



## ITI Examination System

**Prepared By**

Alaa Mahmoud qoura  
Salma magdy ELhadidy  
Ahmed Momtaz elgendy  
Hend Youssef Eleraky  
Fady Gamal AbdElkaream

**Supervised By**

Eng. Nada ELaraby

**INFORMATION TECHNOLOGY INSTITUTE 3-MONTH  
GRADUATION PROJECT**

## Table of Contents

Table of Contents .....	Error! Bookmark not defined.
1. Project Overview.....	3
2. Database Design .....	4
3. Data Sources.....	9
4. Stored Procedures .....	17
5. Data Warehouse Design .....	23
6. ETL Process in DWH .....	25
7. SSRS Reports.....	26
1. Report that returns the students information according to Department No parameter (each Department in one page) .....	26
2. Report that takes the student ID and returns the grades of the student in all courses.	28
3. Report that takes the instructor ID and returns the name of the courses that he teaches and the number of students per course. ....	29
4. Report that takes course ID and returns its topics .....	30
5. Report that takes exam number and returns the Questions in it and choices [freeform report].....	31
6. Report that takes exam number and the student ID then returns the Questions in this exam with the student answers and return exam grade .....	32
8. Power BI Dashboards.....	33
.....	48
9. Challenges & Solutions .....	50
10. Application.....	51
Libraries and Tools Used .....	51

## 1. Project Overview

This project presents a comprehensive data analysis for the Information Technology Institute (ITI) training system, which includes 9 branches (Smart Village , Alexandria, Assiut, Mansoura ,Ismailia , Luxor , Aswan , Beni Suef , Port Said) across different locations. The data involves students from various countries, allowing for multicultural and comparative analysis across the training network.

## **Project Objective :**

This project aims to analyze key aspects of the training structure at the Information Technology Institute, focusing on: The examination structure and performance evaluation of each student based on their

- 1-exam score, which represents 60% of the assessment,  
their graduation project score, which represents 30% of the assessment,  
and their absence score, which represents 10% of the final assessment.
  - 2-The distribution of students and their demographics in each branch.
  - 3-The distribution of trainers across branches, the experience of each trainer, and their  
impact on student performance.
  - 4- The coverage of training topics for each subject and the student's evaluation of them.

## Main Features

Each instructor teaches only one subject.

A teacher may teach the same subject in multiple branches.

Each branch is assigned one teacher per subject.

The project focuses heavily on:

The examination system: performance, structure, and results. Subject coverage: The quality of the subjects distributed and taught across the branches.

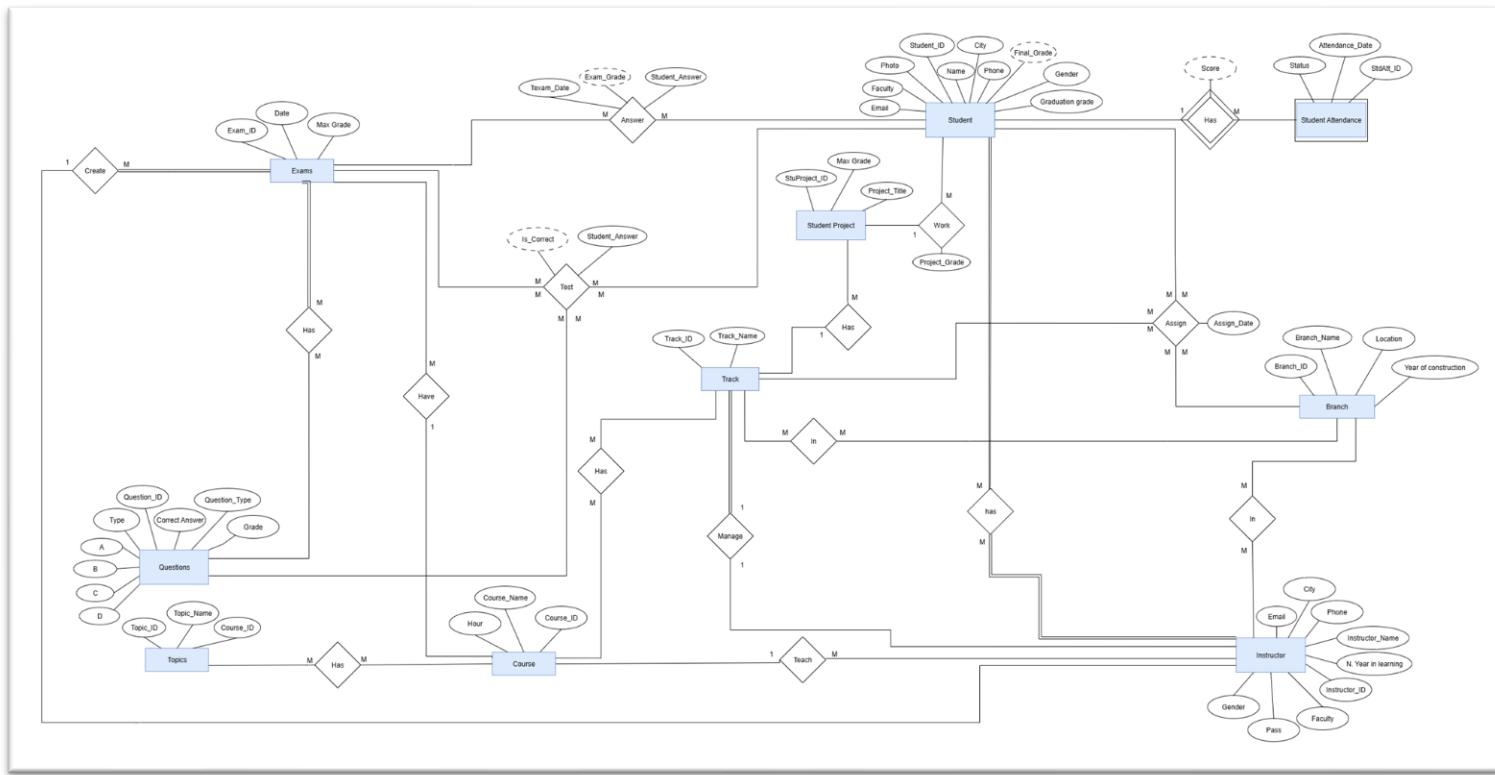
## 2. Database Design

### Relational Database Structure :

The project's relational database was designed to store and manage all core entities in a normalized form. It includes several key tables that reflect real-world training operations at ITI.

### ERD (Entity Relationship Diagram)

This ERD provides a visual representation of the database schema, outlining key entities, their attributes, and the relationships between them. It helps in understanding how data flows and interacts within the system.



## **Main entity :**

Here is an overview of the main tables used: Table Name Description

**Students:** Stores personal information, GPA and nationality for each student.

**Instructors :**Stores instructor data, including subject taught and assigned branches.

**Branches :**Contains all 9 ITI branches with location-related metadata.

**courses :**Defines the subjects offered, linked to instructors and topics.

**Topics :**Sub-divisions of each subject, used to track coverage and analysis.

**Questions :**Contains the questions generated by Python, linked to topics and difficulty.

**Exams :**Represents exam events tied to subjects, branches, and dates

**Student Project :**Project for each course , project title and max grade

**Track :**tracks in each branch

**Student Attendance :** Attendance for each student in each date with status and score

## **Relationships :**

### **1- Instructor — Creates — Exam:**

Explanation: One instructor can create many exams, but each exam is created by one instructor only.

### **2-Exam — Has — Questions:**

Explanation: An exam can contain many questions, and each question can appear in many different exams.

### **3. Topics — Belong To — Course**

Explanation: Each topic is part of one course, but a course can include many topics.

### **4. Instructor — Teaches — Course**

Explanation: Instructors can teach multiple courses, and each course can be taught by multiple instructors.

### **5. Instructor — Works In — Branch**

Explanation: An instructor can work in multiple branches, and each branch can have many instructors.

### **6. Instructor — Manages — Track**

Explanation: An instructor can manage multiple tracks, but each track has one manager.

**7. Track — Has — Student Project**

Explanation: A track can have many student projects, and each project belongs to one specific track.

**8. Track — Includes — Courses**

Explanation: A track contains multiple courses, but each course belongs to one track only.

**9. Student — Track — Branch**

Explanation: Students can be assigned to multiple branches (with assignment dates) , one track , and each branch can have many students.

**10. Student — Has — Attendance**

Explanation: Each student can have multiple attendance records; each attendance record belongs to one student.

**11. Exam - Question – Student**

Explanation: This relationship tracks the answers students gave to specific questions in a specific exam and To know if a student answered a question correctly in a given exam, we need all three entities.

**12. instructor- Student**

Explanation: This relationship tracks the answers students gave to specific each student has multiple instructor and each instructor has multiple student

**13. Student — Has — Student Projects**

Explanation: Each student can work on multiple projects, but each student project is linked to only one student.

## Data Mapping

This section explains the data modelling methodology used in the ITI project. It covers the logical and physical design of the database, including **normalization**, entity definitions, and relationship structuring based on the project requirements. making it easier to perform transformations later during the ETL phase .

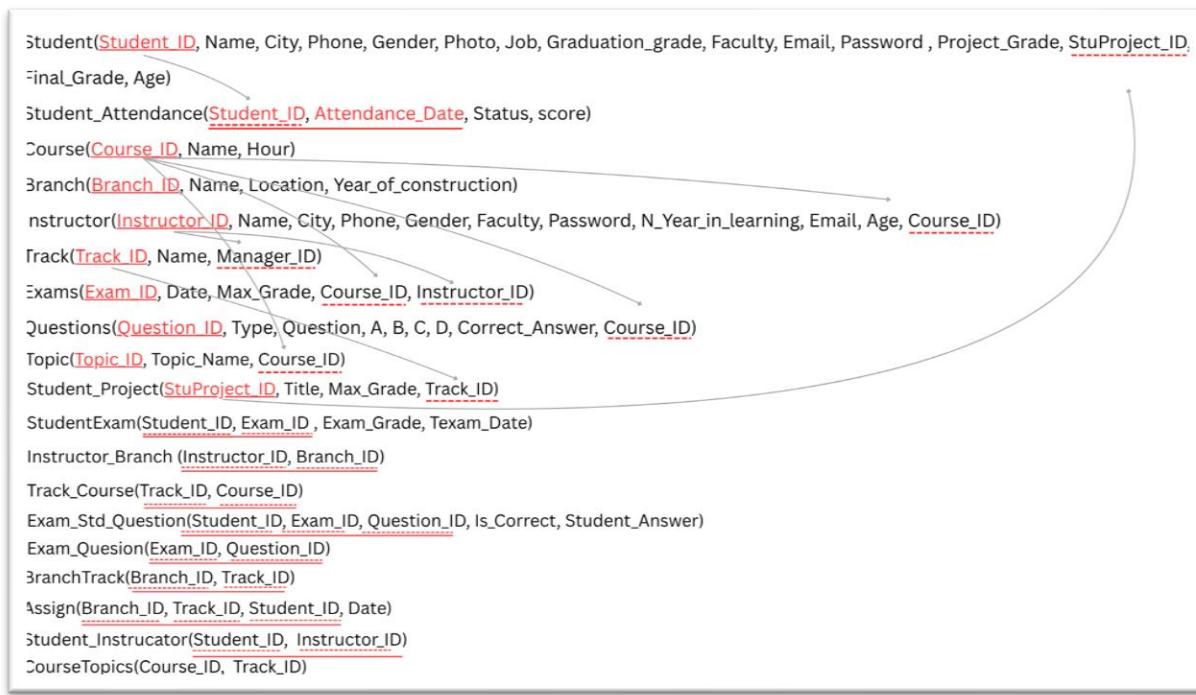


Figure 1 : Mapping

## Database Diagram

Figure 2: ITI System Database Diagram

This image shows the real structure of tables and their connections in the ITI database system

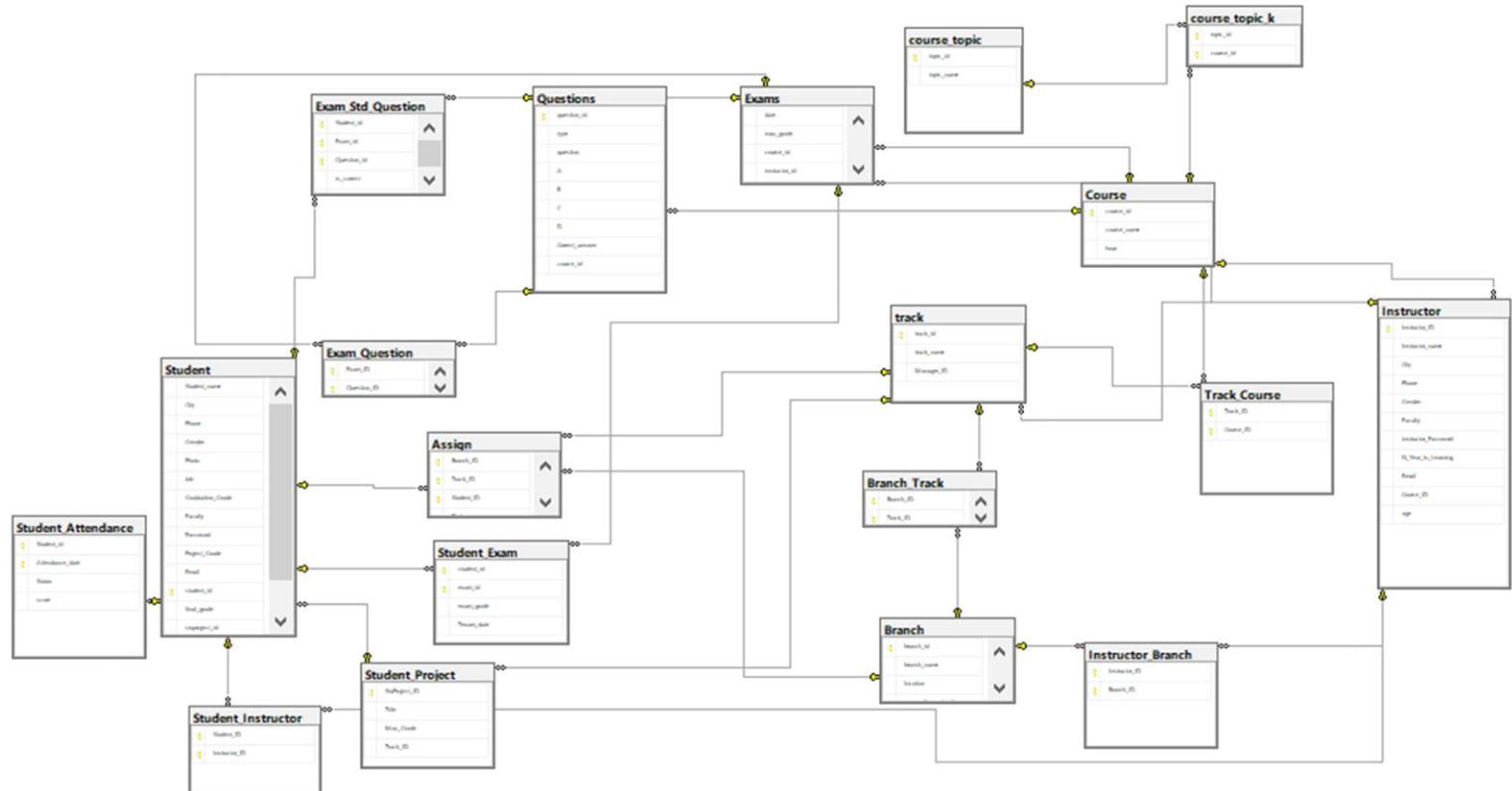


Figure 2 : DATABASE DIAGRAM

### 3. Data Sources

The dataset used in this project was created after the structure was initially prepared using ChatGPT to simulate a realistic training environment for an IT institute. After generating the raw data, it was manually reviewed, cleaned, and optimized to align with the project's structure and analytical needs.

#### Data Preparation Steps

1 -Initial Generation:

Raw data for the student, instructor, branch, subject, and exam were generated using ChatGPT commands.

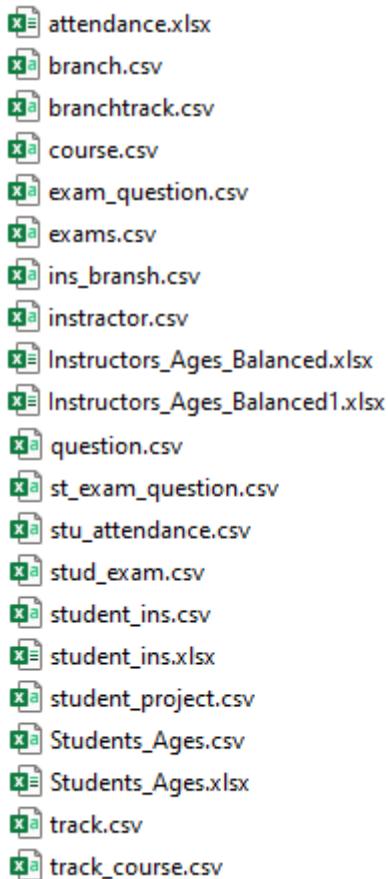


Figure 3 : Tables

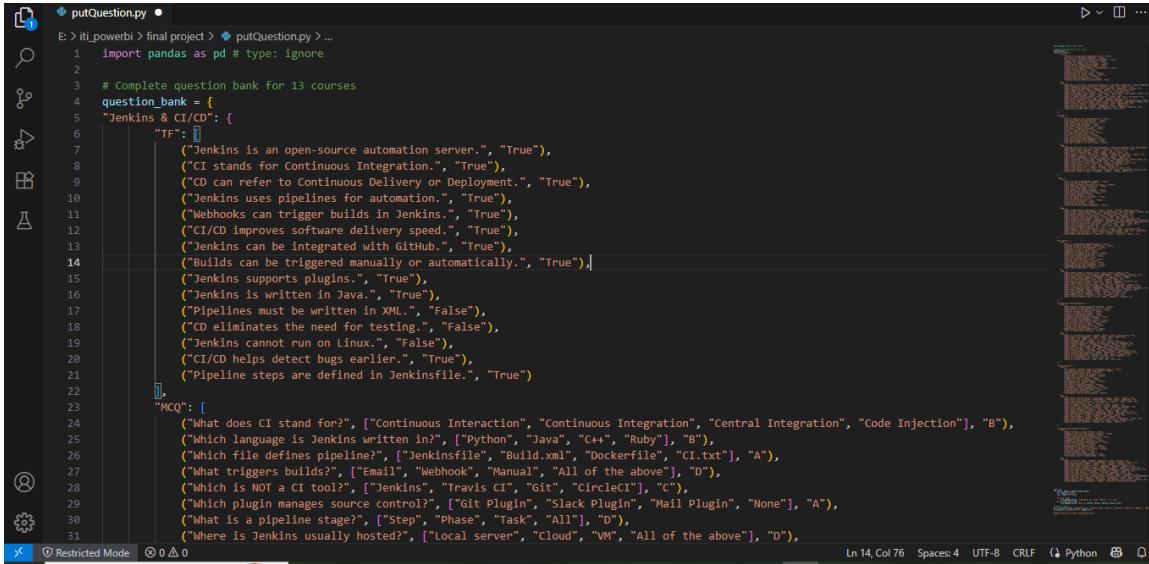
	A	B	C	D	E
	Student_ID	Exam_ID	Question	Student_answer	Is_Correct
1	1	22	3951	A	
2	1	22	3956	B	
3	1	22	3950	A	
4	1	22	3953	A	
5	1	22	3958	C	
6	1	22	3939	D	
7	1	22	3931	C	
8	1	22	3954	D	
9	71	22	3951	C	
0	71	22	3956	C	
1	71	22	3950	A	
2	71	22	3953	C	
3	71	22	3958	A	
4	71	22	3939	B	
5	71	22	3931	D	
6	71	22	3954	C	
7	71	22	3951	B	
8	141	22	3956	D	
9	141	22	3950	A	
0	141	22	3953	D	
1	141	22	3958	A	
2	141	22	3939	B	
3	141	22	3931	D	
4	141	22	3954	C	
5	141	22	3951	B	
6	4	22	3956	B	
7	4	22	3950	D	
8	4	22	3953	C	
9	4	22	3958	D	
0	4	22	3939	A	
1	4	22	3931	D	
2	4	22	3954	C	
3	4	22	3951	B	
4	74	22	3956	A	
5	74	22	3950	D	
6	74	22	3953	C	
7	74	22	3958	D	
8	74	22	3939	B	
9	74	22	3931	A	
0	74	22	3954	D	

st\_exam\_question

Figure 4 : SAMPLE of data

Note: is correct column calculated in SQL

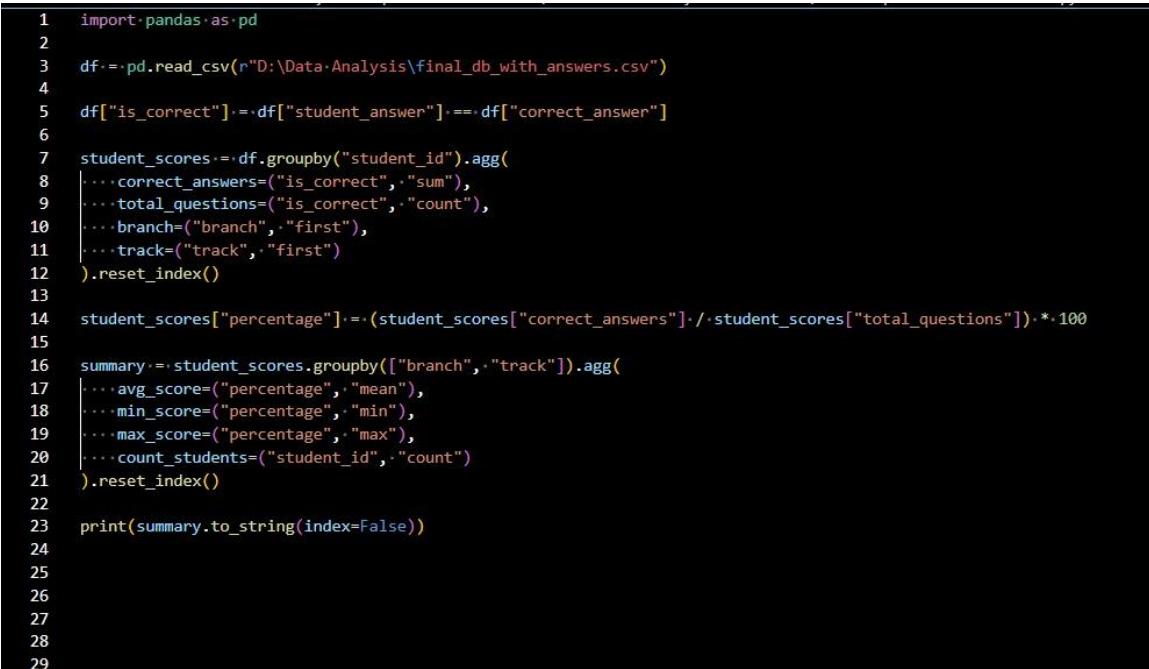
## 2 - Using Python code to generate question data for each course (30 questions per course).



```
E:\itl.powerbi>final project > putQuestion.py ...
1 import pandas as pd # type: ignore
2
3 # Complete question bank for 13 courses
4 question_bank = [
5     "Jenkins & CI/CD": {
6         "TF": [
7             ("Jenkins is an open-source automation server.", "True"),
8             ("CI stands for Continuous Integration.", "True"),
9             ("CD can refer to Continuous Delivery or Deployment.", "True"),
10            ("Jenkins uses pipelines for automation.", "True"),
11            ("Webhooks can trigger builds in Jenkins.", "True"),
12            ("CI/CD improves software delivery speed.", "True"),
13            ("Jenkins can be integrated with GitHub.", "True"),
14            ("Builds can be triggered manually or automatically.", "True"),
15            ("Jenkins supports plugins.", "True"),
16            ("Jenkins is written in Java.", "True"),
17            ("Pipelines must be written in XML.", "False"),
18            ("CD eliminates the need for testing.", "False"),
19            ("Jenkins cannot run on Linux.", "False"),
20            ("CI/CD helps detect bugs earlier.", "True"),
21            ("Pipeline steps are defined in Jenkinsfile.", "True")
22        ],
23        "MCQ": [
24            ("What does CI stand for?", ["Continuous Interaction", "Continuous Integration", "Central Integration", "Code Injection"], "B"),
25            ("Which language is Jenkins written in?", ["Python", "Java", "C++", "Ruby"], "B"),
26            ("Which file defines pipeline?", ["Jenkinsfile", "Build.xml", "Dockerfile", "CI.txt"], "A"),
27            ("What triggers builds?", ["Email", "Webhook", "Manual", "All of the above"], "D"),
28            ("Which is NOT a CI tool?", ["Jenkins", "Travis CI", "Git", "CircleCI"], "C"),
29            ("Which plugin manages source control?", ["Git Plugin", "Slack Plugin", "Mail Plugin", "None"], "A"),
30            ("What is a pipeline stage?", ["Step", "Phase", "Task", "All"], "D"),
31            ("Where is Jenkins usually hosted?", ["Local server", "Cloud", "VM", "All of the above"], "D"),
32        ]
33    }
34]
```

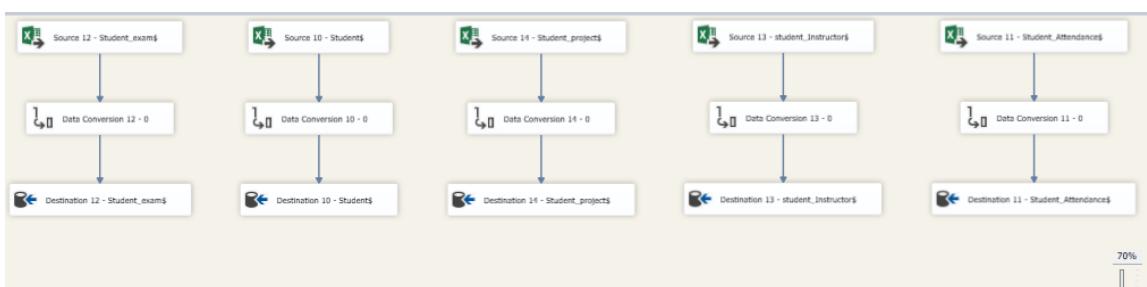
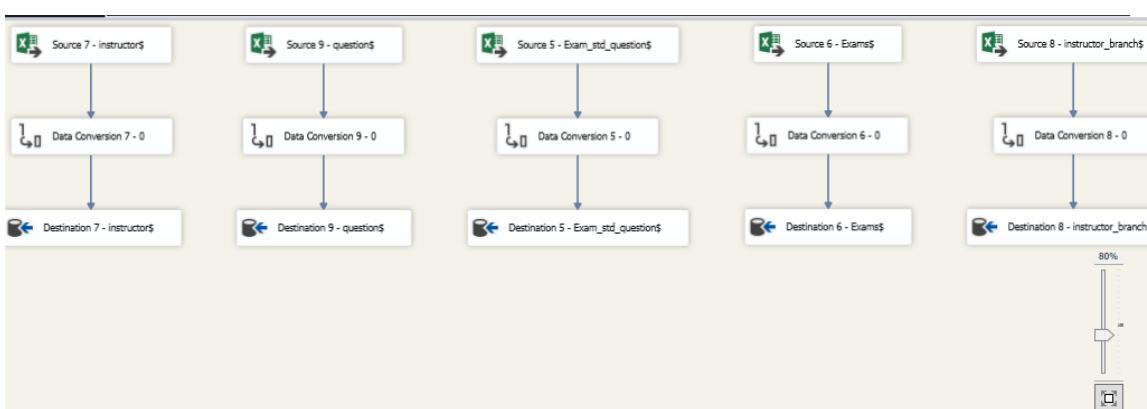
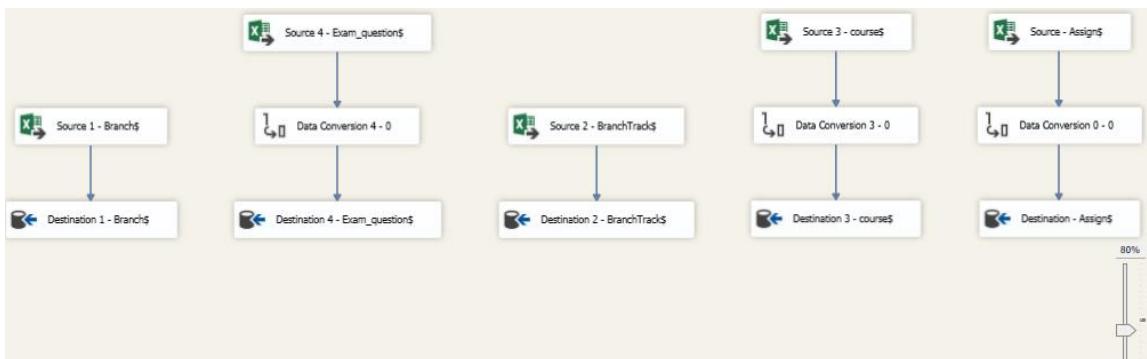
Figure 5 : insert Questions

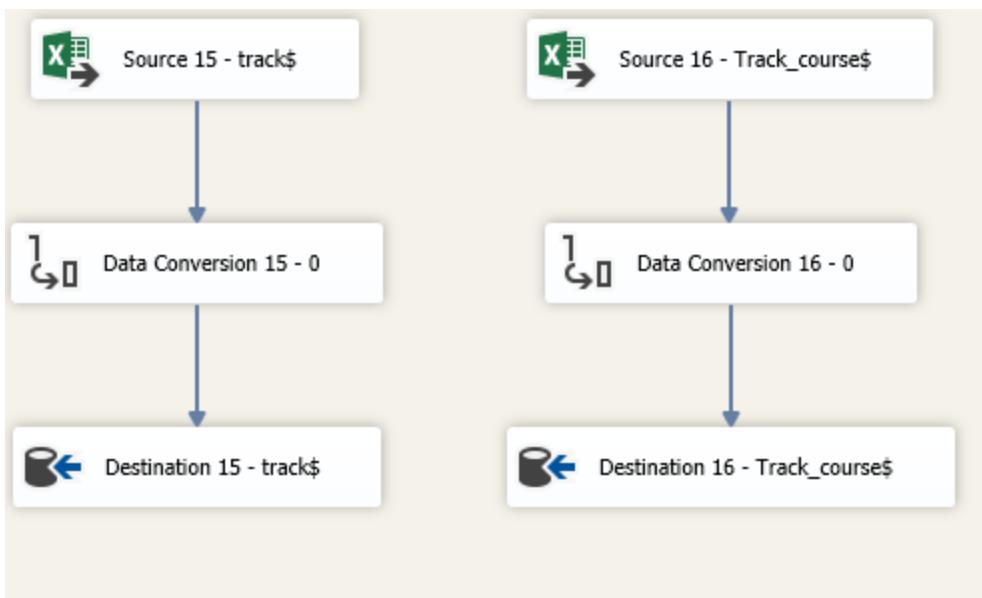
## 3 - Generating random student answers for each exam within logical ranges.



```
1 import pandas as pd
2
3 df = pd.read_csv(r"D:\Data\Analysis\final_db_with_answers.csv")
4
5 df["is_correct"] = df["student_answer"] == df["correct_answer"]
6
7 student_scores = df.groupby("student_id").agg(
8     correct_answers=("is_correct", "sum"),
9     total_questions=("is_correct", "count"),
10    branch=("branch", "first"),
11    track=("track", "first")
12 ).reset_index()
13
14 student_scores["percentage"] = (student_scores["correct_answers"] / student_scores["total_questions"]) * 100
15
16 summary = student_scores.groupby(["branch", "track"]).agg(
17     avg_score=("percentage", "mean"),
18     min_score=("percentage", "min"),
19     max_score=("percentage", "max"),
20     count_students=("student_id", "count")
21 ).reset_index()
22
23 print(summary.to_string(index=False))
24
25
26
27
28
29
```

#### 4 - Insert data From Excel sheets into Database (using SSIS)





## 5 - Comparing student answers with correct answers to fill a column.

```
UPDATE esq
SET esq.is_correct = CASE
    WHEN esq.student_answer = q.Correct_answer THEN 1
    ELSE 0
END
from exam_std_question esq
JOIN Questions q ON esq.question_id = q.question_id
```

The screenshot shows a SQL query being run in a database interface. The query updates the 'is\_correct' column in the 'exam\_std\_question' table ('esq') based on whether the student's answer matches the correct answer ('q.Correct\_answer'). The 'is\_correct' column is set to 1 if they match, and 0 if they don't. It joins the 'exam\_std\_question' table with the 'Questions' table ('q') on the 'question\_id' column. The results are displayed in a table with columns: Student\_id, Exam\_id, Question\_id, is\_correct, and student\_answer. The table has 503,875 rows. A message at the bottom indicates the query was executed successfully.

Student_id	Exam_id	Question_id	is_correct	student_answer
1	13	3960	1	A
1	13	3971	1	B
1	13	3972	1	A
1	13	3974	1	B
1	13	3978	1	A
1	13	3984	1	A
1	13	3987	0	C
1	22	3931	0	B
1	22	3939	1	A
1	22	3950	1	A
1	22	3951	1	B

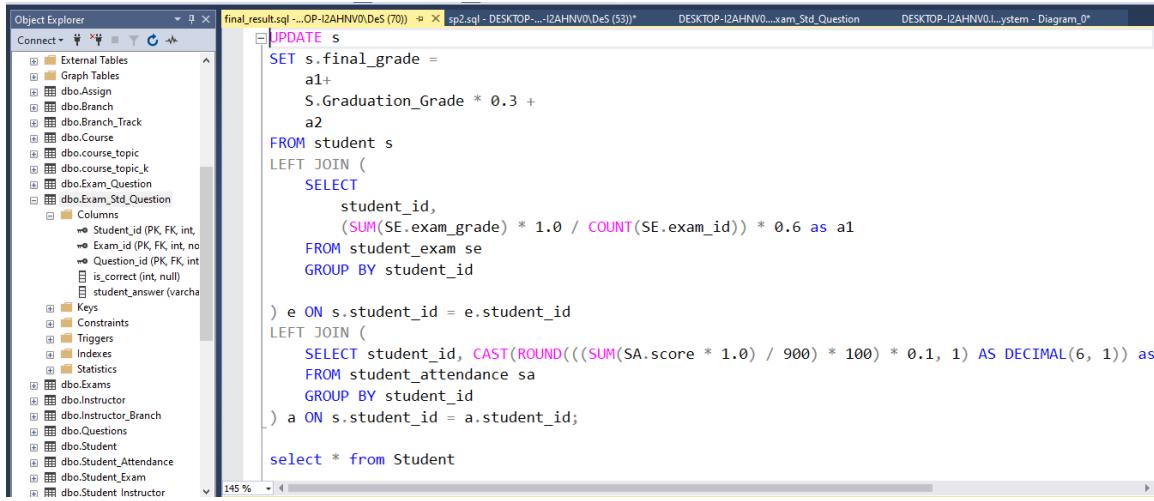
## 6 - Calculating the student's score on each exam to fill a column.

```
UPDATE se
SET se.exam_grade = scores.total_score
FROM student_exam se
JOIN (
    SELECT student_id, exam_id, SUM(is_correct) AS total_score
    FROM exam_std_question
    GROUP BY student_id, exam_id
) scores
ON se.student_id = scores.student_id AND se.exam_id = scores.exam_id;
```

The screenshot shows a SQL query being run in a database interface. The query updates the 'exam\_grade' column in the 'student\_exam' table ('se') with the total score from the 'scores' table. It joins the 'student\_exam' table with a derived table 'scores'. The derived table calculates the total score for each student per exam by summing the 'is\_correct' values. The 'exam\_grade' is set to the 'total\_score'. The results are displayed in a table with columns: student\_id, exam\_id, exam\_grade, and Texam\_date. The table has 55,162 rows. A message at the bottom indicates the query was executed successfully.

student_id	exam_id	exam_grade	Texam_date
1	13	85.71	2022-07-24 00:00:00.000
2	1	75.00	2022-06-22 00:00:00.000
3	1	66.67	2022-06-18 00:00:00.000
4	1	62.50	2022-07-10 00:00:00.000
5	1	71.43	2022-07-22 00:00:00.000
6	1	80.00	2022-06-07 00:00:00.000
7	1	90.00	2022-08-04 00:00:00.000
8	1	90.00	2022-07-13 00:00:00.000
9	2	51	2023-07-06 00:00:00.000
10	2	73	2023-06-22 00:00:00.000

## 7 - Calculating the final score for each student.



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer pane displays a tree view of database objects, including tables like 'dbo.Exam\_Question', 'dbo.Exam\_Std\_Question', and various system tables. The main window contains a T-SQL script for calculating a final grade:

```
final_result.sql (~OP-I2AHNV0.De5 (70))  sp2.sql - DESKTOP-...-I2AHNV0.De5 (53))*
UPDATE s
SET s.final_grade =
    a1+
    S.Graduation_Grade * 0.3 +
    a2
FROM student s
LEFT JOIN (
    SELECT
        student_id,
        (SUM(SE.exam_grade) * 1.0 / COUNT(SE.exam_id)) * 0.6 as a1
    FROM student_exam se
    GROUP BY student_id
) e ON s.student_id = e.student_id
LEFT JOIN (
    SELECT student_id, CAST(ROUND(((SUM(SA.score * 1.0) / 900) * 100) * 0.1, 1) AS DECIMAL(6, 1)) as
    FROM student_attendance sa
    GROUP BY student_id
) a ON s.student_id = a.student_id;
select * from Student
```

## 8 - Simulating exam dates for different branches and subjects.

## Data Cleaning and Validation:

- 1 -Removing duplicate data and invalid entries.
- 2 -Standardizing names, countries, and date formats.
- 3- Data Enhancement: Adding realistic exam results and subject coverage data.
- 4- Associating instructors with specific subjects and branches based on business rules.
- 5- Data preparation included validation, transformation, and alignment with ITI's training structure.

### 6- Temporary Suspension of Incremental Primary Key Process

We temporarily suspended the incremental primary key process to allow for the insertion of new data that does not conform to the existing incremental sequence logic.

```
--HOW TO STOP IDENTITY AND RETURN RUN-----
GO
set identity_insert instructor on
insert into instructor(Instuctor_Name,City,Phone,Gender,Faculty,instructor_Password,N_Year_In_Learning,Email,(select *
from ins_temp
set identity_insert instructor off
GO
select * from instructor
GO
drop table ins_temp
insert into instructor (Instuctor_Name)
values
('body')
select *
from Instructor
where Instuctor_name='body'
```

## 4. Stored Procedures

Eight stored procedures were created to automate data insertion and transformation:

### 1. show\_first\_question

- It is used to ensure that the student has not taken this exam before.
- If this is the first time the student is taking the exam, the first question is returned to him and his data is stored in exam\_std\_question
- Also use it to link to the start button (in the application).

```
create procedure show_first_question @exam_id int , @student_id int
as
begin |
    set nocount on;
    if exists (
        select 1
        from Exam_Std_Question
        where exam_id = @exam_id and Student_id=@student_id)

        BEGIN
            RAISERROR('You Are already take this Exam' , 16, 1);
            RETURN;
        END;
    declare @fq int
    select top 1 @fq=Question_ID
    from Exam_Question
    where Exam_ID=@exam_id
    order by Question_ID;

    select question_id,question,a,b,c,d
    from Questions
    where question_id=@fq;

    insert into Exam_Std_Question
    (Exam_id,Question_id,Student_id)
    values
    (@exam_id,@fq,@student_id);

    insert into Student_Exam
    (student_id,exam_id)
    values
    (@student_id,@exam_id)
end
go
```

## 2. check\_eachQ

- It's used to correct the last question a student answered.
- And enter the correction result in is\_correct column.
- Check whether the exam has finished or not.
- If it hasn't finished, the next question is displayed to the student.
- Store the next question in exam\_std\_question table.
- It's linked to the Next button (in the app).

```
--SP to take next question and correct old question
create procedure check_eachQ @st_id int ,@exam_id int,@answer varchar(50)
as
begin
    set nocount on;
    declare @q_id int;
    declare @correct varchar(50);
    declare @is_correct int;
    declare @N_qi int;

    select @q_id = Question_id
    from exam_Std_Question
    where Student_id = @st_id and Exam_id=@exam_id and student_answer is null;

    select @correct=Correct_answer
    from Questions
    where question_id=@q_id;

    set @is_correct = case when @answer=@correct then 1 else 0 end;

    update Exam_Std_Question
    set is_correct=@is_correct,student_answer=@answer
    where Student_id=@st_id and Question_id=@q_id and Exam_id=@exam_id;

    select top 1 @N_qi = Question_ID
    from Exam_Question
    where Exam_ID=@exam_id and Question_ID not in (
        select Question_id
        from Exam_Std_Question
        where Exam_id=@exam_id and Student_id=@st_id)
    order by Question_ID;
```

```
| IF @N_qi IS NULL
| BEGIN
|     Select 'Finish Exam' As 'message';
|     return;
| END

| insert into Exam_Std_Question
| (Exam_id,Question_id,Student_id)
| values
| (@exam_id,@N_qi,@st_id);

| select question_id,question,a,b,c,d
| from Questions
| where question_id=@N_qi;

End;
Go
```

### 3. Correct\_Student\_Exam

- It is used to correct exams by entering the exam number and student. The student's score for each question is calculated from Is\_Correct column and compared with the final exam score in exams table. The percentage is calculated and the result is stored in exam grade column in student exam table.
- It is linked to the "Show Grade" button( [in the app](#)).

```
CREATE PROCEDURE [dbo].[Correct_Student_Exam]
    @ExamID INT,
    @StudentID INT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @TotalQuestions INT;
    DECLARE @CorrectAnswers INT;
    DECLARE @MaxGrade DECIMAL(5,2);
    DECLARE @StudentGrade DECIMAL(5,2);
    DECLARE @GradePercentage DECIMAL(5,2);

    -- Count student's correct answers
    SELECT @CorrectAnswers = COUNT(*)
    FROM Exam_Std_Question
    WHERE Exam_ID = @ExamID
        AND Student_ID = @StudentID
        AND Is_Correct = 1;

    select @maxGrade = max_grade
    from exams
    where exam_id=@ExamID

    select @TotalQuestions=count(*)
    from Exam_Question
    where exam_id=@ExamID

    -- Calculate grade and percentage
    SET @StudentGrade = (@CorrectAnswers * @MaxGrade) / @TotalQuestions;
    SET @GradePercentage = (@StudentGrade / @MaxGrade) * 100;
    select @GradePercentage

END;
```

#### 4. GetFinalGrade

- Calculate final grade from exam score, which represents 60% of the assessment, their graduation project score, which represents 30% of the assessment, and their absence score, which represents 10% of the final assessment
- Insert value into final grade in student
- This has been linked to the submit button (in the application)

```
--AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @FinalExamGrade FLOAT;
    DECLARE @FinalGraduationGrade FLOAT;
    DECLARE @AttendanceScore FLOAT;
    DECLARE @FinalPercentage FLOAT;

    SELECT @FinalGraduationGrade = S.Graduation_Grade * 0.3 FROM Student S
    WHERE S.Student_ID = @Student_ID;

    SELECT @FinalExamGrade = (sum(se.exam_grade) / count(exam_id)) * 0.6
    FROM Student_Exam se
    WHERE se.Student_ID = @Student_ID;

    SELECT @AttendanceScore = cast(ROUND((SUM(score * 1.0) / 900) * 100) * 0.1, 1)as decimal (6, 1)
    FROM Student_Attendance
    WHERE Student_id = @Student_ID;

    IF @FinalExamGrade = 0
    BEGIN
        PRINT 'No exams found for this student.';
        RETURN;
    END

    set @FinalPercentage = @FinalExamGrade + @FinalGraduationGrade + @AttendanceScore ;
    select @FinalPercentage

END
```

#### 5. ins\_show\_question

- Displaying the available questions for the instructor based on their course to choose from.
- This has been linked to the View Questions button (in the application).

```
--show your questions
create procedure show_q @ins_id int ,@exam_id int
as
begin
    select q.question_id,q.question,q.A,q.B,q.C,q.D,q.Correct_answer
    from Exam_Question e join Questions q
    on e.Question_ID=q.question_id
    where Exam_ID=@exam_id

end
go
```

## 6. sp\_CreateExam

- The student selects the number of true/false questions and multiple choice questions and enters the exam score.
- Ensure that the instructor is providing a specific course.
- Ensure that the question table contains a sufficient number of questions for the course they are requesting.
- If the number is sufficient, an exam is created and placed in exam\_question table.
- Insert exam\_id into exams table with date
- Return exam number .
- It is used and linked to the Create Exam button (in the application).

```
--create random questions
ALTER PROCEDURE sp_CreateExam
    @InstructorID INT,          -- Input: the instructor creating the exam
    @NumTF INT,                 -- Input: number of True/False questions
    @NumMCQ INT,
    @Ex_Grade int               -- Output: the ID of the created exam
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @CourseID INT;
    DECLARE @AvailableTF INT;
    DECLARE @AvailableMCQ INT;

    -- Step 1: Get the course taught by this instructor
    SELECT TOP 1 @CourseID = Course_ID
    FROM Instructor
    WHERE Instructor_ID = @InstructorID;

    -- If instructor doesn't teach any course, stop
    IF @CourseID IS NULL
    BEGIN
        RAISERROR('Instructor does not teach any course.', 16, 1)
        RETURN
    END;

    -- Step 2: Check if there are enough True/False and MCQ questions
    SELECT @AvailableTF = COUNT(*)
    FROM Questions
    WHERE Course_ID = @CourseID AND Type = 'True/False';

    SELECT @AvailableMCQ = COUNT(*)
    FROM Questions
    WHERE Course_ID = @CourseID AND Type = 'MCQ';
```

```
Initial_grade SP-SQL-PIZAFINVU(DES(02)) Required_SPM_APP.S...IZAMINVU(DES(29)) ^ ^ SP-SQL-DES(KIOP-1...-IZAFINVU(DES(71))
-- If not enough TF questions
IF @AvailableTF < @NumTF
BEGIN
    RAISERROR('Not enough True/False questions in the database.', 16, 1)
    RETURN
END;

-- If not enough MCQ questions
IF @AvailableMCQ < @NumMCQ
BEGIN
    RAISERROR('Not enough MCQ questions in the database.', 16, 1)
    RETURN
END;

-- Step 3: Create the exam and get the new ExamID
INSERT INTO Exams (Course_ID, Date,max_grade,instructor_id)
VALUES (@CourseID, GETDATE(),@Ex_Grade,@InstructorID);

declare @ExamID int;
SET @ExamID = SCOPE_IDENTITY();
-- Step 4: Insert randomly selected True/False questions
INSERT INTO Exam_Question (Exam_ID, Question_ID)
SELECT TOP (@NumTF) @ExamID, Question_ID
FROM Questions
WHERE course_id = @CourseID AND Type = 'True/False'
ORDER BY NEWID();

-- Step 5: Insert randomly selected MCQ questions
INSERT INTO Exam_Question (Exam_ID, Question_ID)
SELECT TOP (@NumMCQ) @ExamID, question_id
FROM Questions
WHERE course_id = @CourseID AND Type = 'MCQ'
ORDER BY NEWID();
-- Return the Exam ID
select @ExamID
```

## 7. show\_q

- show questions in your exam

```
--show available questions
alter procedure ins_show_question @ins_id int
as
begin
    declare @Course_ID int;
    SELECT @Course_ID = Course_ID
    FROM Instructor
    WHERE Instructor_ID = @ins_id;

    select question_id,question,A,B,C,D,Correct_answer
    from Questions
    where course_id=@Course_ID

end
go
```

## 8. ins\_stu

- insert data into student

```
create procedure ins_stu (@student_name nvarchar(max), @Email nvarchar(100), @Password nvarchar(250), @Faculty nvarchar(100), @GPA float, @Phone nvarchar(15), @City nvarchar(50), @Gender nvarchar(10)
as
begin
    INSERT INTO Student (Student_name, Email, Password, Faculty, Graduation_Grade, Phone, City,Gender)
    OUTPUT INSERTED.Student_ID, INSERTED.Password
    VALUES (@Student_name , @Email , @Password , @Faculty , @GPA , @Phone , @City ,@Gender )
end
go
```

- It is used and linked to the login button (in the application).

## 9. ins\_inst

- insert data into instructor
- It is used and linked to the login button (in the application).

```
create procedure ins_instruct @ins_name nvarchar(max), @Email nvarchar(100), @Faculty nvarchar(100), @Phone nvarchar(15), @City nvarchar(50), @Gender nvarchar(10)
as
begin
    INSERT INTO Instructor(instructor_name, Email, Faculty, Phone, City,Gender)
    OUTPUT INSERTED.Instructor_ID
    VALUES (@ins_name , @Email , @Faculty , @Phone , @City ,@Gender )
end
go
=====
-----Sp FOR INSTRUCTORS-----
--show available questions
```

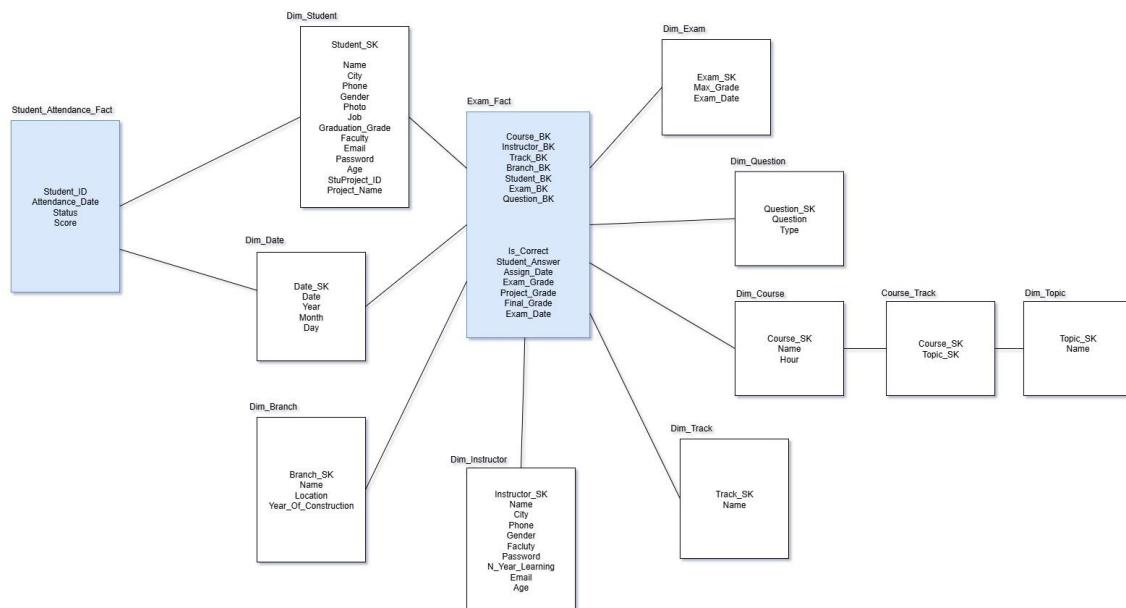
## 5. Data Warehouse Design

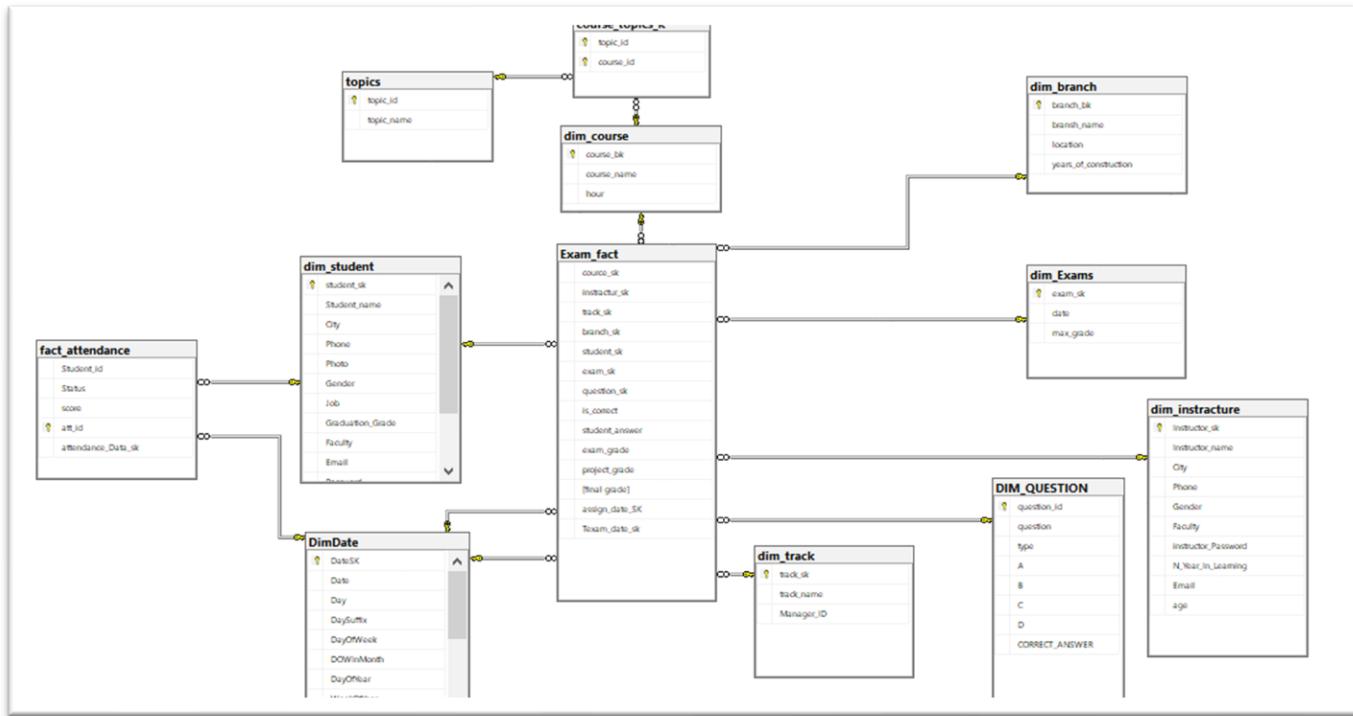
Introduction: Why Do We Need to Transform Data into a Data Warehouse?

As part of the ITI Graduation Project, our goal is to build a robust Data Warehouse solution that enables better data management, reporting, and decision-making.

In today's data-driven world, organizations generate vast amounts of data from various sources — such as operational systems, logs, transactions, and user interactions. However, this raw data is often inconsistent, fragmented, and not optimized for analysis.

A **Galaxy Schema** was designed to support complex analysis, using multiple fact tables and shared dimension tables. This schema allows flexibility and efficiency in reporting.





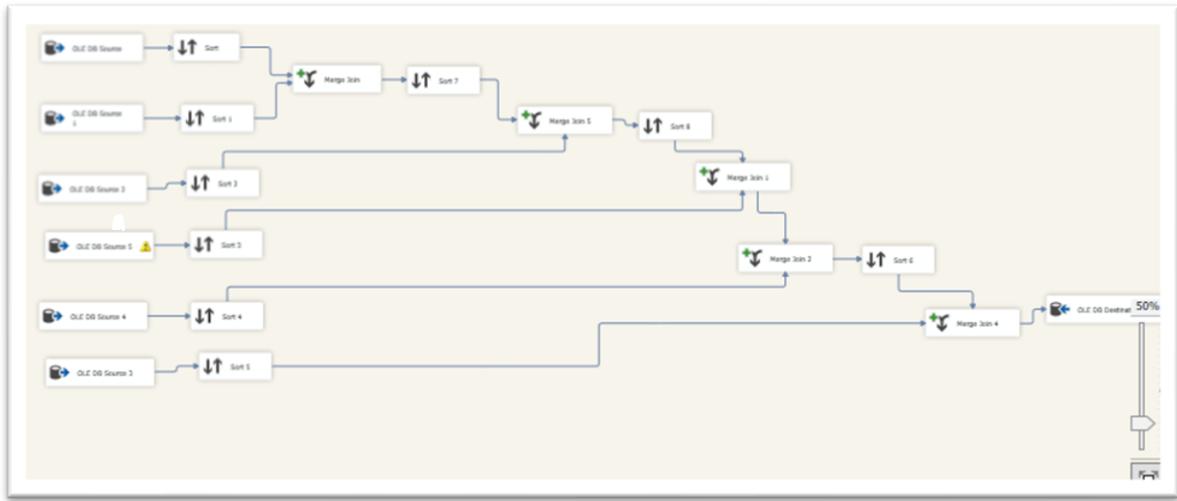
- Exam\_Fact table
- Attendance\_Fact table

dimensions include Students, Instructors, Track, Topics, Branches, Exams, and Questions,Course,DIM\_date.

## 6. ETL Process in DWH

The ETL process was built using Visual Studio with SSIS integration. It included Extract, Transform, and Load stages, ensuring data integrity and business rule enforcement.

- Sort, and merge join were applied, and the output was loaded into the Galaxy Schema for analysis.



- Insert all other tables using SQL queries

Note : also using ETL to insert data into DB (**in Data source section point 4**)

## 7. SSRS Reports

Multiple reports were created using SSRS to deliver printable and parameter-driven insights. Key reports included:

### 1. Report that returns the students information according to Department No parameter (each Department in one page)

- If student grade greater than 50 font color will be blue else will be red  
`=iif(Fields!final_grade.Value <50 , "yellow" , "MidnightBlue")`
- we utilized **SQL Views** as a layer of abstraction on top of the physical tables to simplify complex queries and improve usability for reporting and analytics.

```
create view students_according_track_view
as (
    SELECT
        e.student_sk,
        s.Student_name,
        s.Faculty,
        s.Graduation_Grade,
        s.Gender,
        s.Job,
        s.Phone,
        t.track_name,
        e.project_grade,
        e.[final grade]
    FROM
        Exam_fact e
    INNER JOIN
        dim_student s ON e.student_sk = s.student_sk
    INNER JOIN
        dim_track t ON e.track_sk = t.track_sk)
```

- Then use it inside Stored procedure to create report

```

alter PROCEDURE students_according_track
AS
BEGIN
    SET NOCOUNT ON;
    select * from students_according_track_view

END
exec students_according_track
go

```

- Report in vs

Design Preview

100% Find | Next

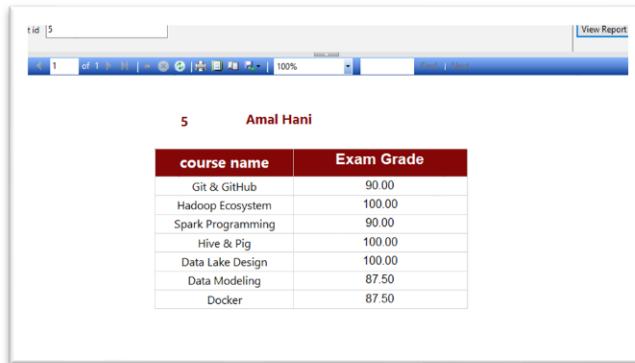
.NET Development

student sk	Student name	Graduation Grade	project grade	final grade	Phone	Faculty	Gender	Job
3443	Reda Haytham	95	99.08	76.87	2.00102e+012	Arts	Female	International
5392	Daoud Tareq	83.55	78.88	78.78	2.00114e+012	Medicine	Female	None
8983	Rafiq Hani	86.35	66.24	73.94	2.00119e+012	Medicine	Male	None
4272	Ali Karam	82.79	68.42	67.84	2.00106e+012	Science	Female	National
2752	Melad Yahya	78.92	81.55	83.95	2.00119e+012	Commerce	Male	International
4062	Ghazal Ashraf	83.65	90.24	87.60	2.00106e+012	Arts	Male	National
5462	Houriya Fathi	65.17	68.46	71.04	2.00109e+012	Engineering	Female	International
1699	Zaina Demian	90.86	67.92	70.37	2.00113e+012	Pharmacy	Male	International
4412	Salma Sultan	98.3	65.36	63.66	2.00106e+012	Computer Science	Male	National
8834	Yara Abdel Tawab	95.77	71.99	75.57	2.0013e+012	Pharmacy	Male	National
3939	Philip Lutfi	81.95	88.97	82.54	2.00109e+012	Science	Female	National
8823	Jumana Ramzy	65.3	67.08	75.34	2.00119e+012	Arts	Male	International
7223	Mirna Lotfy	86.95	62.02	79.15	2.00107e+012	Engineering	Female	International
4849	Affif Mansour	80.82	78.88	76.47	2.00126e+012	Commerce	Female	International
4362	Burhan Lotfy	87.6	99.05	92.56	2.00112e+012	Pharmacy	Female	International

**2. Report that takes the student ID and returns the grades of the student in all courses. %**

- Using stored procedure that take student id and returns grades

```
alter procedure grade_in_course @st_id int  
as  
begin  
select f.student_sk,student_name,c.course_name,max(f.exam_grade) as [Exam Grade]  
from Exam_fact f join dim_student s  
on f.student_sk=s.student_sk  
join dim_course c  
on f.course_sk=c.course_pk  
where f.student_sk=@st_id  
group by f.student_sk,student_name,f.course_sk,c.course_name  
end  
  
exec grade_in_course 5  
go
```



The screenshot shows a report viewer window with a toolbar at the top. The toolbar includes buttons for back, forward, search, and zoom, along with a status bar showing '100%' and a 'View Report' link. Below the toolbar, the report title is '5 Amal Hani'. The report content is a table with two columns: 'course name' and 'Exam Grade'. The data is as follows:

course name	Exam Grade
Git & GitHub	90.00
Hadoop Ecosystem	100.00
Spark Programming	90.00
Hive & Pig	100.00
Data Lake Design	100.00
Data Modeling	87.50
Docker	87.50

**3. Report that takes the instructor ID and returns the name of the courses that he teaches and the number of students per course.**

- View take return the branch he works in, the number of students he teaches in each branch, the course he gives, the tracks, and whether he is a manager or not, and if he is a manager, he is the manager of any branch.

```

CREATE VIEW vw_Instructor_Courses AS
SELECT
    f.instructor_sk,
    i.Instructor_name,
    c.course_name,
    b.branch_name,
    f.student_sk,
    dt.track_name,
    dt.Manager_ID
FROM Exam_fact f
JOIN dim_instructure i ON f.instructor_sk = i.Instructor_sk
JOIN dim_course c ON f.course_sk = c.course_bk
JOIN dim_branch b ON b.branch_bk = f.branch_sk
JOIN dim_track dt ON f.track_sk = dt.track_sk;

```

- Sp take instructor id and insert it into view

```

ALTER PROCEDURE courses_instructor (@ins_id INT)
AS
BEGIN
SELECT
    v.instructor_sk,
    v.Instructor_name,
    v.course_name,
    v.branch_name,
    COUNT(DISTINCT v.student_sk) AS [Number of students],
    CASE
        WHEN @ins_id = v.Manager_ID THEN 'manager'
        ELSE 'not manager'
    END AS position,
    v.track_name
FROM vw_Instructor_Courses v
WHERE v.instructor_sk = @ins_id
GROUP BY v.instructor_sk, v.Instructor_name, v.course_name, v.branch_name, v.Manager_ID, v.track_name;
END;
exec courses_instructor 13

```

course name	branch name	Number of students	track name
C# Programming	Alexandria	143	Game Development
C# Programming	Alexandria	145	.NET Development
C# Programming	Alexandria	143	Augmented & Virtual Reality
C# Programming	Assuit	143	.NET Development
C# Programming	Smart Village	143	.NET Development
C# Programming	Smart Village	143	Augmented & Virtual Reality

**4. Report that takes course ID and returns its topics**

```
create procedure topic_course @course_id int
as
begin
    select distinct course_sk,t.topic_name
    from Exam_fact f
    left join course_topics_k k on f.course_sk=k.course_id
    left join topics t on k.topic_id=t.topic_id
    where f.course_sk=@course_id

end

exec topic_course 1
```

topic name
Data Types
Exception Handling
File I/O
LINQ
OOP
Syntax & Structure

5. Report that takes exam number and returns the Questions in it and choices [freeform report]

```
CREATE PROCEDURE GetExamQuestionsReport
    @ExamID INT
AS
BEGIN
    -- الأسئلة مع خياراتها
    SELECT
        DISTINCT
        Q.Question_ID,
        q.Type,
        q.Question,
        q.A AS Choice1,
        q.B AS Choice2,
        q.C AS Choice3,
        q.D AS Choice4
    from Exam_fact f left join DIM_QUESTIONS q
    on f.question_sk=q.question_id
    WHERE F.exam_sk=@ExamID

END;
EXEC GetExamQuestionsReport 1;
GO
```

Exam ID

1 of 2 ? > | ← × | 50% | ↕

Exam Number : 1

1. Cloud storage allows access to data via internet.

A) TRUE  
B) FALSE  
C) Both  
D) None

---

2. S3 is Amazon's cloud storage service.

A) TRUE  
B) FALSE  
C) Both  
D) None

---

3. Java code runs on the JVM.

A) TRUE  
B) FALSE  
C) Both  
D) None

---

4. Cloud storage is always free.

A) TRUE  
B) FALSE  
C) Both  
D) None

6. Report that takes exam number and the student ID then returns the Questions in this exam with the student answers and return exam grade .

```
alter procedure student_answer_inexam @exam_id int, @student_id int
as
begin
SELECT
    DISTINCT
        Q.Question_ID,
        q.Type,
        q.Question,
        q.A AS Choice1,
        q.B AS Choice2,
        q.C AS Choice3,
        q.D AS Choice4,
        f.student_answer,
        f.exam_grade

    from Exam_fact f
    left join DIM_QUESTIONS q on f.question_sk=q.question_id
    WHERE F.exam_sk=@exam_id and f.student_sk=@student_id
end
```

## 8. Power BI Dashboards

20+ dashboards were developed to provide interactive analytics. Highlights include

### Introductory Pages

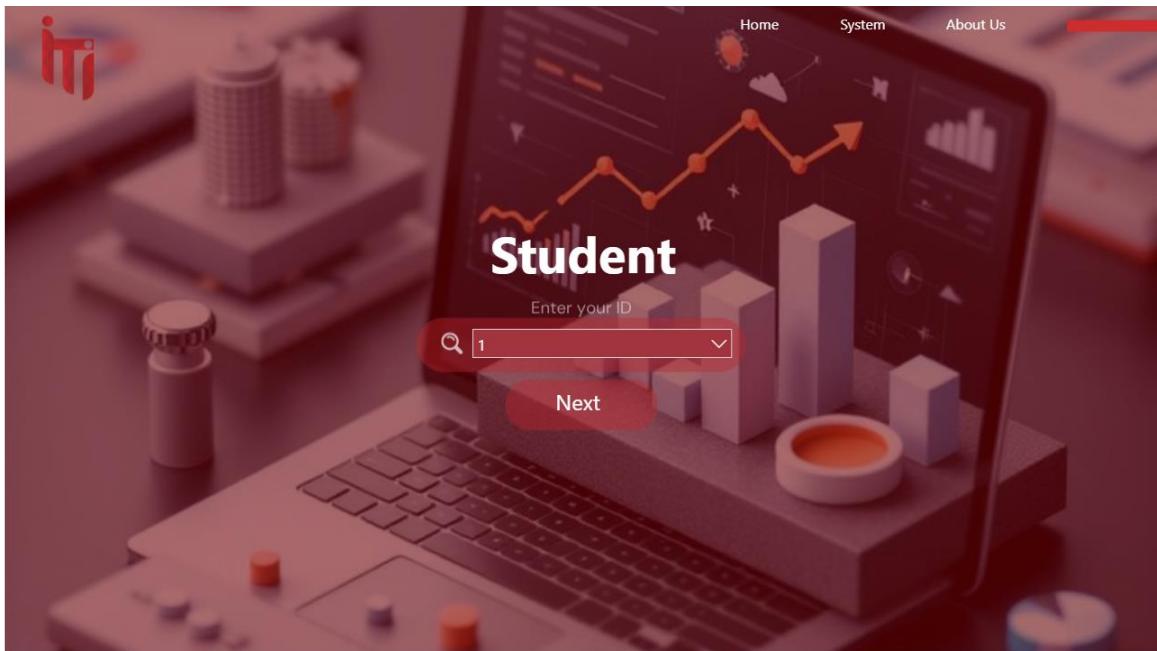
- Dashboard 1: System Welcome Page
- Introduces users to the dashboard system and its purpose. Explains the platform structure, who can access what, and how to use it.
- Dashboard 2: Role-Based Access Overview

Clarifies what students, instructors, and ITI management can view. Table of permissions, navigation tips, and expected use.





**Student Dashboard**



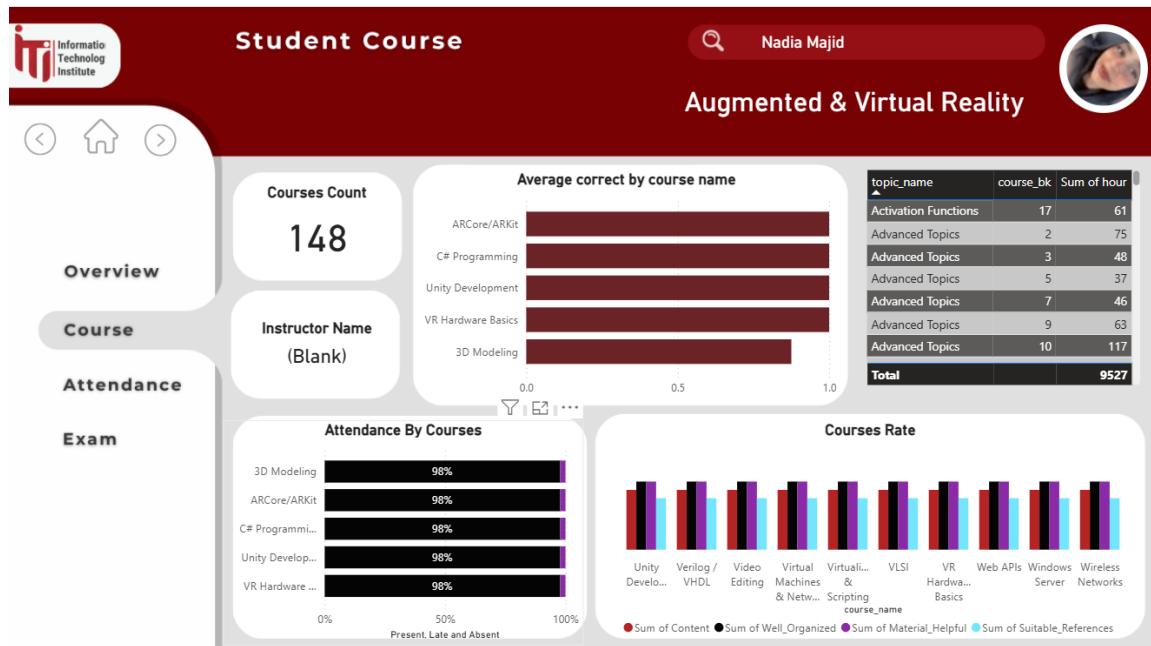


- Dashboard 3: Overall Student Performance

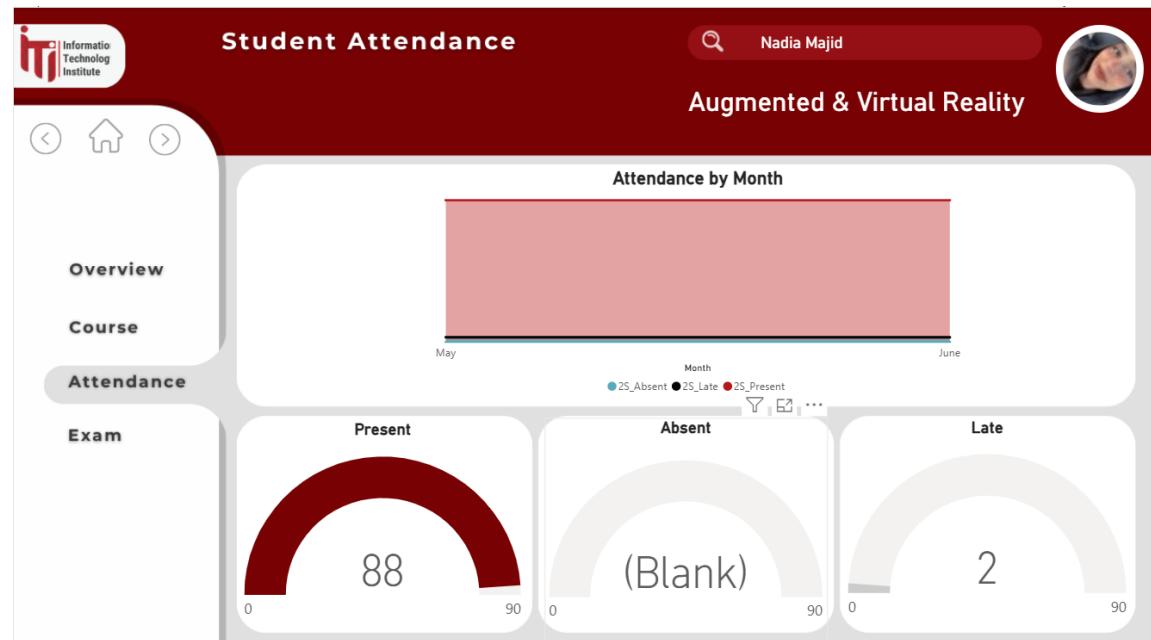
The screenshot displays the 'Student Overview' dashboard for Nadia Majid. The top navigation bar includes the ITI logo, a search bar with 'Nadia Majid', and a user profile icon. On the left, a sidebar lists 'Overview', 'Course', 'Attendance', and 'Exam'. The main area features six data cards:

- # Rank: 4
- Branch: Alexandria
- Final Grade: 96.3 %
- City: Damietta
- Faculty: Science
- Assign Date: 5/1/2022

- Displays the student's progress across attendance, scores, and participation.



- Dashboard 4: Attendance & Absence Overview



- Shows the student's attendance rate, absence days, and late arrivals.

**Student Exam**

Nadia Majid

Augmented & Virtual Reality

Exam ID: 10 | Exam Grade: 100.00 | AVG Exams Grade: 100.00 | #Exams: 1

**Overview**

**Course**

**Attendance**

**Exam**

course\_name: ARCore/ARKit

**Type of Question**

Correct Answer Percentage: 100

C	D	student_answer	CORRECT_ANSWER	Status
Both	None	A	A	Correct
Both	None	A	A	Correct
Both	None	A	A	Correct
Both	None	A	A	Correct
Gyroscope	Flash	B	B	Correct
UIKit	CoreML	B	B	Correct
Camera	Sensor	A	A	Correct
Microphone	Flash	A	A	Correct
SceneForm	OpenGL	A	A	Correct

- Dashboard 5: Assessment Grades

**Student Exam**

Nadia Majid

Augmented & Virtual Reality

Exam ID: 10 | Exam Grade: 100.00 | AVG Exams Grade: 100.00 | #Exams: 1

**Overview**

**Course**

**Attendance**

**Exam**

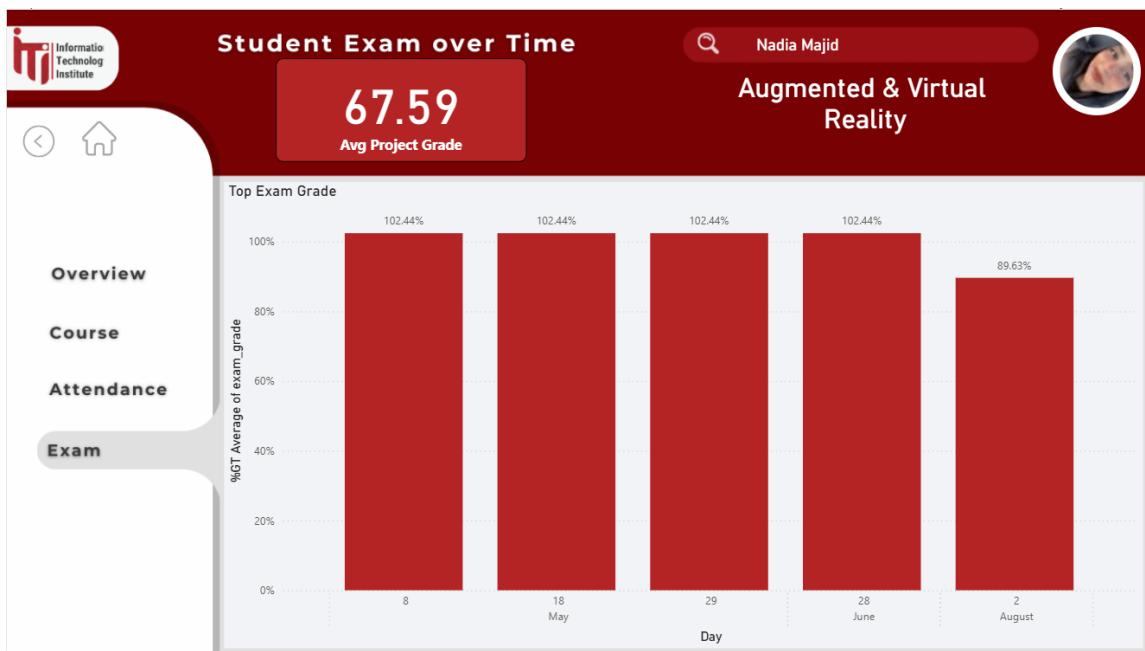
course\_name: ARCore/ARKit

**Type of Question**

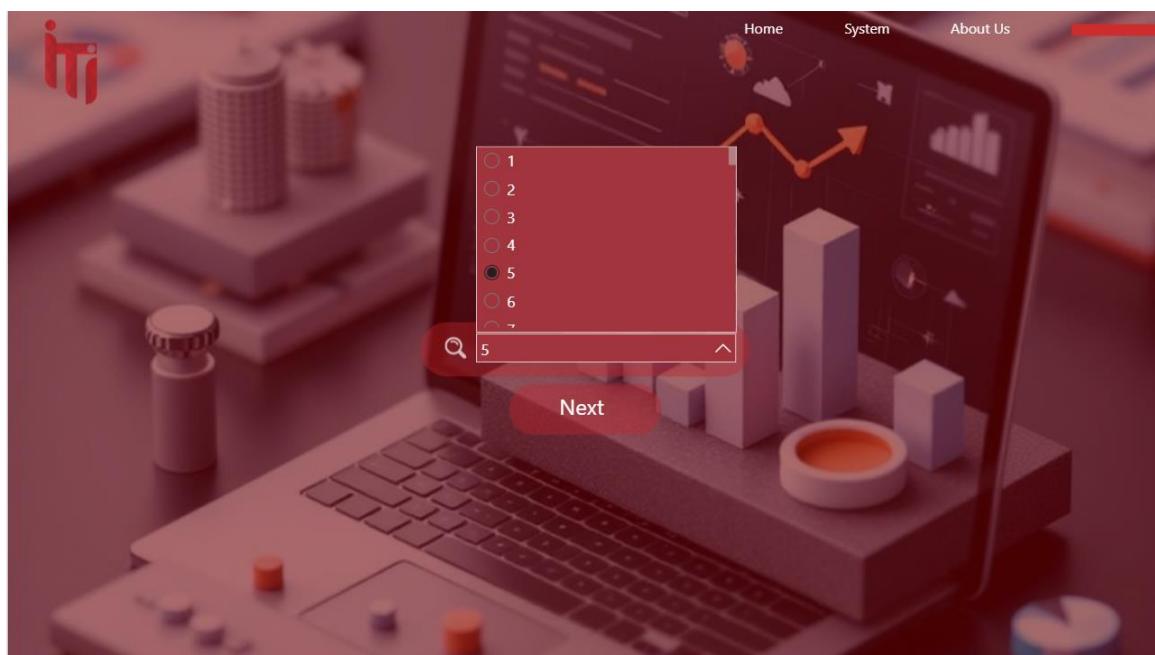
Correct Answer Percentage: 100

question	A	B
AR apps require GPU acceleration.	TRUE	FALSE
ARCore runs on Android.	TRUE	FALSE
ARKit is developed by Apple.	TRUE	FALSE
SLAM helps AR devices map the environment.	TRUE	FALSE
What improves depth detection?	Compass	LiDAR
What library renders 3D objects?	OpenCV	SceneKit
What places virtual content?	Anchor	Shader
Which detects depth?	LiDAR	Compass
Which SDK is for iOS?	ARKit	ARCore

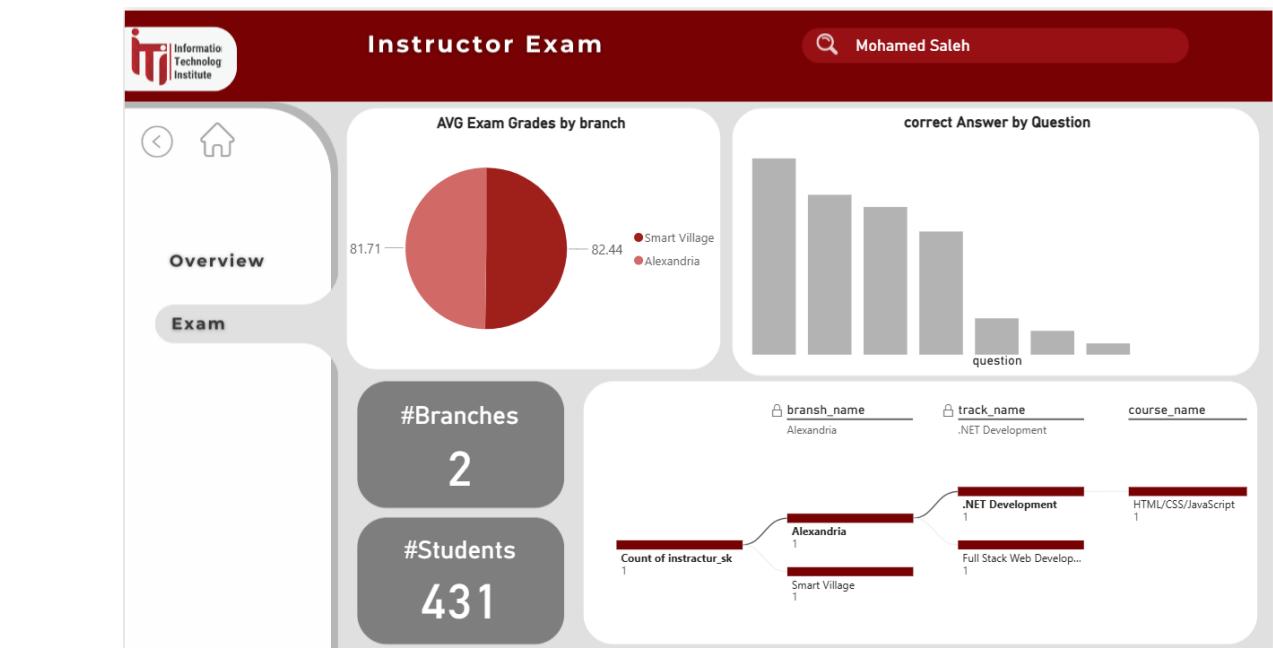
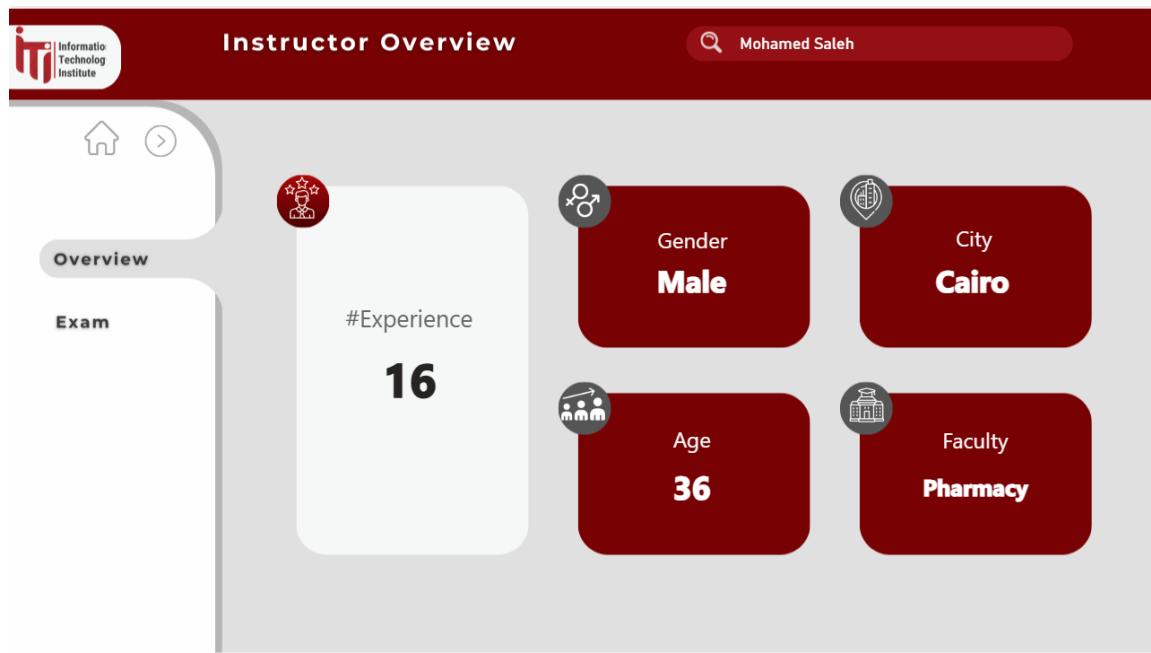
- Visualizes the scores of the student in various evaluations.



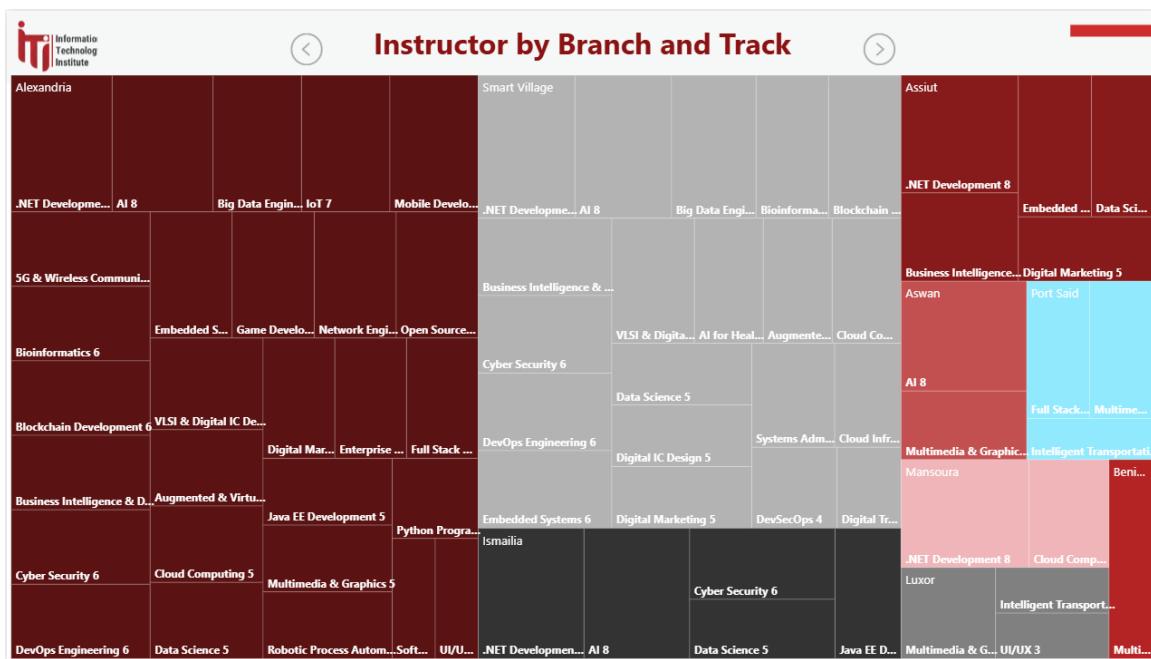
## Instructor overall view

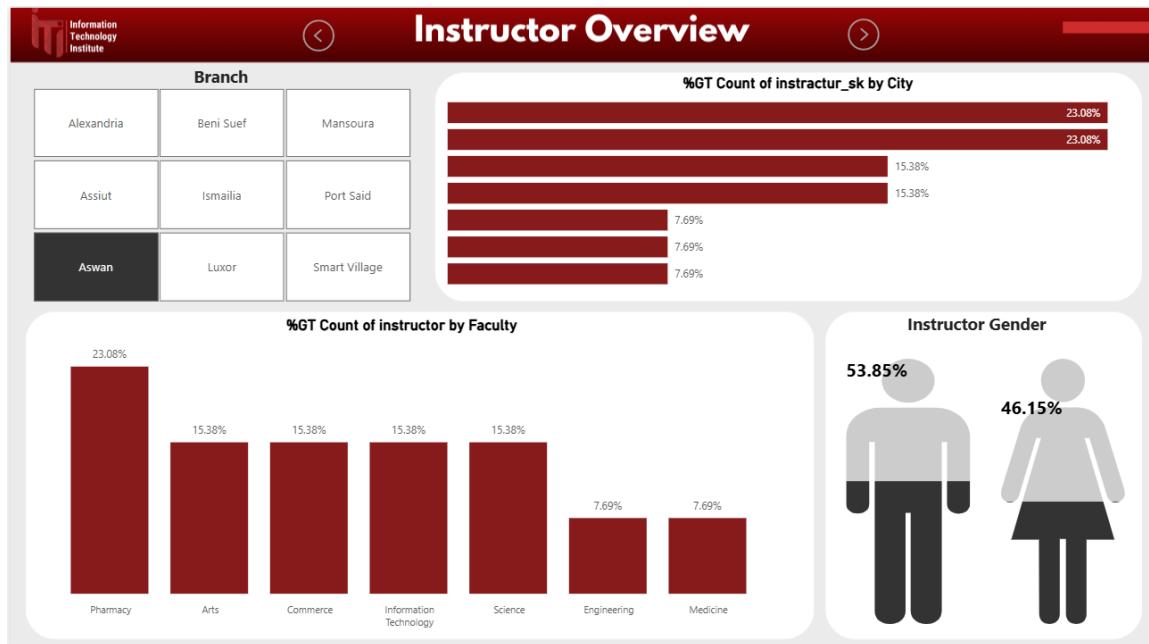
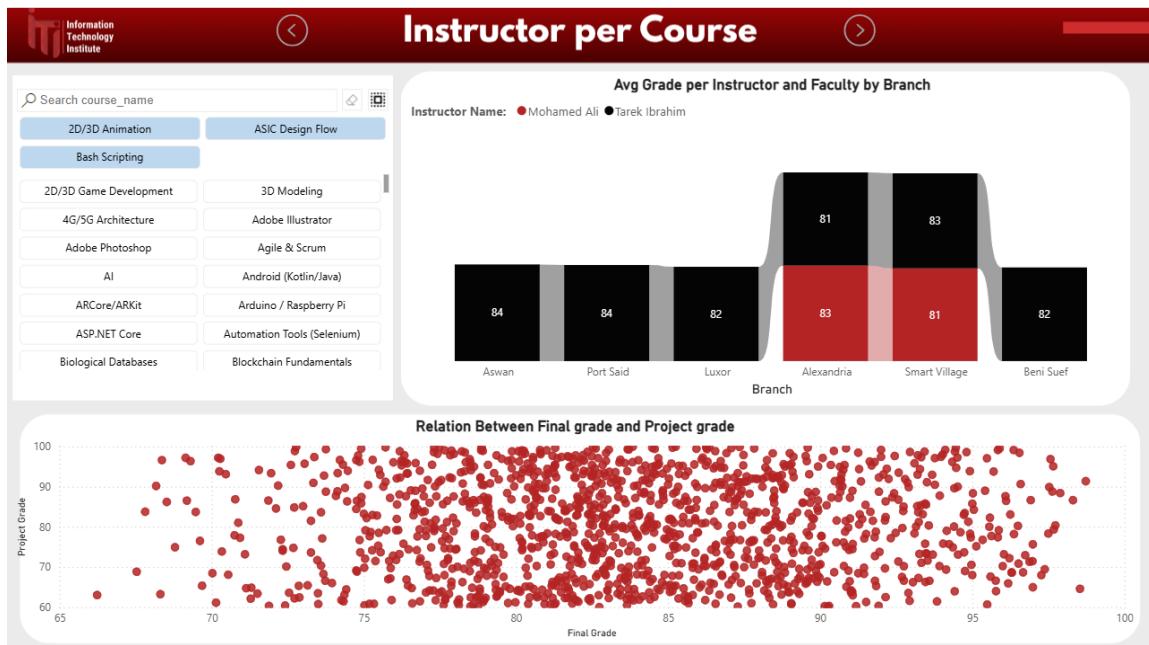


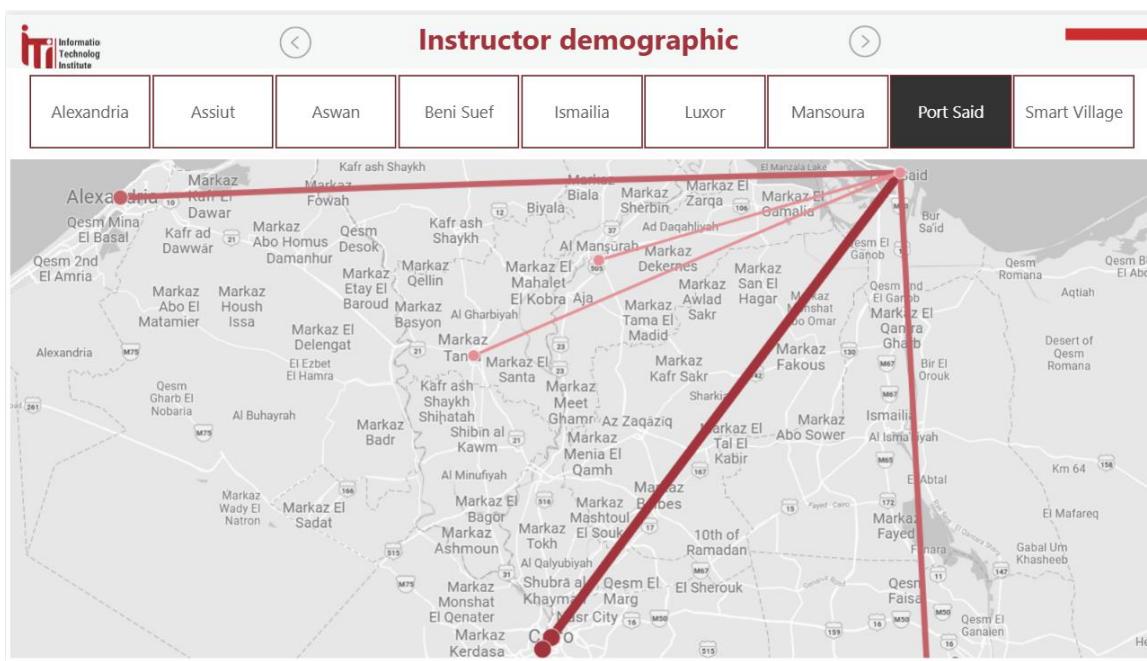
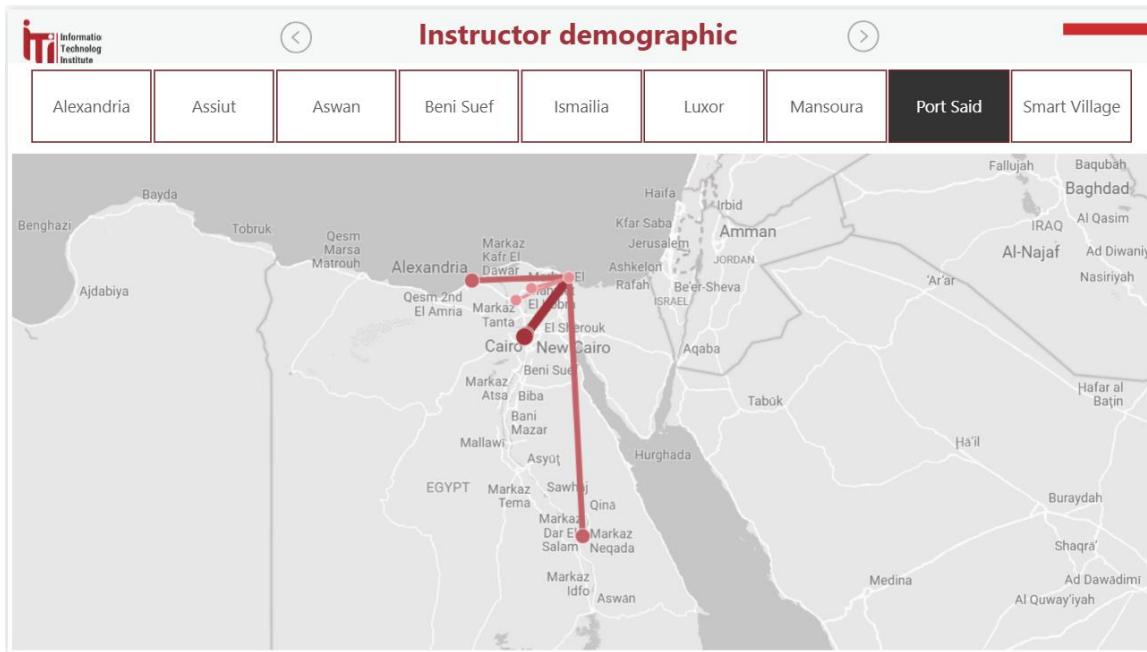
- Dashboard 6: About instructor

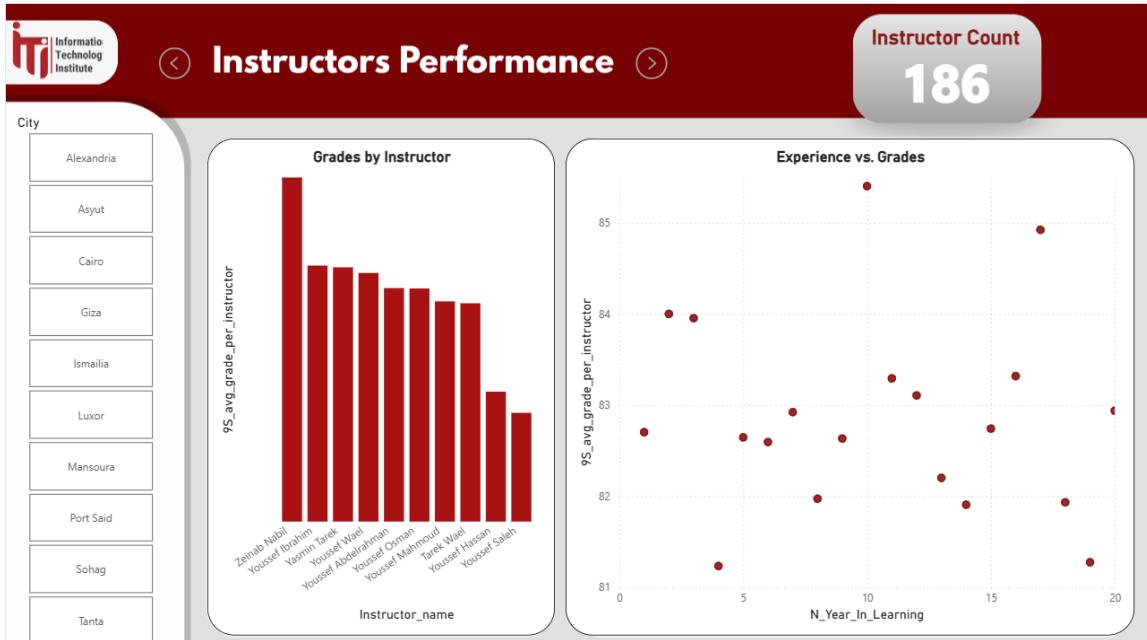
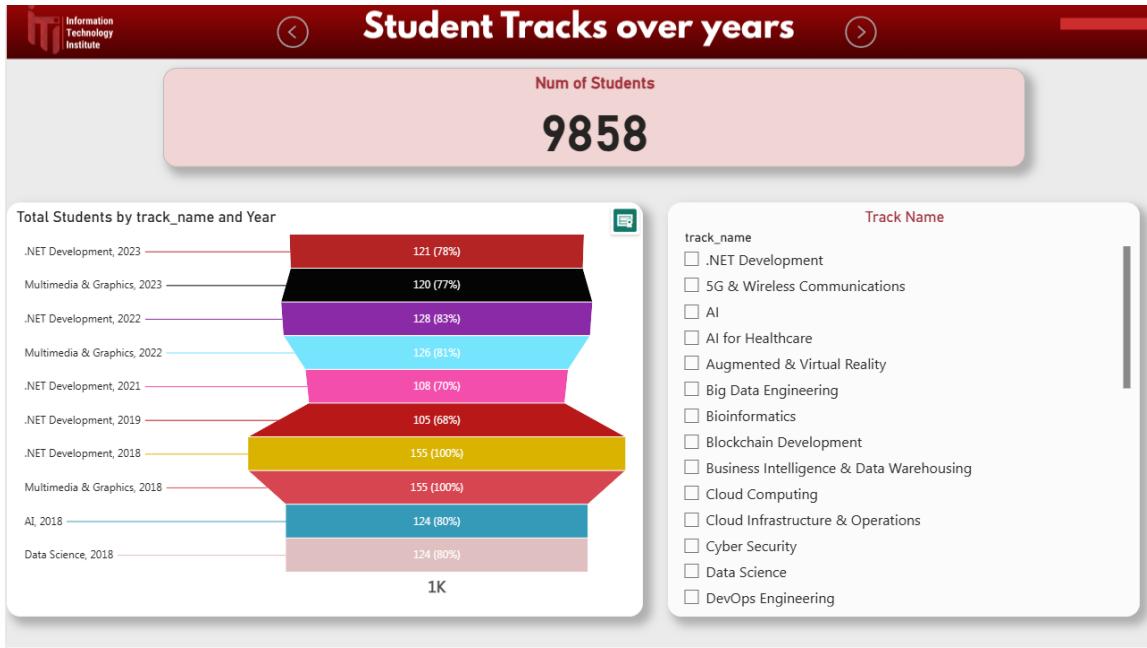


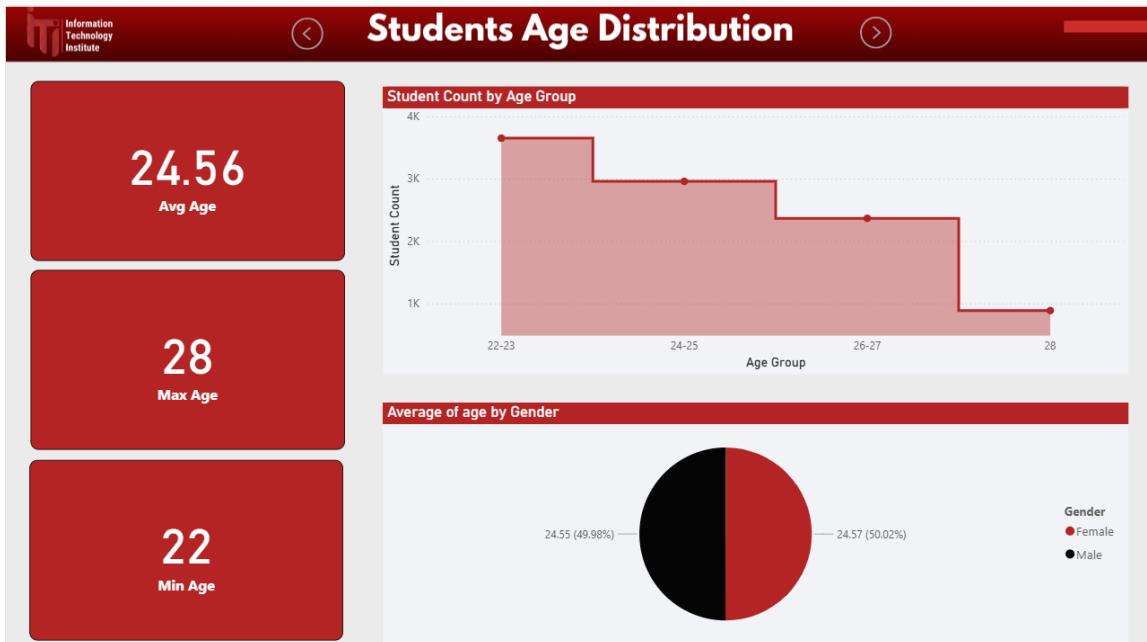
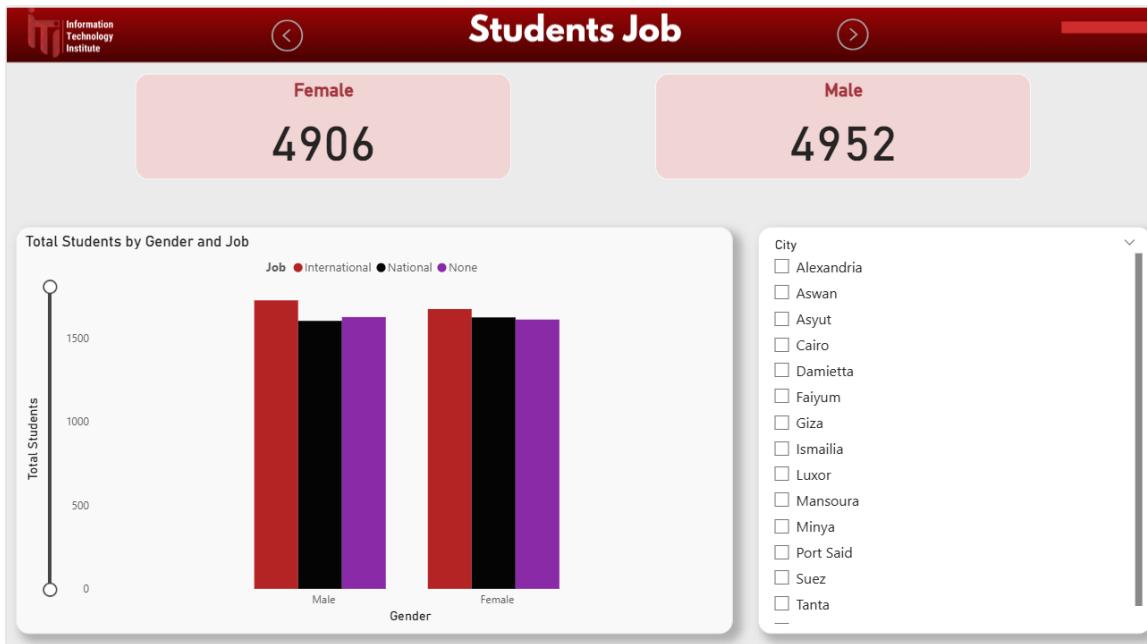
## ITI overview

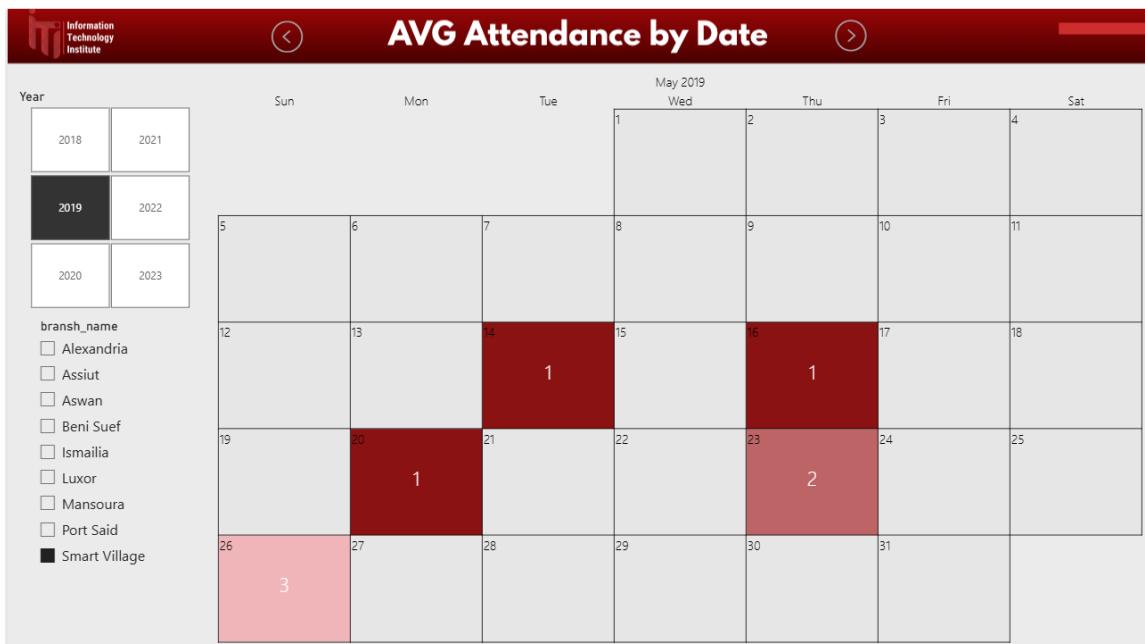
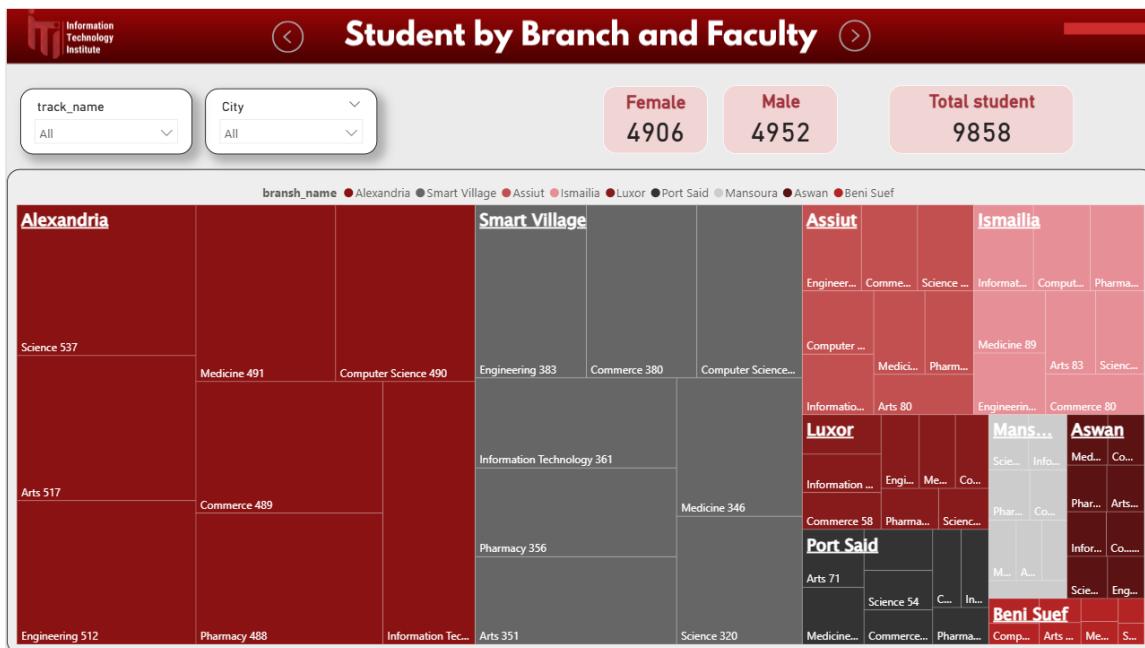


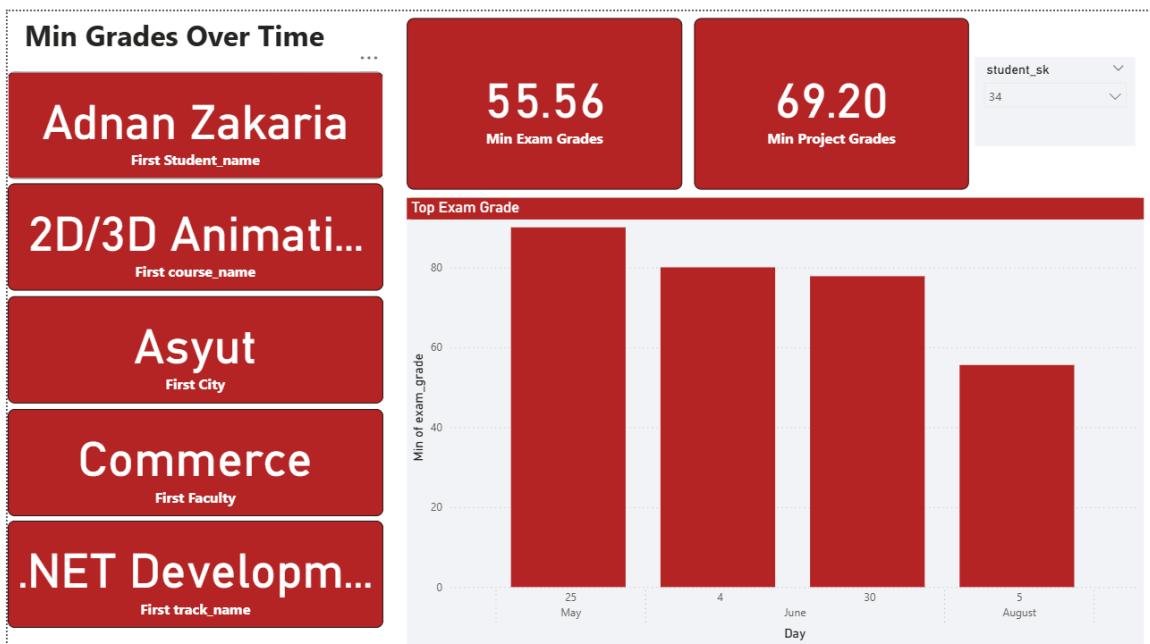
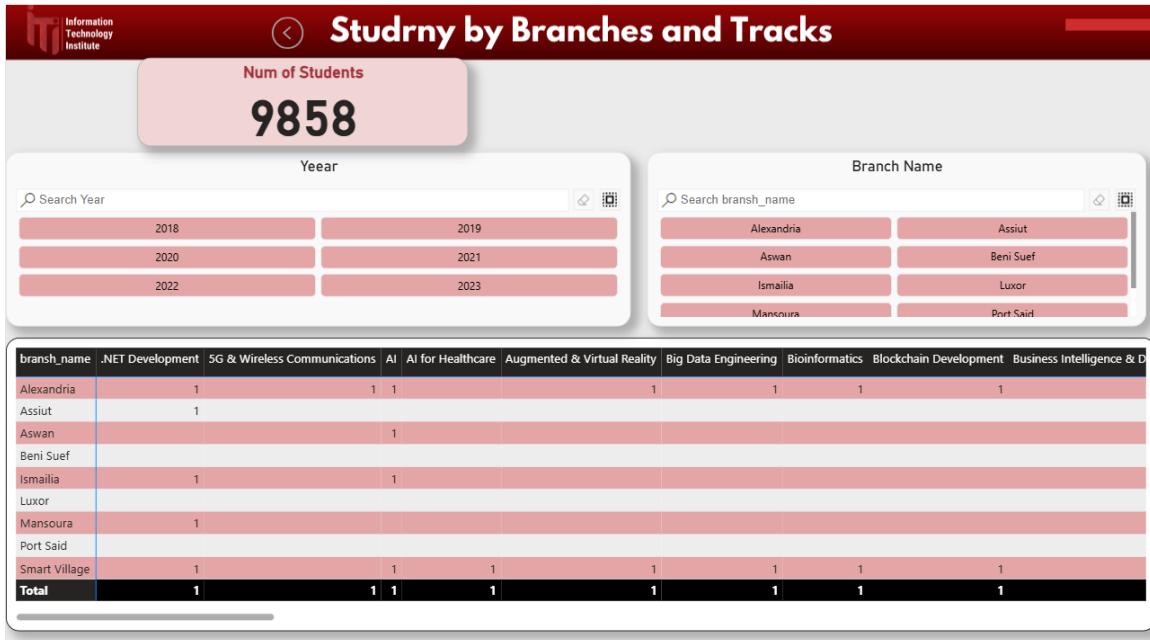


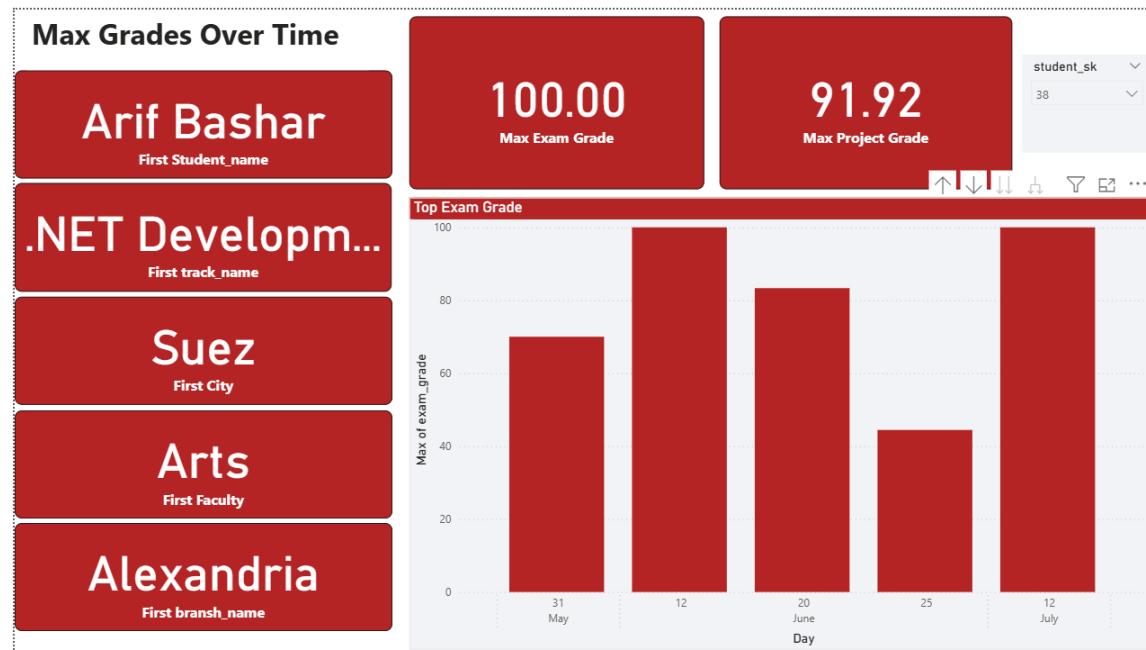
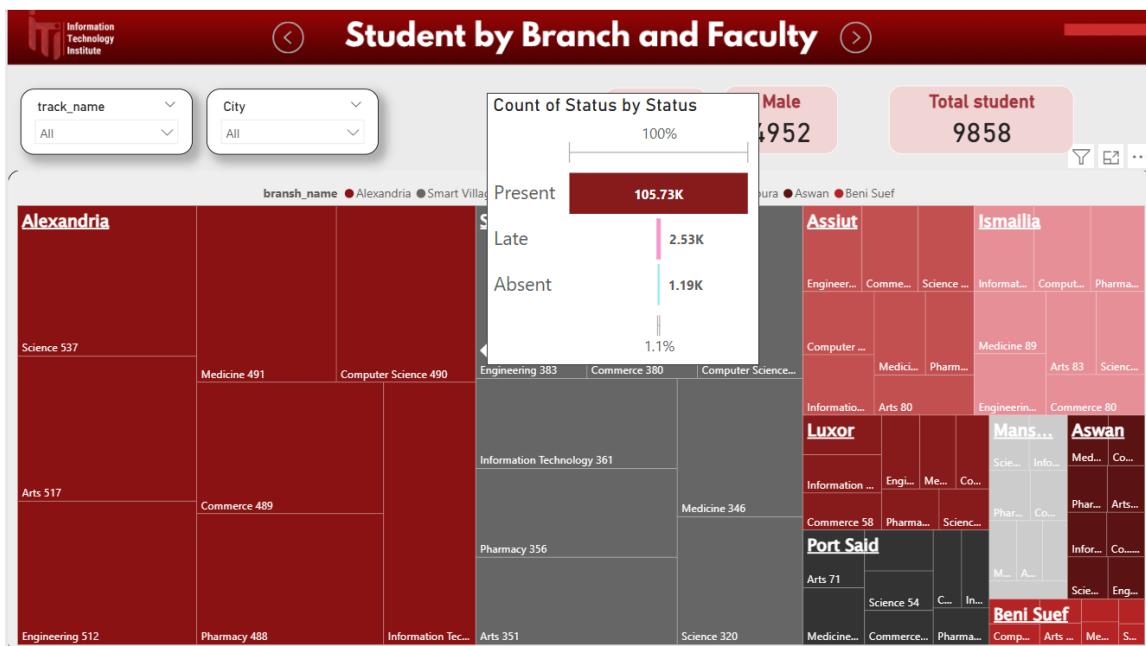












## ITI In Brief

The Information Technology Institute (ITI) was established back in 1993 and became one of the MCIT's affiliates in 2005. Its main mission revolves around ICT human capacity development. For over 30 years, ITI has kept updating its training portfolio through continuous monitoring of future global tech trends, as well as the country's national and mega projects. As a result, it sustained an 85% employment percentage for its graduates. ITI programs served thousands of Egyptian youth from different Egyptian governorates, who graduated from different academic backgrounds, thanks to the rich programs that built on their academic experience leveraging their future employment opportunities. These programs include development, content design, game development, Geographic Information System, media and animation, cloud, security, data analytics, among others. ITI has a wide network of relations with industry and academia, empowering its curriculum and helping in reaching out talents in most of Egypt's universities and governorates.

## 9. Challenges & Solutions

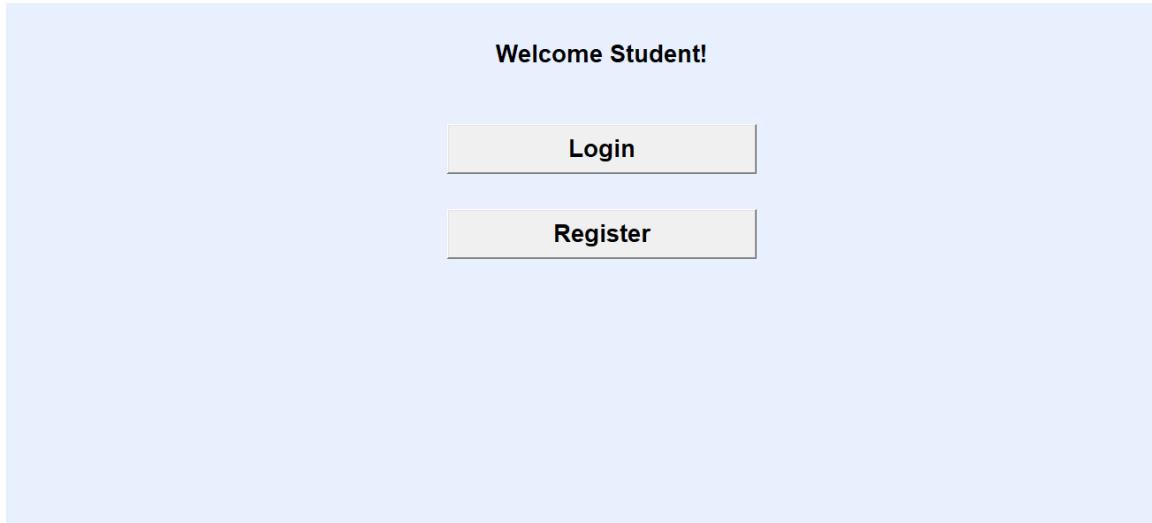
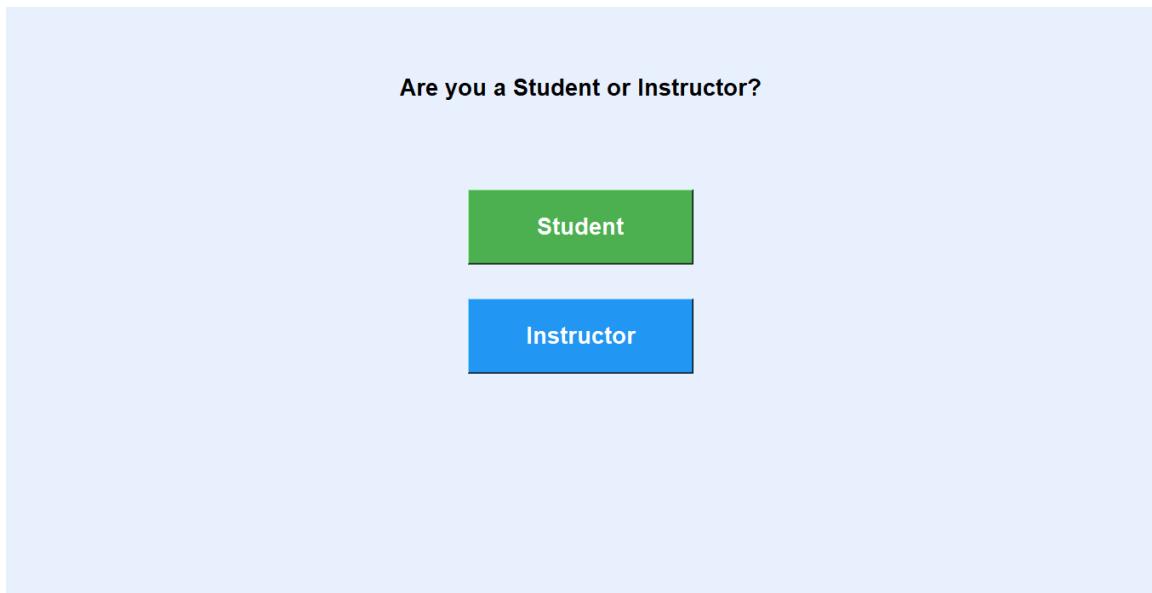
Several challenges were faced and solved during the project:

- Faced difficulty in accessing real-world data, so
  - we generated synthetic data using ChatGPT and a set of Python scripts, as documented.
- Encountered an issue with **Primary Key Identity**, which was
  - resolved by using **Inserted Identity** to manage auto-incremented keys correctly.
- Faced data redundancy due to storing student absence records in the same fact table as exam data. This caused excessive duplication.
  - Solved this by separating the absence records into a dedicated **Fact Table** for attendance.

## 10. Application

### Libraries and Tools Used

- **Tkinter:** Used to create a simple and user-friendly **Graphical User Interface (GUI)** that allows users to interact with the system easily.
- **tkinter.messagebox & tkinter.filedialog:** Utilized for displaying alerts, confirmations, and for enabling file selection features within the GUI.
- **pyodbc:** Used to establish a connection with the **SQL Server Database**, allowing for execution of SQL queries and seamless data manipulation from the Python interface.



### **Student Login**

Student ID:

Password:

**Login**

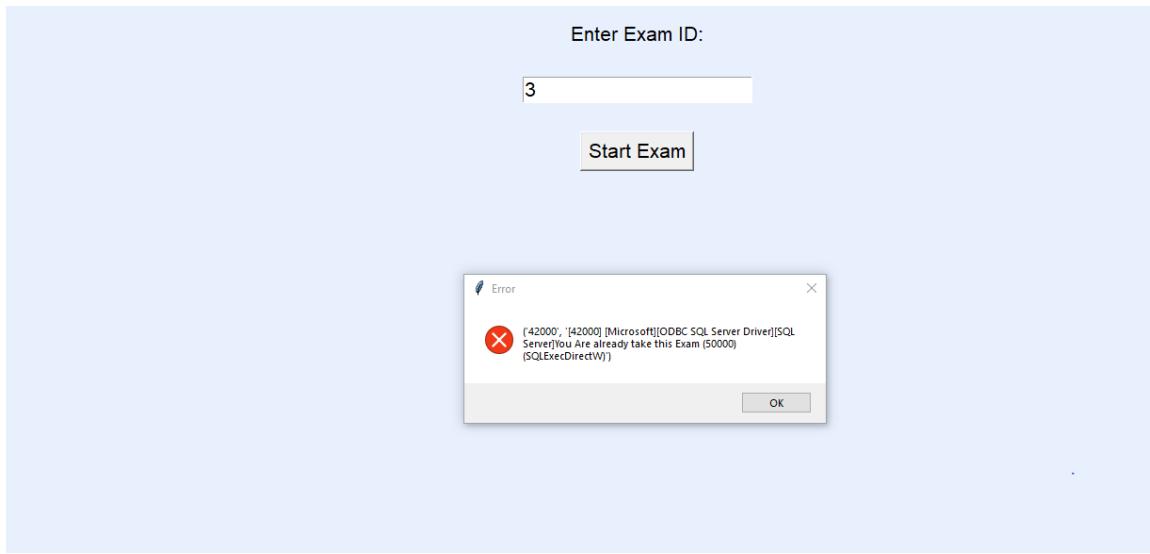


Figure 6 : When student try to enter exam that he already entered

Color correction improves visual quality.

- A) TRUE
- B) FALSE
- C) Both
- D) None

[Next](#)

Figure 7 : All questions will appear like this

**Exam Result : 42.86**

[Back to Menu](#)

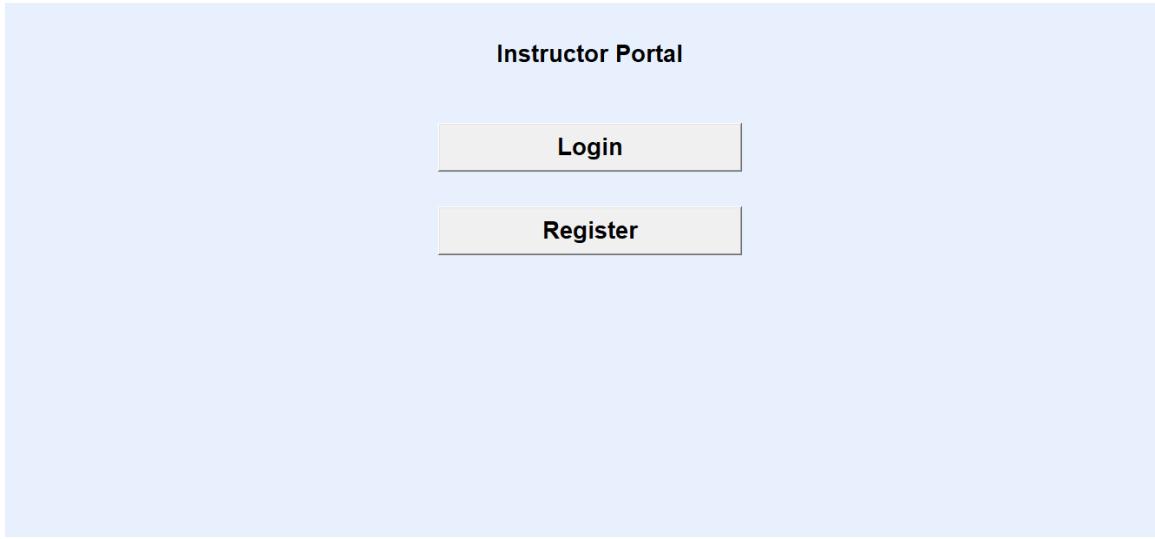


Figure 8 : When press Instructor

A screenshot of a "Instructor Login" form. It includes a label "Instructor ID:" above a text input field containing the value "2", and a "Login" button below it. A vertical red line is drawn to the right of the form area.

## Welcome Instructor!

**Show Available Questions**

**Create Random Exam**

**View Exam Questions**

### Available Questions

ID: 4198

Q: ASP.NET Core is an important topic in its field.

A: TRUE B: FALSE C: Both D: None

Answer: A

ID: 4199

Q: ASP.NET Core is not used in modern applications.

A: TRUE B: FALSE C: Both D: None

Answer: B

ID: 4200

Q: You can master ASP.NET Core without any practice.

A: TRUE B: FALSE C: Both D: None

Answer: B

ID: 4201

Q: ASP.NET Core can be learned with hands-on projects.

A: TRUE B: FALSE C: Both D: None

Answer: A

ID: 4202

Q: ASP.NET Core has no relevance to industry standards.

**Back**

## Create Random Exam

Number of True/False Questions:

Number of MCQ Questions:

Total Exam Grade:

**Create Exam**

**Back**

**Figure 9 : when press create random exam**

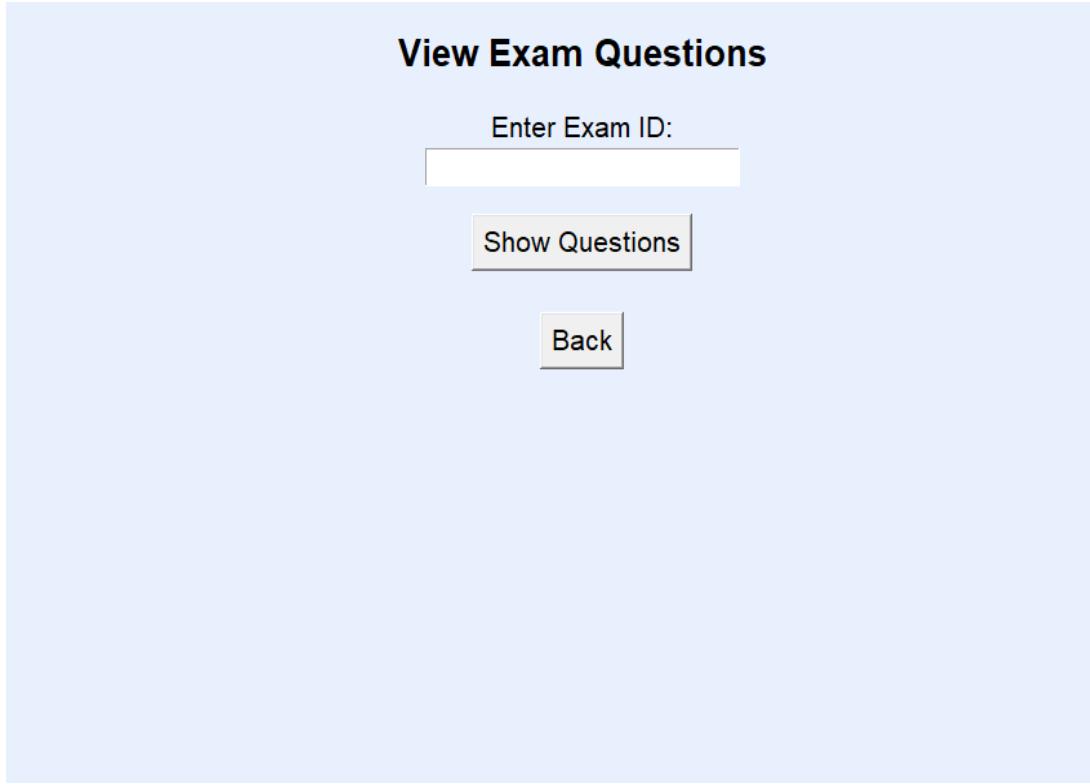


Figure 10 : when press view exam questions to review it