**Executive Summary:**

The University Management System project aims to revolutionize academic administration by unifying operations, automating tasks, and enhancing the overall student and administrator experience. This database was created to be used in Education as well as EdTech industry for use by the Universities and Tutor Companies in the industry who have always been troubled by the vast and intricate student and management data, when looking at the problem over the whole industry, there is huge amounts of data duplication, and data can be reused by the properties using a relational database. This database considers all-round requirements of the industry and takes care of the data in the most efficient way possible.

The database was modelled considering requirements of data fields from various universities and edtech companies. To delve into the essence of the project, Entity-Relationship (EER) and Unified Modelling Language (UML) diagrams were employed during the conceptualization phase. These diagrams served as the blueprint, capturing the intricate relationships between entities like University, Student, and Employee. The EER and UML diagrams were modeled, followed by the mapping of the conceptual model to a relational model with the required primary and foreign keys. The database model was implemented in MySQL and a prototype in a NoSQL graph database. This hybrid approach allows for optimal data management and scalability. Python was employed to connect to the database, unleashing powerful analytics capabilities demonstrated through a set of well-crafted SQL queries.

The create database is a great success, and by connecting it to Python the analytics capabilities are immense, some of which have been shown in the study. This queries can be very helpful tracking students and in which courses, departments and industries their interests are increasing, which will help universities to plan their future directions. The next step would be to include Visualization tools such as Tableau or Power BI so that the database could be more effectively used to make informed decisions. Furthermore, integrating machine learning algorithms with the database can enhance predictive analytics, allowing universities to anticipate trends in student preferences and tailor their offerings accordingly. By leveraging data science techniques, institutions can identify patterns in student behavior, predict enrollment trends, and optimize resource allocation.

**Introduction of the Project:**

In response to the ever-growing need for streamlined administrative processes in the rapidly evolving educational landscape, the "University Management System" project emerges as a strategic initiative to revolutionize how academic institutions manage their resources, students, and academic programs. The project is rooted in a vision to unify operations, automate tasks, and enhance the overall student experience, aligning with the broader goal of academic excellence. Recognizing the inefficiencies inherent in current university management systems, which rely on manual processes and desperate tools leading to delays, errors, and suboptimal resource utilization, the project aims to provide a cohesive solution.

The University College Management System addresses the pressing issues faced by educational institutions, presenting a comprehensive and integrated system to optimize resource allocation, streamline communication, and elevate competitiveness in the education sector. By automating administrative tasks, enhancing data accuracy, and improving overall efficiency, the project not only seeks to resolve existing challenges but also envisions a future where institutions can redirect valuable time and resources toward academic excellence and strategic initiatives. The journey of a student's enrolment sets the tone for the entire academic experience, beginning with a seamless admission process facilitated by a secure online payment system.

The system's intricate database structure efficiently organizes various components, including courses under different departments, and specialized courses. Beyond academic management, the project incorporates essential elements such as student clubs for cultural and extracurricular activities. This outlines the transformative vision and key components of the University College Management System, emphasizing its role in reshaping the educational landscape, enhancing efficiency, and positioning the institution as a leader in education management.

**II. Conceptual Data Modelling**

1. **EER-Model:**

A diagram of a flowchart

Description automatically generated

1. **UML**:

A diagram of a computer

Description automatically generated

**III. Mapping Conceptual Model to Relational Model**

Primary Key- Underlined

Foreign Key- Italicized

**Relational Model:**

University(**University\_ID**, Name, ContactNumber, Address, Email)

Library(**Library\_ID**, Name)

Hostel(**Hostel\_ID**, Hostel\_Name)

Department(**Dep\_ID**, DName, D\_Head)

Employee(**SSN**, Name, *Dep\_ID*)

Dep\_ID is a Foreign Key representing Dep\_ID in table Department; Not Null

Courses(**Course\_ID**, CourseName, Credit, CourseCode, *Dep\_ID*)

Dep\_ID is a Foreign Key representing Dep\_ID in table Department; Not Null

Student (**Student\_ID,** Name, DOB, Email, Address, Fees\_Amount, Fees\_Status, *Dep\_ID, Hostel\_ID*)

Dep\_ID is a Foreign Key representing Dep\_ID in table Department; Not Null

Hostel\_ID is a Foreign Key representing Hostel\_ID in table Hostel; Not Null

Enrollment(**Enrollment\_ID,** Grade, *Student\_ID, Course\_ID*)

Student\_ID is a Foreign Key representing Student\_ID in table Student; Not Null

Course\_ID is a Foreign Key representing Course\_ID in table Course; Not Null

StudentClub(**Club\_ID**, ClubName, *Dep\_ID*)

Dep\_ID is a Foreign Key representing Dep\_ID in table Department; Not Null

ClubVolunteer(**Volunteer\_ID**, *Student\_ID, Club\_ID*)

Student\_ID is a Foreign Key representing Student\_ID in table Student; Not Null

Club\_ID is a Foreign Key representing Club\_ID in table StudentClub; Not Null

**IV. Implementation of Relation Model via MySQL and NoSQL**

**MySQL Implementation:**

The database was created in MySQL and the following queries were performed:

**#1-Retrieve all students' names and their department IDs.**

SELECT Student\_Name, Department\_ID FROM Student;

A screenshot of a computer

Description automatically generated

**#2-Retrieve total students from each department**

SELECT Department\_ID, COUNT(\*) AS TotalStudents

FROM student

GROUP BY Department\_ID;

A screenshot of a computer

Description automatically generated

**#3 - Query to find students course and grades**

SELECT

student.Student\_ID,

student.Student\_Name,

course.Course\_ID,

course.Coursename,

enrollment.Grade

FROM

student

INNER JOIN enrollment ON student.Student\_ID = enrollment.Student\_ID

INNER JOIN course ON enrollment.Course\_ID = course.Course\_ID;

A screenshot of a computer

Description automatically generated

**#4 - Nested Query Write a query for list of student names who are in hostel richards hostel and who are in department 4**

SELECT Student\_Name

FROM Student

WHERE Hostel\_ID = (

SELECT Hostel\_ID

FROM universityhostel

WHERE Hostel\_ID = 1

)

AND Department\_ID = 4;

A screenshot of a computer

Description automatically generated

**#5 - Correlated query to find department with avg credit more then 3**

SELECT Department\_ID, Department\_Name

FROM department d

WHERE (

SELECT AVG(Credit)

FROM course c

WHERE c.Department\_ID = d.Department\_ID

) > 3;

A screenshot of a computer

Description automatically generated

**#6 – uses the EXISTS clause to filter students based on the specified conditions.**

SELECT s.Student\_ID, s.Student\_Name, s.Fees\_status, e.Grade

FROM student s

JOIN enrollment e ON s.Student\_ID = e.Student\_ID

WHERE EXISTS (

SELECT 1

FROM enrollment

WHERE Student\_ID = s.Student\_ID

AND Grade = 'F'

)

AND s.Fees\_status = 'Pending';

A screenshot of a computer

Description automatically generated

**#7 - Total Number of volunteers in each club using Select From**

SELECT

sc.StudentClub\_ID,

sc.StudentClub\_Name,

(

SELECT COUNT(\*)

FROM clubvolunteer

WHERE StudentClub\_ID = sc.StudentClub\_ID

) AS Total\_Volunteers

FROM

studentclub sc;

A screenshot of a computer program

Description automatically generated

**#8- Union query for coding club and It innovators to generate students list in respective clubs.**

**-- Select students in Coding Club**

SELECT s.Student\_ID, s.Student\_Name

FROM clubvolunteer cv

JOIN studentclub sc ON cv.StudentClub\_ID = sc.StudentClub\_ID

JOIN student s ON cv.Student\_ID = s.Student\_ID

WHERE sc.StudentClub\_Name = 'Coding Club'

UNION

**-- Select students in IT Innovators**

SELECT s.Student\_ID, s.Student\_Name

FROM clubvolunteer cv

JOIN studentclub sc ON cv.StudentClub\_ID = sc.StudentClub\_ID

JOIN student s ON cv.Student\_ID = s.Student\_ID

WHERE sc.StudentClub\_Name = 'IT Innovators';

A screenshot of a computer

Description automatically generated

**V. NOSQL IMPLEMENTATION:**

1. **Calculating the number of students in each department:**

pipeline\_count\_students = [

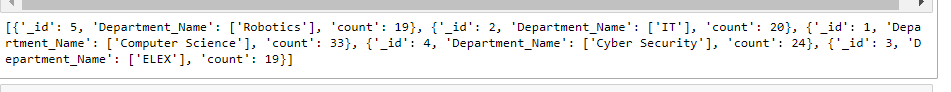
{"$lookup": {"from": "Department", "localField": "Department\_ID", "foreignField": "Department\_ID", "as": "department\_info"}},

{"$group": {"\_id": "$Department\_ID", "Department\_Name": {"$first": "$department\_info.Department\_Name"}, "count": {"$sum": 1}}}

]

result\_count\_students = list(db.Student.aggregate(pipeline\_count\_students))

print(result\_count\_students)



1. **Calculate the average fees paid by students in each department.**

result = db.Fees.aggregate([

{"$group": {"\_id": "$Couse\_name", "average\_fees": {"$avg": "$Fees\_Paid"}}}

])

for avg\_fees in result:

print(avg\_fees)

A screenshot of a computer code

Description automatically generated

1. **Count total fees status for each department, categorizing by 'Pending' and 'Complete' statuses - Aggergate Query**

pipeline\_total\_fees\_status = [

{

'$group': {

'\_id': {

'Department\_ID': '$Department\_ID',

'Fees\_Status': '$Fees\_Status'

},

'total\_count': {'$sum': 1}

}

},

{

'$group': {

'\_id': '$\_id.Department\_ID',

'fees\_status\_counts': {

'$push': {

'Fees\_Status': '$\_id.Fees\_Status',

'total\_count': '$total\_count'

}

}

}

},

{

'$project': {

'Department\_ID': '$\_id',

'fees\_status\_counts': 1,

'\_id': 0

}

}

]

# Assuming you've migrated 'Fees' to 'Student' collection, update the collection name

result\_total\_fees\_status = list(db.Student.aggregate(pipeline\_total\_fees\_status))

print(result\_total\_fees\_status)

A screen shot of a computer code

Description automatically generated

1. **Counts the number of students in each hostel, including those not assigned to a hostel (null value)**

result = db.Student.aggregate([

{"$group": {"\_id": "$Hostel\_ID", "student\_count": {"$sum": 1}}},

{"$lookup": {"from": "Hostel", "localField": "\_id", "foreignField": "Hostel\_ID", "as": "hostel\_info"}},

{"$project": {"hostel\_name": "$hostel\_info.Hostel\_Name", "student\_count": 1, "\_id": 0}}

])

for student\_count in result:

print(student\_count)

A white background with black text

Description automatically generated

**VI. DATABASE ACCESS VIA PYTHON:**

Python facilitates access to the database, and the resulting visual representation of analyzed data is presented here. The integration of MySQL with Python is achieved using the `mysql.connector` library. This process involves executing queries and retrieving data using the `cursor.execute` method. The extracted information is then transformed from a list into a well-organized dataframe through the application of the `pandas` library. To illustrate the analytical findings, graphical plots are generated using the `matplotlib` library.

**Graph 1: The scatter plot shows fees paid by each student and its frequency**

A graph and bar graph

Description automatically generated

**Graph 2: Distribution of students across clubs.**

A pie chart with numbers and text

Description automatically generated

A graph of a bar graph

Description automatically generated

**Graph 3: Total Number of Students in Each Department:**

A graph of blue rectangular objects

Description automatically generated with medium confidence

**Graph 4: Distribution of students in housing category**

A graph of a number of students

Description automatically generated

**Graph 5: Distribution of Grades**

A graph of a distribution of grades

Description automatically generated

**VII. Summary and Recommendation**