# ****Requirement Analysis****

## ****1. Introduction****

Requirement analysis is a critical phase of software development, as it defines the foundation upon which the entire project is built. It ensures that the expectations of users, developers, and stakeholders are clearly understood and documented. For the **Educational AI Assistant**, requirement analysis was performed to identify the features, technical needs, and constraints necessary to achieve its objectives. The goal was to create a reliable, efficient, and user-friendly application that integrates natural language processing with a simple interface for educational purposes.

## ****2. Objectives of Requirement Analysis****

* The objectives of performing requirement analysis for the project were:
* To identify the **functional requirements** that describe the features of the system.
* To determine the **non-functional requirements** such as performance, usability, and security.
* To define the **hardware and software environment** needed for smooth operation.

To document requirements in a structured way, reducing ambiguity and misunderstandings during development.

## ****3. Functional Requirements****

Functional requirements define what the system should do. For the Educational AI Assistant, the key functional requirements are:

1. **Concept Explanation Module**

* The system should allow users to input a concept or topic.
* It should generate a detailed explanation with examples in natural language.

1. **Quiz Generation Module**

* The system should generate at least five quiz questions based on the given topic.
* Questions should include a variety of formats (multiple choice, true/false, short answer).
* The system should provide an “ANSWERS” section for users to check their performance.

1. **User Interface**

* The interface should provide two separate tabs: one for explanations and one for quizzes.
* Input and output should be displayed in textboxes with clear formatting.
* Buttons should trigger the respective functions easily.

1. **Processing and Output**

* The assistant should process input using the Hugging Face tokenizer and IBM Granite model.
* It should generate outputs that are coherent, relevant, and properly formatted.

1. **Error Handling**

* The system should gracefully handle invalid inputs, such as empty prompts.
* It should ensure fallback functionality (e.g., CPU execution when GPU is unavailable).

## ****4. Non-Functional Requirements****

Non-functional requirements specify how the system should perform rather than what it should do. For this project, they include:

1. **Performance**

* The system should generate short explanations within 3–5 seconds on GPU hardware.
* On CPU systems, responses should be delivered within 15–30 seconds.

1. **Usability**

* The interface should be simple and intuitive, requiring no prior technical knowledge.
* Users should be able to access the application via a web browser without installing additional software.

1. **Scalability**

* The system should handle multiple users via shared links without overlapping sessions.
* It should support various topics across domains like science, history, mathematics, etc.

1. **Reliability**

* The assistant should consistently generate meaningful and accurate outputs.
* It should handle long prompts without crashing, even if truncation occurs

1. **Security**

* User inputs should not be stored or shared externally.
* The system should operate within the secure environment of Gradio and Hugging Face.

## ****5. Hardware Requirements****

### ****Minimum Requirements (CPU Mode):****

* Processor: Dual-core CPU (2.0 GHz or higher)
* RAM: 8 GB
* Storage: 5 GB free space
* Internet: Required for initial model download

**Recommended Requirements (GPU Mode):**

* Processor: Quad-core CPU (3.0 GHz or higher)
* RAM: 16 GB or more
* GPU: NVIDIA CUDA-enabled GPU (6 GB VRAM or higher)
* Storage: 10 GB free space
* Internet: Stable broadband for setup and cloud usage

## ****6. Software Requirements****

**1. Operating System**: Windows 10/11, Linux (Ubuntu), or macOS.

**2**. **Programming Language**: Python 3.8 or higher.

**3. Libraries/Frameworks**:

* PyTorch
* Hugging Face Transformers
* Gradio
* Torchvision (for GPU compatibility)

1. **Development Tools**:

* Jupyter Notebook / VS Code
* Git for version control
* Google Colab or Kaggle (optional cloud execution)

## ****7. Conclusion****

The requirement analysis clearly defines the technical and functional expectations of the Educational AI Assistant. By outlining both functional and non-functional requirements along with hardware and software needs, this phase ensures that the project is built on a strong foundation. Meeting these requirements will guarantee that the assistant remains reliable, scalable, and user-friendly, thereby achieving its goal of improving the teaching and learning experience.