## **Day -01**

## Requirement Specification Document

**Project Name**: E-Learning Platform with Auto Evaluation

**Prepared By** : Abdul Samad (dabdusam)

**Date**  : 06 September, 2025

**1. Project Title**

E-Learning Platform with Auto Evaluation

**2. Overview**

The E-Learning Platform with Auto-Evaluation is a web-based application designed to support online education by providing a digital space where instructors can create courses, students can learn and submit assignments, and the system can automatically evaluate those assignments.

The platform aims to:

1. Simplify course delivery (notes, lectures, quizzes, coding exercises).  
2. Enable assignment uploads and auto-evaluation (objective & coding tasks).  
3. Provide instant feedback and performance reports to students.  
4. Reduce manual work for instructors while ensuring fair, fast, and consistent evaluation.  
5. Support scalable, secure, and user-friendly learning experiences.

**3. Actors (Users of the System)**

1. Students learn and submit.  
2. Instructors teach and evaluate.  
3. Admin manages the platform.  
4. Auto-evaluation system handles automated grading.

**4. Functional Requirements (FR)**

These are the core features of my project:

1. **User Management**
   * Student registration and login
   * Teacher/instructor registration and login
   * Admin user management
2. **Course Management**
   * Create, update, delete courses (by instructors/admins)
   * Enroll students into courses
   * View course materials (lectures, PDFs, videos, notes)
3. **Assignment Management**
   * Upload assignments by students
   * Define assignments by instructors
   * Auto-evaluation of assignments (objective or coding-based)
   * Manual grading option (if needed)
4. **Evaluation System**
   * Auto-check objective questions (MCQs, fill-in-the-blanks, etc.)
   * Auto-evaluate coding/programming assignments (test cases execution)
   * Provide instant feedback/scores to students
   * Generate result reports
5. **Communication & Interaction**
   * Announcements (by instructors/admins)
   * Discussion forums or messaging system
   * Notifications (assignment deadlines, results published, etc.)
6. **Reports & Analytics**
   * Student performance reports
   * Instructor dashboard (class performance, assignment status)
   * Downloadable grade reports

## **5. Non-Functional Requirements (System qualities)**

These describehow the system should work:

1. **Performance** : System should handle multiple students submitting assignments at the same time.
2. **Scalability** : Must support growing number of users and courses.
3. **Reliability** : Ensure auto-evaluation works correctly without crashes.
4. **Security** : User data protection, role-based access, secure login.
5. **Usability** : Easy-to-use interface for students and instructors.
6. **Availability** : System should be available 24/7 with minimal downtime.
7. **Maintainability** : Codebase should be modular and easy to update.

## **6. Deliverables & Requirements**

What the project should deliver at the end:

* **Software Deliverables**
  + E-Learning web application (front-end + back-end)
  + Database (student, courses, assignments, results)
  + Auto-evaluation module (objective & coding assignments)
  + Admin dashboard
* **Documentation Deliverables**
  + Requirements Specification Document (FR + NFR)
  + System Design Document (architecture, ER diagrams, UML)
  + User Manual (how students/instructors use the system)
  + Test Report (unit testing, system testing)
  + Final Project Report & Presentation
* **Hardware/Software Requirements**
  + Hardware: Any standard PC or cloud server
  + Software: Java/Python backend, MySQL/PostgreSQL DB, HTML/CSS/JS frontend, Maven/Gradle build tools

**7. Evaluation Flow**

How assignment handling will work:

1. **Instructor** creates an assignment with deadline & instructions.
2. **Student** uploads assignment file (PDF, Word, or code).
3. **System auto-evaluates** (depending on type):  
   * **Objective** : instantly checked against answer key.
   * **Programming** : run student code against test cases; calculate score.
4. **Evaluation Report Generated** : student sees score, feedback, and grade.
5. **Instructor Dashboard** : view all submissions, override grades if needed.