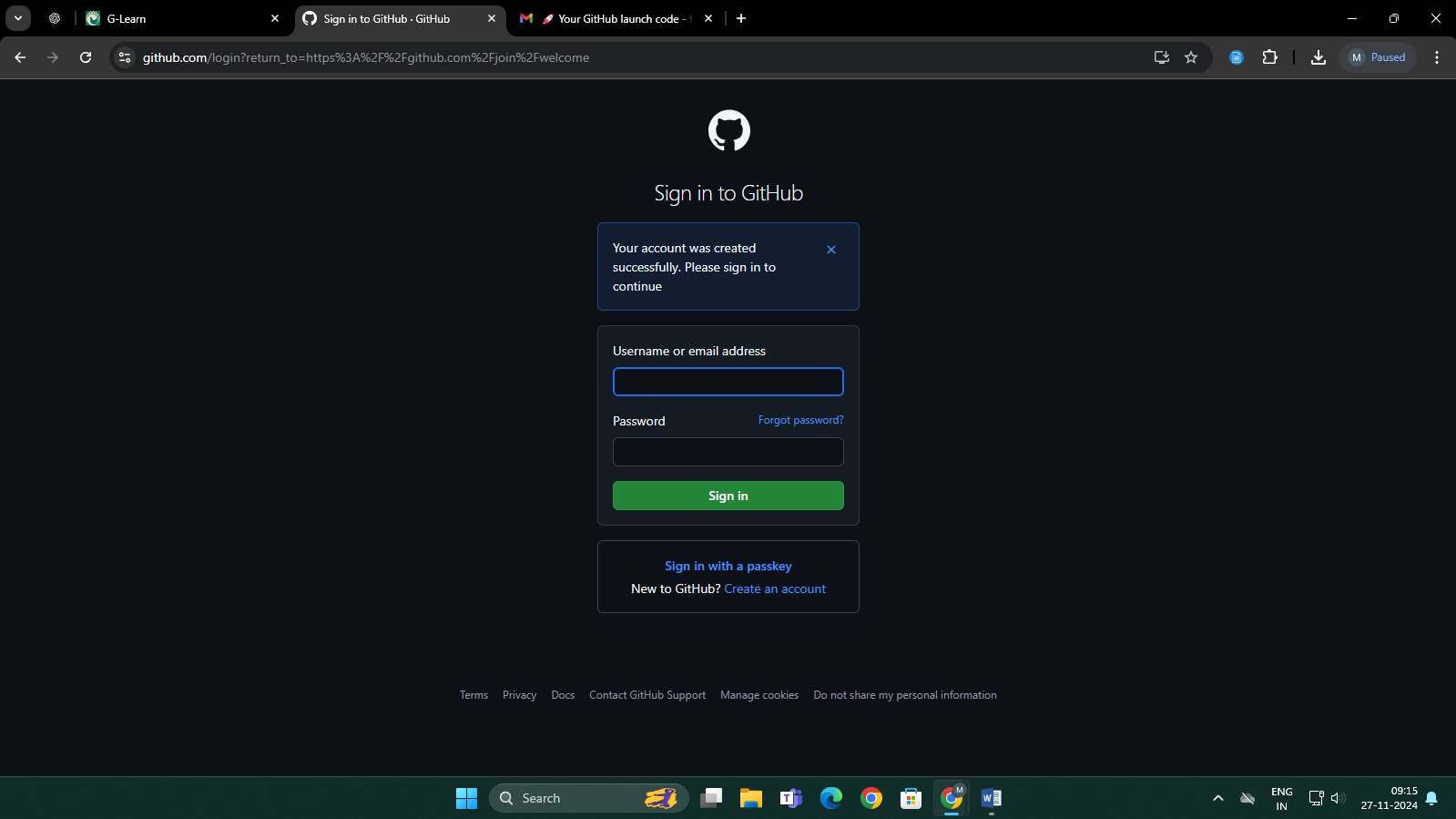
1. Type the password you want to set

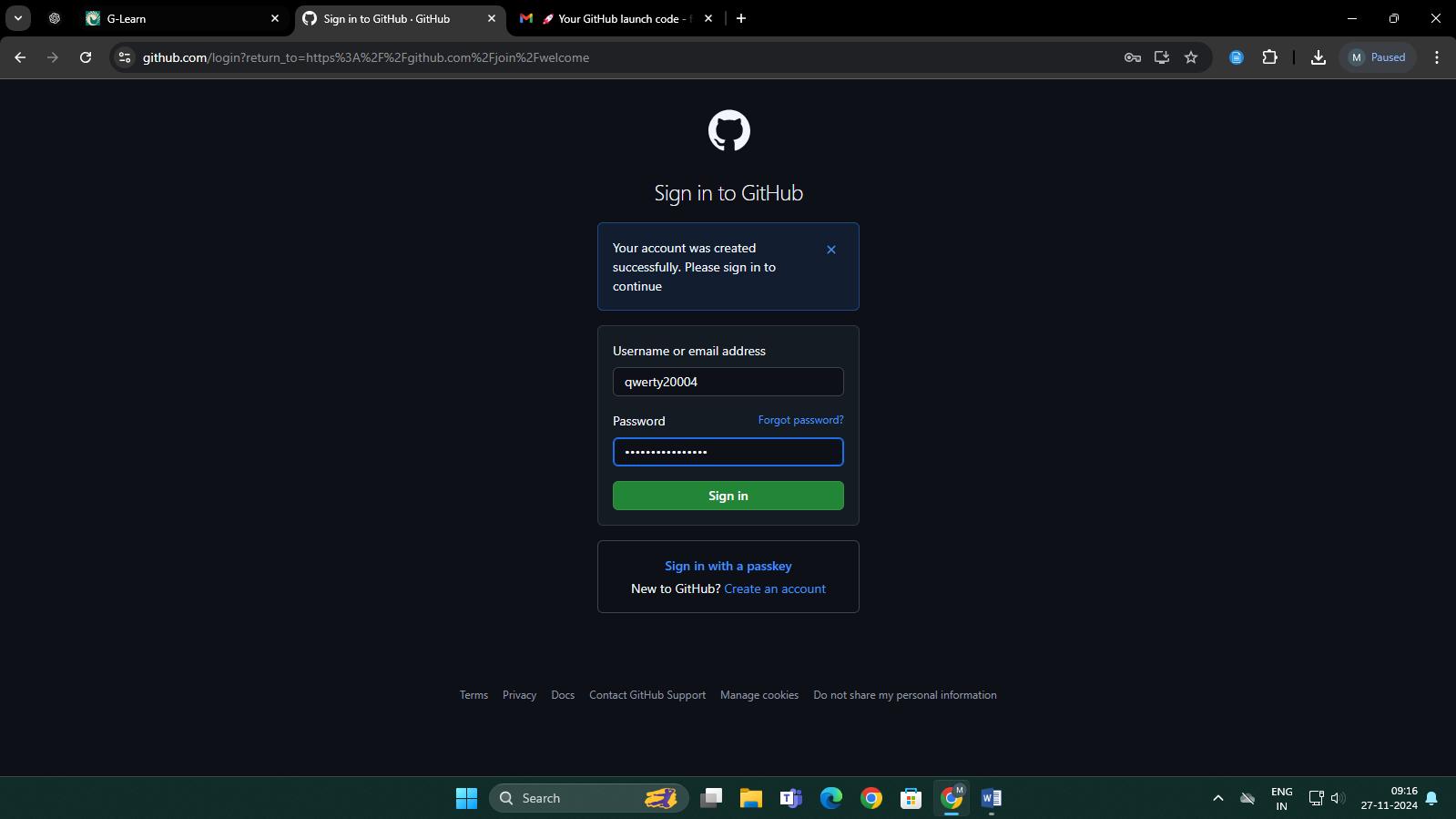


1. Enter Username

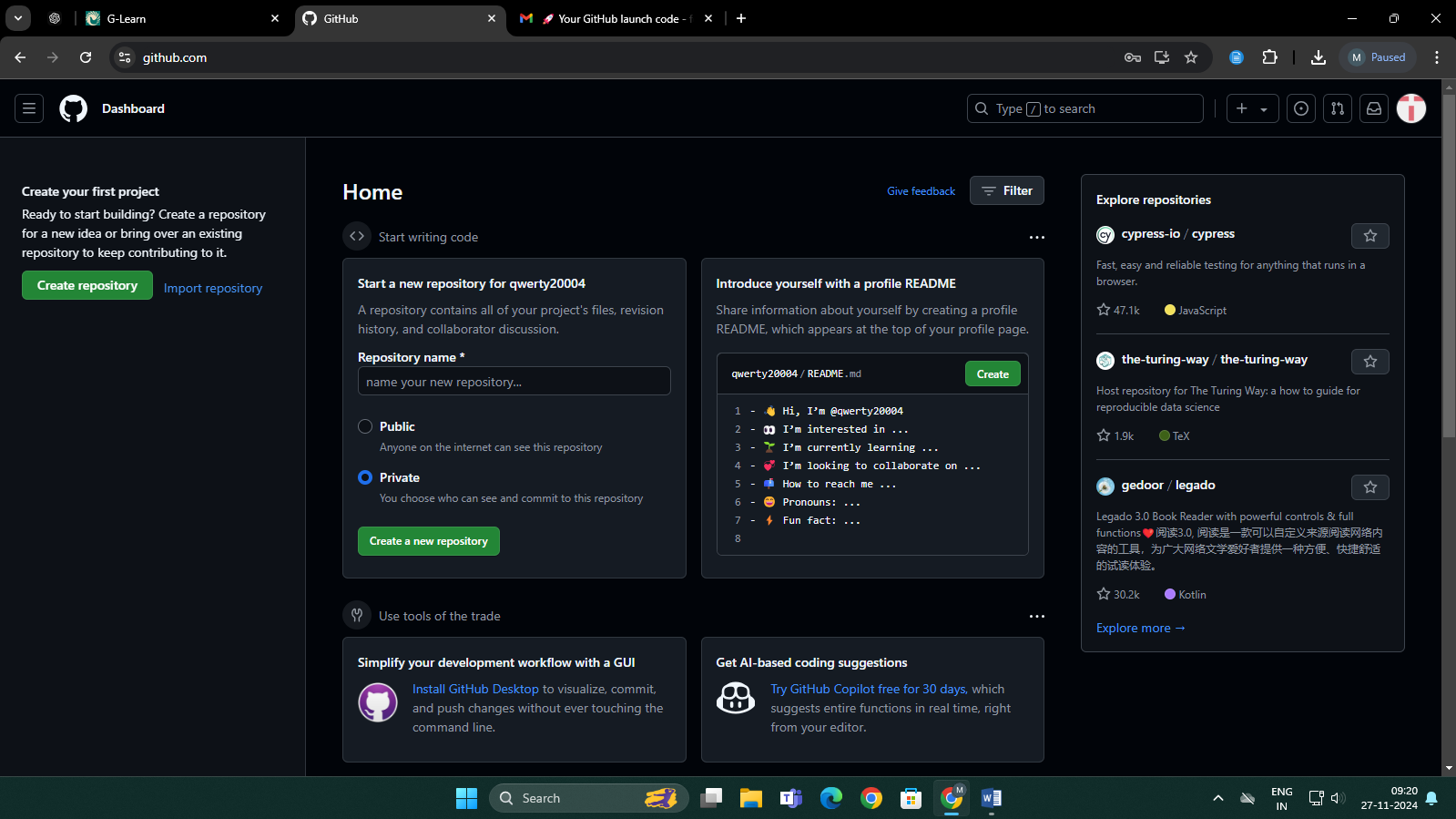


1. Work with required captcha for verification
2. Enter Verification code and Login

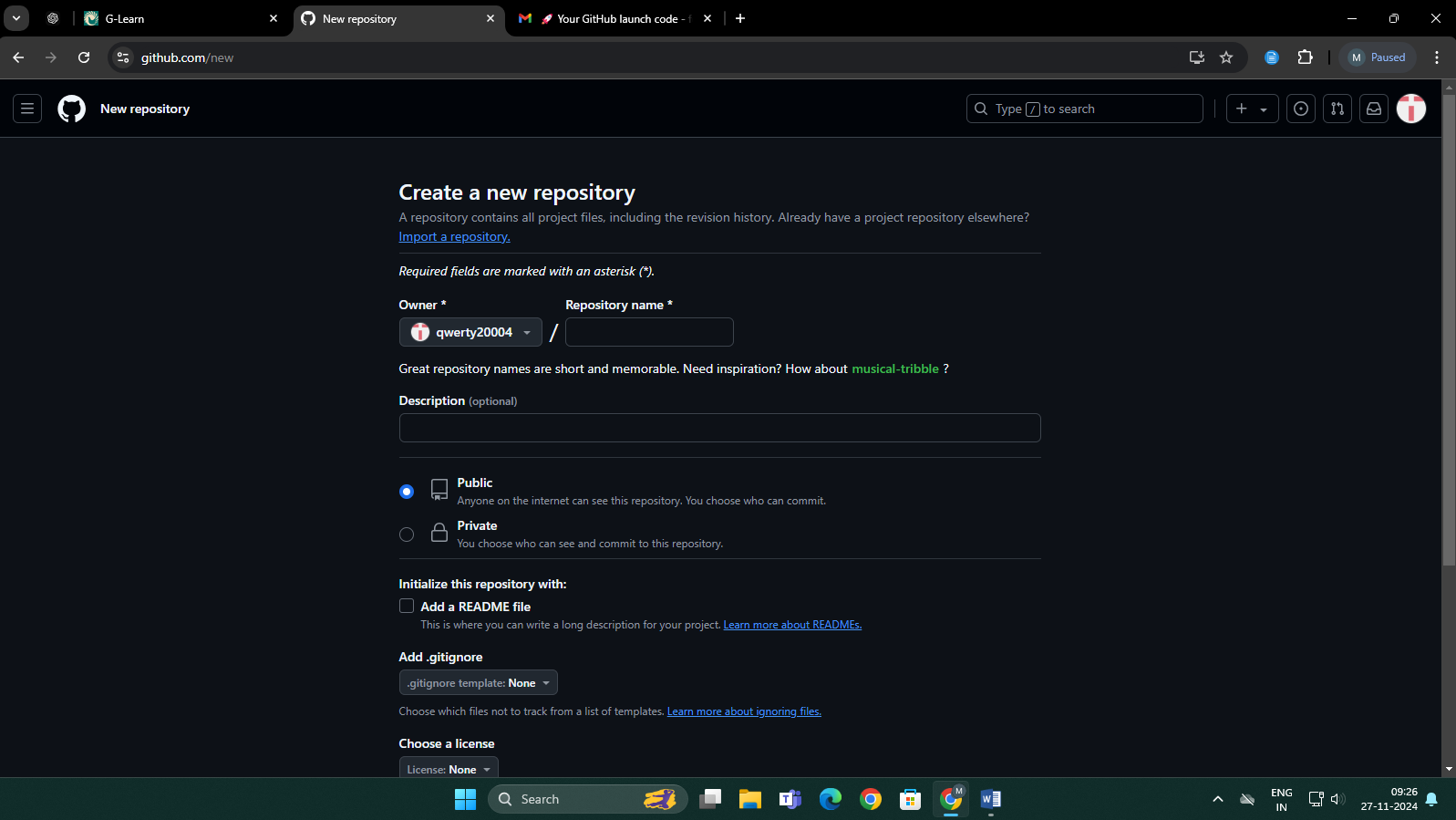




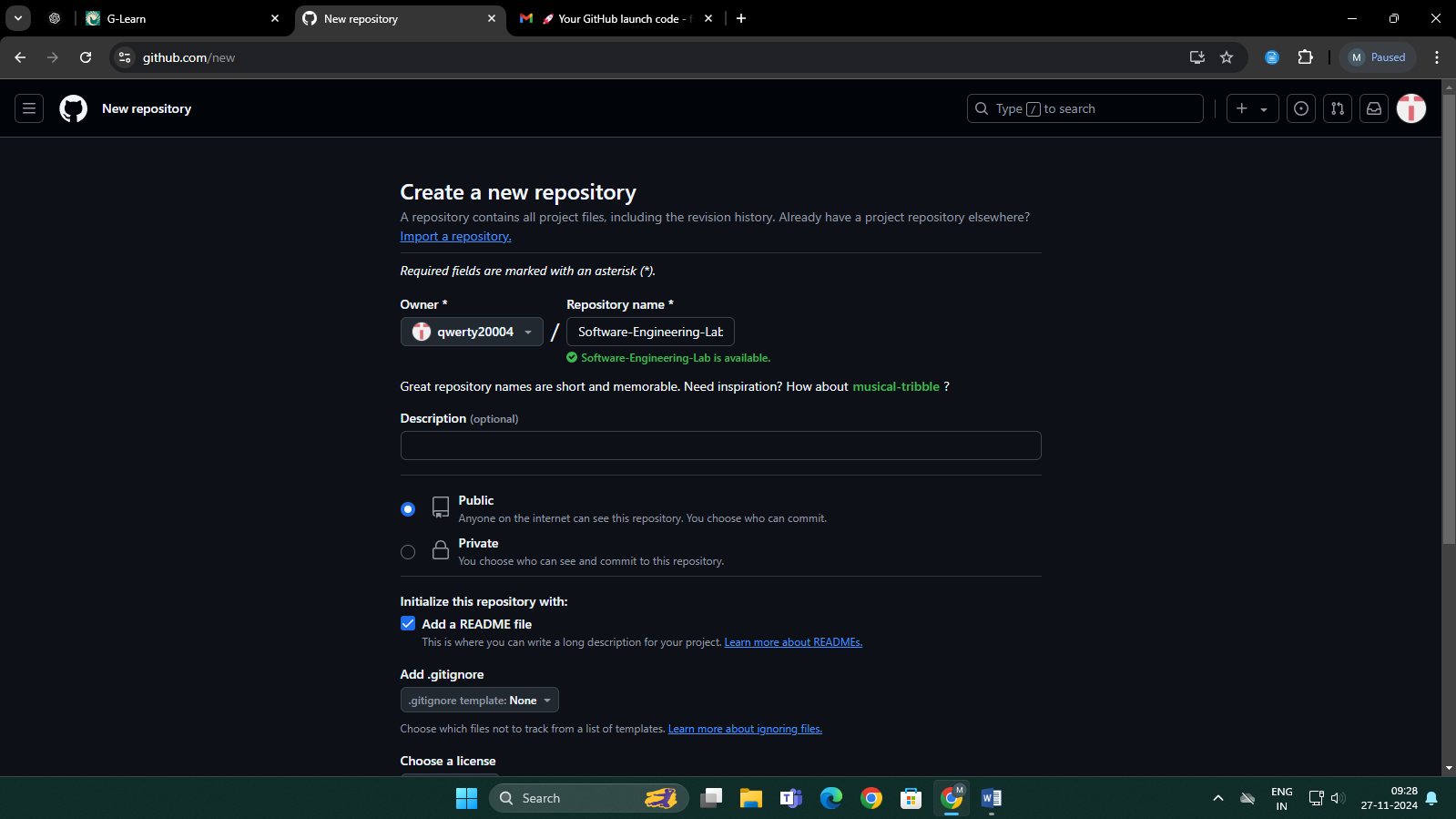
1. Enter necessary details about how you want to use GitHub and it will be redirected to Dashboard



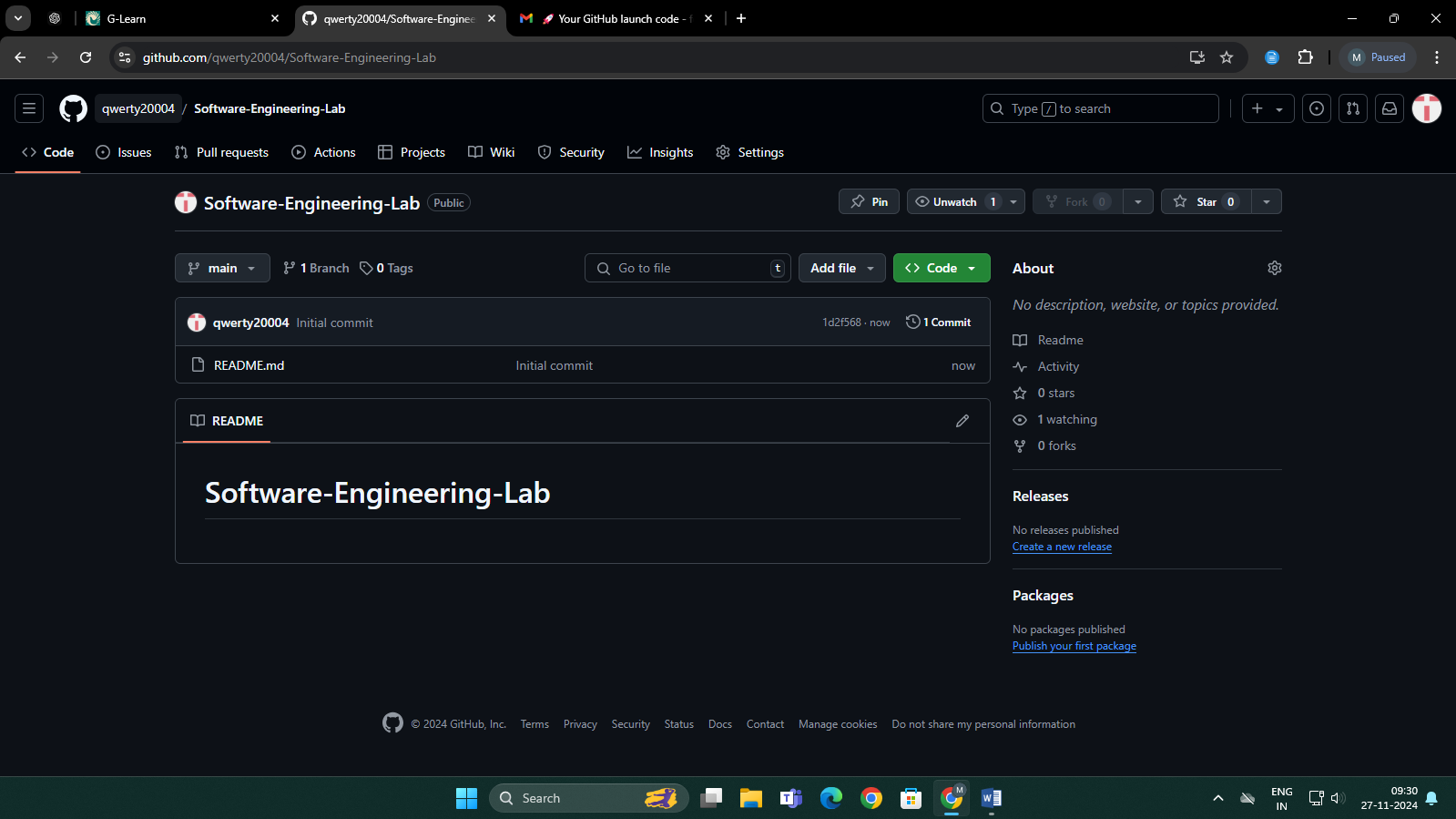
1. Click on “Create Repository” button present in the left panel



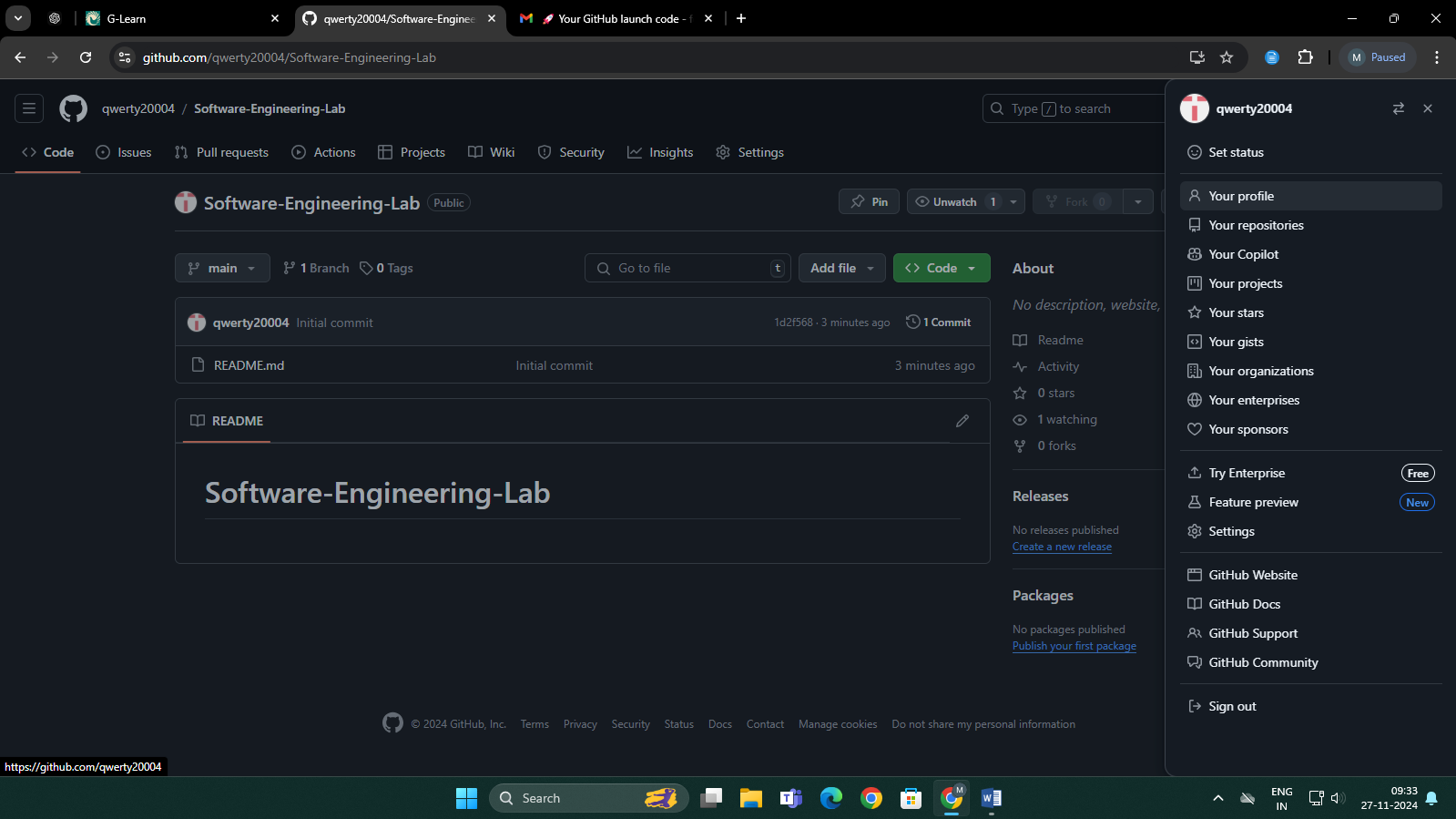
1. Enter Repository Name



1. It will be directed to Repository Dashboard



1. To change the profile image, click profile icon on top right and select profile



1. Change the profile pic

