Author: Sujal Manandhar

Course: Database and Web Application Module

Code: DIG5127

Report: ERM Report (Proposal)

Date: 2024/06/05

Student ID: 23189654

Page Count: 9

Word Count: 1563

TABLE OF CONTENTS

Consultancy System

Description

The consultancy system is a comprehensive platform designed to optimise various operations, including proficiency testing, student enrollment, and university management. It offers a robust user management system for admins, and students, enabling efficient communication and service provision. Furthermore, the system helps admins make and handle thorough university profiles. These profiles give important information like rankings, scholarships, and test requirements. By bringing together helpful services, safe logins, and easy communication, the system becomes a key part in making it easier for students to find the right university. It ensures a simple and helpful experience for everyone involved.

Application Requirements

1. Admin and User Management

A. Admin Registration and Management: Admins can register by providing the following details. They are as follows:

   a. first_name (max 50 characters)
   b. last_name (max 50 characters)
   c. email (max 100 characters)
   d. username (max 50 characters)
   e. password (hashed, max 255 characters)
   f. created_at (timestamp)
   g. session_token (max 100 characters)

B. Student(User) Management: Students can register and manage their profiles by providing:

   a. name (max 100 characters)
   b. email (max 100 characters)
   c. phone(max 20 characters)
   d. address (text)
   e. created_at (timestamp)
   f. admin_id (reference to Admin table)

2. Application Management

A. Proficiency Tests: Admins can create,read,update, and delete proficiency tests, which include:

   a. test_name (max 50 characters)
   b. description (text)
   c. difficulty (text)
   d. admin_id (reference to the Admin table)

3. University Management

A. University Information: Admins can manage university details,which include:
   a. name (max 100 characters)
   b. country (max 50 characters)
   c. location (max 50 characters)
   d. ranking (integer)
   e. admin_id (reference to the Admin table)
   f. minimum_band (float)
   g. Scholarship (max 100 characters)

4. Enrollment and Scoring

A. Student Enrollment:Students can enroll in proficiency tests by providing:
   a. enrollment_date (date)
   b. student_id (reference to the Student table)
   c. test_id (reference to the Proficiency_Test table)

B. Scoring System: Scores are managed with details including:
   a. category (max 50 characters)
   b. score (max 50 characters)
   c. student_id (reference to the Student table)
   d. university_id (reference to the University table)

C. Test Scores: Links scores to specific proficiency tests by providing:
   a. test_id (reference to the Proficiency_Test table)
   b. score_id (reference to the Score table)

Entity-Relationship Model (ERM)

1. **Entities:**
   a. Admin
   b. Student
   c. Proficiency_Test
   d. University
   e. Enrollment
   f. Score
   g. Test_Score

2. **Attributes:**

A. Admin:
   a. 'admin_id': The primary key for the admin table. This is an integer that auto-increments to ensure each admin has a unique ID and is never null.

b. ‘first_name’: The first name of the admin. This is a varchar type with a maximum length of 50 characters, and it must not be null, indicating it's a required field.
c. ‘last_name’: The last name of the admin. This is also a varchar type with a maximum length of 50 characters and must not be null.
d. ‘email’: The email address used by admins for logging in. This is a varchar type with unique and not null constraints, ensuring each email is distinct and mandatory for authentication.
e. ‘username’: The admin's username. This is a varchar type with a length of 50 characters and must not be null, making it a required field.
f. ‘password’: The hashed password for admin login. This is a varchar type with a length of 255 characters and must not be null.
g. ‘created_at’ datetime: The timestamp of when the admin account was created. This is a datetime type that records the creation time automatically.
h. ‘session_token’: The session token for the admin's login session. This is a varchar type with a length of 100 characters and must not be null.

B. Student:
a. ‘student_id’: The primary key for the student table. This is an integer that auto-increments to ensure each student has a unique ID and is never null.
b. ‘name’: The full name of the student. This is a varchar type with a maximum length of 100 characters and must not be null, indicating it's a required field.
c. ‘email’: The email address used by students for logging in. This is a varchar type with unique and not null constraints, ensuring each email is distinct and mandatory for authentication.
d. ‘phone’: The student's phone number. This is a varchar type with a length of 20 characters, ensuring it fits within this limit and has unique and not null constraints, making it a required and distinct field.
e. ‘address’: The address of the student. This is a text type, allowing for detailed address information.
f. ‘created_at’: The timestamp of when the student account was created. This is a datetime type that records the creation time automatically.
g. ‘admin_id’: A foreign key referencing the primary key ‘admin_id’ from the Admin table, indicating the admin responsible for the student.

C. Proficiency_Test:
a. ‘test_id’: The primary key for the proficiency test table. This is an integer that auto-increments to ensure each test has a unique ID and is never null.
b. ‘test_name’: The name of the proficiency test. This is a varchar type with a maximum length of 50 characters and must not be null, indicating it's a required field.
c. ‘description’: A detailed description of the proficiency test. This is a text type.
d. ‘difficulty’: The difficulty level of the proficiency test. This is a text type.

    e. ‘admin_id’: A foreign key referencing the primary key ‘admin_id’ from the Admin table, indicating the admin responsible for the test.

D. University:
    a. ‘university_id’: The primary key for the university table. This is an integer that auto-increments to ensure each university has a unique ID and is never null.
    b. ‘name’: The name of the university. This is a varchar type with a maximum length of 100 characters and must not be null, indicating it's a required field.
    c. ‘country’: The country where the university is located. This is a varchar type with a length of 50 characters and must not be null.
    d. ‘location’: The specific location or city of the university. This is a varchar type with a length of 50 characters and must not be null.
    e. ‘ranking’: The ranking of the university. This is an integer type.
    f. ‘admin_id’: A foreign key referencing the primary key ‘admin_id’ from the Admin table, indicating the admin responsible for the university.
    g. ‘minimum_band’: The minimum proficiency test score required for admission. This is a float type.
    h. ‘scholarship’: The scholarship details provided by the university. This is a varchar type with a maximum length of 100 characters.

E. Enrollment:
    a. ‘enrollment_id’: The primary key for the enrollment table. This is an integer that auto-increments to ensure each enrollment has a unique ID and is never null.
    b. ‘student_id’: A foreign key referencing the primary key ‘student_id’ from the Student table, indicating which student is enrolling.
    c. ‘test_id’: A foreign key referencing the primary key ‘test_id’ from the Proficiency_Test table, indicating which test the student is enrolling in.
    d. ‘enrollment_date’: The date of the enrollment. This is a date type.

F. Score:
    a. ‘score_id’: The primary key for the score table. This is an integer that auto-increments to ensure each score has a unique ID and is never null.
    b. ‘category’: The category of the score. This is a varchar type with a maximum length of 50 characters.
    c. ‘score’: The actual score. This is a varchar type with a length of 50 characters.
    d. ‘student_id’: A foreign key referencing the primary key ‘student_id’ from the Student table, indicating which student achieved the score.
    e. ‘university_id’: A foreign key referencing the primary key ‘university_id’ from the University table, indicating the university associated with the score.

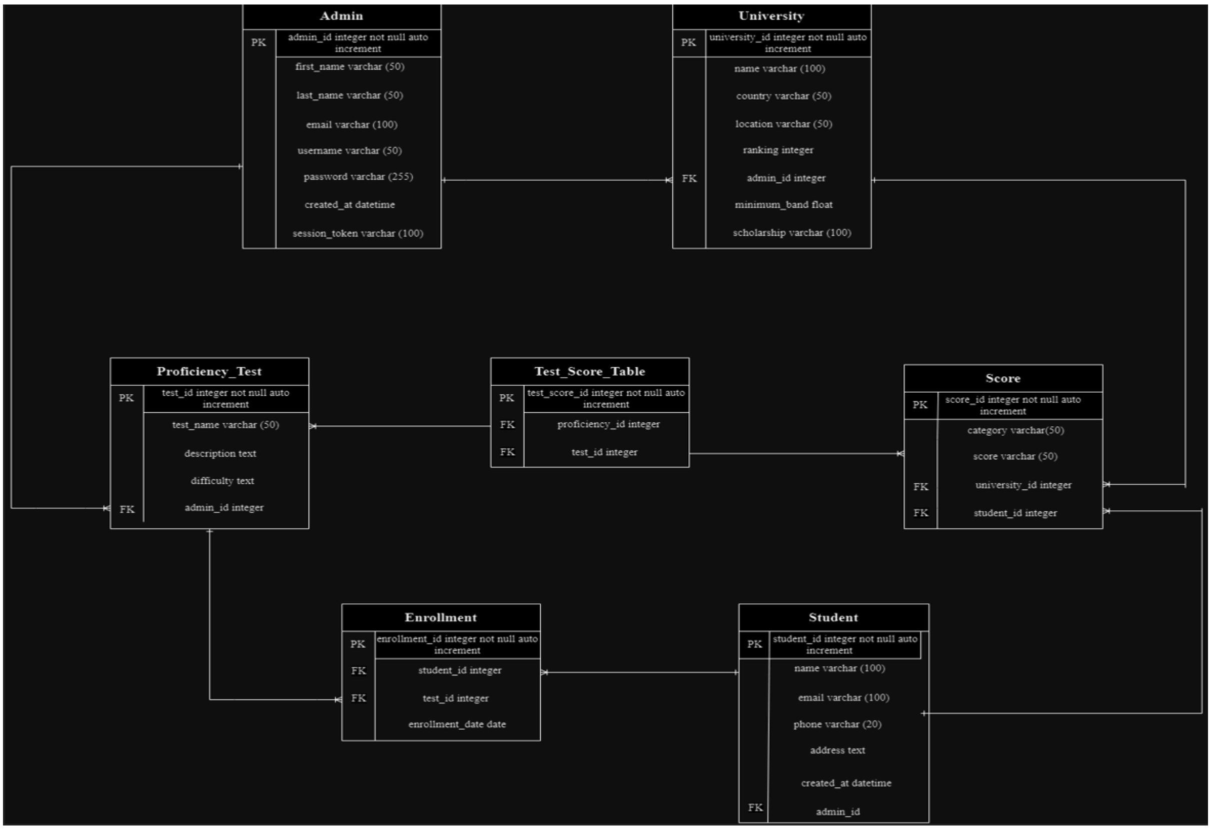
G. Test_Score:

a. ‘test_score_id’: The primary key for the test score table. This is an integer that auto-increments to ensure each test score has a unique ID and is never null.
b. ‘test_id’: A foreign key referencing the primary key ‘test_id’ from the Proficiency_Test table, indicating which test the score is associated with.
c. ‘score_id’: A foreign key referencing the primary key ‘score_id’ from the Score table, indicating which score is associated with the test.

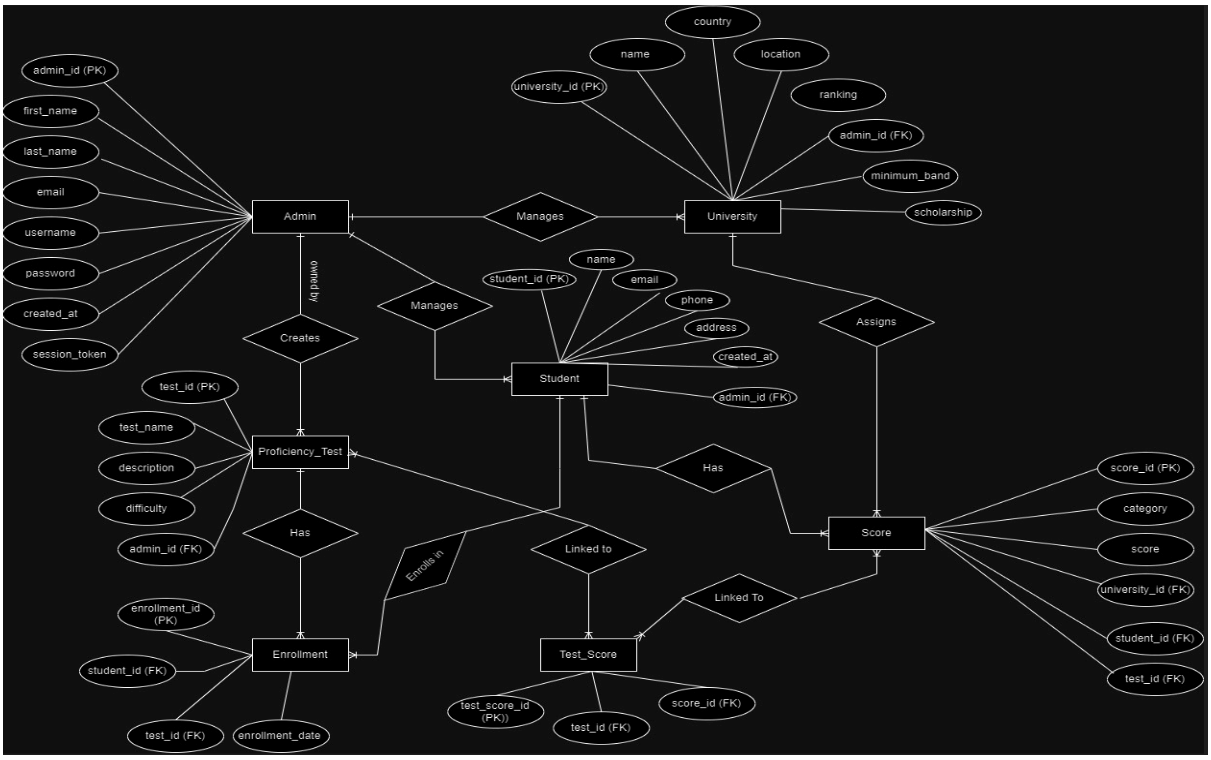
3. Relationships:

A. Admin – Student Relationship: One-to-Many: An admin can oversee multiple students.
B. Admin – University Relationship:One-to-Many: An admin can manage multiple universities.
C. Admin – Proficiency_Test Relationship:One-to-Many: An admin can manage multiple proficiency tests.
D. Student – Enrollment Relationship:One-to-Many: A student can enrol in multiple proficiency tests.
E. Proficiency_Test – Enrollment Relationship:One-to-Many: A proficiency test can have multiple enrollments.
F. Student – Score Relationship:One-to-Many: A student can have multiple scores.
G. University – Score Relationship:One-to-Many: A university can be associated with multiple scores.
H. Proficiency_Test – Test_Score Relationship:One-to-Many: A proficiency test can have multiple test scores.
I. Score – Test_Score Relationship:One-to-Many: A score can be linked to multiple test scores.

Figures:



Fig(1): Crow's foot Notation



Fig(2):Chen Notation

Conclusion

In conclusion, the Consultancy System makes consultancy services much easier and more efficient. It offers user-friendly features for both admins and students, making it simple to manage profiles and handle enrollments and scores. The system organises the management of users, universities, tests, enrollments, and scores in a clear and structured way, ensuring accurate data and smooth operations.The well-defined ERM of the Consultancy System lays out the important relationships and details needed for effective data management and service delivery. As we continue to improve and update the system to fit new needs, it will stay a crucial tool for consultancy services, connecting educational institutions with learners. Our main goal is to link educational services with students, creating a supportive environment that helps them advance in their careers and studies. The Consultancy System shows our dedication to this goal, providing a dependable and effective way to manage consultancy operations.