AI-BASED MTE EVALUATOR FOR AUTOMATED ASSESSMENT AND FEEDBACK

An Internship Report

Submitted by

RITESH SUNIL KATEKAR

(2021BCS117)

in partial fulfilment for the Project of

B. TECH

IN

COMPUTER SCIENCE AND ENGINEERING

At

SHRI GURU GOBIND SINGHJI INSTITUTE OF ENGINEERING AND TECHNOLOGY,
VISHNUPURI, NANDED
(MAHARASHTRA STATE)
PIN 431 606 INDIA

MAY 2025

Certificate

Shri Guru Gobind Singhji Institute of Engineering & Technology,
Vishnupuri, Nanded (M.S.), India – 431606.

Department of Computer Science and Engineering

Certificate

This is to certify that the internship work entitled **"AI-Based MTE Evaluator for Automated Assessment and Feedback"** is submitted by student **Ritesh Sunil Katekar (2021BCS117)** to Shri Guru Gobind Singhji Institute of Engineering & Technology, Nanded for the partial fulfillment of the award for the degree of Bachelor of Technology in Computer Science and Engineering.

This project is a record of **bonafide** work carried out by him under my guidance. The content presented in this report has not been submitted to any other University or Institute to obtain any other degree or diploma.

Guide
Prof. G.S. Malande

Head of Department
Mrs. Suwarna Bansode

Declaration

I hereby declare that this internship report titled **"AI-Based MTE Evaluator for Automated Assessment and Feedback"** submitted in partial fulfillment for the degree of **Bachelor of Technology (B.Tech)** in **Computer Science and Engineering**, is a genuine record of work carried out by me under the guidance of **Prof. G. S. Malande**.

The report has not been submitted to any other University or Institute for the award of any degree or diploma. All sources of information used in the preparation of this report have been duly acknowledged.

Signature of Student
Ritesh Sunil Katekar
2021BCS117

Acknowledgement

I take this opportunity to express my deepest gratitude to all those who helped me during the course of this project.

I extend my sincere gratitude to the **Guruji Education Foundation**, whose trust and support provided me with this meaningful and socially impactful project. Their initiative gave me the opportunity to apply my technical skills to a real-world educational cause, helping students and mentors alike.

I am highly indebted to my guide, **Prof. G. S. Malande**, for his valuable guidance, consistent encouragement, and support throughout the project. His mentorship and constructive feedback helped shape this work to its final form.

I would also like to thank **Mrs. Suwarna Bansode**, Head of the Department, for her encouragement and support that enabled me to carry out this work.

I also thank the faculty and staff of the Department of Computer Science and Engineering, SGGSIE&T, for providing me with the resources and support required to complete this project successfully.

I extend my heartfelt thanks to my batchmates and friends for their constant support, suggestions, and positive motivation throughout this journey.

Lastly, I express my sincere gratitude to my family for their unwavering emotional strength, patience, and support which enabled me to undertake this endeavor with determination.

Abstract

Manual evaluation of Monthly Thinking Exercises (MTEs) is often a tedious, error-prone, and time-consuming process. With increasing student-teacher ratios and academic pressure, there is an urgent need for smarter evaluation systems.

This project introduces an AI-powered system, **MTE Evaluator**, designed to automate the assessment and feedback process. The system integrates technologies like Gmail and Google Drive APIs with a Python backend, Streamlit interface, and AI models to evaluate '.xlsx' submissions. It automatically reads student submissions from unread emails, evaluates them using LLMs, generates PDF feedback reports, and sends personalized emails to students and their mentors.

The automation significantly reduces human workload, ensures consistency in evaluation, and provides faster feedback. The secure handling of credentials and structured Google Drive storage make the system scalable and practical for institutional use. This work demonstrates the potential of AI in academic automation and its role in shaping the future of education.

Contents

List of Figures

Abbreviations

- **AI** – Artificial Intelligence
- **API** – Application Programming Interface
- **CSV** – Comma-Separated Values
- **GUI** – Graphical User Interface
- **HTTP** – Hypertext Transfer Protocol
- **JSON** – JavaScript Object Notation
- **LLM** – Large Language Model
- **MIME** – Multipurpose Internet Mail Extensions
- **MTE** – Monthly Thinking Exercise
- **OCR** – Optical Character Recognition
- **PDF** – Portable Document Format
- **SMTP** – Simple Mail Transfer Protocol
- **UI** – User Interface
- **URL** – Uniform Resource Locator

Introduction

1.1 Introduction

In today's rapidly evolving educational landscape, particularly within socially-driven initiatives like the **Guruji Education Foundation (GEF)**, there is an urgent need to move beyond traditional academic assessments. GEF not only provides **financial assistance to deserving and underprivileged students**, but also facilitates their **holistic development** through structured programs, mentorship, and self-reflection exercises.

GEF is guided by values such as **Total Inclusivity**, **Quality over Quantity**, **Customized Career Planning**, and the principle of a **Cycle of Giving**. These principles ensure that each student is seen as a complete human being — not just a performer — and receives support beyond academics.

One such critical initiative is the **Monthly Thinking Exercise (MTE)** — a reflective self-assessment tool that encourages students to monitor their academic progress, life challenges, and personal growth. However, the process of manually collecting, reading, evaluating, and responding to these reports across a large student base is labor-intensive and error-prone.

To address this challenge, this project titled **AI-Powered MTE Evaluator: Automated Assessment and Feedback Generation System** was developed. The system leverages automation and artificial intelligence to streamline the complete MTE workflow — from reading incoming emails to generating professional feedback reports — with zero manual effort.

The system processes student-submitted Excel files containing qualitative and structured data across multiple domains like academic goals, financial needs, family issues, reading habits, and more. Using large language models (LLMs), it analyzes these fields, generates personalized feedback, and sends it to both students and mentors. Submissions and reports are also stored systematically on Google Drive for long-term access.

1.2 Understanding the Monthly Thinking Exercise (MTE)

The Monthly Thinking Exercise is a foundational activity introduced by the **Guruji Education Foundation** to ensure that students remain engaged in regular introspection, planning, and communication with their mentors. The MTE document is structured to gather insights into the student's academic, personal, and financial circumstances, thereby helping mentors provide targeted guidance and support.

Each section of the MTE serves a specific purpose:

1. **Academic Progress and Vacation Plan:** Captures monthly academic efforts and vacation planning. This supports **Customized Career Planning** and encourages long-term clarity.

2. **Co and Extra Curricular Progress-Plan:** Highlights personal growth through skills, hobbies, and values — nurturing **Holistic Development**, a core GEF pillar.

3. **Financial Requirements for the Next Three Months:** Helps identify real-time needs so that GEF can maintain its commitment to **Total Inclusivity**, ensuring no student is left behind.

4. **Difficulties (Social, Family, etc.):** Encourages expression of personal struggles, fostering an environment of **Unconditional Acceptance**.

5. **Results of the Exams:** Provides academic metrics for mentors to assess and support without judgment, ensuring feedback focuses on improvement, not comparison.

6. **Reading Books / Watching Videos (Completed, To Complete, Reviews):** Promotes **Self-Directed Learning**, curiosity, and diverse knowledge beyond formal education.

7. **Essay on a Topic of Choice:** Builds communication and critical thinking, encouraging **Empathy**, **Expression**, and independent viewpoints.

8. **Action Plan for the Coming Month:** Reinforces discipline and accountability while supporting **Customized Goal Tracking** and mentor alignment.

Each section is designed not just to collect data, but to trigger thought, reflection, and communication — which form the backbone of GEF's long-term student development philosophy.

1.3 Objectives

The main objectives of the AI-Powered MTE Evaluator system are:

- To automate the reading and extraction of MTE submissions from Gmail as '.xlsx' attachments.
- To interpret and evaluate each of the eight critical sections from the student report using LLMs.
- To generate high-quality, personalized feedback in PDF format.
- To automate the delivery of feedback reports to both students and their mentors via email.
- To organize all MTE data securely and systematically on Google Drive.
- To ensure security, scalability, and minimal manual oversight throughout the process.

1.4 Existing System

The current MTE process under GEF relies heavily on manual intervention:

- Students email completed Excel forms to the GEF team.
- Volunteers or mentors manually download, review, and assess each submission.
- Feedback is given inconsistently or delayed due to human bandwidth limitations.
- File storage is unorganized, leading to data loss or duplication.

While this approach is valuable, it is not scalable given the growing number of students and increasing complexity of data.

1.5 Proposed System

The proposed system automates the entire MTE pipeline with the following modules:

- Gmail API automatically fetches unread MTE emails.
- An Excel parser extracts relevant section-wise content.
- An AI-based evaluator (LLM) processes content and generates personalized feedback.
- Feedback and original submissions are stored in student-specific Google Drive folders.
- Feedback reports are automatically sent to both students and their mentors.

This reduces manual work while improving feedback quality, data traceability, and communication.

1.6 Unique Features of the System

The AI-Powered MTE Evaluator brings unique benefits aligned with GEF's mission and principles:

- **AI-Powered Evaluation:** Evaluates subjective responses like essays and personal reflections using advanced LLMs, offering deep insights.
- **Holistic Feedback:** Covers emotional, social, academic, and financial dimensions — rooted in GEF's emphasis on **Holistic Development**.
- **Total Inclusivity:** Provides unbiased processing of reports, ensuring every student is evaluated equally.

- **Cycle of Giving:** By encouraging reflection, responsibility, and growth, the system contributes to GEF's long-term goal of creating givers.
- **Secure and Organized Storage:** Uses Google Drive to maintain student portfolios in an accessible, reliable format.
- **Dual Communication Channel:** Students and mentors both receive valuable feedback, promoting better relationships and engagement.
- **Adaptable and Scalable:** Custom prompts allow flexibility in how MTEs are interpreted and responded to over time.

By blending advanced AI with the humane values of Guruji Education Foundation, this system ensures MTEs become more meaningful, consistent, and impactful.

Technology Stack

The MTE Evaluator system leverages a multi-layered technology stack combining backend automation, AI-based evaluation, cloud services, and secure communication layers. Each component plays a specific role in processing Monthly Thinking Exercises (MTEs) submitted by students, generating AI-based feedback, and distributing results to all stakeholders.

sectionProgramming Language & Core Libraries

The entire backend logic of the system is implemented in **Python 3.x**, chosen for its simplicity, strong community support, and vast ecosystem of libraries that support data science, automation, and cloud integration.

Core libraries and their roles:

- **pandas** and **openpyxl** are used to read, parse, and manipulate Excel files. These libraries help extract individual MTE sections such as academic plans, financial needs, essays, etc., from student-submitted .xlsx files.
- **fpdf** or **reportlab** is used to programmatically generate structured PDF reports. The reports include a clear layout of scores, reasoning, suggestions, and color-coded remarks for easier visual interpretation.
- **imaplib**, **smtplib**, and **email** are used for email integration. These modules allow the system to:
  - Log in to the Gmail inbox using secure credentials.
  - Read unread messages.
  - Extract .xlsx attachments from student emails.
  - Send personalized PDF reports to students, mentors, and administrators.
- **os**, **shutil**, **json**, and **re** are Python standard libraries used for:
  - File and folder management (creating student folders, handling temp files).
  - Loading secure credentials and configuration.
  - Pattern matching during data parsing and validation.

This modular setup ensures each functionality is handled efficiently while maintaining the readability and scalability of the codebase.

2.1 AI/ML Technology (OpenAI GPT via Groq API)

To simulate human-like evaluation, the system integrates the `deepseek-r1-distill-llama-70b` language model via Groq's API infrastructure.

AI Workflow Overview:

1. **User Request Interface:** A student's '.xlsx' file is parsed, and a prompt is constructed.
2. **Backend Processor:** Adds security headers, formats the request.
3. **Groq API Gateway:** Handles load balancing, rate-limiting, and access control. Routes request to appropriate model.
4. **Groq Model Execution:** Runs on GroqChip AI Accelerator, optimized for ultra-low latency and high throughput. Executes the GPT model and returns output.
5. **Output Stream Handler:** Converts the model response into structured YAML or JSON.
6. **Backend Post-Processing:** Parses, filters, and formats content into human-readable PDF.
7. **Response Delivery:** Sends personalized reports back to users.

Security: API Key or OAuth-based authentication, logging, and usage analytics are in place.

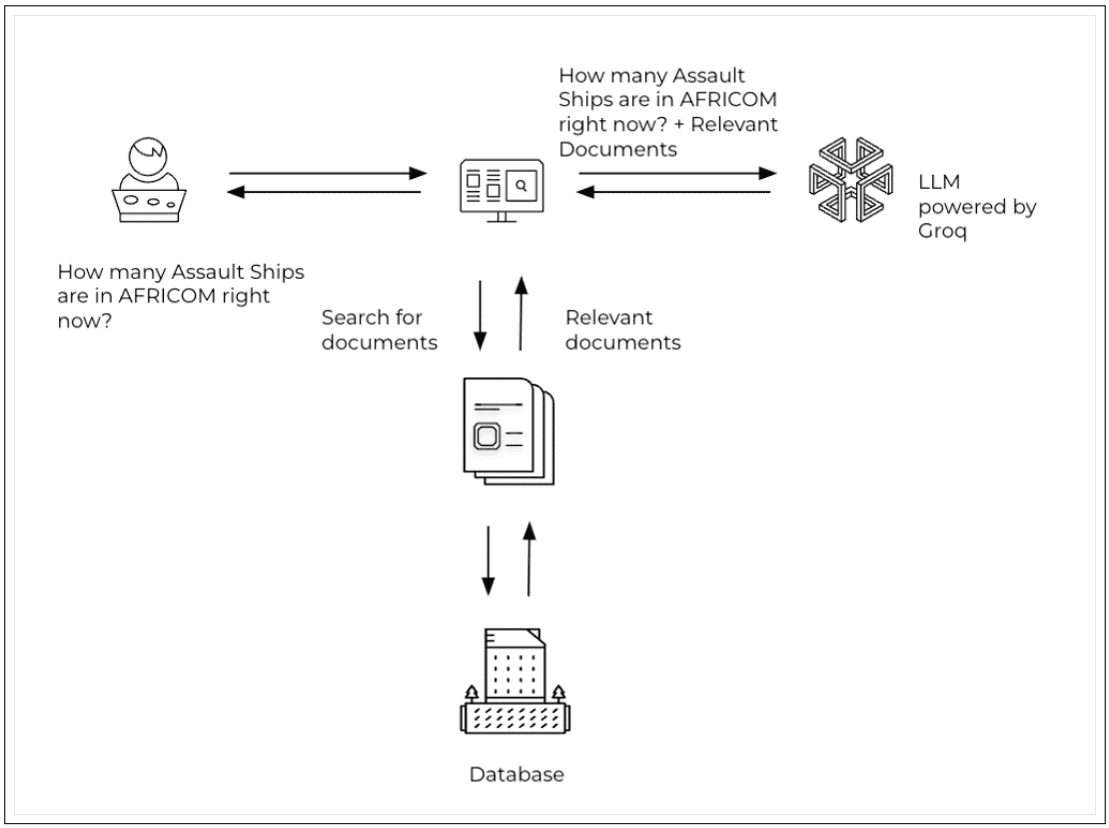

Figure 2.1: Architecture Diagram for Groq API GPT Integration

2.2 Google Cloud APIs

2.2.1 Gmail API Authentication Flow

Step-by-Step Authorization Process:

1. Project is created on Google Developer Console.
2. Unique project ID is generated.
3. User triggers authentication via an authorization servlet.
4. Redirected to Google's OAuth 2.0 consent screen.
5. OAuth token is retrieved after consent.
6. Token is passed to a callback servlet and securely stored.
7. Business logic servlet uses token to make Gmail API calls.
8. Retrieves emails and attachments.
9. Sends structured responses via email.

Key Concepts:

- OAuth 2.0 web server flow.
- Secure token handling.
- Gmail API calls with token-based access.

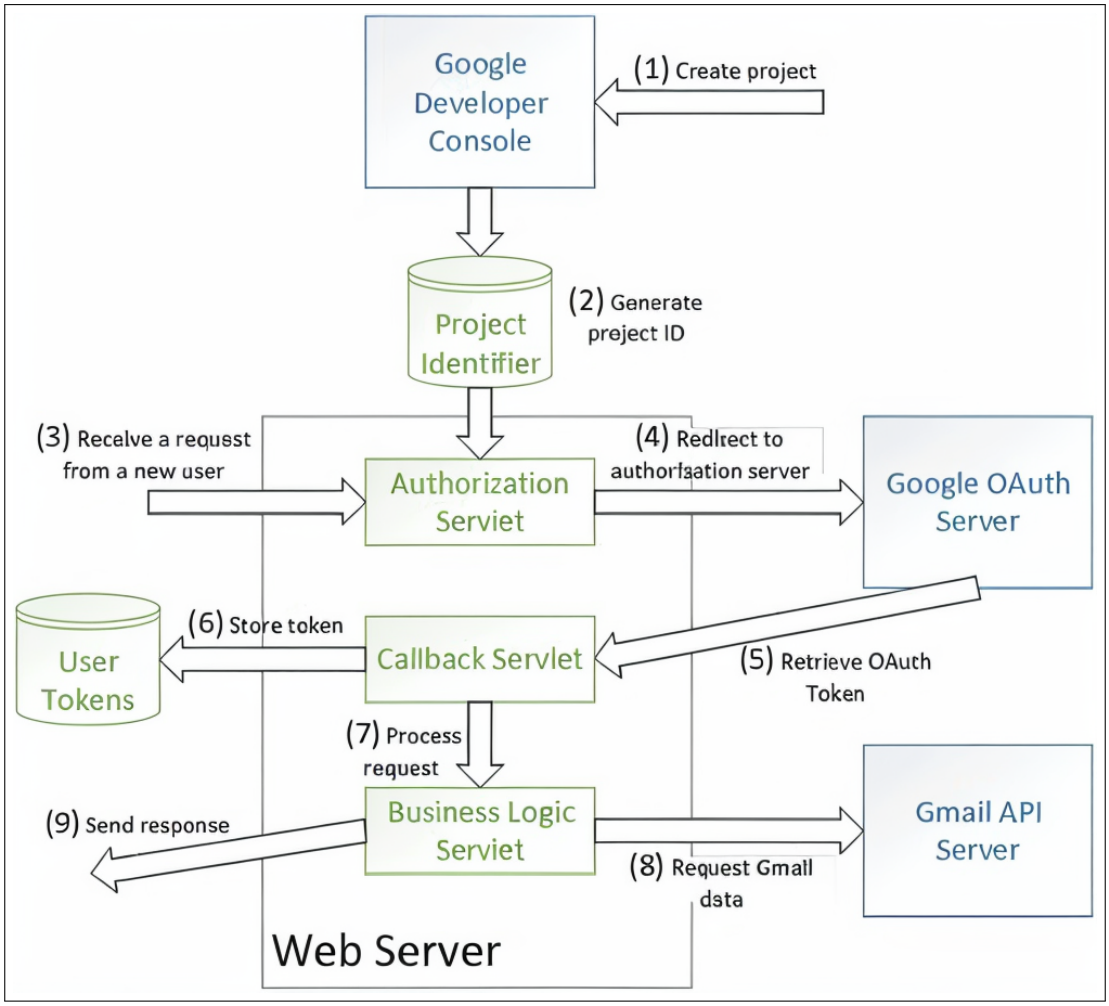

Figure 2.2: Architecture Diagram for Gmail API Integration

2.2.2 Google Drive Architecture

System Components and Flow:

- **Clients:** Desktop/mobile apps initiate file actions.
- **Load Balancer:** Distributes requests to backend services.
- **File Processing Server:** Handles uploads/downloads to/from Cloud Storage.
- **Metadata Server:** Updates and retrieves file metadata from DB.
- **Message Queues:** Decouple services using request/response queues.
- **Notification Server:** Syncs devices via change notifications.

Highlights:

- Microservices + Queues = Scalability.
- Cloud Storage + Metadata DB.

- Real-time sync via Notification Server.

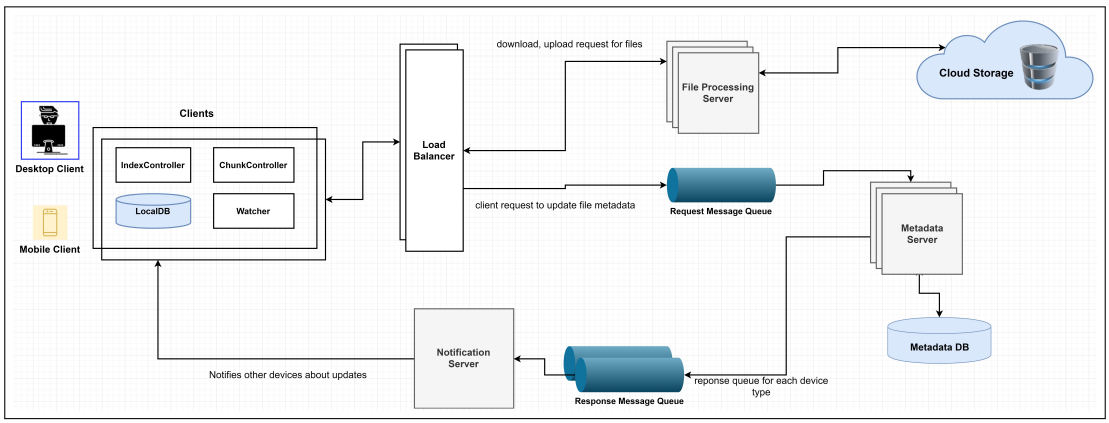

Figure 2.3: Architecture Diagram for Google Drive Backend

2.3 Security & Authentication (OAuth 2.0)

2.3.1 OAuth Flow for WebLogic Applications

Enterprise OAuth Access Process:

1. User accesses WebLogic-hosted app.
2. App requests token from OAM's OAuth Service.
3. Token is issued upon successful authentication.
4. App accesses protected resources using token.
5. Token validation is performed by OAuth Service.
6. If valid, resource is returned to client.

Key Principles:

- Token-based secure access.
- OAuth as centralized auth gateway.
- Application and resource layers are decoupled for modularity.

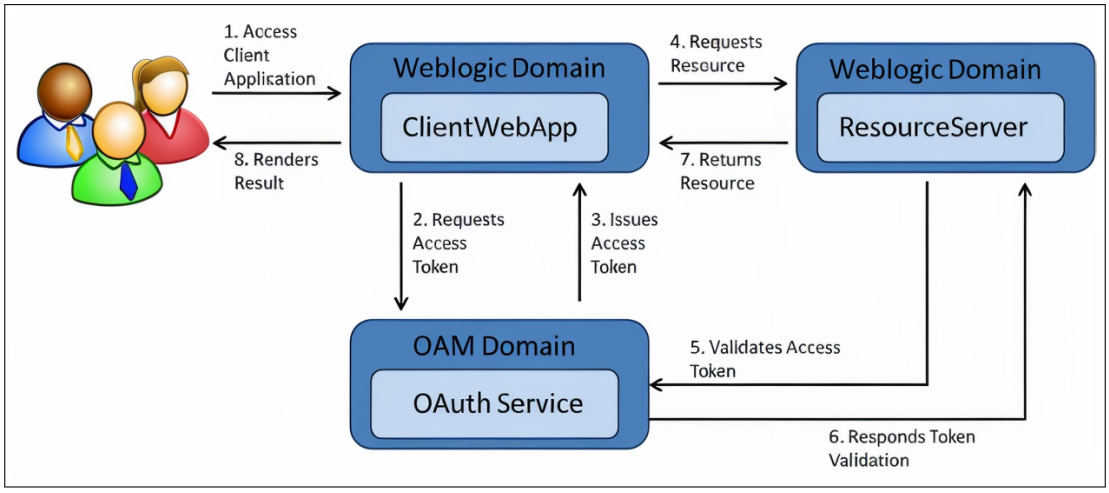

Figure 2.4: OAuth Flow for WebLogic and OAM Integration

2.4 Web Interface (Streamlit)

The UI is powered by **Streamlit**, which supports:

- Uploading ‘.xlsx‘ MTEs.
- Displaying evaluation scores and reasoning.
- Generating downloadable PDF reports.
- Triggering batch processes.

2.5 Conclusion

This architecture enables the automated and intelligent evaluation of student-submitted exercises while ensuring end-to-end security, traceability, and scalability via Google Cloud, Groq APIs, and Python-based automation.

System Design

3.1 Detailed Design

The system is designed to handle the complete automation pipeline of the Monthly Thinking Exercise (MTE) evaluation process. It integrates several key components to streamline reading, interpreting, evaluating, and responding to student submissions effectively.

The workflow begins with the automated retrieval of unread emails sent to the centralized evaluation authority's Gmail account. These emails typically contain '.xlsx' attachments—student-submitted MTE reports. Each Excel file consists of eight essential sections, namely:

1. Academic Progress and Vacation Plan
2. Co and Extra Curricular Progress-Plan
3. Financial Requirements for the Next Three Months
4. Difficulties (Social, Family, etc.)
5. Results of the Exams
6. Reading Books / Watching Videos
7. Essay on a Topic of Choice
8. Action Plan for the Coming Month

Each of these sections is extracted, interpreted, and scored using a Large Language Model (LLM) that not only evaluates the content but also generates constructive feedback.

The evaluated results and feedback are formatted into a professional PDF report, which is then uploaded to the respective student's Google Drive folder. Simultaneously, the original Excel sheet is also backed up. Personalized emails with feedback reports are sent to both the student and their assigned mentor.

3.2 System Architecture

The system is designed as a cloud-based, automated pipeline that handles Monthly Thinking Exercise (MTE) submissions received via Gmail. It integrates email reading, AI-based evaluation, file management, and reporting. The core components include:

- **Gmail Submission:** Students send their '.xlsx' MTE submissions via email to a designated Gmail inbox.

- **Email Reader & Excel Parser:** This module reads incoming emails, extracts '.xlsx' and '.pdf' attachments, and parses their contents using libraries such as `openpyxl` and `pandas`.
- **Google Services Integration:** Extracted files are stored and organized in Google Drive. Google APIs facilitate access to both Gmail and Drive for automation.
- **AI Evaluation Engine:** Parsed data is processed by an LLM (Large Language Model), which evaluates student inputs such as essays or structured answers. The AI engine returns scores, reasoning, and suggestions.
- **Backend Logic:** Core Python scripts orchestrate the entire flow — from prompt construction to response parsing — ensuring seamless integration between modules.
- **Report Generation:** AI evaluations are compiled into a feedback report, which includes performance scores, reasoning, and improvement suggestions. The report is formatted as a professional PDF.
- **Notification and Feedback Delivery:** Finalized reports are sent automatically via Gmail to both the student and mentor, ensuring timely feedback.

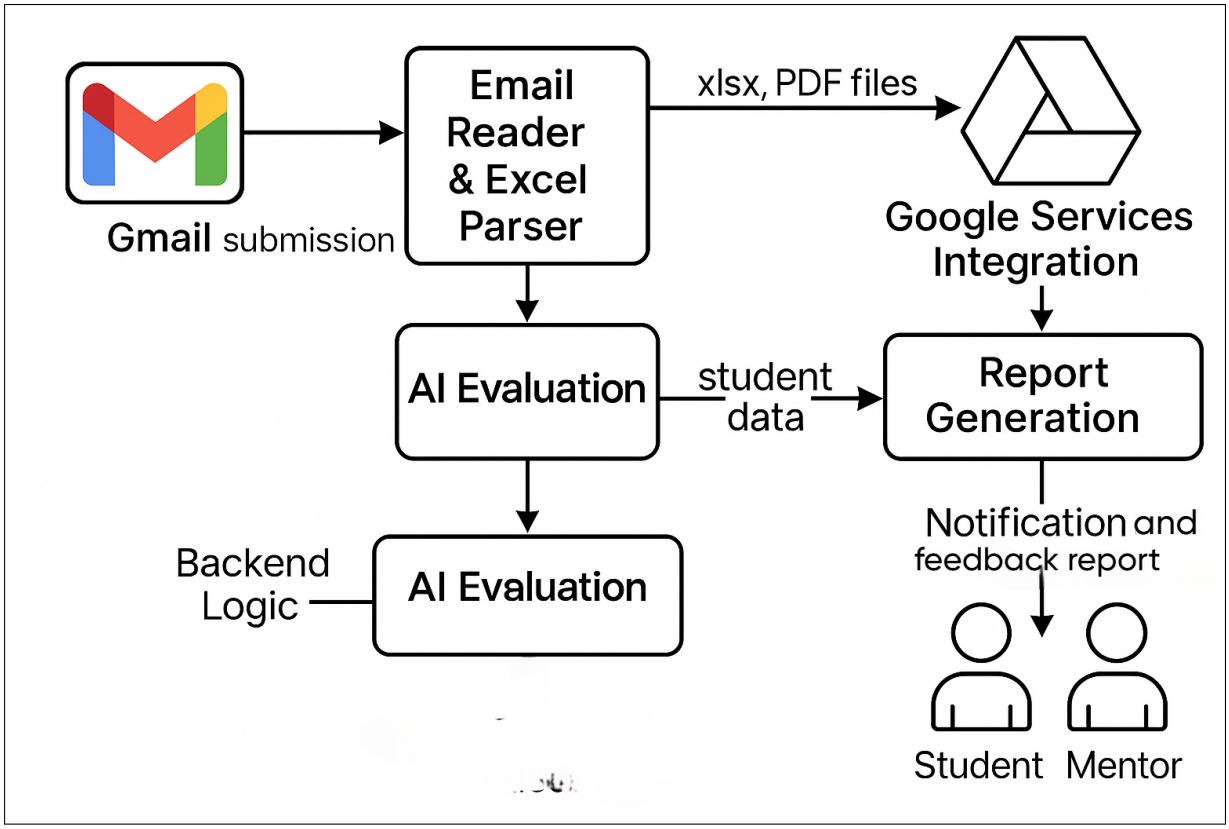

Figure 3.1: System Architecture Diagram

3.3 Module Description

The system is broken down into the following modules:

1. Email Reader Module

Connects securely to the central Gmail account using OAuth 2.0, scans unread emails, and filters only those with '.xlsx' attachments for processing.

2. Excel Parser Module

Reads the structure of the MTE Excel file. Parses key sections such as:

- Academic Progress and Vacation Plan
- Co and Extra Curricular Progress
- Financial Needs
- Personal/Social Challenges
- Exam Results
- Reading/Watching Habits
- Essay on a Chosen Topic
- Action Plan for the Coming Month

3. AI Evaluation Module

Uses a large language model to:

- Interpret narrative content like essays or difficulties.
- Assign objective ratings or qualitative summaries.
- Generate improvement suggestions based on the content.

4. Report Generation Module

Compiles evaluated feedback into a well-formatted PDF, structured with sections corresponding to each part of the student's submission. Includes grading, comments, and suggestions for improvement.

5. Google Drive Integration

Automatically creates and updates folders named after each student. Stores the original Excel file and the corresponding evaluation PDF within the folder for traceability.

6. Email Response Module

Composes and sends an email to the student and mentor, attaching the PDF evaluation. Marks the original submission as read to prevent reprocessing.

3.4 Data Flow

The data flow for the system follows a pipeline pattern:

1. Student submits the MTE via email in '.xlsx' format.
2. System reads unread emails and extracts attachments.
3. Excel data is parsed into structured sections.
4. Data is passed to the AI model for evaluation.
5. PDF report is generated based on evaluation.
6. Files are stored in the appropriate Google Drive folder.
7. Email response with feedback is sent to the student and mentor.

This pipeline ensures minimal human intervention and high accuracy in feedback generation.

3.5 Database Design

Although no traditional relational database is used, the system follows a logical storage structure using Google Drive folders as pseudo-directories:

- Root Directory: `MTE_Submissions`
- Subfolders: `Student_Mail_id`
- Files per Folder:
  - Original MTE Excel Submission
  - Generated Feedback PDF

This hierarchical design facilitates easy access, monitoring, and backup.

3.6 Security Considerations

The system employs secure communication and data storage mechanisms:

- **OAuth 2.0:** All API interactions with Gmail and Google Drive are protected using Google's secure OAuth 2.0 authorization.
- **Token Confidentiality:** Tokens and service account credentials are kept in secure configuration files, not pushed to version control.
- **Read/Write Restrictions:** Scopes for Gmail and Drive are limited to only what's needed (read for Gmail, write for Drive).
- **Access Isolation:** Students and mentors do not access the system backend directly; they receive processed outputs only.

Access Control and Email Integration

4.1 Email-Based Access and Authorization

The AI-Powered MTE Evaluator relies on secure, programmatic access to Gmail and Google Drive APIs for receiving student submissions and organizing evaluation reports. Rather than traditional user login or JWT-based token management, the system is designed to work under a central authority account — typically a department or mentor's email .

All unread emails with '.xlsx' attachments are processed, and the Gmail API provides scoped OAuth 2.0 access to ensure only necessary permissions (read-only access to inbox and read-write access to Drive folders) are granted.

4.1.1 OAuth 2.0 Integration

To ensure secure and authorized access to Google services such as Gmail and Google Drive, the system leverages the OAuth 2.0 protocol. This is the industry-standard mechanism recommended by Google for third-party applications that need access to user data without compromising user credentials.

Once authorized, the access token allows the system to perform the following operations:

- **Fetch Unread Emails:** The Gmail API is queried to fetch all unread emails from a designated inbox, ensuring no submissions are missed or processed multiple times.
- **Download Excel Attachments:** Attachments with the '.xlsx' extension are programmatically identified, extracted, and temporarily stored for evaluation.
- **Upload Feedback to Drive:** Once evaluation is complete, the system uploads the generated PDF feedback reports to a structured student-specific Google Drive directory for archival and traceability.
- **Send Automated Replies:** Using the Gmail API, evaluated reports are emailed back to both students and their assigned mentors, ensuring timely feedback delivery without manual intervention.

4.2 Email Format and Folder Structure

To ensure consistency, automation, and traceability, the MTE Evaluator system relies on a well-defined and structured input-output workflow encompassing email submissions and organized cloud storage. Below is a detailed explanation of the format:

- **Email Format:** Each student is required to submit their Monthly Thinking Exercise (MTE) in the form of an Excel (.xlsx) attachment. These submissions must be sent to

a centralized institutional email address designated for collection and evaluation. Only unread emails with '.xlsx' attachments are processed, ensuring a one-time evaluation per submission.

- **Naming Convention:** To facilitate automated identification and mapping of student records, every Excel submission must follow a strict naming pattern—`FullName.xlsx`. This convention supports seamless indexing, reduces duplication, and ensures traceability across storage systems.
- **Drive Folder Structure:** The system automatically creates and maintains a hierarchical folder structure on Google Drive. Each student has a dedicated folder named after their registered email ID (or roll number if preferred). Within these folders:
  - Original '.xlsx' submissions are preserved for institutional reference.
  - Generated PDF feedback reports are stored chronologically to reflect student progress over time.
- **Root Directory:** All student folders reside within a central root directory titled `MTE Submissions`, which serves as the primary archive for all evaluation records.
- **Access and Visibility:** While students and their mentors receive feedback via email, administrative users can also access these Drive folders manually for audit, review, or academic tracking purposes.

4.3 Security and Privacy Measures

Given the sensitive nature of student data — including financial difficulties, family challenges, and personal reflections — data security is a critical priority in this system. Several mechanisms are implemented to maintain confidentiality:

- OAuth credentials are stored in encrypted files and never exposed in logs or user-facing interfaces.
- All documents are accessible only to the authorized Google account running the automation.
- Emails are never forwarded or stored elsewhere unless explicitly part of the project's communication plan.
- Feedback is sent only to students and their registered mentors based on preconfigured email lists.

4.4 Error Handling and Redundancy

Robustness and fault tolerance are critical to any automation system, particularly one deployed for institutional academic processes. The MTE Evaluator incorporates multiple layers of error handling and redundancy mechanisms to ensure continuity, transparency, and reliability.

- **Graceful Failure Logging:** Any failure encountered during email parsing—such as unreadable messages, missing attachments, or corrupted files—is immediately logged without disrupting the evaluation of subsequent emails. These failed entries are flagged and stored for periodic review, allowing for retries or manual intervention without compromising system uptime.
- **Partial Evaluation Handling:** In cases where specific sections within the Excel-based MTE submission are incomplete, incorrectly formatted, or contain invalid data, the system does not abort the evaluation process. Instead, it proceeds to generate a partial feedback report. A disclaimer is automatically embedded within the generated PDF, notifying the student and mentor about the missing or problematic sections, thereby encouraging data completeness in future submissions.
- **Redundant Email File Tracking:** Every processed email and its corresponding attachments (both input and output) are catalogued and timestamped. This audit trail provides complete traceability for institutional oversight. Additionally, file duplication checks prevent multiple reports from being generated for the same submission, unless explicitly triggered by re-evaluation protocols.
- **Auto-Retry Queues:** A built-in retry mechanism ensures that temporarily failed tasks (e.g., due to API rate limits, network issues, or temporary Gmail/Drive service interruptions) are automatically queued for reprocessing. This minimizes the need for manual oversight and guarantees consistent operation.

These error-handling strategies make the system resilient to unpredictable data anomalies or infrastructural fluctuations, ensuring it can be used reliably in a real-world academic setting without requiring constant supervision.

4.5 Advantages of Email-Based Automation

One of the cornerstone innovations in this project is its reliance on email-based submission and automation rather than a conventional web portal or manual evaluation system. This design choice offers significant benefits in terms of accessibility, simplicity, scalability, and compliance.

- **Zero Onboarding Overhead for Students:** Students are not required to log into any additional platform or undergo complicated registration processes. They simply email their MTE Excel file to a Guruji Foundation and mentors email address—an action they are already familiar with. This ensures ease of adoption and removes technical entry barriers.
- **Credential-Free Monitoring for Mentors:** Respective student mentors, and reviewers can receive reports directly via email without requiring any login credentials or system access. This decentralized delivery model enables prompt feedback and supervision without compromising system security or requiring training.
- **Cost Efficiency and Low Maintenance:** Unlike traditional platforms that require servers, user management, and UI design, this email-centric automation minimizes development and operational costs. Maintenance is simplified, and system updates can be deployed with minimal disruption.
- **Scalability and Replicability:** The modular nature of email processing and file storage allows the system to scale from tens to thousands of users without architectural changes. Moreover, the same design can be replicated across different departments or institutions with only minor configuration changes.
- **Accessibility and Inclusivity:** Students can submit their reports from anywhere and at any time using any device that supports email—thus promoting inclusivity, especially for remote learners or those with limited digital literacy.

Power Search Functionality

5.1 Power Search Methods

The MTE Evaluator is equipped with intelligent search capabilities that enable seamless analysis, retrieval, and assessment of structured data from student-submitted Monthly Thinking Exercise (MTE) Excel sheets. Each `.xlsx` submission contains multiple domains such as academic plans, financial status, reading habits, essays, and more. The system implements context-aware "Power Search" methods powered by natural language understanding.

These methods enable the evaluator to:

- Detect relevant sections automatically based on cell positions and labels.
- Understand varied phrasing of responses (e.g., "Vacation Plan" vs. "Travel plans").
- Retrieve and organize data logically from different sheets or merged cells.
- Extract key points from essay sections and identify sentiment, clarity, and depth.

Using Python libraries like `pandas` and `openpyxl`, data is first transformed into structured dictionaries, which are then parsed semantically using large language models (LLMs) for detailed evaluation.

5.2 Algorithm and Flow Chart for Rendering

5.2.1 Algorithm

The evaluation and search process follows a deterministic yet flexible algorithm:

1. Read unread emails from the Gmail inbox with `.xlsx` attachments.
2. Download the attachment and open the file using `openpyxl`.
3. Identify key sections based on predefined headings:
    - Academic Progress and Vacation Plan
    - Co and Extra Curricular Progress
    - Financial Requirements
    - Family or Social Difficulties
    - Exam Results
    - Reading/Watching Habits
    - Essay on a Selected Topic

- Monthly Action Plan

4. For each section:
   (a) Clean and normalize the text.
   (b) Pass the content to the LLM for summarization, scoring, and feedback generation.
   (c) Generate an evaluation score or remark.
5. Create a feedback report (PDF) and upload it to the respective student's Google Drive folder.
6. Send a response email with the PDF attached to the student and their mentor.

5.2.2 Flow Chart

The flowchart illustrates the complete pipeline for processing Monthly Thinking Exercise (MTE) submissions. The process is designed to be automated and integrates file parsing, AI evaluation, PDF generation, and notification via Gmail. Below are the steps in detail:

1. **Start:** Entry point of the pipeline.
2. **Excel Submission:** Students submit Excel files either through email or a user interface (UI) upload.
3. **Data Preprocessing:** The system uses libraries like `pandas` and `openpyxl` to parse and preprocess the data for further analysis.
4. **Prompt Construction:** The various sections of the Excel sheet are merged into a single, structured prompt suitable for input to a large language model (LLM).
5. **Send Prompt to Groq:** The constructed prompt is sent to a Groq-powered LLM for evaluation.
6. **Parse YAML-Based Output:** The model returns structured YAML output, which is parsed to extract scores, reasoning, and suggestions.
7. **Generate Feedback PDF:** A feedback report is compiled containing:
   - Score
   - Reasoning
   - Suggestions
8. **Upload Report to Gmail API:** The generated PDF is prepared for distribution.

9. **Send Gmail via Gmail API:** The final report is automatically sent to both the student and the mentor via Gmail.

10. **End:** Completion of the pipeline.

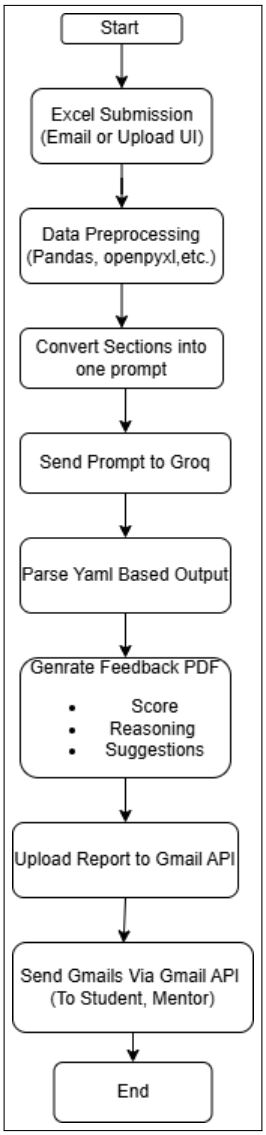

Figure 5.1: Flowchart of the MTE Automated Evaluation Pipeline

5.3 Navigating to Search Route

Although the MTE Evaluator system is primarily designed for autonomous operation, it also provides modular access points that allow developers, testers, or maintainers to manually invoke specific components of the evaluation pipeline. This is especially useful during debugging, integration testing, or refining the AI evaluation logic.

The following internal functions are central to this manual search route:

- `read_excel_sections(file_path)`: Parses the `.xlsx` file submitted by the student and extracts pre-defined MTE sections such as Academic Plan, Financial Needs, Reading Reviews, and Essay topics.
- `evaluate_section_with_ai(section_name, content)`: Sends the content of each extracted section to the AI model for qualitative analysis. The evaluation is based on pre-trained prompts that assess coherence, planning, completeness, and relevance.
- `generate_feedback_pdf(evaluation_data)`: Compiles the evaluated content into a well-structured PDF file with formatted responses and AI-generated remarks for each section.

5.4 Proof of Search

To ensure the system is transparent, auditable, and verifiable, it maintains logs and generates persistent artifacts throughout each stage of the evaluation process. These elements together serve as comprehensive "proof of search" — confirming that each student's Monthly Thinking Exercise was properly analyzed and responded to.

- **Parsed JSON Output:** Every Excel submission is internally converted into a structured JSON format capturing each section (e.g., vacation plans, reading reviews, financial requirements). This format standardizes data processing and simplifies AI input handling.
- **Feedback PDFs:** These documents serve as the primary output of the evaluation system. They include both the student-submitted content and AI-generated feedback, optionally with grading, suggestions, and semantic highlights. These PDFs are timestamped and watermarked for authenticity.
- **Google Drive Uploads:** Both the original `.xlsx` file and the corresponding feedback PDF are stored in clearly labeled, student-specific folders on Google Drive. This ensures longitudinal tracking of student progress and submission history.
- **Sent Emails:** Each evaluation is concluded with an automated email sent to the student and their mentor. These emails include timestamps, download links, and attachments that serve as digital proof of successful evaluation and dispatch.

Collectively, these artifacts ensure transparency, traceability, and accountability in the evaluation process — a key advantage over manual assessment methods.

Advanced Features

The MTE Evaluator system incorporates a variety of advanced features that enhance the intelligence, efficiency, and automation of the Monthly Thinking Exercise (MTE) evaluation workflow. These features distinguish the platform from traditional manual processes and ensure a high level of accuracy, personalization, and scalability. This chapter outlines the most notable advanced capabilities integrated into the project.

6.1 AI-Powered Essay Evaluation

One of the core features of the system is the automated evaluation of student essays using state-of-the-art large language models (LLMs) accessed via the Groq API. The AI is capable of analyzing essays for multiple parameters including clarity, coherence, relevance to the topic, grammar, originality, and tone. Unlike rigid rule-based scoring, the AI understands context and structure, allowing for nuanced and personalized feedback that simulates human evaluation. The essay evaluation also includes constructive suggestions for improvement, making it an effective tool for academic growth.

6.2 Semantic Analysis of Subjective Responses

Several sections of the MTE submission, such as "Financial Requirements", "Family and Social Difficulties", and "Reviews of Books or Videos", are subjective and qualitative in nature. The system employs semantic analysis techniques using LLMs to derive meaningful insights from these inputs. The AI discerns emotional tone, urgency, and significance of the content, ensuring that the feedback generated is both sensitive and contextually accurate. For instance, if a student expresses emotional or financial distress, the system generates empathetic and supportive remarks, rather than generic responses.

6.3 Auto-Personalized Feedback Generation

Based on the parsed content of each section in the Excel document, the system uses dynamic prompt engineering to instruct the AI to generate section-wise, customized feedback. This feedback is tailored to the unique responses of each student, making it feel personal and relevant. The generated remarks are then compiled and embedded into a professional PDF document, enhancing the overall communication quality between students, mentors, and faculty.

6.4 Multi-Section Parsing from Excel

The MTE submission Excel files are structured into multiple content blocks, such as "Academic Progress", "Vacation Plan", "Extra-Curricular Activities", "Action Plan", and so on. The sys-

tem is designed to intelligently parse these sections by dynamically identifying cell headings, content blocks, and structural patterns. It can adapt to slight changes in formatting, ensuring robustness across varying student submissions. This multi-section parsing forms the backbone for accurate content extraction and AI evaluation.

6.5 Email and Attachment Filtering

The system connects to the Gmail API and scans the inbox of the central authority email for unread messages. It incorporates logic to detect whether the email contains a valid '.xlsx' file. If not, the email is ignored. This filtering ensures that only proper MTE submissions are processed, reducing the risk of error and saving computational resources. It also marks emails as "read" after successful processing to avoid reprocessing.

6.6 PDF Report Customization

Each evaluated MTE submission results in a well-structured PDF feedback report. The report includes:

- Student's name and identification.
- Section-wise extracted responses.
- AI-generated feedback.
- Timestamp of submission and evaluation.
- Optional space for mentor or faculty remarks.

The reports are formatted with clear headings, professional typography, and page structure that make them suitable for academic archives.

6.7 Automated Folder Creation in Google Drive

The system integrates with Google Drive API to create structured storage for each student. It checks if a folder already exists by the student's name. If not, it creates one and stores the original '.xlsx' submission along with the generated PDF report inside that folder. This ensures that all documents are automatically archived in a well-organized manner, enabling future access, audits, or reviews.

6.8 Student–Mentor Communication Automation

One of the hallmark features of the MTE Evaluator system is its automated communication mechanism that ensures timely dissemination of evaluation results to all relevant parties. After a successful evaluation cycle:

- The system generates a detailed PDF feedback report personalized for each student.
- This report is automatically attached and emailed to the student using the address extracted from the submission email.
- Simultaneously, a copy of the feedback is also sent to the designated mentor or academic advisor. This ensures that mentors remain in the loop and can track the student's academic and personal development trends.

This dual-dispatch mechanism fosters a transparent and accountable mentoring process. It reduces communication delays, minimizes manual interventions, and allows mentors to provide timely guidance based on structured feedback.

6.9 Secure Configuration Handling

Given that the system integrates with multiple cloud-based services such as Gmail, Google Drive, and large language models (e.g., OpenAI or Groq), maintaining a secure configuration layer is critical. The MTE Evaluator incorporates secure practices to handle credentials and tokens responsibly, ensuring data protection and system integrity.

- **Environment Variables:** Sensitive information such as Gmail OAuth credentials, Google Drive API tokens, and LLM API keys are stored in a '.env' file. This file is not committed to the version control system and remains local to the deployment environment.
- **Credential Isolation:** All configuration values are loaded dynamically at runtime using a secure environment manager (e.g., Python's `os` module or `python-dotenv`). This approach eliminates hard-coded secrets from the codebase.
- **Access Control:** Only authorized personnel with system-level access can modify or regenerate credentials. This limits exposure and minimizes the risk of unauthorized access or credential leakage.

These security measures align with best practices in DevOps and cloud-native application development. They also satisfy common compliance requirements in educational IT systems where student data privacy is paramount.

Back-Office and its Services

The Back-Office component of the AI-Powered MTE Evaluator serves as the administrative engine that governs and oversees the processing, evaluation, storage, and dissemination of Monthly Thinking Exercise (MTE) submissions. It enables faculty or institutional authorities to efficiently monitor student data, ensure organized storage on the cloud (Google Drive), and manage evaluation resources seamlessly. While the system does not operate like a traditional e-commerce backend, the structural concepts of orders, products, and dependencies are metaphorically used to describe internal functionalities and file management mechanisms.

7.1 Orders Management

In the context of the MTE Evaluator, an "Order" represents a complete submission cycle from a student — comprising a filled Excel document that encapsulates their Monthly Thinking Exercise (MTE) reflections. Each submission contains the following structured sections:

- Academic Progress and Vacation Plan
- Co and Extra Curricular Activities Plan
- Financial Requirements for the next 3 months
- Difficulties (Social, Family, etc.)
- Results of the Exams
- Reading and Watching Habits — completed, to be completed, and reviews
- Essay on a chosen topic
- Action Plan for the upcoming month

The system treats each Excel file as a discrete "order" and handles it using an automated pipeline:

- Reads unread Gmail messages with '.xlsx' attachments from the designated institutional inbox.
- Parses and validates the contents of the Excel file for completeness and formatting.
- Logs key metadata including student email, mentor email (if mentioned), submission timestamp, and filename.

Effective order management ensures:

- Each submission is evaluated only once, avoiding duplicate processing.
- Every evaluated submission is linked to a uniquely generated feedback report.
- All submission artifacts (original and processed) are stored in dedicated student folders on Google Drive.

7.2 Products Management

The core "products" generated by the MTE Evaluator are artifacts resulting from the AI-based evaluation. These include:

- Personalized feedback reports in PDF format.
- Evaluation logs and summaries for administrative oversight.
- Final communication emails containing feedback attachments for students and their mentors.

Each product is tailored to the unique content submitted by the student. Text-heavy fields such as essays, progress plans, and reflections are sent through an AI model for qualitative analysis. The generated feedback includes:

- Section-wise strengths and improvement areas.
- Constructive suggestions aligned with the student's goals.
- Holistic remarks on planning, awareness, and growth mindset.

All these products are securely archived on Google Drive. Folder hierarchies ensure that reports can be retrieved chronologically or by student identifiers, supporting both academic counseling and institutional reporting.

7.3 Product Dependencies Management

The MTE Evaluator operates in a modular environment, depending on multiple external services for its seamless functioning. These dependencies are carefully orchestrated and include:

- **Gmail API**: For fetching unread emails and extracting '.xlsx' attachments.
- **Google Drive API**: For dynamically creating student folders and uploading reports.
- **Streamlit**: To offer a minimal graphical interface for manual overrides, testing, or review.

- **OpenAI or Groq LLM APIs**: For evaluating subjective sections using AI-driven language models.

Credential files such as ‘credentials.json‘ are securely managed and excluded from version control using ‘.gitignore‘. Runtime checks are performed to ensure token validity before processing, reducing the risk of authorization errors and improving fault tolerance.

7.4 Pagination

As the number of student submissions increases, efficient data visualization and processing become critical. Pagination is employed in both backend logic and the frontend interface to ensure scalable performance:

- **Frontend Pagination:** Streamlit displays submission logs and report status in a paginated manner, typically in batches of 10–20 entries. This prevents UI clutter and supports responsive design.
- **Backend Batching:** Evaluation logic processes submissions in batches to avoid memory overload and reduce the risk of rate limits from external APIs.

Pagination becomes especially important during review sessions, when faculty may need to browse reports for multiple students. It ensures a smooth user experience without compromising system stability, even under heavy workloads.

Conclusion

8.1 Conclusion

The AI-Powered MTE Evaluator project demonstrates how automation, artificial intelligence, and cloud technology can be integrated to streamline critical educational processes. Manual evaluation of Monthly Thinking Exercises (MTEs) was previously time-consuming, inconsistent, and error-prone. This system not only addresses those issues but also adds value by:

- Automatically reading student submissions via email.
- Extracting and interpreting Excel content that includes academic, financial, personal, and reflective inputs.
- Using AI models to generate human-like, constructive feedback.
- Sending personalized reports to students and mentors.
- Organizing all data in structured Google Drive folders per student.

The system brings significant benefits in terms of:

- Transparency and fairness in evaluation.
- Reduction in faculty workload.
- Faster turnaround for feedback delivery.
- Encouraging students to meaningfully reflect on their academic and personal growth.

The AI-Powered MTE Evaluator is a scalable solution that aligns with the vision of digital transformation in education. With continued enhancements, it has the potential to become a core academic tool that enriches student learning and institutional efficiency.

REFERENCES

- Google Developers. *Gmail API Documentation*. Retrieved from: https://developers.google.com/gmail/api
- Google Developers. *Google Drive API Documentation*. Retrieved from: https://developers.google.com/drive
- OpenAI. *API Reference for Language Models*. Retrieved from: https://platform.openai.com/docs
- Streamlit. *Official Documentation for Building Web Apps*. Retrieved from: https://docs.streamlit.io/
- Python Software Foundation. *Email and MIME Libraries*. Retrieved from: https://docs.python.org/3/library/email.html
- Groq. *LLM Acceleration Platform Documentation*. Retrieved from: https://groq.com
- Stack Overflow. Community discussions on parsing Excel files and automation with Python. Retrieved from: https://stackoverflow.com

Appendix

Appendix A: Internship Offer Letter

GURUJI EDUCATION FOUNDATION
REG. OFFICE: 17 GANADHISH, VEER SAVARKAR SOCIETY, SAHAKARNAGAR 2, PUNE - 411 030.
HEAD OFFICE: B 1902 JASMINE TOWERS VASANTVIHAR THANE WEST 400 610.
PHONE : 091-22-49713197 E-MAIL : info@gurujifoundation.in URL : www.gurujifoundation.in

31st December 2024

To,

The Training and Placement Officer,
Shri Guru Gobind Singhji Institute of Engineering and Technology,
Nanded, Maharashtra.

Subject: Confirmation of Internship of Ritesh Katekar in Guruji Education Foundation

Dear Sir,

This is to confirm that Ritesh Katekar will be doing his internship from January 2025 till May 2025 in Guruji Education Foundation (GEF). Our Foundation helps high-potential students in their holistic development, using education as a foundational block for their growth. We provide a bedrock of personal mentorship, financial assistance and any other resources, setting them up for success so that one day they might continue in our cycle of giving, inclusivity and change.

As part of this internship, Ritesh will be remotely working on the cutting-edge technologies for the development of in-house applications of GEF, which are crucial for the smooth functioning of GEF activities. Ritesh will be awarded a Certificate at the end of the successful completion of his internship in GEF.

Thanking you,

Shridhar Vaishampayan
Mentor and Executive Board Member,
Guruji Education Foundation.