

PROJECT ABSTRACT

Markdown Blog Engine with Static Site Export

A Markdown-powered blogging platform with a split-pane editor, live rendered preview, a published blog front end, post management and one-click static HTML export. Delivered built-to-order with working demo, source code, report, PPT and viva kit.

01. Title

Markdown Blog Engine with Static Site Export

02. Introduction / Background

Developers love writing in Markdown and readers love fast static pages; a Markdown blog engine connects the two. This project builds MarkPress: a split-pane editor where Markdown source renders live into styled HTML (headings, code blocks, blockquotes, lists), a published blog front end with post cards, categories and reading metadata, a management view listing posts with word counts and publish status, and a static-export flow that explains — and simulates — generating pure HTML files with no database or server code. The demo ships with four realistic technical posts so every view has genuine content.

03. Problem Statement

Blog-engine projects often stop at a styled form with no real publishing pipeline. This project addresses that by implementing the complete authoring loop — write Markdown, see it rendered, manage the post inventory, export static HTML — so the student can explain parsing, templating and static generation as one coherent system at the viva.

04. Objectives

- Render Markdown source into styled HTML in a live split-pane editor.
- Present a published blog front end with cards, categories and metadata.
- Manage posts with status, word counts and update dates.
- Document and demonstrate the static-site export pipeline.
- Ship with realistic technical content across four posts.

05. Proposed Methodology

The engine is built in three stages. First, the content model: posts with title, category, body excerpt, author, date and read time, plus full Markdown source for the draft.

Second, the editor view: a two-pane layout pairing the raw Markdown with its rendered HTML, covering headings, bold/italic, lists, code spans, fenced blocks and blockquotes. Third, the publishing views: the public blog grid, the management table with publish/draft states, and the export panel describing the three-step static build.

06. System Architecture / Working

- Content layer: post objects with metadata and Markdown bodies.
- Editor layer: split panes showing source and rendered output side by side.
- Public layer: blog grid rendering cards from published posts.
- Management layer: post table with status and the export pipeline explainer.
- Export concept: Markdown plus theme templates compile to static HTML files.

07. Software Requirements

- Any modern web browser; no build step or dependencies.
- Static hosting — the exported site needs nothing else.
- Text editor for content and code changes.
- Optional: a Markdown parser library if the student rebuilds rendering dynamically.

08. Expected Outcomes

- A working split-pane Markdown editor with faithful rendered preview.
- A complete demo blog with four realistic technical articles.
- Post management with publish/draft workflow.
- A documented static-export pipeline the student can explain end to end.

09. Applications

- Personal technical blogging without a CMS or database.
- Teaching example for Markdown parsing and static site generation.
- Documentation sites for student projects.
- Portfolio piece showing editorial UI design.

10. Future Scope

- Real Markdown-to-HTML build script generating the static files.
- Theme system with multiple templates.
- Tags, search and RSS feed generation.
- Git-backed publishing workflow.