



Chat MD

Project Proposal: Chat MD

Executive Summary

We propose integrating a zero-cost, AI-driven microservice to automatically format session descriptions. This will standardize our website content, eliminate manual "cleanup" work for developers, and help Heads create professional posts without learning Markdown.

The Problem

The "Formatting Bottleneck"

Currently, our content workflow is inefficient:

1. **Skill Gap:** Heads of Committees are not developers. They write descriptions in plain text paragraphs, unaware of the Markdown formatting required by our website.
2. **Inconsistent UX:** This results in "walls of text" on the live site that are hard for students to read.
3. **Manual Toil:** The technical team currently wastes time manually editing these descriptions to add headers, bold text, and bullet points before deployment.

The Solution

"Format with Chat MD" Button

We will add a single button to the description editor. When clicked, it sends the raw text to a secured AI endpoint which returns perfectly structured Markdown.

The "Before & After" Experience:

- **Input (User):**

- Session Name : MLops
- Topic We Discussed : Why Mlops, DVC, MLflow, Training pipeline, Deployment Docker
- Resourecees for it : [Name] [Link of resources]

- **Output (AI):**

MLOps: Building Robust AI Systems

Welcome to our deep dive into **MLOps**! In this session, we explored the critical intersection of Machine Learning and DevOps, learning how to move models from experimental notebooks to reliable production environments.



Key Topics

- **Why MLOps:** Understanding the need for reproducible and scalable machine learning workflows.
- **DVC (Data Version Control):** Managing large datasets and model versions just like code.
- **MLflow:** Tracking experiments and metrics to ensure the best model is selected.
- **Training Pipeline:** Automating the end-to-end process from data ingestion to model training.
- **Deployment with Docker:** Containerizing applications to ensure they run consistently anywhere.



Resources

- [DVC Documentation](#)

3. Technical Architecture

We will use a **Serverless Sidecar** pattern to keep this secure and free.

- **Backend:** Fast API to deploy the project
- **Model:** Google Gemini 1.5 Flash – *Chosen for speed and free tier limits*

⚠️ Security Note: No student data is passed to the AI. Only the generic session topic and description text are processed.

4. Cost Analysis (Zero-Cost Strategy)

This project leverages the "Free Tier" of enterprise-grade tools.

Component	Provider	Tier Limit	Our Usage	Cost
AI API	Google Gemini	15 RPM (Requests Per Min)	~5 RPM	\$0.00
Hosting	Aws Fly.io	5\$	N/A	\$0.00
TOTAL				\$0.00

5. Implementation Plan (Timeline: 3 Days)

Day 1: Backend Development

- Generate the Ai API from Google Cloud.
- Write the API End([api/format](#)).
- Implement the System Prompt (The "Brain").
- Test security (ensure keys aren't exposed).

Day 2: Frontend Integration

- Add "Magic Wand" button to the UI.
- Connect button to the Backend API.

- Final UAT (User Acceptance Testing).
-

6. 🏆 Success Metrics

We will consider this project a success if:

1. **100%** of new session descriptions are Markdown-formatted without developer intervention.
 2. **Zero** complaints from Heads about "breaking the website."
 3. **< 2 seconds** latency for the formatting action.
-