LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

Capstone Tracker with an Integrated Evaluation System

UE22CS441A – Capstone Project Phase – 3

Submitted by:

Dhanush S Jettipalle	**PES2UG22CS175**
Ketan Kancharla	**PES2UG22CS263**
Nitheesh Pugazhanthi	**PES2UG22CS371**
Rohan M G	**PES2UG22CS454**

Under the guidance of

Dr.Richa Sharma
Associate Professor
PES University

August - December 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Bengaluru – 560 100, Karnataka, India

TABLE OF CONTENTS

1. Introduction

1.1. Overview

In our project "Capstone Tracker with an Integrated Evaluation System" aims at two primary things, one is to simplify the process of tracking all the capstone projects, their progresses and their deliverables and the other is to automate the evaluation of the capstone reports and final papers using LLMs.

The capstone tracking system will help address the challenges that a capstone guide might face while managing multiple teams and multiple deliverables for each team which will act as a centralized server where the mentors and panel members can track the progress of each team through every phase. By streamlining the submission and review of deliverables, the tracker allows mentors to focus on guiding students academically while our project will take care of the rest.

In addition to the tracker, we plan on solving the issues mentors face while evaluating reports which are time consuming and too long to evaluate consistently by introducing a LLM which will be fine- tuned to evaluate most of the deliverables that is expected in the capstone process as well as the evaluation of the final paper. The LLM will need to check for the formats and have to check for various parameters like relevance, factual correctness etc.

We plan on introducing a new quality index that will better fit our reports and help us better evaluate the reports accordingly. This way will also give us a fair and objective way of evaluation without any disparities.

1.2. Purpose

Our project's main aim is twofold: one to make the tracking of the various capstone deliverables much easier and two to automate the evaluation of the submitted capstone deliverables.

To reduce the burden of capstone mentors in keeping track of each individual team and their deliverables, we introduce the capstone tracker. This allows mentors and the capstone committee to track the progress of every team phase wise. This allows the mentors to focus on the academic part of the projects and leave the management part to our product. The teams will be able to submit all the deliverables and can be accessed easily anywhere and at any time by the mentors and the committee.

The Project will also take care of evaluating the deliverables submitted by the students. First there will be a plagiarism check done, to avoid any cases of stealing information or content.

After the plagiarism check the deliverables i.e. the power point presentations and the reports. The mentors need not comb through all the reports and the documents for errors and grammatical mistakes. The submitted deliverables will be assessed on various parameters like clarity, readability and technical soundness.

The project will also be able to provide constructive feedback on the report quality and thus act like a reviewer or at most help reinforce the mentor's suggestions. The project will also be able to analyze the reports and predict whether the project can be extended and worked for a longer period of time.

Thus, we propose a fine-tuned LLM which will run at the backend of the project and assess the documents on our newly defined quality index.

1.3. Scope

Capstone Project Tracking Create a comprehensive digital platform to monitor and manage capstone project deliverables

Enable real-time tracking of project phases, milestones, and progress

Reduce administrative burden for capstone committees through automated tracking and reporting

Research Review Automation using Large Language Models (LLMs)

Leverage advanced AI technologies to streamline the academic paper review process

Develop a fine-tuned LLM model to assist in preliminary screening and assessment of research submissions.

The system will feature a custom quality index for assessing submitted papers, generate automated reviews and scores, and enable students to enhance their project impact.

Deliverables include the software system, comprehensive documentation, and a detailed presentation deck covering technical architecture, LLM capabilities, implementation methodology, performance metrics, and future enhancement roadmap.

* Integration of Automated Scoring: Developing a robust algorithm for automated scoring of deliverables while ensuring fairness and transparency can be challenging, especially when subjective evaluation is required but at the same time it is not practical to read line by line of the deliverables to evaluate them accurately.

* Contextual Scoring Systems: A gap exists in creating automated scoring algorithms capable of understanding the context and quality of deliverables beyond surface-level features.

* Auto-Evaluation Complexity: Designing an auto-evaluation system capable of assessing diverse deliverables (e.g., reports, presentations, code) accurately across different capstone phases is a significant technical challenge.

* Real-Time Evaluation Models: Research on developing models that can evaluate deliverables in real-time with minimal latency is still evolving.

2. Design Constraints, Assumptions, and Dependencies

The main design limitations, presumptions, and dependencies that affected the creation of the Capstone Tracker with an Integrated Evaluation System are described in this document. The limited amount of computational resources available was one of the main obstacles encountered during the project. The design had to rely on more compact and effective models like Deepseek V2, LLaMA 3.1, BERT, and SciBERT because access to powerful GPUs and memory-intensive infrastructure was limited. These models were chosen to maintain accuracy and consistency in the outcomes while striking a balance between computational viability and performance. The lack of a common metric to verify the results produced by huge language models presented another difficulty.

This lack of a universal baseline made it difficult to measure the reliability of the model's predictions, which led to the introduction of a custom weighted quality index combining several readability and linguistic metrics.

Several assumptions guided the development process and the preparation of this document. It was assumed that the peer review data from the ICLR conference used for fine-tuning the model was fair and unbiased, representing realistic academic review standards. It was also assumed that readability measures such as the Automated Readability Index (ARI) and the Gunning Fog Index (GFI) would complement one another, with the ARI counteracting the overestimation tendencies observed in GFI. In addition, the team assumed that the combination of rule-based and semantic evaluation techniques could be generalized to a wide range of academic documents beyond the immediate dataset.

The project's implementation also depended on several external frameworks, datasets, and tools. Core dependencies included Python libraries such as Hugging Face Transformers, PyTorch, and scikit-learn, which were used for model training, evaluation, and data preprocessing. The functionality of the deliverable tracker and chatbot relied on stable API integrations and the ability to efficiently call LLM inference endpoints. Moreover, access to large, high-quality datasets—such as past capstone submissions and peer review corpora—was crucial for effective model training and testing. Together, these constraints, assumptions, and dependencies defined the operational boundaries of the project and ensured that all design choices remained practical within the available resources and academic context.

3. Design Description

This document presents the design architecture of the *Capstone Tracker with an Integrated Evaluation System* and explains how the major components interact to achieve the system's objectives. The overall design focuses on modularity, scalability, and clarity, with two primary modules forming the foundation of the system: the Capstone Tracker and the Evaluation Engine.

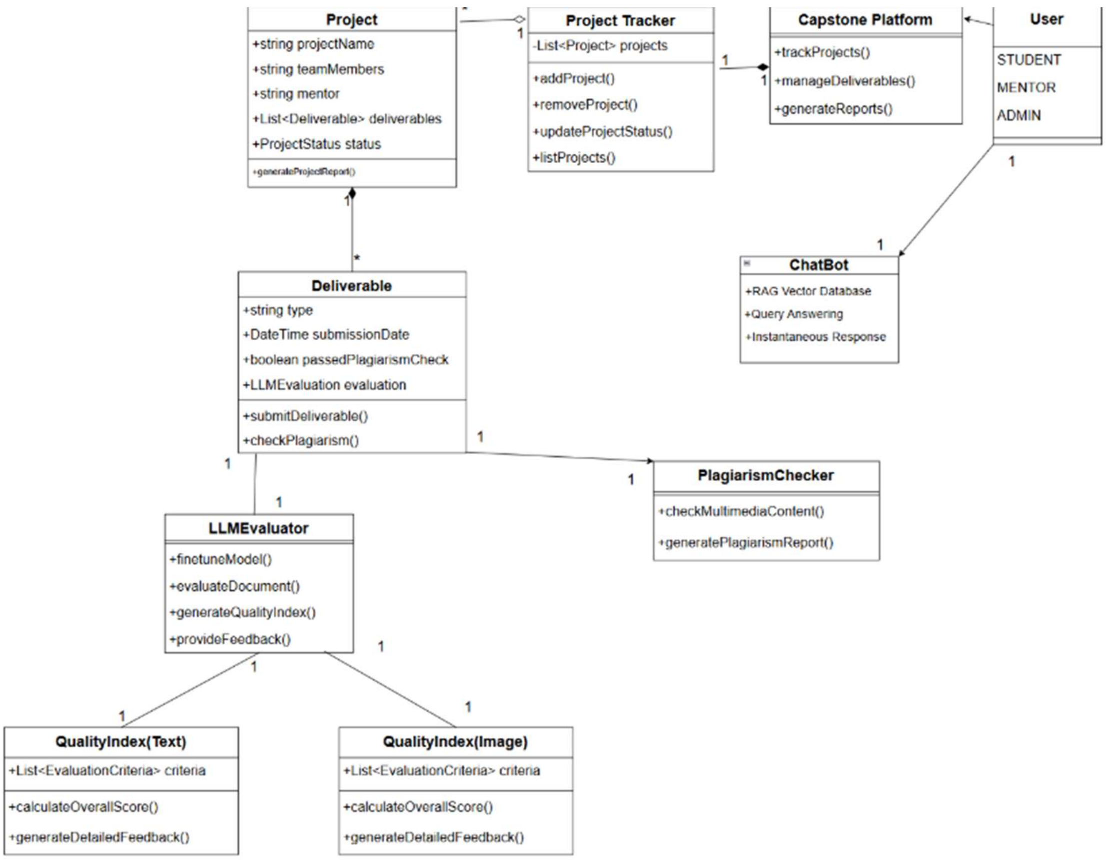
The Capstone Tracker acts as the management layer of the system. It provides a digital platform through which mentors, coordinators, and students can monitor project progress, upload deliverables, and track milestones in real time. The interface is designed to be intuitive and user-friendly, ensuring that both students and faculty can easily navigate through project phases. A chatbot is integrated into this module to assist users by answering queries and offering guidance about submissions, deadlines, and report formats. This not only enhances accessibility but also reduces the need for constant manual supervision by faculty members.

The Evaluation Engine forms the analytical core of the system. It leverages large language models to assess the quality of project reports and presentations based on both linguistic and semantic features. The evaluation is performed in two stages: a rule-based evaluation using the custom weighted quality index and a semantic evaluation using Aspect-Based Sentiment Analysis (ABSA). The quality index combines multiple metrics such as Lexical Density, Gunning Fog Index, Automated Readability Index, and the Indecisive Index, each weighted according to its contribution to readability and clarity. The semantic analysis is powered by models such as Deepseek V2 and LLaMA 3.1, which generate aspect-specific scores for criteria like technical soundness and novelty. For better efficiency, these models are later distilled into smaller, domain-tuned versions like BERT and SciBERT, making the evaluation process faster and more resource-friendly.

The design also defines clear data flow and communication between components. Each module communicates through well-defined APIs that handle data transfer, model inference, and visualization. The system supports cloud-based deployment, allowing for distributed access and seamless management of deliverables from multiple teams simultaneously. Supporting diagrams, including the use case, class, and sequence diagrams, visually represent how the user interacts with the system and how data moves between the components.

Overall, the design emphasizes transparency, modularity, and efficiency. By integrating deliverable tracking with intelligent automated evaluation, the system provides a complete academic management solution that simplifies project monitoring, enhances grading consistency, and supports a more objective and efficient evaluation process.

3.1. Master Class Diagram



3.2. Module 1

Web App for tracking the deliverables:

3.2.1. Description

This module focuses on managing capstone projects from creation to evaluation within the platform. It allows users (students, mentors, and administrators) to track project progress, manage deliverables, and generate reports through an integrated workflow.

Each project is associated with deliverables that are evaluated both for plagiarism and technical quality using an LLM-based evaluation system. The module ensures that all submissions are analyzed thoroughly and feedback is provided automatically to the user.

The **Project Tracker** maintains the list of all active projects, allowing addition, removal, and status updates. The **Deliverable** class manages submissions, ensuring each deliverable undergoes plagiarism checks (handled by the **PlagiarismChecker**) and quality assessment (handled by the **LLMEvaluator**).

The **Capstone Platform** oversees the coordination between these components—tracking projects, managing deliverables, and generating consolidated reports. The

ChatBot component provides quick access to information and assists users with project-related queries through the RAG vector database for instant, intelligent responses.

Module 2

LLM Evaluation Module :

Description

A LLM which will be fine tuned using all the previously submitted capstone deliverables by the previous batches.

A quality index is defined based on various parameters like clarity , technical soundness and originality.

The LLM will also evaluate the relevance of each image present in the in report in accordance to the text present near it.

The fine tuned LLM will evaluate the capstone deliverables based on this index and then finally award a grade based on a final threshold which will be clearly specified.

The LLM will also provide constructive feedback on how it is grading a particular deliverable and what is wrong with it.

Module 3

Chatbot based on the uploaded deliverables:

This module will have a chatbot which will be able to answer various queries about the about the projects, including their objectives, progress, and key details.

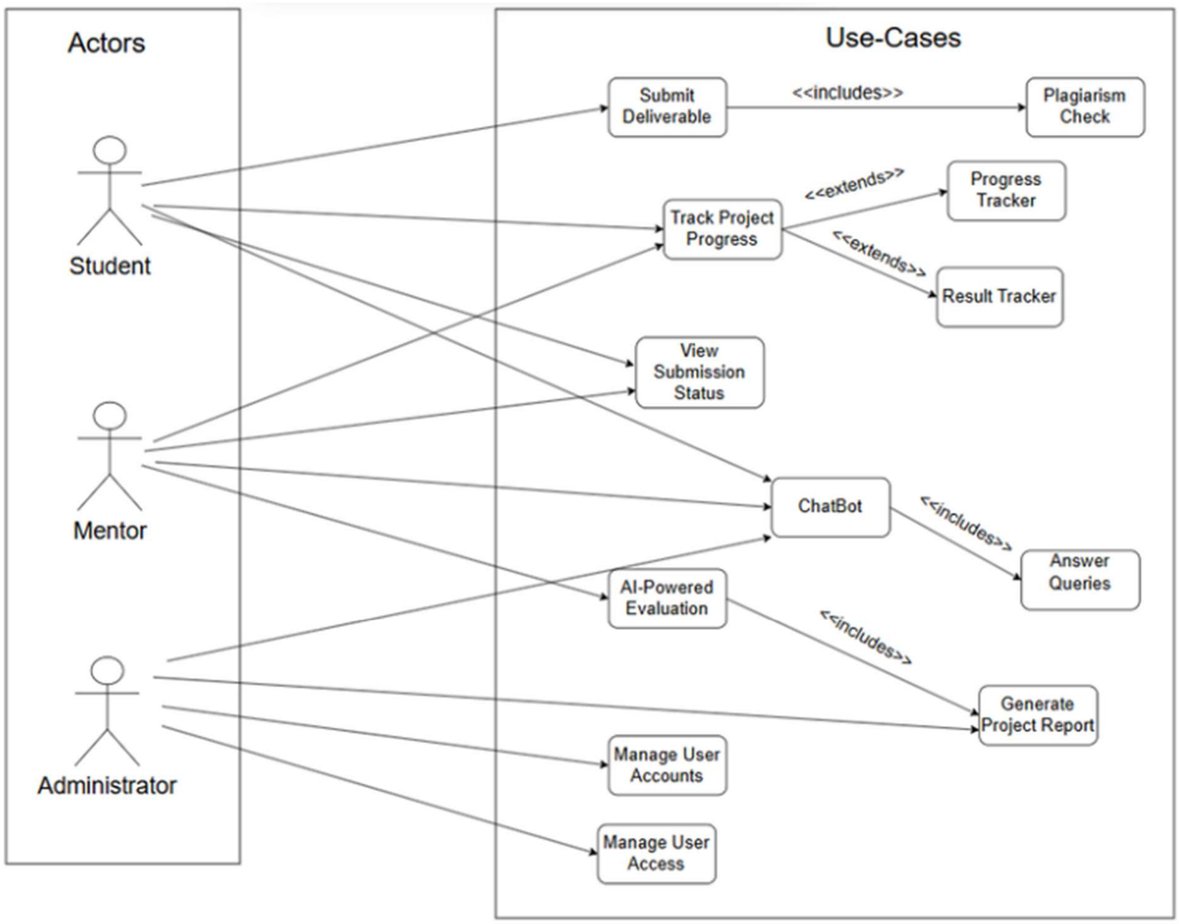
It will be able to generate concise summaries of each project, outlining their design approach, architecture, and overall structure.

It will be able to quickly segregate the project and return the best of any particular year or domain as per choice.

Additionally, the chatbot will address technical queries, such as the technology stack used, implementation details, and potential areas for improvement.

This will be implemented this using a Retrieval-Augmented Generation (RAG) and vector database.

3.2.2. Use Case Diagram



Example:

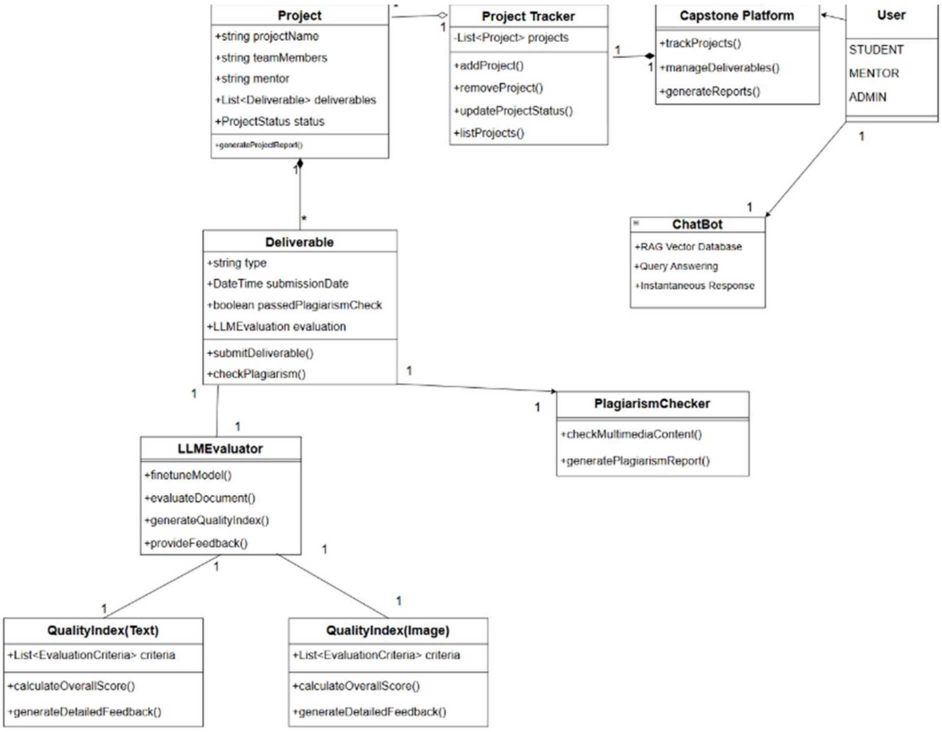
Use Case Item	Description
Submit Deliverable	A user submits a project or assignment to the system. This use case includes a **Plagiarism Check** to ensure the originality of the submission.
Track Project Progress	A user can monitor the status of their project. This use case extends to two sub-functions: a **Progress Tracker** to view overall completion and a **Result Tracker** to see the final outcomes or scores.
View Submission Status	A user can check whether their submitted deliverable has been received, is being processed, or is awaiting a grade.

ChatBot	Users can interact with a chatbot. This use case includes the ability to **Answer Queries** by providing quick, automated responses to common questions.
AI-Powered Evaluation	A user can use an AI-powered tool to evaluate submissions. This use case includes generating a **Project Report** that provides detailed feedback and analysis.
Manage User Accounts	A user with administrative privileges can create, modify, or delete user accounts.
Manage User Access	A user with administrative privileges can control the permissions and access levels for different users within the system.

3.2.3. Class Diagram

[Description of each class in this class diagram will be given. A diagram of the entire system will be given at a high level and then broken down into sub levels. Classes maybe repeated across class diagrams, to show the interfaces with other classes.]

For Example



3.2.3.1. Class Name 1: Project

3.2.3.2. Class Description 1

4. The Project class represents a capstone project. It holds key details such as project name, team members, mentor, and the list of deliverables. It provides methods to generate a report and track project status.

4.2.3.1. Data members 1

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	Project name	Private	Null	Stores the name of the project.
String	Team Members	Private	Null	Stores the list of team members involved.
String	Mentor	Private	Null	Stores the mentor assigned to the project.

List<Deliverable>	Deliverable	Private	[]	Holds all deliverables associated with the project.
Project Status	Status	Private	Started implemention	Indicates the current progress stage of the project.

4.2.3.2. Method 1: generateProjectReport()

The following details shall be defined for the methods:

- Purpose: To generate a comprehensive report of project progress and deliverables.
- Input: None
- Output: Report object or formatted summary
- Parameters: None
- Exceptions: File generation or data retrieval errors

4.2.3.3. Class Name 2: Deliverable

4.2.3.4. Class Description 2

Represents each submission or milestone of a project. It stores information such as submission date, type, plagiarism status, and evaluation results.

4.2.3.5. Data Members 2

Data Type	Data Name	Access Modifiers	Initial Value	Description
String	Type	Private	Null	Specifies the deliverable type
Date time	Submission Date	Private	Null	Stores the date and time of submission.
Boolean	Plagiarism Check	Private	False	Indicates if plagiarism was detected.

4.2.3.6. Methods 1: submitDeliverable()

- Purpose: Allows students to submit their project deliverables.
- Input: Deliverable file or data
- Output: Confirmation message
- Parameters: Deliverable d
- Exceptions: File not found or upload errors

4.2.3.7. Methods 2: checkPlagiarism()

- Purpose: Performs plagiarism verification for the submitted deliverable.
- Input: Deliverable content
- Output: Boolean value (True if passed)
- Parameters: None
- Exceptions: API or system errors

4.2.3.8. Class Name 3: LLMEvaluator

Class Description:

Responsible for evaluating academic deliverables using a fine-tuned language model. Generates a quality index and provides feedback.

Data Type	Data Name	Access Modifiers	Initial Value	Description
Model	finetuneModel	Private	Null	Holds the trained evaluation model.

3.2.3.16 Methods

- Method 1: finetuneModel()

- Purpose: Customizes the base model for academic evaluation.
- Input: Training data
- Output: Updated LLM model

Method 2: evaluateDocument()

- Purpose: Analyzes and scores submitted deliverables.
- Input: Deliverable text or file
- Output: Evaluation results

Method 3: generateQualityIndex()

- Purpose: Produces a composite quality score.
- Input: Evaluation metrics
- Output: Numeric index (0–100)

Method 4: provideFeedback()

- Purpose: Gives textual feedback to students.
- Input: Evaluation report
- Output: Feedback message

3.2.3.17 Class Name 4: ProjectTracker

3.2.3.18 Class Description

Maintains the list of projects in the platform and provides operations to add, remove, update status, and list projects.

3.2.3.19 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
List<Project>	projects	Private	[]	In-memory collection of all projects.

3.2.3.20 Methods

Method: addProject()

- Purpose: Register a new project in the tracker.
- Input: Project
- Output: boolean (success)
- Parameters: Project p
- Exceptions: Duplicate project, validation errors

method 2

- Method: removeProject()
- Purpose: Remove an existing project.
- Input/Output: String projectName → Boolean

method 3

- Method: updateProjectStatus()
- Purpose: Change a project's status.

- Parameters: String projectName, ProjectStatus newStatus

- Output: boolean

method 4

- Method: listProjects()

- Purpose: Return all tracked projects.

- Output: List<Project>

3.2.3.25 Class Name 6: ChatBot
3.2.3.26 Class Description

Provides instant help to users by answering common questions using a vector database for retrieval.

3.2.3.27 Data Members

Data Type	Data Name	Access Modifiers	Initial Value	Description
RAGVectorDatabase	ragDB	Private	null	Stores embeddings for retrieval.
List<Message>	history	Private	[]	Recent conversation turns (optional).

method 1

3.2.3.28 Methods

Method: answerQuery()

- Purpose: Return a concise answer to a user question.

- Parameters: String query

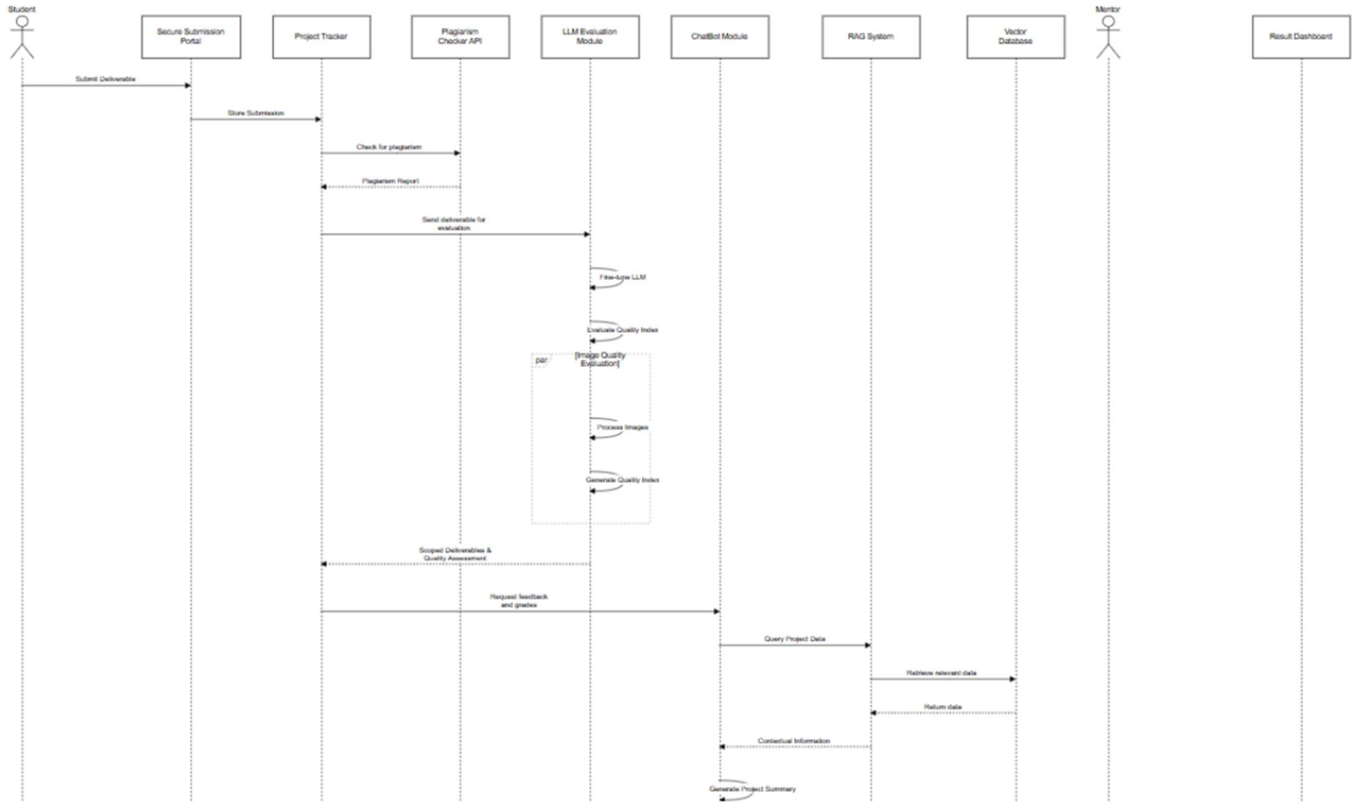
- Output: String (answer)
- Exceptions: Retrieval or parsing errors

3.2.3.28 Methods

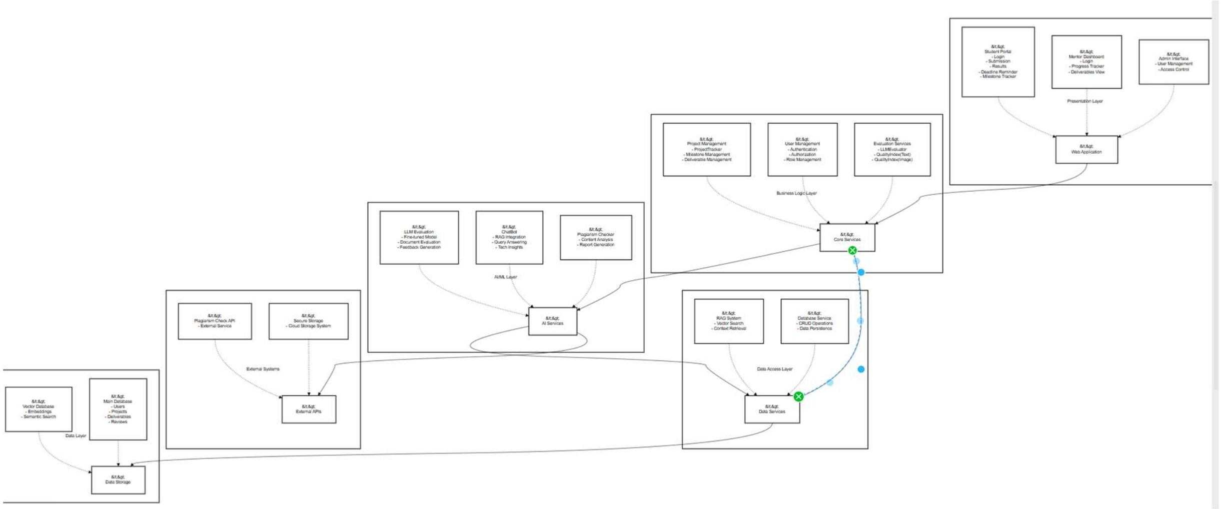
Method: answerQuery()

- Purpose: Return a concise answer to a user question.
- Parameters: String query
- Output: String (answer)
- Exceptions: Retrieval or parsing errors

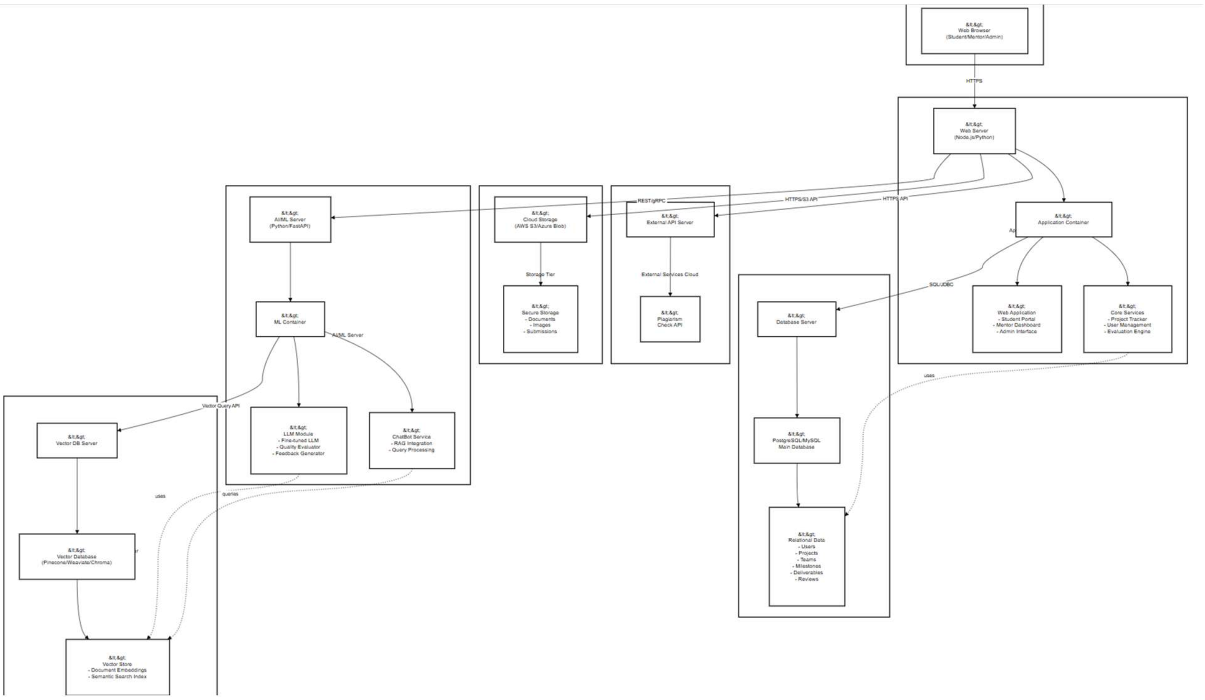
4.2.4. Sequence Diagram



Packaging Diagrams



Deployment Diagrams



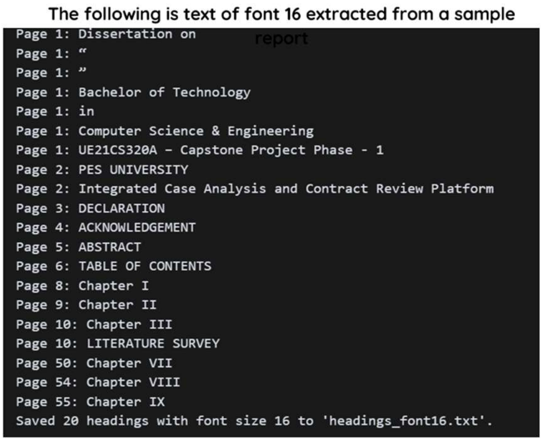
5. Proposed Methodology / Approach

4.1 Algorithm and Pseudocode

Our approach to the Capstone Tracker is centered on a sophisticated, multi-layered framework designed to automatically evaluate project deliverables, integrating traditional metrics with advanced capabilities. A core component is our **custom weighted quality index**, which we use to scientifically assess whether a submitted document meets the required standard. This index is calculated from the normalized scores of several metrics: **Indecisiveness Index** (0.35), **Automated Readability Index (ARI)** (0.3), **Lexical Density** (0.25), and the **Gunning Fog Index (GFI)** (0.2). We deliberately favored the ARI over the GFI to appropriately account for their overlap, as ARI penalizes based on the number of characters per word rather than syllables. Additionally, we employ a **rule-based scoring** mechanism to intelligently combine the base recommendation scores, a method that consistently reflected an underlying pattern present through all years, unlike a simple average. The most advanced component is the use of **Aspect-Based Sentiment Analysis (ABSA)**, which is crucial because peer reviews are complex and multi-faceted, evaluating papers across multiple dimensions. We adopted the holistic reviewing aspects from **ACL conferences** (covering soundness, originality, and clarity) to quantify a **semantic score** from the peer review texts. Instead of traditional lexicon-based methods, we are leveraging modern **Large Language Models (LLMs)** like **Deepseek V2** and **Llama 3.1 8B**, using prompt engineering to perform ABSA and generate semantic-level scores for both **technical soundness** and **novelty** from a dataset of 35,000 text reviews. Ultimately, we **combine** the scores derived from the Rule-Based Scoring, ABSA Semantic Scoring, and the Weighted Quality Index to generate the final, holistic **Composite Quality Score** for the submitted deliverable.

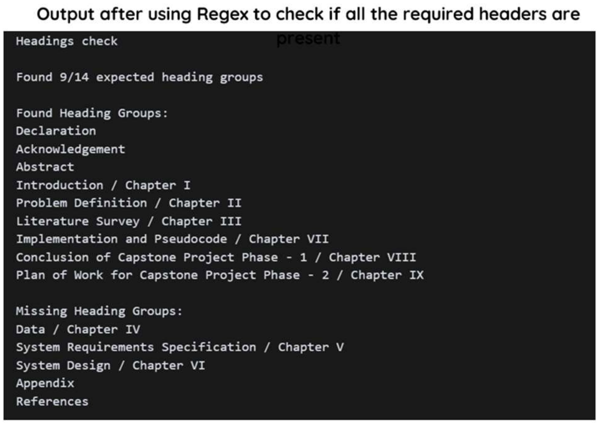
4.1 Implementation and Results

Parser



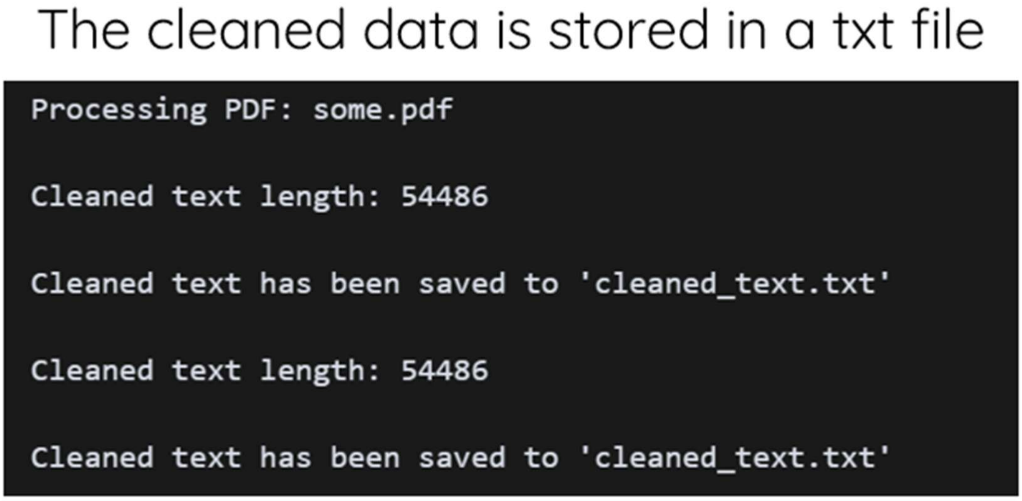
The following is text of font 16 extracted from a sample report

```
Page 1: Dissertation on
Page 1: “
Page 1: ”
Page 1: Bachelor of Technology
Page 1: in
Page 1: Computer Science & Engineering
Page 1: UE21CS320A – Capstone Project Phase - 1
Page 2: PES UNIVERSITY
Page 2: Integrated Case Analysis and Contract Review Platform
Page 3: DECLARATION
Page 4: ACKNOWLEDGEMENT
Page 5: ABSTRACT
Page 6: TABLE OF CONTENTS
Page 8: Chapter I
Page 9: Chapter II
Page 10: Chapter III
Page 10: LITERATURE SURVEY
Page 50: Chapter VII
Page 54: Chapter VIII
Page 55: Chapter IX
Saved 20 headings with font size 16 to 'headings_font16.txt'.
```



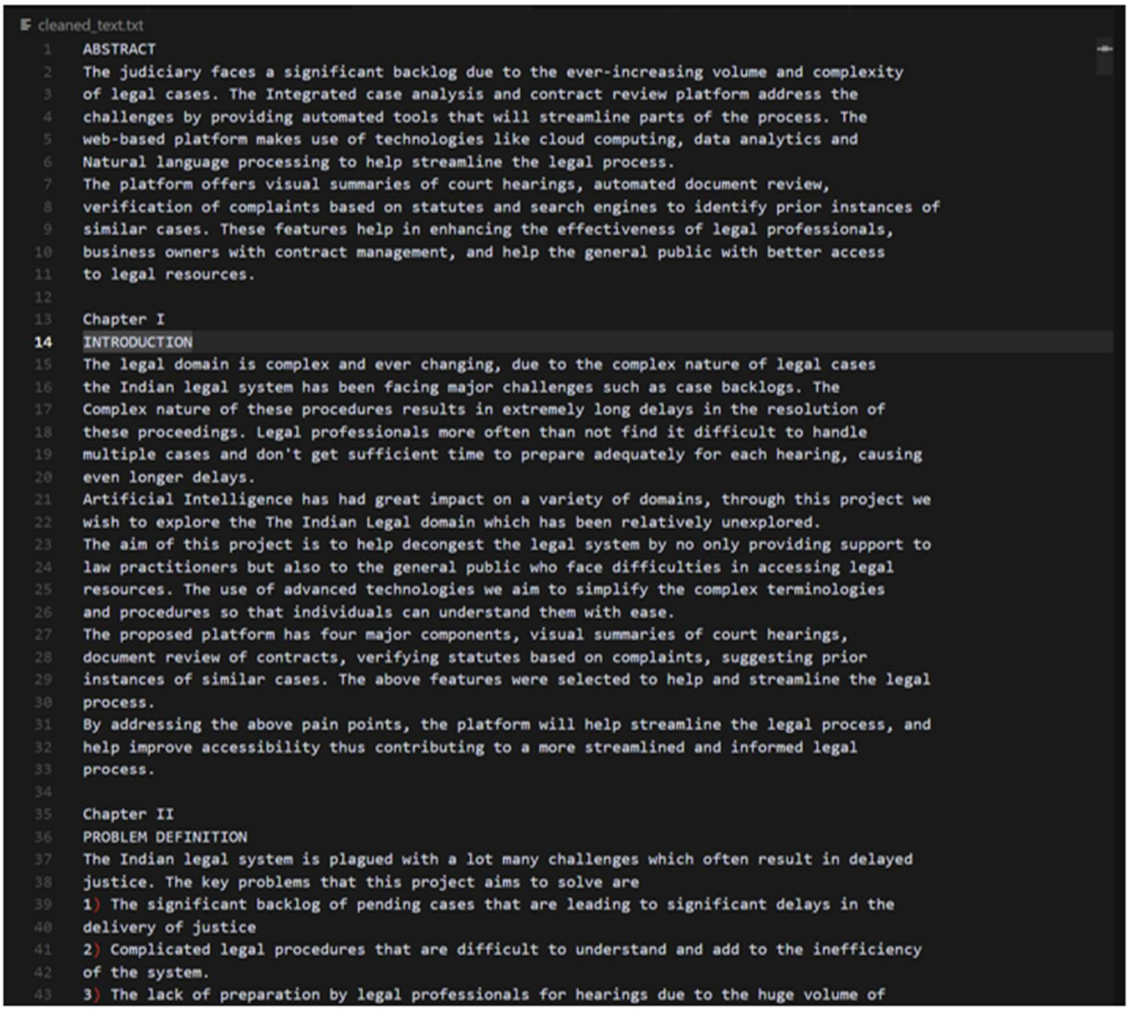
Output after using Regex to check if all the required headers are present

```
Headings check

Found 9/14 expected heading groups

Found Heading Groups:
Declaration
Acknowledgement
Abstract
Introduction / Chapter I
Problem Definition / Chapter II
Literature Survey / Chapter III
Implementation and Pseudocode / Chapter VII
Conclusion of Capstone Project Phase - 1 / Chapter VIII
Plan of Work for Capstone Project Phase - 2 / Chapter IX

Missing Heading Groups:
Data / Chapter IV
System Requirements Specification / Chapter V
System Design / Chapter VI
Appendix
References
```



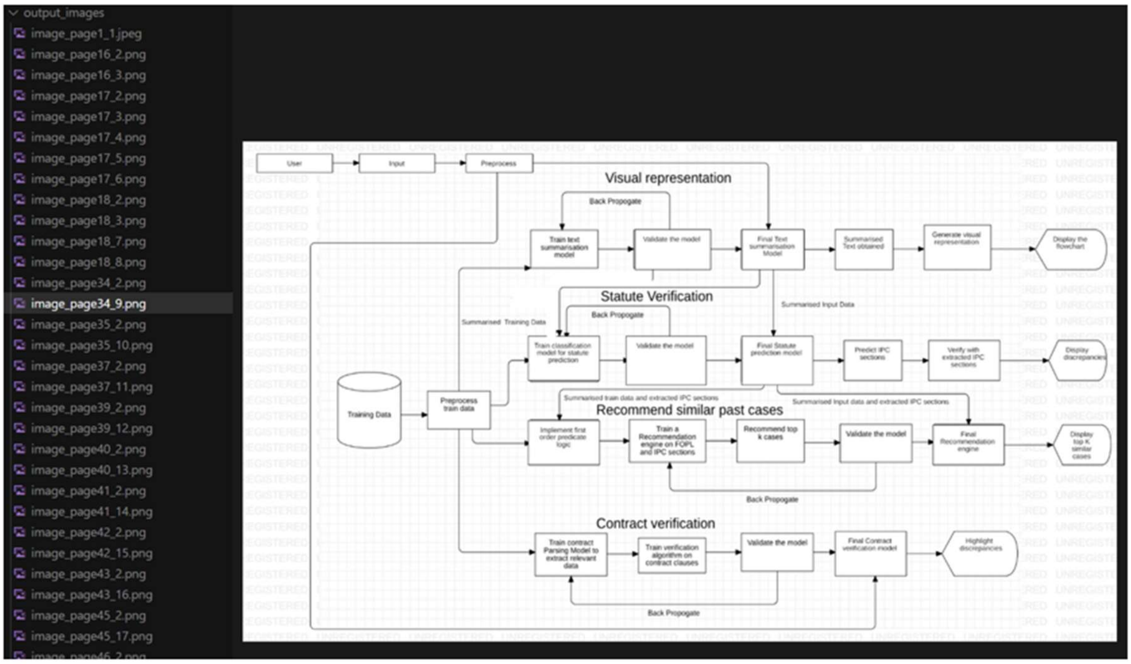
The cleaned data is stored in a txt file

```
Processing PDF: some.pdf

Cleaned text length: 54486

Cleaned text has been saved to 'cleaned_text.txt'

Cleaned text length: 54486

Cleaned text has been saved to 'cleaned_text.txt'
```

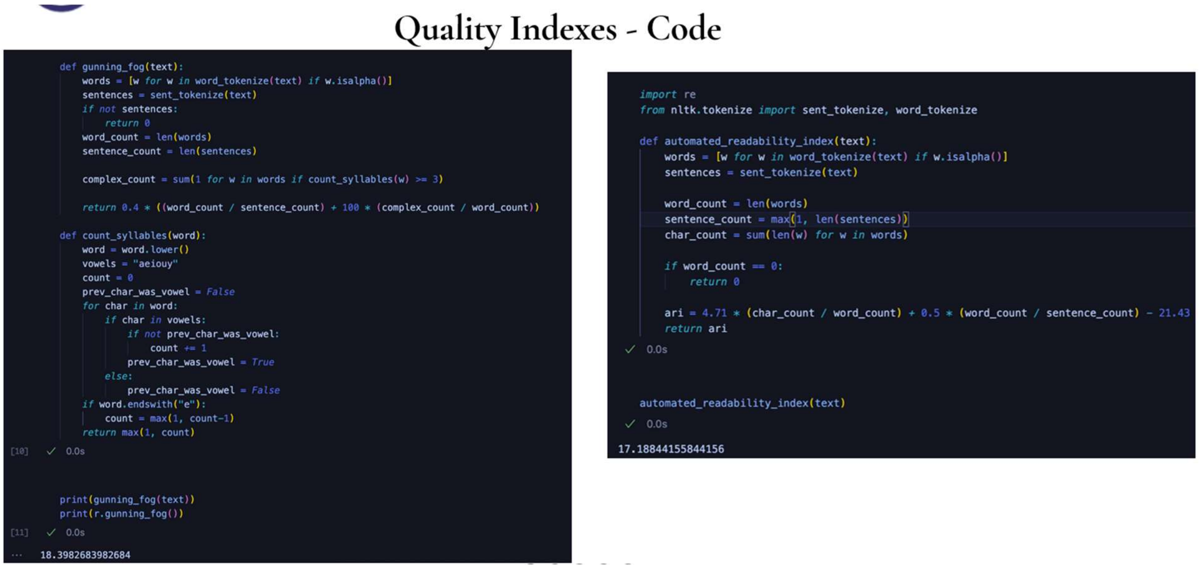


```
ABSTRACT
The judiciary faces a significant backlog due to the ever-increasing volume and complexity
of legal cases. The Integrated case analysis and contract review platform address the
challenges by providing automated tools that will streamline parts of the process. The
web-based platform makes use of technologies like cloud computing, data analytics and
Natural language processing to help streamline the legal process.
The platform offers visual summaries of court hearings, automated document review,
verification of complaints based on statutes and search engines to identify prior instances of
similar cases. These features help in enhancing the effectiveness of legal professionals,
business owners with contract management, and help the general public with better access
to legal resources.

Chapter I
INTRODUCTION
The legal domain is complex and ever changing, due to the complex nature of legal cases
the Indian legal system has been facing major challenges such as case backlogs. The
Complex nature of these procedures results in extremely long delays in the resolution of
these proceedings. Legal professionals more often than not find it difficult to handle
multiple cases and don't get sufficient time to prepare adequately for each hearing, causing
even longer delays.
Artificial Intelligence has had great impact on a variety of domains, through this project we
wish to explore the The Indian Legal domain which has been relatively unexplored.
The aim of this project is to help decongest the legal system by no only providing support to
law practitioners but also to the general public who face difficulties in accessing legal
resources. The use of advanced technologies we aim to simplify the complex terminologies
and procedures so that individuals can understand them with ease.
The proposed platform has four major components, visual summaries of court hearings,
document review of contracts, verifying statutes based on complaints, suggesting prior
instances of similar cases. The above features were selected to help and streamline the legal
process.
By addressing the above pain points, the platform will help streamline the legal process, and
help improve accessibility thus contributing to a more streamlined and informed legal
process.

Chapter II
PROBLEM DEFINITION
The Indian legal system is plagued with a lot many challenges which often result in delayed
justice. The key problems that this project aims to solve are
1) The significant backlog of pending cases that are leading to significant delays in the
delivery of justice
2) Complicated legal procedures that are difficult to understand and add to the inefficiency
of the system.
3) The lack of preparation by legal professionals for hearings due to the huge volume of
```

Clip model



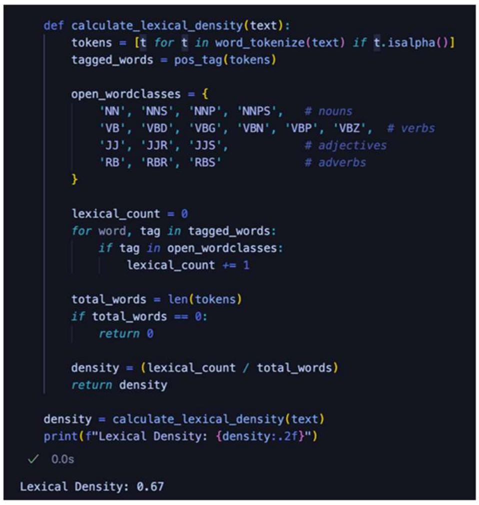
Quality Index


Quality Indexes - Code

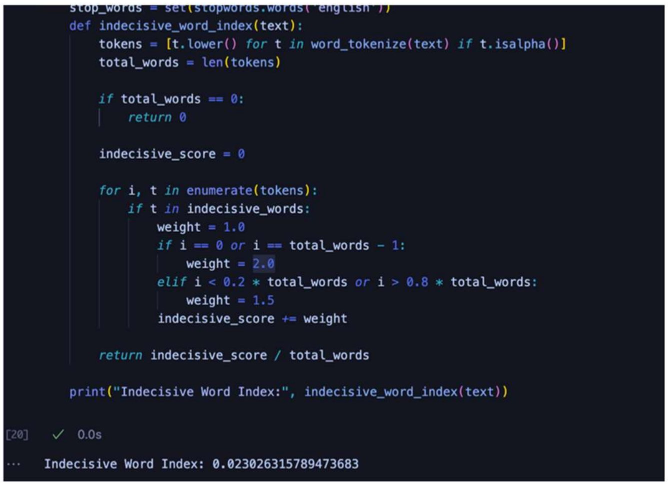
```
def gunning_fog(text):
    words = [w for w in word_tokenize(text) if w.isalpha()]
    sentences = sent_tokenize(text)
    if not sentences:
        return 0
    word_count = len(words)
    sentence_count = len(sentences)

    complex_count = sum(1 for w in words if count_syllables(w) >= 3)

    return 0.4 * ((word_count / sentence_count) + 100 * (complex_count / word_count))

def count_syllables(word):
    word = word.lower()
    vowels = "aeiouy"
    count = 0
    prev_char_was_vowel = False
    for char in word:
        if char in vowels:
            if not prev_char_was_vowel:
                count += 1
            prev_char_was_vowel = True
        else:
            prev_char_was_vowel = False
    if word.endswith("e"):
        count = max(1, count-1)
    return max(1, count)
[10]  ✓ 0.0s


print(gunning_fog(text))
print(r.gunning_fog())
[11]  ✓ 0.0s
...  18.3982683982684
```

```
import re
from nltk.tokenize import sent_tokenize, word_tokenize

def automated_readability_index(text):
    words = [w for w in word_tokenize(text) if w.isalpha()]
    sentences = sent_tokenize(text)

    word_count = len(words)
    sentence_count = max(1, len(sentences))
    char_count = sum(len(w) for w in words)

    if word_count == 0:
        return 0

    ari = 4.71 * (char_count / word_count) + 0.5 * (word_count / sentence_count) - 21.43
    return ari
✓ 0.0s


automated_readability_index(text)
✓ 0.0s
17.18844155844156
```

Quality Indexes - Code

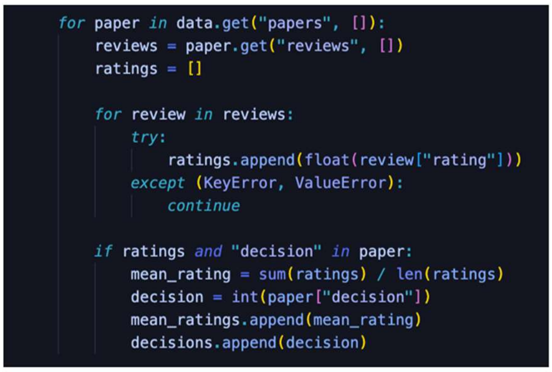


```
def calculate_lexical_density(text):
    tokens = [t for t in word_tokenize(text) if t.isalpha()]
    tagged_words = pos_tag(tokens)

    open_wordclasses = {
        'NN', 'NNS', 'NNP', 'NNPS',    # nouns
        'VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ',  # verbs
        'JJ', 'JJR', 'JJS',            # adjectives
        'RB', 'RBR', 'RBS'             # adverbs
    }

    lexical_count = 0
    for word, tag in tagged_words:
        if tag in open_wordclasses:
            lexical_count += 1

    total_words = len(tokens)
    if total_words == 0:
        return 0

    density = (lexical_count / total_words)
    return density

density = calculate_lexical_density(text)
print(f"Lexical Density: {density:.2f}")
 ✓ 0.0s

Lexical Density: 0.67
```

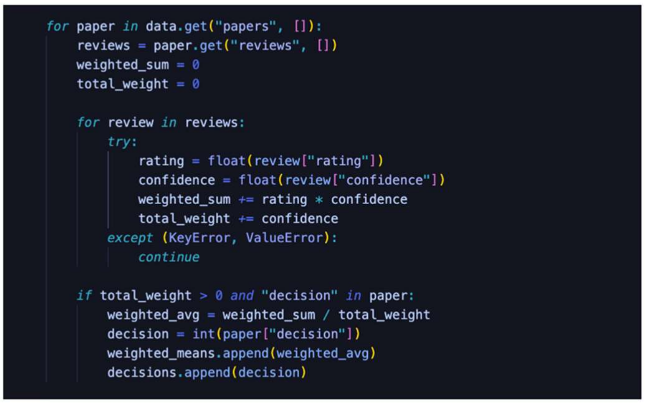


```
stop_words = set(stopwords.words('english'))
def indecisive_word_index(text):
    tokens = [t.lower() for t in word_tokenize(text) if t.isalpha()]
    total_words = len(tokens)

    if total_words == 0:
        return 0

    indecisive_score = 0

    for i, t in enumerate(tokens):
        if t in indecisive_words:
            weight = 1.0
            if i == 0 or i == total_words - 1:
                weight = 2.0
            elif i < 0.2 * total_words or i > 0.8 * total_words:
                weight = 1.5
            indecisive_score += weight

    return indecisive_score / total_words

print("Indecisive Word Index:", indecisive_word_index(text))

[20]  ✓ 0.0s

...  Indecisive Word Index: 0.023026315789473683
```

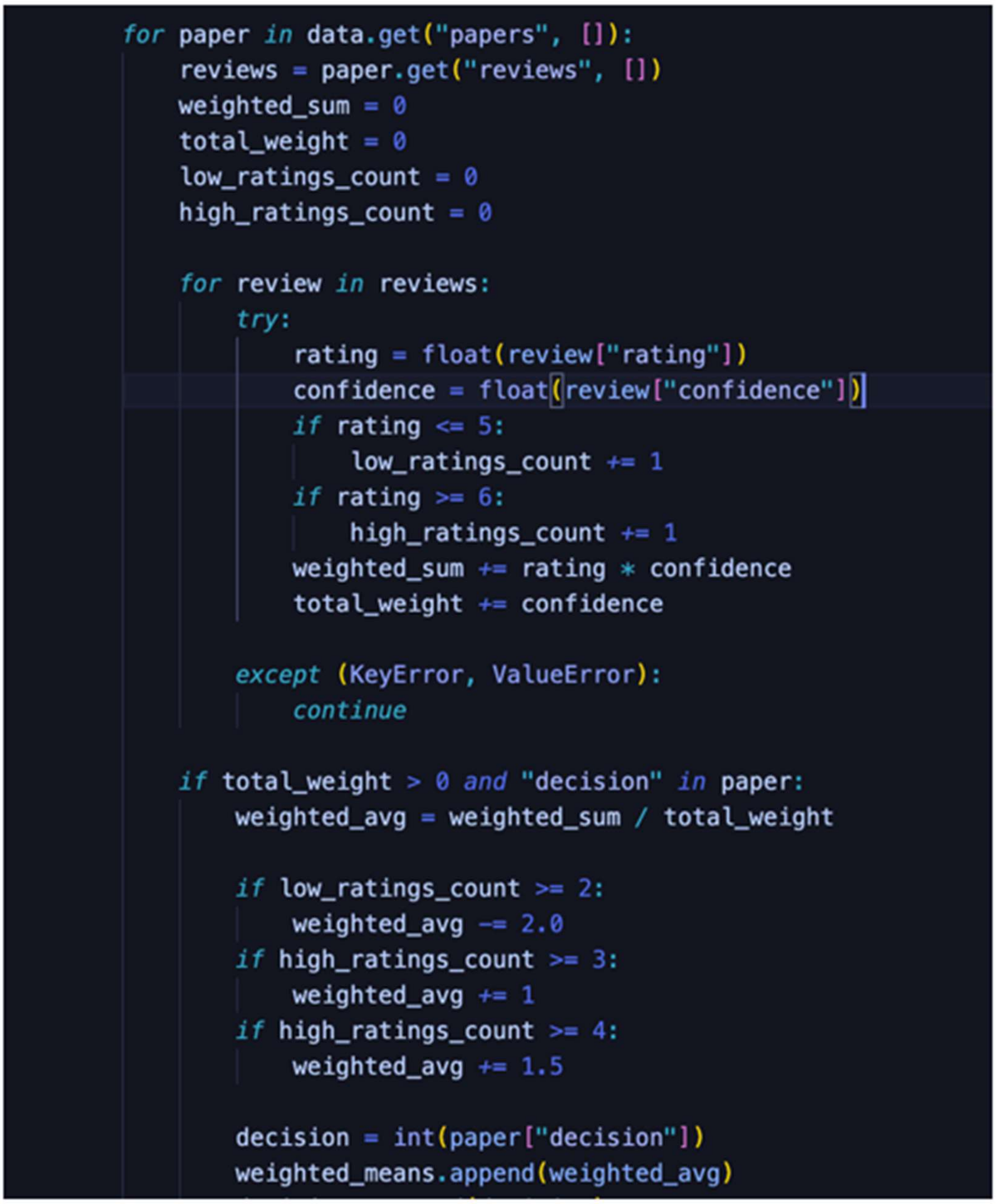
Rule Based Approach



```
for paper in data.get("papers", []):
    reviews = paper.get("reviews", [])
    ratings = []

    for review in reviews:
        try:
            ratings.append(float(review["rating"]))
        except (KeyError, ValueError):
            continue

    if ratings and "decision" in paper:
        mean_rating = sum(ratings) / len(ratings)
        decision = int(paper["decision"])
        mean_ratings.append(mean_rating)
        decisions.append(decision)
```

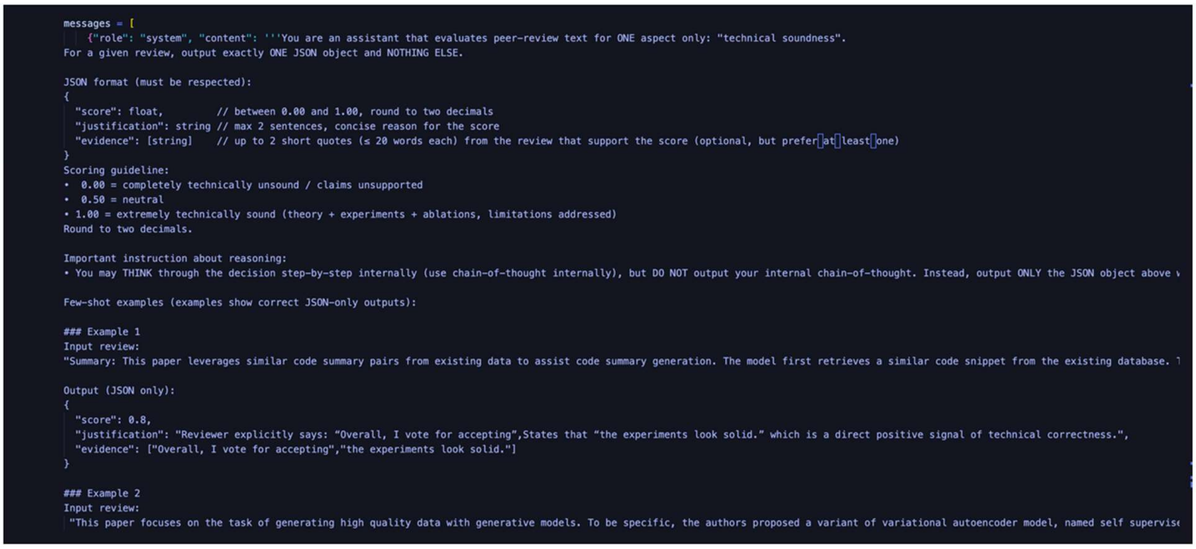


```
for paper in data.get("papers", []):
    reviews = paper.get("reviews", [])
    weighted_sum = 0
    total_weight = 0

    for review in reviews:
        try:
            rating = float(review["rating"])
            confidence = float(review["confidence"])
            weighted_sum += rating * confidence
            total_weight += confidence
        except (KeyError, ValueError):
            continue

    if total_weight > 0 and "decision" in paper:
        weighted_avg = weighted_sum / total_weight
        decision = int(paper["decision"])
        weighted_means.append(weighted_avg)
        decisions.append(decision)
```

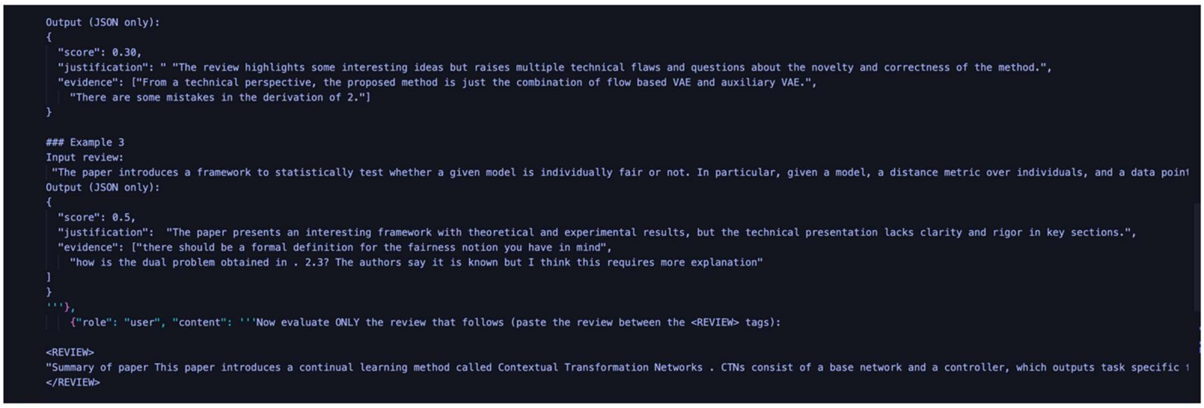


```
for paper in data.get("papers", []):
    reviews = paper.get("reviews", [])
    weighted_sum = 0
    total_weight = 0
    low_ratings_count = 0
    high_ratings_count = 0

    for review in reviews:
        try:
            rating = float(review["rating"])
            confidence = float(review["confidence"])
            if rating <= 5:
                low_ratings_count += 1
            if rating >= 6:
                high_ratings_count += 1
            weighted_sum += rating * confidence
            total_weight += confidence

        except (KeyError, ValueError):
            continue

    if total_weight > 0 and "decision" in paper:
        weighted_avg = weighted_sum / total_weight

        if low_ratings_count >= 2:
            weighted_avg -= 2.0
        if high_ratings_count >= 3:
            weighted_avg += 1
        if high_ratings_count >= 4:
            weighted_avg += 1.5

        decision = int(paper["decision"])
        weighted_means.append(weighted_avg)
```

ABSA on Technical Soundness

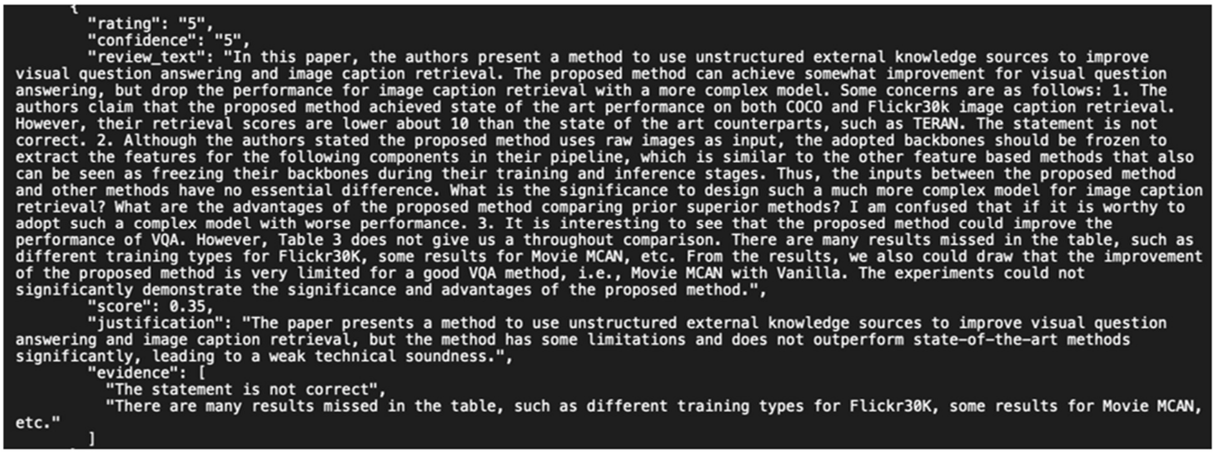


```
messages = [
    {"role": "system", "content": '''You are an assistant that evaluates peer-review text for ONE aspect only: "technical soundness".
For a given review, output exactly ONE JSON object and NOTHING ELSE.

JSON format (must be respected):
{
  "score": float,         // between 0.00 and 1.00, round to two decimals
  "justification": string // max 2 sentences, concise reason for the score
  "evidence": [string]    // up to 2 short quotes (≤ 20 words each) from the review that support the score (optional, but prefer□at□least□one)
}
Scoring guideline:
•  0.00 = completely technically unsound / claims unsupported
•  0.50 = neutral
• 1.00 = extremely technically sound (theory + experiments + ablations, limitations addressed)
Round to two decimals.

Important instruction about reasoning:
• You may THINK through the decision step-by-step internally (use chain-of-thought internally), but DO NOT output your internal chain-of-thought. Instead, output ONLY the JSON object above

Few-shot examples (examples show correct JSON-only outputs):

### Example 1
Input review:
"Summary: This paper leverages similar code summary pairs from existing data to assist code summary generation. The model first retrieves a similar code snippet from the existing database.

Output (JSON only):
{
  "score": 0.8,
  "justification": "Reviewer explicitly says: “Overall, I vote for accepting”,States that “the experiments look solid.” which is a direct positive signal of technical correctness.",
  "evidence": ["Overall, I vote for accepting","the experiments look solid."]
}

### Example 2
Input review:
 "This paper focuses on the task of generating high quality data with generative models. To be specific, the authors proposed a variant of variational autoencoder model, named self supervise
```



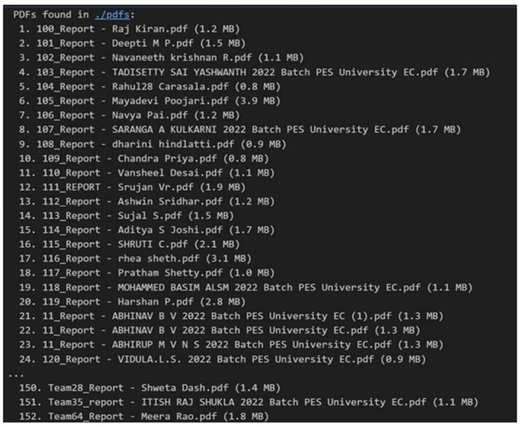
```
Output (JSON only):
{
  "score": 0.30,
  "justification": " "The review highlights some interesting ideas but raises multiple technical flaws and questions about the novelty and correctness of the method.",
  "evidence": ["From a technical perspective, the proposed method is just the combination of flow based VAE and auxiliary VAE.",
    "There are some mistakes in the derivation of 2."]
}

### Example 3
Input review:
 "The paper introduces a framework to statistically test whether a given model is individually fair or not. In particular, given a model, a distance metric over individuals, and a data point
Output (JSON only):
{
  "score": 0.5,
  "justification":  "The paper presents an interesting framework with theoretical and experimental results, but the technical presentation lacks clarity and rigor in key sections.",
  "evidence": ["there should be a formal definition for the fairness notion you have in mind",
    "how is the dual problem obtained in . 2.3? The authors say it is known but I think this requires more explanation"
]
}
'''},
    {"role": "user", "content": '''Now evaluate ONLY the review that follows (paste the review between the <REVIEW> tags):

<REVIEW>
"Summary of paper This paper introduces a continual learning method called Contextual Transformation Networks . CTNs consist of a base network and a controller, which outputs task specific
</REVIEW>
```

Sample Output

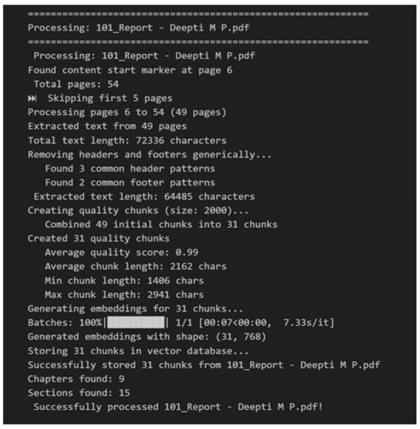


```
      {
        "rating": "5",
        "confidence": "5",
        "review_text": "In this paper, the authors present a method to use unstructured external knowledge sources to improve
visual question answering and image caption retrieval. The proposed method can achieve somewhat improvement for visual question
answering, but drop the performance for image caption retrieval with a more complex model. Some concerns are as follows: 1. The
authors claim that the proposed method achieved state of the art performance on both COCO and Flickr30k image caption retrieval.
However, their retrieval scores are lower about 10 than the state of the art counterparts, such as TERAN. The statement is not
correct. 2. Although the authors stated the proposed method uses raw images as input, the adopted backbones should be frozen to
extract the features for the following components in their pipeline, which is similar to the other feature based methods that also
can be seen as freezing their backbones during their training and inference stages. Thus, the inputs between the proposed method
and other methods have no essential difference. What is the significance to design such a much more complex model for image caption
retrieval? What are the advantages of the proposed method comparing prior superior methods? I am confused that if it is worthy to
adopt such a complex model with worse performance. 3. It is interesting to see that the proposed method could improve the
performance of VQA. However, Table 3 does not give us a throughout comparison. There are many results missed in the table, such as
different training types for Flickr30K, some results for Movie MCAN, etc. From the results, we also could draw that the improvement
of the proposed method is very limited for a good VQA method, i.e., Movie MCAN with Vanilla. The experiments could not
significantly demonstrate the significance and advantages of the proposed method.",
        "score": 0.35,
        "justification": "The paper presents a method to use unstructured external knowledge sources to improve visual question
answering and image caption retrieval, but the method has some limitations and does not outperform state-of-the-art methods
significantly, leading to a weak technical soundness.",
        "evidence": [
          "The statement is not correct",
          "There are many results missed in the table, such as different training types for Flickr30K, some results for Movie MCAN,
etc."
        ]
```

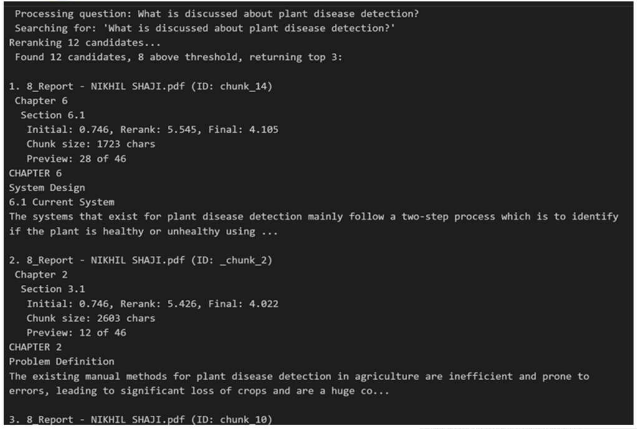
RAG-Based Chatbot



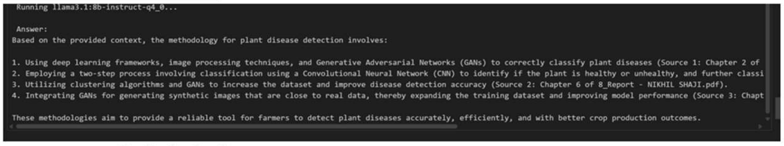
```
PDFs found in ./pdfs:
 1. 100_Report - Raj Kiran.pdf (1.2 MB)
 2. 101_Report - Deepti M P.pdf (1.5 MB)
 3. 102_Report - Navaneeth krishnan R.pdf (1.1 MB)
 4. 103_Report - TADISETTY SAI YASHWANTH 2022 Batch PES University EC.pdf (1.7 MB)
 5. 104_Report - Rahul28 Carasala.pdf (0.8 MB)
 6. 105_Report - Mayadevi Poojari.pdf (3.9 MB)
 7. 106_Report - Navya Pai.pdf (1.2 MB)
 8. 107_Report - SARANGA A KULKARNI 2022 Batch PES University EC.pdf (1.7 MB)
 9. 108_Report - dharini hindlatti.pdf (0.9 MB)
 10. 109_Report - Chandra Priya.pdf (0.8 MB)
 11. 110_Report - Vansheel Desai.pdf (1.1 MB)
 12. 111_REPORT - Srujan Vr.pdf (1.9 MB)
 13. 112_Report - Ashwin Sridhar.pdf (1.2 MB)
 14. 113_Report - Sujal S.pdf (1.5 MB)
 15. 114_Report - Aditya S Joshi.pdf (1.7 MB)
 16. 115_Report - SHRUTI C.pdf (2.1 MB)
 17. 116_Report - rhea sheth.pdf (3.1 MB)
 18. 117_Report - Pratham Shetty.pdf (1.0 MB)
 19. 118_Report - MOHAMMED BASIM ALSM 2022 Batch PES University EC.pdf (1.1 MB)
 20. 119_Report - Harshan P.pdf (2.8 MB)
 21. 11_Report - ABHINAV B V 2022 Batch PES University EC (1).pdf (1.3 MB)
 22. 11_Report - ABHINAV B V 2022 Batch PES University EC.pdf (1.3 MB)
 23. 11_Report - ABHIRUP M V N S 2022 Batch PES University EC.pdf (1.3 MB)
 24. 120_Report - VIDULA.L.S. 2022 Batch PES University EC.pdf (0.9 MB)
...
 150. Team28_Report - Shweta Dash.pdf (1.4 MB)
 151. Team35_report - ITISH RAJ SHUKLA 2022 Batch PES University EC.pdf (1.1 MB)
 152. Team64_Report - Meera Rao.pdf (1.8 MB)
```



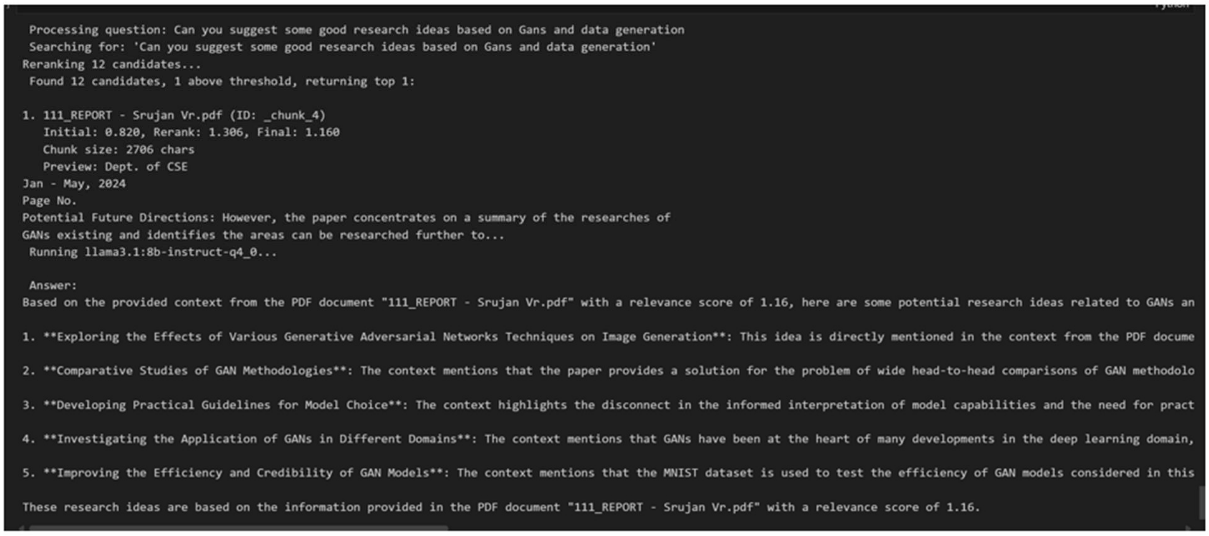
```
============================================================
Processing: 101_Report - Deepti M P.pdf
============================================================
 Processing: 101_Report - Deepti M P.pdf
Found content start marker at page 6
 Total pages: 54
⏭  Skipping first 5 pages
Processing pages 6 to 54 (49 pages)
Extracted text from 49 pages
Total text length: 72336 characters
Removing headers and footers generically...
   Found 3 common header patterns
   Found 2 common footer patterns
 Extracted text length: 64485 characters
Creating quality chunks (size: 2000)...
   Combined 49 initial chunks into 31 chunks
Created 31 quality chunks
   Average quality score: 0.99
   Average chunk length: 2162 chars
   Min chunk length: 1406 chars
   Max chunk length: 2941 chars
Generating embeddings for 31 chunks...
Batches: 100%|██████████| 1/1 [00:07<00:00,  7.33s/it]
Generated embeddings with shape: (31, 768)
Storing 31 chunks in vector database...
Successfully stored 31 chunks from 101_Report - Deepti M P.pdf
Chapters found: 9
Sections found: 15
 Successfully processed 101_Report - Deepti M P.pdf!
```



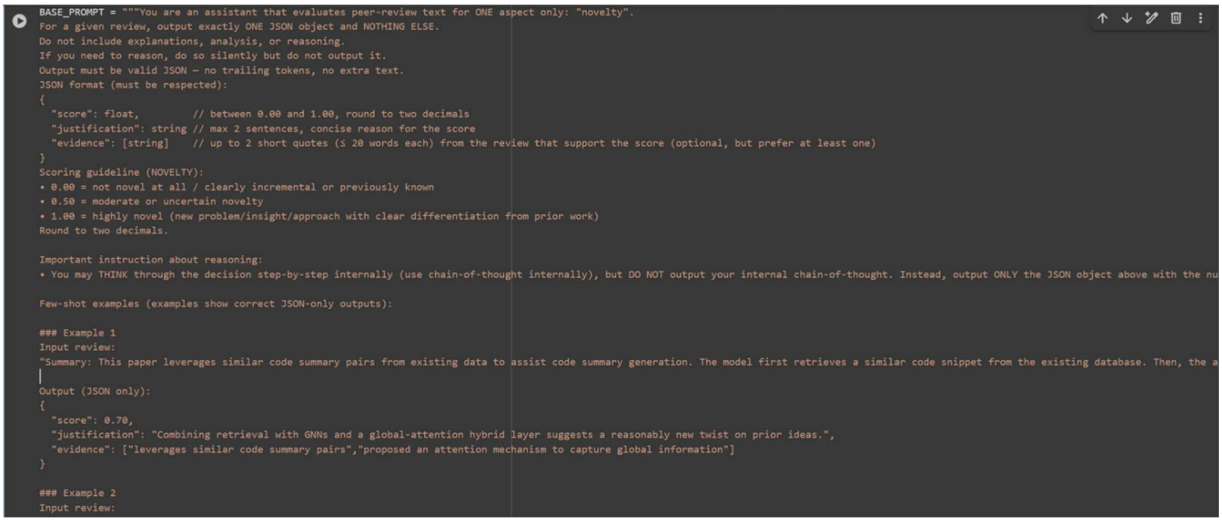
```
 Processing question: What is discussed about plant disease detection?
 Searching for: 'What is discussed about plant disease detection?'
Reranking 12 candidates...
 Found 12 candidates, 8 above threshold, returning top 3:

1. 8_Report - NIKHIL SHAJI.pdf (ID: chunk_14)
 Chapter 6
  Section 6.1
   Initial: 0.746, Rerank: 5.545, Final: 4.105
   Chunk size: 1723 chars
   Preview: 28 of 46
CHAPTER 6
System Design
6.1 Current System
The systems that exist for plant disease detection mainly follow a two-step process which is to identify
if the plant is healthy or unhealthy using ...

2. 8_Report - NIKHIL SHAJI.pdf (ID: _chunk_2)
 Chapter 2
  Section 3.1
   Initial: 0.746, Rerank: 5.426, Final: 4.022
   Chunk size: 2603 chars
   Preview: 12 of 46
CHAPTER 2
Problem Definition
The existing manual methods for plant disease detection in agriculture are inefficient and prone to
errors, leading to significant loss of crops and are a huge co...

3. 8_Report - NIKHIL SHAJI.pdf (ID: chunk_10)
```



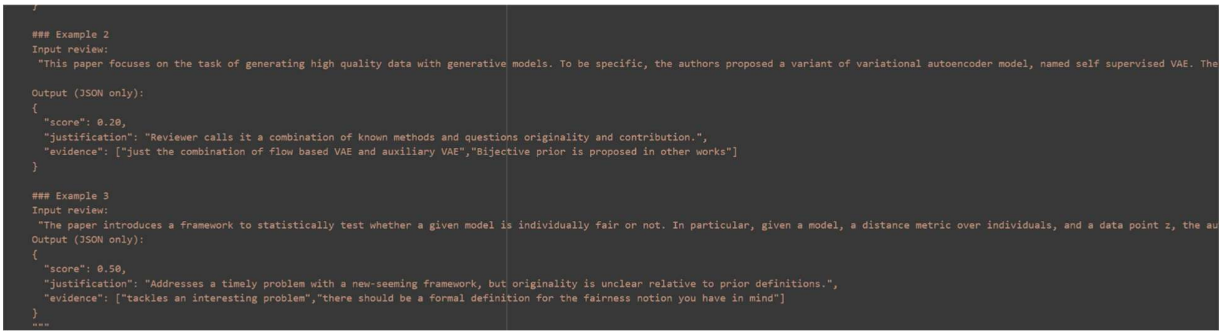
```
 Running llama3.1:8b-instruct-q4_0...

 Answer:
Based on the provided context, the methodology for plant disease detection involves:

1. Using deep learning frameworks, image processing techniques, and Generative Adversarial Networks (GANs) to correctly classify plant diseases (Source 1: Chapter 2 of
2. Employing a two-step process involving classification using a Convolutional Neural Network (CNN) to identify if the plant is healthy or unhealthy, and further classi
3. Utilizing clustering algorithms and GANs to increase the dataset and improve disease detection accuracy (Source 2: Chapter 6 of 8_Report - NIKHIL SHAJI.pdf).
4. Integrating GANs for generating synthetic images that are close to real data, thereby expanding the training dataset and improving model performance (Source 3: Chapt

These methodologies aim to provide a reliable tool for farmers to detect plant diseases accurately, efficiently, and with better crop production outcomes.
```



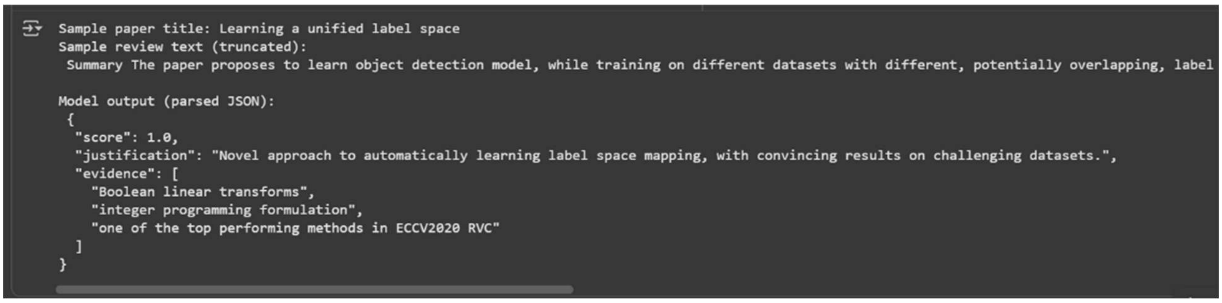
```
 Processing question: Can you suggest some good research ideas based on Gans and data generation
 Searching for: 'Can you suggest some good research ideas based on Gans and data generation'
Reranking 12 candidates...
 Found 12 candidates, 1 above threshold, returning top 1:

1. 111_REPORT - Srujan Vr.pdf (ID: _chunk_4)
   Initial: 0.820, Rerank: 1.306, Final: 1.160
   Chunk size: 2706 chars
   Preview: Dept. of CSE
Jan - May, 2024
Page No.
Potential Future Directions: However, the paper concentrates on a summary of the researches of
GANs existing and identifies the areas can be researched further to...
 Running llama3.1:8b-instruct-q4_0...

 Answer:
Based on the provided context from the PDF document "111_REPORT - Srujan Vr.pdf" with a relevance score of 1.16, here are some potential research ideas related to GANs an

1. **Exploring the Effects of Various Generative Adversarial Networks Techniques on Image Generation**: This idea is directly mentioned in the context from the PDF docume

2. **Comparative Studies of GAN Methodologies**: The context mentions that the paper provides a solution for the problem of wide head-to-head comparisons of GAN methodolo

3. **Developing Practical Guidelines for Model Choice**: The context highlights the disconnect in the informed interpretation of model capabilities and the need for pract

4. **Investigating the Application of GANs in Different Domains**: The context mentions that GANs have been at the heart of many developments in the deep learning domain,

5. **Improving the Efficiency and Credibility of GAN Models**: The context mentions that the MNIST dataset is used to test the efficiency of GAN models considered in this

These research ideas are based on the information provided in the PDF document "111_REPORT - Srujan Vr.pdf" with a relevance score of 1.16.
```



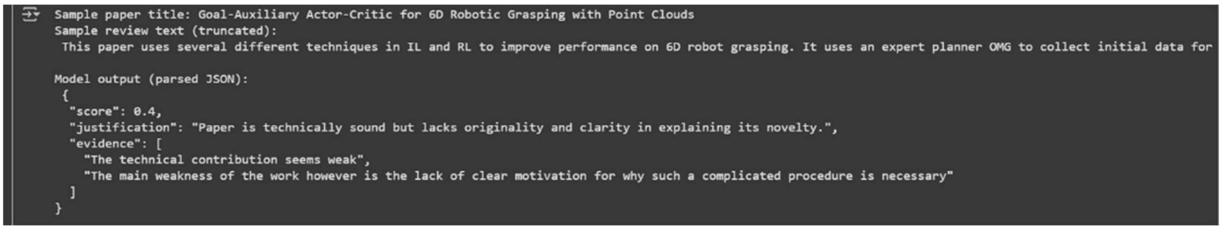
```
BASE_PROMPT = """You are an assistant that evaluates peer-review text for ONE aspect only: "novelty".
For a given review, output exactly ONE JSON object and NOTHING ELSE.
Do not include explanations, analysis, or reasoning.
If you need to reason, do so silently but do not output it.
Output must be valid JSON — no trailing tokens, no extra text.
JSON format (must be respected):
{
  "score": float,         // between 0.00 and 1.00, round to two decimals
  "justification": string // max 2 sentences, concise reason for the score
  "evidence": [string]    // up to 2 short quotes (≤ 20 words each) from the review that support the score (optional, but prefer at least one)
}
Scoring guideline (NOVELTY):
• 0.00 = not novel at all / clearly incremental or previously known
• 0.50 = moderate or uncertain novelty
• 1.00 = highly novel (new problem/insight/approach with clear differentiation from prior work)
Round to two decimals.

Important instruction about reasoning:
• You may THINK through the decision step-by-step internally (use chain-of-thought internally), but DO NOT output your internal chain-of-thought. Instead, output ONLY the JSON object above with the nu

Few-shot examples (examples show correct JSON-only outputs):

### Example 1
Input review:
"Summary: This paper leverages similar code summary pairs from existing data to assist code summary generation. The model first retrieves a similar code snippet from the existing database. Then, the a

Output (JSON only):
{
  "score": 0.70,
  "justification": "Combining retrieval with GNNs and a global-attention hybrid layer suggests a reasonably new twist on prior ideas.",
  "evidence": ["leverages similar code summary pairs","proposed an attention mechanism to capture global information"]
}

### Example 2
Input review:
```



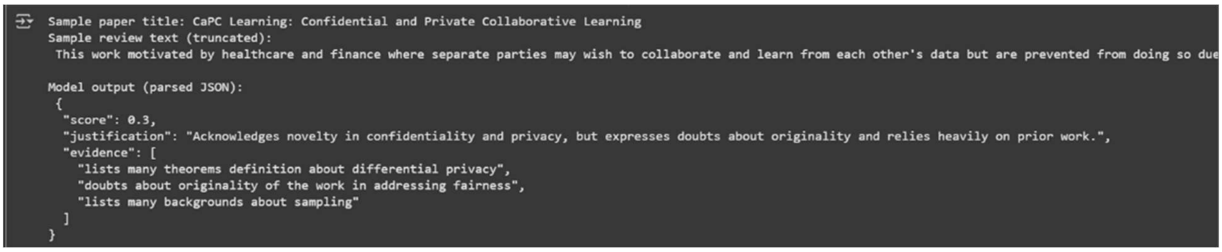
```
}

### Example 2
Input review:
 "This paper focuses on the task of generating high quality data with generative models. To be specific, the authors proposed a variant of variational autoencoder model, named self supervised VAE. The

Output (JSON only):
{
  "score": 0.20,
  "justification": "Reviewer calls it a combination of known methods and questions originality and contribution.",
  "evidence": ["just the combination of flow based VAE and auxiliary VAE","Bijective prior is proposed in other works"]
}

### Example 3
Input review:
 "The paper introduces a framework to statistically test whether a given model is individually fair or not. In particular, given a model, a distance metric over individuals, and a data point z, the au
Output (JSON only):
{
  "score": 0.50,
  "justification": "Addresses a timely problem with a new-seeming framework, but originality is unclear relative to prior definitions.",
  "evidence": ["tackles an interesting problem","there should be a formal definition for the fairness notion you have in mind"]
}
"""
```



```
Sample paper title: Learning a unified label space
Sample review text (truncated):
 Summary The paper proposes to learn object detection model, while training on different datasets with different, potentially overlapping, label

Model output (parsed JSON):
 {
  "score": 1.0,
  "justification": "Novel approach to automatically learning label space mapping, with convincing results on challenging datasets.",
  "evidence": [
    "Boolean linear transforms",
    "integer programming formulation",
    "one of the top performing methods in ECCV2020 RVC"
  ]
 }
```

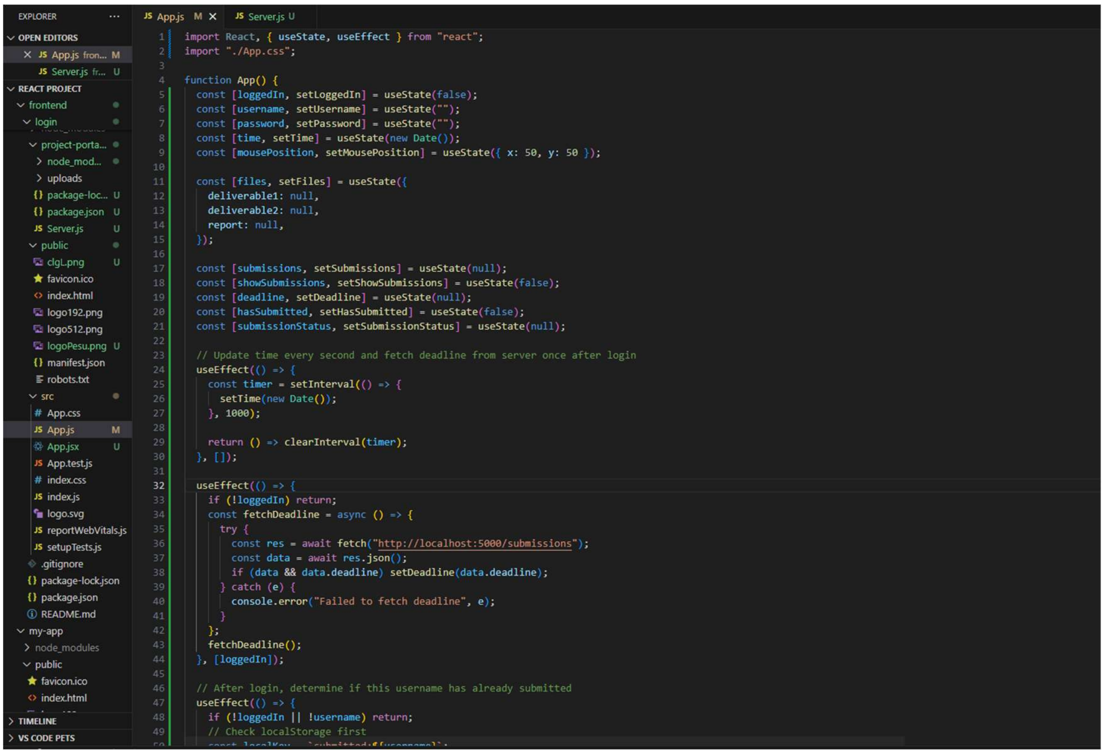


```
Sample paper title: Goal-Auxiliary Actor-Critic for 6D Robotic Grasping with Point Clouds
Sample review text (truncated):
 This paper uses several different techniques in IL and RL to improve performance on 6D robot grasping. It uses an expert planner OMG to collect initial data for

Model output (parsed JSON):
 {
  "score": 0.4,
  "justification": "Paper is technically sound but lacks originality and clarity in explaining its novelty.",
  "evidence": [
    "The technical contribution seems weak",
    "The main weakness of the work however is the lack of clear motivation for why such a complicated procedure is necessary"
  ]
 }
```

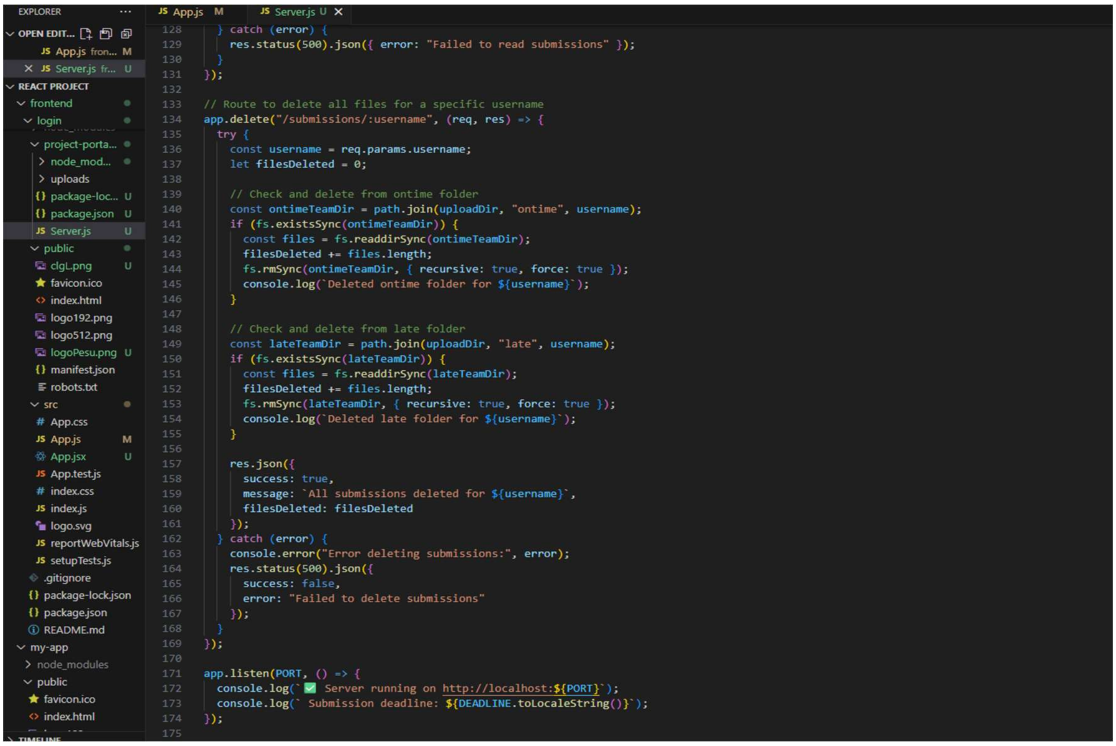


```
Sample paper title: CaPC Learning: Confidential and Private Collaborative Learning
Sample review text (truncated):
 This work motivated by healthcare and finance where separate parties may wish to collaborate and learn from each other's data but are prevented from doing so due

Model output (parsed JSON):
 {
  "score": 0.3,
  "justification": "Acknowledges novelty in confidentiality and privacy, but expresses doubts about originality and relies heavily on prior work.",
  "evidence": [
    "lists many theorems definition about differential privacy",
    "doubts about originality of the work in addressing fairness",
    "lists many backgrounds about sampling"
  ]
 }
```

Frontend Code

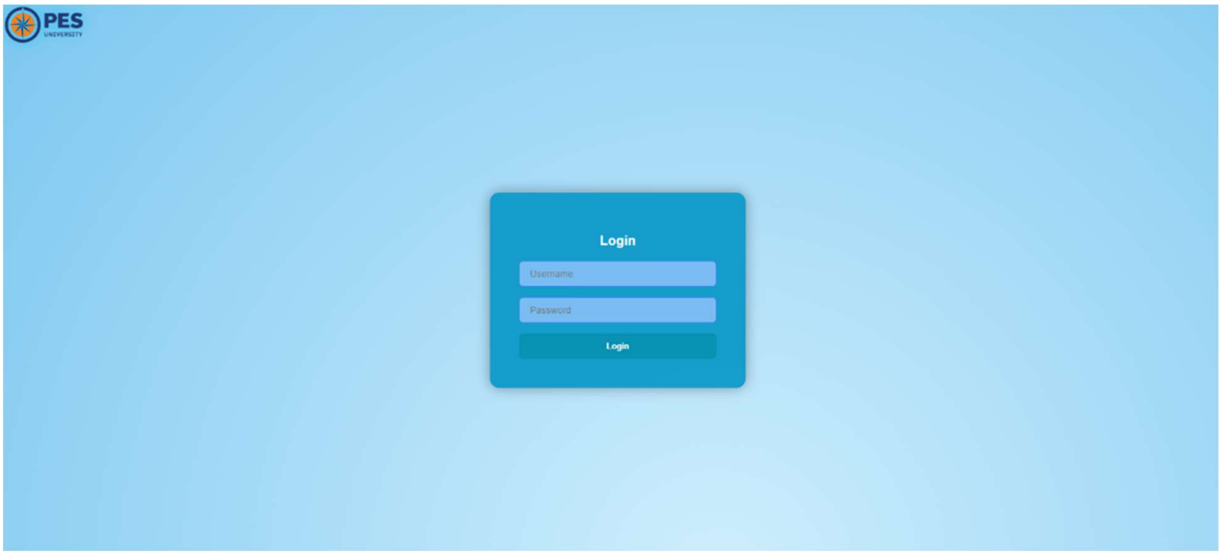


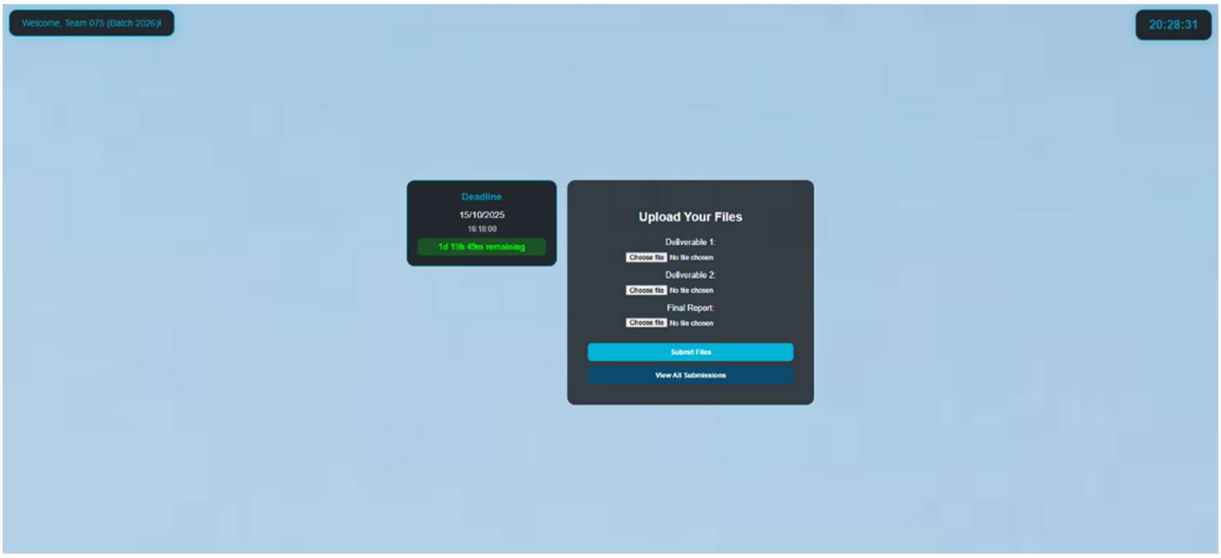
```javascript
import React, { useState, useEffect } from "react";
import "./App.css";

function App() {
  const [loggedIn, setLoggedIn] = useState(false);
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [time, setTime] = useState(new Date());
  const [mousePosition, setMousePosition] = useState({ x: 50, y: 50 });

  const [files, setFiles] = useState({
    deliverable1: null,
    deliverable2: null,
    report: null,
  });

  const [submissions, setSubmissions] = useState(null);
  const [showSubmissions, setShowSubmissions] = useState(false);
  const [deadline, setDeadline] = useState(null);
  const [hasSubmitted, setHasSubmitted] = useState(false);
  const [submissionStatus, setSubmissionStatus] = useState(null);

  // Update time every second and fetch deadline from server once after login
  useEffect(() => {
    const timer = setInterval(() => {
      setTime(new Date());
    }, 1000);

    return () => clearInterval(timer);
  }, []);

  useEffect(() => {
    if (!loggedIn) return;
    const fetchDeadline = async () => {
      try {
        const res = await fetch("http://localhost:5000/submissions");
        const data = await res.json();
        if (data && data.deadline) setDeadline(data.deadline);
      } catch (e) {
        console.error("Failed to fetch deadline", e);
      }
    };
    fetchDeadline();
  }, [loggedIn]);

  // After login, determine if this username has already submitted
  useEffect(() => {
    if (!loggedIn || !username) return;
    // Check localStorage first
```

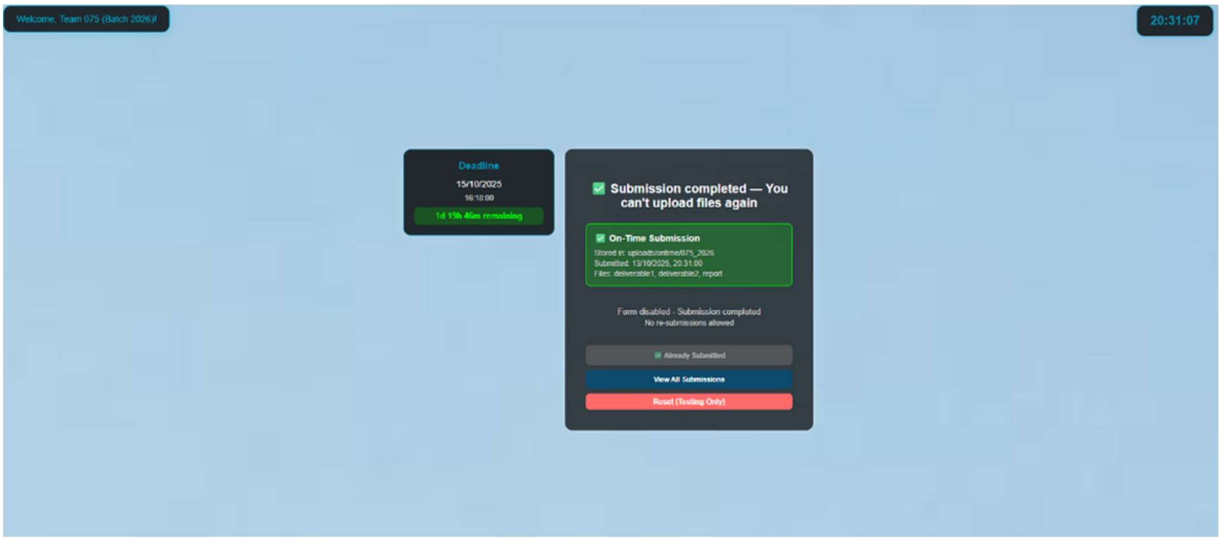


```javascript
  } catch (error) {
    res.status(500).json({ error: "Failed to read submissions" });
  }
});

// Route to delete all files for a specific username
app.delete("/submissions/:username", (req, res) => {
  try {
    const username = req.params.username;
    let filesDeleted = 0;

    // Check and delete from ontime folder
    const ontimeTeamDir = path.join(uploadDir, "ontime", username);
    if (fs.existsSync(ontimeTeamDir)) {
      const files = fs.readdirSync(ontimeTeamDir);
      filesDeleted += files.length;
      fs.rmSync(ontimeTeamDir, { recursive: true, force: true });
      console.log(`Deleted ontime folder for ${username}`);
    }

    // Check and delete from late folder
    const lateTeamDir = path.join(uploadDir, "late", username);
    if (fs.existsSync(lateTeamDir)) {
      const files = fs.readdirSync(lateTeamDir);
      filesDeleted += files.length;
      fs.rmSync(lateTeamDir, { recursive: true, force: true });
      console.log(`Deleted late folder for ${username}`);
    }

    res.json({
      success: true,
      message: `All submissions deleted for ${username}`,
      filesDeleted: filesDeleted
    });
  } catch (error) {
    console.error("Error deleting submissions:", error);
    res.status(500).json({
      success: false,
      error: "Failed to delete submissions"
    });
  }
});

app.listen(PORT, () => {
  console.log(`✅ Server running on http://localhost:${PORT}`);
  console.log(` Submission deadline: ${DEADLINE.toLocaleString()}`);
});
```

Frontend


PES
UNIVERSITY
Login
Username
Password
Login


Welcome, Team 075 (Batch 2026)!
20:28:31
Deadline
15/10/2025
16:18:00
1d 19h 49m remaining
Upload Your Files
Deliverable 1:
Choose file No file chosen
Deliverable 2:
Choose file No file chosen
Final Report:
Choose file No file chosen
Submit Files
View All Submissions



Welcome, Team 075 (Batch 2026)!

20:31:07

Deadline

15/10/2025

16:18:00

1d 19h 46m remaining

☑ Submission completed — You can't upload files again

☑ On-Time Submission

Stored in: uploads/ontime/075_2026

Submitted: 13/10/2025, 20:31:00

Files: deliverable1, deliverable2, report

Form disabled - Submission completed

No re-submissions allowed

4.2 Further Exploration Plans and Timelines (optional)

1. Completing the front end for the deliverable tracking.
2. Deploy the ABSA task on more lightweight, resource-efficient models such as BERT or SciBERT from Decoder based LLMs for downstream applications.
3. Link the aggregated score to the paper abstracts and fine-tune BERT or SciBERT to predict the final score directly from the text.

Appendix A: Definitions, Acronyms and Abbreviations

Acronyms and Abbreviations:

XAI: Explainable Artificial Intelligence
CNN: Convolutional Neural Network
FLIR: Forward Looking InfraRed
MATLAB: MATrix LABoratory
PyTorch: Machine learning library for Python
TensorFlow: Open-source machine learning framework
GPU: Graphics Processing Unit
ICMR: Indian Council of Medical Research
LLM: Large Language Model
SIB: Swiss Institute of Business Administration
BERT: Bidirectional Encoder Representations from Transformers
PCC: Pearson Correlation Coefficient
RMSE: Root Mean Square Error
STEM: Science, Technology, Engineering, and Mathematics
IT: Information Technology
ROGUE: Recall-Oriented Understudy for Gisting Evaluation

Key Definitions

1. **Capstone Project**: A comprehensive academic project that demonstrates a student's accumulated knowledge and skills in their field of study, typically completed in the final year of an academic program.
2. **Large Language Model (LLM)**: An advanced AI model trained on vast amounts of text data, capable of understanding and generating human-like text across various domains.
3. **Systematic Review**: A structured method of collecting, analyzing, and synthesizing research findings from multiple sources to provide a comprehensive overview of a specific research topic.
4. **Fine-Tuning**: The process of adapting a pre-trained AI model to perform better on a specific task or domain by further training it on a specialized dataset.
5. **Peer Review**: A critical evaluation process where experts in a field assess the quality, validity, and significance of academic research before publication.
6. **Quality Index**: A standardized metric used to evaluate the overall quality and effectiveness of academic work based on predefined criteria.
7. **Automated Evaluation**: The use of computational techniques, particularly AI and machine learning, to assess and score academic or creative work with minimal human intervention.

Appendix B: References

1. Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-Rubric: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

2. D.M. Anisuzzaman, Jeffrey G. Malins, Paul A. Friedman, Zachi I. Attia,Fine-Tuning Large Language Models for Specialized Use Cases,Mayo Clinic Proceedings: Digital Health,Volume 3, Issue1,2025,100184,ISSN2949-7612,https://doi.org/10.1016/j.mcpdig.2024.11.005.(https://www.sciencedirect.com/science/article/pii/S2949761224001147)

3. https://arxiv.org/abs/2306.00622.Ryan Liu and Nihar Shah,{ryanliu,nihars}@andrew.cmu.edu,

Carnegie Mellon University

4. https://doi.org/10.48550/arXiv.2410.14165,Chihang Wang, Yuxin Dong, Zhenhong Zhang, Ruotong Wang, Shuo Wang, Jiajing Chen

5. https://doi.org/10.48550/arXiv.2010.05137,David Tran, Alex Valtchanov, Keshav Ganapathy, Raymond Feng, Eric Slud, Micah Goldblum, Tom Goldstein.

6. Dmitry Scherbakov, Nina Hubig, Vinita Jansari, Alexander Bakumenko, MSc2, Leslie A.Lenert,MUSC,South Carolina, USA.

7. Milan Kostic1, Hans Friedrich Witschel2, Knut Hinkelmann1, 2, Maja Spahic-Bogdanovic1, 21University of Camerino (UNICAM)

8. Ruiyang Zhou, Lu Chen, and Kai Yu. 2024. Is LLM a Reliable Reviewer? A Comprehensive Evaluation of LLM on Automatic Paper Reviewing Tasks. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LRECCOLING 2024), pages 9340–9351, Torino, Italia. ELRA and ICCL.

9. Ryan Liu and Nihar Shah,{ryanliu, nihars}@andrew.cmu.edu,Carnegie Mellon University .

10. https://doi.org/10.48550/arXiv.2406.16253,Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, Haoran Ranran Zhang, Vipul Gupta, Yinghui Li, Tao Li, Fei Wang, Qin Liu, Tianlin Liu, Pengzhi Gao, Congying Xia, Chen Xing, Jiayang Cheng, Zhaowei Wang, Ying Su, Raj Sanjay Shah, Ruohao Guo, Jing Gu, Haoran Li, Kangda Wei, Zihao Wang, Lu Cheng, Surangika Ranathunga, Meng Fang, Jie Fu, Fei Liu, Ruihong Huang, Eduardo Blanco, Yixin Cao, Rui Zhang, Philip S. Yu, Wenpeng Yin

Appendix C: Record of Change History

[This section describes the details of changes that have resulted in the current Low-Level Design document.]

#	Date	Document Version No.	Change Description	Reason for Change
1.				
2.				
3.				

Appendix D: Traceability Matrix

[Demonstrate the forward and backward traceability of the system to the functional and non-functional requirements documented in the Requirements Document.]

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.	LLD Reference Section No. Name