

Final Report: Course Recommendation Expert System

An Intelligent Rule-Based System for Academic Course Selection

1. Project Title

Course Recommendation Expert System

2. Team Members & Roles

Team Member	Student ID	Role & Responsibilities
Mohamed Yasser	20100290	Project Lead, Inference Engine Developer, Streamlit UI
Ahmed Hanny	2010390	Knowledge Base Editor, Testing, Documentation
Islam Ali	20100419	Streamlit Frontend, Course Data Validation

3. Application Description

Overview

The Course Recommendation Expert System is an intelligent, rule-based system that recommends university courses to students based on their academic records and institutional policies. Built with Python and the Experta library, it models prerequisites, co-requisites, credit limits, and other academic rules to provide personalized, explainable recommendations.

Functionality

- Recommends courses based on CGPA, semester, passed/failed courses, and track
- Enforces prerequisites, co-requisites, and credit hour limits
- Provides detailed explanations for each recommendation or restriction
- Offers a modern web interface (Streamlit) and a command-line interface
- Includes a knowledge base editor for course management

Domain Modeled

University course selection and advising, including academic policies, prerequisites, and student performance evaluation.

4. Rules: List and Explain All Implemented Rules

Rule 1: Prerequisite Enforcement

Courses are only recommended if the student has passed all required prerequisite courses.

Example:

Course: CSE201 (Data Structures)

Prerequisite: CSE101

 Skipped if CSE101 is not passed.

"Missing prerequisites: CSE101"


Rule 2: Co-requisite Enforcement

Courses with co-requisites require those co-courses to be taken simultaneously or already passed.

Example:

Course: PHY102

Co-requisite: PHY101

 Skipped if PHY101 is not passed or concurrently selected.


"Missing corequisites: PHY101"

Rule 3: Credit Hour Limit Based on CGPA

The system restricts total credit hours per semester based on the student's CGPA:


CGPA Range	Max Credit Hours
Below 2.0	12 credits
2.0 – 3.0	15 credits

CGPA Range	Max Credit Hours
Above 3.0	18 credits

 *Course skipped if it would exceed this limit.*
"Would exceed credit limit (15 > 12)"



Rule 4: Semester and Track Filtering

Courses must match the selected semester and the student's academic track.

 *Skipped if not offered in current semester or outside track scope.*
"Track mismatch - AI Engineering"
"Not offered in Fall semester"

Rule 5: Exclude Passed or Failed Courses



Previously passed or failed courses are not recommended again.

 *Passed: Skipped*
 *Failed: Skipped with "Course previously failed - may need retaking"*

Rule 6: Explanation System

Every recommendation or restriction is stored with a clear explanation.

Example:

 *"CSE301 is recommended because all prerequisites are satisfied."*
 *"CSE402 not recommended due to missing prerequisite MATH204."*

5. Knowledge Base: Content and Editor Functionality

Content

The knowledge base is stored in `CE_Cloud.csv`, containing course codes, names, prerequisites, co-requisites, credits, semester, and track information.

Sample Entry

Code	Name	Prerequisites	Co-requisites	Credits	Semester	Track
CSE101	Intro to CS	None	None	3	Fall	Cloud
CSE201	Data Structures	CSE101	None	3	Spring	Cloud

Editor Functionality

The `kb_editor.py` script allows users to add, edit, or remove courses using a command-line interface, providing full CRUD operations for knowledge base management.

6. Inference Engine: Reasoning Process and Implementation

Overview

The inference engine is the core component responsible for evaluating student profiles against academic policies. It uses the **Experta** rule engine to apply logical conditions and produce explainable recommendations.

Reasoning Process

For each course in the system, the following checks are performed in order:

1. **Prerequisites Met** - Student must have passed all required courses
2. **Co-requisites Met or Included** - Either passed or added to current recommendations
3. **Semester Match** - The course must be offered in the selected semester
4. **Track Match** - The course must match the student's academic track (e.g., Cloud, All)
5. **Credit Limit Respected** - Total credit hours must not exceed the student's limit
6. **Not Already Passed or Failed** - Already taken or failed courses are excluded

Implementation

- Defined with `StudentInfo`, `Course`, and `Recommendation` facts in Experta
- Rule methods are defined with `@Rule()` decorators
- Course facts are declared for each CSV row

- Internal lists hold recommended and skipped courses with explanations
- Rules are dynamically matched and explanations are attached per outcome

☑ Example Flow

Inputs:

CGPA = 3.2, Semester = Fall, Passed = [MAT111, CSE014], Failed = []

Results:

☑ CSE316 passes all checks → Recommended

✗ CSE201 fails prerequisite check → Skipped: *"Missing prerequisites: CSE101"*

7. User Interaction: Streamlit Web Session

Inputs

- CGPA: 3.2
- Semester: Fall
- Passed: MAT111, CSE014
- Failed: (*none*)

Outputs

- ☑ Recommended Courses
- ✗ Skipped Courses with Explanations
- 📄 Download CSV functionality
- ▼ Expandable course cards for detailed information

8. Explanation System

The explanation system is integrated into every decision within the inference engine. All recommendations and restrictions are justified and displayed to the user through the Streamlit interface via expandable sections under "Recommended" and "Not Recommended" categories.

9. Streamlit Interface: Design & Features

Design

- Sidebar for input (CGPA, Semester, Passed/Failed Courses)
- Main view for recommendations, explanations, and download functionality

Features

- Interactive inputs with real-time validation
- Expandable feedback sections
- CSV export capability
- Dynamic credit tracking
- Built with `streamlit`, `pandas`, and `experta`

10. GitHub Repository

Repository: <https://github.com/Mohrizk90/course-recommendation-system>

The repository includes collaborative development via branches and commits, containing all source code, CSV data files, Streamlit application, and comprehensive documentation.

Course Recommendation Expert System - Final Report

Generated on: