ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DAY – 4

Date: June 26, 2025

Git and Git Commands

What is Git?

Git is the free and open-source distributed version control systems that's responsible for everything GitHub related that happens locally on your computer.

Understanding Version Control: Version control, also known as source control, is the technique of tracking and managing changes to codes and these are the systems that are software tools that enable software teams to manage modifications to source code as time passes.

What is GitHub?

GitHub is a widely-used Free-to-use cloud Storage platform with version control and many other essential features that specifically helps developers to manage and deploy their projects on GitHub.

Basic Git Commands

- **git init**: Initializes a new Git repository and begins tracking an existing directory. It creates a new subdirectory named. git that houses all your necessary repository files — a Git repository skeleton. Use this in the root of your project to start using Git.

  ```
  git init
  ```

- **git clone [URL]**: Creates a copy of an existing Git repository. Use this to download and work on a project that has already been started.

  ```
  git clone https://github.com/username/repository.git
  ```

- **git add [file]**: Adds a file to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you commit.

  ```
  git add README.md
  ```

- **git commit -m "[descriptive message]"**: Records file snapshots permanently in the version history. The -m flag lets you add a commit message on the command line.

  ```
  git commit -m "Add initial project version"
  ```

- **git status**: Displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

  ```
  git status
  ```

- **git push [alias] [branch]:** Transmits local branch commits to the remote repository branch. Alias is the name of the remote and branch is the branch you want to push.

  ```
  git push origin master
  ```

- **git pull:** Fetches and merges any commits from the tracking remote branch.

  ```
  git pull origin master
  ```

MYCIN EXPERT SYSTEM

MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections and to recommend antibiotics, with the dosage adjusted for patient's body weight.

MYCIN was developed over five or six years in the early 1970s at Stanford University.

It was written in Lisp

MYCIN operated using fairly simple inference engine, and a knowledge base of approximately 600 rules.

MYCIN is a computer program designed to provide attending physicians with advice comparable to that which they would otherwise get from a consulting physician specializing in bacteremia and meningitis infections. To use MYCIN, the attending physician must sit in front of a computer terminal that is connected to a DEC-20 (one of Digital Equipment Corporation's mainframe computers) where the MYCIN program is stored. When the MYCIN program is evoked, it initiates a dialogue. The physician types answers in response to various questions. Eventually MYCIN provides a diagnosis and a detailed drug therapy recommendation.