UNIVERSITY OF SOUTHERN MAINE

Database Systems Course Project Phase III

Fall 2025, Department of Computer Science
Instructor: Behrooz Mansouri
Due: December 8, 2025

Phase III: Web Application Development & Advanced Features

In Phase III, your team will integrate your database with a fully functional web application. This phase requires implementing at least one **advanced database feature** from the second half of the course, ensuring complete frontend–backend–database integration.

Deliverables must be **submitted as a single ZIP file** via Brightspace, linked to your GitHub repository, and accompanied by both a recorded video and an in-class presentation. The ZIP file must follow the naming convention: **TeamName_Phase3_MMDD.zip**.

Required Deliverables

1. **Fully Functional Web Application**
   - React frontend with a RESTful API backend (Flask recommended; alternatives allowed with prior approval).
   - Must be able to run locally from scratch following instructions in the README.
2. **Advanced Database Feature Implementation**
   - Each student must implement at least one advanced feature from:
     - Transaction management / ACID compliance
     - Database indexing & query optimization
     - Concurrency control & locking
     - Database security & authentication
     - Data warehousing or analytics features
     - Backup & recovery procedures
   - Document the feature, where it's implemented, and its effect.
3. **Recorded Video Demonstration (.mp4 or .mov)**
   - Maximum 20 minutes.
   - Show application setup, frontend functionality, backend API calls, database interactions, and advanced features.
4. **In-Class Presentation**
   - 10 minutes.

- Overview of system architecture, key features, and lessons learned.

5. **GitHub Repository**
   - Complete, well-organized, with clear commit history from all members.
   - Must include all frontend, backend, and database code.
6. **User Manual & System Documentation**
   - Installation/setup instructions, usage guide, and API documentation.
   - Include diagrams where helpful (system architecture, data flow).
7. **Performance Analysis & Optimization Report**
   - Identify performance bottlenecks and describe optimizations performed.

Submission Notes

- All deliverables must be included in one ZIP file and pushed to GitHub.
- The system must be installable and runnable following your README without errors.
- Late submissions are not allowed.

Grading Rubric

Criteria	Description	Points
Web Application Functionality & Usability	Fully functional, easy to use, responsive UI, error-free core operations.	4
Frontend–Backend–Database Integration	Smooth communication between components; API endpoints work correctly.	2
Advanced Database Feature Implementation	Correct, meaningful, and well-documented advanced feature.	3
Performance Analysis & Optimization	Identifies issues and demonstrates measurable improvements.	2
Recorded Video Demonstration	Clear, professional, shows the full system and advanced features.	2
Documentation Quality	Complete README, API docs, setup guide, user manual; organized GitHub.	2