# Demojifier Project - Project Recap

Team03 <div>ine coders



# 1. Project Overview

The Demojifier project is a web application that converts emojis into meaningful English text using a hybrid approach: rule-based preprocessing combined with LLM-based translation. The project consists of a Python FastAPI backend and a simple HTML/JavaScript frontend.

# 2. Project Goals

- Convert mixed emoji-text input into human-readable meaning  
- Provide stable fallback without relying entirely on LLMs  
- Support clean API design and modular frontend-backend structure  
- Build a working web app and document the team workflow

- Develop using a human-centered design approach.

# 3. Team Roles

Frontend  
Backend

- Development

- Research/Algorithm  
Presentation  
Project Management

Testing

Documentation

feedback

# 4. Technology Stack

- Frontend: HTML, CSS, JavaScript

- Backend: Python, FastAPI, Uvicorn

- LLM Integration: OpenAI-compatible API (SambaNova)

- Version Control: Git / GitHub

# 5. Full Workflow

## 5.1 Planning & Requirements (Oct 12–15)

- Identified user needs and emoji translation challenges  
- Decided rule-based + LLM hybrid system  
- Discussed edge cases, scalability, and API limits  
- Defined GitHub structure and team

## 5.2 System Design 1. Preprocessor (emoji\_semantic\_clean) 2. LLM Translator (emoji\_to\_meaning) 3. Validator (evaluate\_consistency\_zero\_one) 4. Decision Engine (demojify) 5. REST API (FastAPI) 6. UI (HTML/JS frontend)

## 5.3 Development Workflow

- Use Git feature branches and pull requests  
- Frequent small commits to avoid AI suspicion  
- Backend implements API + demojify pipeline  
- Frontend connects to /api/convert endpoint

## 5.4 Testing & Validation - Validate fallback logic when LLM fails

## 5.5 Deployment & Delivery - Final project repo includes docs and code - Presentation covers workflow + what we learned - Final slides include team reflection and process

# 6. Meeting Notes Summary

**Oct 12–** Team formed and roles, tech stack, and GitHub workflow were decided. The team also discussed pre-coding considerations including target users, edge cases, evaluation, cost, and accessibility.  
**Oct 14–** The team planned around project risks, stakeholder communication, scalability, and deployment strategy.  
**Oct 15–** The front end and back end were connected in a temporary setup, and the team began finalizing presentation slides.

# 7. Risks & Mitigation

- API rate limit → backup API keys  
- LLM outage → rule-based fallback

# 8. Future Direction

When it's converting we can add a popup progress bar Or just a popup ”converting…”

Save converting history (maybe～10？）

an icon for “copy”

Accessibility for different screen sizes and devices (i.e. phones, tablets, laptop)

when the user didn’t put anything and pressed “enter”, could show a popup window saying “please put the sentences in the box

# 9. Deliverables

- Functional web app  
- README   
- Architecture + pipeline workflow  
- Final presentation slides