Student Portal Development Project Plan

1. Project Overview

Attribute	Detail
Project Name	Student Portal Development
Project Goal	Design, develop, and deploy a functional Student Portal application, fulfilling all academic requirements for UML modeling, documentation, and software implementation.
Project Start	September 15, 2025 (Estimated)
Project Target End	December 25, 2025
Project Manager	(Self-Assign/Lead Developer)
Target Audience	Students and Academic Staff

2. Key Deliverables & Artifacts

The project requires the submission of both functional software and extensive documentation/design artifacts.

2.1. Documentation & Modeling (UML & Reports)

- **Initial Reports:** Feasibility Report, Project Plan, SRS (IEEE Format).
- **Core UML Models:** ERD, Class Diagram, Component Diagram, Object Diagram, Network Diagram.
- **Behavioral Models:** Use Case Diagrams (with descriptions), Sequence Models, Activity Diagrams, Collaboration & State Charts.
- **Architecture & Deployment:** Systems Architecture (2-tier, 3-tier, N-tier analysis), Main Architecture Diagram, Layered Architecture Models, Deployment Diagram.
- **Advanced UML Concepts:** Implementation of all required associations (dual, self, aggregation, composition), Interface Classes, Realizations, Dependencies, Parameterized Classes, Enumerations, Concurrent/Active Objects.
- **Final Output:** Project Report, Presentation, UML exported diagrams (PNG), UML Project File (Papyrus).

2.2. Software Implementation

- Full Source Code of Working Software.
- SQL Scripts and Database Schema Backup.
- Deployed application on a free hosting platform.
- Final demonstration of the working software.

3. Team Structure and Responsibilities

The project utilizes a small, specialized team structure to maximize efficiency and clear ownership.

Role	Primary Responsibilities	Dependencies
Frontend Developer	User Interface (UI), User Experience (UX), client-side scripting, ensuring cross-browser compatibility.	Requires clear requirements (SRS) and design specifications (UML Models).
Backend Developer	Database design (ERD), API endpoints, business logic implementation, security, and data handling.	Requires clear requirements (SRS) and database design (ERD, Class Diagram).

Integration Specialist	Connecting Frontend and Backend components, initial testing, deployment setup, hosting, and final packaging.	Depends directly on completion of both Frontend and Backend development.

4. Work Breakdown Structure (WBS) and Timeline

The plan is divided into six sequential phases, with concurrent work streams in Phase 3.

ID	Phase / Task Name	Estimated Start Date	Duration (Days)	Responsible Role(s)	Predecessors
1.0	**PHASE 1: INITIATION & PLANNING**	**2025-09-15**	**15**	All/Lead	None
1.1	Feasibility Analysis & Project Planning	2025-09-15	7	All	None
1.2	SRS (IEEE Format) & ERD Creation	2025-09-22	8	All/Backend	1.1
2.0	**PHASE 2: DESIGN & MODELING**	**2025-10-01**	**14**	All/Lead	1.2
2.1	All UML Diagrams & Modeling (Behavioral, Structural, Deployment, etc.)	2025-10-01	14	All	1.2
3.0	**PHASE 3: DEVELOPMENT & IMPLEMENTATION**	**2025-10-15**	**27**	Front & Back	2.1

3.1	**Frontend Development** (UI/UX)	2025-10-15	27	Frontend	2.1
3.2	**Backend Development** (API/DB Logic)	2025-10-15	27	Backend	2.1
4.0	**PHASE 4: TESTING & INTEGRATION**	**2025-11-10**	**15**	Integration	3.1, 3.2
4.1	Initial Testing (Front & Back component testing)	2025-11-10	12	Integration	3.1, 3.2
4.2	Integration (Connecting Front + Back)	2025-11-22	3	Integration	4.1
5.0	**PHASE 5: DOCUMENTATION & DEPLOYMENT**	**2025-11-25**	**31**	All/Integration	4.2
5.1	Prepare Project Report & Artifacts (Description of UML/Code)	2025-11-25	20	All	4.2
5.2	Final Code & SQL Scripts Packaging	2025-12-15	5	Integration	5.1
5.3	Hosting & Deployment on Free Platform	2025-12-20	5	Integration	5.2

6.0	**PHASE 6: FINAL SUBMISSION**	**2025-12-25**	**1**	All	5.3
6.1	Final Submission & Demonstration	2025-12-25	1	All	5.3

5. Risk Assessment and Mitigation

Risk	Impact	Mitigation Strategy
Deployment Issues (Hosting)	Critical—Project cannot be demonstrated.	**MUST** initiate hosting setup (Phase 5.3) well in advance (starting mid-December) to allow 5 days for troubleshooting.
Scope Creep (Adding Features)	Schedule delays, incomplete core functionality.	Strict adherence to the SRS document (Task 1.2). Any feature requests must be approved by the Lead/Manager.
Integration Delays	Failure to deliver a working system.	Frontend and Backend APIs/interfaces must be defined clearly during Phase 2 to ensure compatibility.

End of Document