



ITI Examination System

Prepared By

**Omar Adelhady Mostafa Maatouk
Ahmed Hassan Mohamed Hassan
Ziad Mohamed Ali Mohamed
Doaa Mohsen Abdelfattah Mohamed
Ahmed Abdelhamed Hussein Mohamed**

Supervised By

Eng. Eman Gomaa

**INFORMATION TECHNOLOGY INSTITUTE 4-
MONTHS GRADUATION PROJECT**

Table of Contents

1. Project Overview.....	3
2. Database Design.....	4
3. Stored Procedures.....	17
4. Data Warehouse Design.....	23
5. ETL Process in DWH.....	25
6. SSRS Reports	26
8. Power BI Dashboards	33
9. Website	51

1. Project Overview

The **Examination System** is a comprehensive, automated platform developed to manage the full lifecycle of online examinations within an academic institution. Built on the database, the system is designed to streamline exam creation, question management, student participation, automated grading, and detailed reporting.

Key Objectives

- **Automate Exam Processes:** From generating randomized exams to recording and correcting answers, the system minimizes manual intervention and ensures accuracy.
- **Centralized Data Management:** Utilizes a robust SQL Server database to securely store and manage all academic, examination, and operational data.
- **User-Friendly Applications:** Provides both desktop and optional web interfaces for administrators, instructors, and students to interact with the system efficiently.
- **Advanced Reporting & Analytics:** Delivers insightful reports and analytics for students, instructors, and institutional decision-makers.
- **Security & Compliance:** Implements role-based access control, data encryption, and audit logging to protect sensitive information and ensure regulatory compliance.
- **Scalability & Extensibility:** Supports integration with data warehousing and business intelligence tools for future growth and advanced analytics.

Core Features

- **Dynamic Exam Generation:** Randomly selects questions for each exam based on course and predefined criteria, ensuring fairness and variety.
- **Secure Answer Submission:** Allows students to submit answers through a controlled interface, with immediate storage in the database.
- **Automated Grading:** Corrects objective questions automatically and updates student grades, reducing instructor workload and ensuring consistency.
- **Result Management:** Records exam outcomes, calculates percentages, and determines pass/fail status, providing instant feedback.
- **Comprehensive Reporting:** Generates standard and custom reports, including student performance, instructor workloads, course analytics, and graduate outcomes.
- **Backup & Disaster Recovery:** Regularly scheduled backups and documented recovery procedures ensure data integrity and business continuity.

Benefits

- **Operational Efficiency:** Streamlines exam administration, grading, and reporting processes.
- **Data-Driven Insights:** Empowers staff with actionable analytics for academic planning and improvement.
- **Transparency & Fairness:** Ensures objective, consistent grading and feedback for all students.
- **Enhanced Security:** Protects academic data through multi-layered security features.
- **Future-Ready:** Designed for scalability and integration with advanced analytics platforms and reporting tools.

This Examination System provides a modern, reliable, and scalable solution for academic assessment and administration, supporting both immediate operational needs and long-term institutional goals.

2. Database Design

The database design for the Examination System is structured to provide a reliable, scalable, and secure foundation for all system operations. It is implemented using SQL Server and adheres to industry best practices to ensure data integrity, efficient performance, and ease of maintenance^[11].

2.1 Design Principles

- **Normalization:** The schema is normalized to eliminate redundancy and ensure consistency across all tables.
- **Referential Integrity:** Foreign key constraints are used to maintain valid relationships between entities, such as students, courses, exams, and instructors.
- **Clear Naming Conventions:** All tables and fields are named descriptively, making the database self-explanatory and easy to navigate.
- **Security:** Role-based permissions and access controls are enforced to protect sensitive academic and examination data.
- **Documentation:** A comprehensive database dictionary and ERD document the structure, relationships, and purpose of each table and field.

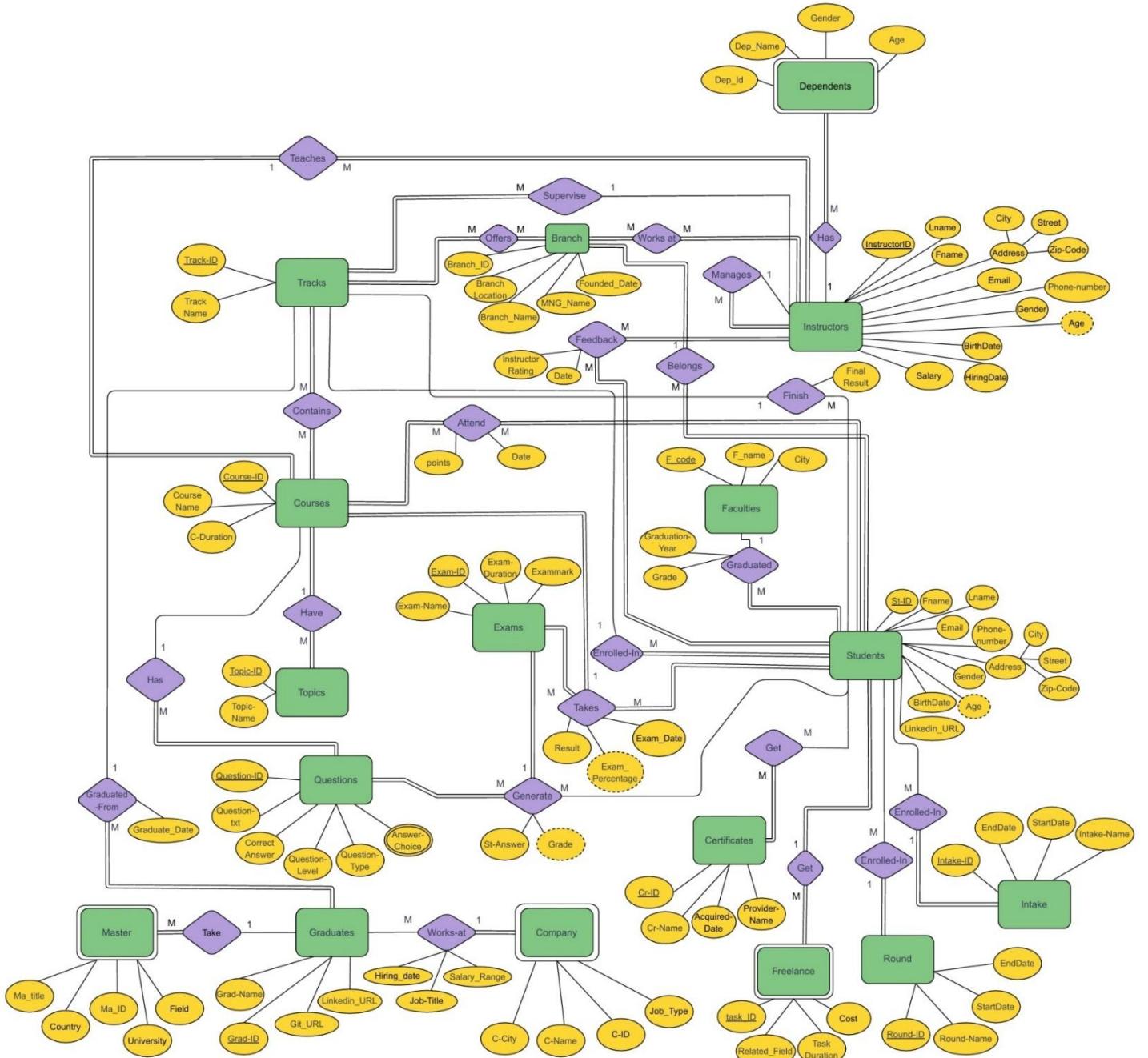
2.2 Entity Relationship Diagram (ERD)

The ERD captures the core entities and their relationships within the system:

- **Student:** Stores student information, including identifiers, names, track, branch, and grades.
- **Instructor:** Contains instructor details such as ID, name, email, and salary.
- **Course:** Catalogues the courses offered, including course ID, name, and duration.
- **Exam:** Details each exam, linking it to a course and specifying its duration and total marks.
- **Question:** Holds exam questions, including text, correct answer, level, and type.
- **Choice:** Stores possible answer choices for each question.
- **Department, Topic, Branch, Track:** Capture academic structure and student assignments.
- **Attendance, Feedback, Graduate, Company, Certificate, Freelance, Master, Round, Intake, Dependents:** Support advanced features such as attendance tracking, graduate outcomes, certifications, and more.

2.3 Table Overview

Table Name	Description	Key Fields
Student	Student records	St_ID, Fname, Lname, Track_ID, Branch_ID, Grade
Instructor	Instructor details	Instructor_ID, Fname, Lname, Email, Salary
Course	Course catalog	Course_ID, Course_Name, C_Duration
Exam	Exam details	Exam_ID, Exam_Name, Exam_Duration, Exam_Mark
Question	Exam questions	Question_ID, Question_Txt, Correct_Answer, Question_Level, Question_Type
Choice	Answer choices	Choice_ID, Question_ID, Answer_Choice
Department	Academic departments	DeptNo, Dept_Name
Topic	Course topics	Topic_ID, Course_ID, Topic_Name
Attendance	Student attendance	Course_ID, St_ID, Date, Degree
Feedback	Student feedback on instructors	St_ID, Instructor_ID, Date, Instructor_Rating
Graduate	Graduate records	Grad_ID, Grad_Name, Git_URL, Linkedin_URL
Company	Graduate employment	Com_ID, Com_Name, C_City, Job_Type
Certificate	Certificates earned	Cer_ID, Cer_Name, Acquired_Date, Provider_Name
Freelance	Freelance tasks by graduates	Task_Title, St_ID, Task_Duration, Cost
Master	Postgraduate studies	Grad_ID, Mas_Title, Country, University, Field
Round	Academic rounds	Round_ID, Round_Name, Start_Date, End_Date
Intake	Student intakes	Intake_ID, Intake_Name, Start_Date, End_Date
Branch	Branch information	Branch_ID, Branch_Name, Branch_Location
Track	Academic tracks	Track_ID, Track_Name, Track_Duration
Dependents	Instructor dependents	Dep_ID, Instructor_ID, Dep_Name, Age, Gender



2.4 Implementation Highlights

- **Stored Procedures:** Key operations such as exam generation, answer submission, grading, and result computation are encapsulated in stored procedures for performance and security.

- **Indexing:** Indexes are applied to primary and foreign keys and frequently queried fields to optimize data retrieval.
- **Backup & Recovery:** Automated backup schedules and disaster recovery procedures are in place to safeguard data.
- **Extensibility:** The schema is designed for future growth, supporting additional features, reporting, and integration with business intelligence tools.

2.5 Design Benefits

This database design ensures that the Examination System can efficiently handle large volumes of data, support complex academic workflows, and provide secure, reliable access to all users. Its structure enables the generation of detailed reports and analytics, empowering ITI staff to make data-driven decisions and maintain high standards of academic integrity.

Database Diagram

Figure 2: ITI System Database Diagram

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



Database Diagram

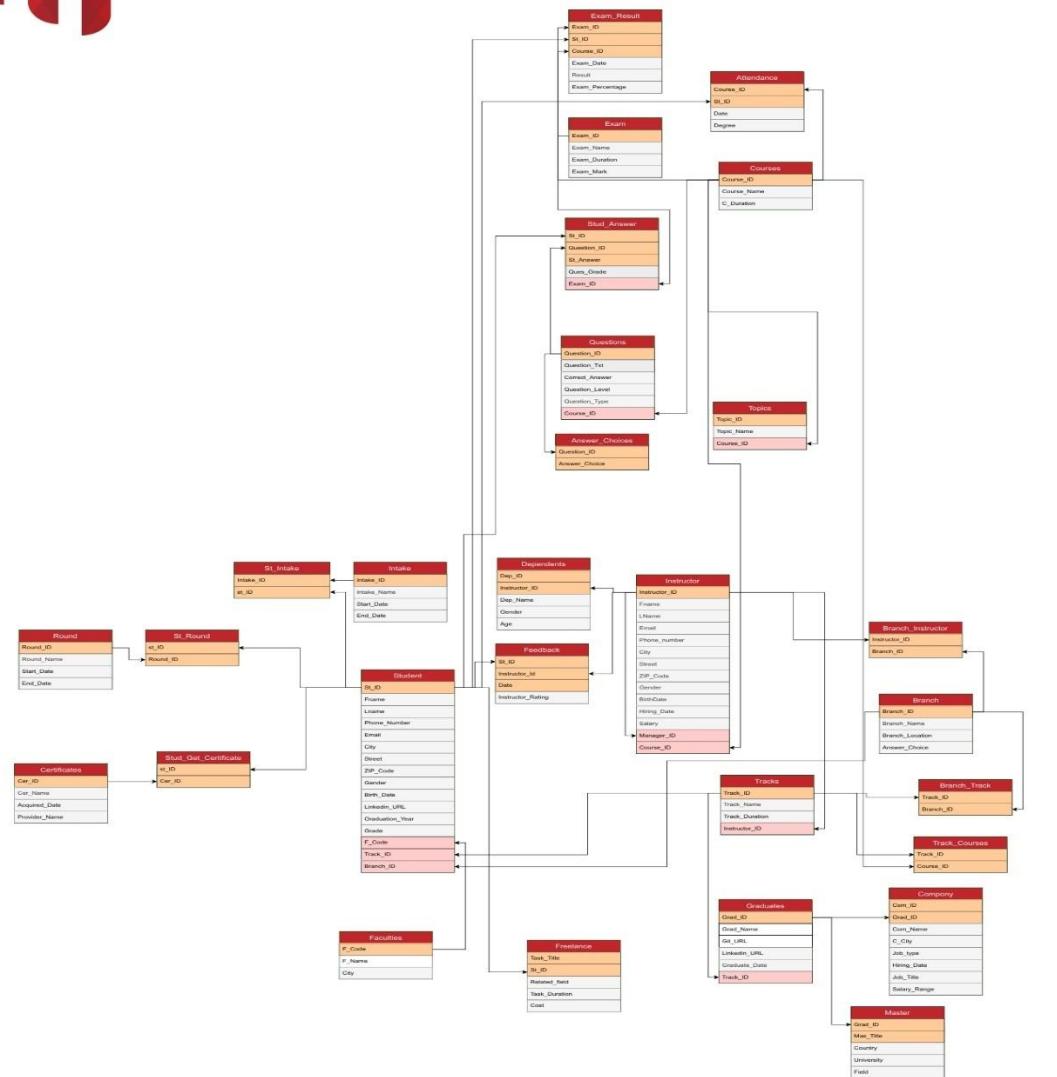


Figure 2 : DATABASE DIAGRAM

3. Stored Procedures

The Examination System database leverages a set of well-structured stored procedures to manage data operations efficiently and securely. These procedures cover standard CRUD (Create, Read, Update, Delete) operations for all main tables, as well as specialized logic to automate exam workflows. Below is a detailed explanation of both standard and special stored procedures, emphasizing their purpose, logic, and significance.

1. Standard CRUD Stored Procedures

For each core table (e.g., Student, Instructor, Course, Exam, Question, Choice, Department, etc.), the following types of stored procedures are implemented:

a. Insert Procedures

- **Purpose:** Add new records to the table.
- **Logic:** Accepts parameters for each field, performs input validation, and inserts the record. Returns the new record's identity value for reference.
- **Example:**

```
CREATE PROCEDURE sp_InsertStudent
    @Fname NVARCHAR(50), @Lname NVARCHAR(50), @Track_ID INT, @Branch_ID INT,
    @Grade DECIMAL(3,2)
AS
BEGIN
    INSERT INTO Student (Fname, Lname, Track_ID, Branch_ID, Grade)
    VALUES (@Fname, @Lname, @Track_ID, @Branch_ID, @Grade);
    SELECT SCOPE_IDENTITY() AS NewStudentID;
END
```

b. Select Procedures

- **Purpose:** Retrieve data, either as full lists or by specific criteria (e.g., by ID, by department).
- **Logic:** Accepts filtering parameters, uses SELECT queries with JOINs where needed, returns requested data.
- **Example:**

```
CREATE PROCEDURE sp_GetStudentByID
    @St_ID INT
AS
```

```
BEGIN  
    SELECT * FROM Student WHERE St_ID = @St_ID;  
END
```

c. Update Procedures

- **Purpose:** Modify existing records.
- **Logic:** Accepts the primary key and fields to update, performs the update, and returns a status or the updated record.
- **Example:**

```
CREATE PROCEDURE sp_UpdateStudentGrade  
    @St_ID INT, @Grade DECIMAL(3,2)  
AS  
BEGIN  
    UPDATE Student SET Grade = @Grade WHERE St_ID = @St_ID;  
    SELECT * FROM Student WHERE St_ID = @St_ID;  
END
```

d. Delete Procedures

- **Purpose:** Remove records from the table.
- **Logic:** Accepts the primary key, deletes the record, and may return a status or affected rows count.
- **Example:**

```
CREATE PROCEDURE sp_DeleteStudent  
    @St_ID INT  
AS  
BEGIN  
    DELETE FROM Student WHERE St_ID = @St_ID;  
    SELECT @@ROWCOUNT AS RowsDeleted;  
END
```

2. Special Stored Procedures

These procedures encapsulate more complex business logic, automating key examination workflows.

2.1 sp_GenerateExamQuestions

- **Purpose:** Randomly selects a specified number of questions for an exam, filtered by course name.
- **Parameters:**
 - @CourseName (partial or full course name)
 - @QuestionCount (number of questions to select; default is 10)
- **Logic:**
 - Joins Questions and Courses tables.
 - Filters questions by course name using a LIKE clause.
 - Uses ORDER BY NEWID() to randomize selection, ensuring each exam is unique.
 - Returns question details: ID, text, correct answer, level, and type.
- **Significance:**
 - Ensures fairness and unpredictability in exams.
 - Supports dynamic exam generation for each student/session.

2.2 sp_InsertStudentAnswer

- **Purpose:** Records a student's answer to a specific question in an exam.
- **Parameters:**
 - @St_ID, @Question_ID, @St_Answer, @Exam_ID
- **Logic:**
 - Inserts the answer into Stud_answers with an initial grade of 0 (pending correction).
 - Returns the new answer's identity value.
- **Significance:**
 - Captures student responses securely and immediately.
 - Prepares data for automated grading.

2.3 sp_CorrectAnswersAndUpdateGrades

- **Purpose:** Grades all answers submitted by a student for a specific exam and calculates performance.

- **Parameters:**
 - @St_ID, @Exam_ID
- **Logic:**
 - Updates Ques_Grade in Stud_answers to 10 if the answer matches the correct answer, 0 otherwise.
 - Counts total questions and correct answers.
 - Calculates the percentage score.
 - Returns total questions, correct answers, and percentage.
- **Significance:**
 - Automates objective grading, ensuring consistency and speed.
 - Provides instant feedback on performance.
 - Reduces instructor workload and human error.

2.4 sp_InsertExamResult

- **Purpose:** Records the final exam result for a student, including pass/fail status.
- **Parameters:**
 - @St_ID, @Exam_ID, @Course_ID, @Exam_Percentage
- **Logic:**
 - Determines pass/fail based on a 60% threshold.
 - Inserts the result into Exam_Result with a timestamp.
 - Returns the percentage and result status.
- **Significance:**
 - Centralizes result management.
 - Enables immediate result communication to students.

2.5 sp_GetAnswerChoices

- **Purpose:** Retrieves all possible answer choices for a given question.
- **Parameters:**
 - @Question_ID

- **Logic:**
 - Selects all choices from `Answer_Choices` for the specified question.
- **Significance:**
 - Supports the exam interface in displaying multiple-choice options.
 - Ensures that students see all valid choices for each question.

3. Summary Table: Key Stored Procedures

Procedure Name	Main Function	Key Tables Affected
<code>sp_Insert[Entity]</code>	Insert new records	All core tables
<code>sp_Select[Entity]</code>	Retrieve records (by ID or criteria)	All core tables
<code>sp_Update[Entity]</code>	Update existing records	All core tables
<code>sp_Delete[Entity]</code>	Delete records	All core tables
<code>sp_GenerateExamQuestions</code>	Random exam question selection	Questions, Courses
<code>sp_InsertStudentAnswer</code>	Record student answers	Stud_answers
<code>sp_CorrectAnswersAndUpdateGrades</code>	Automated grading and performance calculation	Stud_answers, Questions
<code>sp_InsertExamResult</code>	Store exam results and pass/fail status	Exam_Result
<code>sp_GetAnswerChoices</code>	Retrieve answer choices for a question	Answer_Choices

4. Best Practices Reflected

- **Parameterization:** All procedures use parameters to prevent SQL injection and enforce data validation.
- **Transaction Control:** Critical operations (especially updates and inserts) can be wrapped in transactions for atomicity and rollback support.
- **Scalability:** Modular procedures enable easy maintenance and extension.
- **Security:** Role-based execution permissions restrict access to sensitive operations.

These stored procedures form the backbone of the Examination System's data management and exam automation, ensuring reliability, security, and efficiency throughout the academic workflow.

4. 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.

The Data Warehouse component of the Examination System is designed to support advanced analytics, historical tracking, and comprehensive reporting for institutional decision-making. It extends the operational database by integrating, transforming, and storing data in a format optimized for analysis and business intelligence.

4.1 Purpose and Benefits

- **Historical Analysis:** Enables tracking of academic performance, exam results, and student progression over time.
- **Aggregated Reporting:** Supports complex queries and dashboards that would be resource-intensive on the transactional database.
- **Decision Support:** Provides ITI staff, administrators, and instructors with actionable insights for curriculum planning, resource allocation, and student support.
- **Data Consistency:** Ensures a single source of truth for all analytics and institutional reporting.

4.2 Architecture Overview

- **ETL Processes:** Data is extracted from the database, transformed (cleaned, aggregated, and restructured), and loaded into the Data Warehouse. ETL is typically automated using tools like SQL Server Integration Services (SSIS).
- **Star Schema Design:** The warehouse uses a star schema for efficient querying, with fact tables capturing measurable events (e.g., exam results, attendance) and dimension tables providing descriptive context (e.g., student, course, instructor).

- **Batch Updates:** Data is refreshed on a scheduled basis (e.g., nightly) to ensure up-to-date analytics without impacting the performance of the operational system.

4.3 Main Components

Fact Tables

- **Fact_Exam_Result:** Stores exam scores, percentages, pass/fail status, and links to related dimensions (student, exam, course, date).
- **Fact_Student_Finish:** Tracks student completion metrics, graduation status, and related outcomes.
- **Fact_Attendance:** Records attendance events, linking students, courses, and dates.

Dimension Tables

- **Dim_Student:** Contains student attributes (ID, name, track, branch, intake, etc.).
- **Dim_Course:** Details about courses (ID, name, duration, department).
- **Dim_Instructor:** Instructor information (ID, name, email, salary, specialization).
- **Dim_Exam:** Exam metadata (ID, name, duration, total marks).
- **Dim_Branch, Dim_Track, Dim_Graduate, Dim_Feedback, Dim_Date:** Additional context for analytics and drill-down reporting.

4.4 ETL and Data Flow

1. **Extraction:** Data is regularly pulled from the OLTP system, including new exams, student records, answers, and results.
2. **Transformation:** Data is cleaned (duplicates removed, formats standardized), aggregated (e.g., total scores, average grades), and enriched (e.g., joining with dimension data).
3. **Loading:** Transformed data is loaded into the warehouse's fact and dimension tables, ready for analysis.

4.5 Reporting and Analytics

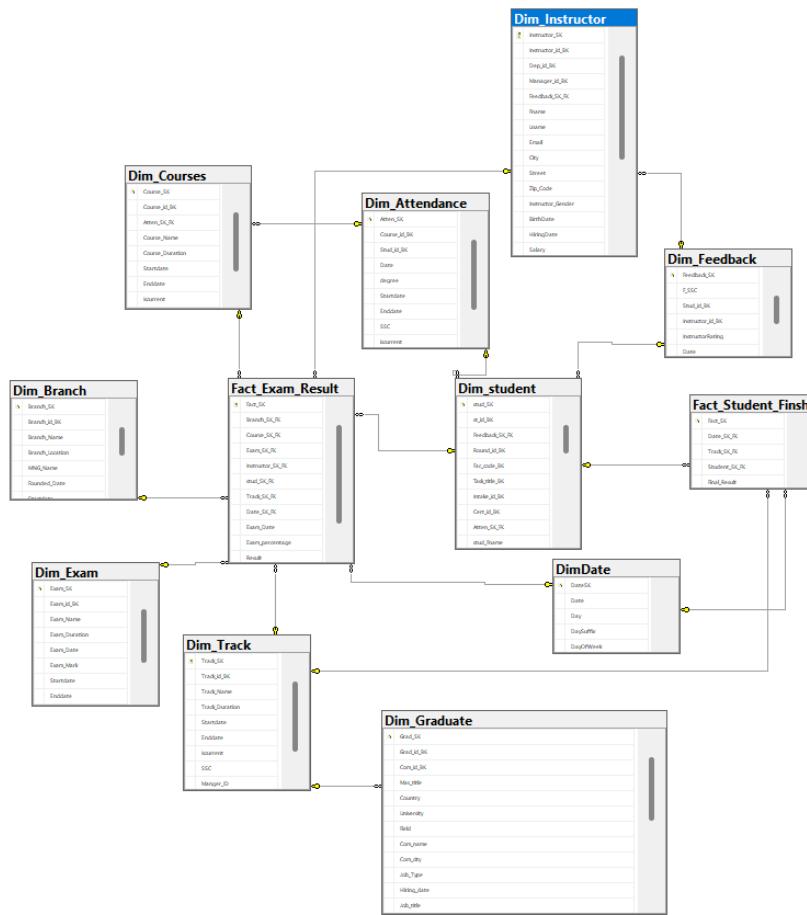
- **Power BI Dashboards:** Visualize trends in student performance, instructor effectiveness, course outcomes, and graduate employment.

- **SSRS/Crystal Reports:** Generate parameterized and ad-hoc reports for academic review, accreditation, and internal audits.
- **Advanced Analytics:** Supports predictive modeling, cohort analysis, and drill-downs by department, course, or instructor.

4.6 Security and Maintenance

- **Access Control:** Only authorized users can access sensitive analytics and reports.
- **Data Refresh Scheduling:** ETL jobs are scheduled during off-peak hours to minimize operational impact.
- **Backup and Recovery:** Regular backups ensure data warehouse integrity and business continuity.

The Data Warehouse is a critical extension of the Examination System, enabling robust analytics and reporting capabilities that drive academic excellence and informed decision-making across the institution.

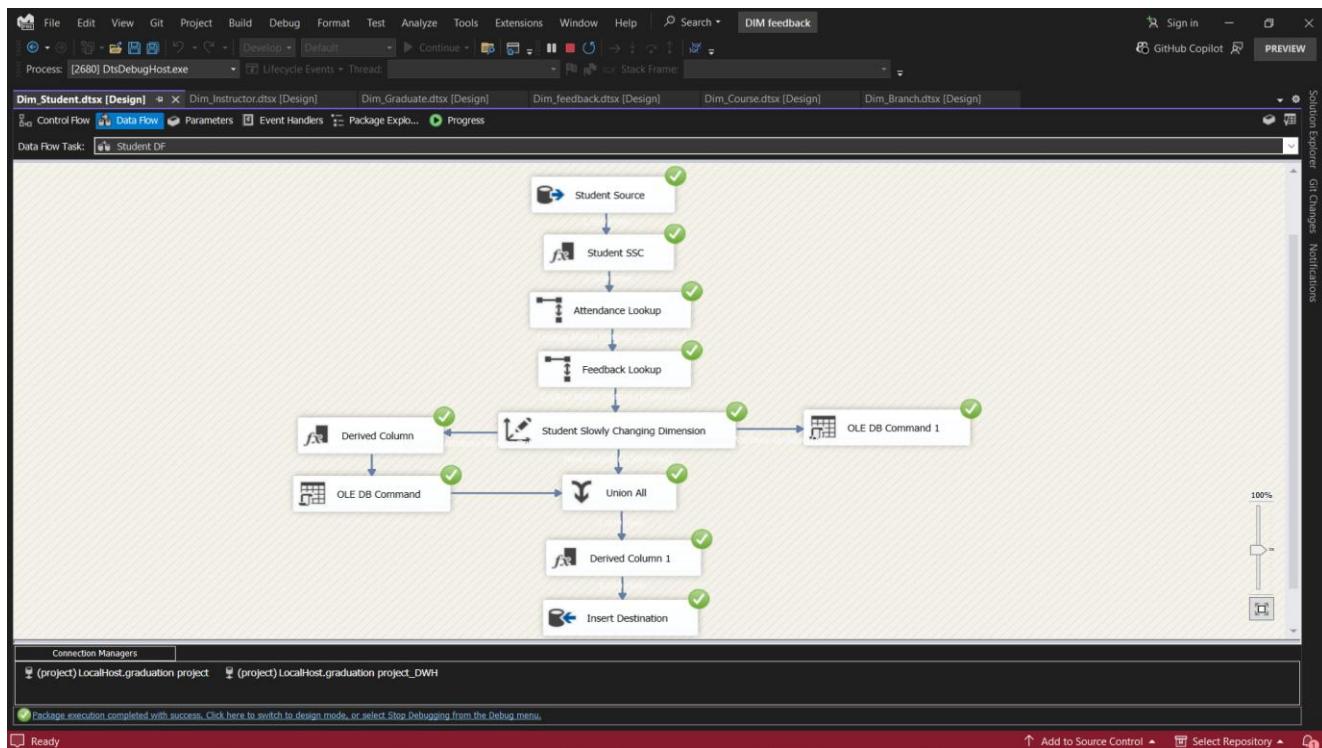


5. ETL Process in DWH

The ETL (Extract, Transform, Load) process is the backbone of the Data Warehouse (DWH) in the Examination System. It ensures that data from the operational database is systematically extracted, transformed, and loaded into the DWH, supporting advanced analytics and reporting. Each SSIS (SQL Server Integration Services) package is designed for a specific dimension or fact table, automating the flow and transformation of data.

Below are detailed explanations for each of the 11 ETL packages, which correspond to the images you will include in your report.

1. Dim_Student



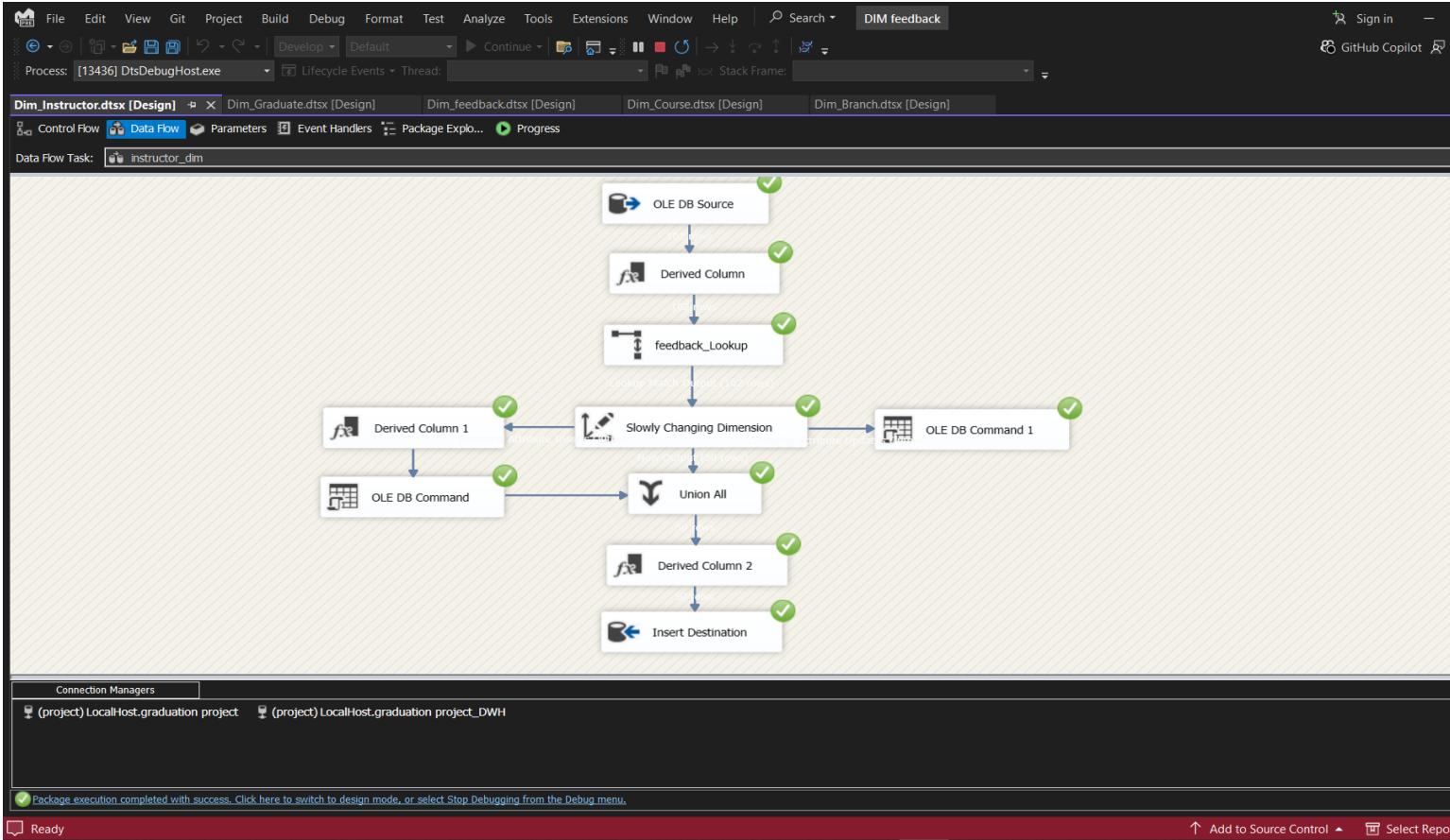
Purpose:

Populates the Student dimension in the DWH with enriched student attributes such as name, track, branch, intake, and graduation status.

Key Steps:

- Extracts student data from the OLTP database.
- Performs lookups to resolve foreign keys (e.g., track, branch).
- Handles Slowly Changing Dimension (SCD) logic to track changes over time.
- Loads the cleaned and enriched data into Dim_Student in the DWH.

2. Dim_Instructor



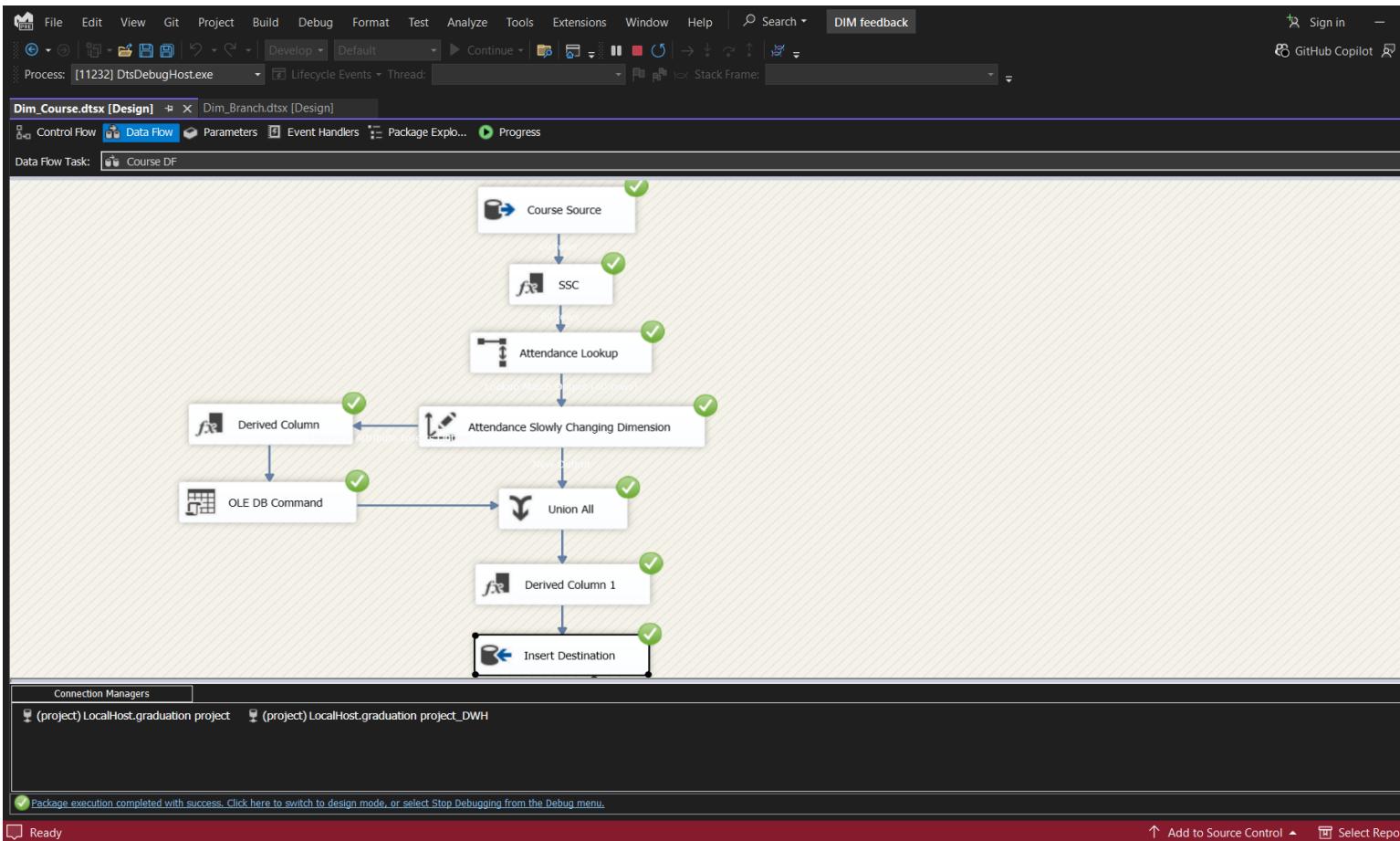
Purpose:

Loads instructor details into the Instructor dimension, capturing attributes like name, email, salary, and specialization.

Key Steps:

- Extracts instructor data from the source.
- Applies data transformations (e.g., formatting names, standardizing emails).
- Resolves department and branch relationships via lookups.
- Loads the data into Dim_Instructor.

3. Dim_Course



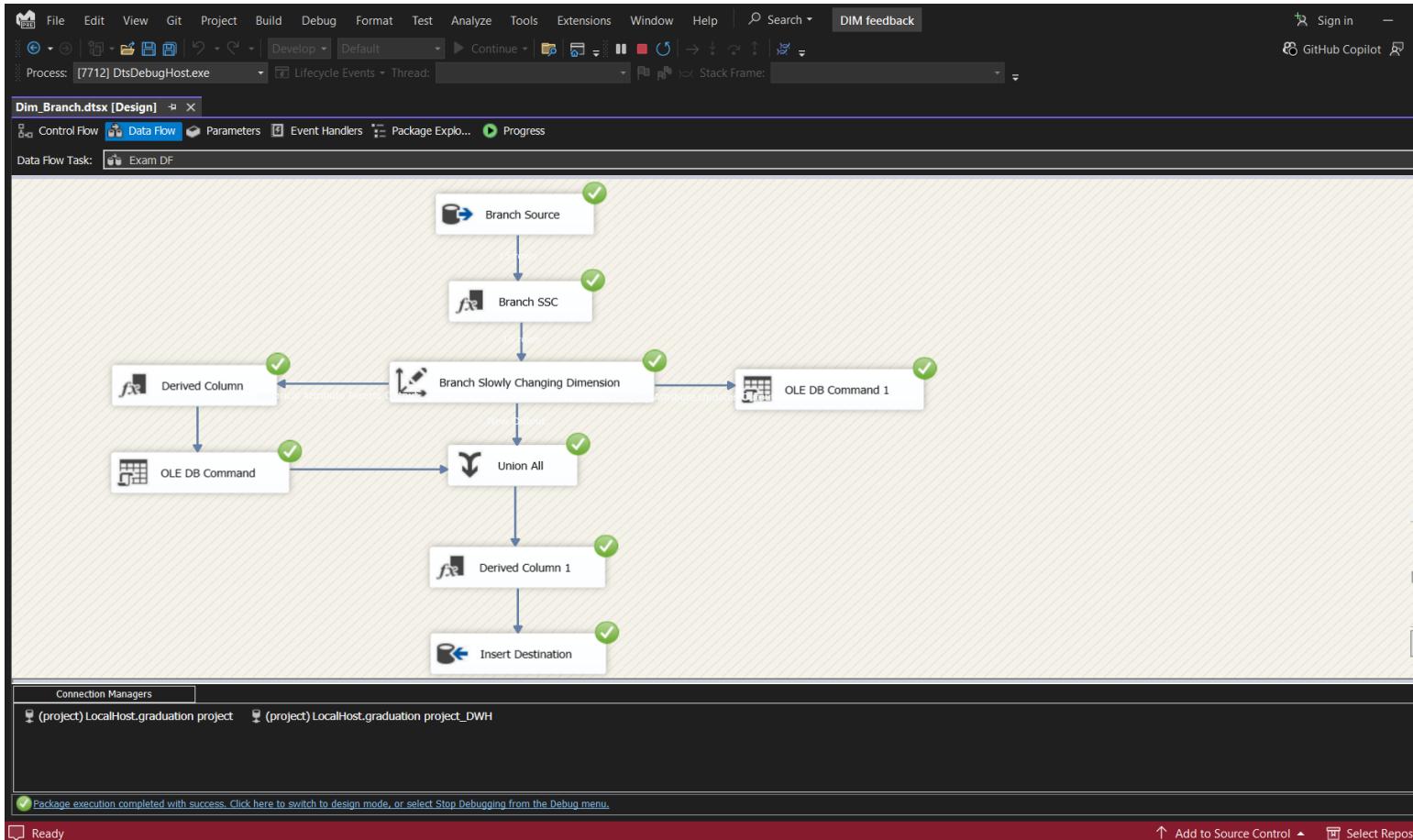
Purpose:

Populates the Course dimension with course metadata, including course name, duration, and department association.

Key Steps:

- Extracts course information from the OLTP system.
- Performs lookups for department and topic associations.
- Cleans and standardizes course names and codes.
- Loads the processed data into Dim_Course.

4. Dim_Branch



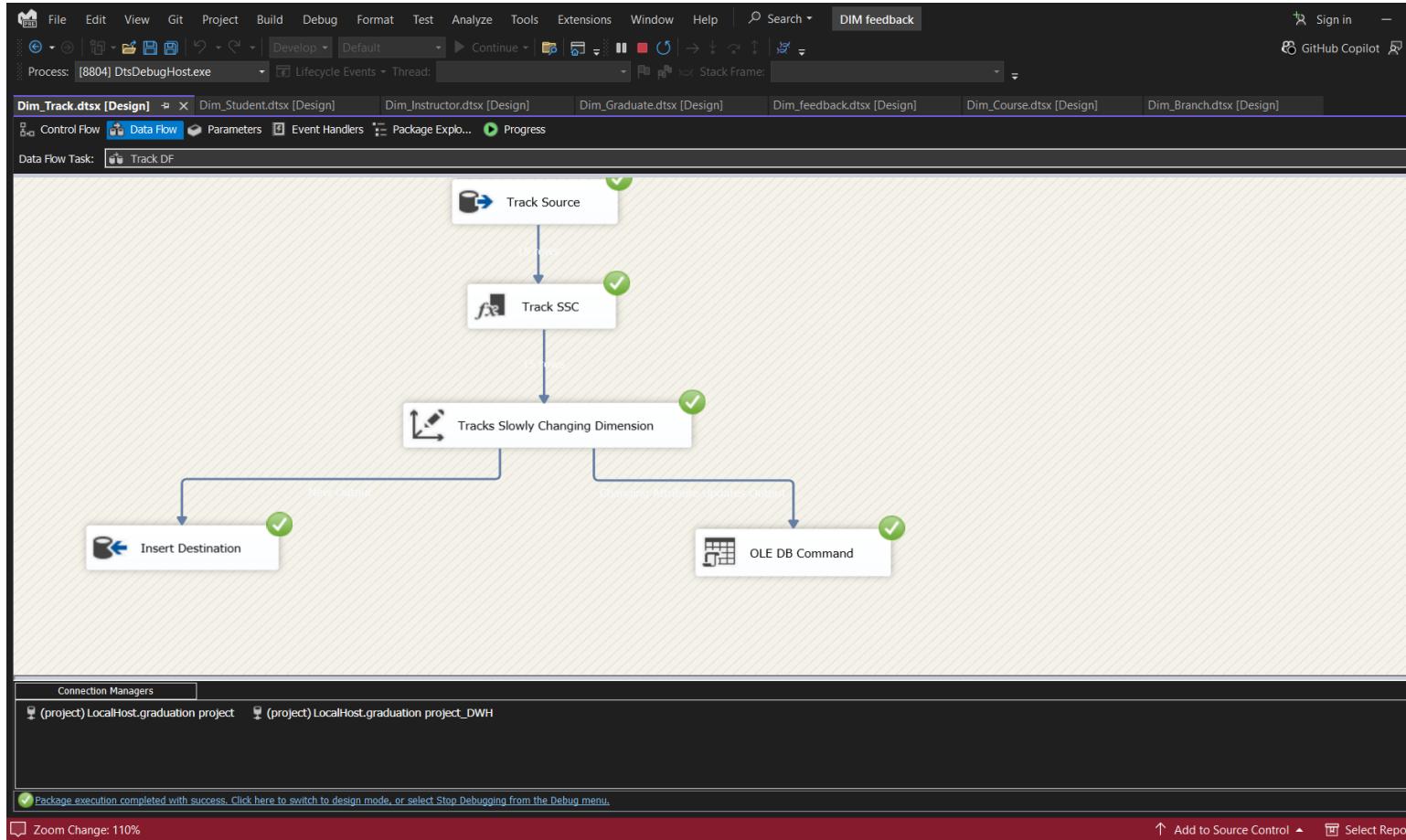
Purpose:

Loads branch information into the DWH, supporting analysis by location or branch.

Key Steps:

- Extracts branch data (name, location) from the source.
- Standardizes location data for consistency.
- Loads the data into **Dim_Branch**.
-

5.Dim_Track



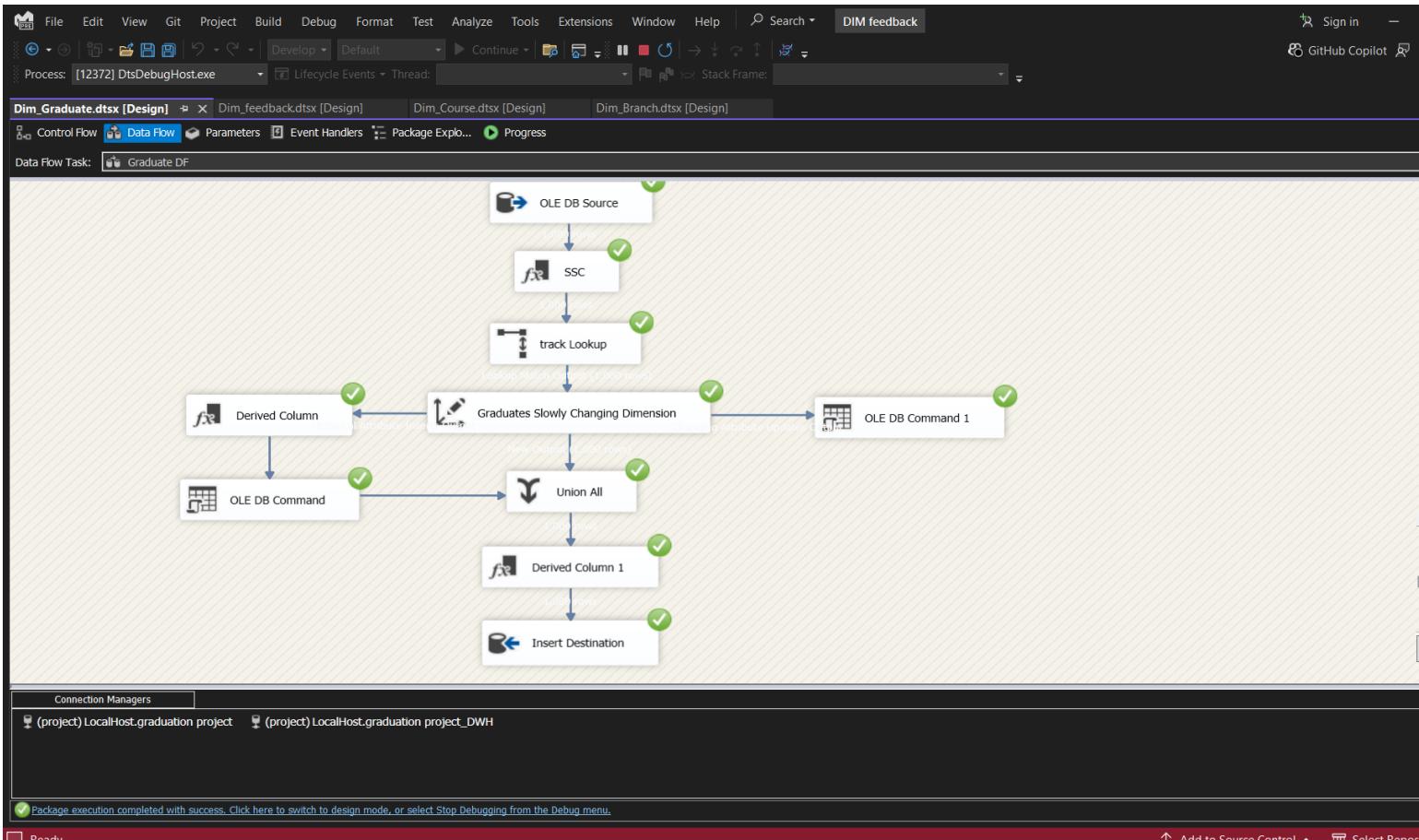
Purpose:

Populates the Track dimension with academic track details.

Key Steps:

- Extracts track names and durations.
- Cleans and formats track data.
- Loads the data into Dim_Track.
-

6.Dim_Graduate



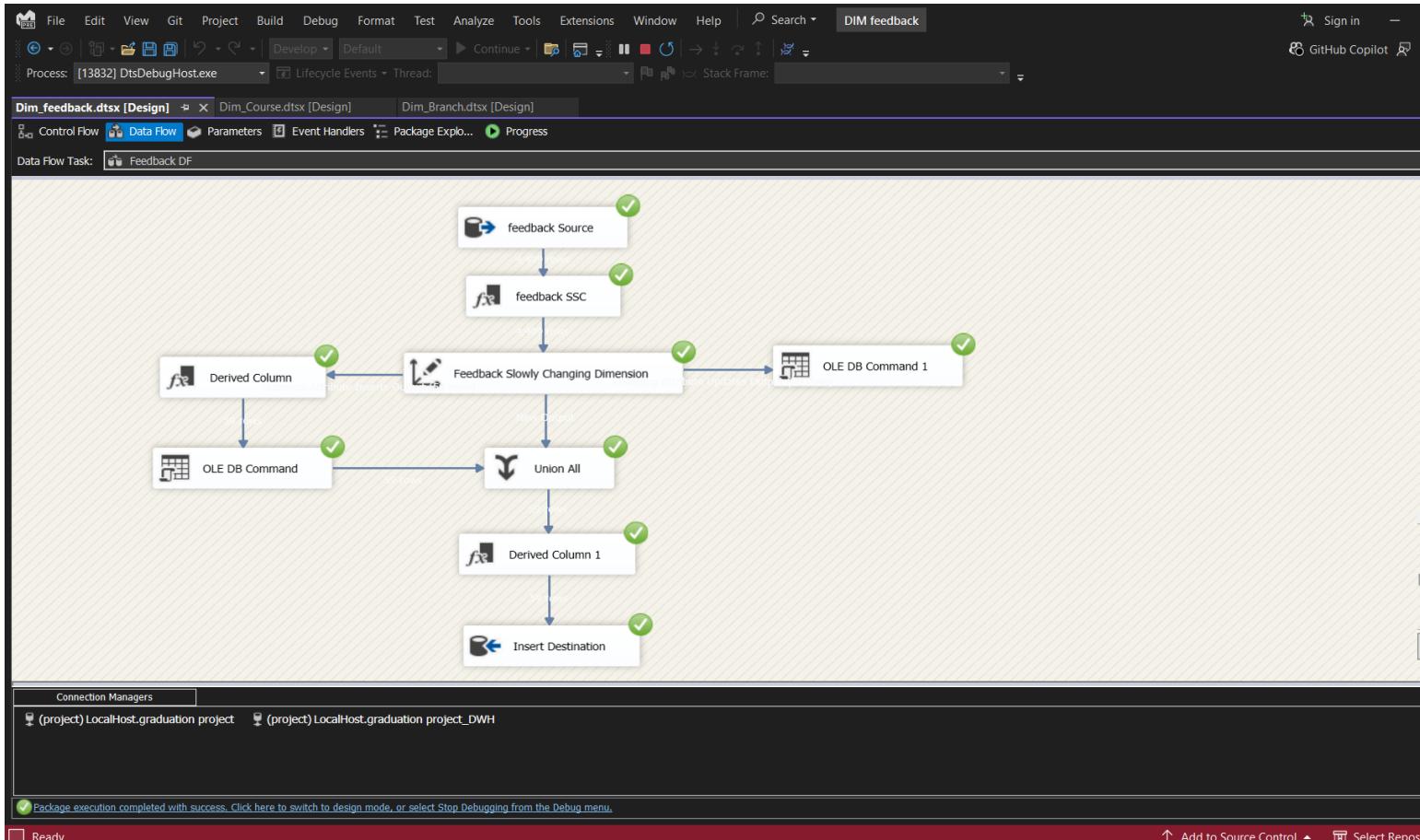
Purpose:

Captures graduate records, including employment and postgraduate details.

Key Steps:

- Extracts graduate data (names, GitHub/LinkedIn URLs).
- Joins with employment and certificate data for enrichment.
- Loads the data into Dim_Graduate.

7.Dim_Feedback



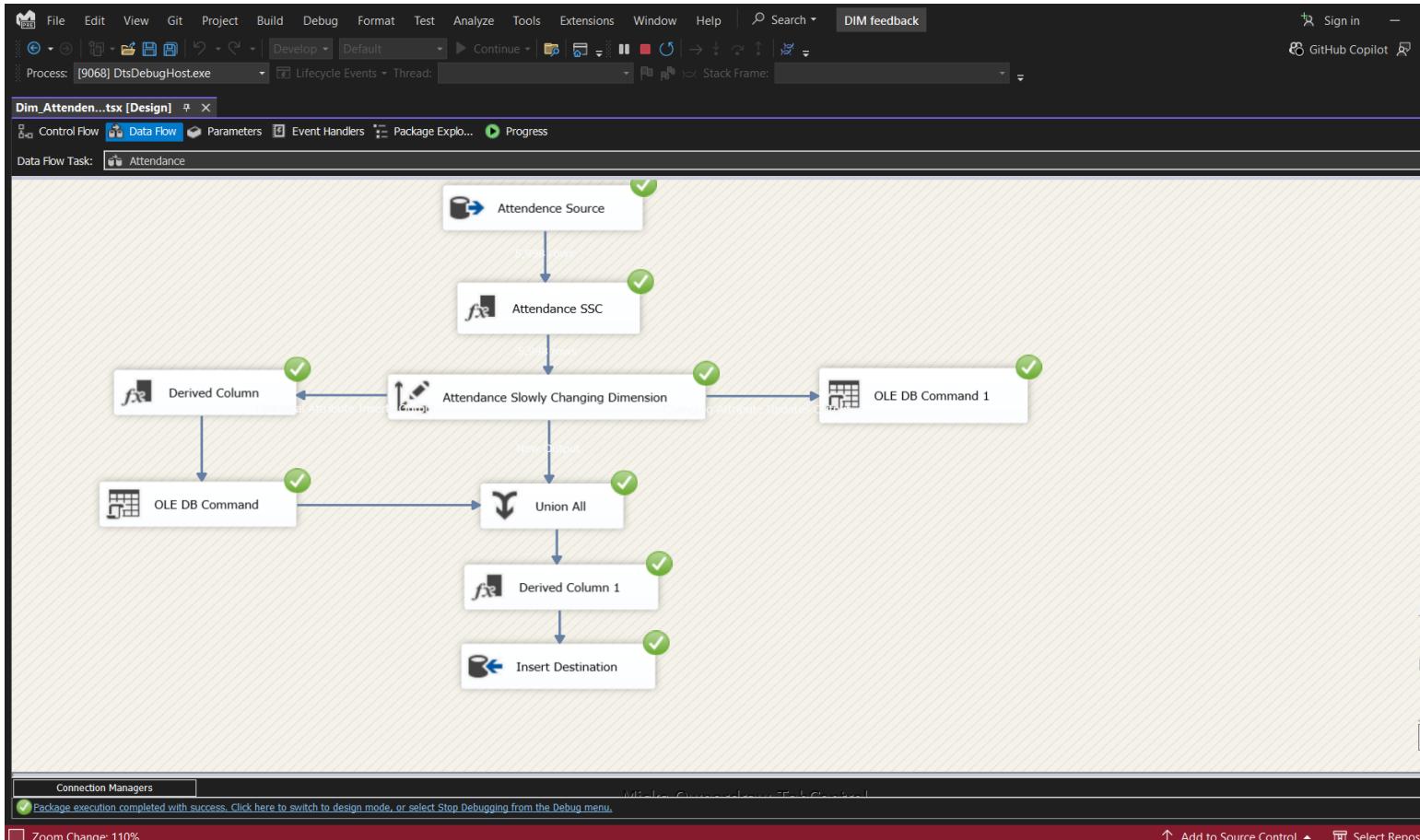
Purpose:

Loads student feedback for instructors, supporting analytics on teaching quality.

Key Steps:

- Extracts feedback records (student ID, instructor ID, rating, date).
- Resolves student and instructor references.
- Loads the data into Dim_Feedback.

8.Dim_Attendance



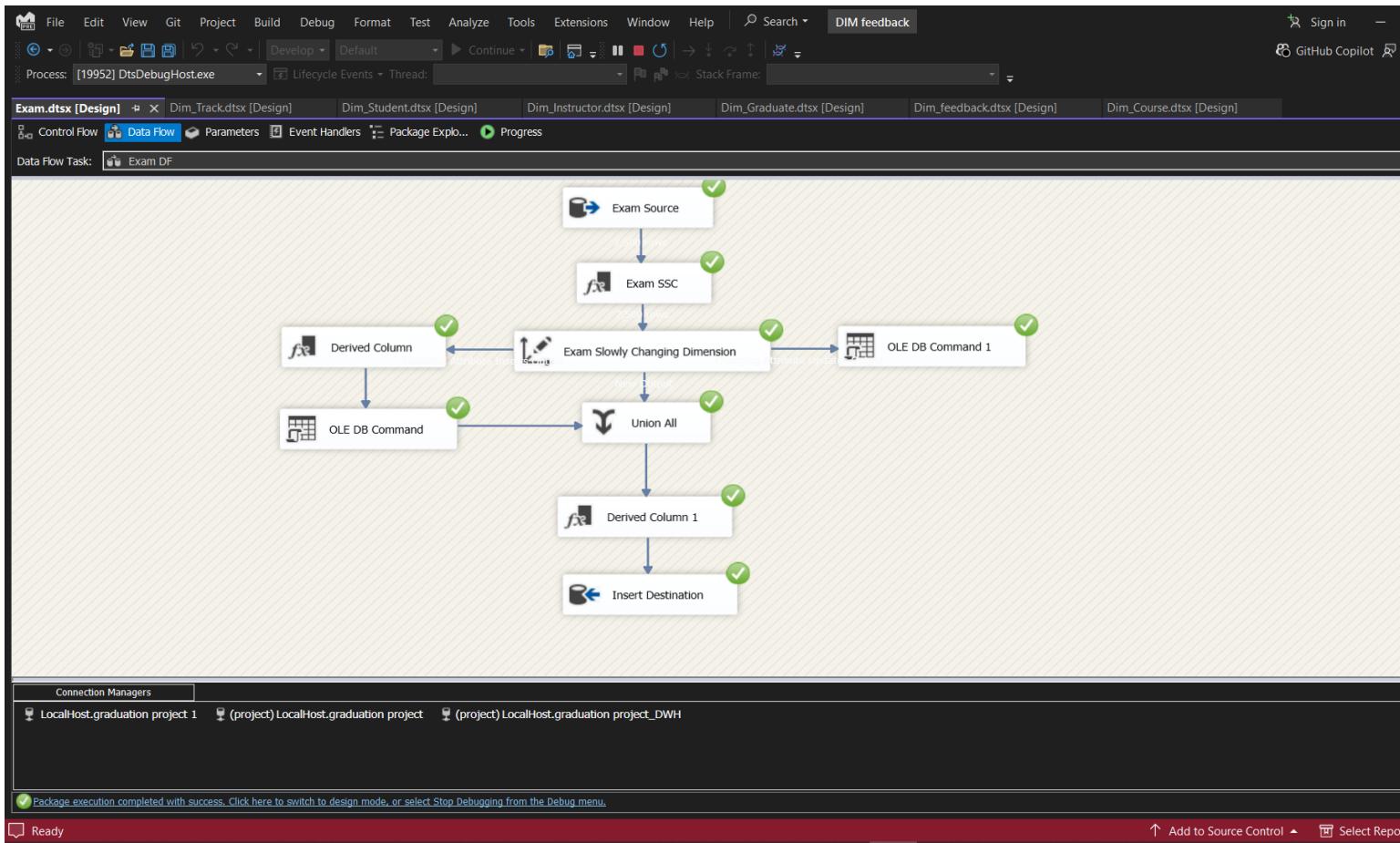
Purpose:

Populates the Attendance dimension for tracking student participation.

Key Steps:

- Extracts attendance events (student, course, date, degree).
- Performs lookups for student and course references.
- Loads the data into Dim_Attendance.

9.Dim_Exam



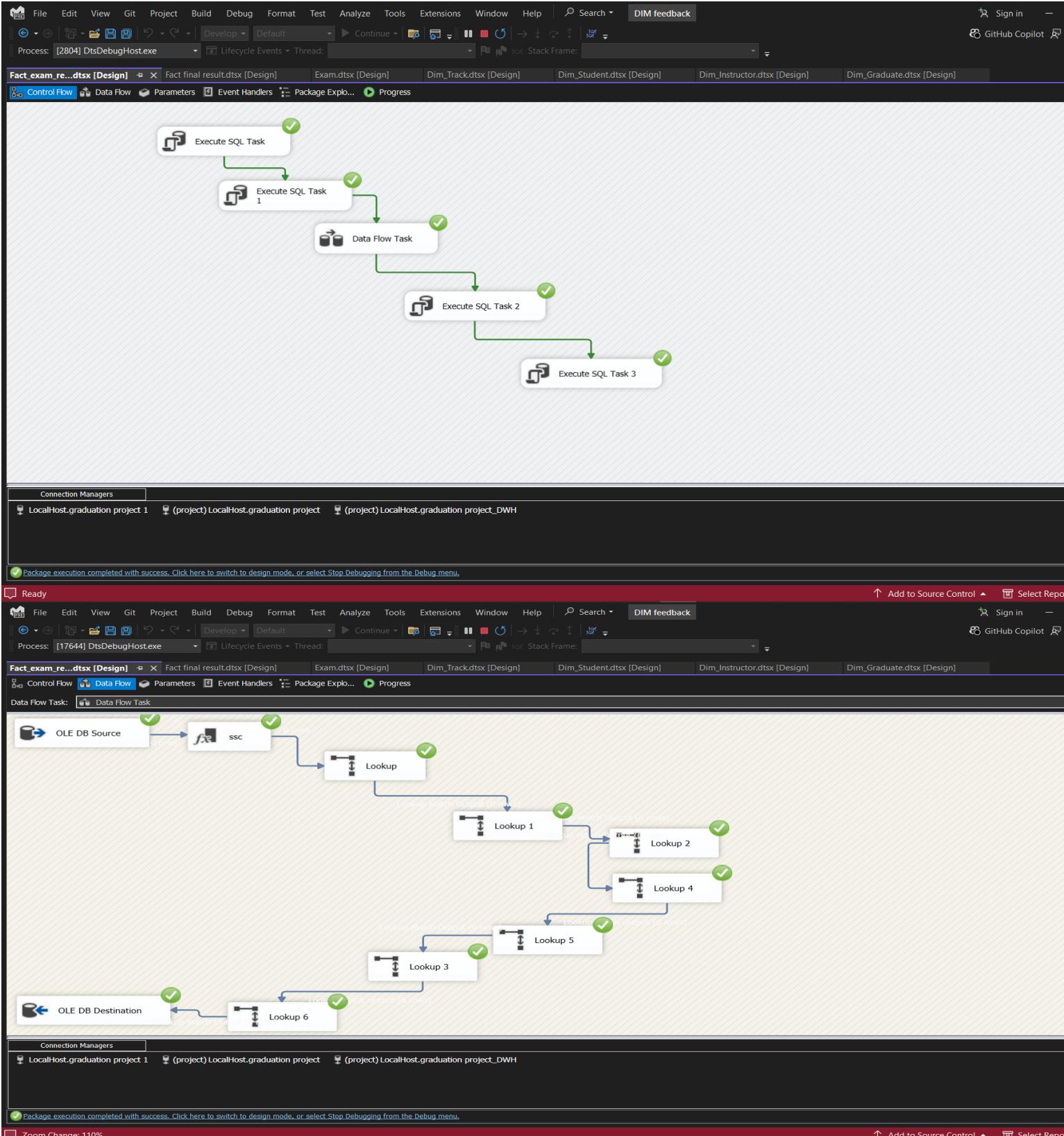
Purpose:

Populates the Exam dimension with exam metadata.

Key Steps:

- Extracts exam data (exam ID, name, duration, total marks).
- Joins with course and instructor data for context.
- Loads the data into Dim_Exam.

10. Fact_Exam_Result



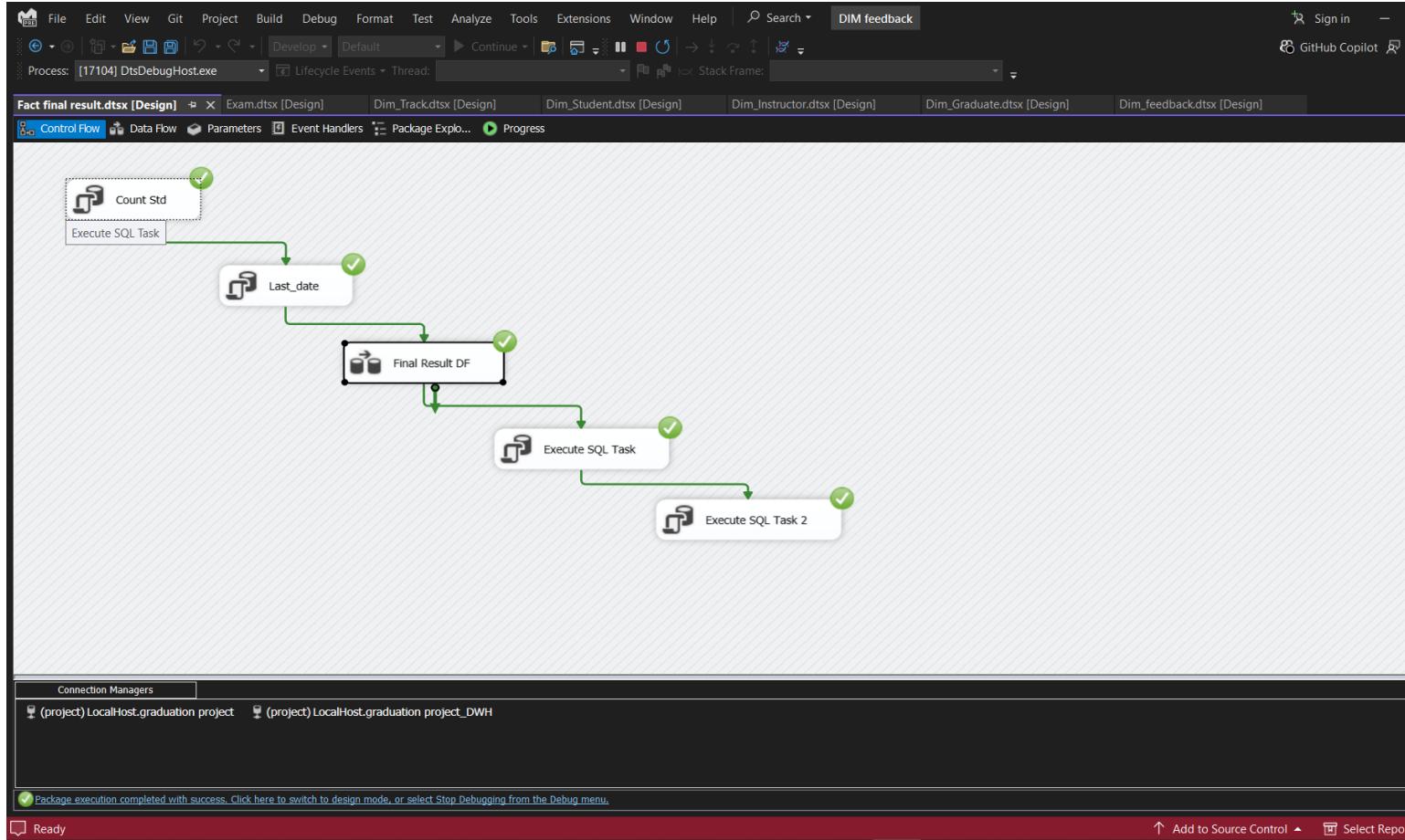
Purpose:

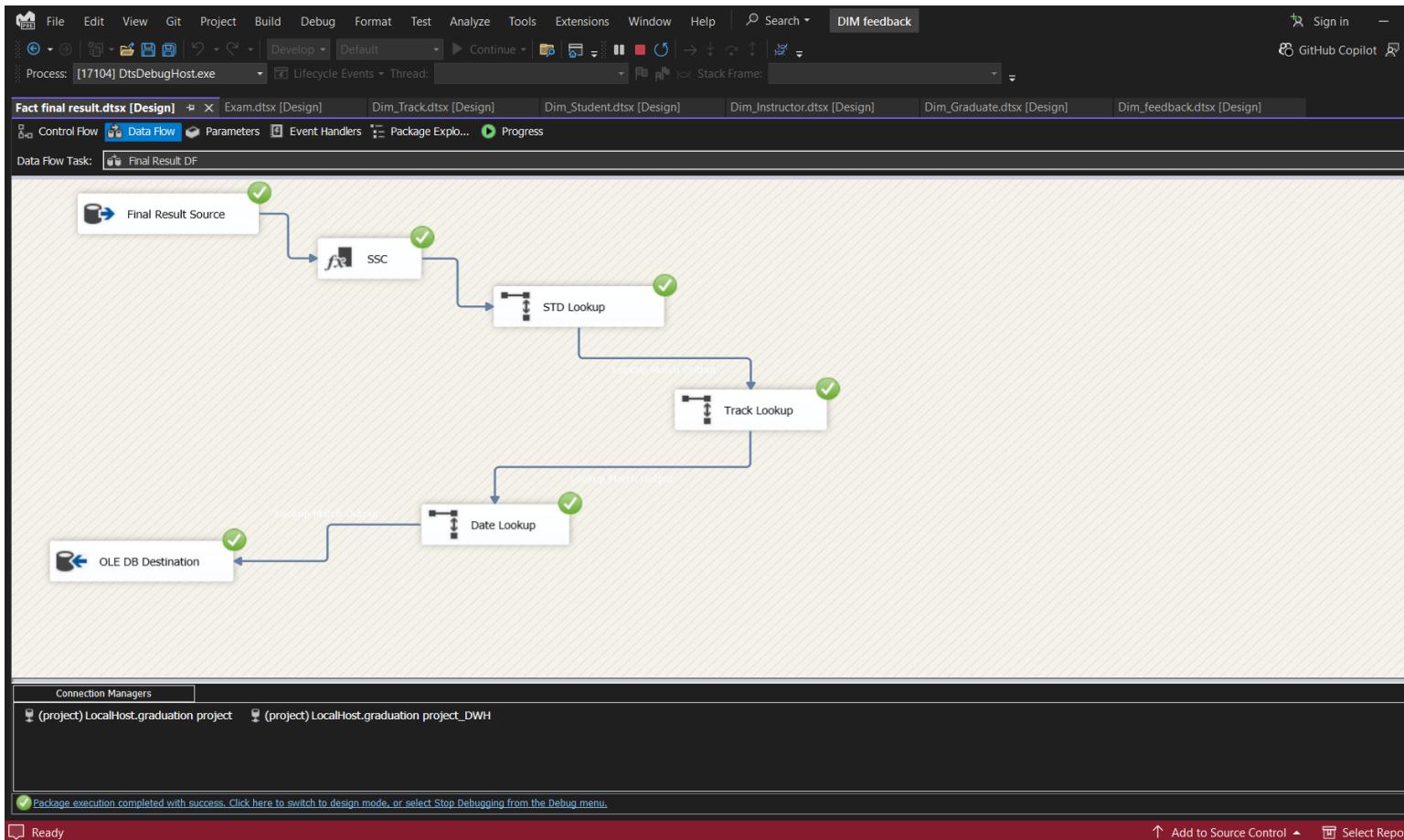
Loads exam results into the fact table, supporting performance analytics.

Key Steps:

- Extracts exam result records (student ID, exam ID, score, percentage, pass/fail).
- Resolves dimension keys (student, exam, course).
- Loads the data into Fact_Exam_Result.

11. Fact_Final_Result





Purpose:

Captures final results for students, enabling graduation and outcome analysis.

Key Steps:

- Extracts final grades and completion status.
- Aggregates results across courses and exams.
- Loads the data into Fact_Final_Result.

ETL Process Summary Table

Package Name	Main Function	Data Target
Dim_Student.dtsx	Student dimension population	DWH.Dim_Student
Dim_Instructor.dtsx	Instructor dimension population	DWH.Dim_Instructor
Dim_Course.dtsx	Course dimension population	DWH.Dim_Course
Dim_Branch.dtsx	Branch dimension population	DWH.Dim_Branch
Dim_Track.dtsx	Track dimension population	DWH.Dim_Track

Dim_Graduate.dtsx	Graduate dimension population	DWH.Dim_Graduate
Dim_Feedback.dtsx	Feedback dimension population	DWH.Dim_Feedback
Dim_Attendance.dtsx	Attendance dimension population	DWH.Dim_Attendance
Dim_Exam.dtsx	Exam dimension population	DWH.Dim_Exam
Fact_Exam_Result.dtsx	Exam results fact table population	DWH.Fact_Exam_Result
Fact_Final_Result.dtsx	Final results fact table population	DWH.Fact_Final_Result

Each ETL package is designed to ensure data quality, consistency, and readiness for business intelligence and reporting. The images you include in your report should visually illustrate the control flow, data flow, and key transformations described above, providing clear documentation of your ETL architecture and implementation.

6. SSRS Reports

The Examination System leverages SQL Server Reporting Services (SSRS) to deliver dynamic, parameterized, and visually rich reports for academic management and decision-making. These reports are directly powered by stored procedures and data warehouse views, ensuring real-time, reliable insights for ITI staff, instructors, and students.

7.1 Overview of SSRS Reports

SSRS is used to create, deploy, and manage a suite of reports that address the core operational and analytical needs of the institution. Reports are designed to be interactive, exportable (PDF, Excel, etc.), and accessible through a web portal or integrated application interface.

7.2 Key Reports Implemented

1. Students Information by Department

- Purpose:** Lists all students in a selected department.
- Parameters:** Department Number.
- Features:** Displays student ID, full name, contact details, address, gender, birth date, LinkedIn URL, graduation year, grade, and department information.
- Sample Output:**



Students Information by Department:10

Student ID	Full Name	Phone number	Email	City	Street	Zip Code	Gender	Birth Date	LinkedIn URL	Graduation Year	Grade	Department No	Department Name
27	Youssif Hassan	01246766086	youssef.hassan27@example.com	Fayoum	Street 15	98734	Male	05/04/2000	https://linkedin.com/in/youssef.hassan27	2019	B	10	Software Testing
31	Khaled Mostafa	01537961832	khaled.mostafa31@example.com	Giza	Street 10	16857	Male	24/02/2002	https://linkedin.com/in/khaled.mostafa31	2020	D	10	Software Testing
33	Nour Elsayed	01033401407	nour.elsayed33@example.com	Zagazig	Street 16	66774	Female	28/06/1999	https://linkedin.com/in/nour.elsayed33	2021	C	10	Software Testing
42	Aya Elsayed	01512377197	aya.elsayed42@example.com	Zagazig	Street 40	51411	Female	13/12/2003	https://linkedin.com/in/aya.elsayed42	2018	D	10	Software Testing
62	Mostafa Elsayed	01142759980	mostafa.elsayed62@example.com	Tanta	Street 19	45420	Male	21/05/1995	https://linkedin.com/in/mostafa.elsayed62	2023	A	10	Software Testing
93	Laila Hassan	01162647829	laila.hassan93@example.com	Giza	Street 3	16717	Female	26/07/2004	https://linkedin.com/in/laila.hassan93	2022	C	10	Software Testing
110	Sara Youssef	01276399665	sara.youssef110@example.com	Cairo	Street 23	41229	Female	02/07/2004	https://linkedin.com/in/sara.youssef110	2019	B	10	Software Testing
112	Hassan Ali	01090006541	hassan.ali112@example.com	Mansoura	Street 39	59526	Male	04/12/1998	https://linkedin.com/in/hassan.ali112	2020	D	10	Software Testing
117	Khaled Mahmoud	01072882217	khaled.mahmoud117@example.com	Tanta	Street 40	54764	Male	26/04/2004	https://linkedin.com/in/khaled.mahmoud117	2023	A	10	Software Testing
122	Ahmed Mostafa	01238897687	ahmed.mostafa122@example.com	Damietta	Street 30	54422	Male	08/05/2004	https://linkedin.com/in/ahmed.mostafa122	2023	C	10	Software Testing
126	Sara Mahmoud	01522235343	sara.mahmoud126@example.com	Damietta	Street 14	43731	Female	23/01/1999	https://linkedin.com/in/sara.mahmoud126	2023	C	10	Software Testing
176	All Mahmoud	01599748099	all.mahmoud176@example.com	Zagazig	Street 49	59584	Male	24/07/2000	https://linkedin.com/in/all.mahmoud176	2023	C	10	Software Testing
186	Omar Youssef	01025768872	omar.youssef186@example.com	Asyut	Street 6	27479	Male	16/02/1999	https://linkedin.com/in/omar.youssef186	2021	C	10	Software Testing
217	Aya Ashraf	01239040411	aya.ashraf217@example.com	Asyut	Street 34	77097	Female	06/08/1998	https://linkedin.com/in/aya.ashraf217	2024	B	10	Software Testing
231	Khaled Ashraf	01566291783	khaled.ashraf231@example.com	Asyut	Street 49	43186	Male	10/09/1996	https://linkedin.com/in/khaled.ashraf231	2019	D	10	Software Testing
251	Nour Ashraf	01072133074	nour.ashraf251@example.com	Damietta	Street 35	42366	Female	22/12/1996	https://linkedin.com/in/nour.ashraf251	2020	C	10	Software Testing
259	Mohamed Elsayed	01546579661	mohamed.elsayed259@example.com	Giza	Street 46	12228	Male	24/01/2000	https://linkedin.com/in/mohamed.elsayed259	2022	B	10	Software Testing
278	Yasmin Hassan	01280734225	yasmin.hassan278@example.com	Damietta	Street 6	93611	Female	28/09/2005	https://linkedin.com/in/yasmin.hassan278	2024	B	10	Software Testing
287	Mohamed Ashraf	0107412717	mohamed.ashraf287@example.com	Asyut	Street 20	29726	Male	05/01/1998	https://linkedin.com/in/mohamed.ashraf287	2018	D	10	Software Testing
288	Ibrahim Hassan	01264261208	ibrahim.hassan288@example.com	Giza	Street 37	89584	Male	25/12/2004	https://linkedin.com/in/ibrahim.hassan288	2020	A	10	Software Testing
296	All Mostafa	01231759702	all.mostafa296@example.com	Tanta	Street 12	77778	Male	17/01/1995	https://linkedin.com/in/all.mostafa296	2021	C	10	Software Testing
323	Huda Abd elrahman	01088295047	huda.abd.elrahman323@example.com	Mansoura	Street 37	80360	Female	23/11/2003	https://linkedin.com/in/huda.abd.elrahman323	2018	D	10	Software Testing
327	Yasmin Ali	01260023616	yasmin.ali327@example.com	Tanta	Street 19	70015	Female	15/10/1996	https://linkedin.com/in/yasmin.ali327	2018	D	10	Software Testing

2. Student Grades in All Courses

- Purpose:** Shows the grades of a specific student across all enrolled courses.
- Parameters:** Student ID.
- Features:** Displays course names and grade percentages for the student.

- **Sample Output:**



Grades of the student : Ahmed Tawfik (ID : 100)

Course Name	Grade Percentage
Game Design Basics	83.21%
Selenium WebDriver	64.41%
AI Ethics	94.82%
Smart Contracts	73.96%
AWS Basics	89.14%

3. Instructor's Courses and Student Count

- **Purpose:** Lists all courses taught by a particular instructor and the number of students enrolled in each.
- **Parameters:** Instructor ID.
- **Features:** Facilitates workload analysis and resource planning.
- **Sample Output:**



Courses taught by instructor : Mai Adel (ID: 50)

Course Name	Number Of Students
Networking Basics	105

4. Course Topics

- **Purpose:** Displays all topics covered within a selected course.
- **Parameters:** Course ID.
- **Features:** Useful for curriculum review and planning.

Sample Output:



Topics for course ID: 50

Course Name:
Networking Basics

Topic Name
TCP/IP Model
IP Addressing
Subnetting
Routing

5. Exam Questions and Choices

- **Purpose:** Retrieves all questions and corresponding answer choices for a specific exam.
- **Parameters:** Exam Number.
- **Features:** Supports exam review and quality assurance.
- **Sample Output:**



Exam Questions

* The technology used to distribute service requests to resources is referred to as

load balancing

load performing

load scheduling

* Which of the following is a more sophisticated load balancer?

rackserve managers

workload managers

workspace managers

* Which of the following is another name for the system virtual machine?

hardware virtual machine

real machine

software virtual machine

* Which of the following is Type 1 Hypervisor?

KVM

LynxSecure

Wind River Simics

6. Exam Questions with Student Answers

- **Purpose:** Shows all questions in a specific exam along with the answers given by a particular student and the grade for each answer.
- **Parameters:** Exam Number, Student ID.
- **Features:** Enables detailed review of student performance and grading transparency.
- **Sample Output:**



Answers of student: Yasmin Ashraf (ID: 1) in Exam #3

Question Text	Student Answer	Question Grade
Network Security offers clear explanations and examples.	False	0.00
Network Security requires understanding of fundamental principles.	True	0.00
Network Security is designed for beginners and professionals.	True	10.00
Network Security requires understanding of fundamental principles.	False	10.00
Network Security offers clear explanations and examples.	True	0.00
Network Security is essential for understanding key concepts.	True	10.00
Network Security focuses on real-life applications.	False	10.00
Network Security covers both theory and practice.	False	10.00
Network Security does not include any advanced topics.	False	10.00
Network Security requires understanding of fundamental principles.	False	10.00

7.3 SSRS Report Features

- **Parameterization:** Users can filter reports by department, student, instructor, course, or exam.
- **Interactivity:** Drill-down and drill-through capabilities allow users to explore data in detail.
- **Export Options:** Reports can be exported to PDF, Excel, Word, and other formats for sharing and archiving.
- **Security:** Access to reports is controlled based on user roles, ensuring data privacy and compliance.

7.4 Integration with Data Warehouse

- **Live Data:** Reports are powered by up-to-date data from the operational database and the data warehouse.
- **Historical Analysis:** Some reports leverage fact and dimension tables in the DWH for trend and cohort analysis.
- **Performance:** SSRS is optimized for efficient querying and rendering, even with large datasets.

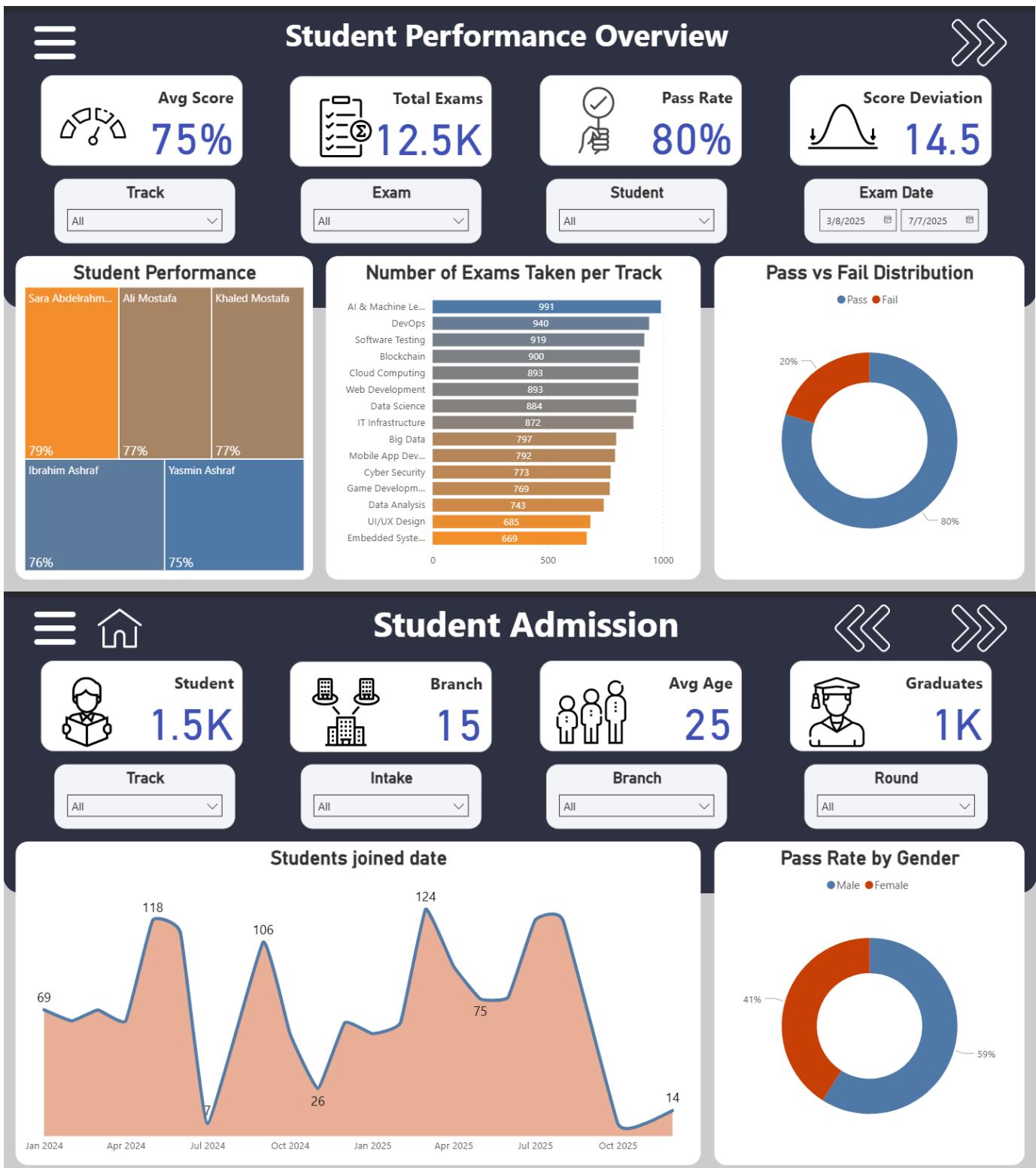
7.5 Benefits

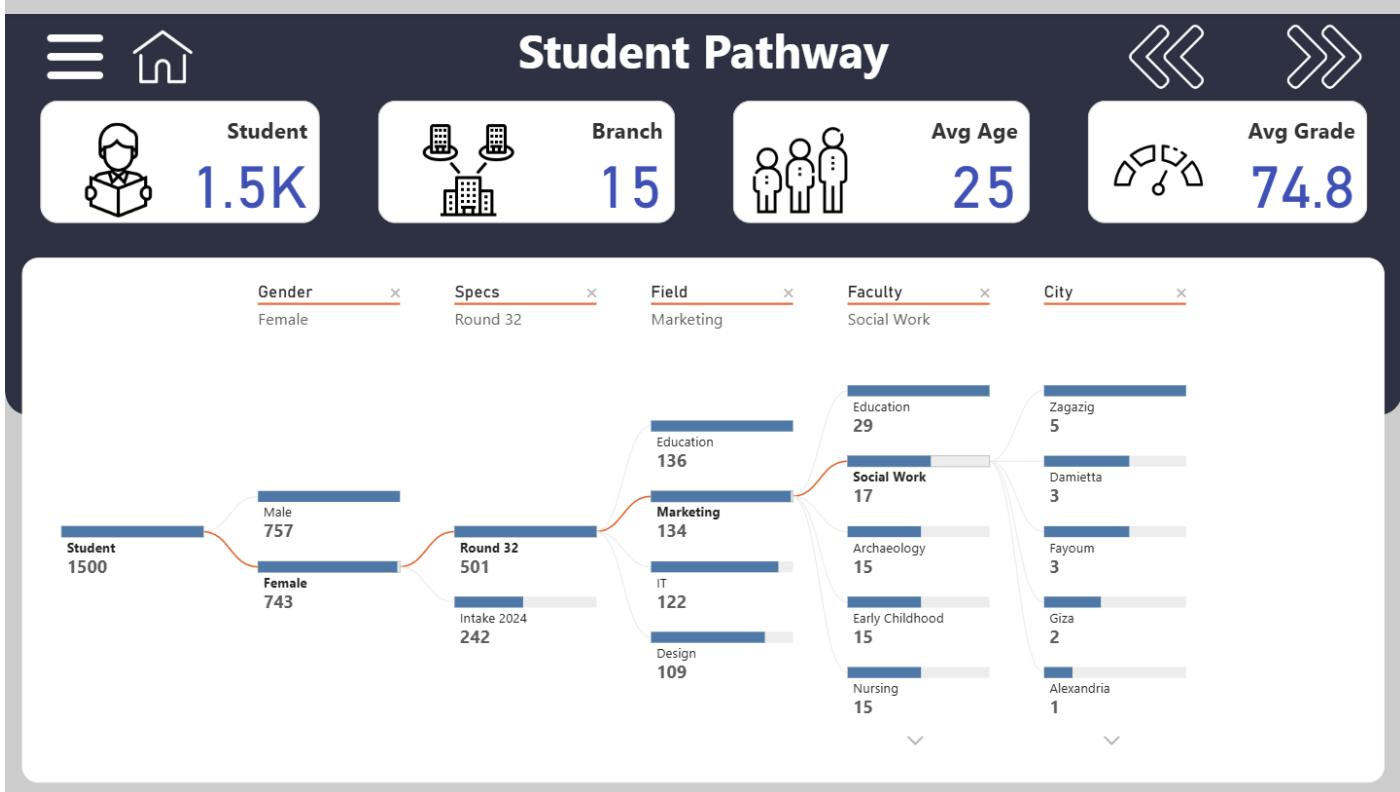
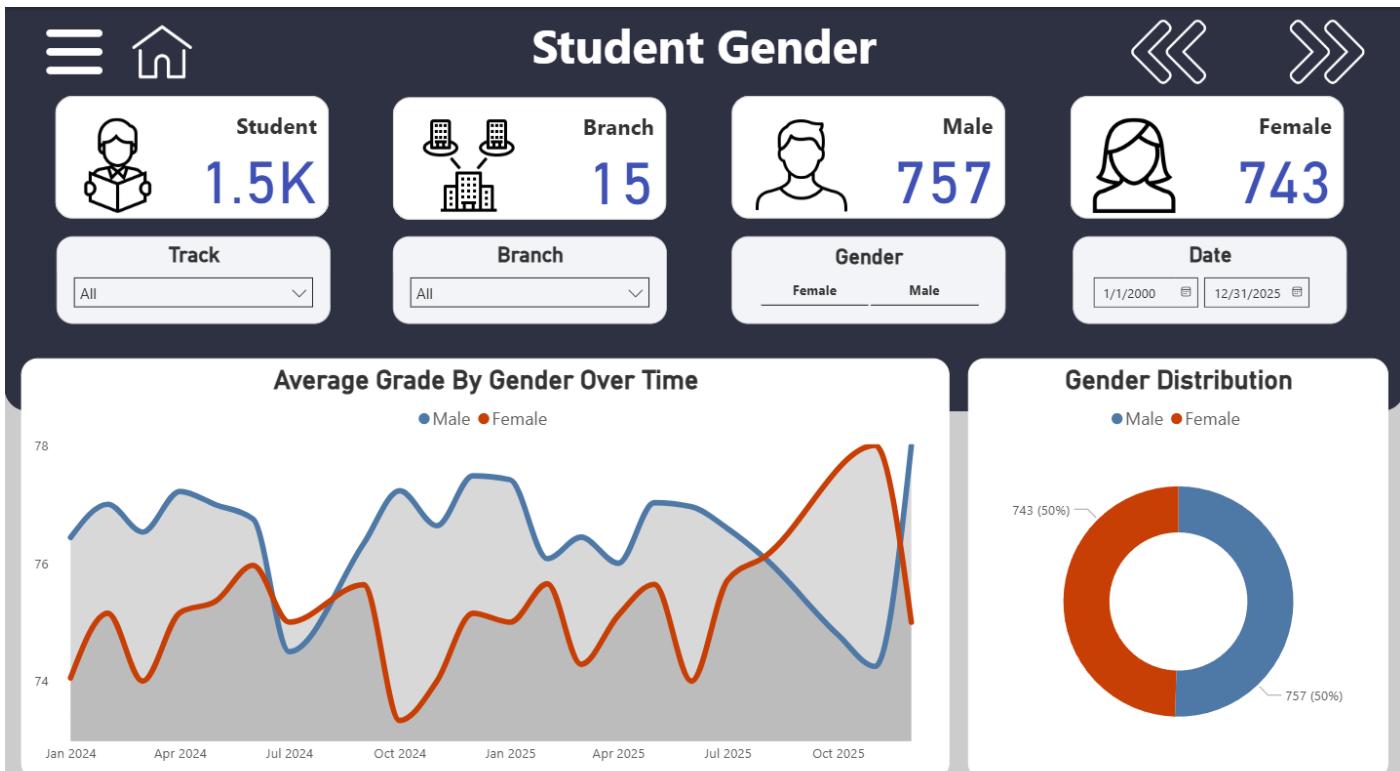
- **Informed Decision-Making:** Provides ITI staff and instructors with actionable insights.
- **Transparency:** Enhances trust by making grading and academic records accessible and auditable.
- **Efficiency:** Reduces manual reporting workload and eliminates errors.
- **Scalability:** Easily accommodates new report types and data sources as the system grows.

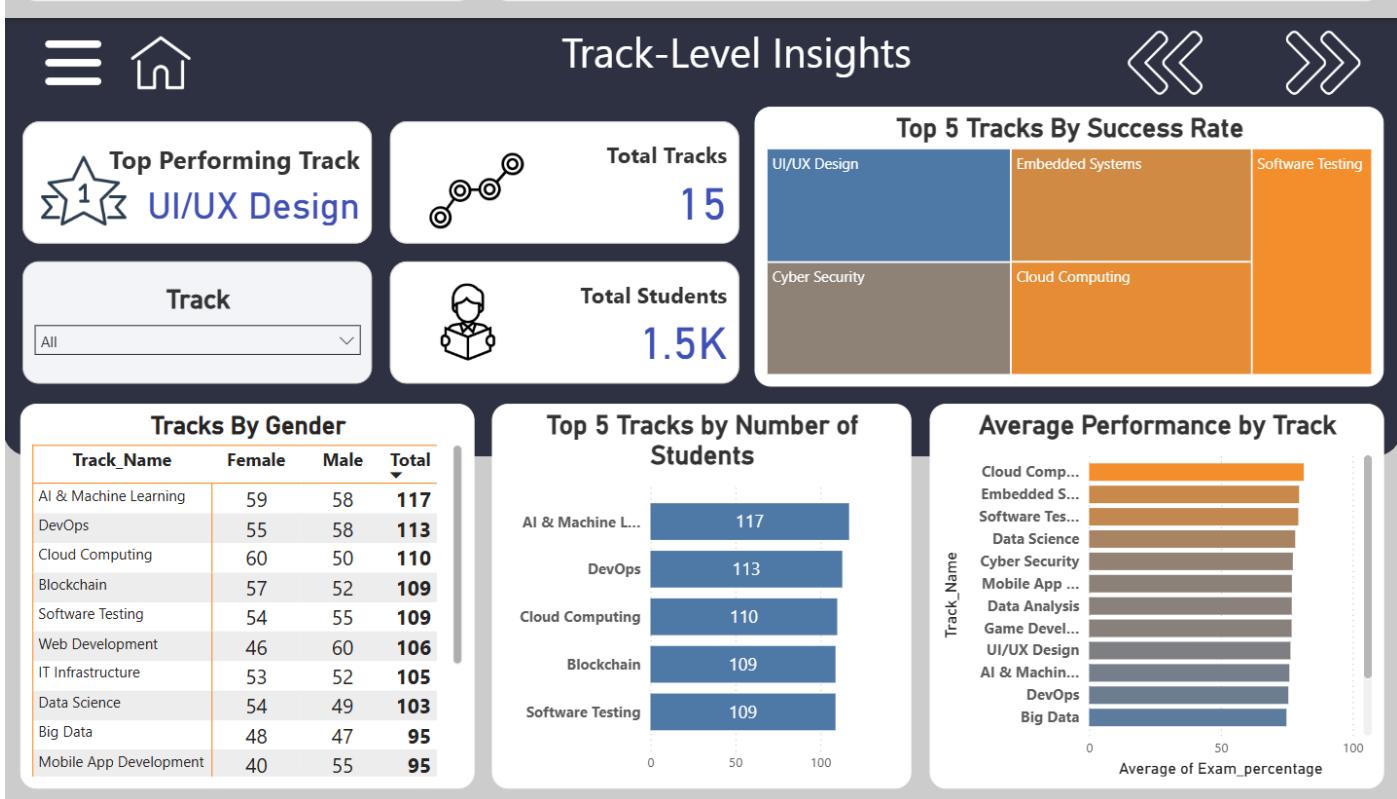
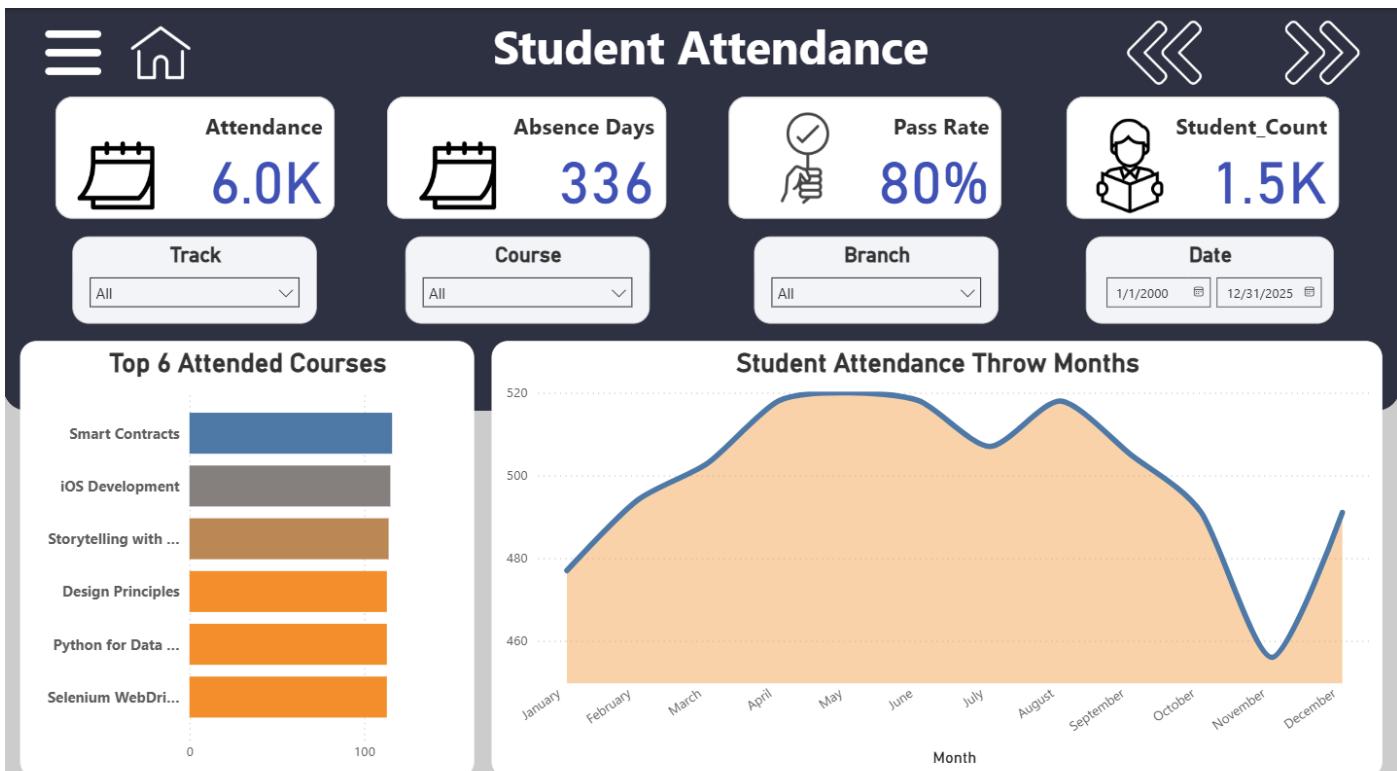
These SSRS reports are central to the Examination System's value, transforming raw academic data into meaningful, actionable information for all stakeholders.

7. Power BI Dashboards

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

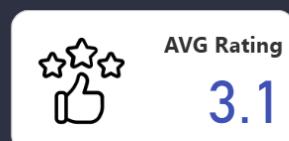








Instructor Dashboard



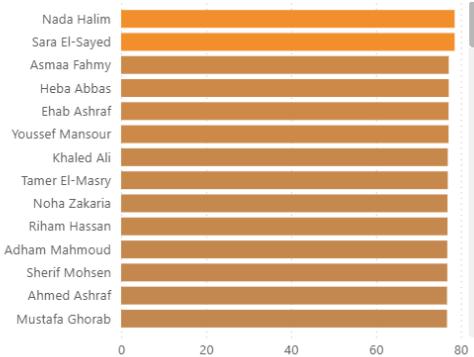
Track

Instructor

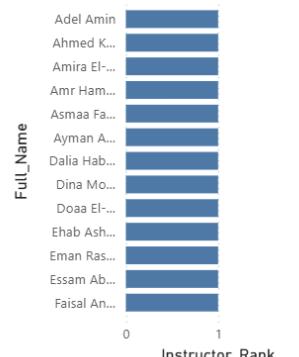
Salary

Gender

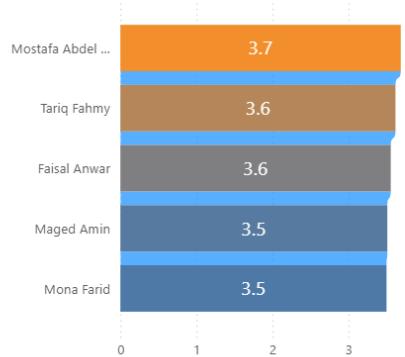
AVG Student Performance Per Instructor



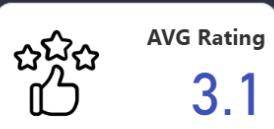
Instructor Rank



Top 5 Instructor in Rating



Instructor Growth



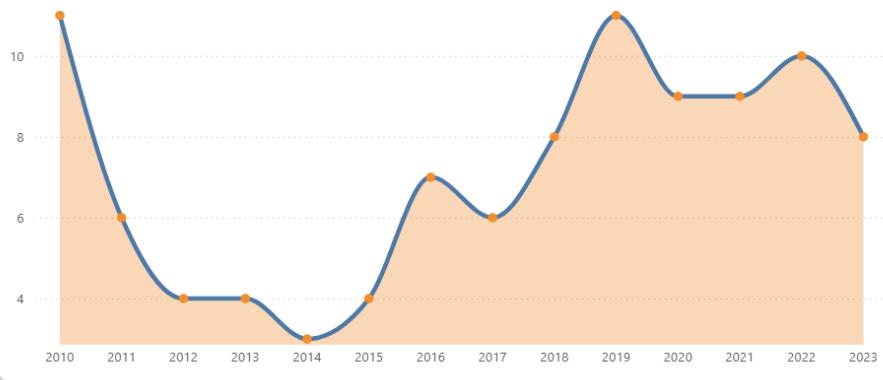
Track

Gender

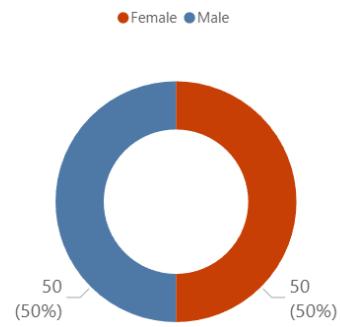
Branch

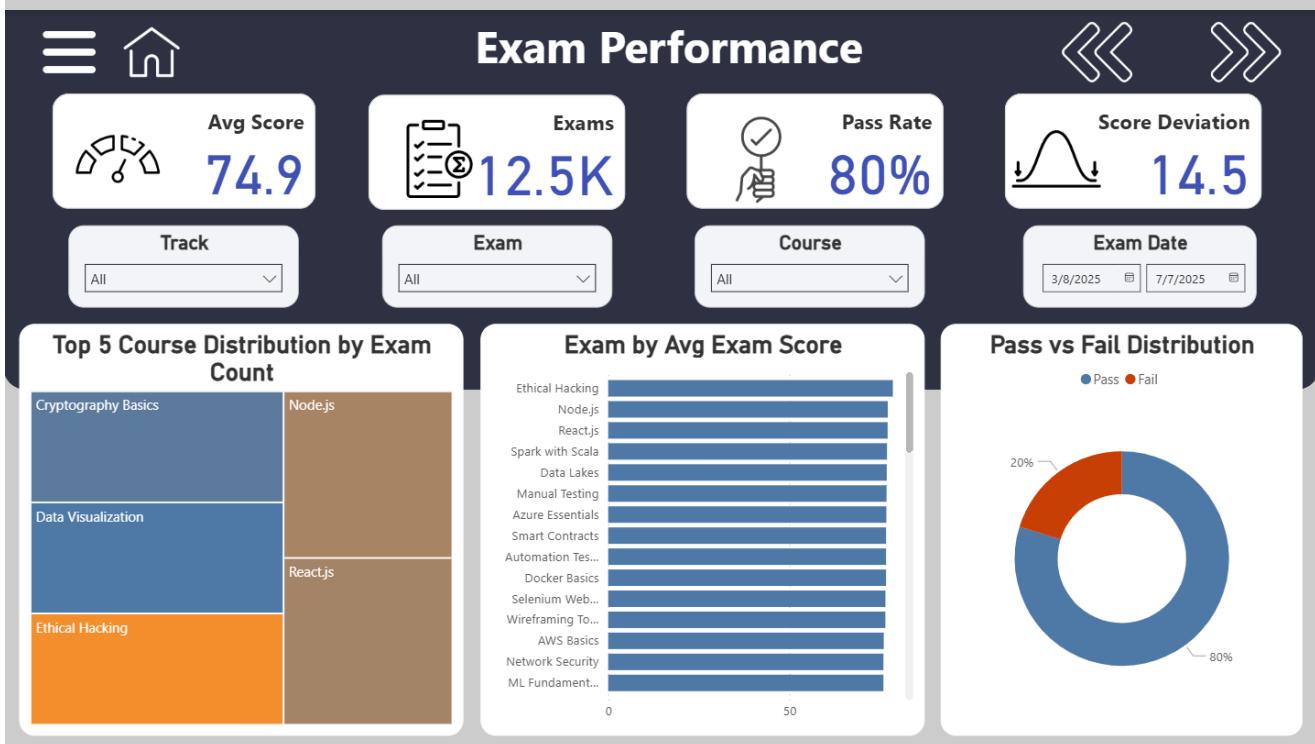
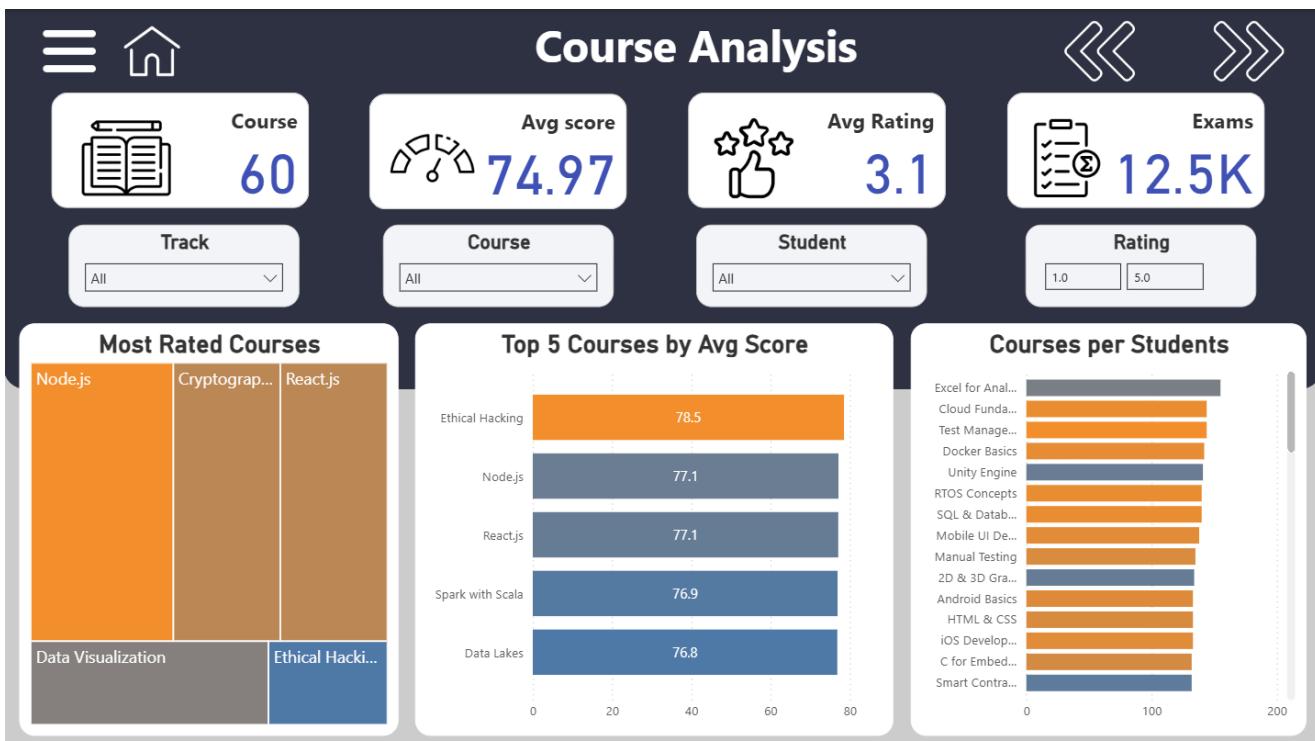
Instructor

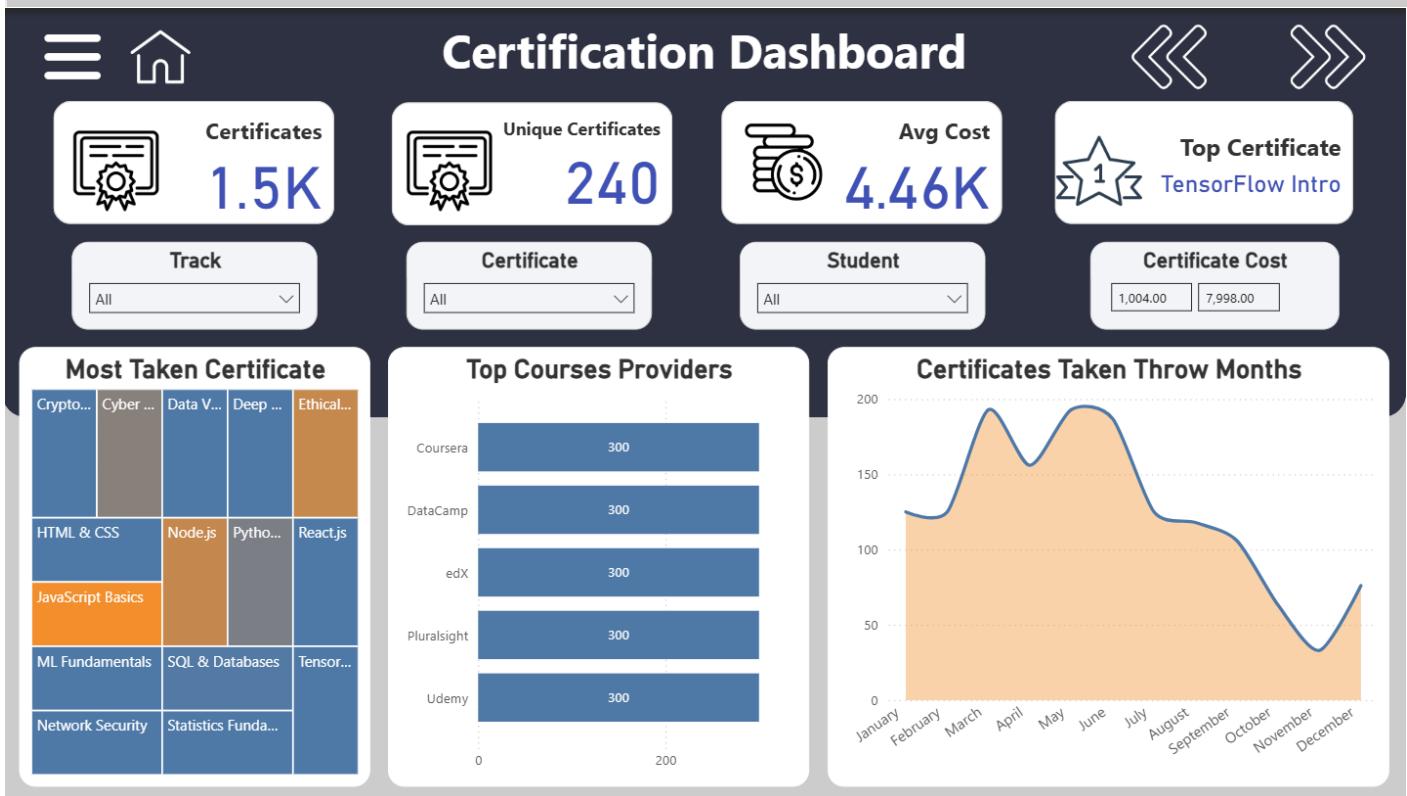
AVG Student Performance Per Instructor

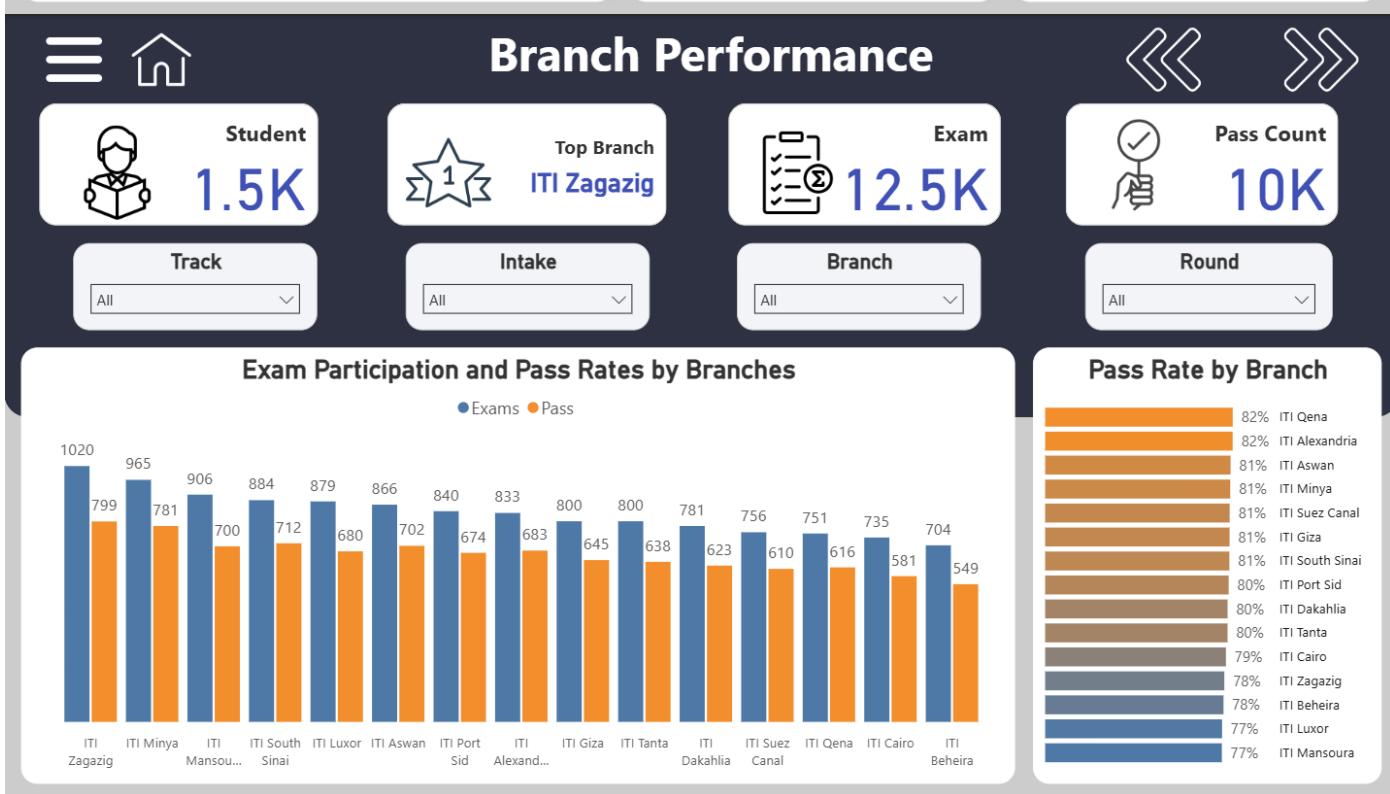
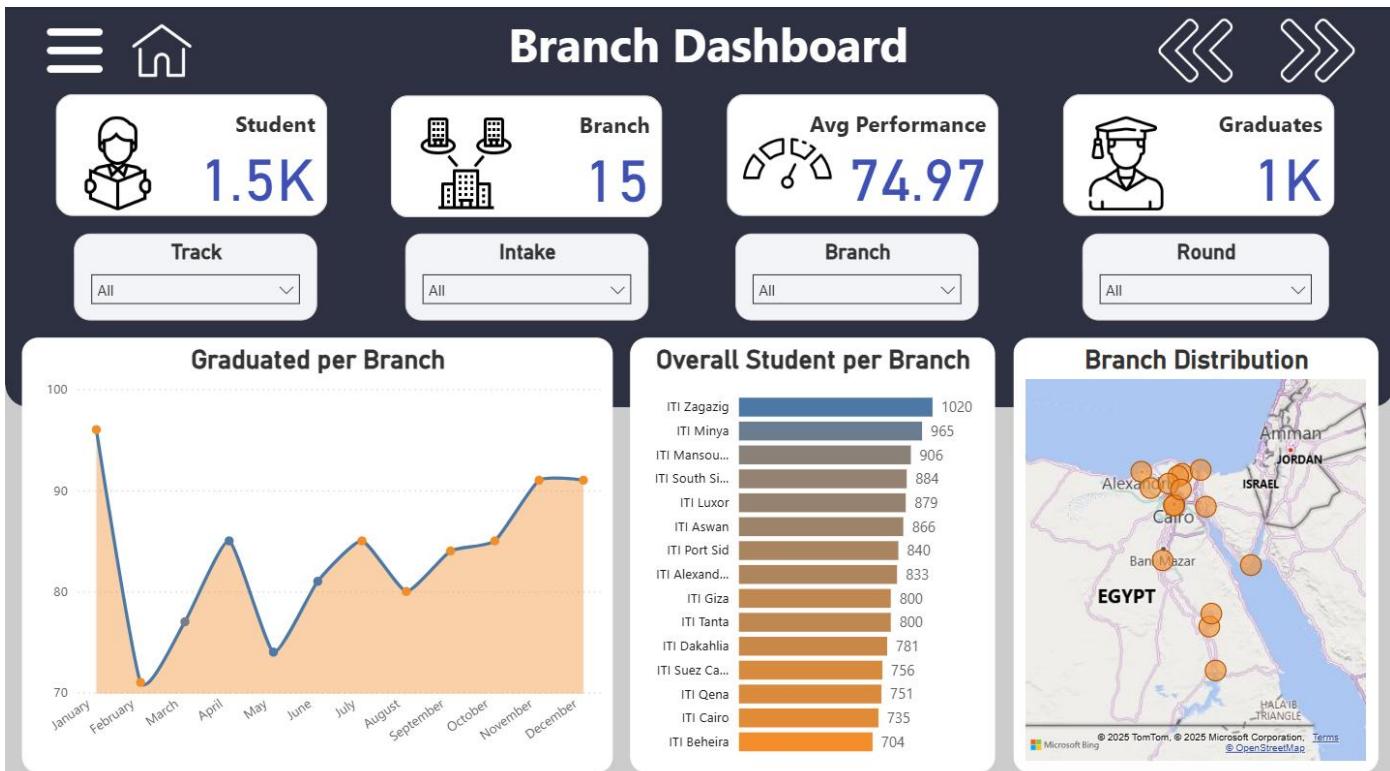


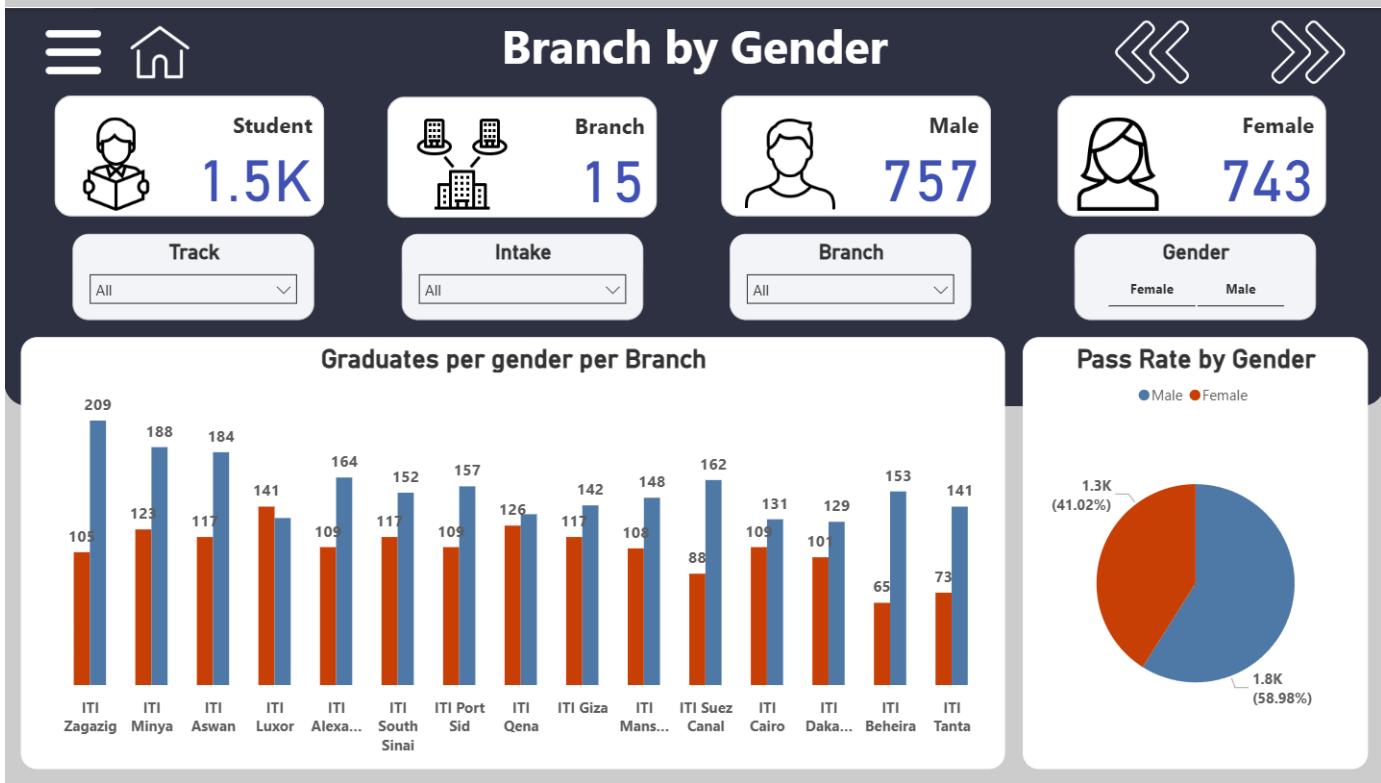
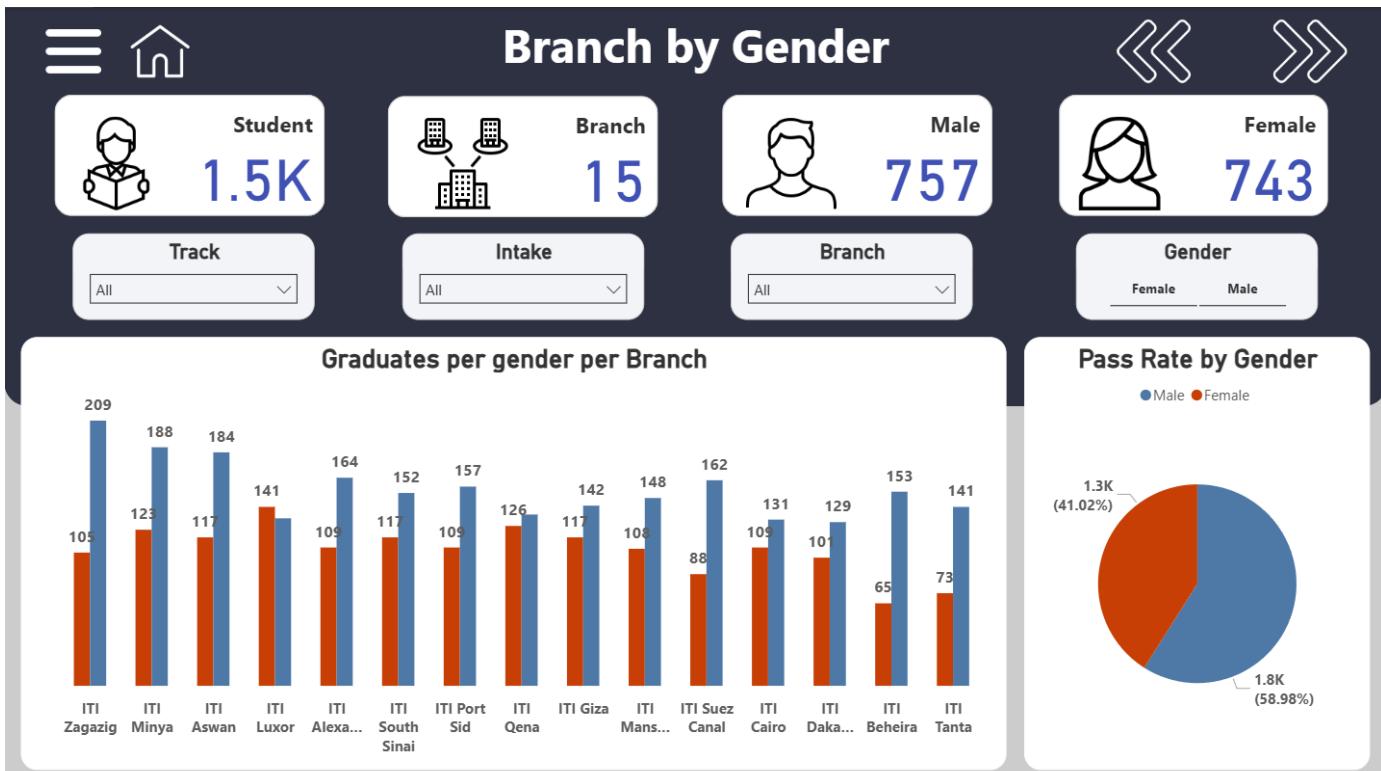
Gender Distribution

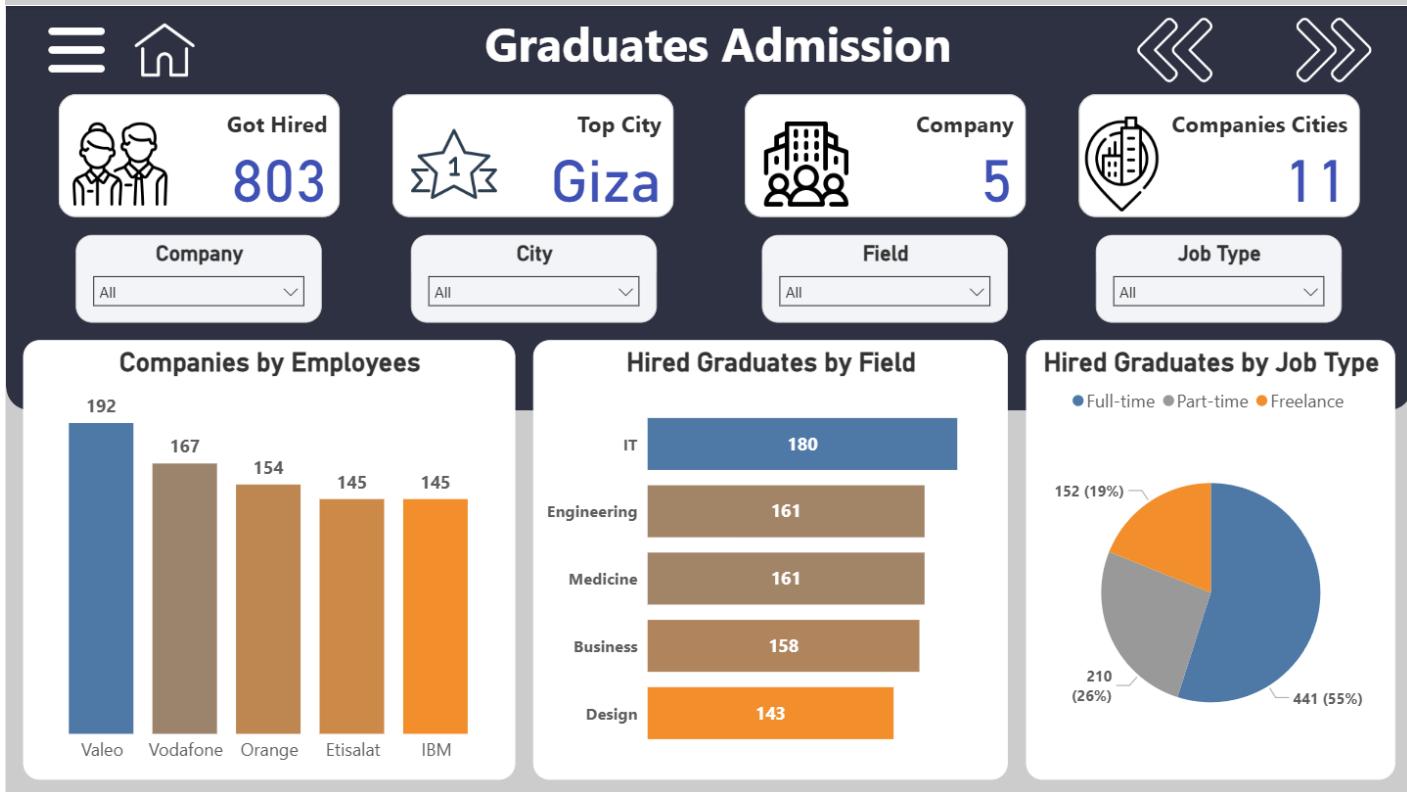
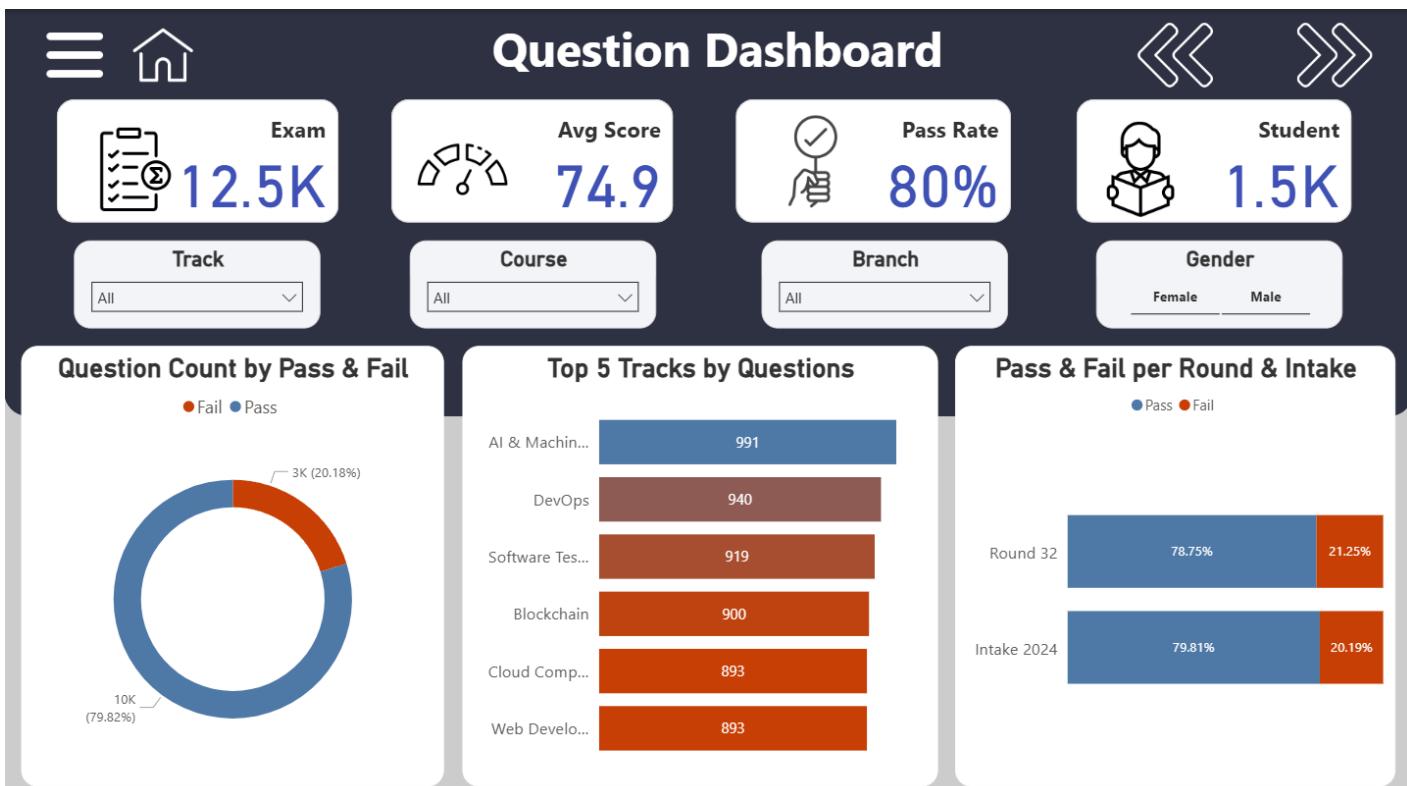


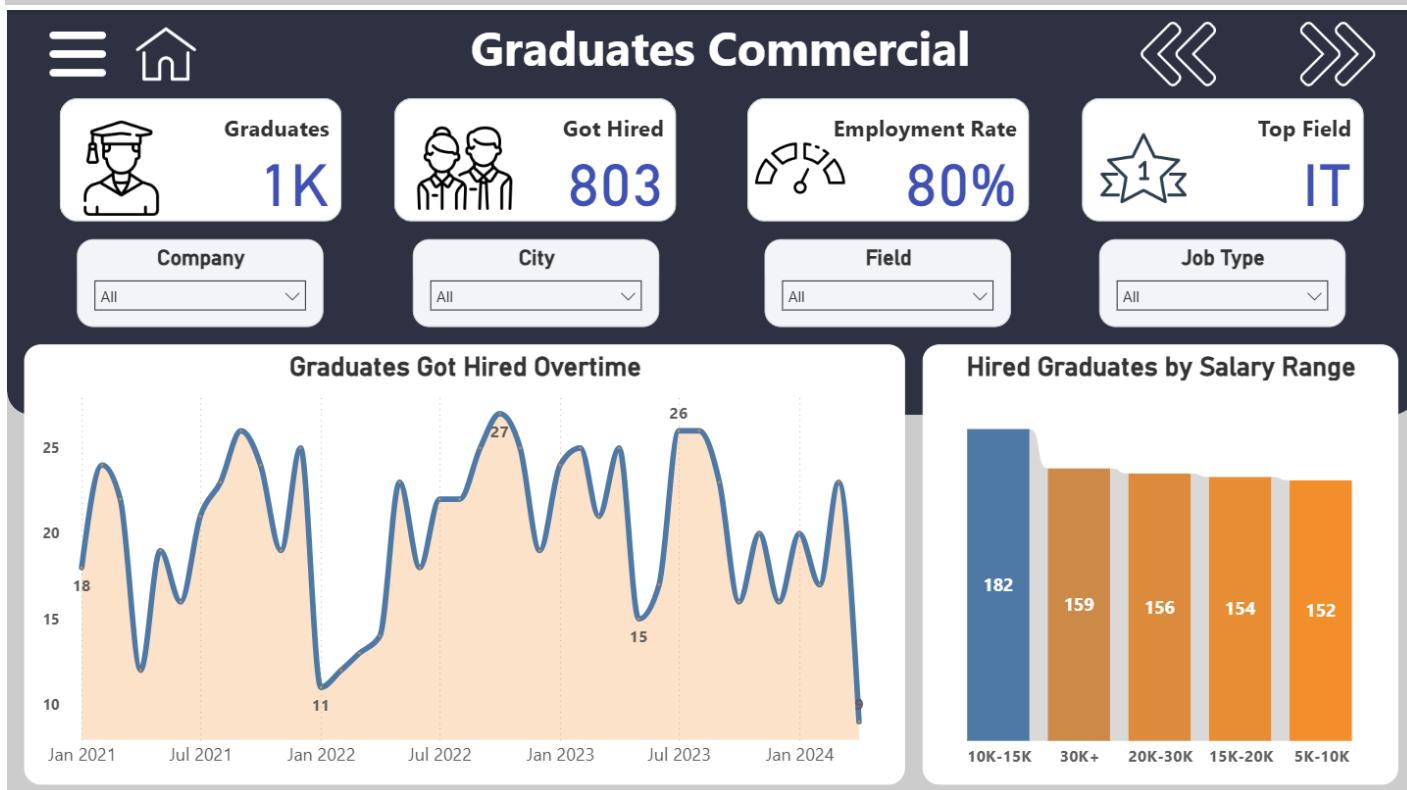


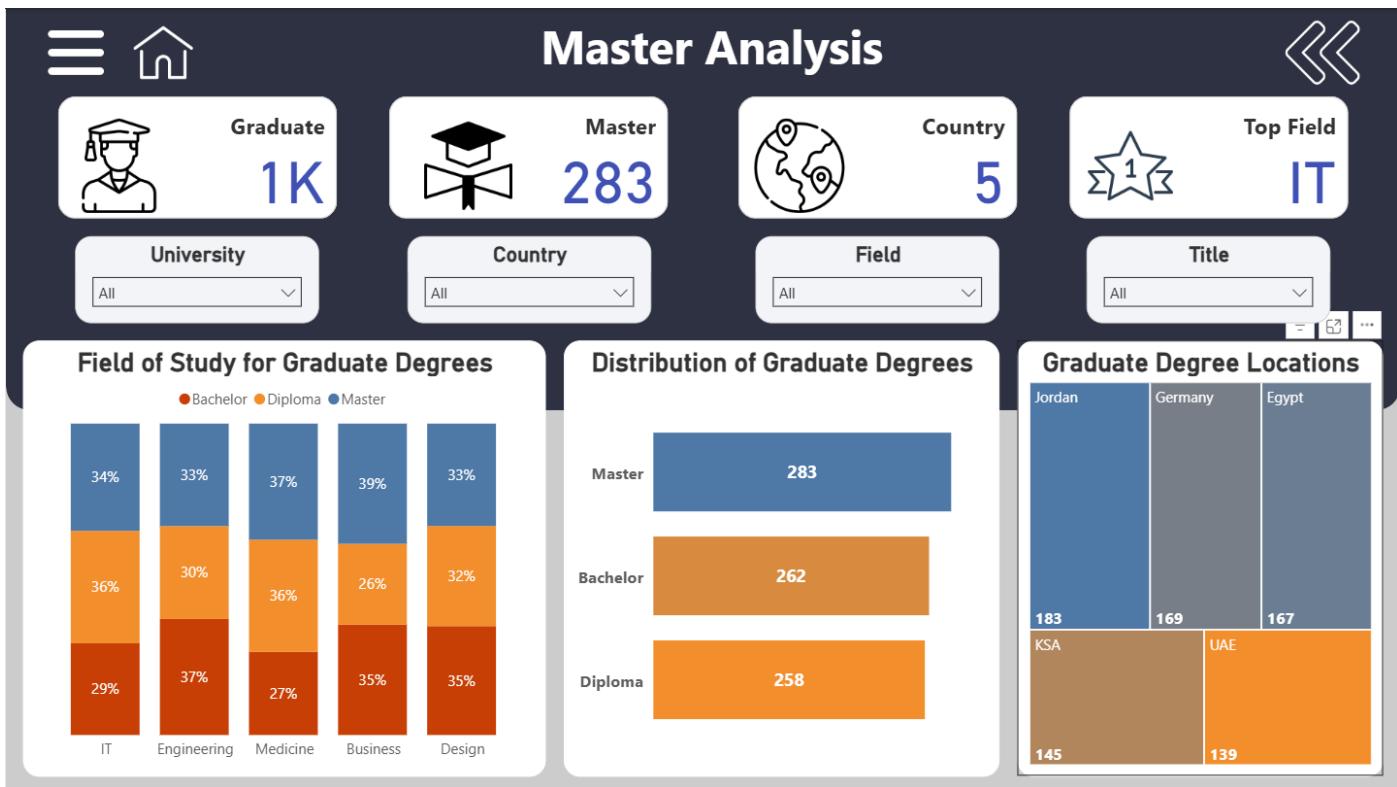












8. Website

web application designed to automate the entire testing process. The platform is built using Python Flask for the backend, a MySQL database for data management, and standard web technologies (HTML, CSS, JavaScript) for a responsive and interactive frontend.

System Architecture

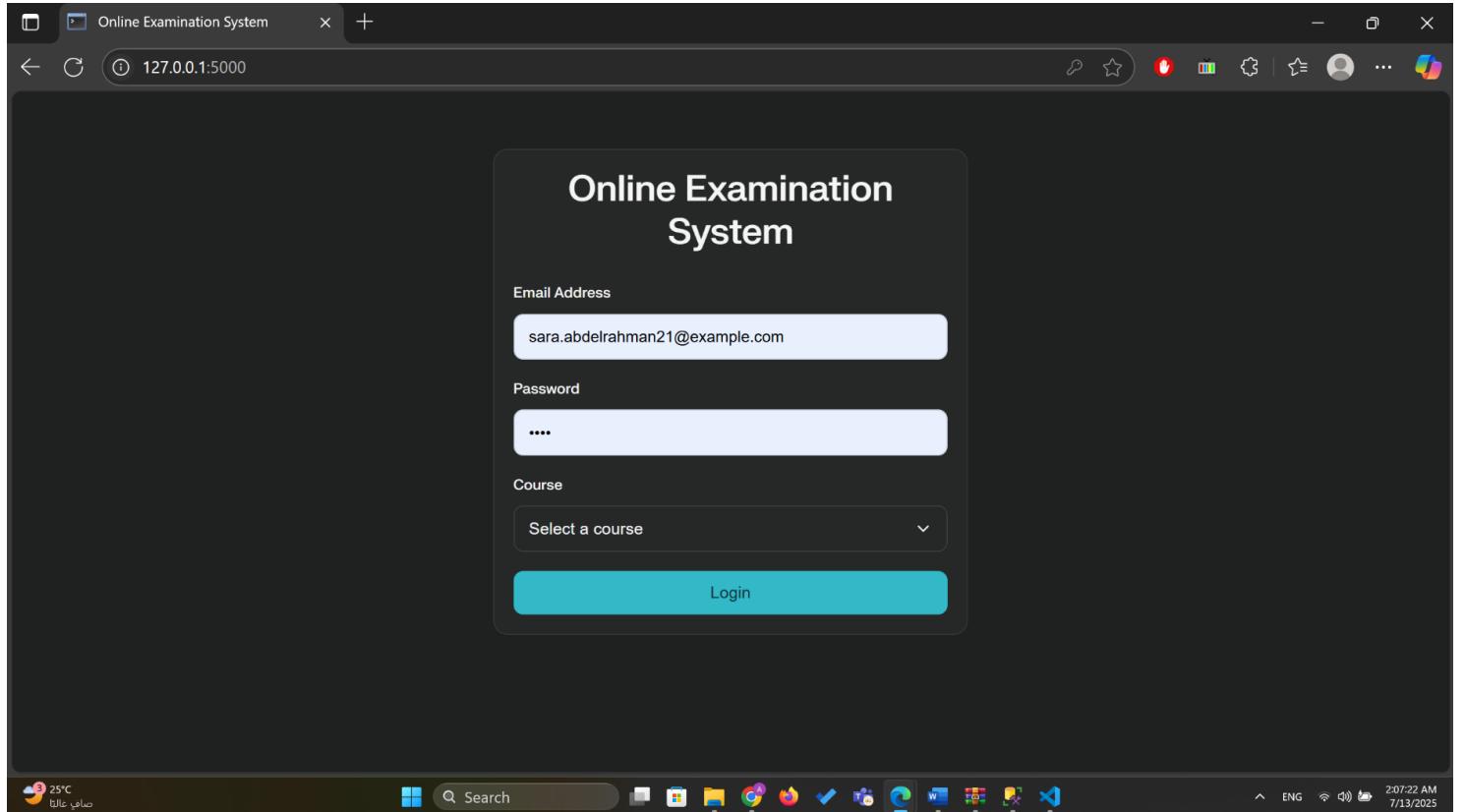
The application employs a modern, multi-tier architecture that separates concerns, ensuring scalability, security, and ease of maintenance.

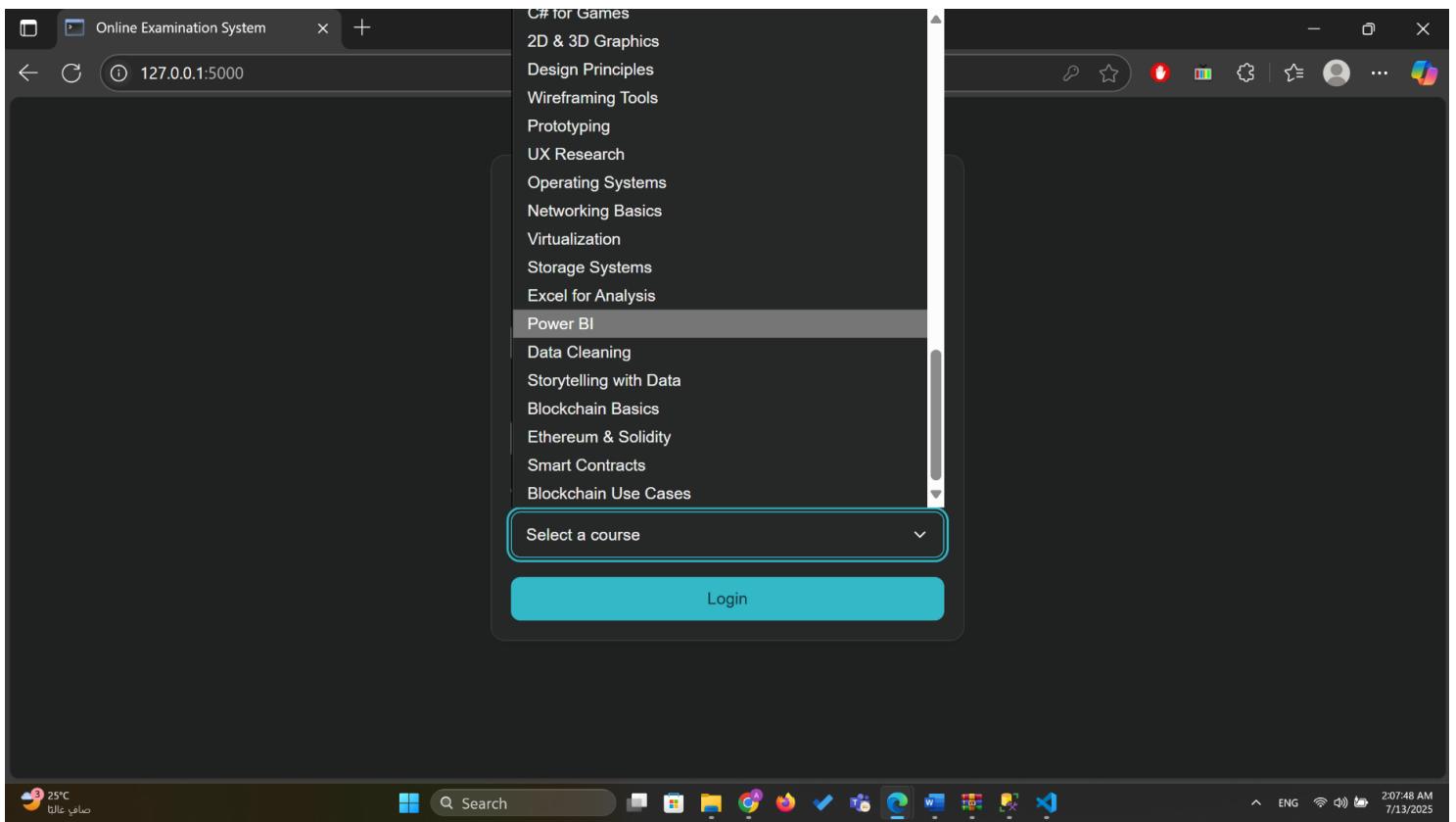
- **Backend (Python Flask):** The server-side logic is managed by Flask, a lightweight and powerful Python web framework. The backend is responsible for handling all API requests, managing user sessions, authenticating users, and orchestrating the entire exam workflow by communicating with the database.
- **Database (MySQL):** At the core of the system is the `Graduation_Project_OLTP` relational database, which stores all data related to students, courses, questions, and exam results. Critically, all database interactions are encapsulated within stored procedures, a key design choice that enhances performance and provides a strong defense against SQL injection attacks.
- **Frontend (HTML, CSS, JavaScript):** The user interface is crafted to be intuitive and responsive, providing a seamless experience on both desktop and mobile devices.
 - **HTML (index.html):** Provides the structural foundation for the application's pages, including the login form, student dashboard, exam interface, and results summary.
 - **CSS (style.css):** Defines the visual presentation, featuring a professional and modern aesthetic with a clean layout, a consistent color scheme, and responsive design principles that adapt to various screen sizes.
 - **JavaScript (app.js):** Powers all client-side interactivity. It handles user input, manages asynchronous API calls to the Flask backend, controls the real-time exam timer, and dynamically updates the user interface without requiring page reloads.

Core Examination Workflow

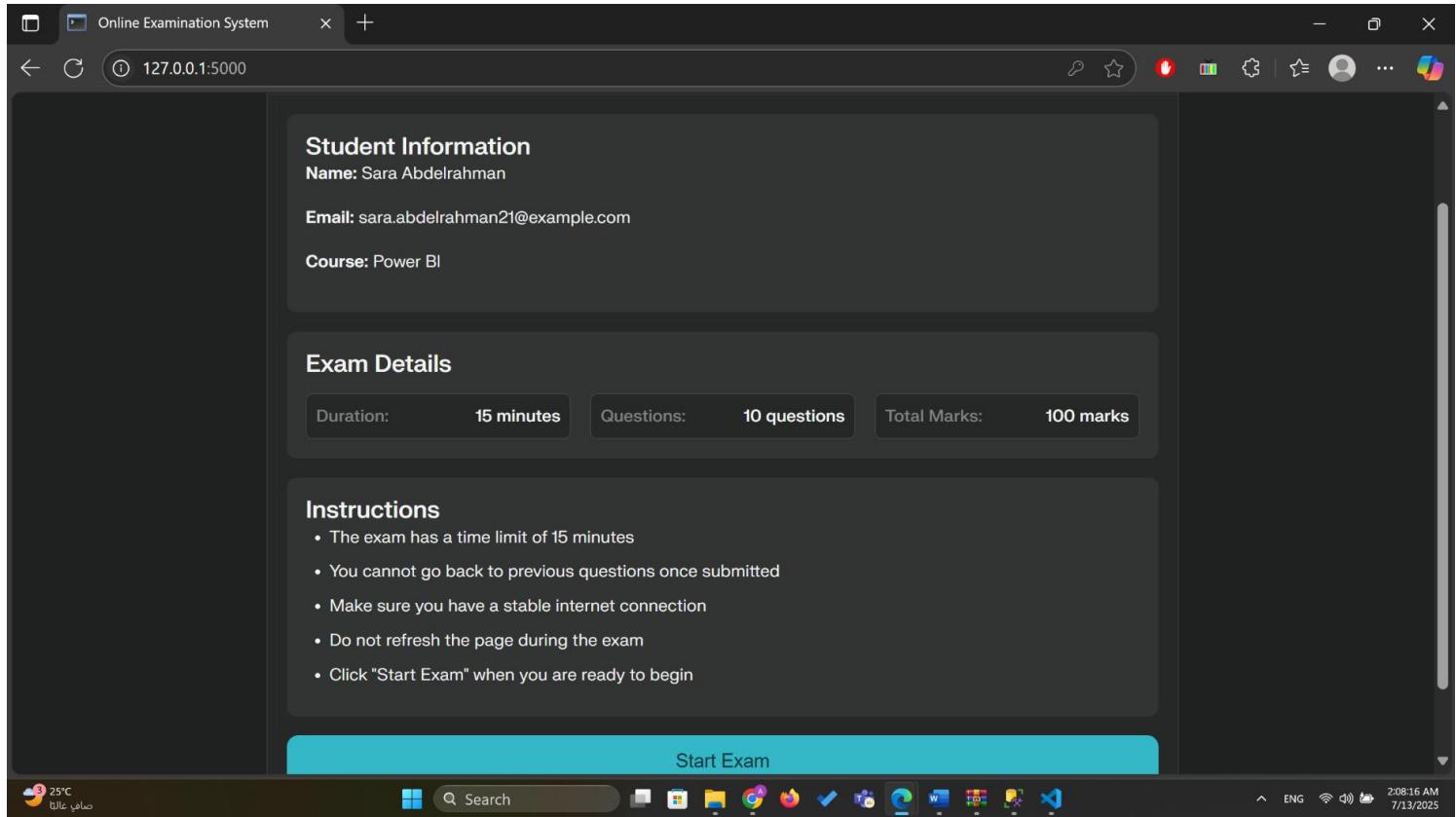
The system guides students through a logical and automated four-step examination process:

- Login and Exam Generation:** A student initiates the process by logging in with their email, password, and selected course. The Flask backend validates these credentials against the `Student` table. Upon successful authentication, the system calls the `sp_generate_exam_on_login` stored procedure, which dynamically creates a new, unique exam instance in the database and returns an `exam_id`. This ensures each exam attempt is a distinct record.





2. **Starting the Exam:** When the student clicks "Start Exam," the backend uses the `sp_start_exam` procedure. This procedure, in turn, calls `sp_GenerateExamQuestions` to randomly select a set of 10 questions for the specified course. To display the options for each question, the system fetches choices using the `sp_GetAnswerChoices` procedure.



3. **Exam Submission and Automated Grading:** After the student completes the exam and submits their answers, the backend executes a precise sequence of stored procedures to automate the grading process:
 - First, `sp_InsertStudentAnswer` is called for each question to securely log the student's chosen answer.
 - Next, `sp_CorrectAnswersAndUpdateGrades` is executed. This procedure compares the student's answers to the correct ones stored in the database, assigns a grade for each question, and calculates the final percentage score.
 - Finally, `sp_InsertExamResult` is called to record the student's overall result, including the percentage and a "Pass" or "Fail" status, into the `Exam_Result` table.

Online Examination System 127.0.0.1:5000 14:54

Examination in Progress

Question 1 of 10

Functions that belong to the category of TEXT FUNCTION.

- CONCATENATE
- COUNT
- DATE

Previous Next

25°C مأوى عاليٌ Search ENG 7/13/2025

Online Examination System 127.0.0.1:5000 13:30

Examination in Progress

Question 10 of 10

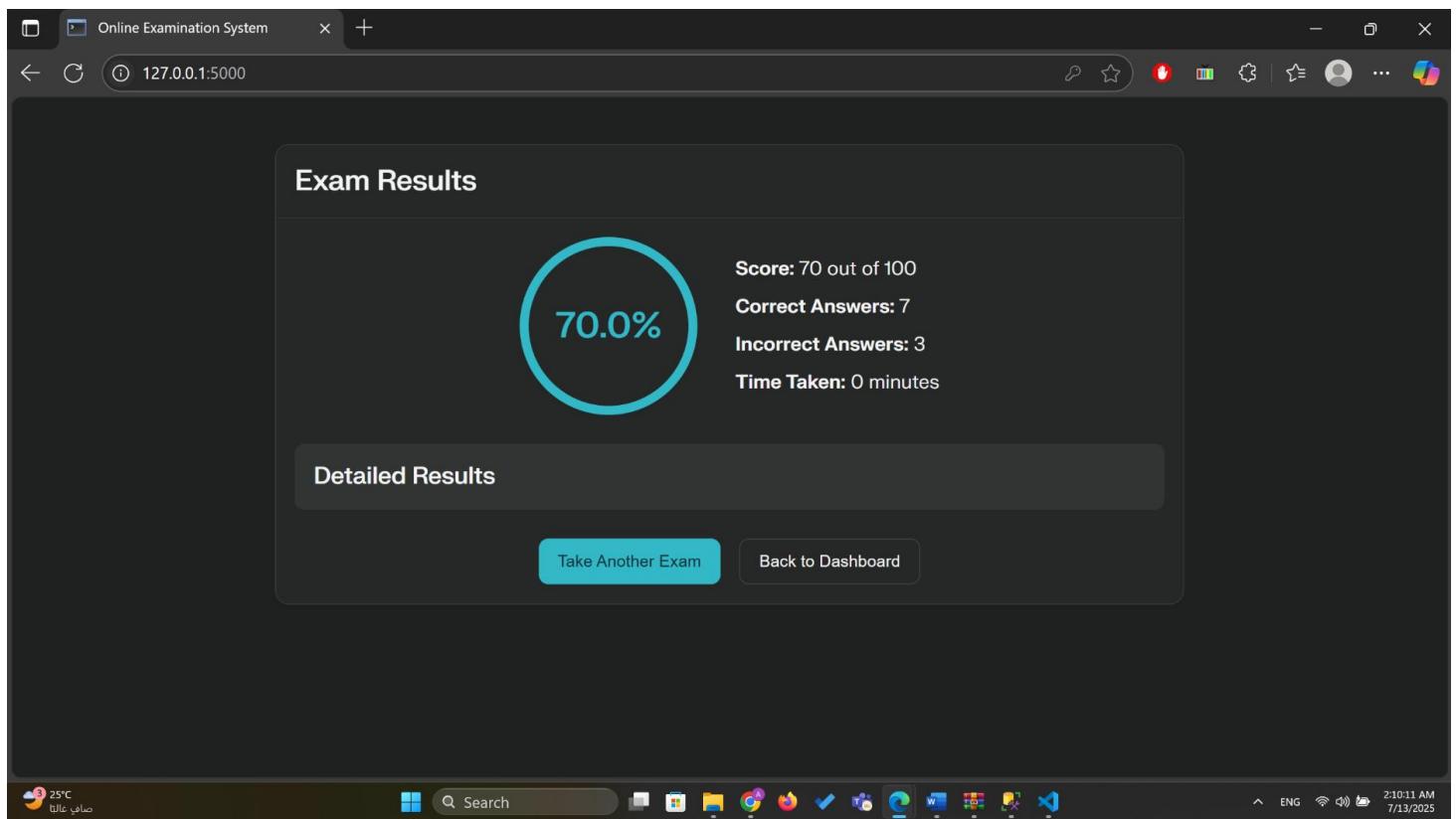
Which of the following is a way to filter data in Power BI?

- Histogram
- Scatter plot
- Slicer

Previous Submit Exam

25°C مأوى عاليٌ Search ENG 7/13/2025

4. **Results Display:** The student is immediately redirected to a results page that provides a comprehensive summary of their performance. This includes their final score, the total number of correct and incorrect answers, and the time taken to complete the exam.



Key Website Features

The application incorporates a range of features designed to enhance functionality, user experience, and security.

User Interface and Experience

- **Responsive Design:** The user interface, governed by `style.css`, is fully responsive and ensures a consistent and accessible experience on desktops, tablets, and smartphones.
- **Real-time Feedback:** The application provides immediate feedback, such as error messages on the login form and a real-time countdown timer during the exam. The timer changes color to warn the student as time runs low, creating a sense of urgency.
- **Interactive Exam Elements:** A visual progress bar keeps the student informed of their progress, while interactive radio buttons allow for easy answer selection.

Security

- **SQL Injection Prevention:** By using parameterized stored procedures for all database operations, the application is protected against SQL injection, one of the most common web security vulnerabilities.
- **Session Management:** Flask's secure session management is used to track authenticated students, ensuring that only logged-in users can access the exam and submit answers.
- **Basic Anti-Cheating Measures:** The frontend JavaScript includes simple but effective measures to deter cheating. It disables the use of developer tools and context menus and alerts the user if they attempt to switch tabs during the exam.

**

1. App.py
2. Project_setup.md
3. Exam_procedures.sql
4. index.html
5. style.css
6. app.js