

PROJECT ABSTRACT

Text Diff and Merge Tool with Side-by-Side View

A browser-based text diff and merge tool that compares two versions of any text with a split side-by-side view, a unified view, word-level change highlighting and block-by-block merge controls — delivered as a single offline HTML file with a live demo, report, PPT and viva kit.

01. Title

Text Diff and Merge Tool with Side-by-Side View

02. Introduction / Background

Every software workflow depends on comparing text: reviewing pull requests, auditing configuration changes before deployment, reconciling two edited copies of a document, or checking what changed between software releases. Command-line diff tools have served this need for decades, but their output is hard for students to read and their merge workflows are intimidating. Visual diff tools solve the readability problem, yet most are heavyweight desktop applications or tied to a specific version-control system. This project builds DiffLens, a lightweight diff and merge tool that runs entirely in a web browser as a single HTML file: paste two texts and it shows exactly what changed, line by line and word by word, then lets you merge the two versions block by block.

03. Problem Statement

Students and junior developers regularly need to answer the question “what changed between these two versions?” — but plain text editors cannot show differences, command-line diff output is cryptic, and full IDE merge tools are heavyweight and tied to version control. There is no simple, offline, explainable tool that (a) visualizes differences at both line and word level, (b) offers both side-by-side and unified presentations, and (c) lets a user resolve each changed block individually into a merged result. This project fills that gap with a self-contained, zero-install web application.

04. Objectives

- Implement a line-level text comparison engine using the longest-common-subsequence algorithm, with word/token-level refinement inside changed lines.
- Provide a split side-by-side view with row alignment and per-line numbering, plus a unified +/- view with collapsible unchanged regions.
- Turn every changed block into a merge conflict card with keep-original, keep-revised and take-both resolution actions.
- Generate a live merged-output preview with standard conflict markers for unresolved blocks, exportable via copy-to-clipboard and .txt download.
- Ship the entire tool as one offline HTML file with no backend, plus three realistic sample comparison pairs for demonstration.

05. Proposed Methodology

The comparison core is a two-stage algorithm. First, both texts are split into lines and compared with a line-level longest-common-subsequence (LCS) dynamic-programming pass ($O(n \cdot m)$ time and memory, easily fast enough for files of a few hundred lines). Optional normalization — collapsing whitespace or folding case — is applied before comparison so reformatted or case-shifted text can be compared semantically. The LCS backtrack is grouped into equal runs and changed blocks. Second, within each changed block, paired lines are tokenized into words and compared with a smaller LCS pass, marking exactly the added, removed and altered tokens for inline highlighting. Rendering is pure DOM: the split view pads the shorter side of each block so rows stay aligned; the unified view emits $-/+$ rows with @@ headers and collapses long unchanged runs behind click-to-expand gaps. Merge state is a per-block choice (left/right/both/null); the merged output concatenates unchanged lines with each block's chosen content, emitting standard <<<<<< / ===== / >>>>>> markers for blocks left unresolved.

06. System Architecture / Working

The application is a single HTML file with embedded CSS and JavaScript — no build step, no backend, no dependencies. The data flow is: (1) two textareas hold the original and revised texts (editable live, or loaded from one of three sample pairs); (2) the diff engine recomputes on every input change (debounced) and produces grouped hunks; (3) the active view — split, unified or merge — renders the hunks; (4) in the merge view, per-block resolution buttons update the merge state and the merged-output preview re-renders instantly; (5) the result is exported through the Clipboard API or a Blob download. A statistics bar (additions, deletions, unchanged lines, changed blocks) updates with every keystroke. Because everything runs client-side, pasted text never leaves the user's machine.

07. Hardware / Software Requirements

- Software: any modern web browser (Chrome, Edge, Firefox, Safari); no server, build tool or runtime required.
- Development: a text editor; the app is plain HTML/CSS/JavaScript with no framework.
- Hardware: any laptop or desktop capable of running a modern browser.
- Optional: Python 3 + Pillow only for the project's own screenshot-generation step, not for the delivered app.

08. Expected Outcomes

- A working single-file diff and merge web app demonstrating the split view, unified view and block-by-block merge on three realistic sample pairs.
- A documented, commented implementation of line-level LCS diff with word-level refinement that the student can explain line by line in a viva.
- Live diff statistics, compare options (ignore whitespace/case, context size) and merged-result export via clipboard and .txt download.
- A project report, review PPT and viva Q&A; covering diff algorithms, LCS complexity and merge strategies.

09. Applications

- Reviewing code or config changes before deployment (e.g. staging vs production nginx configs).
- Classroom teaching of diff algorithms, LCS and merge-conflict concepts with a live visual demo.
- Reconciling two edited copies of documents, changelogs or data files.
- A starting point for a file-upload comparison feature (listed as future scope).

10. Future Scope

- File upload and drag-and-drop comparison via the FileReader API, plus multi-file folder comparison.
- Three-way merge with a common ancestor, and syntax-aware diffing (moved-block detection).
- Performance optimization (Myers $O(ND)$ algorithm, virtualized rendering) for files with tens of thousands of lines.
- Shareable diff links and dark mode.