# **Project Documentation — StagingUpload (LLM)**

## **1. Project Overview**

StagingUpload\_Final is a Python + PHP project designed to simplify and automate staging uploads for Excel files.  
Traditionally, preparing staging scripts manually can take 3+ hours per partner, but this project reduces it to just 2–5 minutes using:  
- LLM-powered automation (Gemini via LangChain)  
- Python for Excel parsing & SQL script generation  
- PHP web interface for staging uploads  
  
The system supports:  
- Uploading Excel files  
- Configuring staging details with Master ID and Header Row  
- Generating AI-assisted SQL scripts (using LangChain + Gemini)  
- Downloading processed outputs

## **2. Objectives**

- Save manual effort and reduce time required for preparing staging scripts  
- Provide a simple web interface (PHP) for uploads  
- Automate Excel header mapping & SQL script generation using LLM (Gemini via LangChain)  
- Ensure version control and reusability across projects

## **3. Project Structure**

StagingUpload\_Final/  
│── src/ # Python source scripts  
│ └── stagingCreator.py # Main Python script with LangChain + Gemini integration  
│── download.php # File download script  
│── index.php # Web interface  
│── Required.txt # Notes / additional requirements  
│── images/ # Walkthrough screenshots  
│── requirements.txt # Python dependencies

## **4. Technology Stack**

- Python 3.8+ → Excel processing, AI-based script generation  
- PHP 7+ → Web interface for upload/download  
- LangChain → Prompt orchestration and workflow for AI processing  
- Gemini (Google AI) → LLM integration for dynamic SQL/script generation  
- Libraries/Dependencies → Listed in requirements.txt  
- Database → Supports MySQL (extendable to PostgreSQL/other RDBMS depending on integration)  
- Version Control → Git

## **5. Setup Guide**

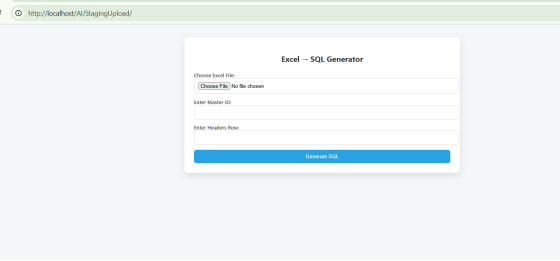
**Install Python**1. Install Python 3.8+  
2. (Optional) Create a virtual environment:  
 python -m venv venv  
 venv\Scripts\activate # Windows  
 source venv/bin/activate # Linux/Mac  
3. Install dependencies:  
 pip install -r requirements.txt

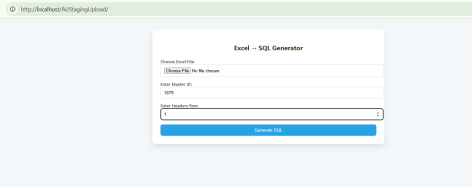
## **6. Usage**

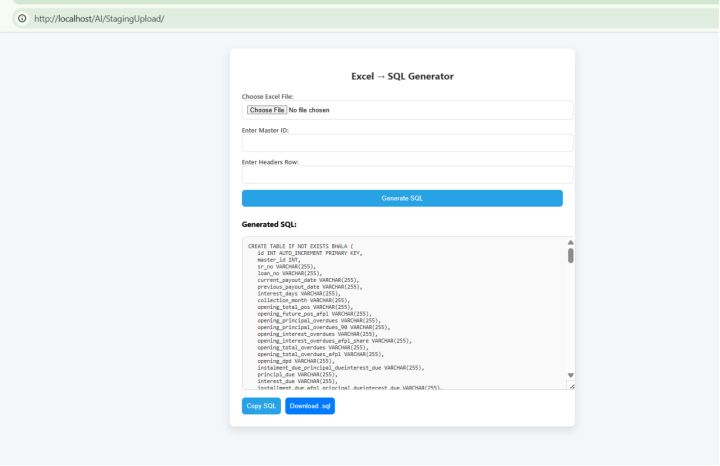
**Python Script**- Parses Excel input  
- Uses LangChain + Gemini to clean headers and generate SQL scripts  
- Generates ready-to-use INSERT/UPDATE statements dynamically  
  
 **Web Interface (index.php)**- Step 1: Upload Excel file  
- Step 2: Enter Master ID  
- Step 3: Select Header Row  
- Step 4: Get processed SQL script (AI-generated via LLM)

## **7. Walkthrough**

**Initial View**Upload Excel, enter Master ID, and select Header Row



**Update With Details**System sends headers to LangChain workflow → Gemini model, which maps headers & generates SQL  
  


**Output**Download AI-generated SQL script  


## **8. Version Control**

- Git-ready setup  
- .gitignore excludes unnecessary files such as:  
- Python cache files (\_\_pycache\_\_/, \*.pyc)  
 - Virtual environments (venv/)  
 - Logs and temporary files  
 - PHP session/cache files

## **9. Benefits**

- Time-saving: From 3 hours → 5 minutes  
- Error reduction: Automated header mapping with LLM (Gemini)  
- Reusable: Works for multiple partners with minimal config  
- Scalable: Easily extendable with new LangChain workflows for staging requirements  
- Future-ready: Any LLM (e.g., GPT, Gemini, Claude) can be integrated via LangChain