BAU

Bahçeşehir University

CAPSTONE PROJECT PRESENTATION

BAU-EVAL: LLM-Based Personalized Assignment Evaluator #1011022

Süleyman Kaan Ataç

Beyza Bayrak

Atena Jafari Parsa

Aleyna Kurt

Advisors:

Dr. Binnur Kurt - Department of Artificial Intelligence Engineering

Dr. Fatih Kahraman - Department of Artificial Intelligence Engineering

CONTENTS

1-Introduction
2-Requirements
3-Conceptual Solutions
4-Physical Architecture
5-Flowchart
6-Work Breakdown Structure (WBS)
7-Responsibility Matrix (RM)
8-Project Network (PN)
9-Gantt Chart
10-Data Preparation
11-LLM Subsystem
12-Backend
13-Frontend
14-Integration
15-Verification
16-Reporting
17-Conclusion

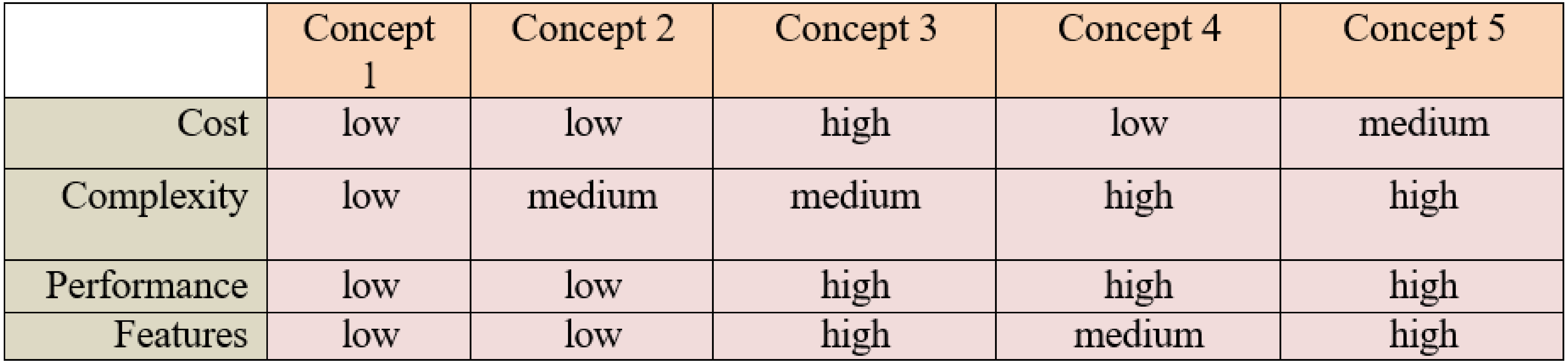
Introduction

This project aims to build an LLM-powered assessment platform for students and instructors in courses like programming, artificial intelligence, and machine learning. The platform generates quizzes and exams, administers time-bound evaluations, and delivers fast, fair feedback to enhance individual learning outcomes.

Conceptual Solutions

- Concept 1: Using ready-to-use LLM
- Concept 2: Open-source LLM
- Concept 3: Closed-source LLM
- Concept 4: Hybrid System with Feedback Mechanism
- Concept 5: Designing Specialized LLM

Comparison of the Conceptual Solutions.



	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
Cost	low	low	high	low	medium
Complexity	low	medium	medium	high	high
Performance	low	low	high	high	high
Features	low	low	high	medium	high

Requirements

Functional Requirements

- Question Generation for Quiz and Exam Generation
- Evaluation and Feedback
- Managing the Evaluations

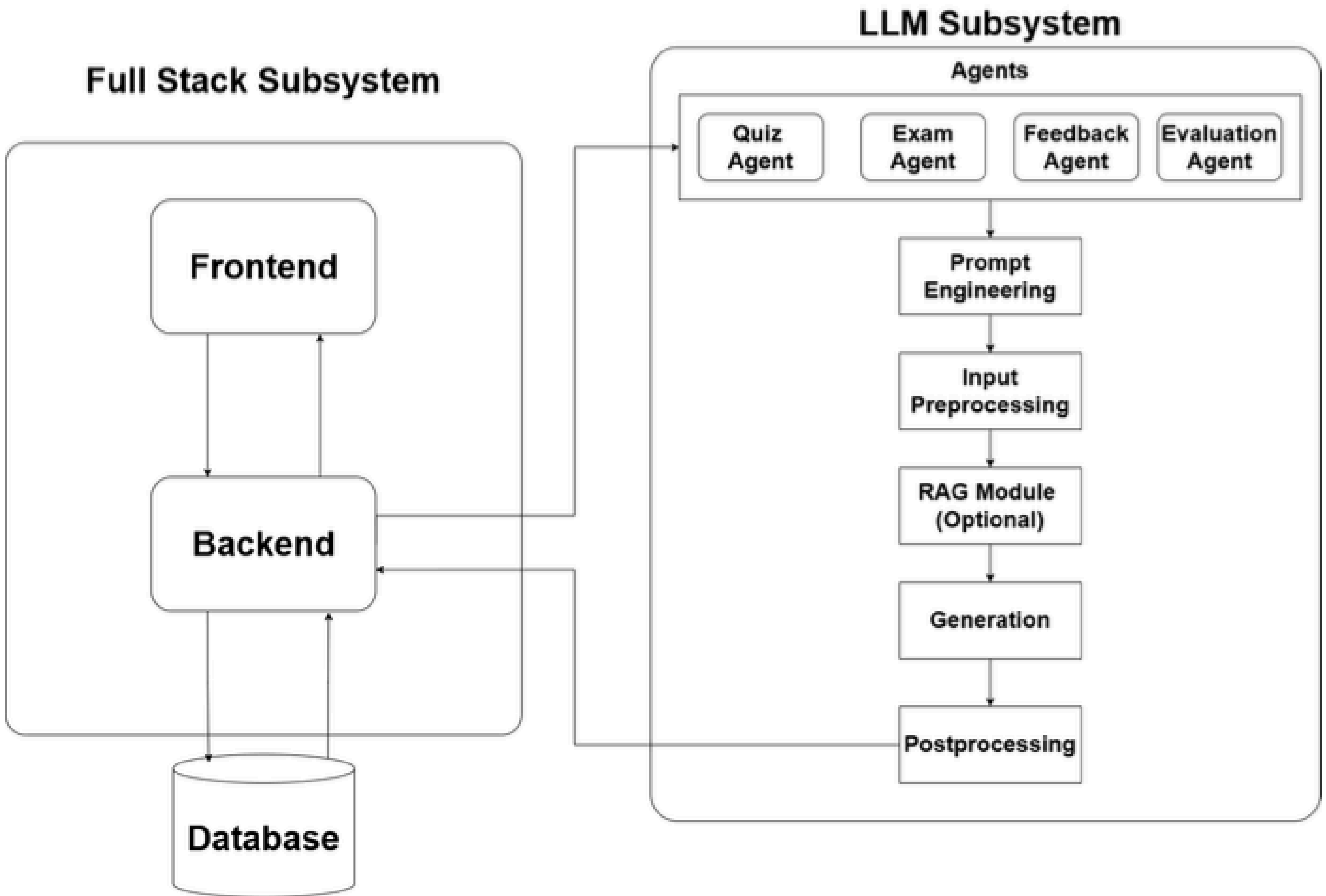
Performance Requirements

- Accuracy
- Time
- Scalability
- Availability

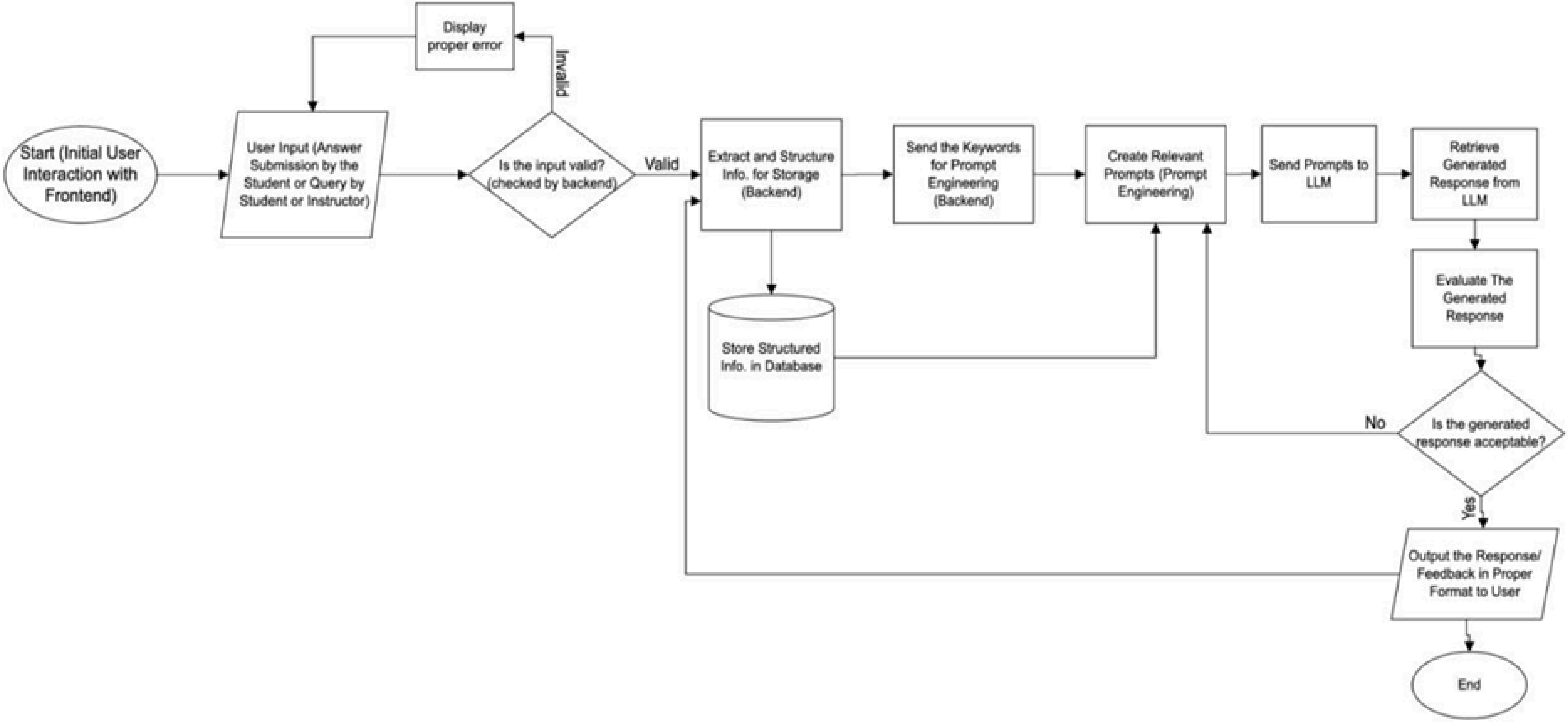
Physical Architecture

The BAU-EVAL architecture is divided into two subsystems:

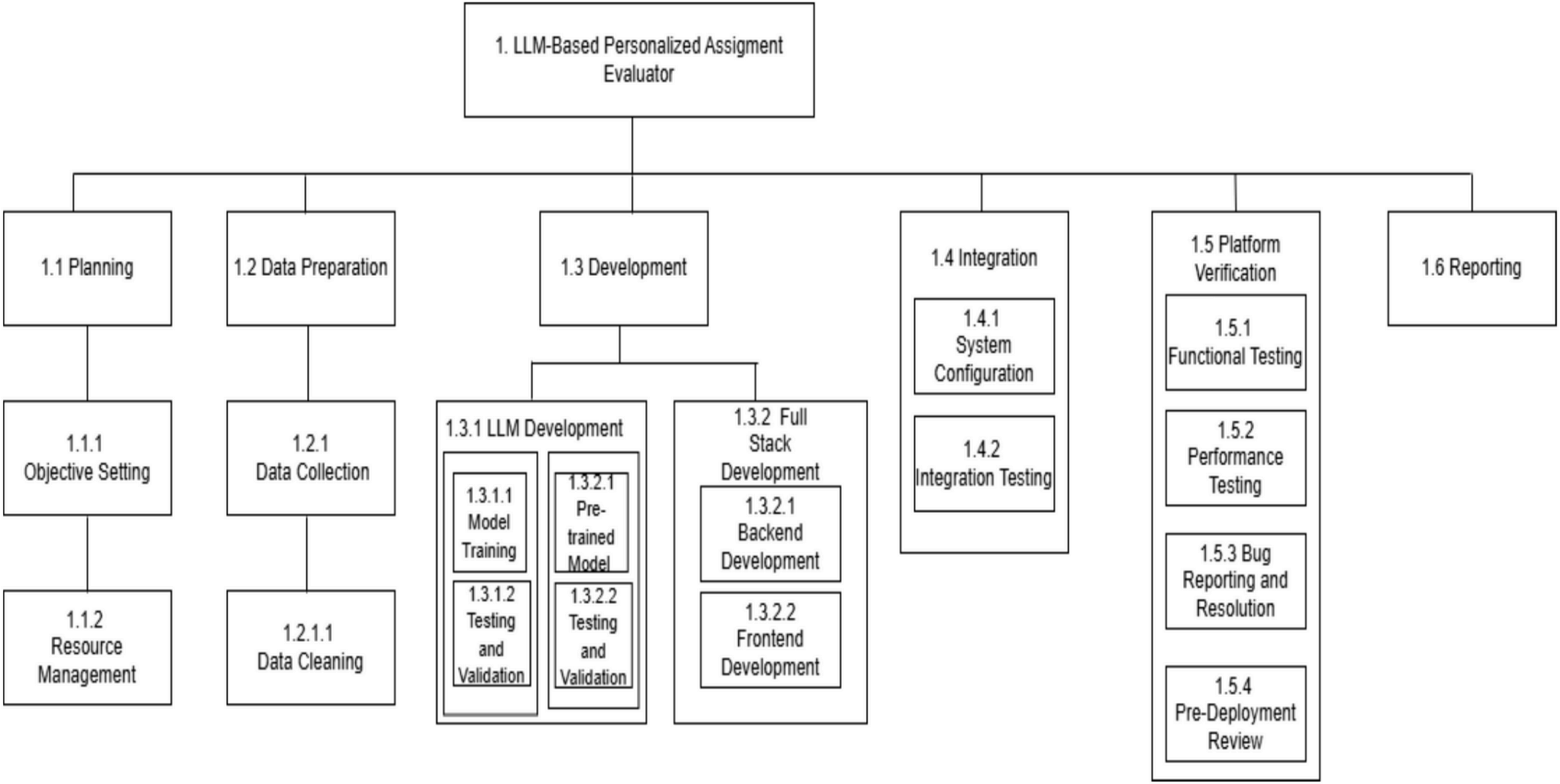
- LLM Subsystem
- Full Stack Subsystem



Flowchart


Start (Initial User Interaction with Frontend)
User Input (Answer Submission by the Student or Query by Student or Instructor)
Is the input valid? (checked by backend)
Invalid
Display proper error
Valid
Extract and Structure Info. for Storage (Backend)
Store Structured Info. in Database
Send the Keywords for Prompt Engineering (Backend)
Create Relevant Prompts (Prompt Engineering)
Send Prompts to LLM
Retrieve Generated Response from LLM
Evaluate The Generated Response
Is the generated response acceptable?
No
Yes
Output the Response/ Feedback in Proper Format to User
End

Work Breakdown Structure (WBS)



1. LLM-Based Personalized Assigment Evaluator

1.1 Planning

1.1.1 Objective Setting

1.1.2 Resource Management

1.2 Data Preparation

1.2.1 Data Collection

1.2.1.1 Data Cleaning

1.3 Development

1.3.1 LLM Development

1.3.1.1 Model Training

1.3.1.2 Testing and Validation

1.3.2.1 Pre-trained Model

1.3.2.2 Testing and Validation

1.3.2 Full Stack Development

1.3.2.1 Backend Development

1.3.2.2 Frontend Development

1.4 Integration

1.4.1 System Configuration

1.4.2 Integration Testing

1.5 Platform Verification

1.5.1 Functional Testing

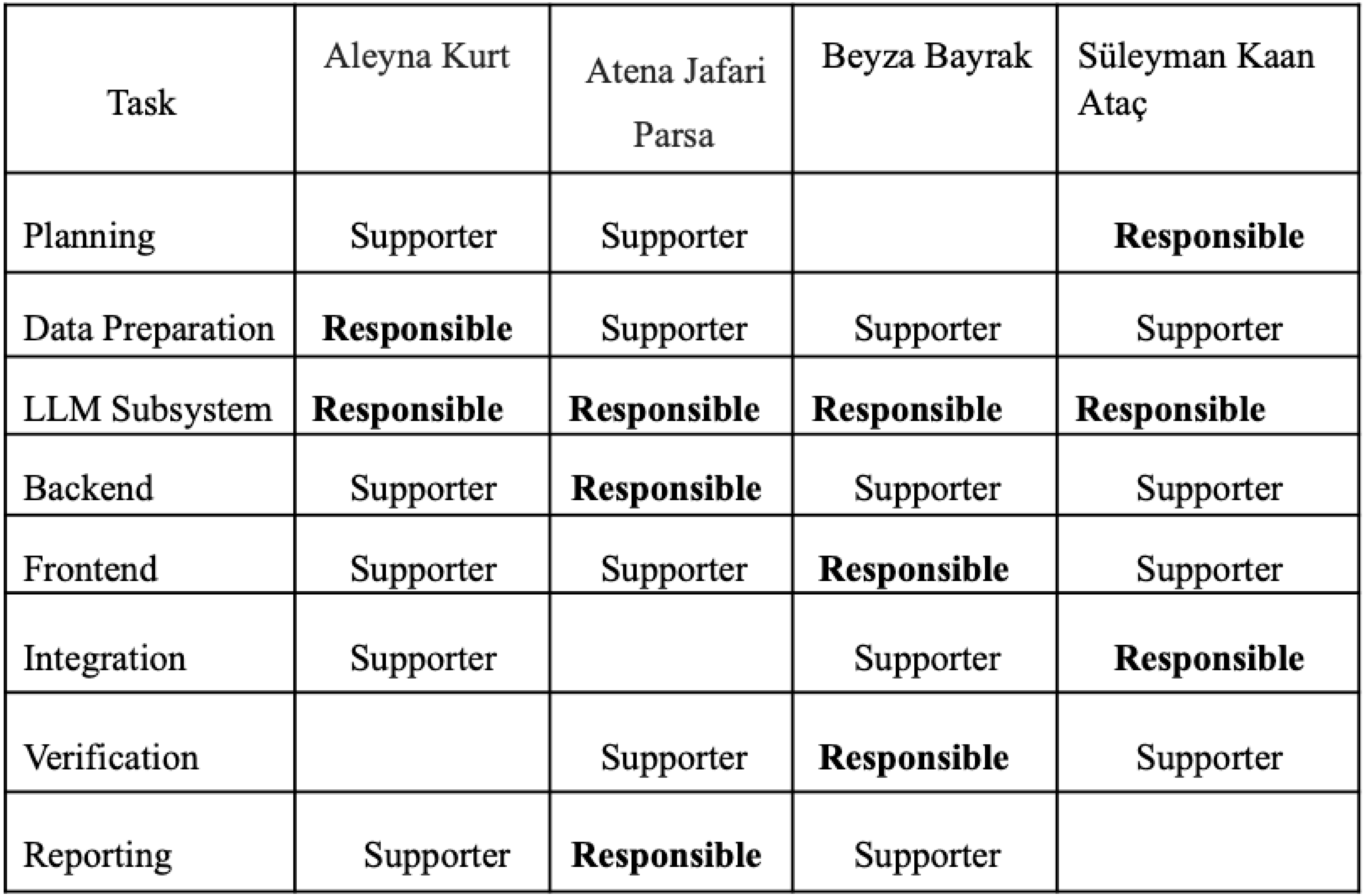
1.5.2 Performance Testing

1.5.3 Bug Reporting and Resolution

1.5.4 Pre-Deployment Review

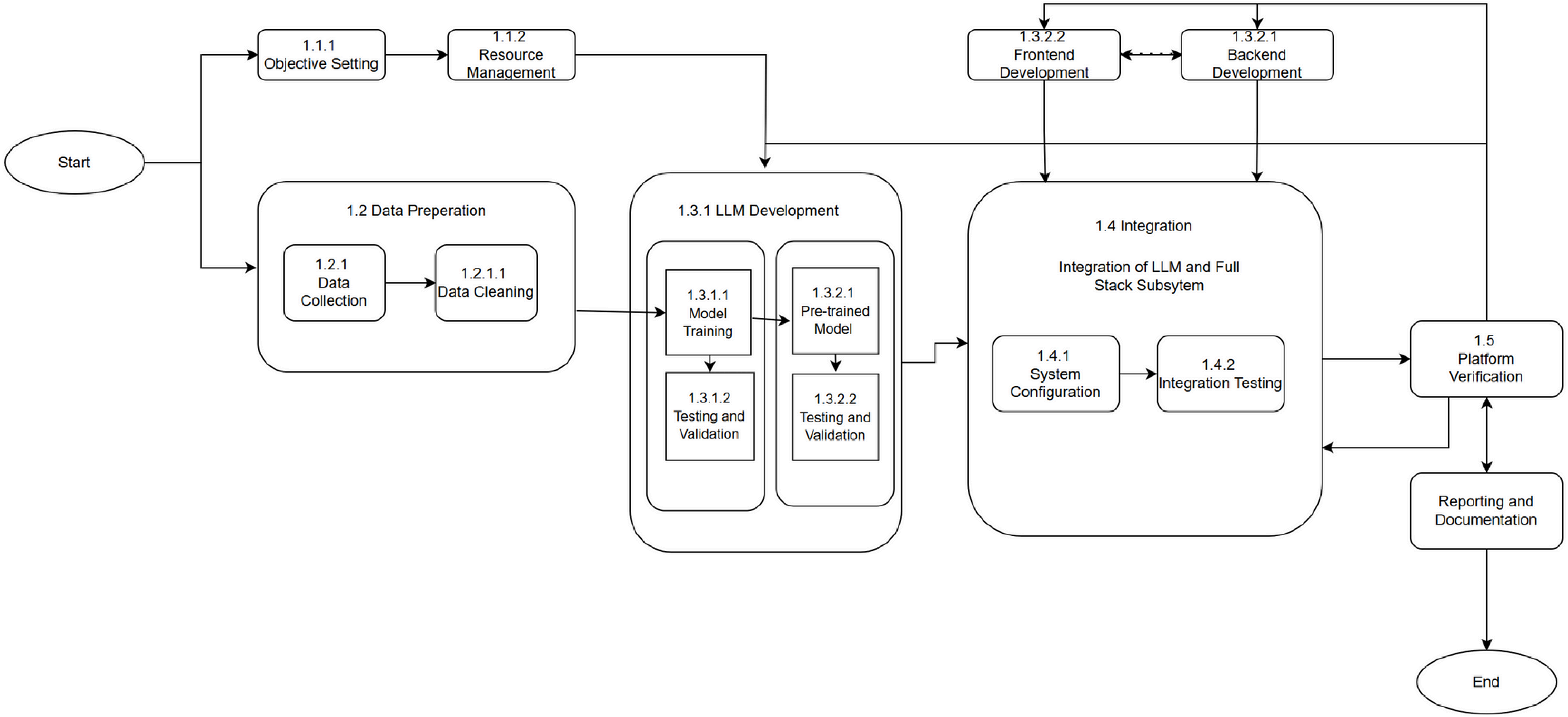
1.6 Reporting

Responsibility Matrix (RM)

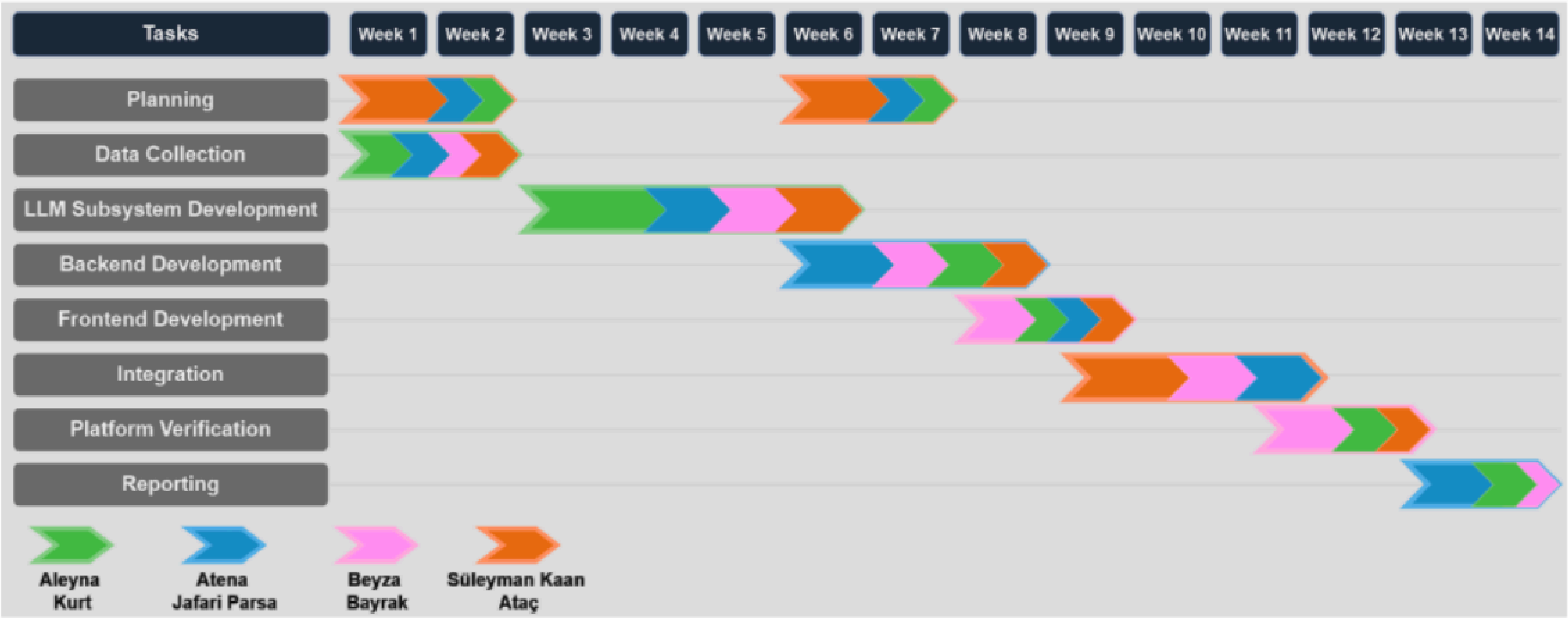


Task	Aleyna Kurt	Atena Jafari Parsa	Beyza Bayrak	Süleyman Kaan Ataç
Planning	Supporter	Supporter		**Responsible**
Data Preparation	**Responsible**	Supporter	Supporter	Supporter
LLM Subsystem	**Responsible**	**Responsible**	**Responsible**	**Responsible**
Backend	Supporter	**Responsible**	Supporter	Supporter
Frontend	Supporter	Supporter	**Responsible**	Supporter
Integration	Supporter		Supporter	**Responsible**
Verification		Supporter	**Responsible**	Supporter
Reporting	Supporter	**Responsible**	Supporter	

Project Network (PN)



Gantt Chart


Tasks
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Planning
Data Collection
LLM Subsystem Development
Backend Development
Frontend Development
Integration
Platform Verification
Reporting
Aleyna Kurt
Atena Jafari Parsa
Beyza Bayrak
Süleyman Kaan Ataç

Data Preparation and Fine-Tuning

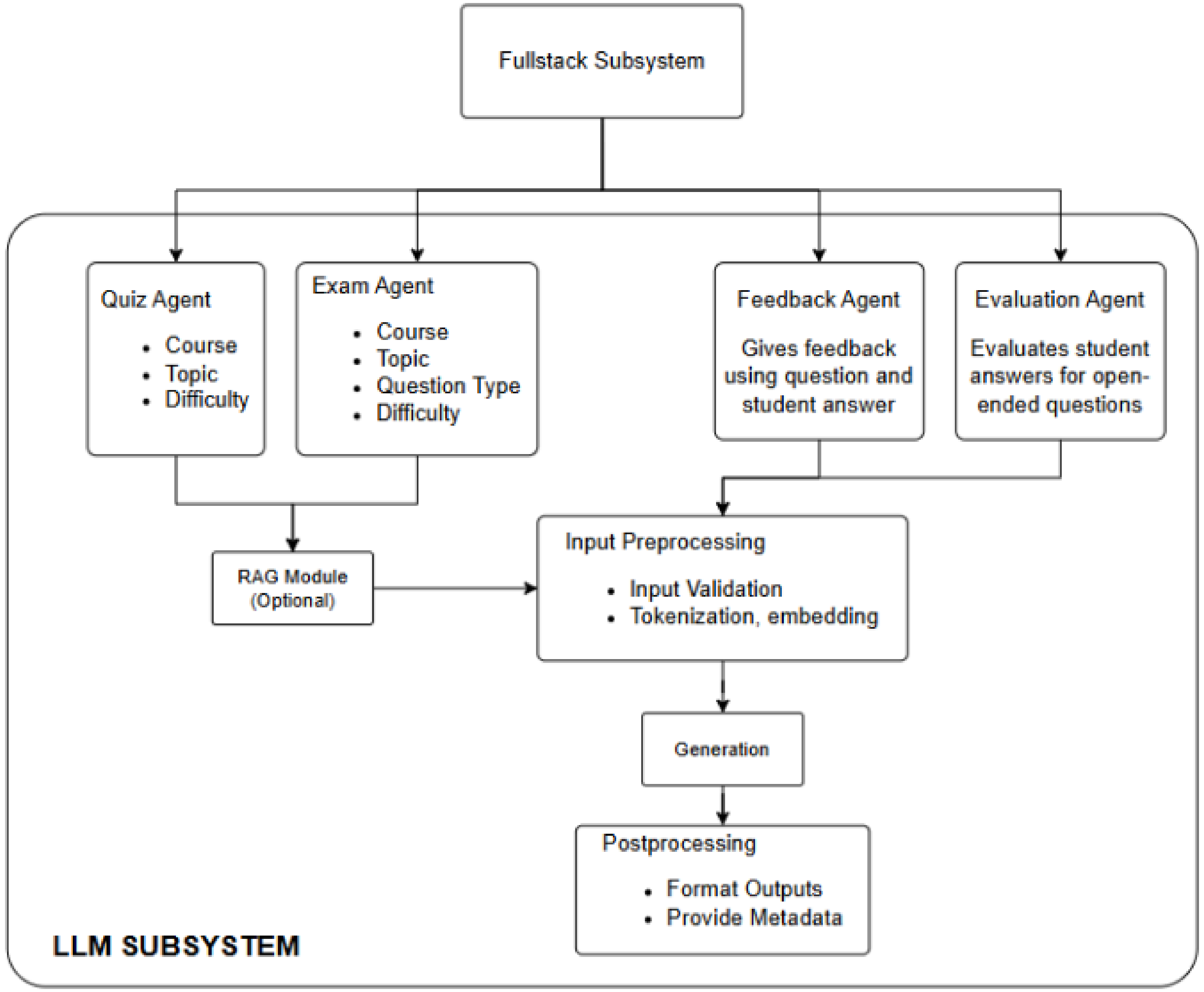
Data Preparation

❌

Fine-Tuning

❌

LLM Subsystem



LLM subsystem focused on question generation and evaluation using Hybrid LLM system with Feedback Mechanism and Open Source LLM.

LLM Subsystem

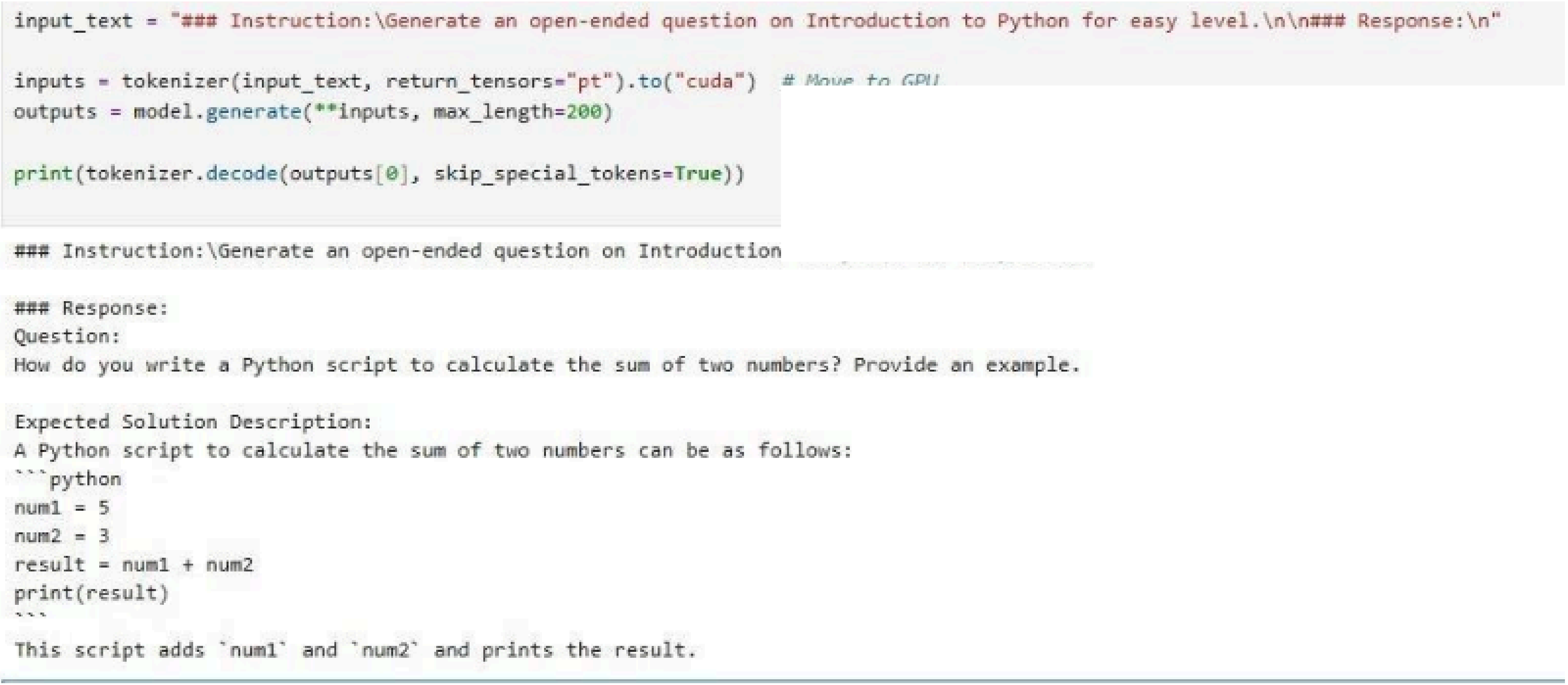
In this module, a LLM system is developed. In accordance with the course material on programming and artificial intelligence, the LLM has been used to create quizzes and exams and assess student answers. It is able to evaluate open-ended submissions as well as open-ended responses and also provides feedback to submitted quizzes and exams for students.

LLM Subsystem

Deployment of a LLM subsystem:
Extensions used:

- **LLM:** Ollama, Llama 3, Mistral, Gemma, Phi3, LangChain (or similar LLM frameworks)

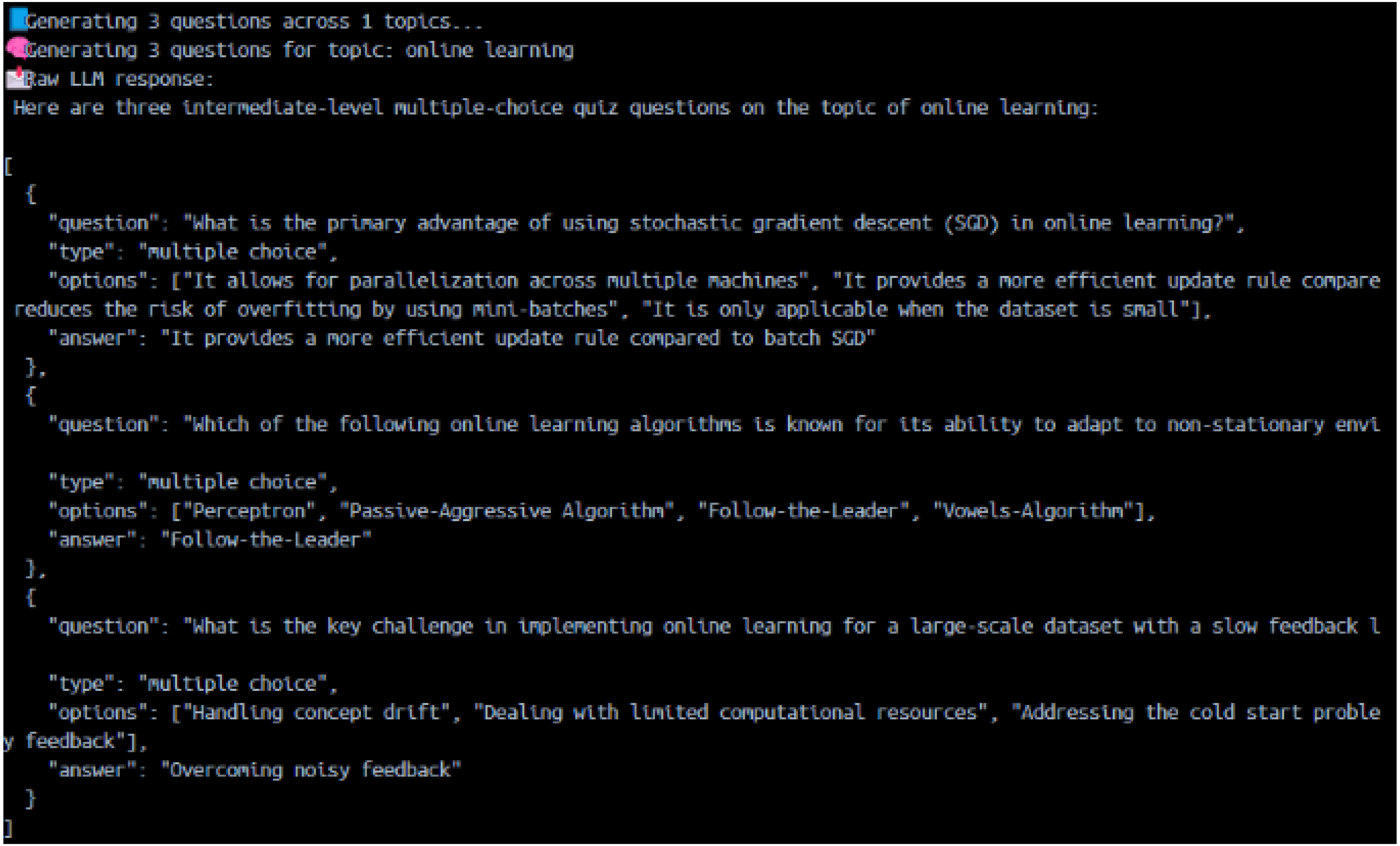
- **Conversational AI GUIs AI Integration:** Hugging Face API, TensorFlow/PyTorch



```
input_text = "### Instruction:\Generate an open-ended question on Introduction to Python for easy level.\n\n### Response:\n"

inputs = tokenizer(input_text, return_tensors="pt").to("cuda")  # Move to GPU
outputs = model.generate(**inputs, max_length=200)

print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

~~~
### Instruction:\Generate an open-ended question on Introduction

### Response:
Question:
How do you write a Python script to calculate the sum of two numbers? Provide an example.

Expected Solution Description:
A Python script to calculate the sum of two numbers can be as follows:
```python
num1 = 5
num2 = 3
result = num1 + num2
print(result)
```
This script adds `num1` and `num2` and prints the result.
~~~

Example of Fine-tuning Generation



```
🟦Generating 3 questions across 1 topics...
💭Generating 3 questions for topic: online learning
📩Raw LLM response:
 Here are three intermediate-level multiple-choice quiz questions on the topic of online learning:


[
  {
    "question": "What is the primary advantage of using stochastic gradient descent (SGD) in online learning?",
    "type": "multiple choice",
    "options": ["It allows for parallelization across multiple machines", "It provides a more efficient update rule compare
 reduces the risk of overfitting by using mini-batches", "It is only applicable when the dataset is small"],
    "answer": "It provides a more efficient update rule compared to batch SGD"
  },
  {
    "question": "Which of the following online learning algorithms is known for its ability to adapt to non-stationary envi

    "type": "multiple choice",
    "options": ["Perceptron", "Passive-Aggressive Algorithm", "Follow-the-Leader", "Vowels-Algorithm"],
    "answer": "Follow-the-Leader"
  },
  {
    "question": "What is the key challenge in implementing online learning for a large-scale dataset with a slow feedback l

    "type": "multiple choice",
    "options": ["Handling concept drift", "Dealing with limited computational resources", "Addressing the cold start proble
y feedback"],
    "answer": "Overcoming noisy feedback"
  }
]
```

Pre-trained Model Responses

Backend Architecture

- The backend serves as the central processing unit for LLM-Based Assignment Evaluator.
- It securely manages user accounts, quiz and exam submissions.
- It interacts with the LLM to generate questions and provide evaluations
- It provides real-time feedback to students.

Core Backend Modules

- API: Provides a communication interface between the frontend and backend.
- Database: Securely stores user data, quizzes, exams, and evaluation results.
- LLM Integration Module: Manages communication and data exchange with the Large Language Model (LLM).
- Authentication/Authorization Module: Secures user access and permissions.
- Verification Module (Instructor Validation): Instructors validate LLM-generated assessments ensuring fairness and accuracy.

Technology Platforms

- Backend: FastAPI
- Frontend: HTML, CSS for Style, Javascript
- Database management: MongoDB, Docker
- Containerization: Docker
- Cloud: Azure

API Design

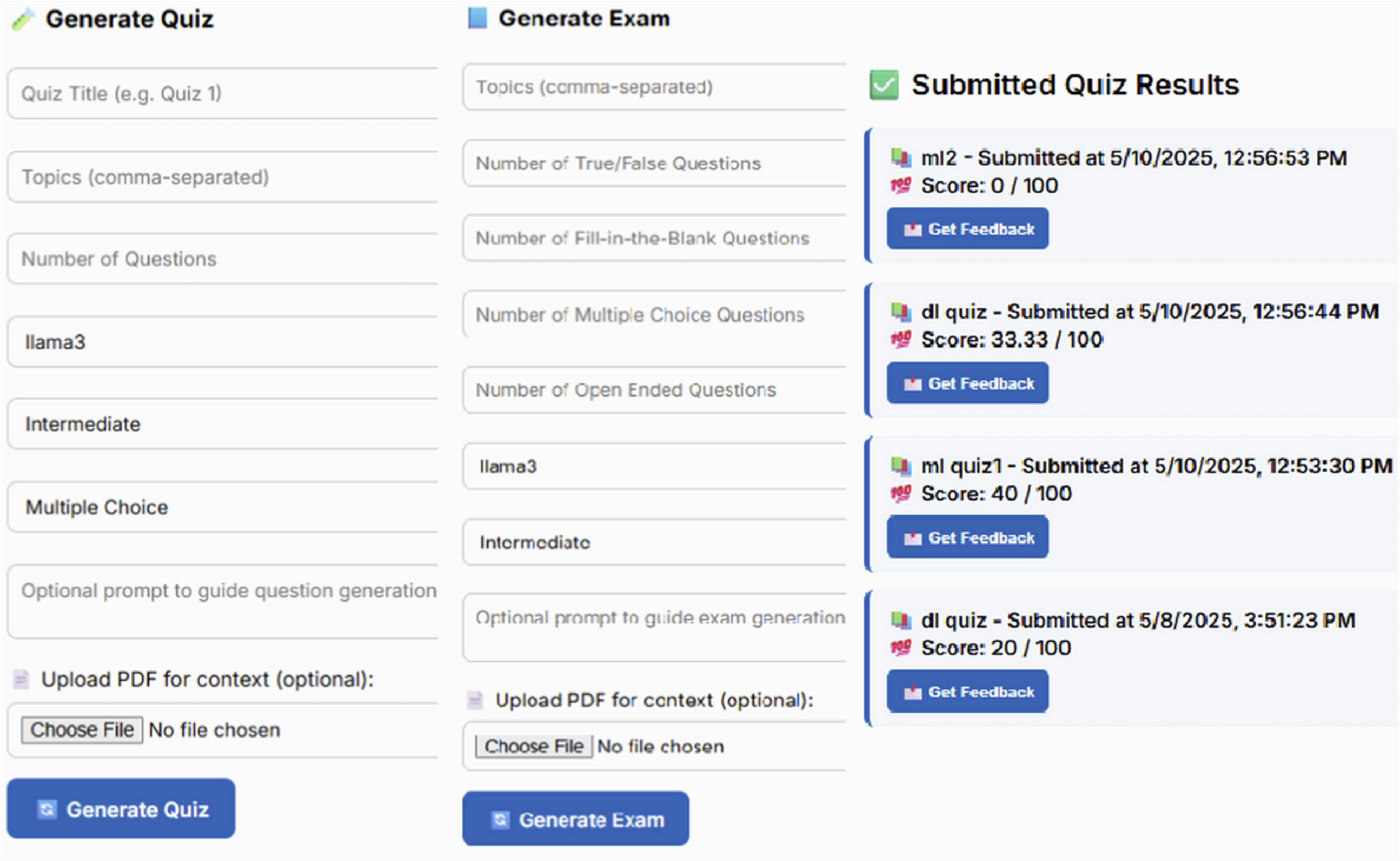
Communicating with the Backend

- RESTful API was used for efficient communication using FastAPI.
- Endpoints were designed for user authentication, exam and quiz submission, evaluation requests, and feedback retrieval.

Frontend

Modern User Interface

- Dynamic interface; effortlessly interaction for users
- Real-time response and feedback



Generate Quiz

Quiz Title (e.g. Quiz 1)

Topics (comma-separated)

Number of Questions

llama3

Intermediate

Multiple Choice

Optional prompt to guide question generation

Upload PDF for context (optional):

Choose File No file chosen

Generate Exam

Topics (comma-separated)

Number of True/False Questions

Number of Fill-in-the-Blank Questions

Number of Multiple Choice Questions

Number of Open Ended Questions

llama3

Intermediate

Optional prompt to guide exam generation

Upload PDF for context (optional):

Choose File No file chosen

✅ Submitted Quiz Results

ml2 - Submitted at 5/10/2025, 12:56:53 PM
Score: 0 / 100

dl quiz - Submitted at 5/10/2025, 12:56:44 PM
Score: 33.33 / 100

ml quiz1 - Submitted at 5/10/2025, 12:53:30 PM
Score: 40 / 100

dl quiz - Submitted at 5/8/2025, 3:51:23 PM
Score: 20 / 100

Scability and Speed

- Fast question/answer generation and instant feedback

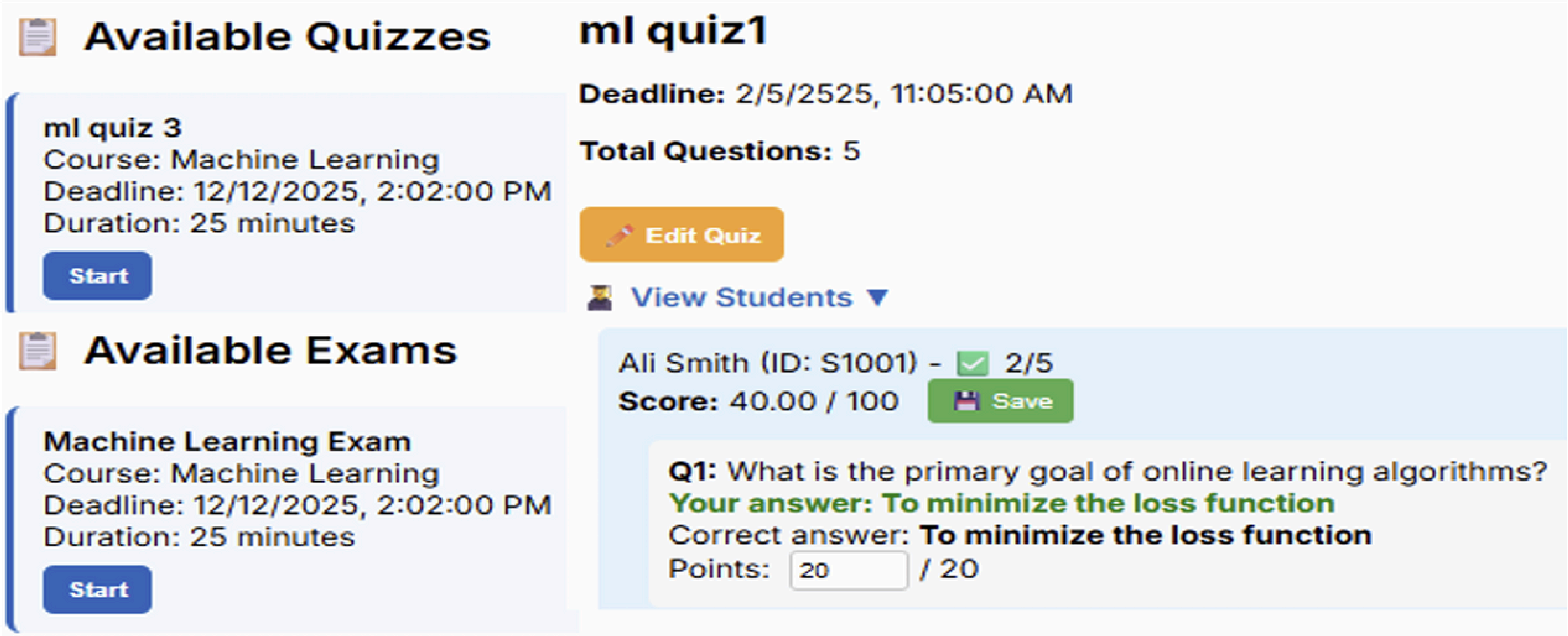
Tests showed that generating a question takes about 2.1 seconds; grading takes 1 second; feedback takes approximately 2.5 seconds per question.

- Managing high number of users simultaneously

Tests showed that the system can handle 1000 users at once successfully.

Enhanced User Experience

- Clear dashbord for instructions to monitor
- Easy to navigate for students



Available Quizzes

ml quiz 3
Course: Machine Learning
Deadline: 12/12/2025, 2:02:00 PM
Duration: 25 minutes

Available Exams

Machine Learning Exam
Course: Machine Learning
Deadline: 12/12/2025, 2:02:00 PM
Duration: 25 minutes

ml quiz1

Deadline: 2/5/2525, 11:05:00 AM

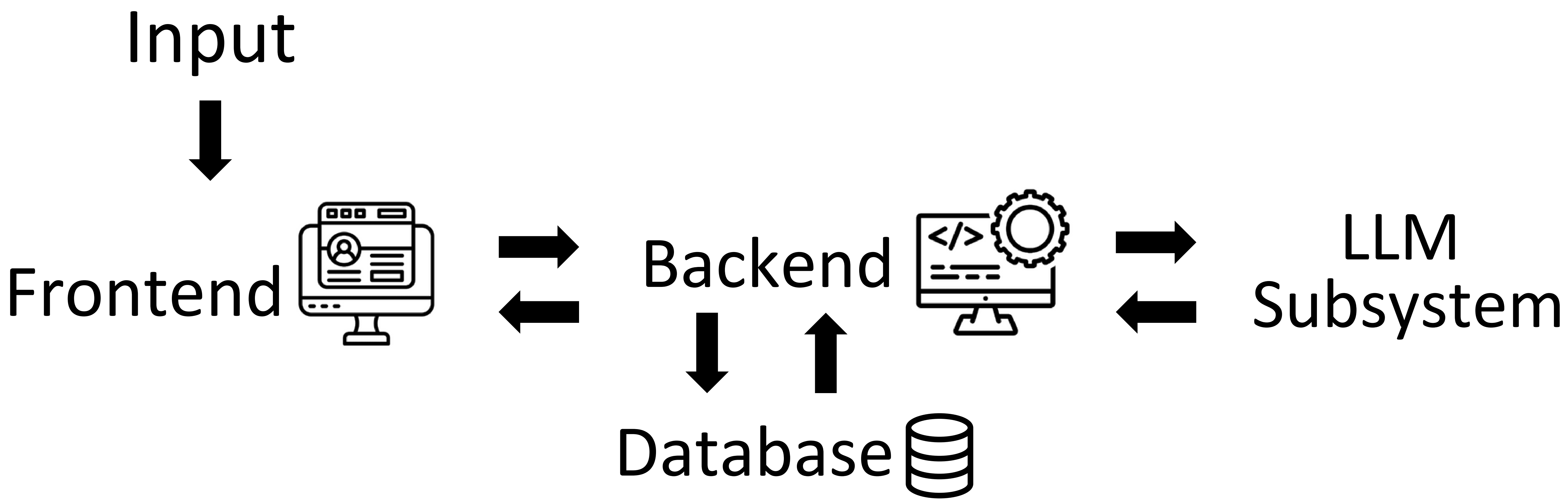
Total Questions: 5

Ali Smith (ID: S1001) - ✓ 2/5
Score: 40.00 / 100

Q1: What is the primary goal of online learning algorithms?
Your answer: To minimize the loss function
Correct answer: **To minimize the loss function**
Points: 20 / 20

Integration

Data Flow



Input

Frontend

Backend

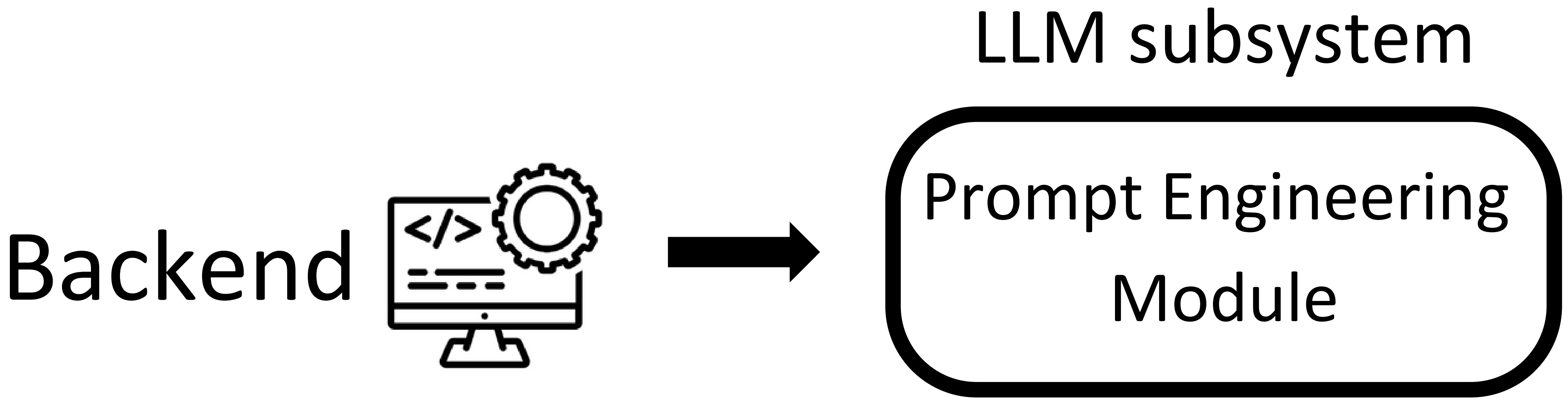
LLM Subsystem

Database

Integration

Prompt Processing

The Backend sends the processed data to the Prompt Engineering module in the LLM subsystem, where prompts are created and preprocessed for the LLM.



Integration

APIs and Endpoints

Establishing robust APIs between the Backend and the LLM subsystem to ensure efficient and error-free data transfer.

Data Consistency and Integrity

Ensuring that the data remains consistent and intact throughout the process, from initial input through to final output.

Verification

Accuracy and Efficieny

- Various tests were conducted to verify the LLM model’s accuracy in generating and scoring questions.
- RAG allowed the system to generate high-quality, context-aware questions directly from instructional materials.

Scalability and Availability

- Tested for 1000+ users simultaneously
- Consistent accessibility with 99.9% uptime guarentees

Reporting

The system comprises of three core components:

- An LLM model API with the option to upload relevant course materials to ensure accurate and unbiased evaluations.
- A web-based personalized examination panel providing a seamless and user-friendly interface for students to take quizzes and exams with provided feedbacks and auto-grading, as well as a panel for instructors for course setup, exam/quiz generation and preview and manual grade updates.
- A cloud-native backend providing scalable and secure infrastructure to manage data, handle LLM interactions, and deliver results.

The intended users of this platform are students and instructors in various courses at the university level.

Challenges, Limitations and Setbacks

1. LLM-Related Challenges

- LLM Performance Limitations
- LLM Cost and Resource Constraints
- LLM API Limitations
- Maintaining LLM Up-to-Date (for future)

2. Backend Development Challenges

- Database Scalability Limitations
- Testing Challenges
- Integration Complexity
- Security Vulnerabilities

3. Frontend Development Challenges

- User Interface Design Limitations
- Frontend Responsiveness and Performance

4. Overall Project Management Challenges

- Time Constraints
- Resource Constraints
- Teamwork and Communication Challenges

5. Data Challenges

- Data Acquisition and Preparation Challenges
- Data Privacy and Security Concerns

Conclusion

Objective

Develop a quiz, exam generation and assessment system with feedbacks using LLM technology for Bahçeşehir University

Achievements

- Automated the generation and evaluation of exams and quizzes.
- Integrated cutting-edge LLM with full-stack development
- Implemented a LLM system with optional RAG implementation to get content related results.

Conclusion

Future Directions

- Further enhancements based on user feedback and technological advancements.
- Expand system capabilities to enhance educational outcomes and learning experiences.

Impact: Streamlined assessment processes and improved educational outcomes through AI-driven personalization and efficiency to set a new standard for educational excellence

REFERENCES

C.-H. Chiang, W.-C. Chen, C.-Y. Kuan, C. Yang, and H.-Y. Lee, “Large Language Model as an Assignment Evaluator: Insights, Feedback, and Challenges in a 1000+ Student Course”, [Online]. Available: https://aclanthology.org/2024.emnlp-main.146.pdf [Accessed: Dec. 21, 2024].

Y.-T. Lin and Y.-N. Chen, “LLM-EVAL: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models”, [Online]. Available: https://arxiv.org/pdf/2305.13711 [Accessed: May 23, 2023]

Y. Liu et al.,” Aligning with Human Judgement: The Role of Pairwise Preference in Large Language Model Evaluators”, [Online]. Available: http://arxiv.org/pdf/2403.16950 [Accessed: Aug. 10, 2024].

D. Hirunyasiri, D. R. Thomas, J. Lin, K. R. Koedinger, and V. Aleven, “Comparative Analysis of GPT-4 and Human Graders in Evaluating Praise Given to Students in Synthetic Dialogues”, [Online]. Available: https://arxiv.org/pdf/2307.02018 [Accessed: Jul. 05, 2023].

Z. Chu, Q. Ai, Y. Tu, H. Li, and Y. Liu, “PRE: A Peer Review Based Large Language Model Evaluator”, [Online]. Available: https://arxiv.org/pdf/2401.15641 [Accessed: Jun. 03, 2024].

K. D. Dunnell, T. Painter, A. Stoddard, and A. Lippman, “Latent Lab: Large Language Models for Knowledge Exploration”, [Online]. Available: https://arxiv.org/pdf/2311.13051 [Accessed: Nov. 21, 2023].

X. Yue et al., “A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI” , [Online]. Available: https://mmmu-benchmark.github.io [Accessed: Sep. 05, 2024].

M. Fariz, S. Lazuardy, and D. Anggraini, “Modern Front End Web Architectures with React.Js and Next.Js,” International Research Journal of Advanced Engineering and Science, [Online]. Available: https://irjaes.com/wp-content/uploads/2022/02/IRJAES-V7N1P162Y22.pdf [Accessed: 2022].
R. Sawhney, Beginning Azure Functions: Building Scalable and Serverless Apps. 2019.

A. Luca, “POLITECNICO DI TORINO User Interface Development of a Modern Web Application Candidate Marzieh SOMI,” Available: https://webthesis.biblio.polito.it/secure/30076/1/tesi.pdf [Accessed: 2021].

Thank you