## **Day 5 - Git Bash Commands & Concepts**

### **Basic Git Commands**

1. **Check Current Directory**
   * pwd → Prints the **present working directory**.
2. **Create & Navigate Directories**
   * mkdir <folder\_name> → Creates a new **directory**.
   * cd <path> → Navigates to a specified **directory**.

### **Git Configuration**

* git config --global user.name "Your Name"
  + Sets the **global username** for Git.
* git config --global user.email "your.email@example.com"
  + Sets the **global email** for Git.

### **Git Repository Initialization**

* git init
  + Initializes a **new Git repository** in the current directory.
* echo "text we want to save" > file\_name.txt
  + Creates a file and writes the given text into it.

### **Git Stash (Saving Local Changes Temporarily)**

* git stash
  + Saves **uncommitted** changes temporarily without committing.
* git stash list
  + Displays the list of **stashed changes**.
* git stash apply
  + Restores the **most recent stash** but **keeps it in the stash list**.
* git stash pop
  + Restores the **most recent stash** and **removes it from the stash list**.
* git stash drop stash@{0}
  + Deletes a **specific stash** (e.g., stash@{0}).
* git stash clear
  + Removes **all stashed changes**.
* git stash push -m "message"
  + Creates a **named stash** with a custom message.

### **Scenario: Handling Merge Conflicts**

**Issue:**

* Changes were made to the **main branch** from the UI.
* Local repository also has changes.
* A git pull would cause a **merge conflict**.

#### **Steps Followed**

1. **Switch to Feature Branch**
   * git checkout feature/intrest
2. **Modify File & Switch to Main**
   * vi a.txt (edit file)
   * git checkout main
3. **Merge Attempt & Check Differences**
   * git merge feature/intrest (attempt to merge changes)
   * git diff a.txt (check the differences in a.txt)
4. **Pull Changes from Remote Main**
   * git pull origin main (fetch latest changes)
5. **Add & Commit Changes**
   * git add a.txt
   * git commit -m "Changes"
6. **Push Changes to Remote Repository**
   * git push -u origin main
7. **Work on Another File (module.py)**
   * vi module.py (edit a new file)
   * git add module.py
   * git commit -m "Changes adding Jinesh"
   * git push -u origin main
8. **Merge & Final Commit**
   * git pull
   * git merge
   * git add module.py
   * git commit -m "Changes"
   * git push -u origin main

### **Key Git Commands Summary**

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a Git repository. |
| git checkout <branch> | Switch to a specific branch. |
| git merge <branch> | Merge changes from another branch. |
| git pull origin main | Fetch and merge changes from the remote main branch. |
| git add <file> | Add a file to staging. |
| git commit -m "message" | Commit staged changes. |
| git push -u origin main | Push commits to the remote repository. |
| git diff <file> | Show differences in a file. |
| git stash | Temporarily save uncommitted changes. |
| git stash apply | Restore stashed changes without removing them. |
| git stash pop | Restore and remove the stash. |
| git stash list | Show all stashed changes. |
| git stash clear | Delete all stashed changes. |