**Acadify - Academic Record Management System**

**Project Overview**

**Title:** Develop **Acadify**, a comprehensive web-based academic record management system for Norzagaray College.

**Purpose:** Acadify will serve as a centralized digital platform for managing academic records, streamlining grade management, automating academic achievement recognition, and enhancing overall student success tracking.

**Technology Stack**

* **Frontend:** HTML + CSS + JS with Dark/Light Mode support
* **Backend:** Flask (Connection to database and Some Functions)
* **Database:** MySQL via XAMPP (phpMyAdmin for database management)

**Core Features & User Roles**

**1. Student Dashboard**

* **Grade Viewing:** Complete academic history from 1st-4th year, organized by semester
* **Grade Equivalency Display:** Real-time conversion showing numerical grades and their equivalent points (eg.., Grades is 97 the equivalent is 1.00)
* **Dean's List Tracking:** Personal eligibility status and qualification tracking based on qualification criteria.
* **Theme Toggle:** Dark/Light mode preference with system detection
* **Academic Progress:** Visual representation of academic performance trends

**2. Instructor Portal**

* **Grade Entry:** Manual grade input with validation
* **Batch Upload:** CSV/Excel import functionality for multiple students
* **Auto-Computation:** Automatic calculation of final grades and averages (eg.., Instructor add raw grades Prelim grade is 95, Midterm is 97, Final is 85 and the Final Grade is 92.3 equivalent is 1.50 Superior, formula (Prelim + Midterm + Final and Divide by 3))
* **Grade Submission:** Workflow for submitting grades to registrar approval
* **Class Management:** View assigned classes and the student list coming from the registrar.

**3. Registrar/Admin Dashboard**

* **Student Records Management:** Complete oversight of all student academic data
* **Grade Approval System:** Lock/unlock and approve instructor-submitted grades
* **Report Generation:** Comprehensive academic reports and official transcripts
* **Audit Trail Management:** Track all system changes and access logs
* **Dean's List Processing:** Automated computation and ranking of academic achievers
* **Grade Approval System:** Lock/unlock and approve instructor-submitted grades.
* **Workflow**: The Registrar assigns the subject to the instructor, then forwards it to them so they can encode the grades. Once the instructor has entered the grades, they can no longer change them unless the MIS/IT resets the access to allow re-editing.

**4. Dean's Office Interface**

* **Academic Rankings:** Real-time view of top-performing students
* **Dean's List Management:** Semester-based achiever listings
* **Recognition Reports:** Generate official PDF reports for ceremonies
* **Department Analytics:** Performance statistics and trends

**5. MIS/IT Staff Panel**

* **System Security:** User access management and security monitoring
* **Backup Management:** Automated and manual backup procedures
* **System Logs:** Comprehensive logging and error tracking
* **Maintenance Tools:** Database optimization and system health monitoring
* **Account Management:** The MIS/IT staff will create the accounts of professors and students, then forward these accounts to the Registrar. MIS/IT is also the one who can add or remove accounts from the system. If a password is forgotten, they are also responsible for handling it.
* **Scope of Departments:** This applies to the departments at Norzagaray College:
* BSCS/ACT – Bachelor of Science in Computer Science / Associate in Computer Technology
* BEED – Bachelor of Elementary Education
* BSHM – Bachelor of Science in Hospitality Management
* BSED – Bachelor of Secondary Education
* Grade Encoding Schedule: MIS/IT will also schedule when instructors can encode grades. They will set a date range during which grade encoding is allowed. After that date passes, instructors can no longer input grades; they will only be able to view them.

**Detailed Functional Specifications**

**Grade Management System**

* **Grading Scale (Norzagaray College Standard):**
  + 98-100 = 1.00 (Excellent)
  + 95-97 = 1.25 (Outstanding)
  + 92-94 = 1.50 (Superior)
  + 89-91 = 1.75 (Very Good)
  + 86-88 = 2.00 (Good)
  + 83-85 = 2.25 (Satisfactory)
  + 80-82 = 2.50 (Fairly Satisfactory)
  + 76-79 = 2.75 (Fair)
  + 75 = 3.00 (Passed)
  + Below 75 = 5.00 (Failed)
  + AW (Authorized Withdrawal)
  + UW (Unauthorized Withdrawal)
  + INC (Incomplete)

**Dean's List Automation**

* **Qualification Criteria:**
  + A general Weighted Average 1.75 or better
  + No failing grades (5.00) in any subject
  + No incomplete (INC) or withdrawal (AW/UW) marks
  + Regular student with 18 units and above
  + No grade lower than 2.00
* **Automated Processing:**
  + Real-time qualification checking
  + Automatic student tagging and notification
  + Semester-based ranking generation
  + Export functionality for recognition ceremonies

**Dark/Light Mode Implementation**

* **Auto-Detection:** System theme preference detection using prefers-color-scheme
* **Manual Toggle:** User-accessible theme switcher in navigation bar
* **Persistence:** Theme preference storage (localStorage and database backup)
* **Accessibility:** WCAG compliance for both themes

**Technical Requirements**

**Security & Compliance**

* **Role-Based Access Control (RBAC)**
* **CHED (Commission on Higher Education) compliance**
* **Data Privacy Act compliance**
* **Secure password handling and session management**
* **Audit logging for all critical operations**

**Performance & Scalability**

* **Optimized database queries** for grade computations
* **Efficient ranking algorithms** for large student populations
* **Responsive design** for various devices and screen sizes
* **Scalable architecture** to handle 1,500+ students

**UI/UX Requirements**

**Design Principles**

* **Modern, clean interface** following current web design trends
* **Intuitive navigation** with clear user flow
* **Responsive layout** that works on desktop, tablet, and mobile
* **Accessibility features** including keyboard navigation and screen reader support
* **Fast loading times** with optimized assets and lazy loading

**Branding Elements**

* **Custom Logo Integration:** Include logo.png in the top-left corner of navigation bar
* **Clickable Logo:** Logo should redirect to homepage when clicked
* **Consistent Branding:** Acadify logo and color scheme throughout the application
* **Professional Appearance:** Maintain institutional credibility and trust

**Development Deliverables**

**Code Organization Requirements**

* **Well-commented code** with clear explanations for each function and module
* **Modular structure** with separate files for different functionalities
* **Configuration files** for easy environment setup and deployment
* **Documentation** including setup instructions and user guides

**XAMPP Integration Code**

* **Database connection setup** using Flask-SQLAlchemy
* **MySQL configuration** for XAMPP environment
* **Error handling** for database operations
* **Connection pooling** for optimal performance

**Expected Final Output**

A complete, production-ready **Acadify web application** featuring:

1. **Comprehensive grade management** from freshman to senior year
2. **Automated Dean's List computation and ranking** system
3. **Modern, responsive UI** with full Dark/Light mode support
4. **Role-based access** for all user types with appropriate permissions
5. **Robust reporting system** with PDF generation capabilities
6. **Secure, scalable architecture** ready for institutional deployment
7. Even if the database isn't working yet, just show me how to connect to the database. Just the code only
8. improve the login dashboard with a demo at the bottom for student, instructor, mis/it, registrar, dean account and there is a toggled password.
9. Start with the overall design first, then the organization of the files: put all dashboard designs in the **templates** folder, and keep all functions in a single Flask file — put every possible function there (including the MySQL database connection and the login) so it won't be scattered. In short: a **templates** folder and a **main.py** to run.