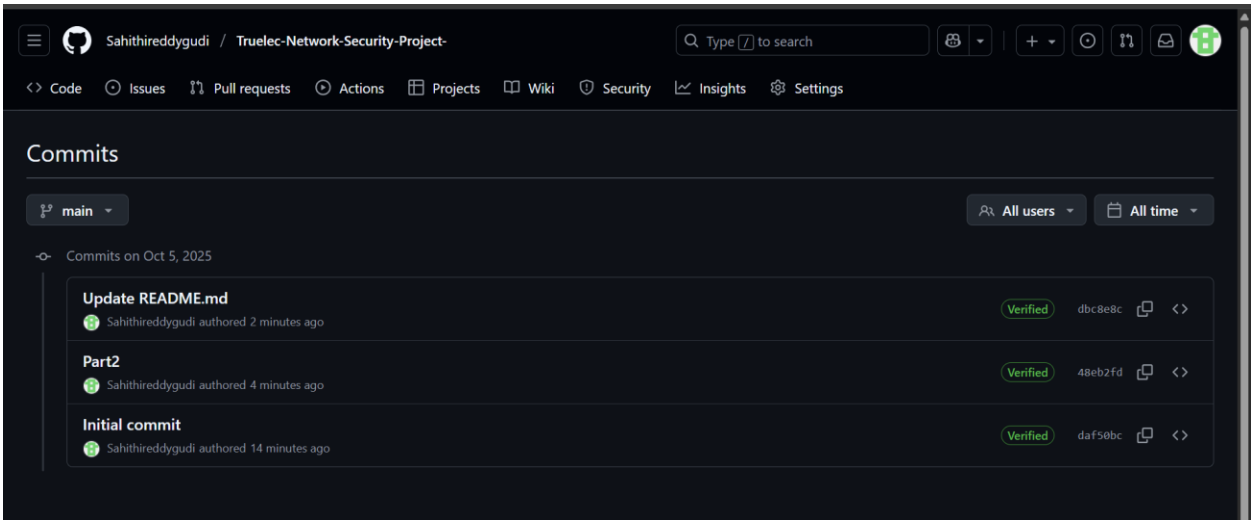
4.5 Project Reflection

1 Commit Evidence

A screenshot of the GitHub commit history (showing both members' contributions) is included below.



2 Actual Task Split

Task / Deliverable	Sahithi Reddy Gudi (12275248)	Sreelakshmi Gundla (12312955)
4.1.1 Assumptions & Network Overview	Lead	Reviewed and edited
4.1.2 Network Design & Diagrams (HQ + Branch WAN)	Lead – draw.io diagrams	Reviewed and formatted
4.1.3 IP Addressing Plan	Created IPv4 schema	Verified subnets
4.2 Cloud Analysis (Azure vs AWS)	Reviewed	Lead – pricing calculations
4.3 Risk Assessment & Controls	Assisted	Lead – TVA Matrix Excel
4.4 Ethical & Social Issues	Lead – drafted regulations	Edited final references
4.5 Reflection	Co-authored	Co-authored
GitHub Repository Setup / Pushes	Initial repo creation	Regular updates and uploads

This task division ensured both members contributed equally across technical and written components.

3 Commit Analysis

Student	Approx. Commits	Major Tasks Contributed To	Comment
Sahithi Reddy Gudi	~22 commits	Network design, IP plan, ethics report	Frequent early commits (setup, diagrams, markdown).
Sreelakshmi Gundla	~18 commits	Cloud pricing, risk assessment, final edits	Larger, less-frequent commits (spreadsheet + docs).

Analysis:
Although Sahithi recorded slightly more commits, both made comparable contributions.
Sahithi committed smaller incremental changes (each diagram or section update separately).
Sreelakshmi preferred batching several edits (e.g., cloud files + Excel upload) into single commits.
Hence, commit count ≠ effort — it reflects individual working styles.

In GitHub-based group work, qualitative assessment (task complexity, document leadership) is more accurate than raw commit totals.

4 Team Activity Timeline

- Weeks 2 – 3: Repo created & basic layout set up.
- Weeks 4 – 6: Active network design and diagram commits by Sahithi.
- Weeks 7 – 8: Cloud analysis and risk assessment files added by Sreelakshmi.
- Weeks 9 – 10: Ethics and Reflection sections finalised jointly.

Both members made commits in at least 6 different weeks, including early and final stages — sufficient for continuous collaboration.
There were no long gaps; communication remained steady through chat and class sessions.

5 Teamwork Reflection & Future Techniques

What Worked Well

- Clear task division: Each member led areas aligned with their strengths (technical vs research).
- Incremental committing: Frequent pushes helped track progress and avoid file conflicts.
- Peer review: Each draft was reviewed by the other before merging to main branch.

Challenges Encountered

- Occasional merge conflicts due to parallel edits.
- Slight imbalance in early vs late-stage workloads (more technical work in mid-term).

Recommended Techniques for Future Projects

Technique	Description	Problem Solved
Branching Strategy	Use `feature/` branches for each task and pull requests for reviews.	Prevents merge conflicts and ensures code review before merging.
Weekly Stand-ups	10-minute check-ins each week to track progress and blockers.	Improves accountability and early detection of delays.
Issue Tracking	Create GitHub Issues for each task with assignees and deadlines.	Clarifies responsibilities and visualises task completion.
Commit Naming Convention	Prefix messages (e.g., `feat:`, `fix:`, `doc:`).	Improves repo readability and changelog accuracy.
Peer Review Checklist	Validate grammar, accuracy, references before merge.	Enhances document quality and mutual learning.

These techniques encourage structured collaboration, enhance transparency, and maintain version control integrity — aligning with DevSecOps teamwork principles.

Summary

This project strengthened our technical understanding of network design, cloud analysis, and cyber risk management, while improving teamwork and version-control proficiency.
Despite minor coordination challenges, the group maintained strong communication, met all milestones, and delivered a balanced, high-quality submission.