**Scenario : Developing a Student Management Portal**

The **Student Management System (SMS)** is a web-based platform designed to enhance the efficiency of student enrollment, course management, attendance tracking, and performance reporting for educational institutions. This system provides a seamless and user-friendly experience for administrators, teachers, and students, ensuring effective academic management.

**Sprint Breakdown**

**Sprint 1: User Authentication and Dashboard Development**  
- Implement secure user authentication, including login, registration, and role-based access control (RBAC).    
- Design and develop an intuitive dashboard tailored for administrators, teachers, and students, ensuring seamless navigation and accessibility.

**Sprint 2: Student and Course Management**  
- Develop functionality for student profile registration, management, and updates.    
- Implement course creation, enrollment processes, and assignment of students to courses.    
- Enable teachers to manage course content and student assignments efficiently.

**Sprint 3: Attendance Tracking and Reporting**  
- Develop an attendance tracking system that allows teachers to mark and monitor student attendance for each course.    
- Generate comprehensive reports on student attendance and performance analytics, providing valuable insights for academic decision-making.

This structured sprint-based development approach ensures systematic implementation, scalability, and enhanced user experience.

**Frontend(User Interface and User Experience)**

* React.js(for dynamic UI) + Tailwind CSS (for styling)

**Backend**

* node

**Database**

* MongoDB

SCRUM ACTIVITY:

Scenario: Developing a Student Management Portal

Sprint 1: User Authentication and Dashboard UI (2 Weeks)

* User authentication:
* setup backend authentication using JWT for student registration and login.
* adding login logout and registration tab
* Front end authentication integration:
* create login and registration page using React framework.
* The Dashboard UI:
* where we are going to create the dashboard layout including different tabs consisting of student attendance and details using the react
* It is a user-friendly environment and it has a secure login and authenticated responsive and user friendly interface

Retrospectives:

Goals:

Implementing a secure and scalable authentication system, ensuring responsiveness across devices.

What went well:

* + Authentication System Stability
  + Performance Optimization

What Challenges you have faced:

API Response Time- some backend authentication requests were slower than expected

Improvement:

* Documentation improvement with encryption methods for storing and explain how JWT tokens are handled securely.
* Common UI bugs and solutions.

SCRUM ACTIVITY:

Scenario: Developing a Student Management Portal

Sprint 2: Student and Course Management (3 Weeks)

* Student management:
* Creating API for managing student details and handling attendance section
* Implementing features like insert, update, delete and search with the help of students ID and name.
* Course management:
* Creating the API for the course management details where we are going to add credit of the course, course code and the total completion number for the course.
* Assigning course to the students tracking the progress of the course.
* For the extended time limit we can add the certification.

Retrospectives:

Goals:

Efficiently handling student enrolment, profiles and progress tracking, automating scheduling, assignments and grading.

What went well:

* + Smooth enrolment process
  + User-friendly dashboards, course content management

What Challenges you have faced:

Course scheduling conflicts like overlapping of cources.

Improvement:

* Better analytics and reporting.
* Automated Course Scheduling.

SCRUM ACTIVITY:

Scenario: Developing a Student Management Portal

Sprint 3: Attendance Tracking and Reporting (4 Weeks)

* Attendance tracking:
* Uploading student attendance using the barcode with the help of NM prebuilt packages.
* With the help of toaster we can maintain the attendance which typically refers to the UI notification components so with the help of that we can get the attendance sections
* automated reminder for the attendance submission
* Reporting:
* We can generate the attendance reports which are Daily, weekly or monthly basis
* The graphical representation of the attendance.

Retrospectives:

Goals:

Provides real time tracking for accurate record keeping, provide visual insights and data accuracy.

What went well:

* + Automated Attendance Capture
  + Comprehensive Reporting

What Challenges you have faced:

Privacy concerns and scalability issues which caused system slowdowns while processing large datasets.

Improvement:

* Enhance check-in reliability.
* Improve offline mode support.