System Overview

- **AcademIQ** is an intelligent academic advisory system designed to help university students understand, track, and enhance their academic performance. The system collects academic data and analyze it to generate personalized insights, predictions, and improvement recommendations. The goal is to replace traditional static advising with a **dynamic, data-driven guidance system** that adapts to each student's behavior and performance patterns.

- **Stackholders:**

Students (Primary Users)	Receive insights, forecasts, and personalized guidance to improve academic performance.
Software Developers / AI Engineers	Responsible for designing, implementing, and maintaining the system.
Project Supervisor / Committee	Ensure validity, monitor progress, evaluate deliverables.
Data Providers (Registration, LMS Systems)	Provide academic and attendance datasets.

- **Use-cases brief Description:** to be implemented

2. Requirements

Functional Requirements

.1. User Management

FR1.1 User Registration

- Students must be able to create an account using email and password.

FR1.2 User Login & Authentication

- The system must authenticate users using secure credential validation.
- Role-based access must differentiate Student and Admin users.

FR1.3 Profile Management

- Students can update personal information and academic preferences.

.2. Academic Data Management

FR2.1 Manual Grade Entry

- Students can manually input grades for courses per semester.

FR2.2 Bulk Grade Upload (Optional)

- Students can upload transcripts or CSV files to automatically extract grades.

FR2.3 Course & Semester Management

- Students can add, remove, or edit semesters and the courses taken in each.

FR2.4 Course Catalog Access

- Students can browse a list of available courses, including prerequisites and difficulty.

FR2.5 Course Import

- Students can import his their course data from LMS.

.3. Performance Insights & Analytics

FR3.1 Academic Dashboard View

- Display GPA, semester averages, course performance, and progress charts.

FR3.2 Insights Generation

- The system must compute academic insights such as:
    - Strongest/weakest subjects
    - Performance patterns
    - Study habits indicators
    - Historical performance trends

FR3.3 Performance Detection

- Detect sudden GPA drops or inconsistent grades and notify the student.

.4. Prediction Services (AI & ML Models)

FR4.1 GPA Prediction

- Predict future GPA based on:
    - Previous semesters
    - Course difficulty
    - Expected load

FR4.2 Course Grade Prediction

- Predict expected grade for each enrolled or planned course.

FR4.3 Semester Outcome Forecast

- Predict if the upcoming semester is high risk, medium risk, or safe.

FR4.4 Overload Detection

- Identify if the student’s planned course load is too heavy.

5. Recommendation Engine

FR5.1 Personalized Study Recommendations

- Provide targeted suggestions such as:
    - Time allocation
    - Learning strategies
    - Prioritization tips

FR5.2 Course Load Optimization

- Recommend a balanced selection of courses.

FR5.3 Improvement Tips

- Suggest actions when a student's performance declines.

6. AI Advisor Chatbot

FR6.1 AI Conversation

- Students can ask academic questions and get guidance.

FR6.2 Insight Explanation

- Chatbot can explain predictions in simple terms.

FR6.3 Personalized Study Planning

- Users can request:
    - A study plan
    - Advice for specific courses
    - Motivation and reminders

FR6.4 Thread-Based History

- Conversations are stored and reloaded as threads.

7. Admin Features

FR7.1 Course Management

- Admins can add/update/delete courses.

FR7.2 User Monitoring

- Admins can view:
    - Student metrics
    - Risk levels
    - System-wide analytics

FR7.3 Dataset Management

- Admins can manage datasets used for training ML models.

8. System Management

FR8.1 Logging

- System logs critical actions, errors, and performance metrics.

FR8.2 Data Backup

- Academic and user data must be backed up periodically.

FR8.3 Model Updating

- System admins can retrain/update ML models.

Non-Functional Requirements:

- **Performance**:
- The system shall return dashboard analytics within **2 seconds**.
  AI prediction requests shall be processed within **1–3 seconds**.
  The server shall handle at least 100 concurrent users.

- **Scalability**:

- **Security**: Authentication, authorization, encryption.
- **Reliability**: Disaster recovery, failover systems, automatic database backups every **24 hours,**
- **Maintainability**:

3. System Architecture

- **High-Level Diagram**: in progress
- **Technology Stack**:
- **Frontend**: (React)
- **Backend**: (Python)
- **Database**: (PostgreSQL)
- **Infrastructure**: (Docker)
- **System Components**:

Student Web

Provides the user interface for all student interactions:

1.1 Student Dashboard

- Academic dashboard
- GPA calculator, forecasting, insights
- Course difficulty predictions (did not mention)
- Personalized recommendations
- Study plan generator
- Advisor chatbot
- Goal setting
- Grade entry/import

1.2 Admin Dashboard

Used by admins/faculty:

- Manage student data
- View aggregated analytics

- Manage courses
- Monitor risk indicators

Handles:

- Secure user login/registration
- Token-based authentication (JWT)
- Role-based access control (Student, Admin)
- Password encryption
- Session management

2. Backend Services Layer

The heart of the system, Communications.

2.1 API Gateway

Provides unified REST API endpoints to frontend apps.

Responsibilities:

- Authentication & authorization
- Routing requests to internal services

FastAPI

2.2 Authentication & User Management Service

Handles:

- Secure user login/registration

- Token-based authentication (Bcrypt)
- Role-based access control (Student, Admin)
- Password encryption
- Session management

2.3 Student Profile Service

Maintains all student-related functional data:

- Personal info
- Academic history
- Transcript/grades
- Semester logs
- Goals and preferences

2.4 Grades Processing Service

Responsible for:

- Predicting GPA from planned courses
- Detecting inconsistencies (missing grades, invalid entries)

2.5 Academic Insights Service

++++

Generates:

- Personalized academic insights
- Performance patterns
- Strength/weakness identification
- Behavior analysis

Features:

- Uses machine learning predictions
- Triggers risk alerts
- Provides tailored recommendations

2.6 Risk Prediction & Forecasting Service

AI module utilizing ML models to forecast:

- GPA
- Probability of course failure

Inputs:

- Student history
- Study behavior
- Past student outcomes (performance vectors)

Outputs:

- Risk classification (Low / Medium / High)

- Expected grades per course
- GPA forecast range

2.8 Recommendation Engine

Generates:

- Weekly study advice
- Course load adjustments (Course Advising)
- Time allocation suggestions (schedule planning)
- Personalized tips

Uses:

- Prediction service outputs
- Student behavior patterns
- Smart rule-based logic + AI models

2.9 AI Advisor Chatbot Service

Uses:

- OpenAI API
- Context/state memory
- Thread-style interaction

Functions:

- Explains insights
- Guides study planning
- Answers academic questions
- Generates personalized understanding

2.10 System Reports & Visualization Service

Handles:

- Academic reports
- Performance graphs
- Admin monitoring analytics

Exports:

- PDF
- Web visual dashboards

3. Data & Intelligence Layer

The backbone of the system.

3.1 Primary Database

Stores all core system data:

Tables:

- Students

- Courses
- Grades
- Predictions
- Recommendations
- Chatbot message threads

Tech: PostgreSQL

3.2 File Storage

Stores:

- Reports
- Logs
- Visual data

Tech**:**

- Google Cloud Storage

4. Third-Party Services & External APIs

Your project will involve several external integrations:

4.1 OpenAI API (Main AI Engine)

Used for:

- AI advisor chatbot
- Explanation of insights
- NLP-based guidance
- Data classification and summarization tasks

4.2 Authentication Providers

- Firebase Auth
- OAuth for university integration

5. How All Components Work Together (High-Level Summary)

1. **Frontend** sends requests to →
2. **API Gateway** authenticates →
3. Routes request to the **Backend Service** →
4. Backend pulls data from **Database** →
5. If AI processing is needed →
    - It communicates with **AI Models**
    - Or sends tasks to **OpenAI API**
6. The backend returns results to the **frontend**
7. Visualized in dashboards, reports, or chatbot responses

Module 1: Authentication & User Management Module

Purpose

Handles secure login, registration, role-based access control.

Inputs

- Username / email
- Password
- User role (Student)

Outputs

- JWT / OAuth tokens
- Authenticated session
- User profile information
- Access permissions

Dependencies

- User database
- Encryption library (bcrypt)

- JWT/OAuth provider
- Admin panel for role management

Flow

1. User enters credentials
2. System verifies hashed password
3. Generates JWT token
4. Loads the appropriate dashboard
5. Applies role-based access control

Module 2: Student Data Ingestion & Processing Module

Purpose

Collects raw academic data and processes it for analytics and model consumption.

Inputs

- Student Transcript
- Attendance data
- Course information
- Assessment breakdowns
- Uploaded CSV/Excel datasets (optional)

Outputs

- Cleaned, validated, standardized dataset
- Student performance vectors
- Stored records in the database

Dependencies

- University LMS
- Database storage

Flow

1. Receive academic data
2. Validate format, completeness
3. Clean & normalize data
4. Compute derived metrics (percentages, GPA)
5. Store structured data

Module 3: Analytics & Visualization Module

Purpose

Generates charts, dashboards, and analytical summaries of student performance.

Inputs

- Clean student dataset

- Historical academic performance
- Attendance rates
- Course difficulty profiles

Outputs

- GPA trend charts
- Strength/weakness metrics
- Performance breakdown visualizations
- Comparative/charts dashboards

Dependencies

- Data processing module
- Visualization libraries (Chart.js)
- Frontend user interface

Flow

1. Fetch student data from DB
2. Calculate performance metrics
3. Generate analytical summaries (LLMs)
4. Render visual charts on UI

Module 5: Recommendation Engine Module

Purpose

Produces personalized academic guidance based on analytics, predictions, and performance patterns.

Inputs

- AI predictions
- Student strengths/weaknesses (Performance module)
- Risk scores

Outputs

- Study strategy recommendations
- Alerts (e.g., “low attendance risk”)
- Course load suggestions
- Behavioral insights (e.g., “you do better in morning exams”)

Dependencies

- Analytics module
- AI module
- NLP component for generating text
- Rule-based engine (optional)

Flow

1. Receive prediction results
2. Analyze weaknesses, risks, and trends
3. Apply recommendation rules + AI personalization
4. Generate natural-language advice
5. Send insights to UI

Module 6: Notification & Alerts Module

This module is essential because your system sends:

- Risk warnings
- Performance alerts
- Recommendation updates
- Weekly summaries

Without a dedicated module, this behavior looks incomplete in the design.

Module 7: Intelligent Quiz Generation & Assessment Engine

Provides students with personalized quizzes that help reinforce weak areas and improve academic performance.

Inputs

- Course
- AI Core
- Student strengths/weaknesses
- Course difficulty data
- Performance trends

Outputs

- Personalized Quiz
- Answer Explanation
- Correction

Dependencies

- Analytics module
- Course Info
- AI module
- NLU/NLP component for generating text

Flow

1. Receive Course Info
2. Receive performance results
3. Analyze weaknesses, risks, and trends

4. Generate Quiz
5. Correct & Explain
6. Save results and measure performance
7. Update AI

Website (Frontend + Experience Layer)

1.1 Dashboard & Insights

- Personalized academic dashboard
- GPA, CGPA, course progress, and trajectory visualization
- Highlight strong/weak subjects
- Early warnings (risk flags)

1.2 Grade Tracking

- Automated grade import OR manual grade entry
- GPA forecasting (expected vs achieved)
- Grade improvement tracking

1.3 Personalized Recommendations

- Study improvement suggestions
- Weekly performance summaries
- Behavior-based learning patterns

1.4 Goal Setting

- Student sets targets (GPA goals, course goals)
- AI checks progress and notifies student if they are at risk or ahead

1.5 Study Plan Generator

AI uses historical patterns + current workload to produce:

- Weekly study plan
- Highlight overloaded weeks
- Recommend time allocation per course
-

4. Advisor-Student Interaction Module

4.1 Smart Advisor Chatbot

AI-powered academic assistant that can:

- Answer student academic questions
- Interpret student performance
- Suggest study techniques
- Explain insights
- Help with course planning based on student performance vector and course catalogs.

4.2 Message Threads

- Student and advisor have a conversation timeline
- Stored as private journaling-style threads
- Context-aware, not one-off replies

5. Admin & Management Module

5.1 Admin Dashboard

- Manage student accounts
- View aggregated analytics

5.2 University/Faculty Integration

- Import academic data
- Manage course structure

- Set curriculum details

7. Security, Privacy & User Management

7.1 Authentication

- Login, registration
- Secure token sessions

7.2 Role Management

- Admin
- Student

7.3 Data Privacy

- Encrypted academic data
- “My AI data settings” panel

📈 8. Reports & Visualizations Module

8.1 Smart Reports

- Student performance evolution
- Semester insights
- Course difficulty breakdown

8.2 Data Visualizations

- GPA time series
- Progress bars
- Risk indicators
- Pie charts & heatmaps

9. Overall Value Proposition

Your system:

- Understands the student academically
- Learns from their history
- Predicts their future performance
- Advises them personally
- Helps them plan
- Uses AI to deliver meaningful insights
- Fixes the chaos of grading systems with a unified model
- Gives students personalized academic development tools