# Examination System Documentation

Defining the problem

- We were asked to build a student examination system

Accepted Solution

- Build a desktop application from which we can manage this student examination system

System components

- Desktop application
- Database (SQL Server)
- Reports (SSRS)
- Dashboards (PowerBI)

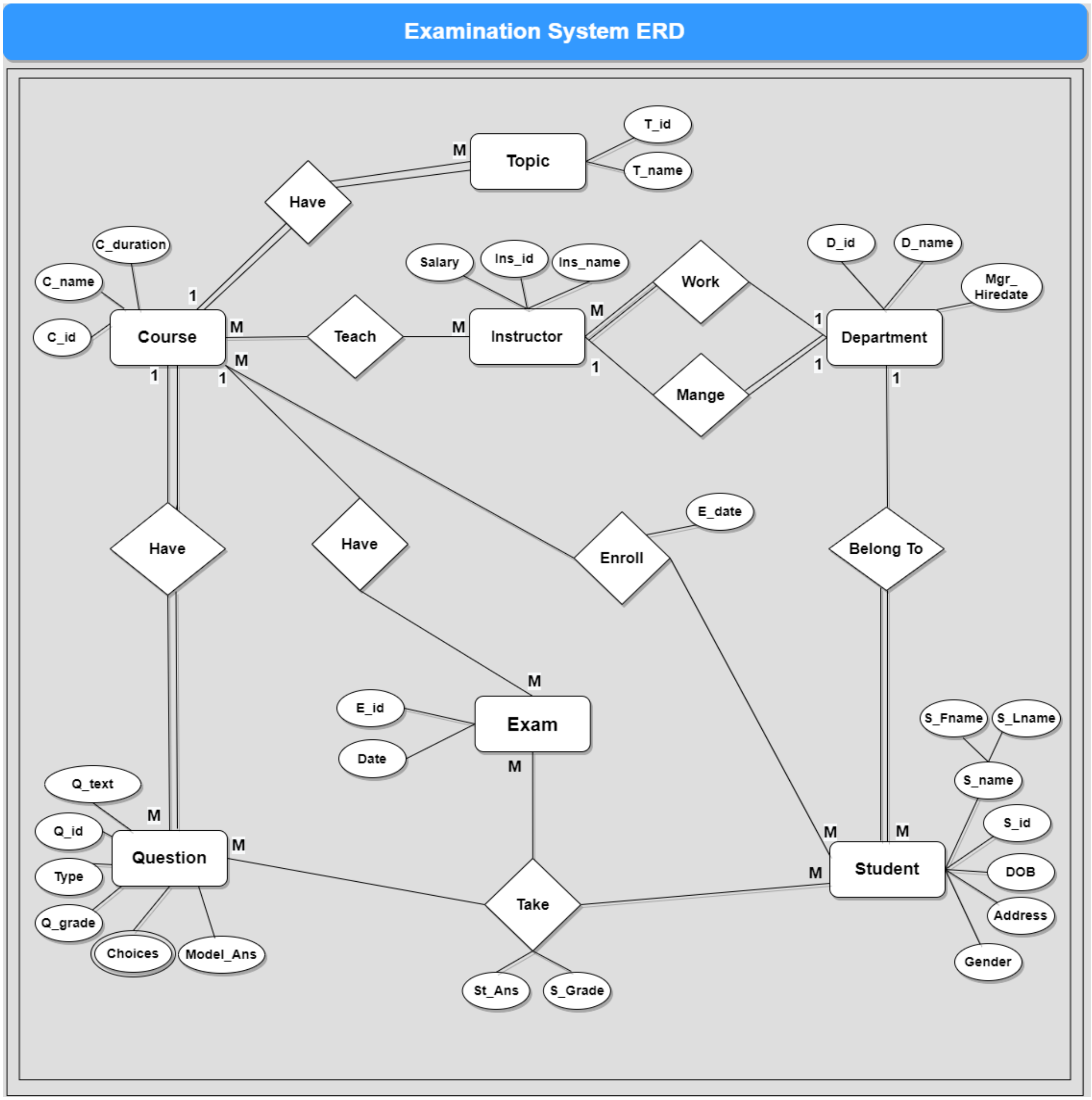
How this system works

- The system database was designed to store.
  1- Students' data (names, courses, exams, grades, and departments).
  2- Instructors' data (names, departments, salaries, courses)
  3- Exams data (students' exams, students, answers, students' grades).
  4- Questions data (questions’ grade, model answer).
  5- Choices (three wrong choices for each exam).
  6- Courses data.
- Students have to take exams on this desktop application and their answers have to be stored in the database.
- Students’ grades can be shown through some reports.
- These reports are made using SSRS.
- Reports were published locally on the machine.
- From the application, you can access multiple reports that were made (based on requirements).
- As for requirements, students’ data needed to be analyzed through some dashboards made on PowerBI.
- Dashboards as well can be accessed from the desktop application

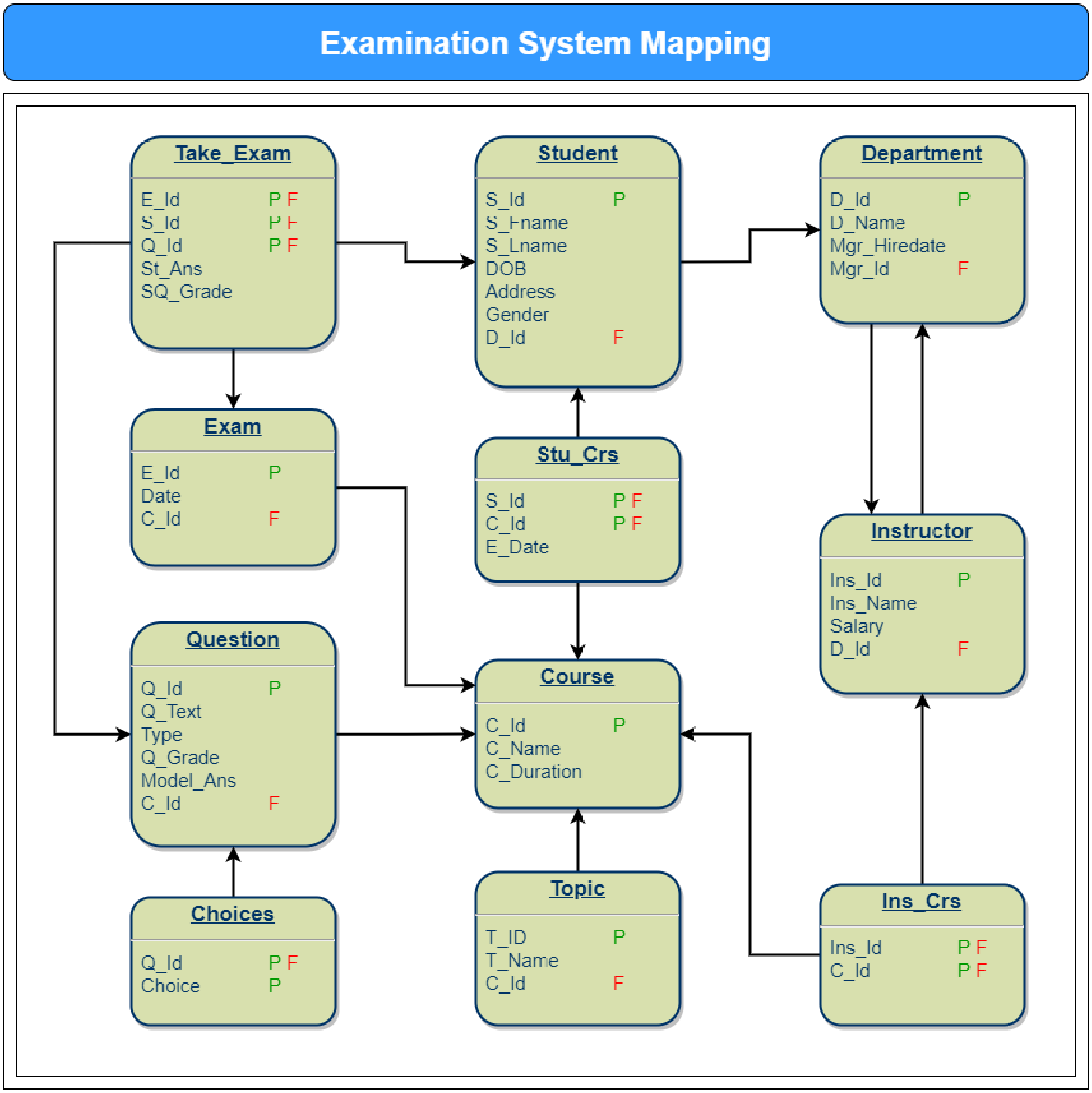
# STEPS OF WORK

## 1- Design Phase

- Building database ERD.



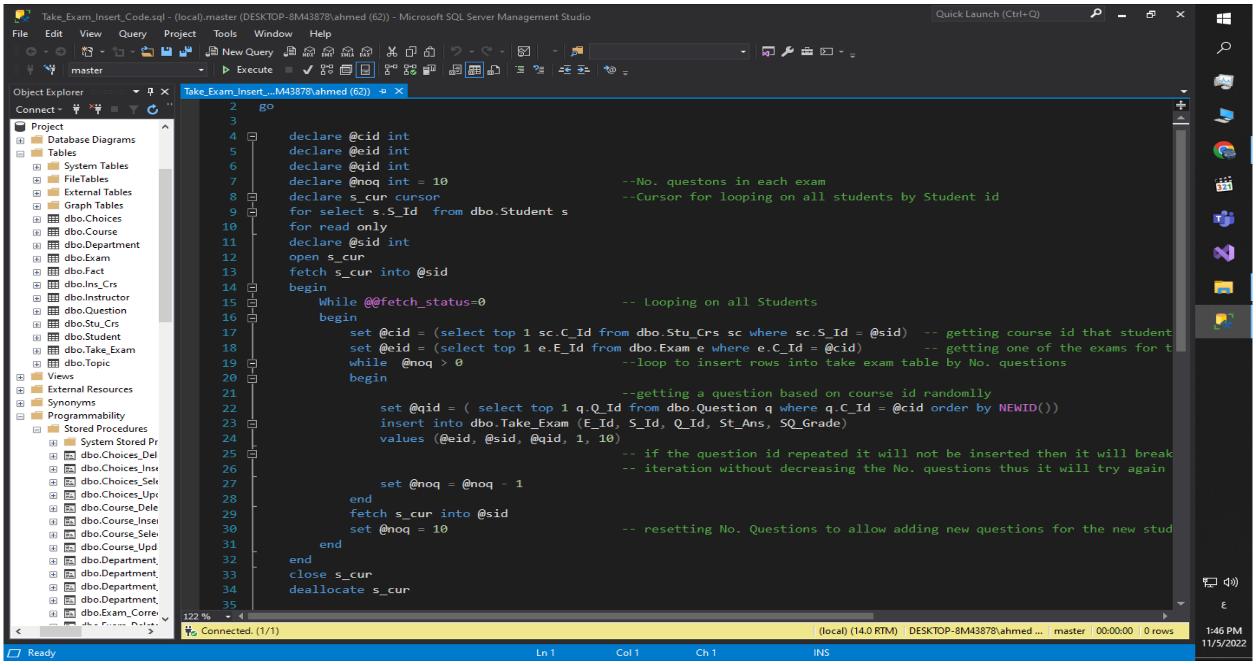
- Mapping for database ERD.


Examination System Mapping

# 2- Populating the DB Phase

- Now we already have an accepted form of database BUT with no records
- Through some web searching, we could get students' data to fill student’s table
- As for exams, and through some web scrapping using python BeatifulSoup we could get exam questions for (Python, SW engineering, Cyber Security, …..)

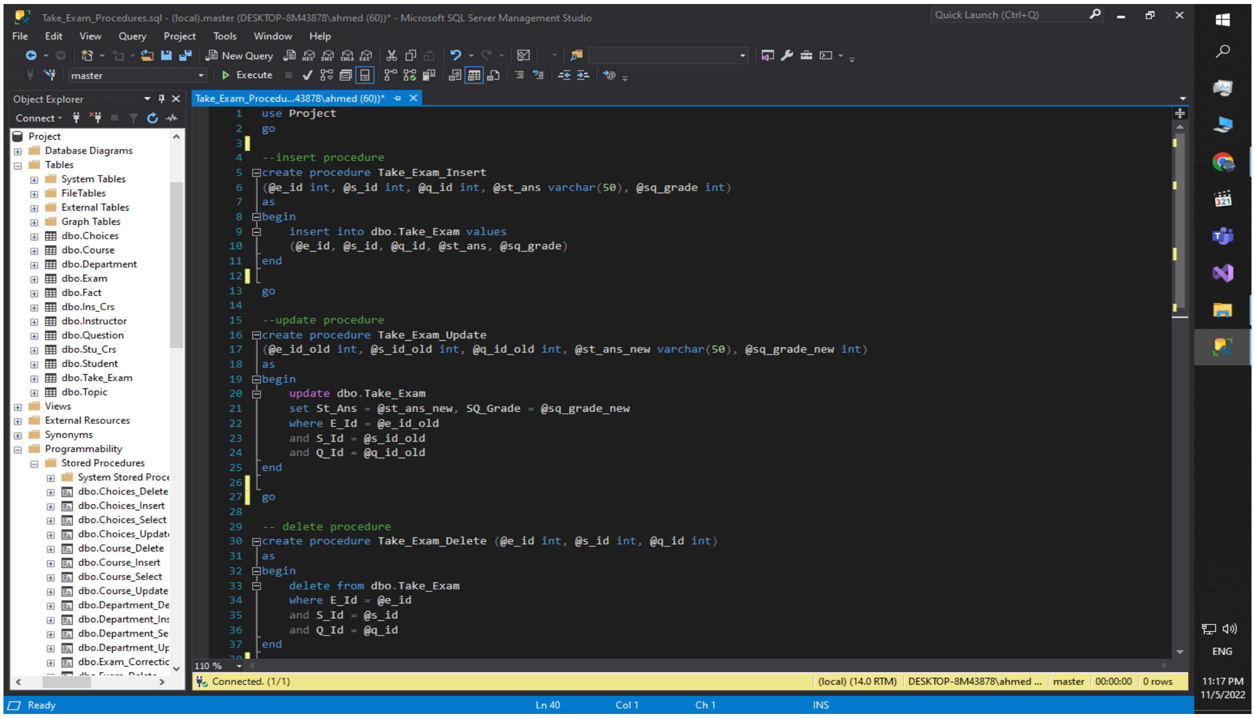
- As for analysis and to Test and to be able to build reports on students’ data we had to build some stored procedures to fill the Take_Exam table in which exam_id, s_id, and students’ answers were stored
- This SQL Script fill random exam for random students and make them (students) answer random answers to be able to build reports and dashboards



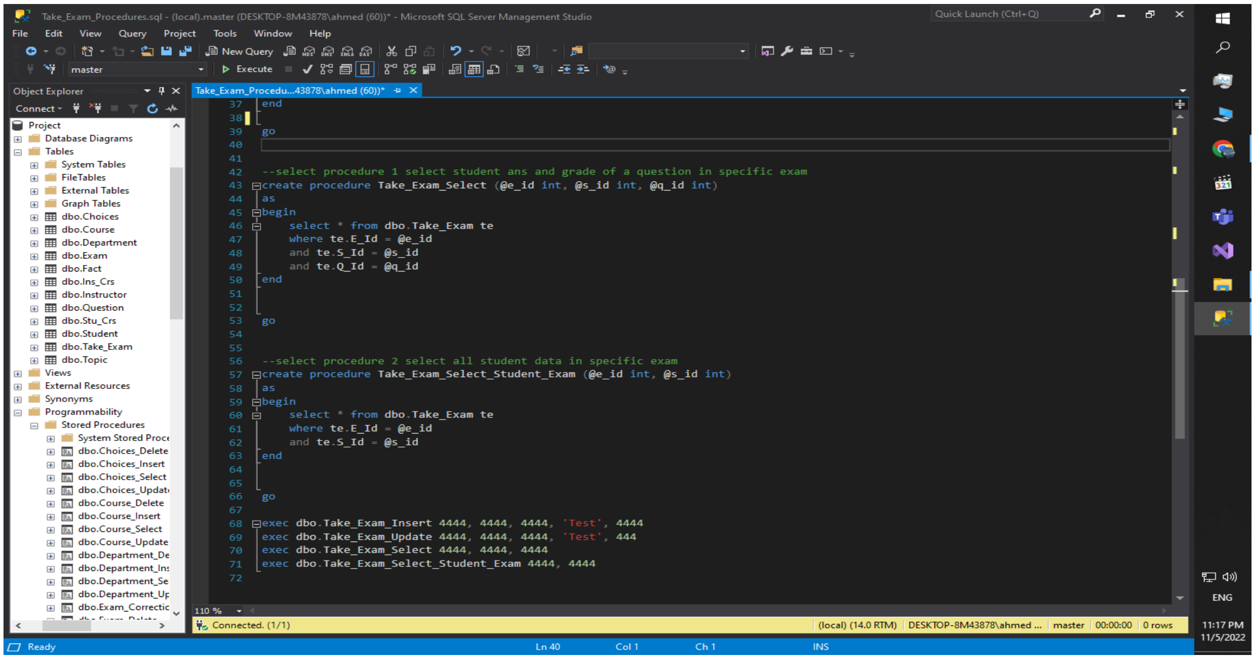
```sql
go

    declare @cid int
    declare @eid int
    declare @qid int
    declare @noq int = 10                       --No. questons in each exam
    declare s_cur cursor                        --Cursor for looping on all students by Student id
    for select s.S_Id  from dbo.Student s
    for read only
    declare @sid int
    open s_cur
    fetch s_cur into @sid
    begin
        While @@fetch_status=0                  -- Looping on all Students
        begin
            set @cid = (select top 1 sc.C_Id from dbo.Stu_Crs sc where sc.S_Id = @sid)  -- getting course id that student
            set @eid = (select top 1 e.E_Id from dbo.Exam e where e.C_Id = @cid)        -- getting one of the exams for t
            while  @noq > 0                     --loop to insert rows into take exam table by No. questions
            begin
                                                --getting a question based on course id randomlly
                set @qid = ( select top 1 q.Q_Id from dbo.Question q where q.C_Id = @cid order by NEWID())
                insert into dbo.Take_Exam (E_Id, S_Id, Q_Id, St_Ans, SQ_Grade)
                values (@eid, @sid, @qid, 1, 10)
                                                -- if the question id repeated it will not be inserted then it will break
                                                -- iteration without decreasing the No. questions thus it will try again
                set @noq = @noq - 1
            end
            fetch s_cur into @sid
            set @noq = 10                       -- resetting No. Questions to allow adding new questions for the new stud
        end
    end
    close s_cur
    deallocate s_cur
```

# 3- Stored Procedures Phase

- Stored procedures we built to insert, update, delete, and select data from each table in the model (**50** SP)



```sql
use Project
go

--insert procedure
create procedure Take_Exam_Insert
(@e_id int, @s_id int, @q_id int, @st_ans varchar(50), @sq_grade int)
as
begin
    insert into dbo.Take_Exam values
    (@e_id, @s_id, @q_id, @st_ans, @sq_grade)
end

go

--update procedure
create procedure Take_Exam_Update
(@e_id_old int, @s_id_old int, @q_id_old int, @st_ans_new varchar(50), @sq_grade_new int)
as
begin
    update dbo.Take_Exam
    set St_Ans = @st_ans_new, SQ_Grade = @sq_grade_new
    where E_Id = @e_id_old
    and S_Id = @s_id_old
    and Q_Id = @q_id_old
end

go

-- delete procedure
create procedure Take_Exam_Delete (@e_id int, @s_id int, @q_id int)
as
begin
    delete from dbo.Take_Exam
    where E_Id = @e_id
    and S_Id = @s_id
    and Q_Id = @q_id
end
```



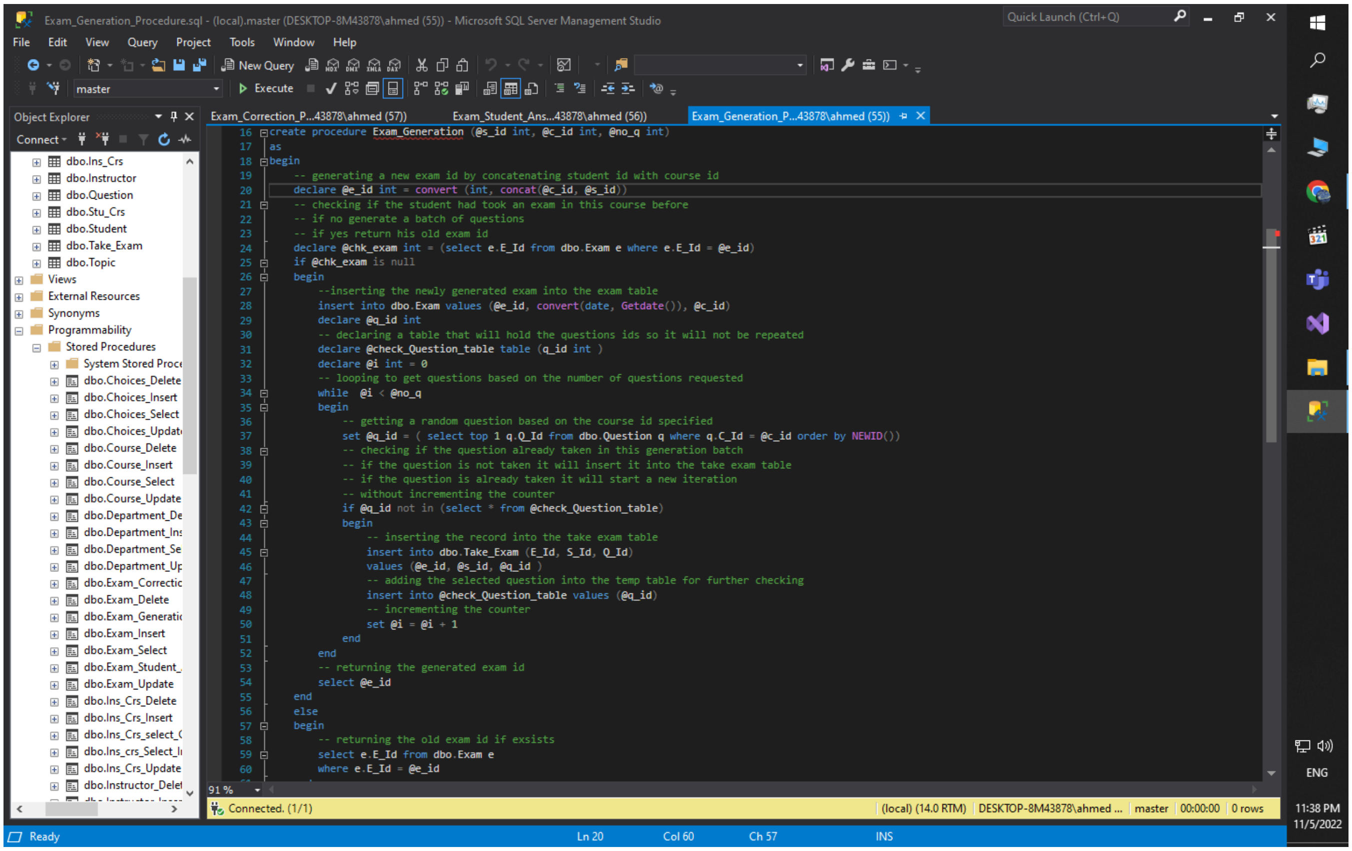
Take_Exam_Procedures.sql - (local).master (DESKTOP-8M43878\ahmed (60))* - Microsoft SQL Server Management Studio
Take_Exam_Procedu...43878\ahmed (60))*
Connected. (1/1)
Ready
Ln 40
Col 1
Ch 1
INS

```sql
end

go


--select procedure 1 select student ans and grade of a question in specific exam
create procedure Take_Exam_Select (@e_id int, @s_id int, @q_id int)
as
begin
    select * from dbo.Take_Exam te
    where te.E_Id = @e_id
    and te.S_Id = @s_id
    and te.Q_Id = @q_id
end


go


--select procedure 2 select all student data in specific exam
create procedure Take_Exam_Select_Student_Exam (@e_id int, @s_id int)
as
begin
    select * from dbo.Take_Exam te
    where te.E_Id = @e_id
    and te.S_Id = @s_id
end


go

exec dbo.Take_Exam_Insert 4444, 4444, 4444, 'Test', 4444
exec dbo.Take_Exam_Update 4444, 4444, 4444, 'Test', 444
exec dbo.Take_Exam_Select 4444, 4444, 4444
exec dbo.Take_Exam_Select_Student_Exam 4444, 4444
```

- Other vital Stored procedures that we built to perform the main operations of the system (**5** SP) we will briefly mention 3

1. Exam Generation SP:

-- Input-----> Student id, Course id, Number of Questions
-- Action----> map the relations between a student and a set
of questions in a course  by inserting these
data
into take exam table & generate new exam id
if the student took an exam before in this
course it will return the old exam id
-- Output----> return an Exam id in this course



```sql
create procedure Exam_Generation (@s_id int, @c_id int, @no_q int)
as
begin
    -- generating a new exam id by concatenating student id with course id
    declare @e_id int = convert (int, concat(@c_id, @s_id))
    -- checking if the student had took an exam in this course before
    -- if no generate a batch of questions
    -- if yes return his old exam id
    declare @chk_exam int = (select e.E_Id from dbo.Exam e where e.E_Id = @e_id)
    if @chk_exam is null
    begin
        --inserting the newly generated exam into the exam table
        insert into dbo.Exam values (@e_id, convert(date, Getdate()), @c_id)
        declare @q_id int
        -- declaring a table that will hold the questions ids so it will not be repeated
        declare @check_Question_table table (q_id int )
        declare @i int = 0
        -- looping to get questions based on the number of questions requested
        while  @i < @no_q
        begin
            -- getting a random question based on the course id specified
            set @q_id = ( select top 1 q.Q_Id from dbo.Question q where q.C_Id = @c_id order by NEWID())
            -- checking if the question already taken in this generation batch
            -- if the question is not taken it will insert it into the take exam table
            -- if the question is already taken it will start a new iteration
            -- without incrementing the counter
            if @q_id not in (select * from @check_Question_table)
            begin
                -- inserting the record into the take exam table
                insert into dbo.Take_Exam (E_Id, S_Id, Q_Id)
                values (@e_id, @s_id, @q_id )
                -- adding the selected question into the temp table for further checking
                insert into @check_Question_table values (@q_id)
                -- incrementing the counter
                set @i = @i + 1
            end
        end
        -- returning the generated exam id
        select @e_id
    end
    else
    begin
        -- returning the old exam id if exsists
        select e.E_Id from dbo.Exam e
        where e.E_Id = @e_id
```

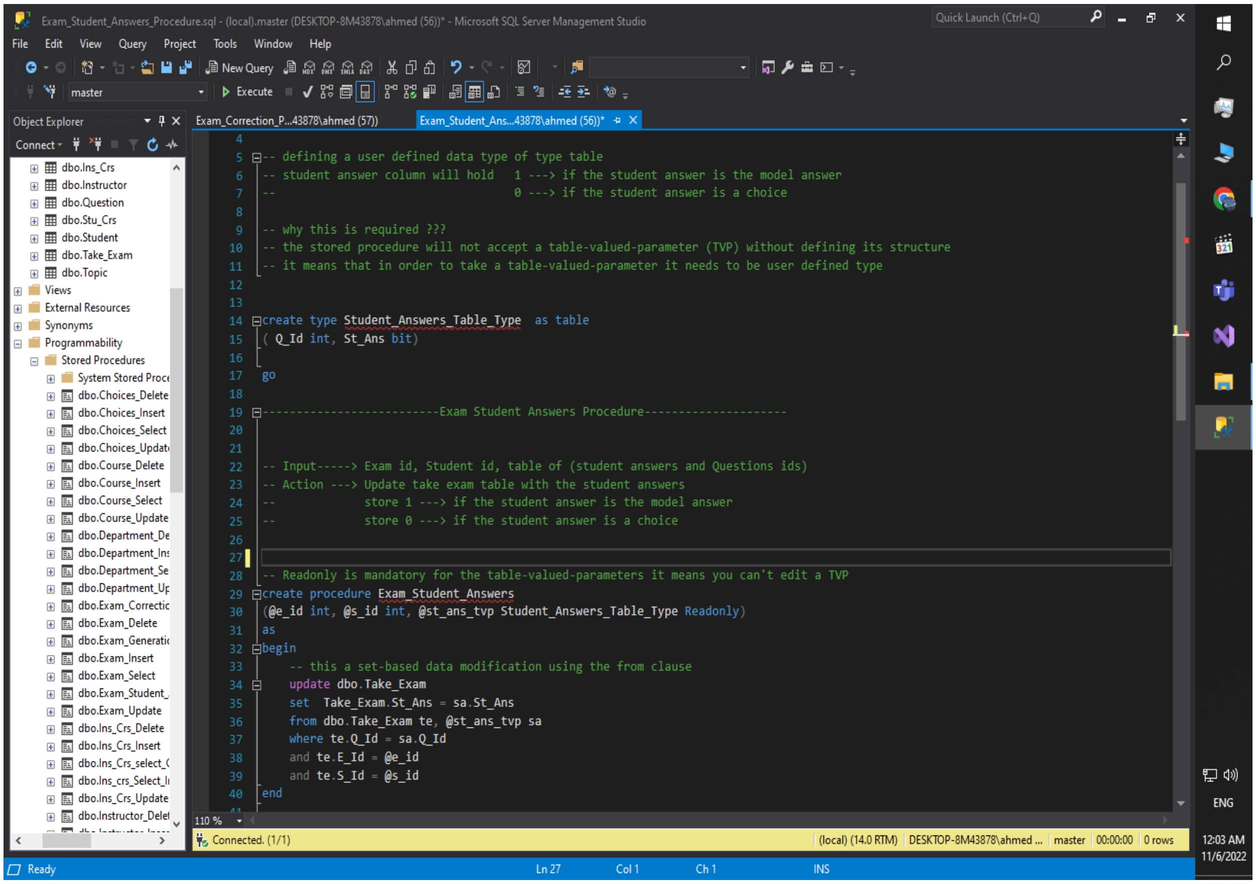
2. Exam Answers SP:

- Input---> Exam id, Student id, table of (student answers and Questions ids)

-Action --> Update take exam table with the student answers

store 1 -> if the student answer is  the model answer

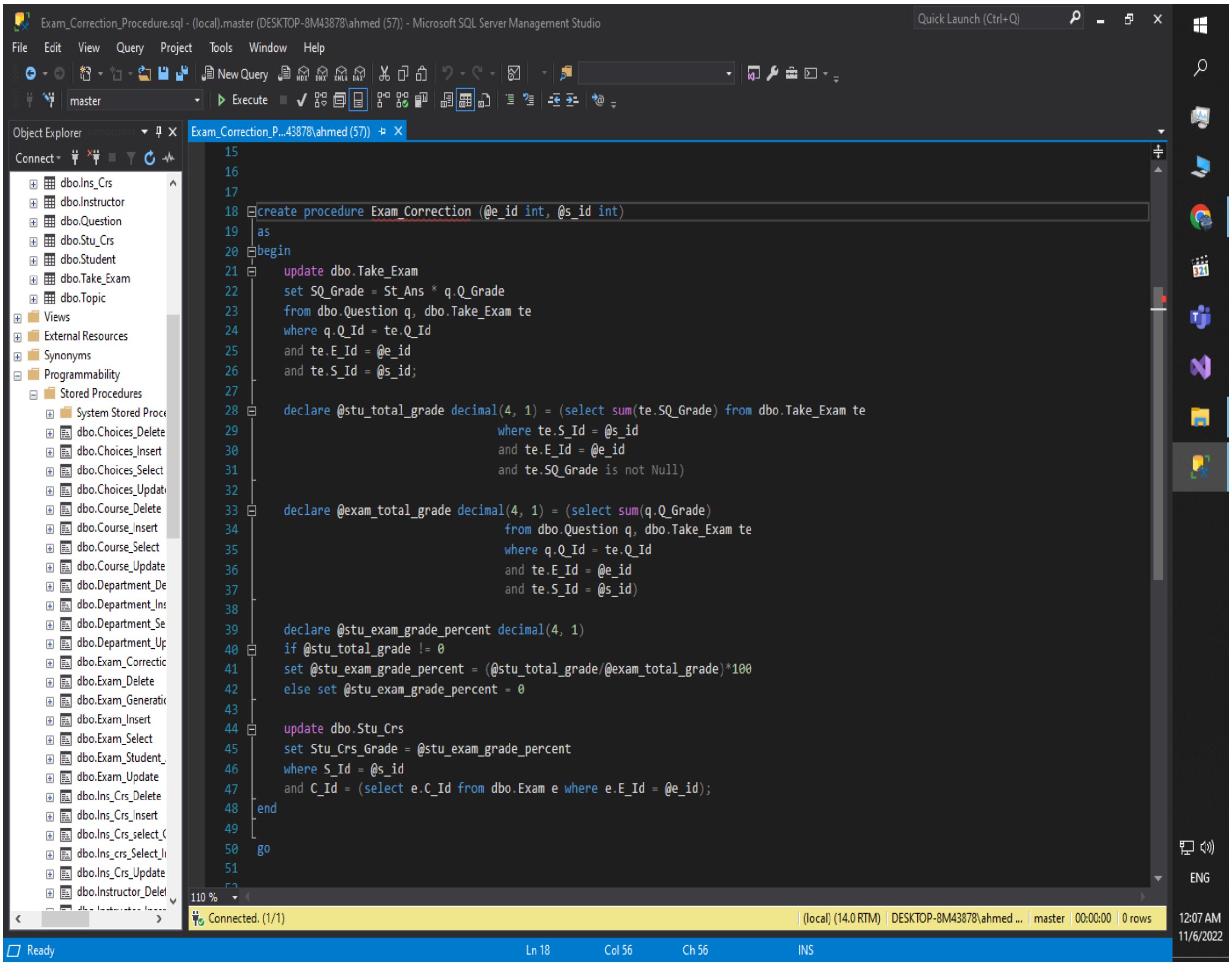
store 0 ---> if the student answer is a choice



```sql

-- defining a user defined data type of type table
-- student answer column will hold   1 ---> if the student answer is the model answer
--                                   0 ---> if the student answer is a choice

-- why this is required ???
-- the stored procedure will not accept a table-valued-parameter (TVP) without defining its structure
-- it means that in order to take a table-valued-parameter it needs to be user defined type


create type Student_Answers_Table_Type  as table
( Q_Id int, St_Ans bit)

go

------------------------Exam Student Answers Procedure--------------------


-- Input-----> Exam id, Student id, table of (student answers and Questions ids)
-- Action ---> Update take exam table with the student answers
--             store 1 ---> if the student answer is the model answer
--             store 0 ---> if the student answer is a choice


-- Readonly is mandatory for the table-valued-parameters it means you can't edit a TVP
create procedure Exam_Student_Answers
(@e_id int, @s_id int, @st_ans_tvp Student_Answers_Table_Type Readonly)
as
begin
    -- this a set-based data modification using the from clause
    update dbo.Take_Exam
    set  Take_Exam.St_Ans = sa.St_Ans
    from dbo.Take_Exam te, @st_ans_tvp sa
    where te.Q_Id = sa.Q_Id
    and te.E_Id = @e_id
    and te.S_Id = @s_id
end
```

3. Exam Correction SP:

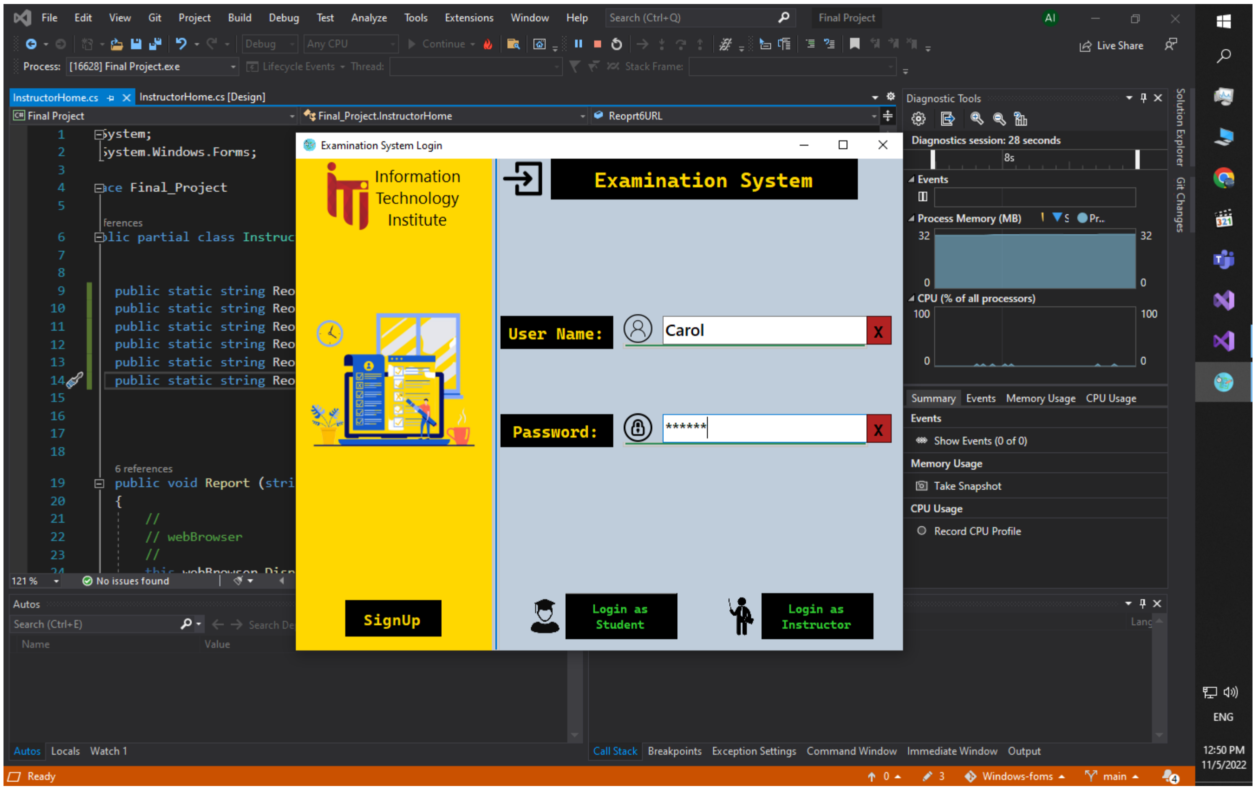
    -- Input---> Exam id, Student id
    -- Action--> 1. update the student question grade column in
                take exam table based on  the the student
                answer column
             2. calculate the student grade percentage and
                store it in  the student course table
    - Output--> return the student grade percentage in this
    exam



```sql



create procedure Exam_Correction (@e_id int, @s_id int)
as
begin
    update dbo.Take_Exam
    set SQ_Grade = St_Ans * q.Q_Grade
    from dbo.Question q, dbo.Take_Exam te
    where q.Q_Id = te.Q_Id
    and te.E_Id = @e_id
    and te.S_Id = @s_id;

    declare @stu_total_grade decimal(4, 1) = (select sum(te.SQ_Grade) from dbo.Take_Exam te
                                            where te.S_Id = @s_id
                                            and te.E_Id = @e_id
                                            and te.SQ_Grade is not Null)

    declare @exam_total_grade decimal(4, 1) = (select sum(q.Q_Grade)
                                             from dbo.Question q, dbo.Take_Exam te
                                             where q.Q_Id = te.Q_Id
                                             and te.E_Id = @e_id
                                             and te.S_Id = @s_id)

    declare @stu_exam_grade_percent decimal(4, 1)
    if @stu_total_grade != 0
    set @stu_exam_grade_percent = (@stu_total_grade/@exam_total_grade)*100
    else set @stu_exam_grade_percent = 0

    update dbo.Stu_Crs
    set Stu_Crs_Grade = @stu_exam_grade_percent
    where S_Id = @s_id
    and C_Id = (select e.C_Id from dbo.Exam e where e.E_Id = @e_id);
end

go

```

# 4- Desktop Application Phase

- To be able to build desktop applications we were required to use C# through windows forms.



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Final Project
Process: [16628] Final Project.exe
InstructorHome.cs
InstructorHome.cs [Design]
Final Project
Final_Project.InstructorHome
Reoprt6URL
Diagnostic Tools
Diagnostics session: 28 seconds
Events
Process Memory (MB)
CPU (% of all processors)
Summary Events Memory Usage CPU Usage
Show Events (0 of 0)
Take Snapshot
Record CPU Profile
Autos
Search (Ctrl+E)
Name
Value
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
121 %
No issues found
Ready
Windows-foms
main
Live Share
ENG
12:50 PM
11/5/2022

```
ystem;
ystem.Windows.Forms;

ace Final_Project

ferences
lic partial class Instruc


    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo




    6 references
    public void Report (stri
    {
        //
        // webBrowser
        //
```

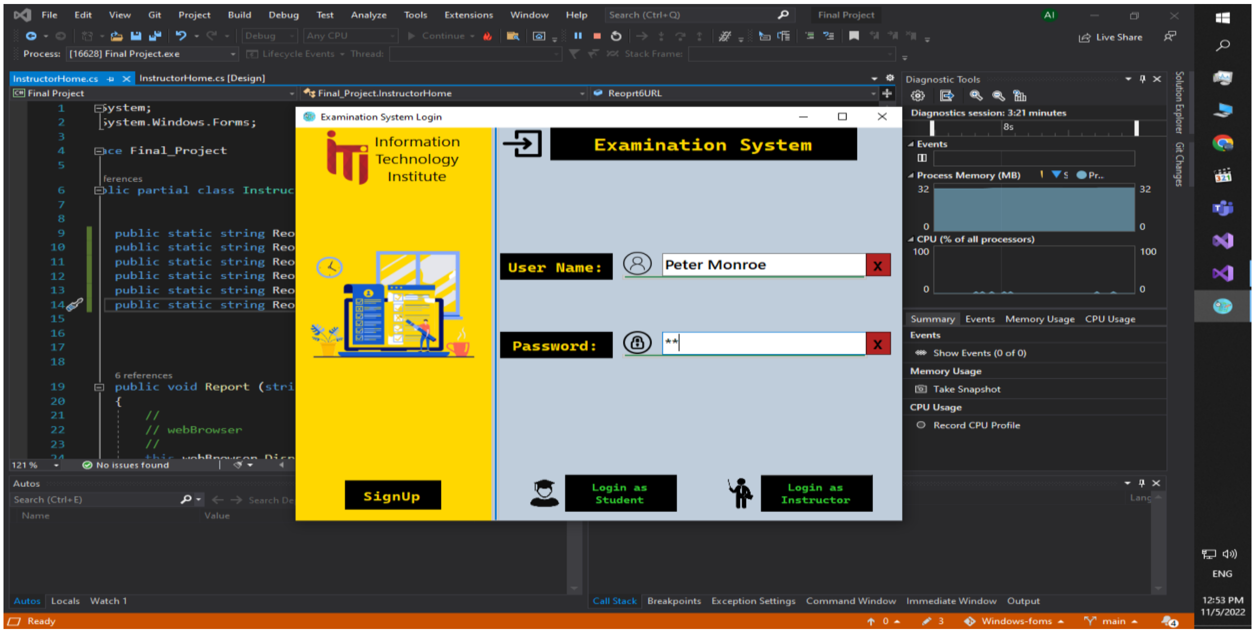
Examination System Login

Information Technology Institute
Examination System
SignUp
Login as Student
Login as Instructor

User Name: Carol

Password: ******

- A student or an instructor could log in to the desktop application.

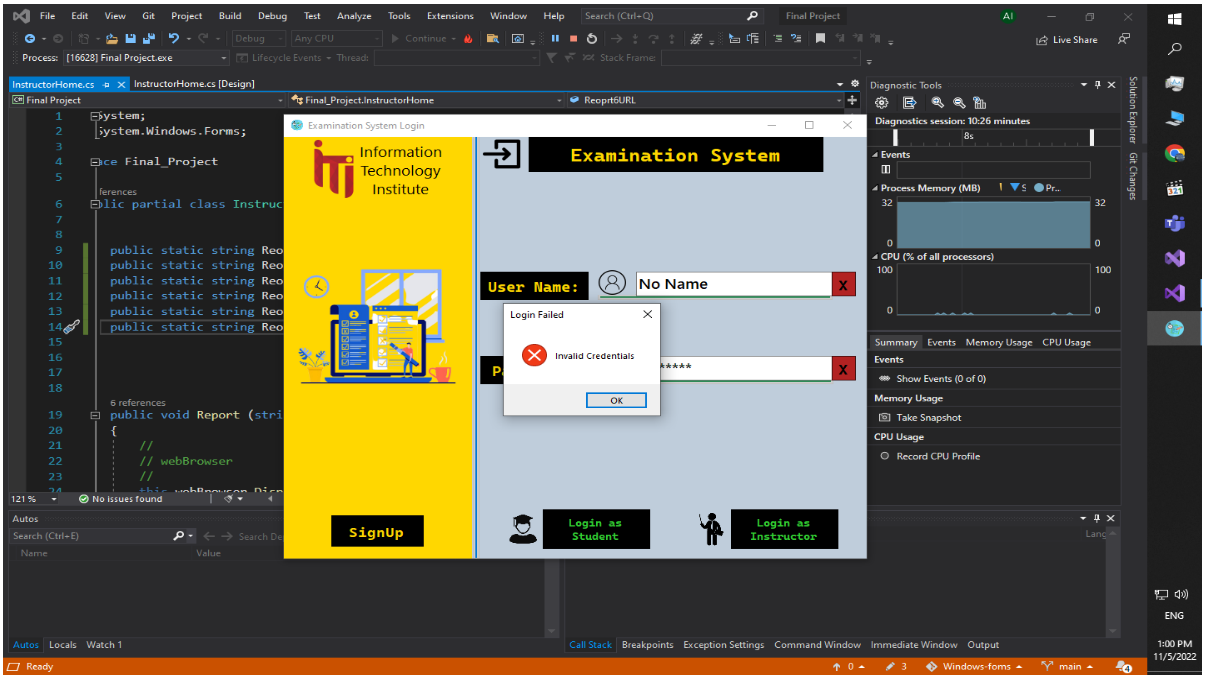


```
system;
ystem.Windows.Forms;

ce Final_Project

ferences
lic partial class Instruc

    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo
    public static string Reo

    6 references
    public void Report (stri
    {
        //
        // webBrowser
        //
```

User Name: Peter Monroe

Password: **

- As for Authentication, we build a form in which student and instructor could insert their username and password.



User Name: No Name

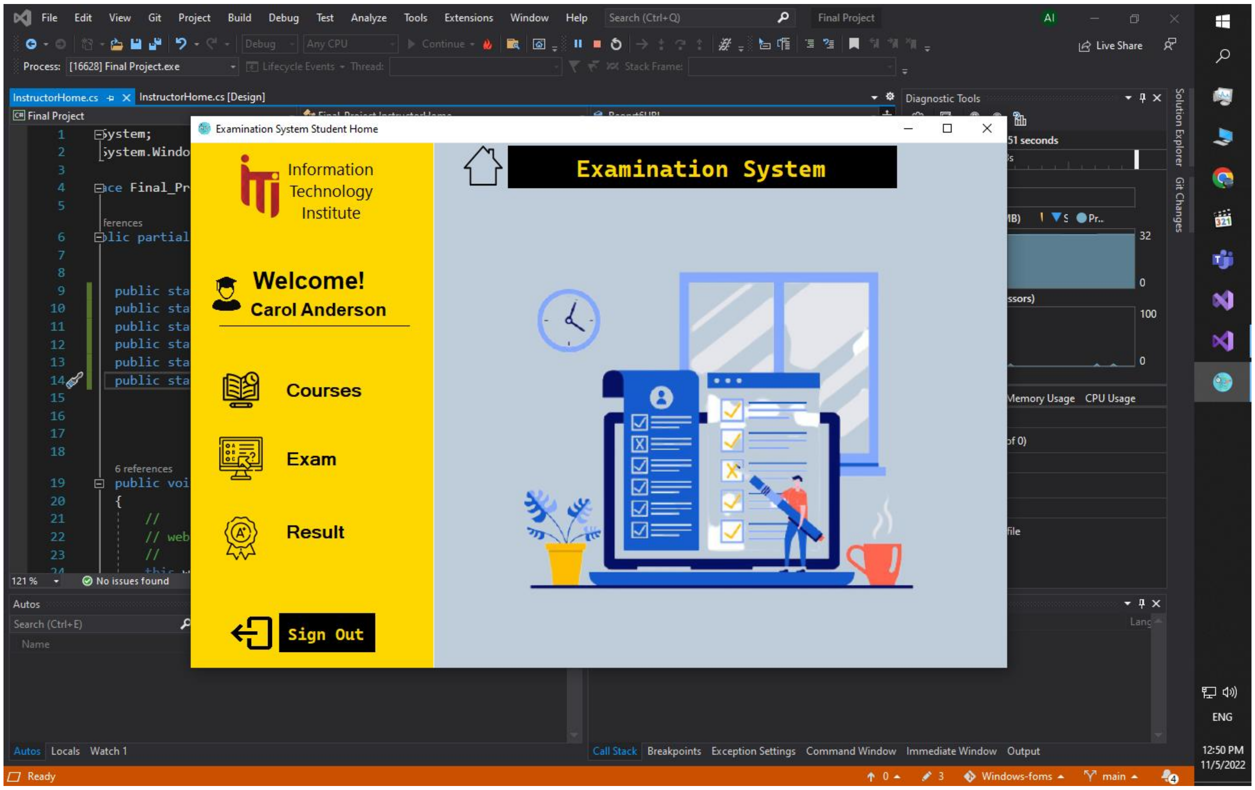
Password: *****

Login Failed

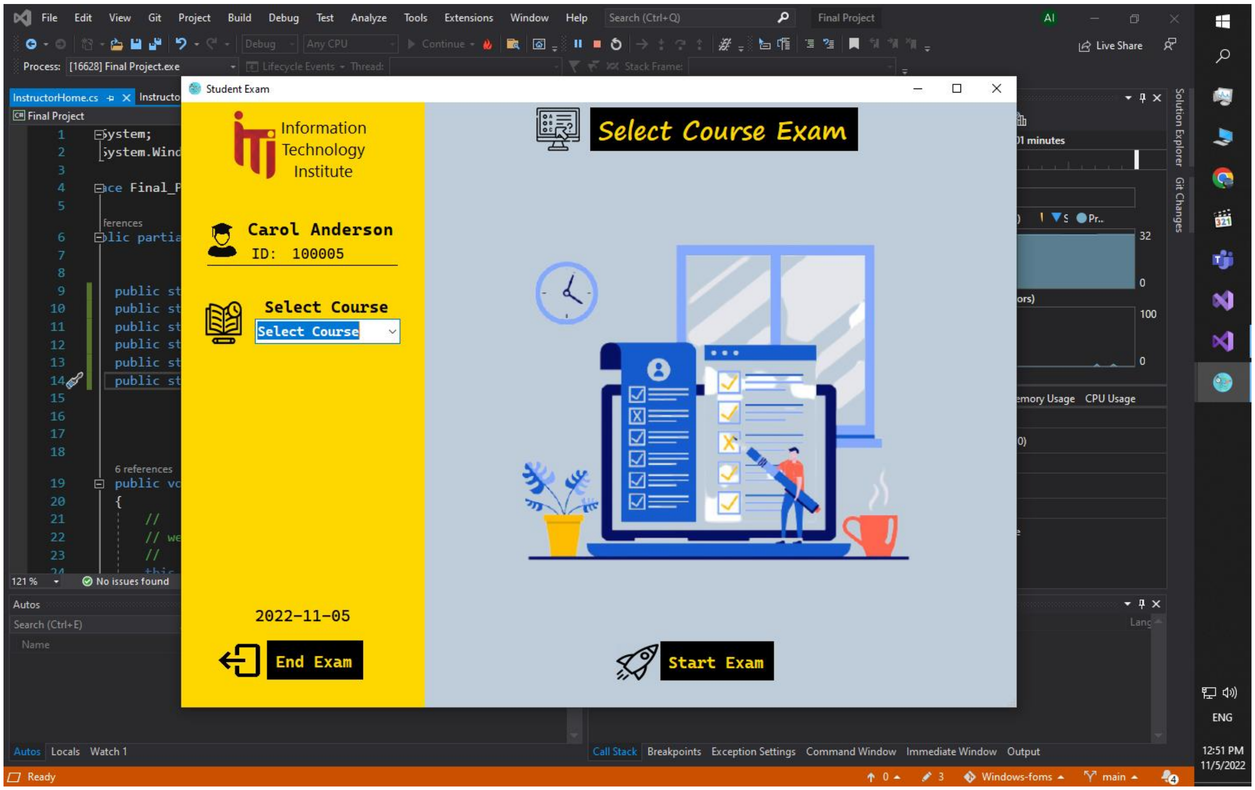
Invalid Credentials

OK

- As for students we chose the student's name to be his username and his id to be his password, this might make no sense but the idea was that this authentication was made by establishing a connection from the desktop application to the database(windows authentication) to check if there is a student with the username and id (at the same time (AND operator)), and similar was made for instructors.
- We build other forms to be student homes and instructor homes.
- A student could look at their grades and take exams from this form.



- Exam Form



```
System;
System.Wind
ce Final_P
ferences
lic partia
    public st
    public st
    public st
    public st
    public st
    public st
    6 references
    public vo
    {
        //
        // we
        //
```

Student Exam

Select Course Exam

Information Technology Institute

Carol Anderson
ID: 100005

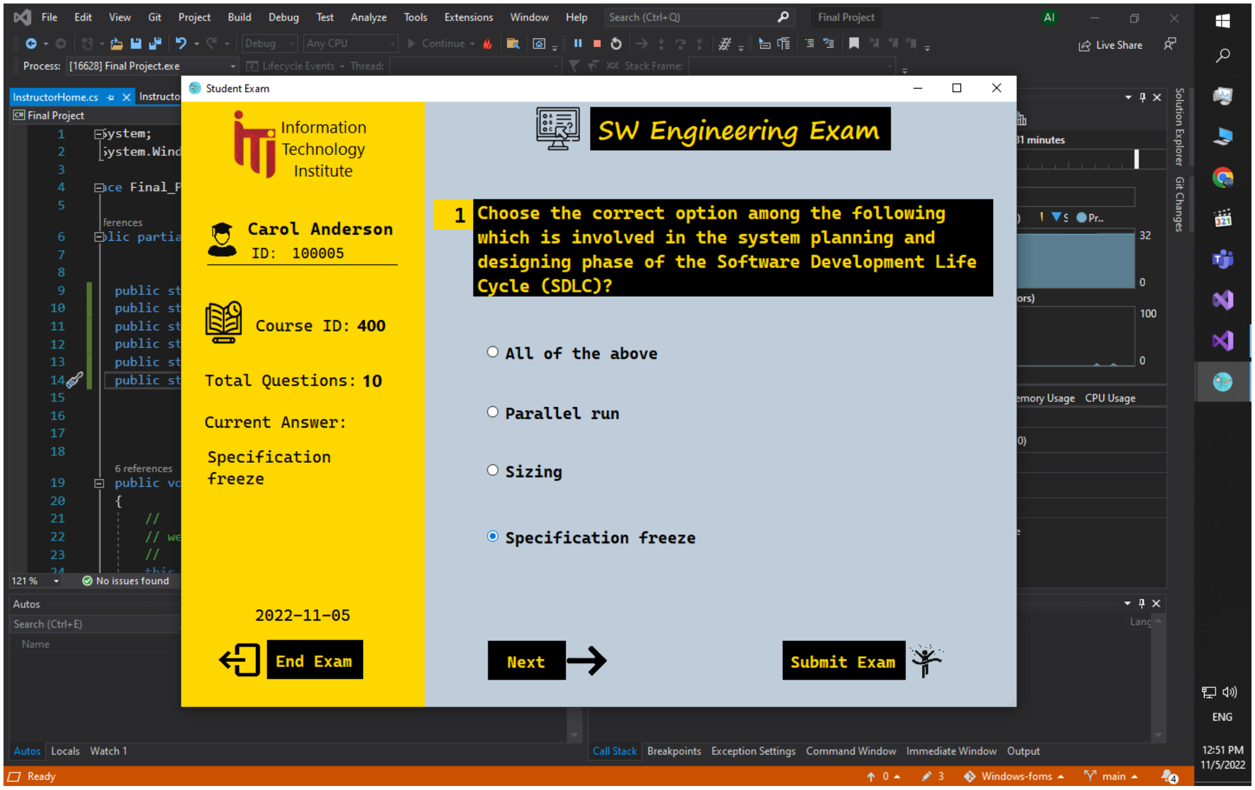
Select Course

Select Course

2022-11-05

End Exam

Start Exam



```
System;
System.Wind
ce Final_P
ferences
lic partia
    public st
    public st
    public st
    public st
    public st
    public st
    6 references
    public vo
    {
        //
        // we
        //
```

Student Exam

SW Engineering Exam

Information Technology Institute

Carol Anderson
ID: 100005

Course ID: 400

Total Questions: 10

Current Answer:

Specification freeze

1 Choose the correct option among the following which is involved in the system planning and designing phase of the Software Development Life Cycle (SDLC)?

○ All of the above

○ Parallel run

○ Sizing

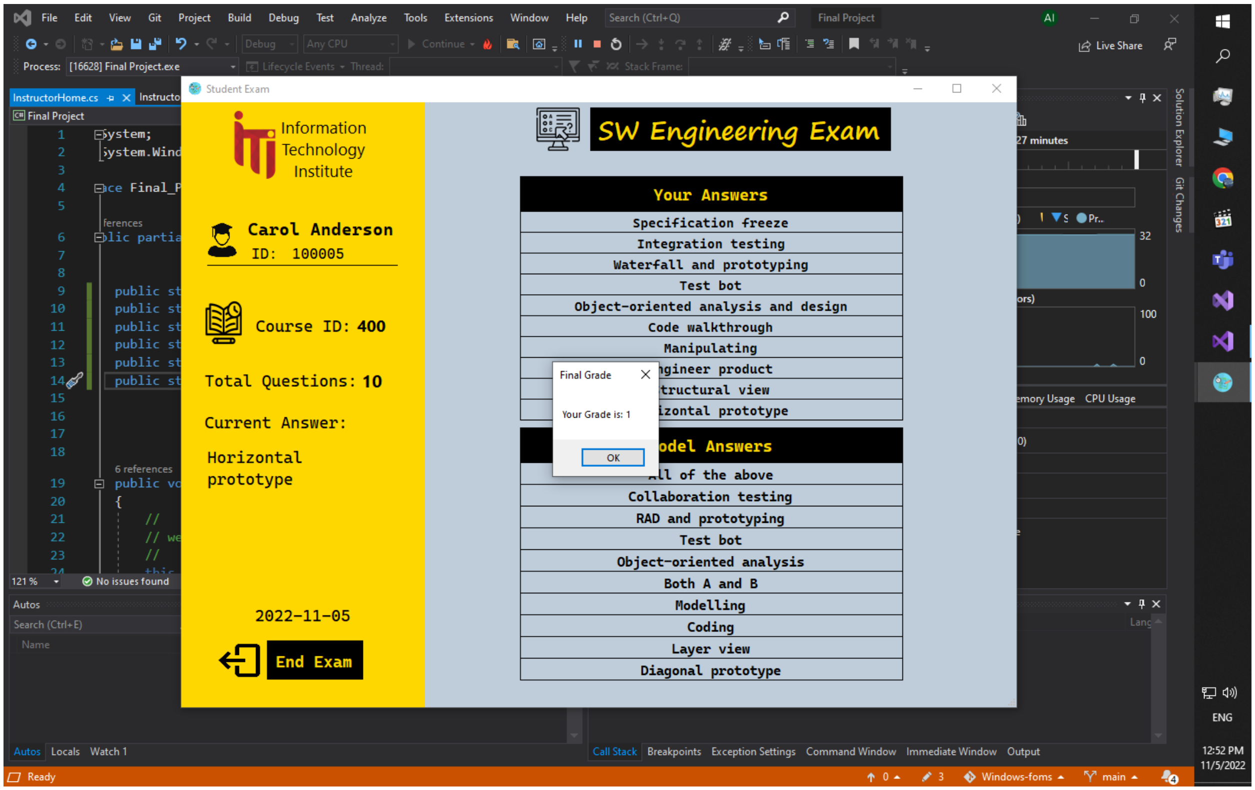
● Specification freeze

2022-11-05

End Exam

Next

Submit Exam



Student Exam

Information Technology Institute

Carol Anderson
ID: 100005

Course ID: 400

Total Questions: 10

Current Answer:

Horizontal prototype

2022-11-05

End Exam

## SW Engineering Exam

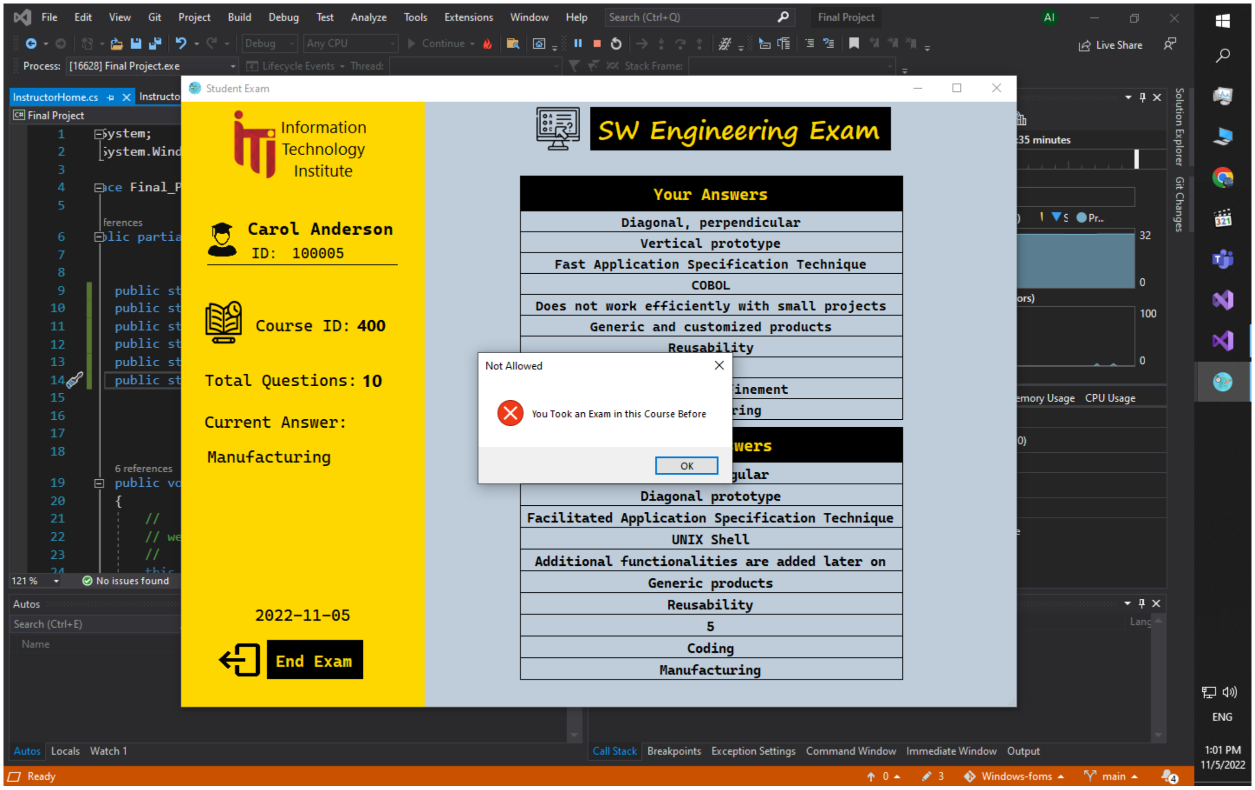
| Your Answers |
|---|
| Specification freeze |
| Integration testing |
| Waterfall and prototyping |
| Test bot |
| Object-oriented analysis and design |
| Code walkthrough |
| Manipulating |
| ngineer product |
| tructural view |
| izontal prototype |

| odel Answers |
|---|
| ll of the above |
| Collaboration testing |
| RAD and prototyping |
| Test bot |
| Object-oriented analysis |
| Both A and B |
| Modelling |
| Coding |
| Layer view |
| Diagonal prototype |

Final Grade

Your Grade is: 1

OK



Student Exam

Information Technology Institute

Carol Anderson
ID: 100005

Course ID: 400

Total Questions: 10

Current Answer:

Manufacturing

2022-11-05

End Exam

## SW Engineering Exam

| Your Answers |
|---|
| Diagonal, perpendicular |
| Vertical prototype |
| Fast Application Specification Technique |
| COBOL |
| Does not work efficiently with small projects |
| Generic and customized products |
| Reusability |
| |
| inement |
| ring |

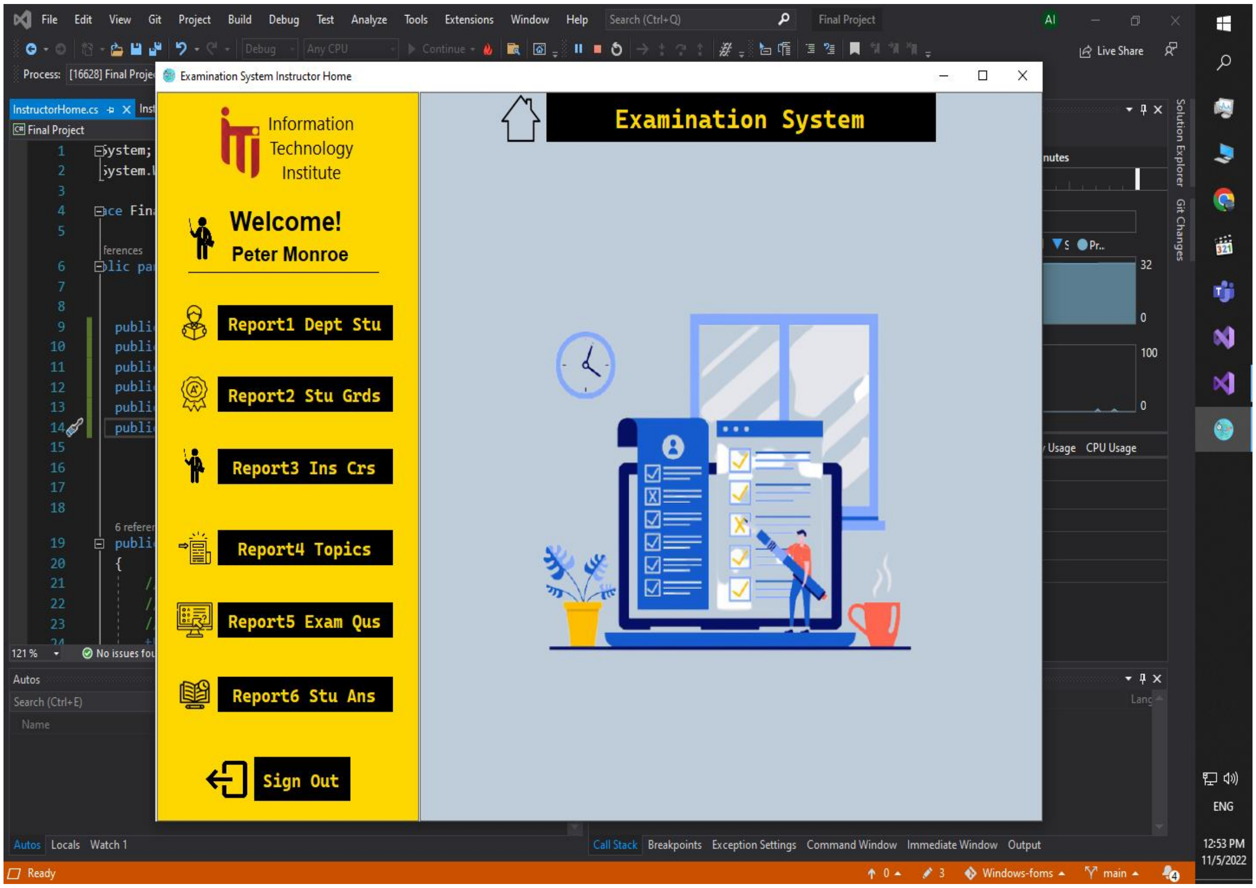
| wers |
|---|
| gular |
| Diagonal prototype |
| Facilitated Application Specification Technique |
| UNIX Shell |
| Additional functionalities are added later on |
| Generic products |
| Reusability |
| 5 |
| Coding |
| Manufacturing |

Not Allowed

You Took an Exam in this Course Before

OK

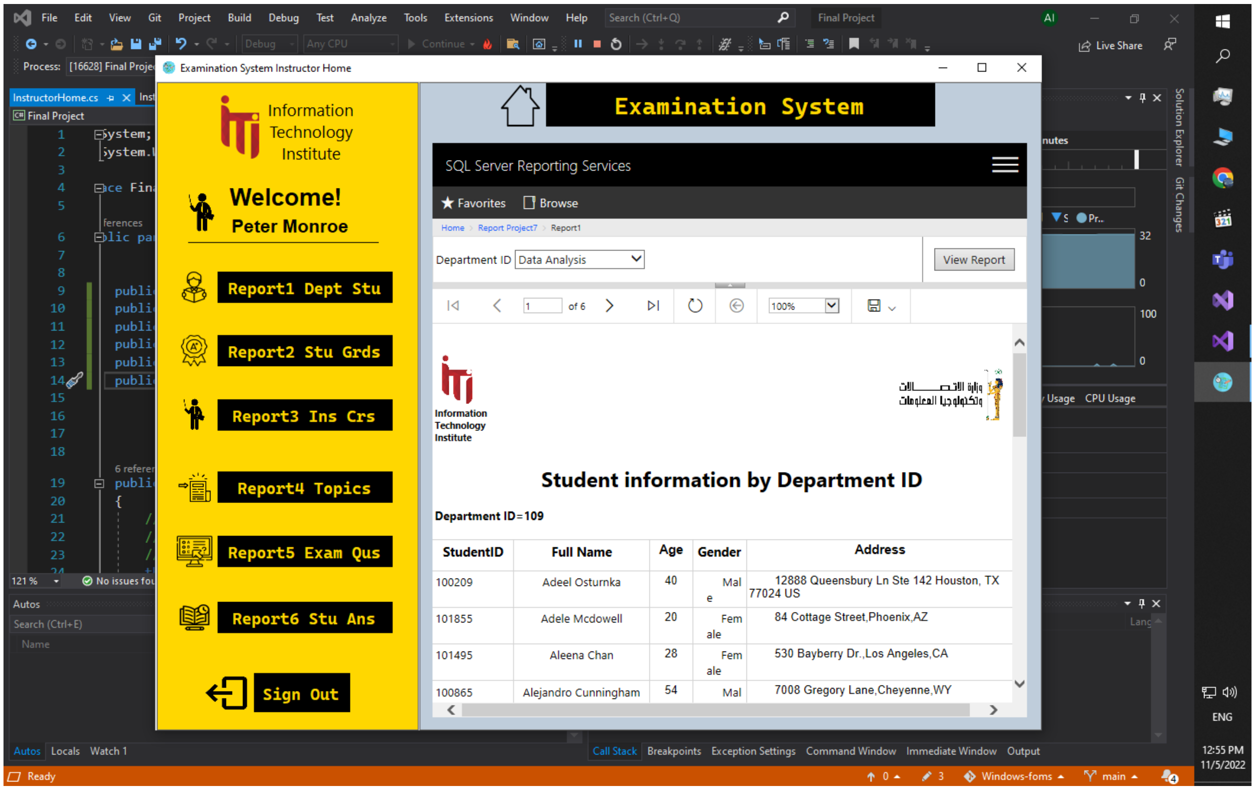
- An instructor had also an instructor home form from which he can display some reports and dashboards
- We have implemented 6 reports according to what is required, each of them has a stored procedure and parameters to return to us what is required


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Final Project
Live Share
Process: [16628] Final Proje
InstructorHome.cs
Examination System Instructor Home
Information Technology Institute
Welcome!
Peter Monroe
Examination System
Report1 Dept Stu
Report2 Stu Grds
Report3 Ins Crs
Report4 Topics
Report5 Exam Qus
Report6 Stu Ans
Sign Out
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
Windows-foms
main
12:53 PM
11/5/2022

# 5- SSRS Reports Phase

- Report 1

  The stored procedure called (Report1_Department_Student_Info) has a parameter (department id) that returns Student information by department id.



Department ID: Data Analysis

Information Technology Institute

وزارة الاتصـــــــالات وتكنولوجيا المعلومات

**Student information by Department ID**

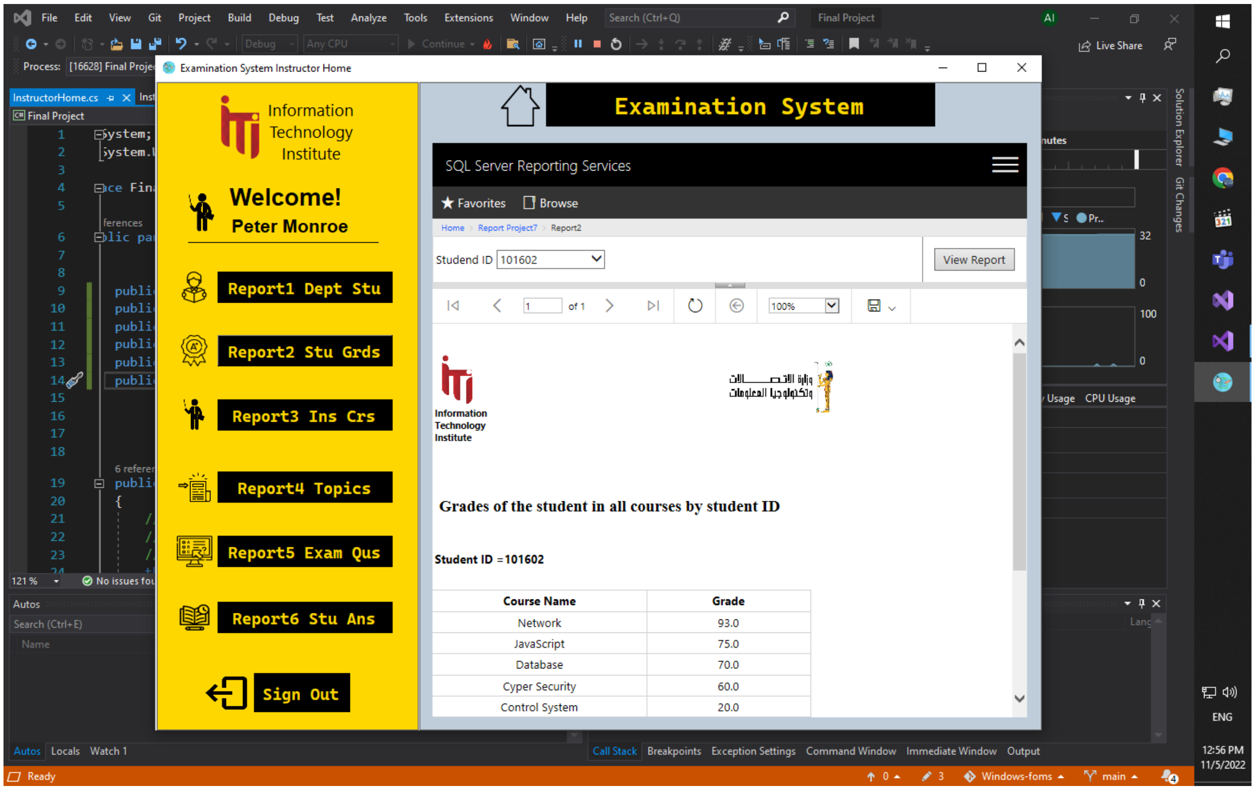
**Department ID=109**

| StudentID | Full Name | Age | Gender | Address |
|---|---|---|---|---|
| 100209 | Adeel Osturnka | 40 | Male | 12888 Queensbury Ln Ste 142 Houston, TX 77024 US |
| 101855 | Adele Mcdowell | 20 | Female | 84 Cottage Street,Phoenix,AZ |
| 101495 | Aleena Chan | 28 | Female | 530 Bayberry Dr.,Los Angeles,CA |
| 100865 | Alejandro Cunningham | 54 | Mal | 7008 Gregory Lane,Cheyenne,WY |

- Report 2

  The stored procedure called (Report2_Student_Grade_All_Courses) has a parameter (student id)

  that returns the grades of the student in all courses.



Studend ID 101602

Information Technology Institute

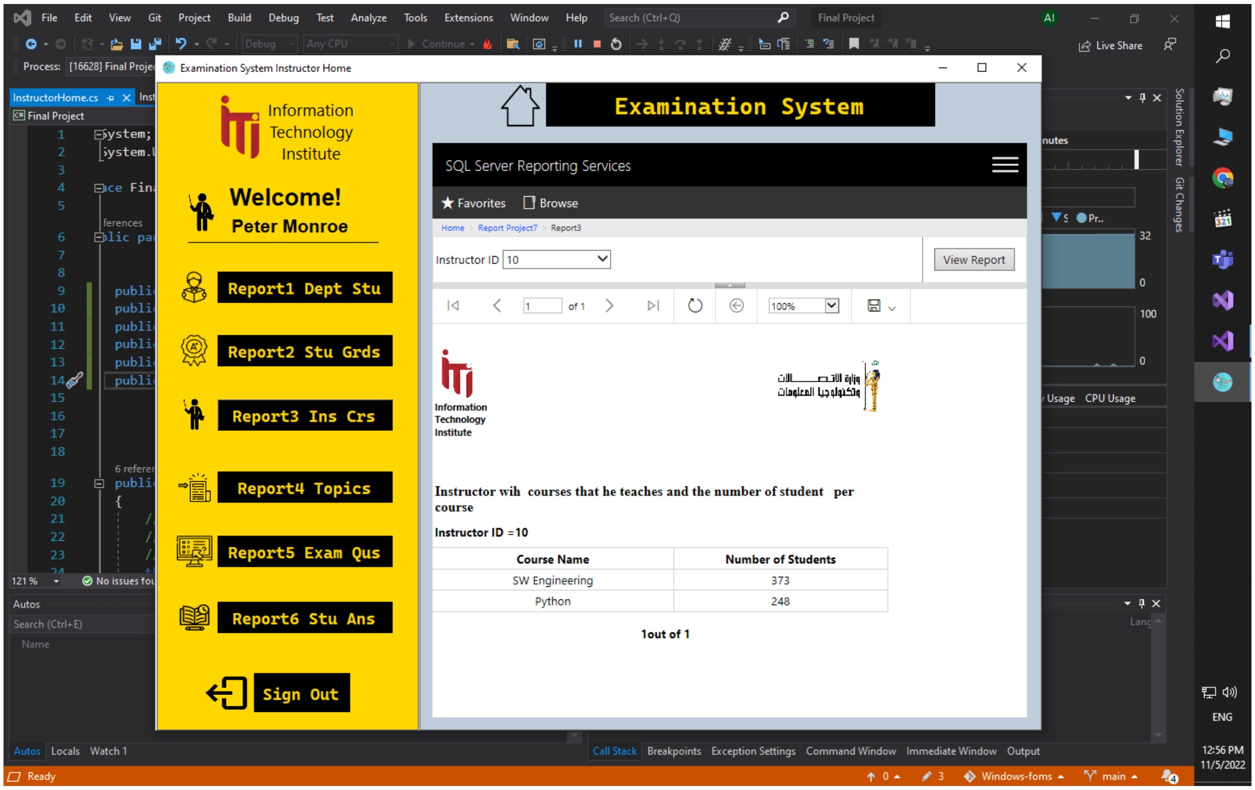
وزارة الاتصـــــــالات وتكنولوجيا المعلومات

Grades of the student in all courses by student ID

Student ID =101602

| Course Name | Grade |
|---|---|
| Network | 93.0 |
| JavaScript | 75.0 |
| Database | 70.0 |
| Cyper Security | 60.0 |
| Control System | 20.0 |

- Report 3
  The stored procedure called (Report3_Instructor_Course_Name_No_Students) has a parameter (instructor ID) that returns the name of the courses that he teaches and the number of students per course.



Instructor ID 10

Information Technology Institute

وزارة الاتصـــــــالات وتكنولوجيا المعلومات

**Instructor wih  courses that he teaches and the number of student   per course**

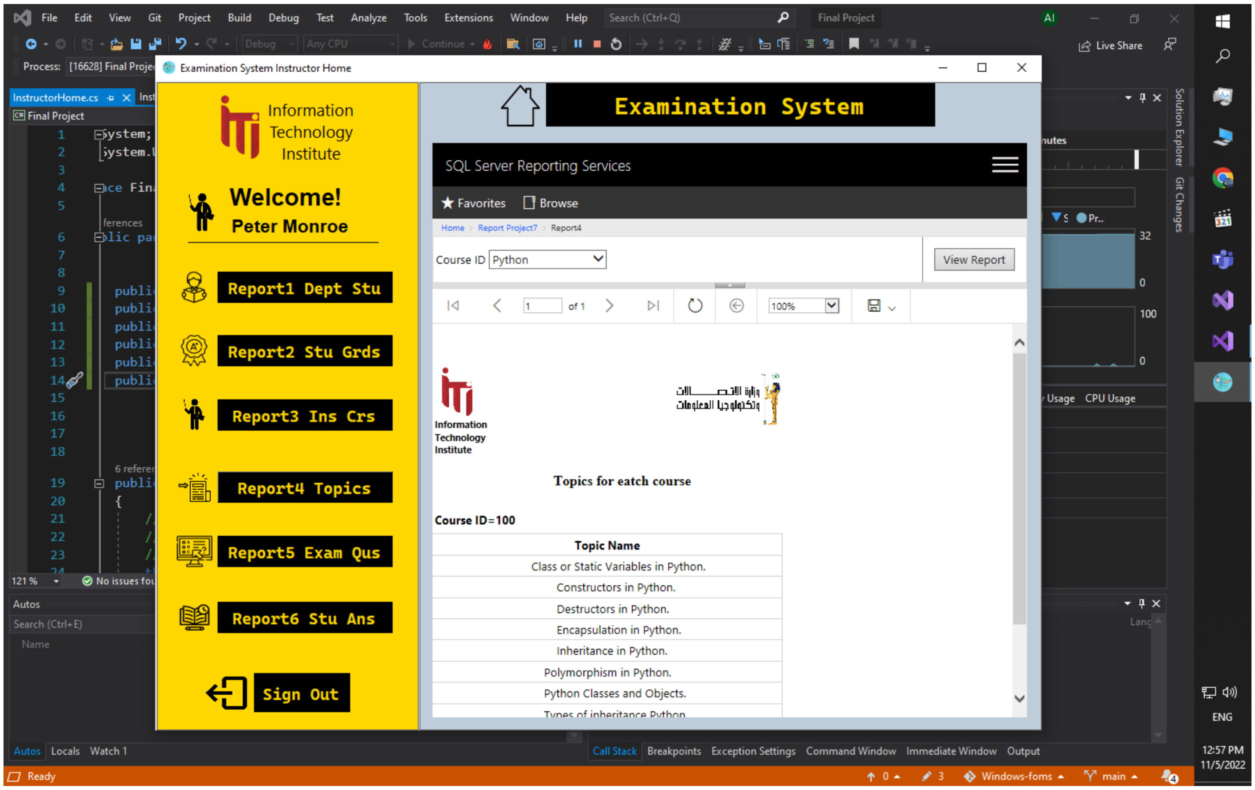
**Instructor ID =10**

| Course Name | Number of Students |
|---|---|
| SW Engineering | 373 |
| Python | 248 |

1out of 1

- Report 4

  The stored procedure called (Report4_Course_Topics) has a parameter (course id) that returns topics of its course.



Course ID: Python

Information Technology Institute

وزارة الاتصـــــــالات وتكنولوجيا المعلومات

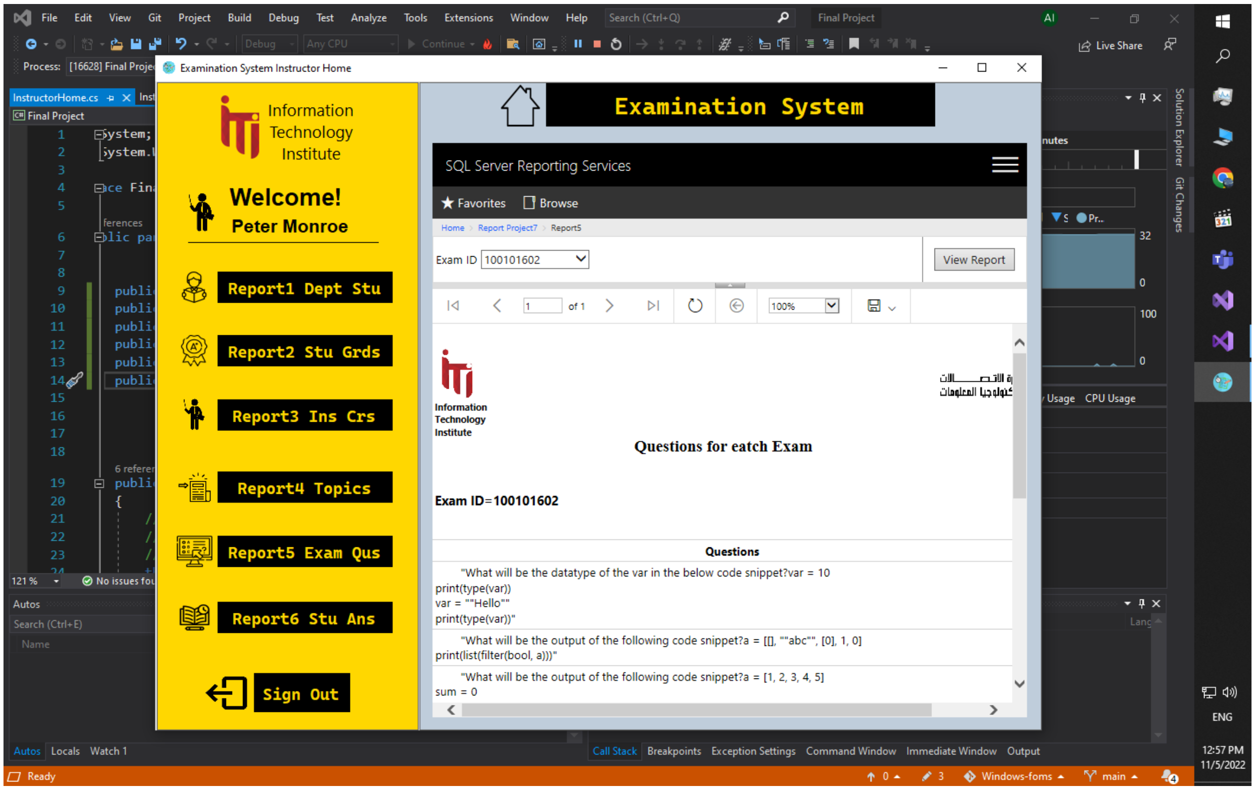
Topics for eatch course

Course ID=100

| Topic Name |
|---|
| Class or Static Variables in Python. |
| Constructors in Python. |
| Destructors in Python. |
| Encapsulation in Python. |
| Inheritance in Python. |
| Polymorphism in Python. |
| Python Classes and Objects. |
| Types of inheritance Python |

- Report 5

  The stored procedure called(Report5_Exam_Questions) has a parameter (exam id) that returns Questions in it.



Exam ID 100101602

Information Technology Institute

Questions for eatch Exam

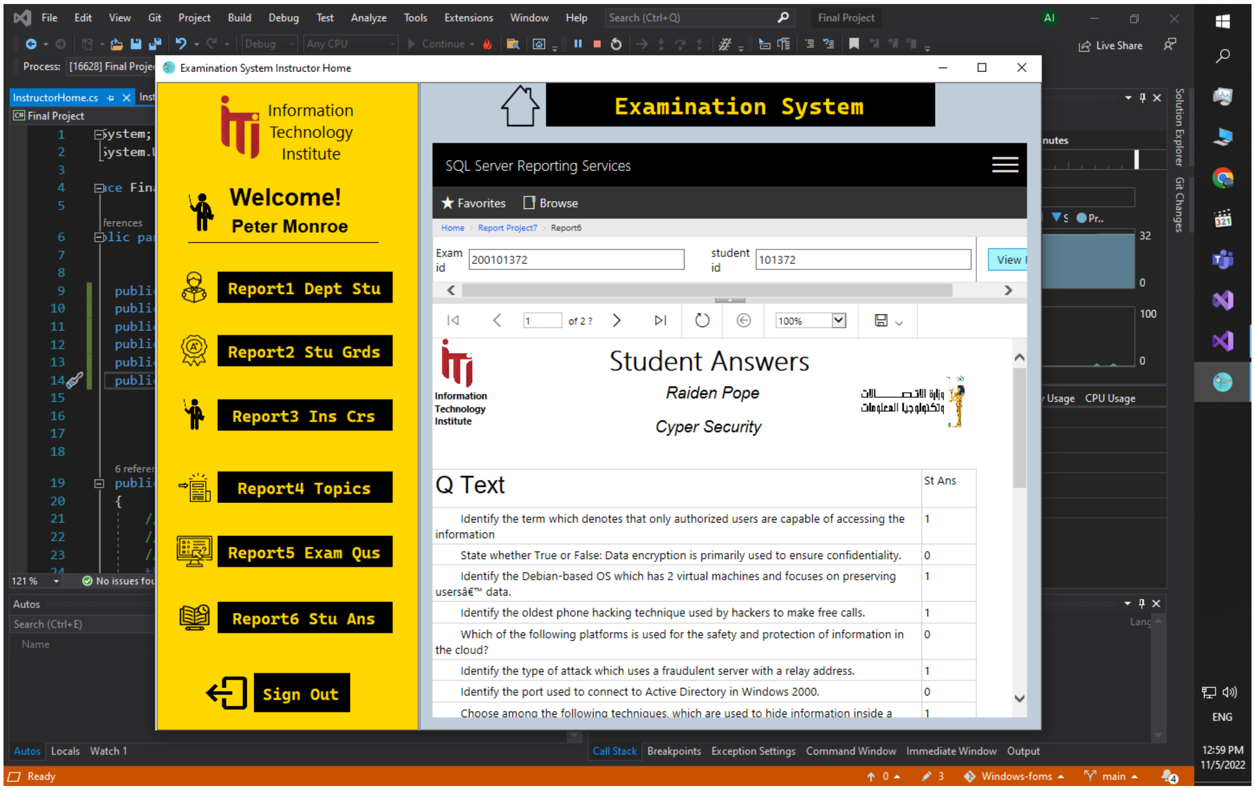
Exam ID=100101602

| Questions |
|---|
| "What will be the datatype of the var in the below code snippet?var = 10<br>print(type(var))<br>var = ""Hello""<br>print(type(var))" |
| "What will be the output of the following code snippet?a = [[], ""abc"", [0], 1, 0]<br>print(list(filter(bool, a)))" |
| "What will be the output of the following code snippet?a = [1, 2, 3, 4, 5]<br>sum = 0 |

- Report 6

  The stored procedure called(Report6_Exam_Student_Questions_Answers) that has 2 parameters

  (exam id) and (student id) that returns the Questions in this exam with the student's answers.



Exam id: 200101372

student id: 101372

Student Answers

Raiden Pope

Cyper Security

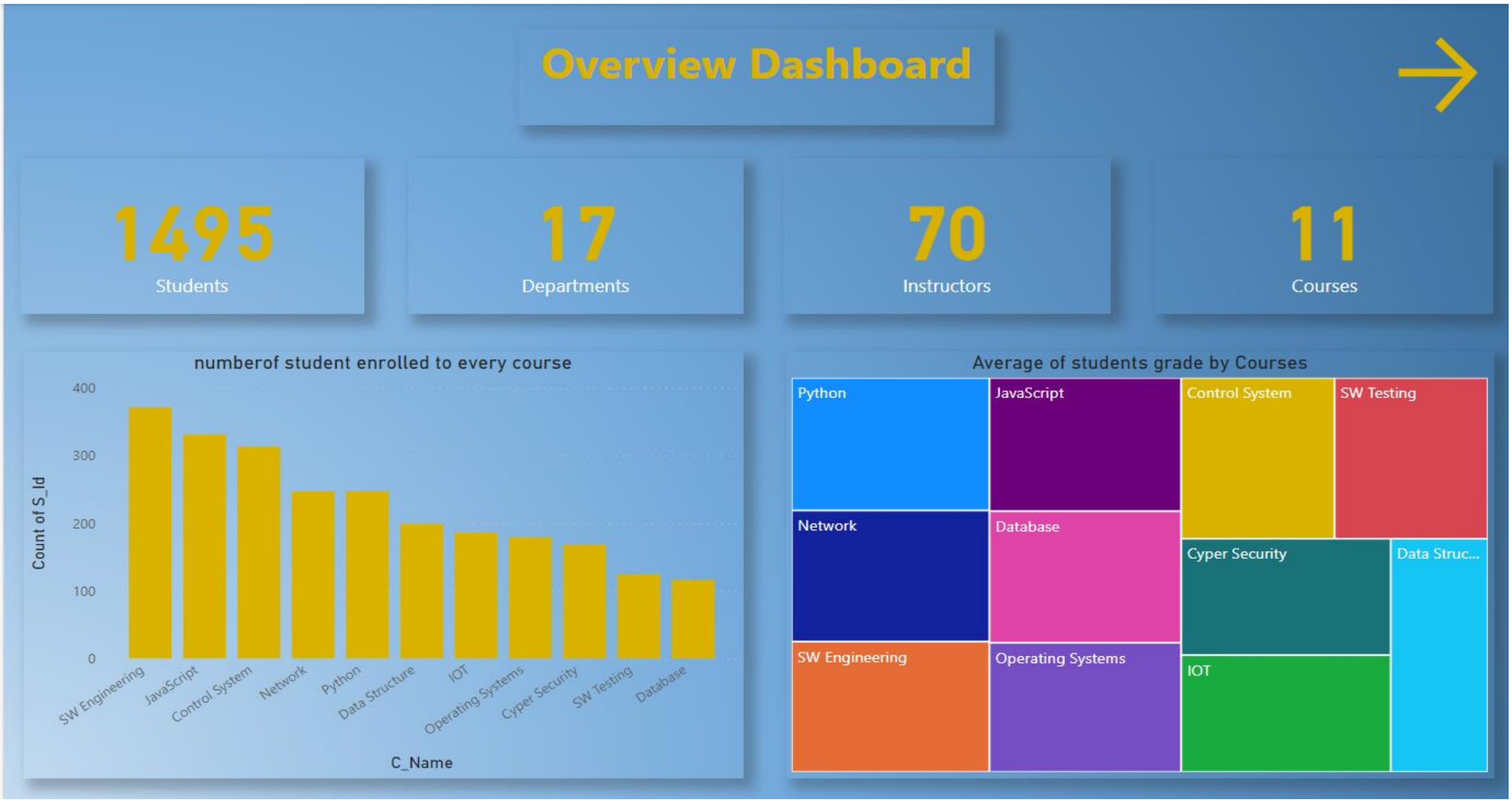
| Q Text | St Ans |
|---|---|
| Identify the term which denotes that only authorized users are capable of accessing the information | 1 |
| State whether True or False: Data encryption is primarily used to ensure confidentiality. | 0 |
| Identify the Debian-based OS which has 2 virtual machines and focuses on preserving usersâ€™ data. | 1 |
| Identify the oldest phone hacking technique used by hackers to make free calls. | 1 |
| Which of the following platforms is used for the safety and protection of information in the cloud? | 0 |
| Identify the type of attack which uses a fraudulent server with a relay address. | 1 |
| Identify the port used to connect to Active Directory in Windows 2000. | 0 |
| Choose among the following techniques, which are used to hide information inside a | 1 |

# 6- Dashboard Phase

for any system or business needs to be carefully monitored to make sure we are moving forward day by day, to see the system running in a harmony we build a Power bi dashboard, this is an implementation of the visualization magic, we choose to make two dashboards one for
an overview visualization and the other for relatively deep insights about the students

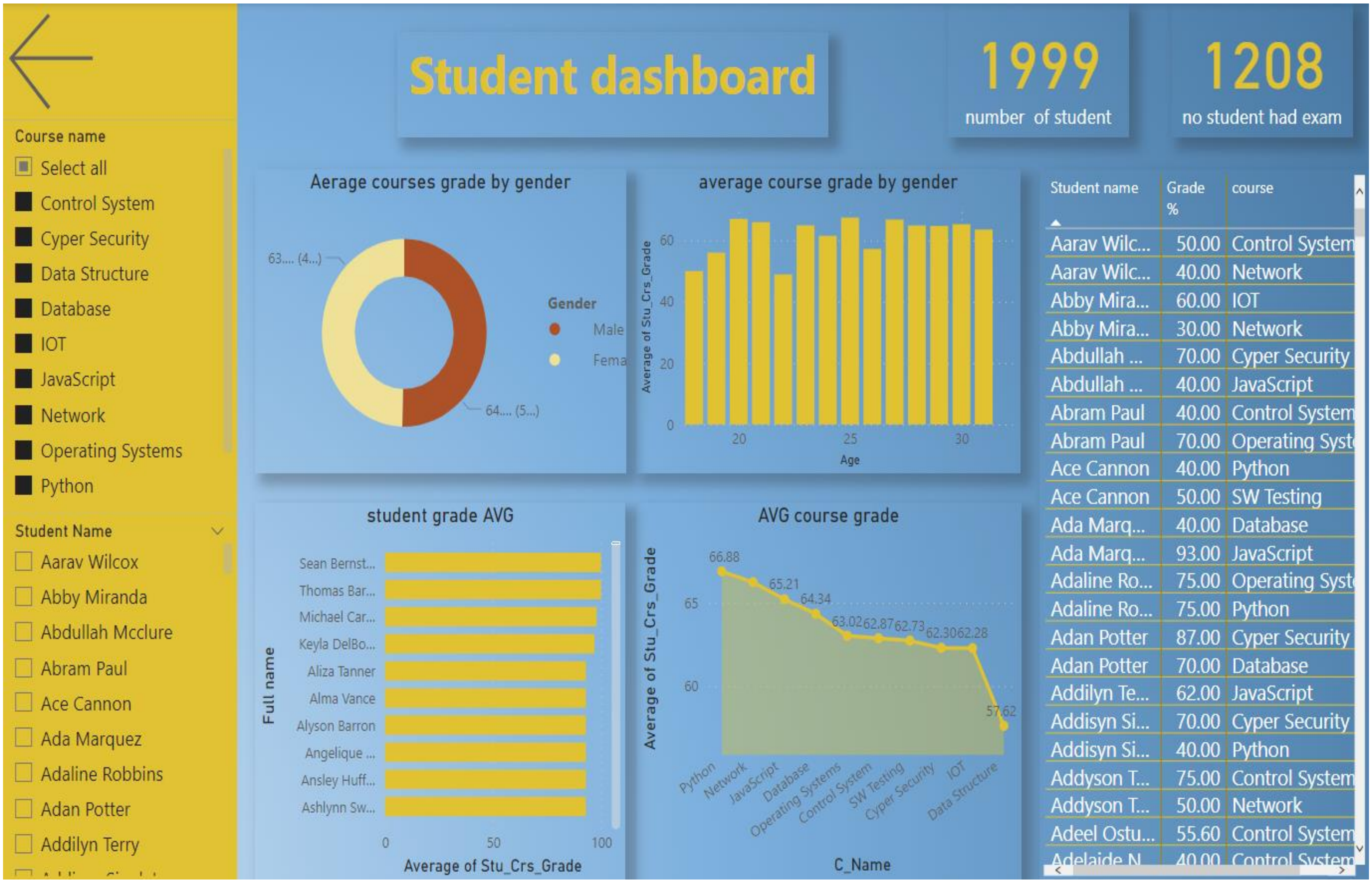
we tried to represent basic visuals such as bar plot and area plot, just to represent an overall how the examination system behaive.

Overview Dashboard:



at the student dashboard we deeply represented how students perform at there exams and highlight the best performing students, we also represented the courses grades if  to highlight the courses with low average grades, to dig deep on every student/course we added a slicer to represent each student and the another slicer to represent courses.

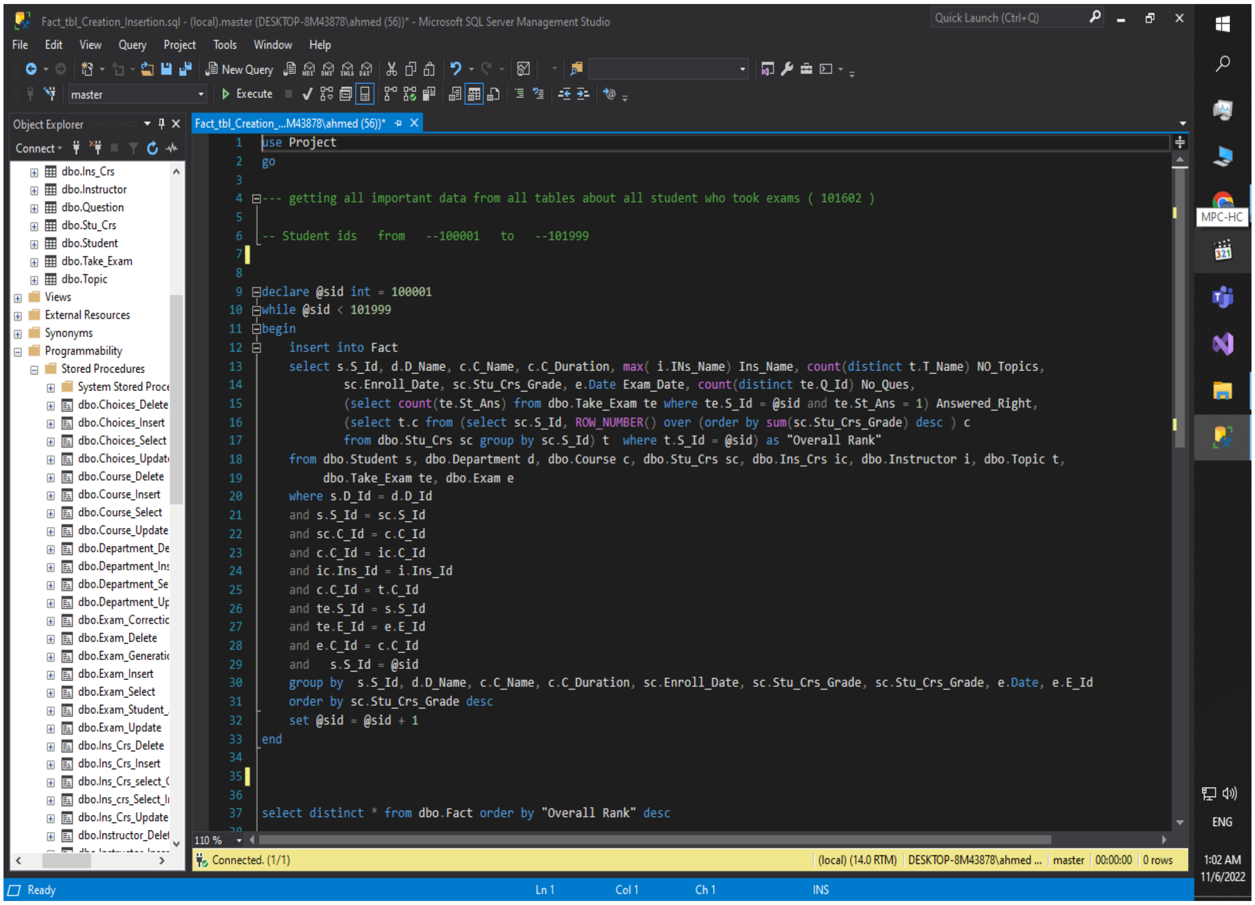
Student dashboard:



| Student name | Grade % | course |
|---|---|---|
| Aarav Wilc... | 50.00 | Control System |
| Aarav Wilc... | 40.00 | Network |
| Abby Mira... | 60.00 | IOT |
| Abby Mira... | 30.00 | Network |
| Abdullah ... | 70.00 | Cyper Security |
| Abdullah ... | 40.00 | JavaScript |
| Abram Paul | 40.00 | Control System |
| Abram Paul | 70.00 | Operating Syst |
| Ace Cannon | 40.00 | Python |
| Ace Cannon | 50.00 | SW Testing |
| Ada Marq... | 40.00 | Database |
| Ada Marq... | 93.00 | JavaScript |
| Adaline Ro... | 75.00 | Operating Syst |
| Adaline Ro... | 75.00 | Python |
| Adan Potter | 87.00 | Cyper Security |
| Adan Potter | 70.00 | Database |
| Addilyn Te... | 62.00 | JavaScript |
| Addisyn Si... | 70.00 | Cyper Security |
| Addisyn Si... | 40.00 | Python |
| Addyson T... | 75.00 | Control System |
| Addyson T... | 50.00 | Network |
| Adeel Ostu... | 55.60 | Control System |
| Adelaide N... | 40.00 | Control System |

To analyze and visualize all these data and to be able to get all the data about a student in the DB, We created a SQL Script that joins nearly all the tables in the DB, Which we used in creating a Fact table.

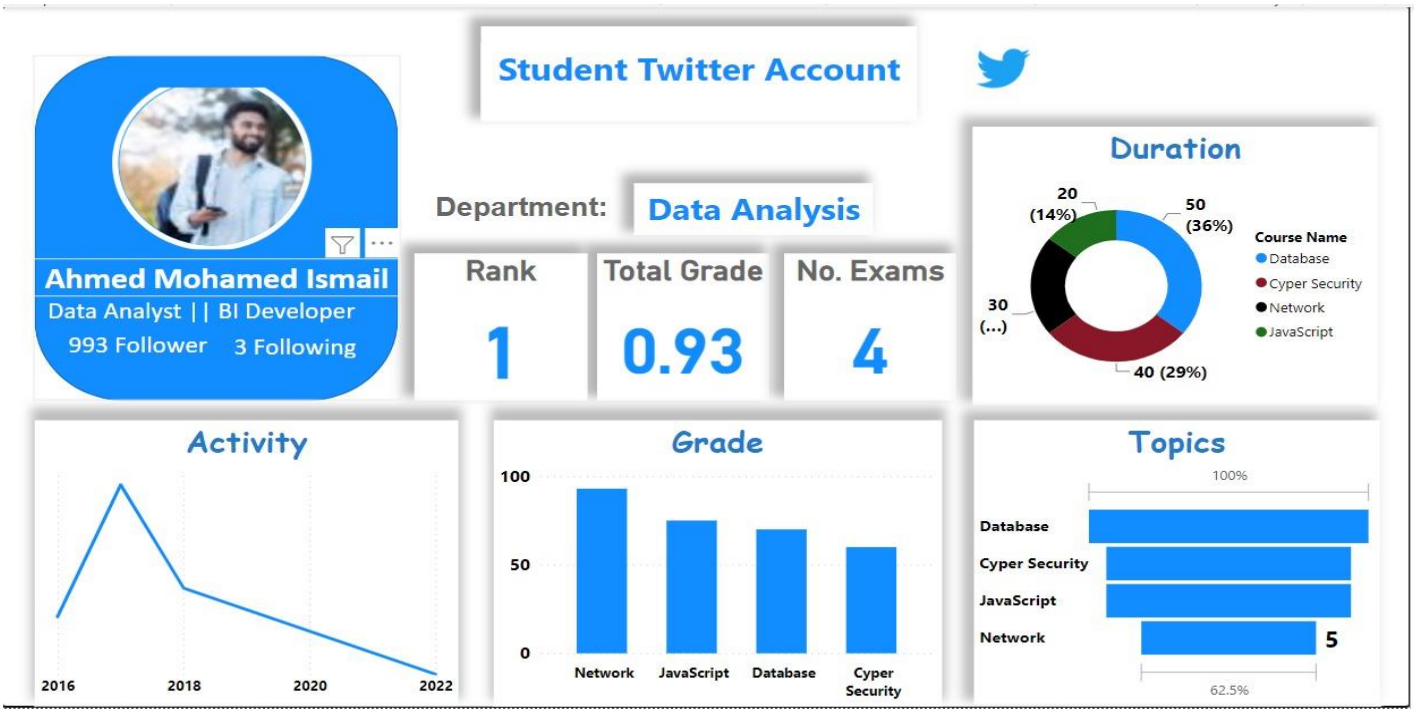
Fact Table SQL Script:



```sql
use Project
go

--- getting all important data from all tables about all student who took exams ( 101602 )

-- Student ids   from   --100001   to   --101999


declare @sid int = 100001
while @sid < 101999
begin
    insert into Fact
    select s.S_Id, d.D_Name, c.C_Name, c.C_Duration, max( i.INs_Name) Ins_Name, count(distinct t.T_Name) NO_Topics,
            sc.Enroll_Date, sc.Stu_Crs_Grade, e.Date Exam_Date, count(distinct te.Q_Id) No_Ques,
            (select count(te.St_Ans) from dbo.Take_Exam te where te.S_Id = @sid and te.St_Ans = 1) Answered_Right,
            (select t.c from (select sc.S_Id, ROW_NUMBER() over (order by sum(sc.Stu_Crs_Grade) desc ) c
            from dbo.Stu_Crs sc group by sc.S_Id) t  where t.S_Id = @sid) as "Overall Rank"
    from dbo.Student s, dbo.Department d, dbo.Course c, dbo.Stu_Crs sc, dbo.Ins_Crs ic, dbo.Instructor i, dbo.Topic t,
         dbo.Take_Exam te, dbo.Exam e
    where s.D_Id = d.D_Id
    and s.S_Id = sc.S_Id
    and sc.C_Id = c.C_Id
    and c.C_Id = ic.C_Id
    and ic.Ins_Id = i.Ins_Id
    and c.C_Id = t.C_Id
    and te.S_Id = s.S_Id
    and te.E_Id = e.E_Id
    and e.C_Id = c.C_Id
    and   s.S_Id = @sid
    group by  s.S_Id, d.D_Name, c.C_Name, c.C_Duration, sc.Enroll_Date, sc.Stu_Crs_Grade, sc.Stu_Crs_Grade, e.Date, e.E_Id
    order by sc.Stu_Crs_Grade desc
    set @sid = @sid + 1
end



select distinct * from dbo.Fact order by "Overall Rank" desc
```

some times social media activities affects how the student perform at there exams, therefore we decided to link the students twitter accounts to discuss if there any relation between their activities and their grades.

Twitter Dashboard:



Student Twitter Account

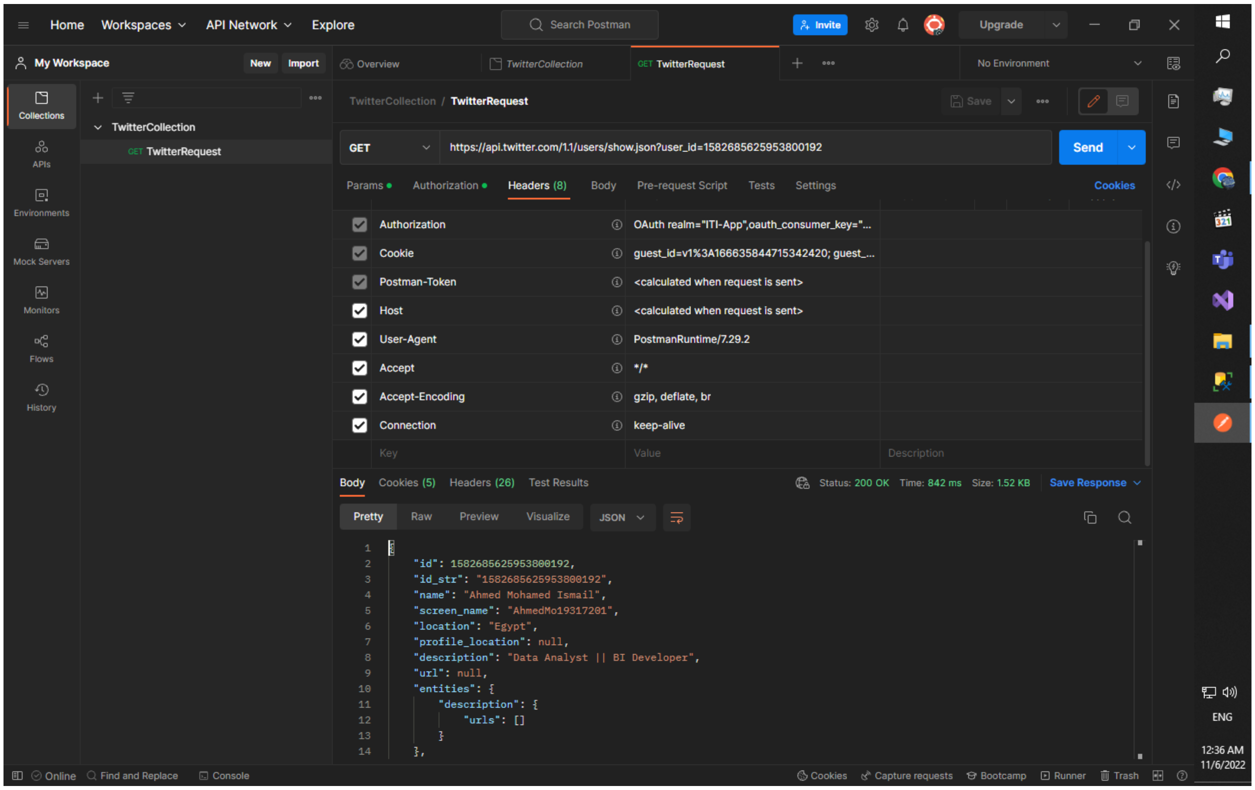
Ahmed Mohamed Ismail
Data Analyst || BI Developer
993 Follower 3 Following

Department: Data Analysis

| Rank | Total Grade | No. Exams |
|---|---|---|
| 1 | 0.93 | 4 |

We used Postman Program to construct the API request and the authorization header that will be used in Power BI to connect to Twitter API.

# Postman API Request:



TwitterCollection / TwitterRequest

GET https://api.twitter.com/1.1/users/show.json?user_id=1582685625953800192

| | Key | Value | Description |
|---|---|---|---|
| ☑ | Authorization | OAuth realm="ITI-App",oauth_consumer_key="... | |
| ☑ | Cookie | guest_id=v1%3A166635844715342420; guest_... | |
| ☑ | Postman-Token | <calculated when request is sent> | |
| ☑ | Host | <calculated when request is sent> | |
| ☑ | User-Agent | PostmanRuntime/7.29.2 | |
| ☑ | Accept | */* | |
| ☑ | Accept-Encoding | gzip, deflate, br | |
| ☑ | Connection | keep-alive | |

Body Cookies (5) Headers (26) Test Results

Status: 200 OK Time: 842 ms Size: 1.52 KB

```json
{
    "id": 1582685625953800192,
    "id_str": "1582685625953800192",
    "name": "Ahmed Mohamed Ismail",
    "screen_name": "AhmedMo19317201",
    "location": "Egypt",
    "profile_location": null,
    "description": "Data Analyst || BI Developer",
    "url": null,
    "entities": {
        "description": {
            "urls": []
        }
    },
```