



ABESEC Ghaziabad
Department of Computer Science & Engineering

SYNOPSIS REPORT
(Session 2025-26)

Project Title: Smart Result				
Project Type : Website				
	Name	Roll Number	Section	Signature
Group member (1)	Gourav Giri	2300320120095	A	
Group member (2)	Harsh Saini	2300320120106	A	
Group member (3)	Kapil Kumar	2300320120121	A	
Project Guide	Dr. Somya Srivastava	Remarks:		
Signature				
Date of submission				

1.1. Problem Introduction

Manual evaluation and result declaration in educational institutions often lead to delays, errors, and lack of transparency. Teachers spend excessive time preparing mark sheets and calculating grades, which can be automated. The **Smart Result** system aims to simplify and digitalize result management by automating result computation, storage, and visualization for both teachers and students.

1.1.1. Motivation

- To reduce manual errors and time consumption during result preparation.
- To provide instant access to marks, grades, and performance analytics.
- To create a transparent system that can be easily scaled across departments.
- To integrate smart technologies that save class time and effort during result distribution

1.1.2. Project Objective

- To develop an automated result management web application using modern technologies.
- To allow faculty to upload marks, compute totals, and publish results efficiently.
- To allow students to view their results securely using their credentials.
- To maintain a digital record for performance tracking and analysis.

1.1.3. Scope of the Project

- Applicable in schools, colleges, and universities for automated result generation.
- Can be integrated with student management systems (attendance, exam scheduling, etc).
- Scalable for multi-department or multi-institution environments.
- Can be extended to generate performance analytics or dashboards for faculty/admin.

1.2. Related Previous Work

"A Review on Student Result Management System" (IRJET) This review paper analyzes various existing Student Result Management Systems, highlighting the problems with traditional manual processing, such as wasted time, effort, and security risks. It identifies the common modules for administrators, faculty, and students and discusses the security flaws found in some systems, emphasizing the need for features like audit trails to ensure data integrity .

"Student Result Management System" (Quest Journals) This paper proposes the design of a specific, modern online Student Result Management System. It details a system intended to automate result-focused tasks like **inputting student grades, automating scoring, and grade reports** for students, teachers, and administrators. The system's proposed design uses a modern microservices architecture.

"Understanding the architecture of a 3-tier application" .This article explains the foundational 3-tier application architecture, a software design pattern that structures an application into three separate logical layers. It details the distinct roles of the **Presentation tier** (the user interface), the **Logic tier** (the "brain" that handles business rules), and the **Data tier** (the database for storage). The article emphasizes that this "separation of concerns" is a major advantage, making the application more scalable, secure, and easier to maintain.

1.3 Software and Hardware requirements

Software Requirements:

- Programming Language: Java / Spring Boot
- Frontend: HTML, CSS, JavaScript, Thymeleaf
- Database: MySQL
- Server: Apache Tomcat
- IDE: IntelliJ IDEA / Eclipse
- Tools: Postman, GitHub

Hardware Requirements:

- Processor: Intel i5 or higher
- RAM: 8 GB minimum
- Storage: 100 GB
- System: Windows / Linux OS

1.4 Proposed Method

Requirement Analysis: Identify user roles (Admin, Faculty, Student) and functional modules.

System Design: Create ER diagrams, data flow diagrams, and architecture layout.

Backend Development: Implement APIs for student management, marks entry, and result calculation using Spring Boot

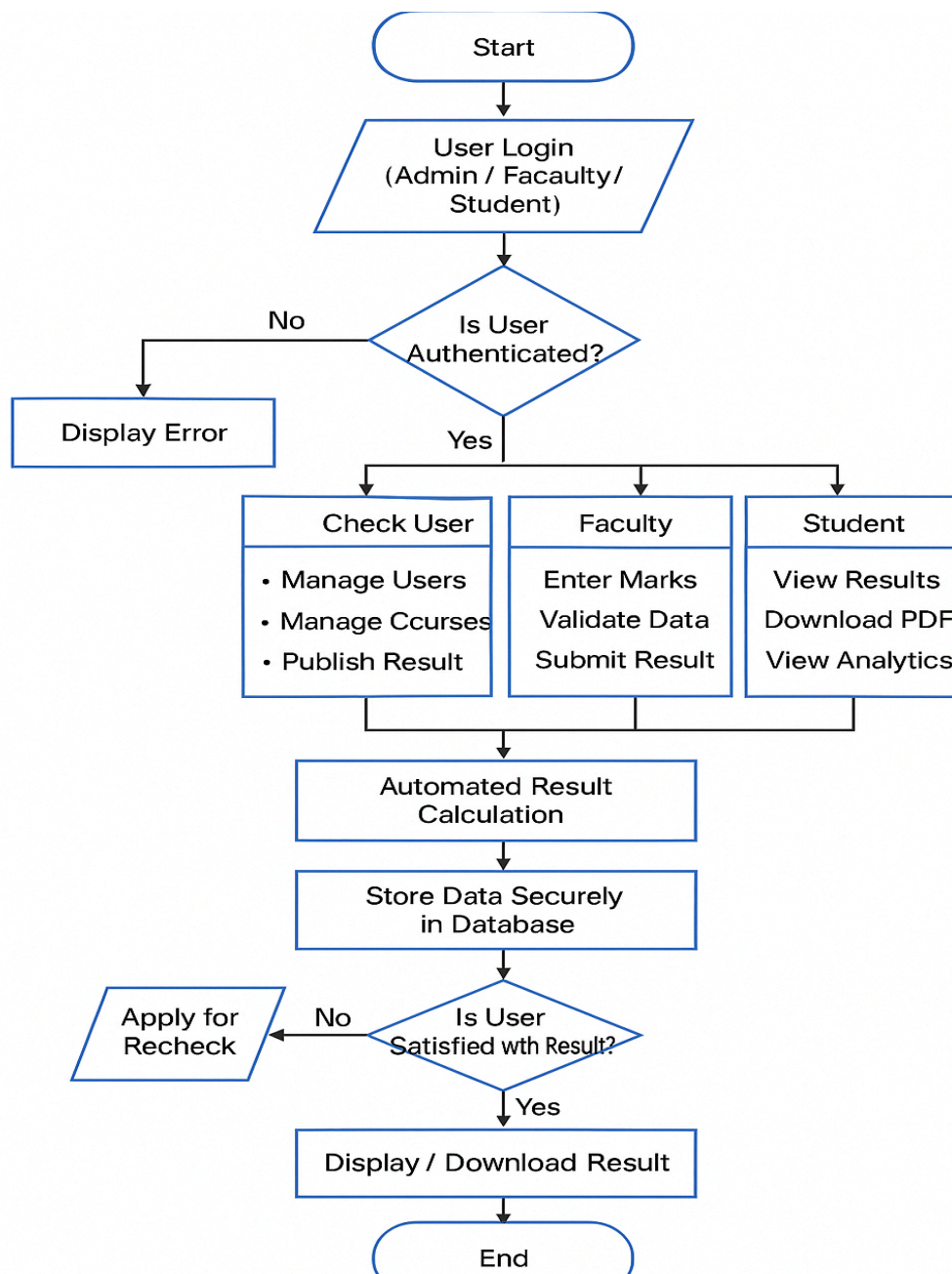
Frontend Development: Build a responsive UI using HTML, CSS, and Thymeleaf templates.

Integration & Testing: Combine frontend with backend and test all modules for accuracy and performance.

Deployment: Deploy on a local or cloud server for testing and demonstration.

Result Visualization: Display reports, graphs, and downloadable result sheets.

1.5 FLOWCHART



1.6 Comparison Table — Proposed Methodology

<i>Aspect</i>	<i>Existing Smart Result Management Systems</i>	<i>Proposed Smart Result Management System</i>
<i>System Architecture</i>	<i>Mostly monolithic or loosely structured 3-tier architecture</i>	<i>Modular Spring Boot–based layered architecture</i> following separation of concerns
<i>Backend Processing</i>	<i>Partially automated result computation with manual verification</i>	<i>Fully automated result processing engine</i> with rule-based computation
<i>Role-Based Access Control</i>	<i>Basic authentication with limited role isolation</i>	<i>Role-Based Access Control (RBAC)</i> with Admin, Faculty, and Student privileges
<i>Database Management</i>	<i>Simple relational storage without optimization</i>	<i>Normalized MySQL database</i> with indexing, relationships, and transaction support
<i>Security Mechanism</i>	<i>Basic login system</i>	<i>Secure authentication, encrypted credentials, and controlled data access</i>
<i>Result Analytics</i>	<i>Static result display</i>	<i>Dynamic analytics with subject-wise, student-wise, and performance reports</i>
<i>Rechecking Mechanism</i>	<i>Manual or offline rechecking process</i>	<i>Online rechecking request workflow</i> integrated into the system
<i>Integration Capability</i>	<i>Standalone system</i>	<i>Extensible design</i> for integration with LMS, attendance, and examination modules

1.5 References

- D. Patel and H. Modi, “A Review on Student Result Management System,” 2008. Available: <https://www.irjet.net/archives/V9/i7/IRJET-V9I7604.pdf>
- D. Sharma and H. Singh, “Student Result Management System Rainy,” Quest Journals Journal of Software Engineering and Simulation, vol. 9, no. 4, pp. 2321–3809, 2023, Available: <https://www.questjournals.org/jses/papers/Vol9-issue-4/09045361.pdf>
- M. Chiaramonte, “Understanding the architecture of a 3-tier application,” vFunction, May 13, 2024. <https://vfunction.com/blog/3-tier-application/>
- [1] “Spring Boot Reference Documentation,” *docs.spring.io*. <https://docs.spring.io/spring-boot/docs/current/reference/html/>