PROJECT TITLE : COACHING MANAGEMENT SYSTEM

**NAME:**  
SANJANA SALUKHE(23011031)  
SHRUSHTI PANCHDHANE (23011035)

# AIM :

In this experiment, we will learn how to identify functional and non-functional requirements from a given problem statement. These are the primary components of a Software Requirements Specification (SRS) for the Coaching Management System.

# INTRODUCTION :

Requirement identification is the first step in any software project. For the Coaching Management System, it is essential to define what features the system should provide—such as student registration, batch management, attendance tracking, and performance analysis. Until these requirements are clearly understood and verified, development cannot proceed. Analysts usually work with institute administrators, staff, and students to gather accurate and complete requirements.

# THEORY :

## Objectives :

- Identify ambiguities, inconsistencies, and incompleteness from a requirements specification for a Coaching Management System  
- Identify and state functional requirements specific to managing coaching classes and student activities  
- Identify and state non-functional requirements such as system reliability, usability, and performance

# REQUIREMENTS :

A requirement specifies what the system should do without explaining how it will be done. For a Coaching Management System, requirements define expected behaviors such as student enrollment, course assignment, scheduling, and result reporting.  
  
Requirements engineering involves understanding stakeholder needs and documenting them clearly. This becomes the foundation for system design, development, and testing.

# CHARACTERISTICS OF REQUIREMENTS :

1. Unambiguity  
There should be no confusion about what the system should do.  
Example: "The system should allow a student to enroll in a batch and receive confirmation."  
  
2. Consistency  
Requirements should not contradict each other.  
Example: Admin should not be allowed to delete an active batch while also being asked to edit it.  
  
3. Completeness  
Requirements should specify what the system must do and must not do.  
Example: Only the admin can assign a teacher to a course, and students cannot do this.

# CATEGORIZATION OF REQUIREMENTS :

**User Requirements:**  
Written in natural language for teachers, admin staff, and students to understand.  
Example: "Students should be able to view their timetable and attendance status."  
 **System Requirements:**  
Technical specifications for developers and testers.  
Example: "The system shall allow only admin users to create or delete a batch."

# BASED ON DESCRIPTION :

**Functional Requirements (FRs):**  
These describe what the system should do.  
Examples:  
- Register students and assign them to courses or batches.  
- Add and manage teaching staff.  
- Track attendance and test scores.  
- Schedule lectures and notify students.  
- Generate progress and fee reports.  
 **Non-Functional Requirements (NFRs):**  
These describe how the system should perform.  
Examples:  
- The system should be accessible via both web and mobile platforms.  
- Data backup should be performed daily.  
- System should respond within 2 seconds for any user query.  
- The system should support 500 simultaneous users.

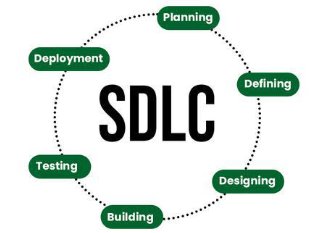
# SIMULATION :

Functional Requirements for Coaching Management System:  
  
- The system should allow users to register and log in based on roles: admin, teacher, student.  
- Admin should be able to add/edit/delete student profiles, assign batches, and schedule classes.  
- Teachers should be able to mark attendance, upload study material, and enter test marks.  
- Students should be able to view schedules, attendance, results, and download study material.  
- The system must generate reports like attendance summary, result sheets, and fee status.

# PROCEDURE :

**General Instructions:**  
To perform the experiments in the Software Engineering Virtual Lab using the Coaching Management System as the chosen topic, follow these general steps:  
1. Read the theory related to the experiment.  
2. Understand the simulation based on the CMS problem statement.  
3. (Optional) Take the self-evaluation to assess your understanding.  
4. Attempt and solve the list of given exercises.  
  
**Experiment-Specific Instructions (for CMS):**- Carefully read the problem statement of the Coaching Management System.  
- Identify if there are any inconsistencies or missing information in the requirement specifications.  
- Clearly determine the functional and non-functional requirements of the system.  
- Select the appropriate checkboxes or fields (if using the virtual lab interface), and click on the 'Submit' button to proceed.

# FLOW DIAGRAM:



# E. Preparing Software Requirements Specifications

An SRS (Software Requirements Specification) is a formal document that captures complete software requirements. It includes all the functional and non-functional requirements, use cases, user roles, and interface design. It acts as a legal agreement between the developer and the client. In the context of the Coaching Management System, the SRS will ensure all expected features like student enrollment, teacher assignment, class scheduling, and report generation are well-documented and agreed upon.

# F. Case Study

The Coaching Management System will be used by a coaching institute to manage daily academic operations. It helps automate student registrations, assign teachers to batches, maintain attendance records, schedule classes, upload study materials, and generate progress reports. This system will reduce manual efforts and provide transparency and accountability for both faculty and students.

# G. References

- Ian Sommerville, Software Engineering, 10th Edition  
- IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications  
- Software Engineering Virtual Labs – NPTEL

# H. Conclusion

Requirement analysis is a critical step in software development. By understanding and documenting the expectations of all stakeholders through functional and non-functional requirements, a system can be built with fewer risks and higher user satisfaction. The Coaching Management System project provides a practical understanding of how requirements guide the success of a software solution.