SOFTWARE ENGINEERING TASK 1

27-11-2024

Eshwar Deshmukh Chavan

HU22CSEN0100999

What is GitHub?

GitHub is a website and platform where people can store, share, and collaborate on computer code. It's like an online "storage space" for projects, but it also helps teams work together by keeping track of changes made to the code over time.

It uses a tool called **Git** to manage these changes. Git allows you to save different versions of your code, so you can see who made changes and when. It also makes it easier to combine everyone's changes together, even if they are working on different parts of the project.

In simple terms, GitHub is like a big online notebook for programmers where they can write, edit, and share their code, and work together on coding projects with others, no matter where they are.

Introduction to GitHub

GitHub is a web-based platform primarily used for version control and collaboration in software development. It allows developers to store their code, track changes, and collaborate with others, all in one place. GitHub is built on **Git**, a popular version control system that helps developers manage different versions of their code, making it easier to work on projects as a team.

It's widely used by developers, teams, and even large organizations to create open-source projects, manage private repositories, and contribute to others' work. GitHub has become the go-to platform for software collaboration and has grown beyond just code hosting to become a hub for knowledge sharing, project management, and community engagement.

Key Features of GitHub

1. Version Control with Git:
   o GitHub uses Git to track changes in your code. Git allows you to save snapshots of your project at different points in time, so you can review, compare, or revert back to previous versions of your code.
2. Repositories (Repos):
   o A repository (or "repo") is where all the files for a project are stored, including code, documentation, and assets. Repositories can be public (anyone can see) or private (only selected people can access).
3. Collaboration Tools:
   o GitHub allows multiple developers to work on the same project. It provides features like:

- Pull Requests: Developers can propose changes (called "pull requests") to a repository. Others can review, discuss, and approve the changes before merging them into the main project.
- Forking: You can make a copy of someone else's repository (called "forking") to work on it independently, and then later propose changes back to the original project.
- Branches: You can create different versions of the project (branches) to experiment with changes without affecting the main project (called "master" or "main" branch).

4. Issues and Project Management:
   o GitHub has an integrated Issues feature, which helps track bugs, tasks, or new features in your project.
   o You can organize work by creating to-do lists, assigning tasks to team members, and adding labels to categorize issues.

5. Community and Open-Source Contributions:
   o GitHub is home to millions of open-source projects that anyone can contribute to. You can browse repositories, contribute to projects by fixing bugs or adding features, and learn from others' code.
   o Discussions and wikis allow teams and open-source contributors to communicate and share ideas.

6. Actions and Automation:
   o GitHub Actions is a feature that lets you automate workflows, like automatically testing your code, deploying your application, or sending notifications when something happens in your repository.
   o This helps streamline processes and reduce manual work.

7. Code Review and Collaboration:
   o GitHub offers tools for reviewing code through comments, inline discussions, and suggesting changes.
   o It allows teams to collaborate efficiently, with features for assigning tasks, reviewing pull requests, and merging code changes.

8. Documentation and Wikis:
   o GitHub provides features to document your project, either by adding README files directly in your repositories or creating a full Wiki to explain your project in detail.
   o This is particularly useful for open-source projects, so others know how to use and contribute to the project.

9. GitHub Pages:
   o GitHub Pages lets you host static websites directly from a GitHub repository. This is ideal for personal projects, portfolios, or documentation.

10. Security Features:
   o GitHub offers security alerts for known vulnerabilities in your code or dependencies, helping you stay on top of security risks.
   o Dependabot automatically creates pull requests to update vulnerable dependencies.

11. Insights and Analytics:
   o GitHub provides project insights, including metrics about repository activity, contributions, and community engagement, to help you understand how your project is progressing.
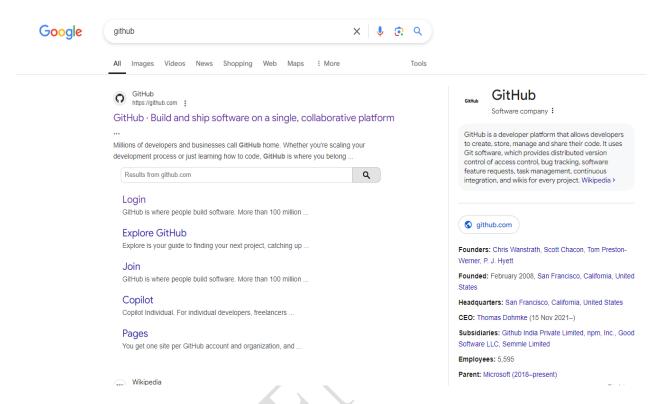
12. Integration with Other Tools:

- GitHub integrates with many other tools like CI/CD services (for automated testing and deployment), Slack (for notifications), and project management tools like Trello and Jira.

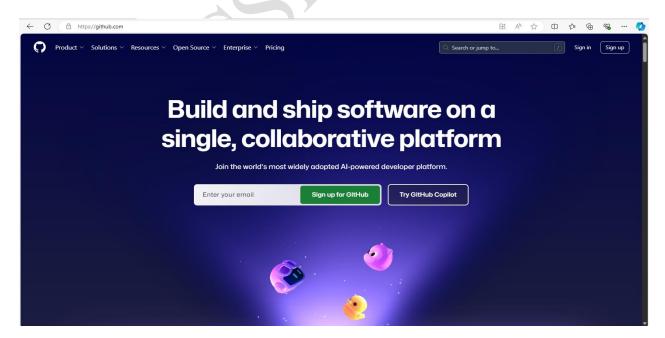## Difference between Git and GitHub

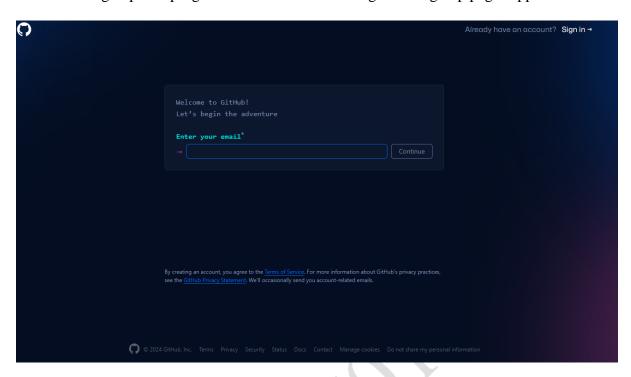| Feature	Git	GitHub
What It Is	Version control system	Cloud-based platform for hosting Git repositories
Functionality	Tracks changes, manages versions, and allows branching locally	Hosts Git repositories and adds collaboration features
Where It Runs	Locally on your computer	Online, accessible through a web browser or GitHub tools
Purpose	Manages and tracks code versions	Stores, shares, and collaborates on code remotely
Collaboration	No collaboration features (works locally)	Collaboration tools: pull requests, issues, code reviews
Access	Runs offline on your computer	Requires an internet connection to access repositories
Storage	Manages local versions of code	Stores code in the cloud, accessible from anywhere
Usage	Used for tracking changes, branching, and versioning	Used for hosting repositories, sharing code, and collaborating with others
Installation	Installed locally on your machine	No installation required; accessed via web or GitHub Desktop
Internet Requirement	Works offline	Requires internet to push, pull, and clone repositories

Getting Started with GitHub

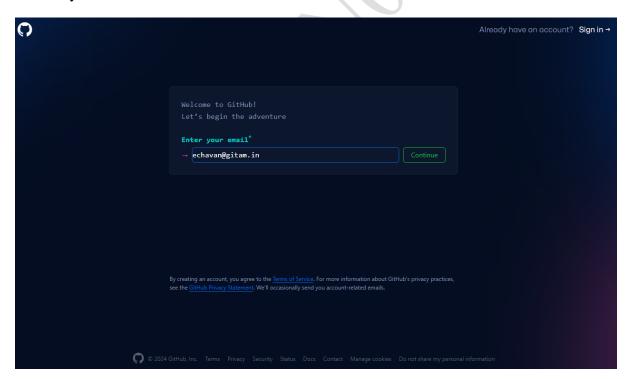1. Search "GitHub" in any browser and click enter.

2. Click on the GitHub website link which is shown in above picture then GitHub Website opens
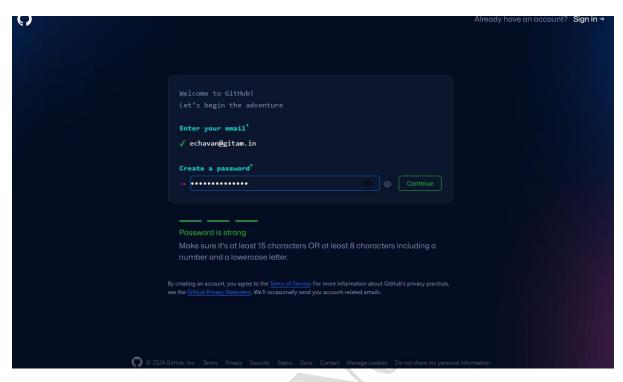
3. Click on sign up in top right corner of the above image then sign up pages appears
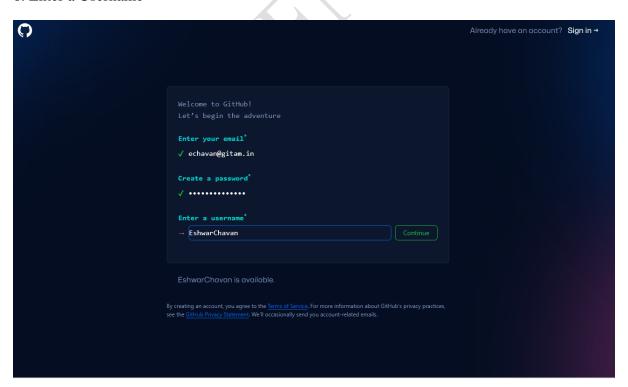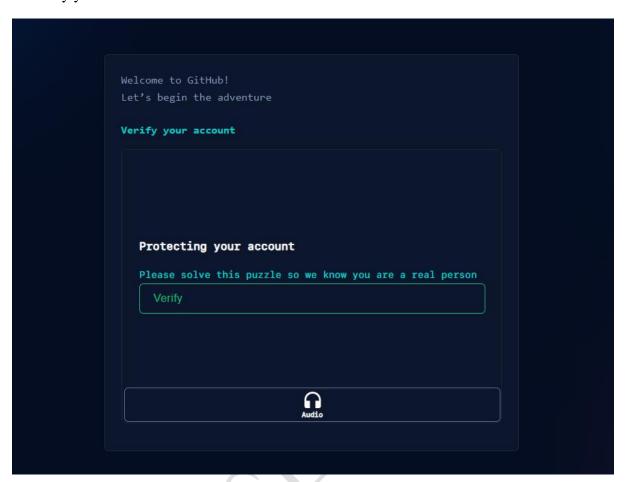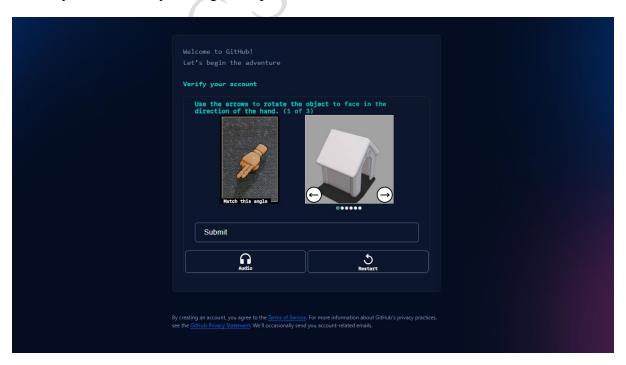

4. Enter your Email

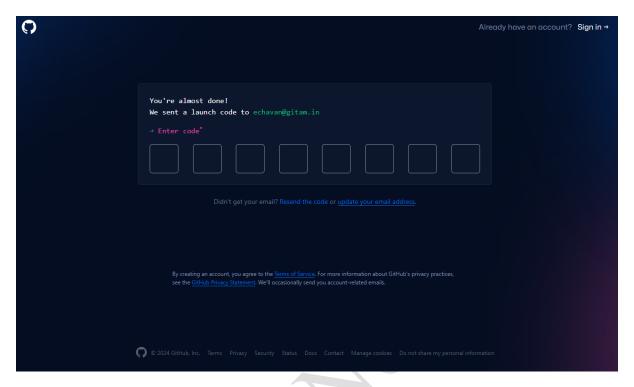## 5.Create a password and continue

6. Enter a Username

7. Verify your account
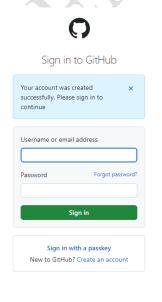

8. Verify the account by solving the captcha
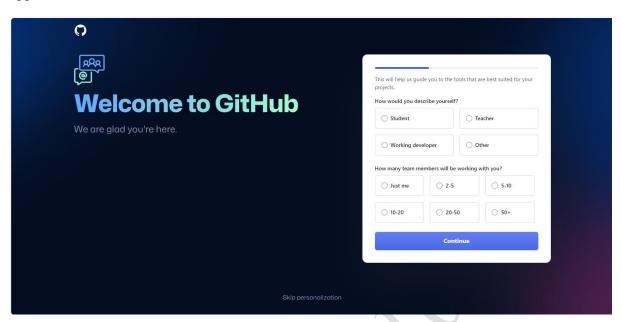
9. Enter the verification/launch code sent to registered email

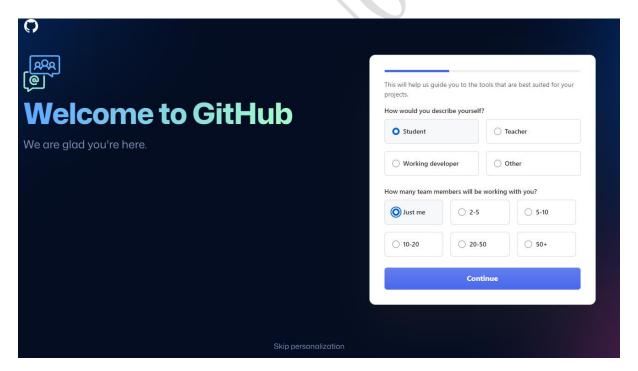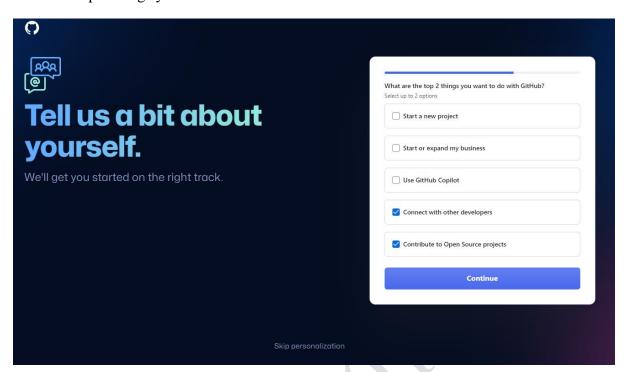10. Account is successfully created a pop up page appears

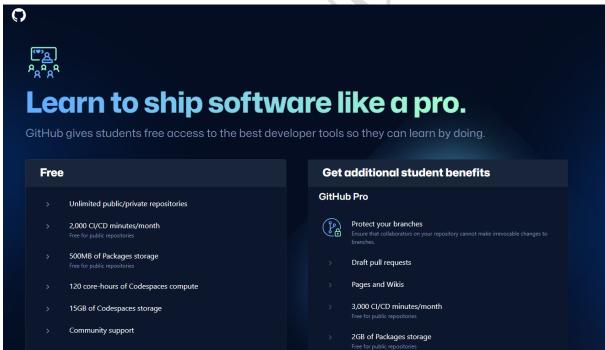11. Login in your account using Username/email and password after that a welcome page appears

12. Select the required options accordingly, here I am selecting student and just me and continue

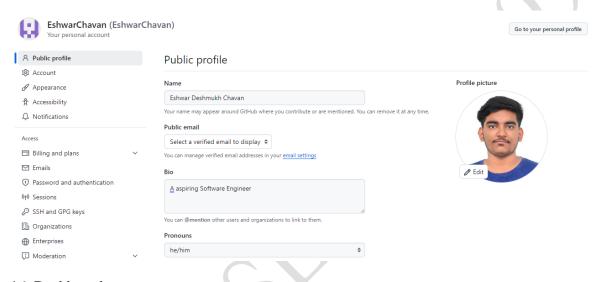13.Select top 2 things you want to do with GitHub and continue
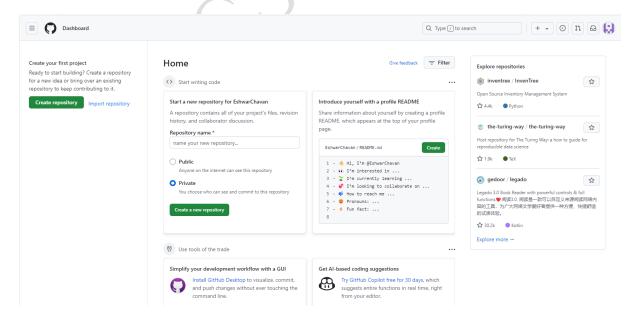

14.Click on free account and continue

GitHub Student Developer Pack

Free access to the industry's best developer tools
Hundreds of offers, including Digital Ocean, Microsoft Azure, Heroku, MongoDB, DataDog, Twilio, and Stripe.

GitHub Campus Expert training

Enrich your college technical community
Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub.

Continue for free

Apply for your GitHub student benefits

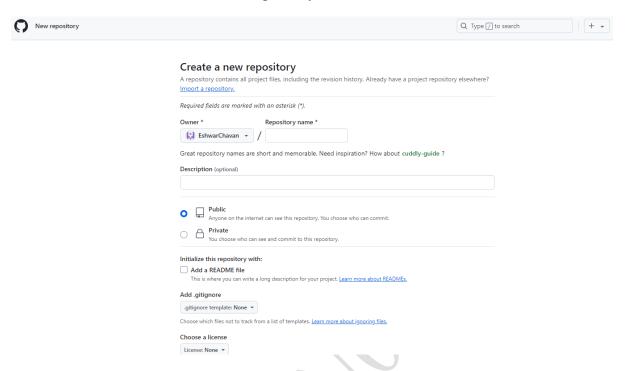## 15. Add a profile pic and fill other details to your GitHub Account
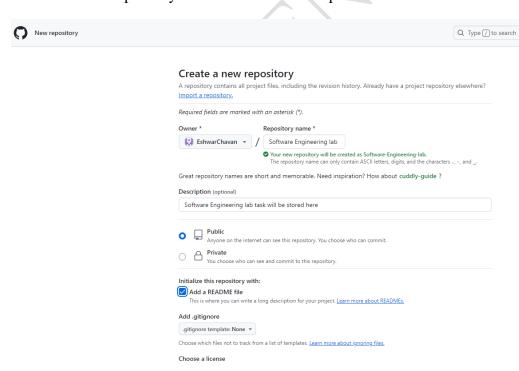

16. Dashboard appears

Steps to Create a Repository
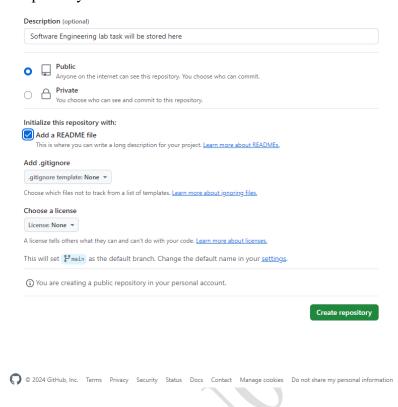
17.Click on the button Create a new Repository

18.Now Write Repository Name and write Description of it and add a README file

19.Click on Create repository button

20.Repository has been Created