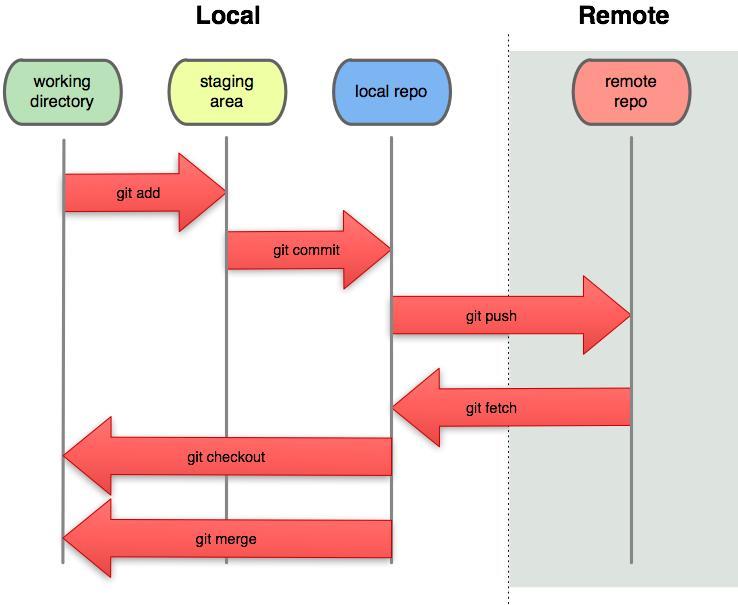
**GIT Concepts**

* **Definition:** Git is a distributed version control system designed to track changes in source code during software development.
* **Purpose:** Enables collaboration among developers, maintains history of changes, and allows for easy branching and merging.



**Basic Commands**

* Creating a Repository: git init
* Cloning a Repository: git clone <repository-url>
* Checking Status: git status
* Staging Changes: git add <file>
* Committing Changes: git commit -m "commit message"
* Viewing History: git log
* Branching: git branch <branch-name>, git checkout <branch-name>, git checkout -b new\_branch\_name
* Merging: git merge <branch-name>
* Pushing Changes: git push origin <branch-name>
* Pulling Changes: git pull

**SSH Into Git:**

Perform the following commands on local

---> cd ~/.ssh

---> ssh-keygen -o -t rsa -C "business.idcard@gmail.com"

---> cat id\_rsa.pub

---> GitHub SSH key in the settings of your online account

---> Log into GitHub

---> Navigate to your account settings

---> Go To : SSH and GPG keys

---> Click on this link to create a new GitHub SSH key

---> Provide a unique name and paste the value of the private GitHub SSH key.

**Handling Merge Conflicts:**

* Through Git Pull and Automatic Merge conflicts – show in demo class.
* How to resolve and how to push it again in real time?
* Creating a New repo for Each Batch and Add people as collaborators to contribute.

**Better Branching Strategy on Real time:**

* Production changes generally will be there on ‘main’ or ‘master’ branch.
* Whenever you start implementing new feature, will create a new branch from ‘master’ with upto date.
* Parallelly will keep few testing branches – like uat 1, uat 2..uatn (based on requirements) for testing the code changes and validating the changes for testers.
* These Uat branches will always keep up to date with the ‘master’
* So once you have created a feature branch – start working on requirements and make the necessary changes and do developer validations and create a PR for any one of the uat branch.
* Reviewer/Lead will do review the changes and approve if everything good.
* Merge the changes into uat branch and asks QA to validate your changes in that environment.
* Once QA gives signoff – Create a new PR to move the changes from feature branch to ‘master’. Merge the changes to ‘Master’ and trigger the build to reflect in production.

**Common Errors:**

$ git clone git@github.com:vinodreddem2/educard\_backend.git

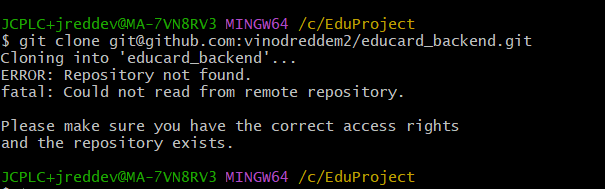
Cloning into 'educard\_backend'...

ERROR: Repository not found.

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.



This Error occurs when you are trying to clone the repo using ssh url into local, but you didn’t add your ssh key to github. So follow above steps to add ssh key into git.

Git Conflicts:

