ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DIPLOMA WALLAH

UNIT 03 :- GITHUB

---

1. GitHub - Basics of Distributed Git

Definition:
GitHub is an online platform that hosts Git repositories, allowing developers to store, share, and collaborate on code. It uses **distributed version control**, so every collaborator has a local copy of the repository with full history.

Explanation:
GitHub extends Git's capabilities by providing remote hosting, collaboration tools, pull requests, issue tracking, and version control. It helps teams work together seamlessly, even from different locations, and maintain a history of changes. Developers can push local repositories, manage branches, and track project evolution.

Example:
A developer creates a new project locally, pushes it to GitHub, and team members can clone it to their machines to contribute.

Hinglish Summary:
GitHub ek online platform hai jo Git repositories host karta hai aur team collaboration easy banata hai.

Applications:

- Hosting code online
- Team collaboration
- Open-source contributions
- Tracking project history
- Managing versions and releases

Advantages:

- Distributed system support
- Easy collaboration
- Online backup of code
- Branching and pull requests
- Integration with CI/CD tools

Disadvantages:

- Internet required for remote operations
- Can be complex for beginners
- Large repositories can be slow
- Public repositories are visible (unless private)
- Merge conflicts may occur in team projects

---

2. Account Creation and Configuration on GitHub

Theory:

- To use GitHub, you must create an account using an email address.
- After account creation, GitHub provides a remote repository platform for storing your projects.
- Configure Git locally with GitHub credentials to push and pull repositories.

Practical Steps:

1. Go to https://github.com and click **Sign Up**.
2. Enter **username**, **email**, and **password**.
3. Verify email to activate the account.
4. Configure Git locally with:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

5. Generate SSH key for secure communication:

```
ssh-keygen -t ed25519 -C "your.email@example.com"
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
cat ~/.ssh/id_ed25519.pub
```

- Copy the public key to GitHub **SSH keys** settings.

Example:
After configuration, you can push local changes to GitHub without entering credentials every time.

Hinglish Summary:
GitHub account banake aur local Git configure karke aap apne projects remote repository par host kar sakte hain.

---

3. Create and Push a Repository (Practical + Theory)

Theory:

- A repository is a project folder tracked by Git, and on GitHub, it is hosted remotely.
- Pushing a repository uploads your local files and commit history to GitHub.

Practical Steps:

1. **Create a repository on GitHub**:
    - Click **New Repository**
    - Name: mini-project
    - Add description (optional)
    - Choose **public** or **private**
    - Click **Create repository**
2. **Create local repository:**

mkdir mini-project

cd mini-project

git init

3. **Add files and commit:**

echo "# Mini Project" >> README.md

git add README.md

git commit -m "Initial commit"

4. **Connect local repo to GitHub:**

git remote add origin https://github.com/username/mini-project.git

5. **Push to GitHub:**

git push -u origin main

Example:
Your local project is now on GitHub at https://github.com/username/mini-project. Team members can clone it using:

git clone https://github.com/username/mini-project.git

Hinglish Summary:
Local repository banakar aur commit karke GitHub par push karna se aapka project online host ho jata hai aur team ke liye accessible hota hai.

---

4. Versioning on GitHub

Definition:
Versioning means keeping track of all changes in a project so you can revert to previous states or compare different versions.

Explanation:
Each commit in GitHub acts as a version snapshot. You can view commit history, check differences between versions, and restore older versions if needed. GitHub also allows tagging specific commits as releases for version management.

Example:

- Commit v1: Basic homepage created
- Commit v2: Added login module
- Tag release v1.0 after testing

Hinglish Summary:
Versioning se project ke saare changes track hote hain aur older versions restore kiye ja sakte hain.

---

5. Collaboration on GitHub

Theory:

- GitHub allows multiple developers to work on the same project.
- Features include **forking**, **pull requests**, **branching**, **code review**, and **issues**.
- Pull requests allow collaborators to review and merge changes safely into the main branch.

Practical Example:

1. Developer A creates a branch feature-login
2. Makes changes and pushes branch:

git push origin feature-login

3. Opens a **Pull Request** on GitHub

4. Team reviews and merges into main

Hinglish Summary:
GitHub par multiple developers safely code share, review aur merge kar sakte hain.

---

6. Migration to GitHub

Definition:
Migration means moving an existing project from local storage or another VCS to GitHub.

Practical Steps:

1. Initialize local Git repository:

git init

2. Add files and commit:

git add .

git commit -m "Initial commit"

3. Create repository on GitHub
4. Connect remote and push:

git remote add origin <GitHub-repo-URL>

git push -u origin main

Example:

- Local college project migrated to GitHub for online backup and collaboration.

Hinglish Summary:
Existing project ko GitHub par migrate karke aap online host aur team collaboration enable karte hain.

---

7. Practical Mini Project Example

Steps:

1. Create folder: mini-project
2. Initialize Git and add README.md
3. Commit initial version
4. Create repository on GitHub
5. Connect remote and push changes

6. Create a new branch feature-login
7. Make changes, commit, push, open pull request
8. Merge feature branch into main

Hinglish Summary:
Mini project create karke GitHub par push, branch aur merge karke collaboration workflow practice kiya.

Diploma Wallah

Made with 💗 by Sagar Sangam