

Intelligent Tutoring System (ITS)

Objective:

The purpose of this assignment is to **design a software architecture** and **implement** the **Intelligent Tutoring System (ITS)** using **SOLID** principles. This system provides personalized, adaptive learning experiences for students. The ITS should be capable of assessing students' knowledge, providing feedback, and recommending customized learning paths. The goal is to create a flexible and scalable architecture that can handle a variety of learning contexts.

Context:

An Intelligent Tutoring System (ITS) is a type of software application that uses artificial intelligence techniques to provide individualized instruction to learners. These systems are capable of assessing student performance, providing hints, and offering feedback based on student behavior and progress. An ITS aims to replicate some aspects of one-on-one human tutoring, where the system adjusts dynamically to the learner's pace, abilities, and preferences.

The system should be able to handle multiple subjects, topics, and difficulty levels, and must also offer administrative features for instructors, such as tracking student progress and generating reports.

Scope of the System:

- **Personalized Learning:** The system should assess individual students' strengths, weaknesses, and learning styles, and provide customized instructional content.
- **Feedback:** Students should receive feedback on their performance, including guidance and hints when necessary.
- **Assessment and Evaluation:** The ITS should include mechanisms for evaluating student progress through quizzes, exercises, or projects.
- **Instructor Dashboard:** Teachers can monitor student progress, manage content, and generate reports.
- **Learning Content Management:** Ability to manage, update, and curate learning materials across different topics, formats (e.g., text, video, interactive exercises).

Assignment Tasks:

Task 1: Software Architecture Design

1. **Thoroughly describe the context** of the **Intelligent Tutoring System (ITS)** based on the basic information provided. This includes clearly identifying functional and non-functional requirements, outlining the objectives and scope of the project, etc.
2. **Create the software architecture** for the **ITS**: The software architecture should include:

- **Architecture Characteristics** define the success criteria of the ITS.
 - **Structure:**
 - Compare and choose suitable architecture styles to apply to the ITS.
 - Present the software architecture in different views including module views, component-and-connector views, and allocation views.
 - **Architecture Decisions** define the rules for how the ITS should be constructed.
 - **Design Principles:** Guidelines for constructing the ITS.
3. **Apply SOLID Principles:** explain how the **SOLID** principles have been applied in your design.
 4. **Reflection Report:** Write a brief reflection on how applying SOLID principles helped you improve the design of the **ITS**. Discuss challenges you faced and how adhering to these principles made your system more modular, maintainable, and extensible.

Task 2: Code Implementation (choose **at least one main module** to implement)

1. **Implement Core Functionalities:** Implement key system features based on your design, ensuring the code adheres to the SOLID principles.

Submission Instructions:

- Submit a detailed report containing a GitHub URL (source code) on the LMS website.