

**NAME: ALI FAROOQ,
MUHAMMAD ALI REHMAN**

RED NO. 2023094, 2023372

COURSRE CODE: ES-221

**COURSE NAME:
DATA STRUCTURE AND
ALGORITHM**

Initial Project Report: GIT-Lite represents a simplified version control system built from Git principles.

Project Title: The Data Structure Design and Implementation Phase of GIT-Lite is the initial focus of this report.

Team Members:

ALI FAROOQ 2023094

MUHAMMAD ALI REHMAN 2023372

1. Introduction:

The first development phase of GIT-Lite tracks the initial stages of creating a basic version control system derived from Git's functionality. The goal of GIT-Lite is to provide an educational version control tool which aids students and developers in understanding essential version control concepts and core data structures.

2. Purpose of the Project:

Engineers need to develop core version control operations such as commit and branch and checkout and merge by creating them from fundamental elements. The project needs to develop suitable data structures incorporating DAGs, hash maps and trees to optimize version tracking capabilities. The tool delivers straightforward operational capabilities which replicate Git functionality basics. The project aims to improve understanding of internal systems used by tools like Git when they handle file content modifications along with their version history maintenance. The application functions as an educational aid for student use which demonstrates version control principles through hands-on interaction.

3. Project Description:

GIT-Lite uses a design structure which duplicates Git's essential features through standalone code instead of Git libraries. The tool will include: The system creates objects which store both metadata and snapshots of changes made to the repository. The directory-based file tracker implements both content indexing together with hashing functionality. Through Directed Acyclic Graphs (DAGs) the system enables users to explore history navigation as well as branching operations. The system allows users to perform staging area and repository simulation to execute add and remove file operations before making a commit. Through a command-line interface the system offers basic Git-like operation capabilities to users.

4. Key Components & Data Structures:

Multiple tailor-made data structures form the backbone of this system. The system employs Blobs to maintain file content storage. One key data structure of the system is trees which serve to organize directory structures along with their hierarchy. The system stores commit with commit metadata and timestamps for each entry and parent commit data and user-provided notes. The Index serves as a staging area which tracks all modified files that users plan to add or remove from their project. The system uses references known as branches to track active commit positions along with HEAD for reference management.

5. Expected Outcomes By the end of the project we hope to achieve:

- A working CLI tool for simple version control operations.
- Simple, clear and module ready codebase with emphasis on readability and readability for learning.
- A tech report explaining design rationals, data structure, and implementation details.
- A demo session which will illustrate how GITLite can be applied to manage a real-world project.

6. Timeline Overview:

- Phase 1: Requirement analysis, data structure design.

- Phase 2: Project scope usability wrapper core feature implementation (init, add, commit).
- Phase 3: Feature Advanced (branching, checkout, merge).
- Phase 4: Testing, Documentation, Final Demo.

7. Tools and Technologies:

- Programming Language: [e.g., Python/C++/Java].
- Version Control for Development: Git.
- Text Editors/IDEs: VS Code, IntelliJ, etc.