|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SPRINT | TIMELINE | GOAL | KEY TASK | RESPONSIBLE ROLES | EXPECTED OUTCOMES | Value Vs Effort Analysis |
| 1 | Weeks  1-3 | System Setup & User Management | ✅ Setup project repository and development environment  ✅ Design database schema for users and authentication  ✅ Implement User Registration & Login (RBAC, Password Recovery)  ✅ Develop Question Bank System (Storage, Categorization, Search)  ✅ Build Student Dashboard (Performance Overview, Test Results)  ✅ Initial UI Design and Navigation Structure | **Backend Developers** (Authentication, Database)  **Frontend Developers:**  - Build authentication pages  - Design Student Dashboard UI  - Develop Question Bank UI  - Implement navigation & responsiveness  **QA Team** (Testing User Registration) | 🎯 Users can log in with authentication and access role-based features.  🎯 Faculty can store and retrieve questions in the question bank.  🎯 Students can view their dashboard and test history. | High Value – Low Effort |
| 2 | Weeks  4-5 | Test Management & Auto-Grading | ✅ Develop Basic Test Creation Module (Interface for selecting and arranging questions)  ✅ Implement Automated Scoring for Objective Questions (MCQs)  ✅ Enable Question Printing System for school exams  ✅ Implement Basic Question Randomization to ensure test uniqueness  ✅ Optimize performance for test retrieval and randomization algorithms | **Backend Developers** (Test Logic, Auto-Grading)  **Frontend Developers:**  - Build test creation UI  - Implement automated scoring UI  - Design print-friendly test layouts  - Optimize test-taking experience  **QA Team** (Validate Test Functionality) | 🎯 Faculty can create tests, randomize questions, and print test papers.  🎯 Students can take quizzes with automated scoring. | High Value – Medium Effort |
| 3 | Weeks  6-7 | Approval System & Secure Testing | ✅ Develop Approval System for Faculty Updates (Review and approve question modifications)  ✅ Implement Secure Online Testing Environment (Anti-cheating, Full-Screen Lock)  ✅ Create Tagged Question Removal System (Identify underperforming and overperforming questions)  ✅ Generate Statistical Analysis Reports for question performance | **Backend Developers** (Approval Logic, Secure Test Logic)  **Frontend Developers:**  - Develop approval system UI  - Implement secure test-taking interface  - Create reports dashboard  - Improve UI accessibility  **QA Team** (Security Testing & Bug Fixes) | 🎯 Faculty can submit, review, and approve question changes.  🎯 Secure online tests prevent students from cheating.  🎯 Tagged question reports improve question bank quality. | High Value – High Effort |
| 4 | Weeks  8-9 | Faculty Control & Randomization | ✅ Implement Faculty Assignment & Reassignment System (Admin assigns faculty to roles)  ✅ Develop Subject & Difficulty Selection Control (Super Admin can adjust subject difficulties)  ✅ Improve Advanced Question Randomization (Smart difficulty-based selection)  ✅ Enable System Notifications (Approvals, Faculty Updates) | **Backend Developers** (Admin Roles, Randomization)  **Frontend Developers:**  - Build Faculty Management UI  - Design Subject & Difficulty Control UI  - Improve question selection interface  - Implement system notifications  **QA Team** (Test Faculty Assignments & Notifications) | 🎯 Super Admins can manage faculty roles and subject difficulty.  🎯 Faculty and students receive system notifications for approvals and changes. | Low Value – Low Effort |
| 5 | Weeks 10-11 | UI Enhancements & Campus-Specific Features | ✅ Implement Theme Customization (Light/Dark Mode)  ✅ Develop Gamification Elements (Badges, Leaderboards) (Tentative)  ✅ Restrict access via Campus-Specific Access Control  ✅ Create Inter-Campus Performance Dashboard (Comparison of student performance across campuses)  ✅ Integrate JRSMSU Email for Login (Restrict logins to @jrsmsu.edu.ph) | **Frontend Developers:**  - Implement theme switcher  - Build gamification UI (badges, leaderboards)  - Develop campus-specific access UI  - Design Inter-Campus Dashboard  **Backend Developers** (Campus-Based Features)  **QA Team** (Test Campus-Specific Restrictions) | 🎯 Users can switch themes for better UI experience.  🎯 Gamification (if implemented) encourages student participation.  🎯 Campus-based access and analytics enhance system security and usability. | Low Value – High Effort |
| Deployment | Week 12 | System Testing & Launch | ✅ Conduct User Testing with real students and faculty  ✅ Perform Bug Fixes & Performance Optimization  ✅ Obtain Final Approval from Super Admins and Stakeholders  ✅ Deploy System to live servers | **DevOps Team** (Deployment & Optimization)  **Frontend Developers:**  - Final UI/UX testing  - Fix UI issues before deployment  - Ensure consistency in UI elements  - Implement onboarding tooltips  **QA Team** (Final Testing)  **Project Manager** (Approval & Deployment) | 🎯 Fully deployed, optimized, and ready for use by all JRSMSU users. 🎉 | High Value – High Effort |
|  |  |  |  |  |  |  |