

AI Curriculum Planner

Phillip Diec

922800213

Nexusstar12

Checkpoint #	Date Submitted
Checkpoint 1 V1	09/18/2023
Checkpoint 1 V2	09/25/2023
Checkpoint 2 V1	10/04/2023
Checkpoint 3 V1	10/25/2023

Table of contents

1. Introduction	1
1.1. Overview	1
2. Project Description	4
2.1. Purpose and Scope	4
2.2. Key Features	5
3. Use Cases	6
3.1. Working Students	6
3.2. Students Who Don't Want to Rely on Counselors	6
3.3. Students Balancing Athletic and Academic Pursuits	7
3.4. Parent-Teacher Performance Review	7
3.5. Group Presentation Preparation	8
4. Potential Benefits for Existing Software/Products	9
4.1. Canvas	9
4.2. iLearn	9
5. Functional Database Requirements	11
5.1. User	11
5.2. Registered User	11
5.3. Administration	11
5.4. Student	11
5.5. Educator	12
5.6. Courses	12
5.7. AI Algorithm	13
5.8. Feedback System	13
5.9. Global Tren Integration	13
5.10. Group Task and Presentation	13
5.11. Lab Booking	14

5.12. Performance Review	14
5.13. Notifications	14
5.14. Community	14
5.15. Resources	14
5.16. Account	14
5.17. Tutorials	15
5.18. Articles	15
5.19. Feedback	15
6. Non-functional Database Requirements	16
6.1. Performance	16
6.2. Storage	16
6.3. Security	16
6.4. Scalability	16
6.5. Reliability	16
6.6. Usability	17
6.7. Integration	17
6.8. Maintainability	17
6.9. Compliance	17
6.10. Environmental	17
6.11. Legal	17
6.12. Content	17
6.13. Compatibility	18
6.14. Organizational	18
7. Entity Relationship Diagram (ERD)	19
8. Entities Description	19
9. EER (Entity Establishment Relationship) Diagram	27
10. Constraints Description	28

Project Description

- **AI Curriculum Planner**

Within the realm of education, the traditional education systems often rely on set curriculums, leaving students navigating a confusing and static academic pathway. These methods don't consider the personal lifestyles and needs of today's students. This is where our AI Curriculum Planner comes into play. Using advanced AI, our system offers real-time course adjustments that are specifically tailored to students who are caught in complicated or unique situations.

Regardless of whether you're a working student, an athlete looking to balance sports and academics, or someone who just wants to work independently, our tool is designed to help make your journey less of a hassle. This system doesn't stop at individual planning either, it also encourages people to work together and enable group collaboration, streamlining and making group projects go much smoother than ever before. Teachers and parents can also take advantage of this by accessing detailed student data to make sure their performance is up to pace with their peers, this opens new opportunities to not only help guide students who are in need, but also allows teachers, parents, and students to all connect with one another. Overall, the AI Curriculum Planner's purpose is to make the journey easier for students by allowing them to simplify, personalize, and enhance their academic planning, ensuring that they get the best learning experience possible in an environment that can be stressful at times.

- **High Level Non-Technical Description**

Our AI Curriculum Planner database system will act as a central repository of curriculum plans, learning modules, and assessment results along with real-time feedback. It will utilize AI algorithms to check and search through data to identify trends that will benefit students in finding their educational requirements all the while suggesting curriculum changes accordingly. And to make things easier for students, we have a feature where students shall be able to input their academic organization and it will provide them with previous helpful comments from students who have attended said school in the past, this can be filtered to make it easier for students to find useful tips on their academic journey. We'll also have a feature where the AI will automatically suggest changes in curriculum based on individual student's progress, preferences, and availability, that way, students won't have to stress sifting through an abundance of classes to find the ones they need to graduate, students shall be able to modify these suggestions to fit with their respective schedules. There will be other features such as real-time feedback, integration with global trends, and holistic data views. Real-time feedback allows students to have access to the most up-to-date information on their classes as fast as possible, while the integration of global trends and holistic data views will allow the AI to adjust and swap courses based on educational trends such as demographic, performance, and feedback data.

- Use Cases

Use Case: Working Students

Actors: Working Student (Sarah)

Sarah is a university student who is juggling between her part-time job and her university courses, she faces challenges in managing her schedule. And using the AI Curriculum Planner, she's able to input her working hours and academic preferences and the system recognizes her limited availability. Doing so, the AI delicately selects courses that fit within her schedule and range of preferences. Additionally, the real-time feedback feature provides Sarah with insights from other students who are also working and going to school at the same time, offering tips to Sarah on how to balance work and studies effectively.

Use Case: Students Who Don't Want to Rely on Counselors

Actors: Independent Student (John)

John is a student who relies on himself most of the time, and on the rare occasions that he has visited academic advisors or counselors, he always left each session unsatisfied. When John stumbled upon the AI Curriculum Planner, he was automatically drawn to it, as it removed the need for him to meet with counselors in person and take matters into his own hands related to his academic courses. Upon analyzing John's past performances and goals, the AI generates a list of recommended courses and resources for John to utilize. The system also tells John what prerequisites or corequisites he might need to take certain courses, ensuring that John stays fully informed and can independently make academic decisions without having to ever worry about meeting counselors or advisors again.

Use Case: Students Balancing Athletic and Academic Pursuits

Actors: Student Athlete (James)

John is a prominent member of his college basketball team. As an athlete, his schedules are filled with rigorous training sessions, team meetings, and travels for matches. His commitment to the team is unwavering, but he is just as equally determined to excel academically, finding that balance, however, has been challenging for John. Upon registering with the AI Curriculum Planner, the system prompts him to input his training schedules, match days, and potential out-of-town tournaments. Using this data, the planner provides James with a fully tailored and personalized course schedule. It prioritizes courses that have flexible timings, attendance options, and professors known for their understanding of student-athlete schedules.

Use Case: Parent-Teacher Performance Review

Actors: Teachers (Ms. Durham), Students (Alex), and Parents

Ms. Durham, a high school teacher, schedules a performance review meeting with the parents of Alex, one of her students. Before the meeting takes place, she utilizes the AI Curriculum Planner to pull up Alex's performance metrics, feedback, and course progression. On the other hand, Alex also adds his own feedback and reflections on his performance, areas that he feels good about, and areas that he feels are still lacking and need improving. During the meeting, his parents have access to this data, which leads to more productive conversations on Alex's academic development, strengths, weaknesses, and future goals in the course.

Use Case: Group Presentation Preparation

Actors: Students (Anna, Ben, Carlos), Teachers (Professor Kent), Media Lab Technician (Ms. Anders)

Professor Kent's class requires a collaborative presentation where students must come together and form a team to produce multimedia-rich reports. Anna, Ben, and Carlos form a team to explore a certain topic. Their goal is to integrate textual analysis, historical context, and multimedia elements like videos and audio clips into their presentation. Using the AI Curriculum Planner, the team lists their tasks and gets to work. Carlos finds that he needs to use the university's media lab for editing, using the Planner's system, he is able to book specific hours with Ms. Anders, the media lab technician, and she receives a notification about the booking. After the three put together their findings, they then submit a draft via the Planner. Professor Kent reviews it and provides them with feedback directly on the platform, highlighting areas needed for improvement. With the teamwork enabled by the AI Curriculum Planner, the three students were able to combine their efforts to deliver a compelling presentation that earns them high marks from Professor Kent, showcasing the capabilities of the AI Curriculum Planner in streamlining collaboration and resource management for a more straightforward academic task.

• Potential Benefits for Existing Software/Products

- Canvas Learning Management System (LMS)

Why: Canvas is one of the more popular Learning Management System used by educational institutions worldwide to manage online courses, facilitate communications between teachers and students, and to distribute assignments along with grades. By integrating with the AI Curriculum Planner, Canvas can enhance its already existing management system even further. For example, Canvas could proactively suggest classes to students based on their academic history, personal schedules and preferences, allowing students to have a much more personalized and intuitive time when it comes to the enrollment process. Not only that, but with the real-time feedback feature, the AI Curriculum Planner can offer students insights and advice from peers that will help them in finding courses and even professors who match their preferences in the learning environment. This wouldn't only benefit students, but educators can also use this AI system to modify or suggest changes in the course content based on the trends that are identified by the AI, ensuring that the curriculum stays up-to-date and relevant to allow for a more engaging and enriching learning experience.

- iLearn

Why: iLearn is an e-learning platform also used by various institutions to offer online courses and modules. Its main features include course creation, distribution, communication tools, and assessment capabilities. There are significant advantages iLearn can gain from integrating the AI Curriculum Planner such as the ability for students to personalize their courses based on their past performance, work commitments, and extracurricular activities, the AI system can suggest the best classes to take each semester, allowing students to effectively handle the workload and

achieve their desired results. Real-time feedback will let students on iLearn to receive instant reviews and insights from their peers who've taken the same classes in the past to let them know what to expect, this, in turn, will help students when it's time to consider which courses to enroll in next. The AI Curriculum Planner will also be keeping up with global trends to potentially introduce new courses, what this means is that the AI system will adapt to global educational shifts, for example, if there's a global emphasis on specific topics, institutions using iLearn will have the opportunity to introduce these courses to their academic plans. With these features in mind, regardless of whether you're an educator or student, the ability to have these personalized features will enhance the already existing system to cater to a much wider audience to ensure that anyone, regardless of their challenges, can find an academic pathway that suits them. Integrating the AI Curriculum Planner will move iLearn beyond a platform that just hosts courses, to a dynamic learning environment that is tailored to individual needs and is always evolving based on real-world trends and feedback.

• Functional Database Requirements

1. User:

- 1.1. A user shall create only one account.
- 1.2. A user shall create an account using a unique email address.
- 1.3. A user shall be able to view resources like articles, feedback, and tutorials.
- 1.4. A user is an unregistered user, registered user, or administrator.

2. Registered User:

- 2.1. A registered user shall log into the system from multiple devices.
- 2.2. A registered user shall deactivate or reactivate their account upon choice.
- 2.3. A registered user shall be classified as either a student or an educator.
- 2.4. A registered user shall bookmark courses of interest.
- 2.5. A registered user shall access and download system resources, including articles, tutorials, and other educational content.
- 2.6. A registered user shall view feedback and reviews.
- 2.7. A registered user shall manage their privacy settings.
- 2.8. A registered user shall customize their content preferences.
- 2.9. A registered user shall opt-in or opt-out of receiving system notifications, including course updates, feedback alerts, or system announcements.

3. Administration:

- 3.1. Administrators shall add or remove courses.
- 3.2. Administrators shall monitor system usage metrics.
- 3.3. Administrators shall address flagged feedback in the system.
- 3.4. An admin can change the role of a registered user from student to educator and vice versa.
- 3.5. Administrators shall moderate or manage community content.

4. Student:

- 4.1. A student shall input their academic and work schedules.
- 4.2. A student shall view feedback provided by other users.
- 4.3. A student shall receive course recommendations based on their schedules.
- 4.4. A student shall modify AI-generated curriculum suggestions.
- 4.5. A student shall provide real-time feedback for courses they've taken.
- 4.6. A student shall view performance metrics and course progression.
- 4.7. A student shall join group tasks or presentations initiated by other students.
- 4.8. A student shall view their academic history.
- 4.9. A student shall be able to link their academic organization to the system.
- 4.10. A student shall bookmark courses of interest.
- 4.11. A student shall customize their content preferences.
- 4.12. A student shall opt-in or opt-out of receiving system notifications, including course updates, feedback alerts, or system announcements.

5. Educator:

- 5.1. Educators shall view feedback and recommendations provided by other users.
- 5.2. Educators shall initiate courses and define prerequisites.
- 5.3. Educators shall view collective feedback from students for courses they teach.
- 5.4. Educators shall set meeting hours for individual student consultations.
- 5.5. Educators shall upload course materials.
- 5.6. Educators shall initiate group projects or assignments within their courses.
- 5.7. Educators shall be able to link their academic organization to the system.
- 5.8. Educators shall bookmark courses of interest.
- 5.9. Educators shall opt-in or opt-out of receiving system notifications, including course updates, feedback alerts, or system announcements.

6. Course:

- 6.1. A course shall have prerequisites that must be met before enrollment.
- 6.2. A course shall integrate real-time feedback from students.
- 6.3. A course shall be updated or replaced based on collective student feedback.
- 6.4. A course shall show its relevance to current global educational trends.
- 6.5. Each course shall display its duration, credits, and syllabus.
- 6.6. A course shall highlight professors or educators associated with it.
- 6.7. Courses shall be categorized by department or field of study.

7. AI Algorithm:

- 7.1. The AI shall analyze student input for work and academic schedules.
- 7.2. The AI shall consider past performance metrics when recommending courses.
- 7.3. The AI shall provide curriculum adjustments based on student progress.
- 7.4. The AI shall detect and incorporate global educational trends.
- 7.5. The AI shall use student feedback to assess course quality.

8. Feedback System:

- 8.1. Feedback shall be linked to a specific course or module.
- 8.2. Feedback shall have a timestamp to indicate when it was given.
- 8.3. Feedback shall be anonymous unless the student chooses to reveal their identity.
- 8.4. Feedback shall be upvoted or downvoted by other students for relevance.

9. Global Trend Integration:

- 9.1. The system shall have a feature to detect emerging educational trends.
- 9.2. Trends shall be ranked based on popularity and relevance.
- 9.3. Courses related to top trends shall be highlighted for students.
- 9.4. The system shall provide articles or resources on top global trends.

10. Group Task and Presentation:

- 10.1. A group task shall have a defined list of student participants.
- 10.2. Students shall share resources and communicate within the group task environment.
- 10.3. Group tasks shall allow file uploads for presentations or reports.

11. Lab Booking:

- 11.1. Students shall be able to book specific lab hours.
- 11.2. Lab technicians shall receive notifications about bookings.
- 11.3. The system shall show available slots for lab bookings.
- 11.4. Lab equipment or software needed shall be specified during booking.

12. Performance Review:

- 12.1. Educators shall initiate performance reviews for students.
- 12.2. Reviews shall integrate student performance metrics.
- 12.3. Parents shall access these reviews for a holistic understanding of their child's progress.
- 12.4. Students shall be able to respond or add reflections to their reviews.

13. Notifications:

- 13.1. Educators shall receive notifications about feedback trends on their courses.
- 13.2. Students shall get notifications about changes in course schedules or timings.

14. Community:

- 14.1. Registered users shall create discussion threads related to courses.
- 14.2. Students shall join academic clubs or societies through the system.
- 14.3. Registered users shall upvote valuable discussions or comments.

15. Resources:

- 15.1. A resource is either an article, tutorial, or feedback

16. Account

- 16.1. An account shall belong to only one user
- 16.2. An account shall have at least one profile
- 16.3. An account can have many roles

17. Tutorials

- 17.1. Tutorials shall be viewed by many accounts.

17.2. Tutorials shall be downloaded by many registered users.

18. Articles

18.1. Articles shall be viewed by many accounts.

18.2. Articles shall be downloaded by many registered users.

18.3. Articles shall be either original or an exact copy of themselves.

19. Feedback

19.1. Feedback shall be viewed by many accounts.

19.2. Feedback shall be provided by students.

19.3. Feedback shall be provided by educators.

● Non-functional Database Requirements

1. Performance:

- 1.1 The database system shall support concurrent transactions.
- 1.2 Response times for common queries shall not exceed 2 seconds.
- 1.3 The system shall handle up to 10,000 users simultaneously without any noticeable lag.
- 1.4 Indexing shall be applied to frequently searched columns to improve search performance.

2. Storage:

- 2.1 The database system shall assign 10 MB of memory per table.
- 2.2 The database system shall support persistent storage.
- 2.3 Data redundancy shall be minimized to conserve storage space.
- 2.4 The database system shall support storage partitioning for better data management.

3. Security:

- 3.1 Only encrypted passwords shall be supported by the database system.
- 3.2 All the values inserted into the database shall be consistent with the attribute's datatype and domain.
- 3.3 The database shall be automatically backed up every day at 11:59 pm.
- 3.4 Role-based access control shall be enforced, ensuring that users only have access to data relevant to their roles.
- 3.5 All database transactions shall be logged for audit purposes.
- 3.6 The database system shall have protection against SQL injection attacks.

4. Scalability:

- 4.1 The database shall support scaling out to handle increased load by distributing data across multiple servers.
- 4.2 The system shall add additional hardware resources without downtime.

5. Reliability:

- 5.1 The database system shall have a 99.9% uptime, excluding scheduled maintenance.

5.2 Any failed transaction shall be rolled back to maintain database consistency.

5.3 The system shall have failover capabilities in case the primary server goes down.

6. Usability:

6.1 The database system's interfaces shall be user-friendly and intuitive for end-users and administrators.

6.2 Errors, warnings, and notifications shall be meaningful and help the user understand the issue and its potential solution.

7. Integration:

7.1 The database shall support APIs or connectors for integration with other systems like Learning Management Systems, Financial Systems, etc.

7.2 It shall support file formats like JSON and XML for seamless integration.

8. Maintainability:

8.1 The database system shall support online upgrades without affecting the availability.

8.2 Regular health checks and optimizations, like defragmentation of storage, shall be feasible without system interruption.

9. Compliance:

9.1 The database system shall comply with regulations like GDPR or FERPA regarding student data.

9.2 Regular audits shall be conducted to ensure compliance is maintained.

10. Environmental:

10.1 The database system shall operate effectively under different environmental conditions where the servers are located.

10.2 The system shall optimize energy consumption to reduce its environmental impact.

11. Legal:

11.1 The database system shall adhere to all applicable local, state, federal, and international laws and regulations.

11.2 The system shall facilitate compliance reporting and documentation as required by law.

12. Content:

12.1 The database system shall support multilingual content to cater to a diverse user base.

12.2 The content displayed shall be accurate, up-to-date, and free of errors.

13. Compatibility:

13.1 The database system shall be compatible with various operating systems including Windows, Linux, and MacOS.

13.2 The system shall be compatible with different database management systems to ensure smooth integration.

14. Organizational:

14.1 The database system shall align with the organization's existing workflows and processes.

14.2 The system shall be adaptable to changes in organizational structure and policies



- system_id: key, numeric
- version: alphanumeric
- last_update: timestamp

- club_id: key, numeric
- club_name: alphanumeric
- members_count: numeric

- details_id: key, numeric
- syllabus: text
- objectives: text

- prerequisite_id: key, numeric
- course_id: key, numeric

- requirement: text
- **Department (Strong)**
 - department_id: key, numeric
 - department_name: alphanumeric
 - head: alphanumeric
- **Privacy Settings (Strong)**
 - settings_id: key, numeric
 - visibility: alphanumeric (e.g., public/private)
 - data_sharing: alphanumeric
- **Content Preference (Strong)**
 - preference_id: key, numeric
 - content_type: alphanumeric
 - frequency: alphanumeric
- **AI Suggestions (Strong)**
 - suggestion_id: key, numeric
 - course_id: key, numeric
 - suggestion_reason: text
- **Performance (Strong)**
 - performance_id: key, numeric
 - score: numeric
 - feedback: text
- **Notifications (Strong)**
 - notification_id: key, numeric
 - content: text
 - timestamp: timestamp
- **Academic History (Strong)**
 - history_id: key, numeric
 - user_id: foreign key, numeric
 - courses_taken: multivalue, alphanumeric
 - grades: multivalue, alphanumeric
- **Academic Organization (Strong)**
 - org_id: key, numeric
 - org_name: alphanumeric
 - org_type: alphanumeric (e.g., University, College, Institute)
 - location: alphanumeric
- **AI Algorithm (Strong)**
 - algorithm_id: key, numeric
 - version: alphanumeric
 - last_updated: timestamp
 - function: text (description of what the algorithm does)
- **Curriculum Adjustments (Weak)**
 - adjustment_id: key, numeric
 - student_id: foreign key, numeric

- suggested_changes: text
 - reason: text
- **User (Strong)**
 - user_id: key, numeric
 - email: alphanumeric
 - password: alphanumeric (encrypted preferably)
 - registration_date: timestamp
- **UserType (Strong)**
 - userType_id: key, numeric
 - type_name: alphanumeric
 - description: text
- **Account (Weak)**
 - account_id: key, numeric
 - user_id: foreign key, numeric
 - account_status: alphanumeric (e.g., active, suspended)
 - last_login: timestamp
- **Tutorials (Weak)**
 - tutorial_id: key, numeric
 - title: alphanumeric
 - content: text
 - creation_date: timestamp
- **Articles (Weak)**
 - article_id: key, numeric
 - title: alphanumeric
 - author: alphanumeric
 - publication_date: timestamp
- **Feedback (Weak)**
 - feedback_id: key, numeric
 - user_id: foreign key, numeric
 - content: text
 - submission_date: timestamp
- **Address (Addressing an Issue) (Weak)**
 - address_id: key, numeric
 - feedback_id: foreign key, numeric
 - issue_description: text
 - resolution: text
 - status: alphanumeric (e.g., open, closed, in-progress)
- **Course Material (Strong)**
 - material_id: key, numeric
 - course_id: foreign key, numeric
 - type: alphanumeric (e.g., video, document)
 - link: alphanumeric (URL)
- **Device (Strong)**

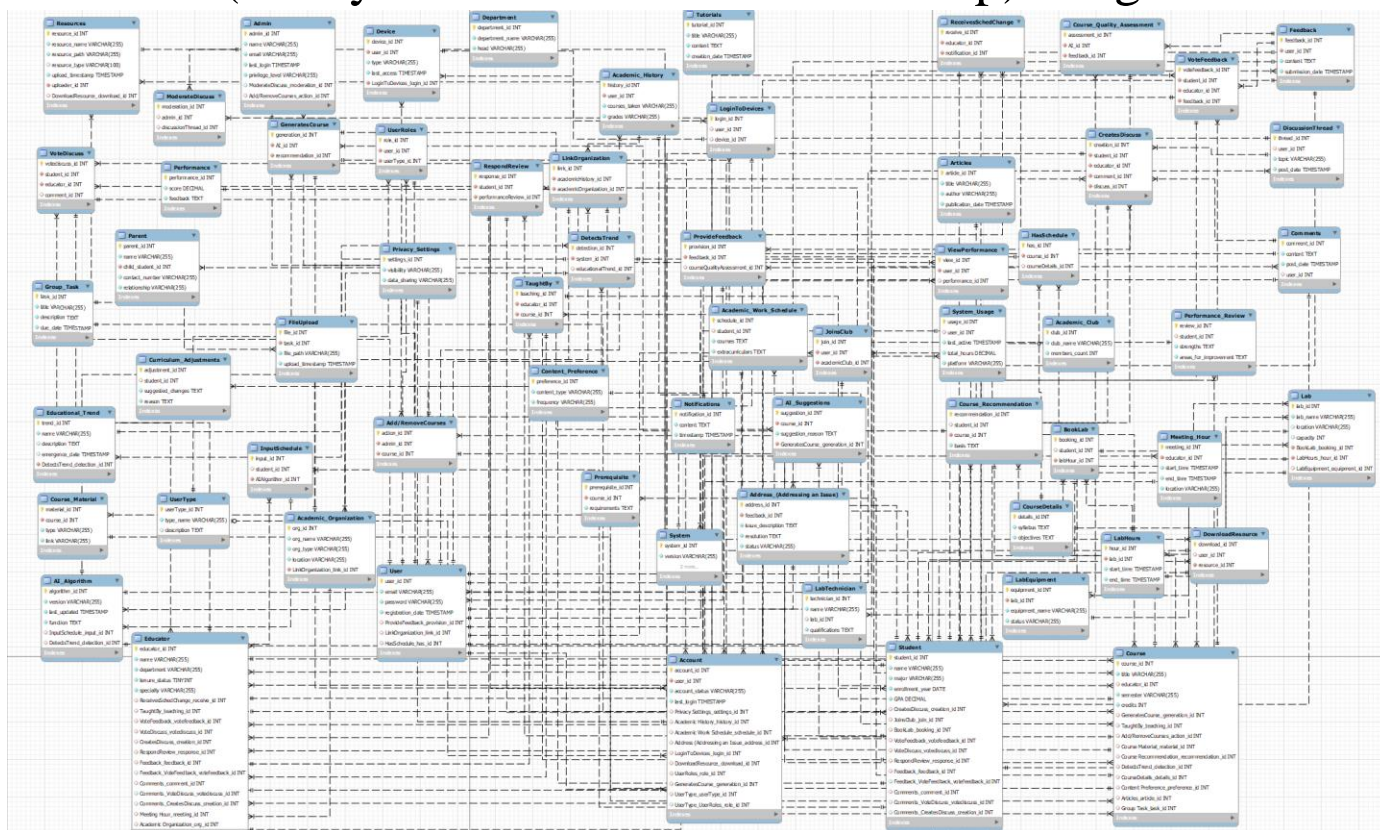
- device_id: key, numeric
- user_id: foreign key, numeric
- type: alphanumeric (e.g., mobile, PC)
- last_access: timestamp
- **Resources (Strong)**
 - resource_id: key, numeric
 - resource_name: alphanumeric
 - resource_path: alphanumeric
 - resource_type: alphanumeric
 - upload_timestamp: timestamp
 - uploader_id: foreign key, numeric
- **Admin (Weak)**
 - admin_id: key, numeric
 - name: alphanumeric
 - email: alphanumeric
 - last_login: timestamp
 - privilege_level: alphanumeric
- **Student (Weak)**
 - student_id: key, numeric
 - name: alphanumeric
 - major: alphanumeric
 - enrollment_year: date
 - GPA: numeric
- **Educator (Weak)**
 - educator_id: key, numeric
 - name: alphanumeric
 - department: alphanumeric
 - tenure_status: boolean
 - speciality: alphanumeric
- **System Usage (Weak)**
 - usage_id: key, numeric
 - user_id: foreign key, numeric
 - last_active: timestamp
 - total_hours: numeric
 - platform: alphanumeric (e.g., web, mobile)
- **Meeting Hour (Weak)**
 - meeting_id: key, numeric
 - educator_id: foreign key, numeric
 - start_time: timestamp
 - end_time: timestamp
 - location: alphanumeric
- **Parent (Strong)**
 - parent_id: key, numeric

- name: alphanumeric
 - child_student_id: foreign key, numeric
 - contact_number: alphanumeric
 - relationship: alphanumeric (e.g., mother, father)
- **Course Recommendation (Weak)**
 - recommendation_id: key, numeric
 - student_id: foreign key, numeric
 - course_id: foreign key, numeric
 - basis: text (e.g., based on interests, previous courses)
- **AcademicWork/Schedule (Weak)**
 - schedule_id: key, numeric
 - student_id: foreign key, numeric
 - courses: text
 - extracurriculars: text
- **Group Task (Strong)**
 - task_id: key, numeric
 - title: alphanumeric
 - description: text
 - due_date: timestamp
- **FileUpload (Weak)**
 - file_id: key, numeric
 - task_id: foreign key, numeric
 - file_path: alphanumeric
 - upload_timestamp: timestamp
- **Performance Review (Weak)**
 - review_id: key, numeric
 - student_id: foreign key, numeric
 - strengths: text
 - areas_for_improvement: text
- **Lab (Strong)**
 - lab_id: key, numeric
 - lab_name: alphanumeric
 - location: alphanumeric
 - capacity: numeric
- **LabTechnician (Strong)**
 - technician_id: key, numeric
 - name: alphanumeric
 - lab_id: foreign key, numeric
 - qualifications: text
- **LabHours (Strong)**
 - hour_id: key, numeric
 - lab_id: foreign key, numeric
 - start_time: timestamp

- end_time: timestamp
- **LabEquipment (Strong)**
 - equipment_id: key, numeric
 - lab_id: foreign key, numeric
 - equipment_name: alphanumeric
 - status: alphanumeric (e.g., functional, broken)
- **DiscussionThread (Weak)**
 - thread_id: key, numeric
 - user_id: foreign key, numeric
 - topic: alphanumeric
 - post_date: timestamp
- **Course (Weak)**
 - course_id: key, numeric
 - title: alphanumeric
 - educator_id: foreign key, numeric
 - semester: alphanumeric
 - credits: numeric
- **Educational Trend (Strong)**
 - trend_id: key, numeric
 - name: alphanumeric
 - description: text
 - emergence_date: timestamp
- **Course Quality Assessment (Weak)**
 - assessment_id: key, numeric
 - AI_id: foreign key, numeric
 - feedback_id: foreign key, numeric
- **HasSchedule (Weak)**
 - has_id: key, numeric
 - course_id: foreign key, numeric
 - courseDetails_id: foreign key, numeric
- **DetectsTrend (Weak)**
 - detection_id: key, numeric
 - system_id: foreign key, numeric
 - educationalTrend_id: foreign key, numeric
- **ProvideFeedback (Weak)**
 - provision_id: key, numeric
 - feedback_id: foreign key, numeric
 - courseQualityAssessment_id: foreign key, numeric
- **ViewPerformance (Weak)**
 - view_id: key, numeric
 - user_id: foreign key, numeric
 - performance_id: foreign key, numeric
- **InputSchedule (Weak)**

- input_id: key, numeric
- student_id: foreign key, numeric
- AIAAlgorithm_id: foreign key, numeric
- **LinkOrganization (Weak)**
 - link_id: key, numeric
 - academicHistory_id: foreign key, numeric
 - academicOrganization_id: foreign key, numeric
- **JoinsClub (Weak)**
 - join_id: key, numeric
 - user_id: foreign key, numeric
 - academicClub_id: foreign key, numeric
- **ReceivesSchedChange (Weak)**
 - receive_id: key, numeric
 - educator_id: foreign key, numeric
 - notification_id: foreign key, numeric
- **BookLab (Weak)**
 - booking_id: key, numeric
 - student_id: foreign key, numeric
 - labHour_id: foreign key, numeric
- **TaughtBy (Weak)**
 - teaching_id: key, numeric
 - educator_id: foreign key, numeric
 - course_id: foreign key, numeric
- **GeneratesCourse (Weak)**
 - generation_id: key, numeric
 - AI_id: foreign key, numeric
 - recommendation_id: foreign key, numeric
- **Comments (Weak)**
 - comment_id: key, numeric
 - content: text
 - post_date: timestamp
 - user_id: foreign key, numeric
- **VoteFeedback (Weak)**
 - vote_id: key, numeric
 - student_id: foreign key, numeric
 - educator_id: foreign key, numeric
 - feedback_id: foreign key, numeric
- **VoteDiscuss (Weak)**
 - vote_id: key, numeric
 - student_id: foreign key, numeric
 - educator_id: foreign key, numeric
 - comment_id: foreign key, numeric
- **ModerateDiscuss (Weak)**

- moderation_id: key, numeric
- admin_id: foreign key, numeric
- discussionThread_id: foreign key, numeric
- **CreatesDiscuss (Weak)**
 - creation_id: key, numeric
 - student_id: foreign key, numeric
 - educator_id: foreign key, numeric
 - comment_id: foreign key, numeric
 - discuss_id: foreign key, numeric
- **LoginToDevices (Weak)**
 - login_id: key, numeric
 - user_id: foreign key, numeric
 - device_id: foreign key, numeric
- **DownloadResource (Weak)**
 - download_id: key, numeric
 - user_id: foreign key, numeric
 - resource_id: foreign key, numeric
- **Add/RemoveCourses (Weak)**
 - action_id: key, numeric
 - admin_id: foreign key, numeric
 - course_id: foreign key, numeric
- **RespondReview (Weak)**
 - response_id: key, numeric
 - student_id: foreign key, numeric
 - performanceReview_id: foreign key, numeric
- **UserRoles (Weak)**
 - role_id: key, numeric
 - user_id: foreign key, numeric
 - userType_id: foreign key, numeric (could be Admin, Student, Educator, etc.)



• Constraints Description

Table	FK	ON DELETE	ON UPDATE	Comment
Resources	Resource_uploader_id	CASCADE	CASCADE	If the user who uploaded the resource is deleted, we want to reflect those changes in the resources table
Resources	Fk_Resources_DownloadResource1	SET NULL	CASCADE	When a resource download entry is deleted, we want to ensure consistency and that's what Cascade does for us
ModerateDiscuss	Moderatediscuss_admin_id	SET NULL	CASCADE	If an admin is removed, the discussion they moderated might still be relevant
ModerateDiscuss	moderatediscuss_discussionThread_id	SET NULL	CASCADE	Same thing here, if the main discussion id changes, it should reflect in the moderation records
Admin	fk_Admin_Add/RemoveCourses1	SET NULL	CASCADE	Admin's actions are linked to courses, and if an action is removed, we might remove the connection, but not necessarily the entire action record
Device	Device_user_id	CASCADE	CASCADE	If a user's id changes or the user is deleted, it's good to keep the device records updated

Device	Fk_Device_LoginToDevices1	CASCADE	CASCADE	This represents a record of devices a user has logged into, if this record changes, all associated device logs should be updated
Academic History	Academichist_user_id	CASCADE	CASCADE	Academic histories are tied to users, if a user's record changes, cascade ensures that the academic history reflects those changes
ReceivesSchedChange	Schedchange_educator_id	CASCADE	CASCADE	Educators receiving schedule changes should have their records updated if their id changes
ReceivesSchedChange	Schedchange_notification_id	CASCADE	CASCADE	Notifications related to schedule changes are vital, if a notification record is removed or changed, it should be reflected in the related entries
Course Quality Assessment	Assess_feedback_id	CASCADE	CASCADE	Quality assessments are linked to feedback and an AI system, changes or deletions should cascade to ensure data consistency
Course Quality Assessment	Assess_AI_id	CASCADE	CASCADE	Same idea as above
VoteFeedback	Votefeedback_student_id	CASCADE	CASCADE	The actions of voting and providing feedback are

				closely related to users, feedback records, and specific topics, any change or deletion should cascade to maintain integrity
VoteFeedback	Votefeedback_educator_id	CASCADE	CASCADE	Same as above
VoteFeedback	Votefeedback_feedback_id	CASCADE	CASCADE	Same as above
Feedback	Feedback_user_id	CASCADE	CASCADE	Same as above
Feedback	Fk_feedback_VoteFeedback1	CASCADE	CASCADE	Same as above
VoteDiscuss	votediscuss_student_id	CASCADE	CASCADE	Voting on discussions is closely tied to student, educators, and specific comments or topics, cascade makes sure that the data remains consistent if there are changes
VoteDiscuss	votediscuss_educator_id	CASCADE	CASCADE	Same as above
VoteDiscuss	votediscuss_comment_id	CASCADE	CASCADE	Same as above
GeneratesCourse	coursegenerate_AI_id	CASCADE	CASCADE	Course generation that is tied to AI recommendations needs to remain consistent, if the AI or recommendation changes, it should reflect in the course generation
GeneratesCourse	coursegenerate_recommendation_id	CASCADE	CASCADE	Same as above
UserRoles	userroles_user_id	CASCADE	CASCADE	Roles assigned to users need to remain consistent, if a user's role or type changes, it should cascade to ensure consistency

UserRoles	userroles_userType_id	CASCADE	CASCADE	Same as above
RespondReview	reviewrespond_student_id	CASCADE	CASCADE	Responses to reviews made by students should cascade to maintain data accuracy and consistency
RespondReview	reviewrespond_performanceReview_id	CASCADE	CASCADE	Same as above
LinkOrganization	linkorg_academicHistory_id	CASCADE	CASCADE	The linking between academic histories and organizations should reflect changes to maintain clarity and consistency
LinkOrganization	linkorg_academicOrganization_id	CASCADE	CASCADE	Same as above
LoginToDevices	logindevice_user_id	SET NULL	CASCADE	If a user's device or login record changes, it's best to nullify the relationship while preserving the primary record
LoginToDevices	logindevice_device_id	SET NULL	CASCADE	Same as above
CreatesDiscuss	creatediscuss_student_id	CASCADE	CASCADE	Discussions and threads are tied to students, educators, and specific topics, cascade ensures that the data remains consistent across any changes or deletions
CreatesDiscuss	creatediscuss_educator_id	CASCADE	CASCADE	Same as above
CreatesDiscuss	creatediscuss_comment_id	CASCADE	CASCADE	Same as above
CreatesDiscuss	creatediscuss_discuss_id	CASCADE	CASCADE	Same as above
DiscussionThread	discussionthread_user_id	SET NULL	CASCADE	Same as above
DiscussionThread	fk_DiscussionThread_CreatesDiscuss1	CASCADE	CASCADE	Same as above

Parent	child_student_id	CASCADE	CASCADE	In the rare case that a student's id changes, it should be reflected upon that parent's link to the student, and a parent entry should not exist without an associated student
DetectsTrend	detect_educationalTrend_id	SET NULL	CASCADE	If the id of the education trend changes, the link in DetectsTrend should also update
DetectsTrend	detect_system_id	CASCADE	CASCADE	If the system ID changes, the link should update, and if it's removed, it makes sense to remove the associated detection record since they might no longer be relevant
ProvideFeedback	providefeedback_feedback_id	CASCADE	CASCADE	If feedback is deleted, its association with course quality assessment might no longer be needed
ProvideFeedback	providefeedback_courseQualityAssessment_id	SET NULL	CASCADE	If an assessment is deleted, we might still want to keep the feedback, but without pointing to a specific assessment
ViewPerformance	performanceview_user_id	CASCADE	CASCADE	This reflect any user ID changes, and if a user is

				removed, their performance view might no longer be relevant
ViewPerformance	performanceview_performance_id	SET NULL	CASCADE	If a performance record is deleted, we might still want to keep the review
HasSchedule	hassched_course_id	CASCADE	CASCADE	If a course is deleted, the associated schedules are no longer relevant
HasSchedule	hassched_courseDetails_id	SET NULL	CASCADE	If specific courses details are removed, we might still want to keep the schedule
Comments	comments_user_id	SET NULL	CASCADE	If a user is deleted, we might still want to keep comments for context in discussions
Comments	fk_Comments_VoteDiscuss1	CASCADE	CASCADE	If a vote or discussion is removed, associated comments might no longer be relevant
Comments	fk_Comments_CreatesDiscuss1	CASCADE	CASCADE	If a discussion record is deleted, its related comments may no longer be meaningful
Curriculum Adjustments	curradjustment_student_id	SET NULL	CASCADE	Curriculum adjustments linked to students should also update if a student's id changes, and if a student is deleted, the

				curriculum adjustments might remain for historical/auditing purposes
FileUpload	fileupload_task_id	CASCADE	CASCADE	If the task id changes, associated file uploads should also update, and if a task is removed, its associated file uploads might not make sense to keep around
TaughtBy	taughtby_educator_id	CASCADE	CASCADE	If an educator is removed, their teaching record would no longer be relevant, and if an educator's id changes, the courses they teach should update accordingly
TaughtBy	taughtby_course_id	CASCADE	CASCADE	If a course id changes, the association with the educators should also update, and if a course is deleted, all educator associations to that course should also be removed
Academic Work Schedule	schedule_student_id	SET NULL	CASCADE	If a student's id changes, their academic schedule should also update, and if a student is removed, the academic schedule might remain for

				historical/auditing purposes
JoinsClub	clubjoin_user_id	CASCADE	CASCADE	If a user's id changes, their club associations should also update, and if a user is removed, their records of club associations should also be removed
JoinsClub	clubjoin_academicClub_id	CASCADE	CASCADE	If a club's id changes, user associations should also update, and if a club is deleted, all user associations to that club should be removed
System Usage	sysusage_user_id	SET NULL	CASCADE	If a user's id changes, the system usage record should also update, and if a user is deleted, the system usage records might remain for analysis or auditing
Performance Review	reviewperformance_student_id	SET NULL	CASCADE	If a student's id changes, their performance reviews should also update, if a student is removed, performance reviews might remain for historical/auditing purposes
Performance Review	fk_Performance Review_ViewPerformance1	CASCADE	CASCADE	If the view record id changes,

				associated performance reviews should also update, and if a view record is removed, any performance review associated might not make sense anymore and should be removed
Performance Review	fk_Performance Review_RespondReview1	CASCADE	CASCADE	If the response record id changes, associated performance reviews should also update, and if a response record is removed, the related performance review might no longer be relevant and should be removed
Educational Trend	fk_Educational Trend_DetectsTrend1	CASCADE	CASCADE	If the detects trend id changes, the education trend record should also update, if it's deleted, the associated educational trend might not be relevant anymore
InputSchedule	inputsched_student_id	SET NULL	CASCADE	If a student's id changes, associated input schedules should update, and if a student is removed, the input schedule may remain but won't be linked to any student

InputSchedule	inputsched_AIAlgorithm_id	CASCADE	CASCADE	If the AI Algorithm's id changes, associated input schedules should also update, and if deleted, related input schedules should also be removed as they might not be valid without the algorithm
Add/RemoveCourses	add_remove_admin_id	CASCADE	CASCADE	If an admin's id changes, their course addition/removal records should also update, and if an admin is removed, the associated records of adding/removing courses would no longer be relevant and should also be removed
Add/RemoveCourses	add_remove_course_id	CASCADE	CASCADE	If a course id changes, the records of it being added/removed should also update, and if a course is deleted, associated addition/removal records by admins should also be removed
AI Suggestions	Alsuggestions_course_id	CASCADE	CASCADE	If a course id changes, associated AI suggestion should also update, and if a course is removed, its AI suggestions might

				not make sense anymore and should be removed
AI Suggestions	fk_AI Suggestions_GeneratesCourse 1	CASCADE	CASCADE	If the course generation id changes, AI suggestions linked to it should also update, and if the course generation record is deleted, associated AI suggestions should also be removed
Course Recommendation	student_id_recommendation	SET NULL	CASCADE	If a student's id changes, their course recommendation should update, if a student is removed, the course recommendations may remain but won't be linked to any student
Course Recommendation	course_id_recommendation	CASCADE	CASCADE	If a course id changes, associated recommendations should update, and if a course is deleted, related student recommendations for that course should also be removed
BookLab	labbook_student_id	SET NULL	CASCADE	If a student's id changes, their lab bookings should also update, and if a student is removed, the lab booking may remain but won't

				reference any specific student
BookLab	labbook_labHour_id	CASCADE	CASCADE	If the lab hour id changes, associated lab bookings should be updated, and if a lab hour is removed, any bookings during that time should also be deleted
Meeting Hour	meethour_educator_id	CASCADE	CASCADE	If an educator's id changes, their meeting hours should also be updated, if an educator is removed, their set meeting hours might not be relevant anymore and should be deleted
Lab	fk_Lab_BookLab1	CASCADE	CASCADE	If a booking id changes, the lab associated with it should also update, and if a lab booking is deleted, the lab record might no longer be valid and should be removed
Lab	fk_Lab_LabHours1	CASCADE	CASCADE	If the lab hours change, the lab records should also update, and if certain lab hours are removed, labs operating during those times might not be valid and should also be removed

Lab	fk_Lab_LabEquipment1	SET NULL	CASCADE	If the equipment id changes, the lab that uses this equipment should also update, and if a piece of equipment is removed, labs that used it can still exist but without a link to that equipment
Course Material	coursematerial_course_id	CASCADE	CASCADE	If course id change, associated course materials should also update, if a course is deleted, its materials might not be relevant anymore
UserType	fk_UserType_UserRoles1	CASCADE	CASCADE	If a user role's id changes, the associated user type should also update, and if a user role is deleted, the user type associated with that role might not be valid anymore
Academic Organization	fk_Academic Organization_LinkOrganization1	CASCADE	CASCADE	If an organization link's id changes, the related academic organization should also update, and if an organization link is removed, the corresponding academic organization might no longer be relevant
User	fk_User_ProvideFeedback1	SET NULL	CASCADE	If the feedback id changes, the

				user's link to that feedback should also update, and if that feedback is deleted, the user's record can remain without pointing to that specific feedback
User	fk_User_LinkOrganization1	SET NULL	CASCADE	If the organization link's id changes, the associated user should also update, if an organization link is deleted, the user can remain without being linked to any specific organization
User	fk_User_HasSchedule1	SET NULL	CASCADE	If a schedule id changes, the associated user record should also update, if a schedule is deleted, the user can still exist but won't have a link to that schedule
Prerequisite	prerequisite_course_id	CASCADE	CASCADE	If a course's id changes, the associated prerequisites should also update to maintain the relation, if a course is deleted, its prerequisites might no longer make sense
System	fk_System_System Usage1	CASCADE	CASCADE	If a system usage id changes, the system associated with that usage should also

				update, if a system usage record is removed, the system linked to it might not be valid anymore
Address (Addressing an Issue)	addressing_feedback_id	CASCADE	CASCADE	If the feedback id changes, associated addressing record should update, and if feedback is deleted, addressing that feedback might not be relevant anymore
LabHours	labhours_lab_id	CASCADE	CASCADE	If a lab's id changes, the associated lab hours should also update, and if a lab is deleted, its operating hours might no longer be valid
DownloadResource	resourcedownload_user_id	SET NULL	CASCADE	If a user's id changes, their resource download records should also update, and if a user is removed, the resource download records can remain
DownloadResource	resourcedownload_resource_id	CASCADE	CASCADE	If a resource's id changes, the download records associated with it should also update, if a resource is deleted, associated download records should also be

				removed as they reference that resource
AI Algorithm	fk_AI Algorithm_InputSchedule1	SET NULL	CASCADE	Update the AI Algorithm in the Input Schedule id changes, maintain the AI Algorithm record without referencing a specific Input Schedule
AI Algorithm	fk_AI Algorithm_DetectsTrend1	SET NULL	CASCADE	Updates the AI Algorithm if the DetectsTrend id changes, maintain AI Algorithm record without referencing a specific trend detection
Educator	fk_Educator_ReceivesSchedule1 fk_Educator-TaughtBy1 fk_Educator_VoteFeedback1 fk_Educator_VoteDiscuss1 fk_Educator_CreatesDiscuss1 fk_Educator_RespondReview1 fk_Educator_Feedback1 fk_Educator_Comments1 fk_Educator_Meeting Hour1 fk_Educator_Academic Organization1	SET NULL	CASCADE	For each foreign key: Update the Educator's association if related table id changes, and maintain the Educator record without a specific association
Account	account_user_id	CASCADE	CASCADE	Update Account if the User id changes, delete the Account if the associated User is deleted
Account	fk_Account_Privacy Settings1 fk_Account_Academic History1 fk_Account_Academic Work Schedule1 fk_Account_Address (Addressing an Issue1 fk_Account_LoginToDevices1	SET NULL	CASCADE	For each foreign key: Update the Account's association if related table id changes, and maintain the Account record

	fk_Account_DownloadResource1 fk_Account_UserRoles1 fk_Account_GeneratesCourse1 fk_Account_UserType1			without a specific association
LabTechnician	labtechnician_lab_id	SET NULL	CASCADE	Update LabTechnician if the Lab id changes, maintain the LabTechnician record without referencing a specific lab
Student	fk_Student_CreatesDiscuss1 fk_Student_JoinsClub1 fk_Student_BookLab1 fk_Student_VoteFeedback1 fk_Student_VoteDiscuss1 fk_Student_RespondReview1 fk_Student_Feedback1 fk_Student_Comments1	SET NULL	CASCADE	For each foreign key: Update the Student's association if related table id changes, and maintain Student record without a specific association
LabEquipment	labequipment_lab_id	CASCADE	CASCADE	Update LabEquipment if the Lab id changes and delete the LabEquipment if the associated Lab is removed
Course	course_educator_id	SET NULL	CASCADE	Update Course if the Educator id changes, and maintain Course record without referencing a specific Educator
Course	fk_Course_GeneratesCourse1 fk_Course-TaughtBy1 fk_Course_Add/RemoveCourses1 fk_Course_Course Material1 fk_Course_Course Recommendation1 fk_Course_DetectsTrend1	SET NULL	CASCADE	For each foreign key: Update Course's association if the related table id changes, and maintain Course record without a

	fk_Course_CourseDetails1 fk_Course_Content Preference1 fk_Course_Articles1 fk_Course_Group Task1			specific association
--	--	--	--	-------------------------