

The process I took to import this csv to a database is by utilizing an app to generate an SQL script from the csv and executing it in the database. Here is the process:

First, I exported the csv file into an excel spreadsheet to test if the data has been cleaned and no errors are found.

The screenshot shows a Microsoft Excel spreadsheet titled "Unisa Portal Final.xlsx". The table contains the following data:

ID	StartTime	CompTime	Email	Name	StdNum	ModCode	Integrity	DocType	M1	M2	M3
743	2021/04/14 21:56	2021/04/14 22:03	51460386@mylife.unisa.ac.za	VERONICA KEROMAMANG PULE	51460386	AAH6522	No	Document Upload			
845	2021/04/22 19:17	2021/04/23 01:26	53886879@mylife.unisa.ac.za	BOKAMOSO PAAPAA SEGAGE	53886879	AAK3212	No	Fill-in			
865	2021/04/26 23:10	2021/04/26 23:35	32901046@mylife.unisa.ac.za	N MASEKO	32901046	AAM7285	Yes	MCQ	B	C	
618	2021/03/11 22:27	2021/03/11 22:28	40080269@mylife.unisa.ac.za	DANIELABRAHAM LOUW	40080269	AAN0650	Yes	MCQ	B	A	B
288	2021/02/16 12:47	2021/02/16 12:52	54709695@mylife.unisa.ac.za	TA LUNGA	54709695	ABE3371	Yes	MCQ	A	C	B
640	2021/03/25 21:31	2021/03/25 21:34	34802193@mylife.unisa.ac.za	NDIDZULAFHI SMIKE NENZELELE	34802193	ABIS523	Yes	Fill-in			
802	2021/04/19 22:30	2021/04/19 22:35	49690493@mylife.unisa.ac.za	P N MAHLANGU	49690493	ABT2635	Yes	Fill-in			
739	2021/04/14 15:35	2021/04/14 15:39	49391933@mylife.unisa.ac.za	SIVUYILE HAPPY KWAZA	49391933	ABX6127	Yes	Fill-in			
964	2021/05/03 11:44	2021/05/03 11:46	53885767@mylife.unisa.ac.za	Silvester Mantji Mamabolo	53885767	ACC1502	Yes	MCQ	B	C	D
981	2021/05/03 13:06	2021/05/03 13:09	51300265@mylife.unisa.ac.za	M N MARIPANE	51300265	ACC6079	Yes	Fill-in			
34	2021/01/12 14:04	2021/01/12 14:07	56661118@mylife.unisa.ac.za	RICCARDO RESSA	56661118	ACR5568	Yes	Fill-in			
337	2021/02/22 22:27	2021/02/22 22:37	31518567@mylife.unisa.ac.za	William Ramogasane Matlala	31518567	ACT2300	Yes	Document Upload			
500	2021/03/02 16:34	2021/03/02 16:38	48783854@mylife.unisa.ac.za	ZUKISWA VIVIAN MAPEVI	48783854	ADF1908	No	MCQ	A	B	C
491	2021/03/02 11:46	2021/03/02 11:49	47655453@mylife.unisa.ac.za	LEDWABA VICTOR MASHIANE	47655453	ADH8006	Yes	MCQ	A	A	A
1259	2021/05/14 23:30	2021/05/14 23:33	55603378@mylife.unisa.ac.za	MAMPE ISABELLA MAMPURU	55603378	ADL2602	Yes	MCQ	A	A	C
1078	2021/05/10 05:55	2021/05/10 05:58	54362474@mylife.unisa.ac.za	ALEX KAMOGELO MANAKA	54362474	ADY5584	Yes	Document Upload			
510	2021/03/03 21:54	2021/03/21:57	54657628@mylife.unisa.ac.za	Lehlohonolo Mathewus Lekhelebane	54657628	AEP2721	Yes	MCQ	B	A	C
692	2021/04/10 06:30	2021/04/10 06:35	54511909@mylife.unisa.ac.za	SISANDA KIMBILI	54511909	AFO5530	Yes	Document Upload			

I exported this excel file into Microsoft's Access for the file to be read in Microsoft Visual Basic 2015.

The screenshot shows a Microsoft Access database window with a table named "Exam". The table structure is identical to the one in the Excel file, with columns for ID, Start Time, Comp Time, Email, Name, StdNum, ModCode, Integrity, Obsolete, DocType, and Question. The data in the Access table is as follows:

ID	Start Time	Comp Time	Email	Name	StdNum	ModCode	Integrity	Obsolete	DocType	Question
1	2021/01/07 08:55:21	2021/01/07 09:16:06	47576545@my	DUSTIN BOOYS	47576545	DMA3675	Yes	MCQ	A	
2	2021/01/07 09:29:37	2021/01/07 09:29:55	47576545@my	DUSTIN BOOYS	47576545	AVL0582	Yes	Fill-in		
3	2021/01/07 09:33:03	2021/01/07 09:36:58	47576545@my	DUSTIN BOOYS	47576545	ADM1924	No	Document Upd		
4	2021/01/07 09:34:36	2021/01/07 09:36:58	47576545@my	DUSTIN BOOYS	47576545	SNF0001	Yes	Document Upd		
5	2021/01/08 13:07:27	2021/01/08 13:07:38	44974825@my	Gary Harry Cun	44974825	ICT3715	Yes	Document Upd		
6	2021/01/10 00:30:20	2021/01/10 00:48:48	54811163@my	Hendrik Abrah	54811163	ILCT3715	Yes	Document Upd		
7	2021/01/10 00:58:50	2021/01/10 00:58:55	54811163@my	Hendrik Abrah	54811163	LNF8731	Yes	Fill-in		
8	2021/01/10 10:13:35	2021/01/10 10:16:32	54811163@my	Hendrik Abrah	54811163	IC3691	Yes	Document Upd		
9	2021/01/10 10:16:35	2021/01/10 10:18:34	54811163@my	Hendrik Abrah	54811163	IC1425	Yes	MCQ	A	
10	2021/01/10 15:58:37	2021/01/10 16:14:38	34576851@my	Craig Brickhill	34576851	INF3612	Yes	Fill-in		
11	2021/01/10 16:14:43	2021/01/10 16:19:55	34576851@my	Craig Brickhill	34576851	ITV0353	No	MCQ	A	
12	2021/01/10 16:19:59	2021/01/10 16:29:27	34576851@my	Craig Brickhill	34576851	KUK2442	Yes	Document Upd		
13	2021/01/10 16:46:43	2021/01/10 16:52:27	34576851@my	Craig Brickhill	34576851	FGA7301	Yes	Fill-in		
14	2021/01/10 22:52:33	2021/01/10 22:08:05	34576851@my	Craig Brickhill	34576851	HLU8077	Yes	MCQ	A	
15	2021/01/10 19:25:24	2021/01/10 19:28:16	44974825@my	Gary Harry Cun	44974825	XVU6034	Yes	Fill-in		
16	2021/01/10 19:28:18	2021/01/10 19:29:27	44974825@my	Gary Harry Cun	44974825	TA2882	Yes	MCQ	A	
17	2021/01/08 13:07:45	2021/01/08 21:27:05	44974825@my	Gary Harry Cun	44974825	ITC3641	Yes	Document Upd		
18	2021/01/20 21:27:09	2021/01/20 21:40:03	44974825@my	Gary Harry Cun	44974825	DFH4827	Yes	Fill-in		
19	2021/01/20 22:05:29	2021/01/20 22:07:15	56605234@my	KURT RUSTIN	56605234	ICT3715	Yes	Document Upd		
20	2021/01/20 22:07:32	2021/01/20 22:11:40	56605234@my	KURT RUSTIN	56605234	IC3611	Yes	Document Upd		
21	2021/01/20 22:14:30	2021/01/20 22:22:56	56605234@my	KURT RUSTIN	56605234	NHG2156	Yes	Fill-in		
22	2021/01/20 22:22:59	2021/01/20 22:25:05	56605234@my	KURT RUSTIN	56605234	GFJ2615	Yes	Fill-in		
23	2021/01/10 00:33:00	2021/01/10 00:35:56	64235955@mv	KHANGWELO N	64235955	RQJ5431	Yes	MCQ	C	

I made use of the code used for the CleanUp app and made modifications to it to generate the SQL file. Below showcases the classes, variables and properties used for a class to read data from the csv file.

Class used to read data from the access file.

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Process: [30640] CleanUpApp.vshost.exe

CleanUp App CleanUp.vb Clean.vb Form1.vb [Design]

Form1.vb # Clean.vb # Form1.vb [Design]

Public Sub New(ID As String, starttime As String, Comptime As String, Email As String, Name As String, stdnum As String, ModCode As String, Integrity As String, DocType As String, M1 As String, M2 As String)

Me.ID = ID

Me.Starttime = starttime

Me.Comptime = Comptime

Me.Email = Email

Me.Name = Name

Me.Stdnum = stdnum

Me.ModCode = ModCode

Me.Integrity = Integrity

Me.DocType = DocType

Me.M1 = M1

Me.M2 = M2

Me.M3 = M3

Me.M4 = M4

Me.M5 = M5

Me.M6 = M6

Me.M7 = M7

Me.M8 = M8

Me.M9 = M9

Me.M10 = M10

Me.M11 = M11

Me.Q1 = Q1

Me.Q2 = Q2

Me.Q3 = Q3

Me.Q4 = Q4

Me.QAns = QAns

Me.Examinate = Examinate

End Sub

Public Property ID As String

Get

Return m_ID

End Get

Set(value As String)

m_ID = value

End Set

End Property

Public Property Starttime As String

Get

Return m_Starttime

End Get

Set(value As String)

m_Starttime = value

End Set

End Property

Public Property Comptime As String

Get

Return m_Comptime

End Get

Set(value As String)

m_Comptime = value

End Set

End Property

Public Property Email As String

Get

Return m_Email

End Get

Set(value As String)

m_Email = value

End Set

End Property

Public Property Name As String

Get

Return m_Name

End Get

Set(value As String)

m_Name = value

End Set

End Property

Public Property stdnum As String

Get

Return m_Stdnum

End Get

Set(value As String)

m_Stdnum = value

End Set

End Property

Public Property ModCode As String

Get

Return m_ModCode

End Get

Set(value As String)

m_ModCode = value

End Set

End Property

Public Property Integrity As String

Get

Return m_Integrity

End Get

Set(value As String)

m_Integrity = value

End Set

End Property

Public Property DocType As String

Get

Return m_DocType

End Get

Set(value As String)

m_DocType = value

End Set

End Property

Public Property M1 As String

Get

Return m_M1

End Get

Set(value As String)

m_M1 = value

End Set

End Property

Public Property M2 As String

Locals

Name Value

Output

Show output from: Debug

The thread 0x6004 has exited with code 0 (0x0).

The thread 0x6208 has exited with code 0 (0x0).

The thread 0x600C has exited with code 0 (0x0).

The thread 0x600D has exited with code 0 (0x0).

The thread 0x64F4 has exited with code 0 (0x0).

The thread 0x5F54 has exited with code 0 (0x0).

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List

Ready

Database configurations

```

VB CleanUp App (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [30640] CleanUp App.vshost.exe Lifecycle Events Thread: Stack Frame:
Form1.vb CleanDB.vb Clean.vb Form1.vb [Design]
CleanDB.vb
    connection.Close()
End Sub

Public Shared Sub GetStdNum(clean As Clean)
    ' Select Case SUBSTRING(customerName, 1, 5) As ExtractString
    'FROM Exam;
End Sub

Public Shared Function GetResults() As List(Of Clean)
    Dim cleanList As New List(Of Clean)
    Dim result As Clean
    Dim connectionString As String =
        "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=[DataDirectory]\Portal.mdb"
    Dim connection As New OleDbConnection(connectionString)

    Dim selectStatement As String = "SELECT * "
    "FROM Exam"

    Dim selectCommand As New OleDbCommand(selectStatement, connection)

    connection.Open()
    Dim reader As OleDbDataReader = selectCommand.ExecuteReader()
    Do While reader.Read
        result = New Clean
        result.ID = reader("ID").ToString
        result.StartTime = reader("StartTime").ToString
        result.Comptime = reader("Comptime").ToString
        result.Email = reader("Email").ToString
        result.Name = reader("Name").ToString
        result.StdNum = reader("Stdnum").ToString 'this one needs changing
        result.ModCode = reader("ModCode").ToString
        result.Integrity = reader("Integrity").ToString
        result.DocType = reader("DocType").ToString
        result.M1 = reader("M1").ToString
        result.M2 = reader("M2").ToString
        result.M3 = reader("M3").ToString
        result.M4 = reader("M4").ToString
        result.M5 = reader("M5").ToString
        result.M6 = reader("M6").ToString
        result.M7 = reader("M7").ToString
        result.M8 = reader("M8").ToString
        result.M9 = reader("M9").ToString
        result.M10 = reader("M10").ToString
        result.M11 = reader("M11").ToString
        result.M12 = reader("M12").ToString
        result.M13 = reader("M13").ToString
        result.M14 = reader("M14").ToString
        result.M15 = reader("M15").ToString
        result.Upload = reader("Upload").ToString
        result.Q1 = reader("Q1").ToString
        result.Q2 = reader("Q2").ToString
        result.Q3 = reader("Q3").ToString
        result.Q4 = reader("Q4").ToString
        result.QAns = reader("QAns").ToString
        result.ExamDate = reader("ExamDate").ToString
        cleanList.Add(result)
    Loop
    reader.Close()
    connection.Close()
    Return cleanList
End Function

Public Shared Function GetModList() As List(Of ValidMod)

```

This method is used to write the generated SQL file to the text. Save

SQL

```

Public Shared Sub SaveSQL(str As String)
    'this code writes values to the text file
    Dim input As New StreamWriter(New FileStream(Location, FileMode.Create, FileAccess.Write))

    input.WriteLine(str)

    input.Close()
End Sub

```

These are the variables to be used for the process.

```

Imports System.Text.RegularExpressions
Public Class Form1
    Dim cleanList As List(Of Clean)
    Dim backList As List(Of Clean)
    Dim modList As New List(Of ValidMod)
    Dim correctModList As New List(Of Clean) 'try next time with as list only
    Dim examTypeList As New List(Of Clean)
    Dim finalList As New List(Of Clean)
    Public ValidMod As ValidMod
    Public Clean As Clean

    Dim sqlScript As String

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Randomize()

        Dim validMod = New ValidMod("cleanlist(0).ModCode", 0, 0, 0, 0, "")
        cleanList = CleanDB.GetResults 'procedure to fill in cleanlist
        modList = CleanDB.GetModList 'add a value to modlist
        correctModList = CleanDB.GetCorrectMod
        examTypeList = CleanDB.GetExamType
        finalList = GetFinalList() 'Me.finalList wonder if this might work
        FillBoxes()
    End Sub

```

The following method is what I used to filter out certain information to be used for the INSERT queries. What it filters will depend on the arguments presented. For example, filtering out all but student information, module information etc.

CleanUp App - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Form1.vb CleanDB.vb Clean.vb Form1.vb [Design] > Click

```

Private Sub removeDuplicate(remove As String) 'add parameter you wish to remove
    finalList.Clear()
    Dim studentNumber, modCode As String
    Dim exist, chosenOne As Integer

    If remove Is "student" Then 'provides list of students
        'is used by student table
        For i As Integer = 0 To cleanList.Count - 1
            exist = 0
            studentNumber = cleanList(i).StdNum
            If finalList.Count > 0 Then 'checks through list if it already exists, go next if it already does, if it doesn't
                For l As Integer = 0 To finalList.Count - 1
                    If studentNumber = finalList(l).StdNum Then
                        exist = 1
                    End If
                Next
            If exist < 1 Then 'if it doesn't exist, check through examTypeList, make an integer which can be used as an identifier.
                For m As Integer = 0 To cleanList.Count - 1
                    If studentNumber = cleanList(m).StdNum Then
                        chosenOne = m
                    End If
                Next
                finalList.Add(cleanList(chosenOne))
            End If
        Else
            For m As Integer = 0 To cleanList.Count - 1
                If studentNumber = cleanList(m).StdNum Then
                    chosenOne = m
                End If
            Next
            finalList.Add(cleanList(chosenOne))
        End If
    End If

    If remove Is "module" Then 'provides list of modules
        'is used by module, exam table
        For l As Integer = 0 To finalList.Count - 1
            exist = 0
            modCode = cleanList(l).ModCode
            If finalList.Count > 0 Then 'checks through list if it already exists, go next if it already does, if it doesn't
                For i As Integer = 0 To finalList.Count - 1
                    If modCode = finalList(i).ModCode Then
                        exist = 1
                    End If
                Next
            If exist < 1 Then 'if it doesn't exist, check through examTypeList, make an integer which can be used as an identifier.
                For m As Integer = 0 To cleanList.Count - 1
                    If modCode = cleanList(m).ModCode Then
                        chosenOne = m
                    End If
                Next
                finalList.Add(cleanList(chosenOne))
            End If
        Else
            For m As Integer = 0 To cleanList.Count - 1
                If modCode = cleanList(m).ModCode Then
                    chosenOne = m
                End If
            Next
            finalList.Add(cleanList(chosenOne))
        End If
    End If

    If remove Is "enroll" Then 'provides registry list of modules students have registered for.
        'is used by enroll table
        For i As Integer = 0 To cleanList.Count - 1
            exist = 0
            studentNumber = cleanList(i).StdNum
            modCode = cleanList(i).ModCode

            If finalList.Count > 0 Then
                'code to check through the list if it already exists, go next if it already does, if it doesn't
                For l As Integer = 0 To finalList.Count - 1
                    If studentNumber & modCode = finalList(l).StdNum & finalList(l).ModCode Then 'will this even work???
                        exist = 1
                    End If
                Next
            If exist < 1 Then
                'if it doesn't exist, check through examTypeList, make an integer which can be used as an identifier.
                For m As Integer = 0 To cleanList.Count - 1
                    If studentNumber & modCode = cleanList(m).StdNum & cleanList(m).ModCode Then 'will this even work???
                        chosenOne = m
                    End If
                Next
                finalList.Add(cleanList(chosenOne))
            End If
        Else
            For m As Integer = 0 To cleanList.Count - 1
                If studentNumber & modCode = cleanList(m).StdNum & cleanList(m).ModCode Then 'will this even work???
                    chosenOne = m
                End If
            Next
            finalList.Add(cleanList(chosenOne))
        End If
    End If

    'code checks if value already exists in the table
End If

If remove = "MCQ" Or remove = "Fill-in" Or remove = "Document Upload" Then 'removes module duplicates
    For i As Integer = 0 To cleanList.Count - 1
        If (cleanList(i).Octype = remove) Then
            finalList.Add(cleanList(i))
        End If
    Next

    For i As Integer = 0 To finalList.Count - 1
        finalList(i).ID = i + 1
    Next
    'MCQ
    'Fill-in

```

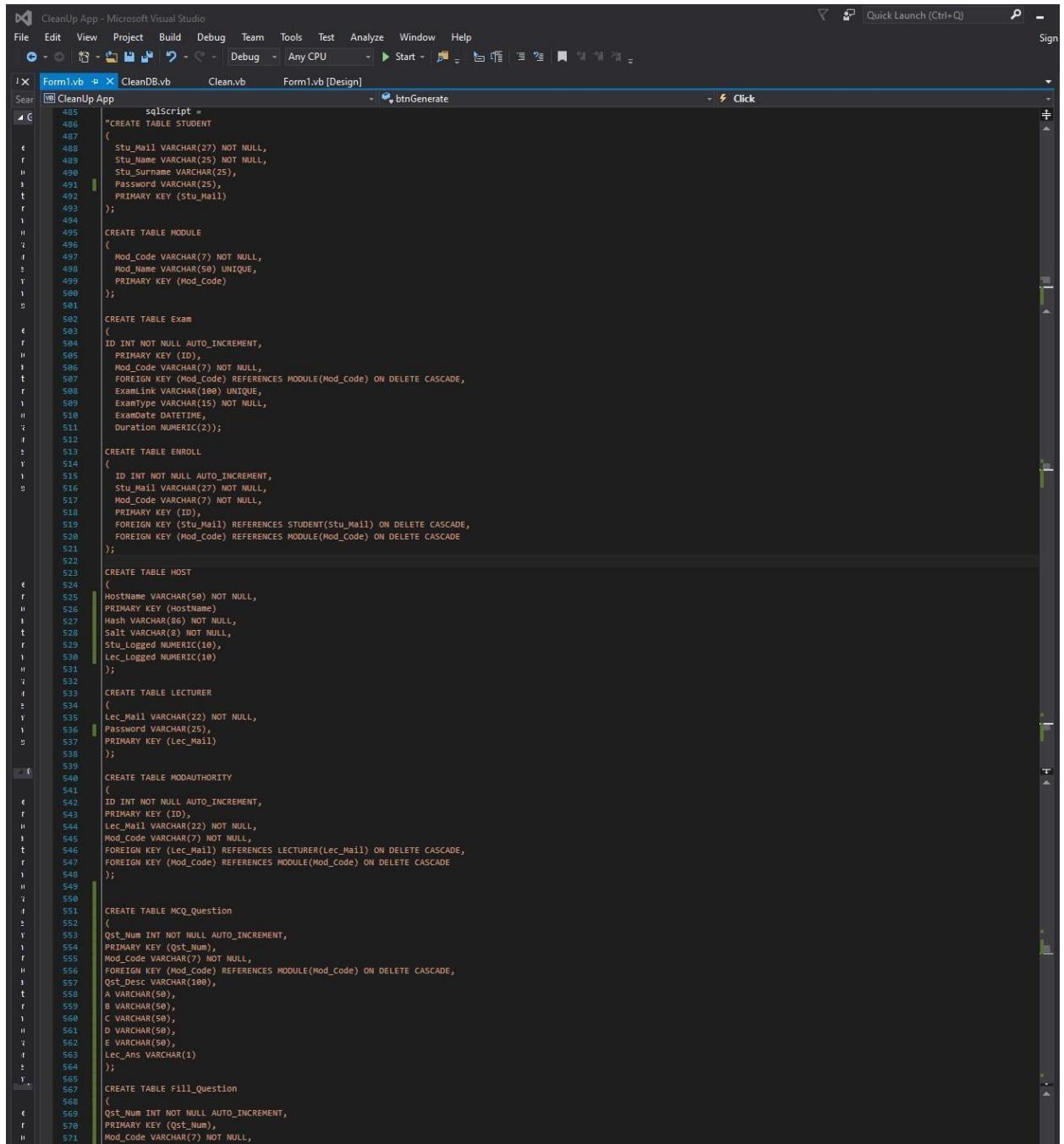
Show output from: Debug
 ('63627922_KAD3867_2021/05/17 22:50:47', '63627922@mylife.unisa.ac.za', 'KAD3867', '2021/05/17 22:45:44', '2021/05/17 22:50:47'),
 ('46160329_ICT3715_2021/05/18 09:39:08', '46160329@mylife.unisa.ac.za', 'ICT3715', '2021/05/18 09:30:43', '2021/05/18 09:39:08').
 The thread 0x7630 has exited with code 0 (0x0).
 The program '[20616] CleanUp App.vshost.exe' has exited with code -1 (0xffffffff).

Output

Ready Ln 531 Col 2 Ch 2 INS

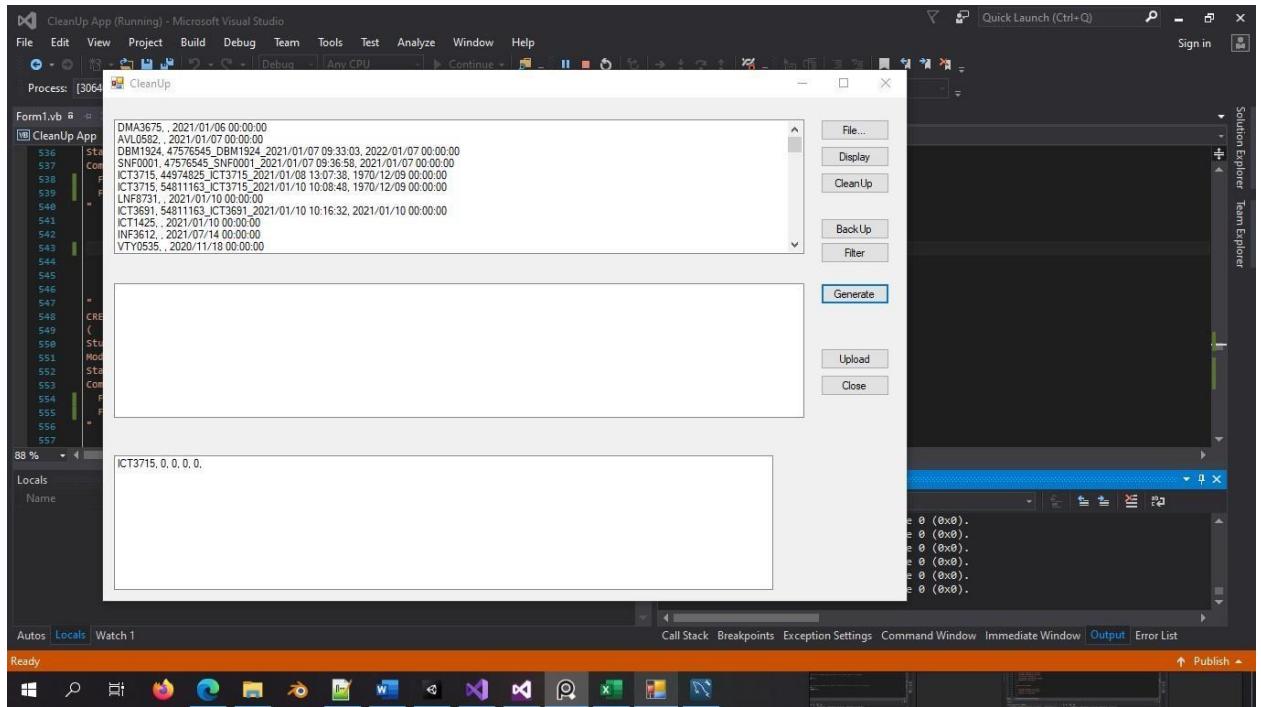
16:55 2021/06/02

The string that will be generated when app is executed:

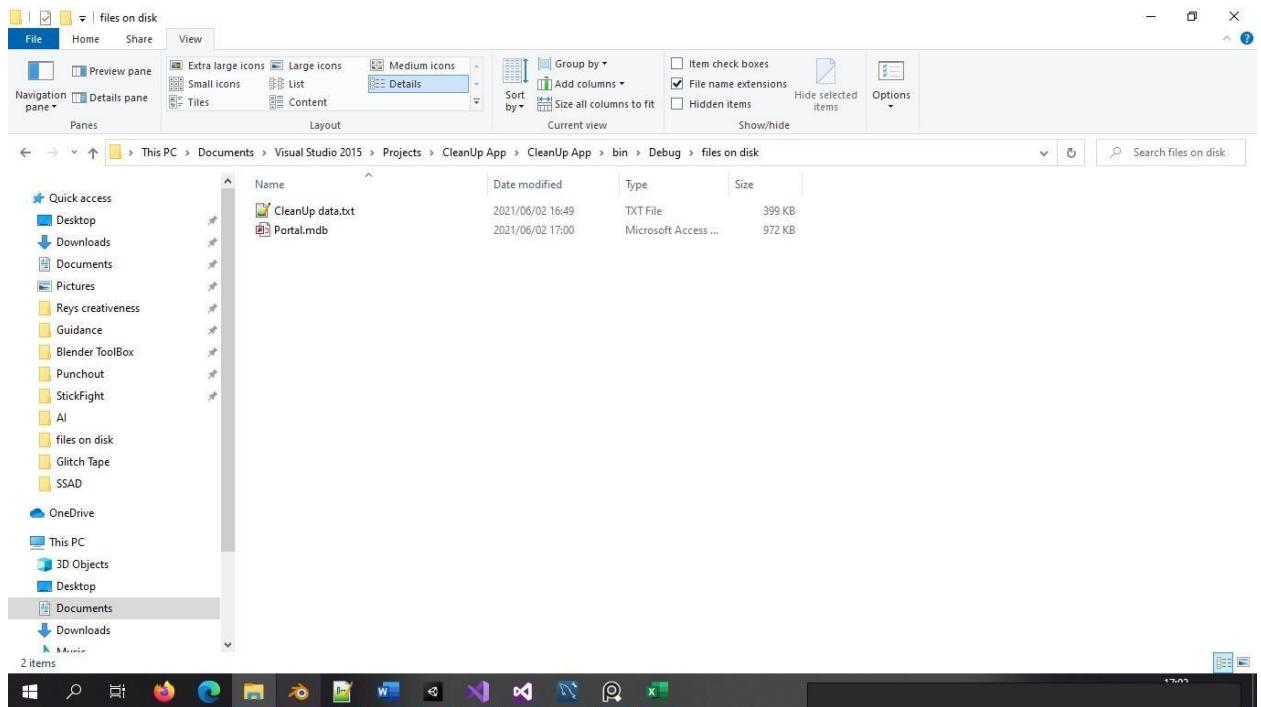


```
sqlScript =  
485 "CREATE TABLE STUDENT  
486 (  
487     Stu_Mail VARCHAR(27) NOT NULL,  
488     Stu_Name VARCHAR(25) NOT NULL,  
489     Stu_Surname VARCHAR(35),  
490     Password VARCHAR(25),  
491     PRIMARY KEY (Stu_Mail)  
492 );  
493  
494 CREATE TABLE MODULE  
495 (  
496     Mod_Code VARCHAR(7) NOT NULL,  
497     Mod_Name VARCHAR(50) UNIQUE,  
498     PRIMARY KEY (Mod_Code)  
499 );  
500  
501 CREATE TABLE EXAM  
502 (  
503     ID INT NOT NULL AUTO_INCREMENT,  
504     PRIMARY KEY (ID),  
505     Mod_Code VARCHAR(7) NOT NULL,  
506     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,  
507     ExamLink VARCHAR(100) UNIQUE,  
508     ExamType VARCHAR(15) NOT NULL,  
509     ExamDate DATETIME,  
510     Duration NUMERIC(2));  
511  
512 CREATE TABLE ENROLL  
513 (  
514     ID INT NOT NULL AUTO_INCREMENT,  
515     Stu_Mail VARCHAR(27) NOT NULL,  
516     Mod_Code VARCHAR(7) NOT NULL,  
517     PRIMARY KEY (ID),  
518     FOREIGN KEY (Stu_Mail) REFERENCES STUDENT(Stu_Mail) ON DELETE CASCADE,  
519     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE  
520 );  
521  
522 CREATE TABLE HOST  
523 (  
524     HostName VARCHAR(50) NOT NULL,  
525     PRIMARY KEY (HostName),  
526     Hash VARCHAR(86) NOT NULL,  
527     Salt VARCHAR(8) NOT NULL,  
528     Stu_Logged NUMERIC(10),  
529     Lec_Logged NUMERIC(10)  
530 );  
531  
532  
533 CREATE TABLE LECTURER  
534 (  
535     Lec_Mail VARCHAR(22) NOT NULL,  
536     Password VARCHAR(25),  
537     PRIMARY KEY (Lec_Mail)  
538 );  
539  
540 CREATE TABLE MODAUTHORITY  
541 (  
542     ID INT NOT NULL AUTO_INCREMENT,  
543     PRIMARY KEY (ID),  
544     Lec_Mail VARCHAR(22) NOT NULL,  
545     Mod_Code VARCHAR(7) NOT NULL,  
546     FOREIGN KEY (Lec_Mail) REFERENCES LECTURER(Lec_Mail) ON DELETE CASCADE,  
547     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE  
548 );  
549  
550  
551 CREATE TABLE MCQ_Question  
552 (  
553     Qst_Num INT NOT NULL AUTO_INCREMENT,  
554     PRIMARY KEY (Qst_Num),  
555     Mod_Code VARCHAR(7) NOT NULL,  
556     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,  
557     Qst_Desc VARCHAR(100),  
558     A VARCHAR(50),  
559     B VARCHAR(50),  
560     C VARCHAR(50),  
561     D VARCHAR(50),  
562     E VARCHAR(50),  
563     Lec_Ans VARCHAR(1)  
564 );  
565  
566 CREATE TABLE Fill_Question  
567 (  
568     Qst_Num INT NOT NULL AUTO_INCREMENT,  
569     PRIMARY KEY (Qst_Num),  
570     Mod_Code VARCHAR(7) NOT NULL,  
571
```


The SQL generator app in action



File that gets generated



```
1 CREATE TABLE STUDENT
2 (
3     Stu_Mail VARCHAR(27) NOT NULL,
4     Stu_Name VARCHAR(25) NOT NULL,
5     Stu_Surname VARCHAR(25),
6     Password VARCHAR(25),
7     PRIMARY KEY (Stu_Mail)
8 );
9
10 CREATE TABLE MODULE
11 (
12     Mod_Code VARCHAR(7) NOT NULL,
13     Mod_Name VARCHAR(50) UNIQUE,
14     PRIMARY KEY (Mod_Code)
15 );
16
17 CREATE TABLE Exam
18 (
19     ID INT NOT NULL AUTO_INCREMENT,
20     PRIMARY KEY (ID),
21     Mod_Code VARCHAR(7) NOT NULL,
22     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
23     ExamLink VARCHAR(100) UNIQUE,
24     ExamType VARCHAR(15) NOT NULL,
25     ExamDate DATETIME,
26     Duration NUMERIC(2));
27
28 CREATE TABLE ENROLL
29 (
30     ID INT NOT NULL AUTO_INCREMENT,
31     Stu_Mail VARCHAR(27) NOT NULL,
32     Mod_Code VARCHAR(7) NOT NULL,
33     PRIMARY KEY (ID).
```



```
28 CREATE TABLE ENROLL
29 (
30     ID INT NOT NULL AUTO_INCREMENT,
31     Stu_Mail VARCHAR(27) NOT NULL,
32     Mod_Code VARCHAR(7) NOT NULL,
33     PRIMARY KEY (ID),
34     FOREIGN KEY (Stu_Mail) REFERENCES STUDENT(Stu_Mail) ON DELETE CASCADE,
35     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE
36 );
37
38 CREATE TABLE HOST
39 (
40     HostName VARCHAR(50) NOT NULL,
41     PRIMARY KEY (HostName)
42     Hash VARCHAR(86) NOT NULL,
43     Salt VARCHAR(8) NOT NULL,
44     Stu_Logged NUMERIC(10),
45     Lec_Logged NUMERIC(10)
46 );
47
48 CREATE TABLE LECTURER
49 (
50     Lec_Mail VARCHAR(22) NOT NULL,
51     Password VARCHAR(25),
52     PRIMARY KEY (Lec_Mail)
53 );
54
55 CREATE TABLE MODAUTHORITY
56 (
57     ID INT NOT NULL AUTO_INCREMENT,
58     PRIMARY KEY (ID),
59     Lec_Mail VARCHAR(22) NOT NULL,
60     Mod_Code VARCHAR(7) NOT NULL..
```



```
66 CREATE TABLE MCQ_Question
67 (
68     Qst_Num INT NOT NULL AUTO_INCREMENT,
69     PRIMARY KEY (Qst_Num),
70     Mod_Code VARCHAR(7) NOT NULL,
71     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
72     Qst_Desc VARCHAR(100),
73     A VARCHAR(50),
74     B VARCHAR(50),
75     C VARCHAR(50),
76     D VARCHAR(50),
77     E VARCHAR(50),
78     Lec_Ans VARCHAR(1)
79 );
80
81
82 CREATE TABLE Fill_Question
83 (
84     Qst_Num INT NOT NULL AUTO_INCREMENT,
85     PRIMARY KEY (Qst_Num),
86     Mod_Code VARCHAR(7) NOT NULL,
87     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
88     Qst_Desc VARCHAR(100),
89     Qst_Type VARCHAR(6),
90     Lec_Ans VARCHAR(1)
91 );
92
93
94 CREATE TABLE Upload_Ans
95 (
96     ID INT NOT NULL AUTO_INCREMENT,
97     PRIMARY KEY (ID),
98     File_name VARCHAR(100) NOT NULL,
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
```

C:\Users\Rey\Documents\Visual Studio 2015\Projects\CleanUp App\CleanUp App\bin\Debug\files on disk\CleanUp data.txt - Notepad++

```

79
80 CREATE TABLE Fill_Ans
81 (
82 ID INT NOT NULL AUTO_INCREMENT,
83 PRIMARY KEY (ID),
84 Stu_Mail VARCHAR(27) NOT NULL,
85 Mod_Code VARCHAR(7) NOT NULL,
86 StartTime DateTime,
87 CompTime DateTime,
88 FOREIGN KEY (Stu_Mail) REFERENCES STUDENT(Stu_Mail) ON DELETE CASCADE,
89 FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
90 Q1 VARCHAR(200),Q2 VARCHAR(200),Q3 VARCHAR(200),Q4 VARCHAR(200),Q5 VARCHAR(200),Q6 VARCHAR(200),Q7 VARCHAR(200),Q8 VARCHAR(200),Q9 VARCHAR(200)
91
92
93 CREATE TABLE MCQ_Ans
94 (
95 ID INT NOT NULL AUTO_INCREMENT,
96 PRIMARY KEY (ID),
97 Stu_Mail VARCHAR(27) NOT NULL,
98 Mod_Code VARCHAR(7) NOT NULL,
99 StartTime DateTime,
100 CompTime DateTime,
101 FOREIGN KEY (Stu_Mail) REFERENCES STUDENT(Stu_Mail) ON DELETE CASCADE,
102 FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
103 M1 VARCHAR(1),M2 VARCHAR(1),M3 VARCHAR(1),M4 VARCHAR(1),M5 VARCHAR(1),M6 VARCHAR(1),M7 VARCHAR(1),M8 VARCHAR(1),M9 VARCHAR(1),M10 VARCHAR(1),M11 VARCHAR(1)
104
105 INSERT INTO STUDENT(Stu_Mail, Stu_Name)
106 VALUES('47576545@mylife.unisa.ac.za', 'DUSTIN BOOYSEN MARCUS'),
107 ('44974825@mylife.unisa.ac.za', 'Gary Harry Cummings'),
108 ('54811163@mylife.unisa.ac.za', 'Hendrik Abraham Kruger'),
109 ('34576851@mylife.unisa.ac.za', 'Craig Brickhill'),
110 ('56605234@mylife.unisa.ac.za', 'KURT RUSTIN'),
111 ('64239555@mylife.unisa.ac.za', 'KHANGWELA NTHONIUS MAPHAHA').
<
```

Normal text file length: 411 681 lines: 5 073 Ln: 51 Col: 20 Pos: 1:151 Windows (CRLF) UTF-8 INS

C:\Users\Rey\Documents\Visual Studio 2015\Projects\CleanUp App\CleanUp App\bin\Debug\files on disk\CleanUp data.txt - Notepad++

```

350 INSERT INTO MODULE(Mod_Code)
351 VALUES('AVL0582'),
352 ('DEMI924'),
353 ('SNF0001'),
354 ('ICT3715'),
355 ('INF8731'),
356 ('ICT3691'),
357 ('ICT1425'),
358 ('INF3612'),
359 ('VYY0535'),
360 ('KUK2442'),
361 ('FAG7301'),
362 ('LHU8077'),
363 ('XVU6034'),
364 ('GTA2882'),
365 ('ICT3641'),
366 ('OPH4872'),
367 ('ICT3611'),
368 ('NHG2156'),
369 ('GEJ2615'),
370 ('RQJ5431'),
371 ('QWJ8855'),
372 ('ICT3722'),
373 ('ARN7183'),
374 ('DWD2184'),
375 ('CIS1082'),
376 ('EUG3054'),
377 ('INF3708'),
378 ('FMK0006'),
379 ('JLO7012'),
380 ('ACR5568'),
381 ('ICT2151'),
382 ('ICT2501').
```

Normal text file length: 407 998 lines: 5 023 Ln: 1 Col: 1 Pos: 1 Windows (CRLF) UTF-8 INS

```

1358 INSERT INTO Exam(Mod_Code, ExamType, ExamDate)
1359 VALUES('AVL0582', 'Fill-in', '2021/01/07 00:00:00'),
1360 ('DBM1924', 'Document Upload', '2022/01/07 00:00:00'),
1361 ('SNF0001', 'Document Upload', '2021/01/07 00:00:00'),
1362 ('ICT3715', 'Document Upload', '1970/12/09 00:00:00'),
1363 ('LNF8731', 'Fill-in', '2021/01/10 00:00:00'),
1364 ('ICT3691', 'Document Upload', '2021/01/10 00:00:00'),
1365 ('ICT1425', 'MCQ', '2021/01/10 00:00:00'),
1366 ('INF3612', 'Fill-in', '2021/07/14 00:00:00'),
1367 ('VYU0535', 'MCQ', '2020/11/18 00:00:00'),
1368 ('KUK2442', 'Document Upload', '2021/09/10 00:00:00'),
1369 ('FAG7301', 'Fill-in', '2021/08/20 00:00:00'),
1370 ('LHU8077', 'MCQ', '2010/09/07 00:00:00'),
1371 ('XVU6034', 'Fill-in', '2021/01/11 00:00:00'),
1372 ('GTA2882', 'MCQ', '2021/01/12 00:00:00'),
1373 ('ICT3641', 'Document Upload', '2021/01/12 00:00:00'),
1374 ('OFH4872', 'Fill-in', '2021/01/14 00:00:00'),
1375 ('ICT3611', 'Document Upload', '2021/01/27 00:00:00'),
1376 ('NHG2156', 'Fill-in', '2021/03/17 00:00:00'),
1377 ('GFJ2615', 'Fill-in', '2021/06/15 00:00:00'),
1378 ('RQJ5431', 'MCQ', '2021/06/08 00:00:00'),
1379 ('QWJ8855', 'MCQ', '2021/06/14 00:00:00'),
1380 ('ICT3722', 'Fill-in', '2021/01/31 00:00:00'),
1381 ('ARN7183', 'Document Upload', '2021/06/22 00:00:00'),
1382 ('DYD2184', 'MCQ', '2021/01/14 00:00:00'),
1383 ('CIS1082', 'Fill-in', '2021/01/08 00:00:00'),
1384 ('EUG3054', 'MCQ', '2021/01/11 00:00:00'),
1385 ('INF3708', 'Document Upload', '2021/01/11 00:00:00'),
1386 ('FMK0006', 'MCQ', '2021/01/11 00:00:00'),
1387 ('JL07012', 'Fill-in', '2021/01/11 00:00:00'),
1388 ('ACR5568', 'Fill-in', '2021/01/12 00:00:00'),
1389 ('ICT2151', 'Document Upload', '2021/01/12 00:00:00').
1390

```

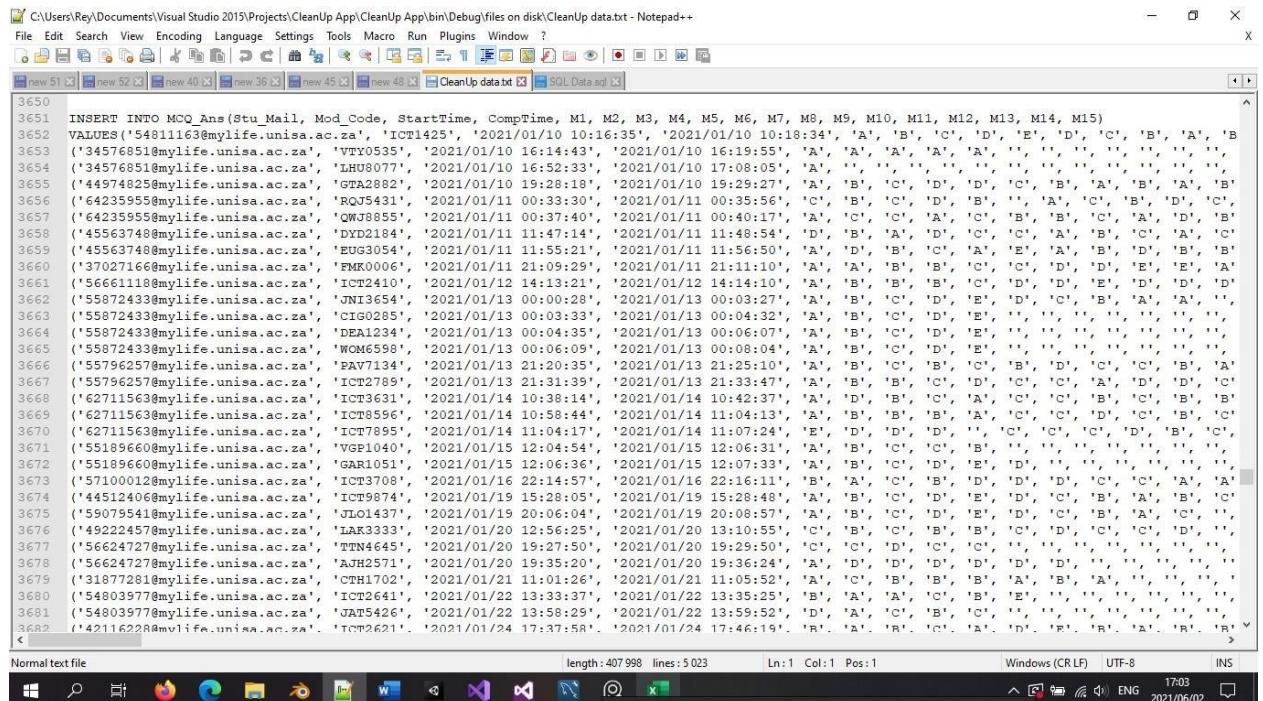
Normal text file length: 407 998 lines: 5 023 Ln:1 Col:1 Pos:1 Windows (CRLF) UTF-8 INS

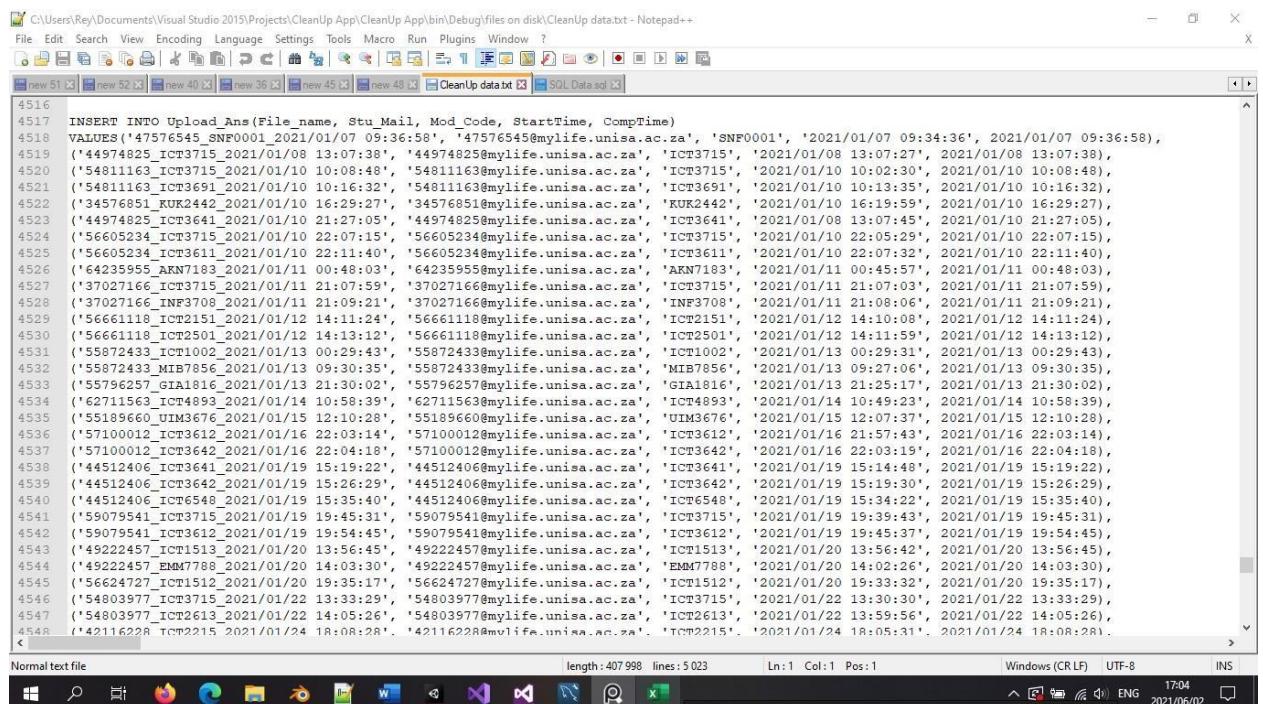
```

2367
2368
2369 INSERT INTO ENROLL(Stu_Mail, Mod_Code)
2370 VALUES('47576545@mylife.unisa.ac.za', 'AVL0582'),
2371 ('47576545@mylife.unisa.ac.za', 'DBM1924'),
2372 ('47576545@mylife.unisa.ac.za', 'SNF0001'),
2373 ('44974825@mylife.unisa.ac.za', 'ICT3715'),
2374 ('54811163@mylife.unisa.ac.za', 'ICT3715'),
2375 ('54811163@mylife.unisa.ac.za', 'LNF8731'),
2376 ('54811163@mylife.unisa.ac.za', 'ICT3691'),
2377 ('54811163@mylife.unisa.ac.za', 'ICT1425'),
2378 ('34576851@mylife.unisa.ac.za', 'INF3612'),
2379 ('34576851@mylife.unisa.ac.za', 'VYU0535'),
2380 ('34576851@mylife.unisa.ac.za', 'KUK2442'),
2381 ('34576851@mylife.unisa.ac.za', 'FAG7301'),
2382 ('34576851@mylife.unisa.ac.za', 'LHU8077'),
2383 ('44974825@mylife.unisa.ac.za', 'XVU6034'),
2384 ('44974825@mylife.unisa.ac.za', 'GTA2882'),
2385 ('44974825@mylife.unisa.ac.za', 'ICT3641'),
2386 ('44974825@mylife.unisa.ac.za', 'OFH4872'),
2387 ('56605234@mylife.unisa.ac.za', 'ICT3715'),
2388 ('56605234@mylife.unisa.ac.za', 'ICT3611'),
2389 ('56605234@mylife.unisa.ac.za', 'NHG2156'),
2390 ('56605234@mylife.unisa.ac.za', 'GFJ2615'),
2391 ('64235955@mylife.unisa.ac.za', 'RQJ5431'),
2392 ('64235955@mylife.unisa.ac.za', 'QWJ8855'),
2393 ('64235955@mylife.unisa.ac.za', 'ICT3722'),
2394 ('64235955@mylife.unisa.ac.za', 'ARN7183'),
2395 ('45563748@mylife.unisa.ac.za', 'DYD2184'),
2396 ('45563748@mylife.unisa.ac.za', 'CIS1082'),
2397 ('45563748@mylife.unisa.ac.za', 'EUG3054'),
2398 ('37027166@mylife.unisa.ac.za', 'ICT3715'),
2399 ('37027166@mylife.unisa.ac.za', 'TNF3708').

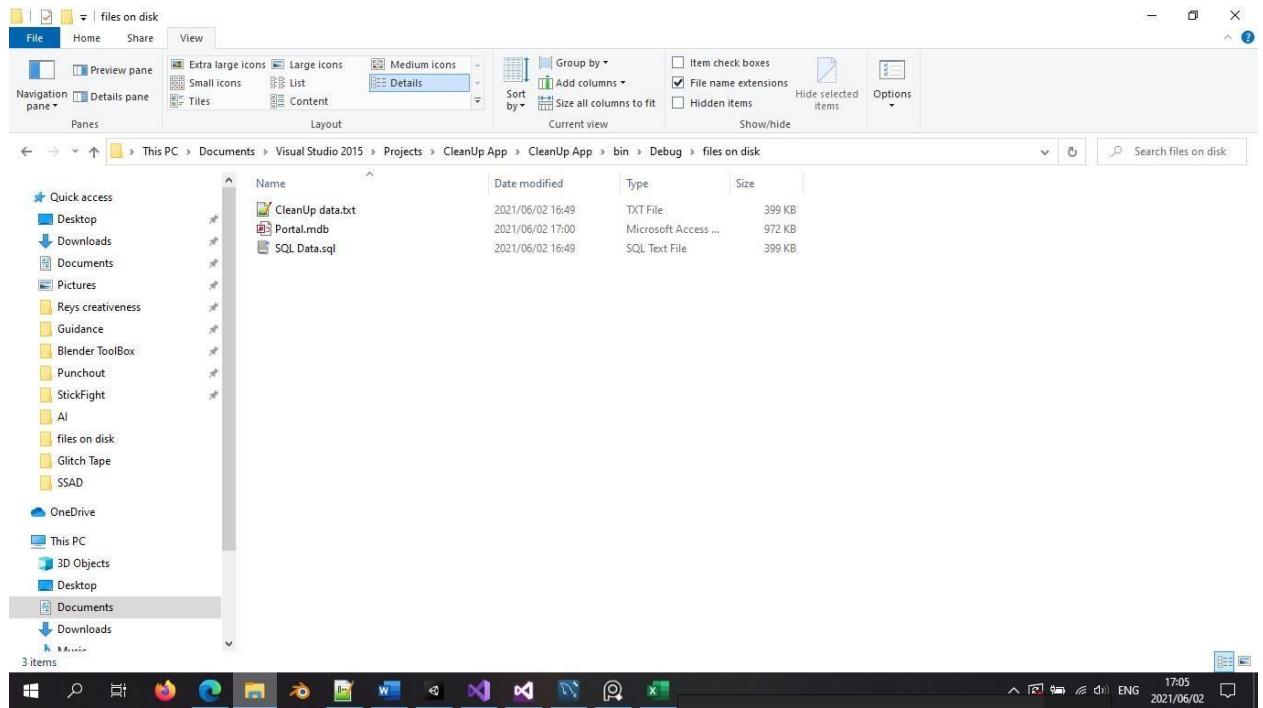
```

Normal text file length: 407 998 lines: 5 023 Ln:1 Col:1 Pos:1 Windows (CRLF) UTF-8 INS





I then renamed the file to SQL Data.sql



The script is now ready to be executed.

```

1 CREATE TABLE STUDENT
2 (
3     Stu_Mail VARCHAR(27) NOT NULL,
4     Stu_Name VARCHAR(25) NOT NULL,
5     Stu_Surname VARCHAR(25),
6     Password VARCHAR(25),
7     PRIMARY KEY (Stu_Mail)
8 );
9
10 CREATE TABLE MODULE
11 (
12     Mod_Code VARCHAR(7) NOT NULL,
13     Mod_Name VARCHAR(50) UNIQUE,
14     PRIMARY KEY (Mod_Code)
15 );
16
17 CREATE TABLE Exam
18 (
19     ID INT NOT NULL AUTO_INCREMENT,
20     PRIMARY KEY (ID),
21     Mod_Code VARCHAR(7) NOT NULL,
22     FOREIGN KEY (Mod_Code) REFERENCES MODULE(Mod_Code) ON DELETE CASCADE,
23     ExamLink VARCHAR(100) UNIQUE,
24     ExamType VARCHAR(15) NOT NULL,
25     ExamDate DATETIME,
26     Duration NUMERIC(3));
27
28 CREATE TABLE ENROLL
29 (
30     ID INT NOT NULL AUTO_INCREMENT,
31     Stu_Mail VARCHAR(27) NOT NULL,
32     Mod_Code VARCHAR(7) NOT NULL,
33     PRIMARY KEY (ID),
34     FOREIGN KEY (Stu_Mail) REFERENCES STUDENT(Stu_Mail) ON DELETE CASCADE

```

This screenshot shows a Notepad++ window displaying a SQL script. The script contains four CREATE TABLE statements: STUDENT, MODULE, Exam, and ENROLL. The STUDENT table has columns for Stu_Mail (VARCHAR(27), primary key), Stu_Name (VARCHAR(25)), Stu_Surname (VARCHAR(25)), and Password (VARCHAR(25)). The MODULE table has columns for Mod_Code (VARCHAR(7), primary key) and Mod_Name (VARCHAR(50), unique). The Exam table has columns for ID (auto-increment, primary key), Mod_Code (VARCHAR(7)), ExamLink (VARCHAR(100), unique), ExamType (VARCHAR(15)), ExamDate (DATETIME), and Duration (NUMERIC(3)). The ENROLL table has columns for ID (auto-increment, primary key), Stu_Mail (VARCHAR(27), foreign key referencing STUDENT(Stu_Mail)), and Mod_Code (VARCHAR(7), foreign key referencing MODULE(Mod_Code)). The script uses MySQL syntax, including AUTO_INCREMENT and ON DELETE CASCADE.

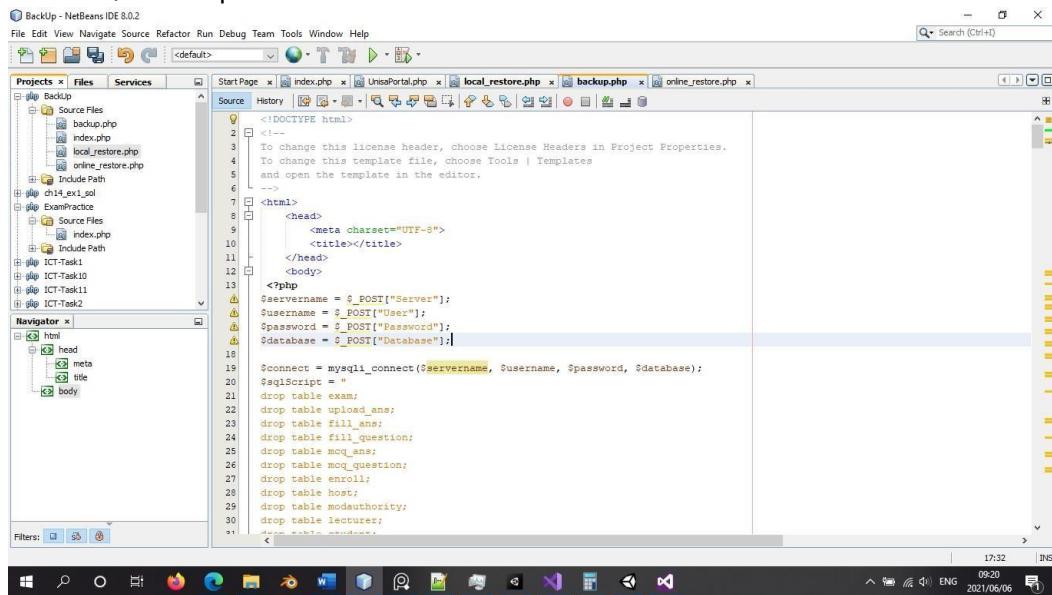
2. Include Screen captures as evidence of data imported in the database. (Attach as PDFs) (30)

SECTION B

For the backup procedures, I have created an app called UNISA Database Backup. Its main responsibility is to periodically make backups of the portal's database.

Its interface was designed with the Unity app being the frontend whilst the backend is entirely operated by php and its connection to the database.

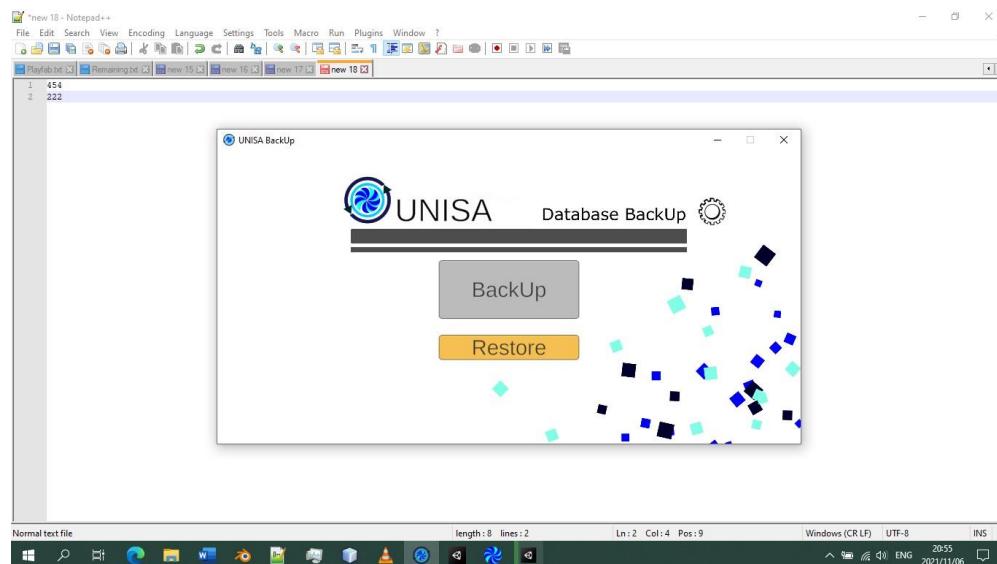
A string is generated containing executable SQL scripts that will then be appended to an existing or created SQL Data.sql file on the local drive or online drive.



The screenshot shows the NetBeans IDE interface with the following details:

- File Menu:** File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
- Search Bar:** Search (Ctrl+I)
- Project Explorer:** Projects > Backup > Backup.php, index.php, UnisPortal.php, local_restore.php, backup.php, online_restore.php
- Navigator:** Navigator > Html > Head > meta, title, body
- Code Editor:** Content pane showing PHP code for generating SQL scripts. The code includes variables for servername, username, password, and database, and loops through tables to generate drop statements.
- Bottom Status Bar:** 17:32, INs, Windows (CR LF), UTF-8, 20-55, 2021/06/06

Within the app, it consists of 3 buttons, BackUp, Restore and Settings button at the top right corner.



Settings

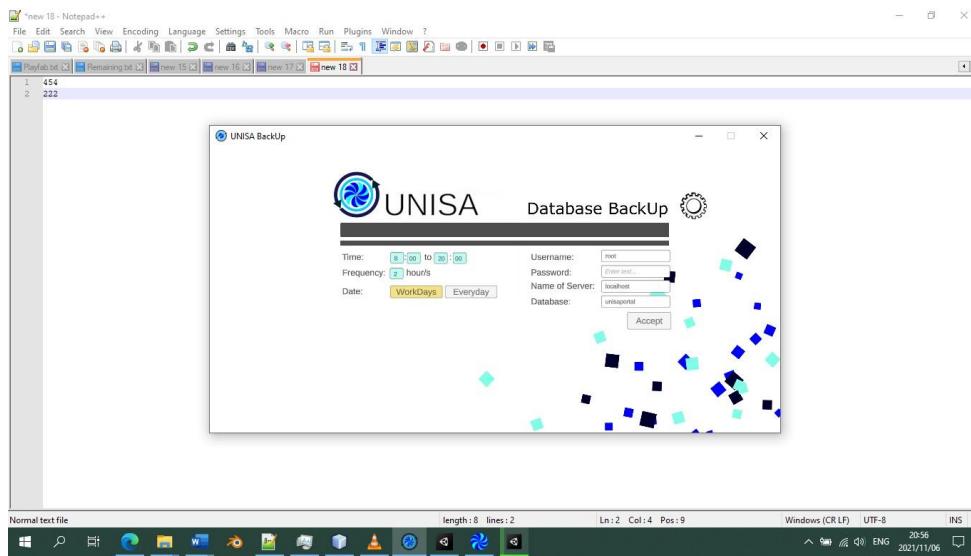
When this button is clicked, it opens the settings menu. Here, configurations can be made:

Time = Start and end time of backups e.g., 06:00 - 21:00

Frequency = Specifying the time and date of these backups e.g., every 2 hours

Date = Backup operating hours.

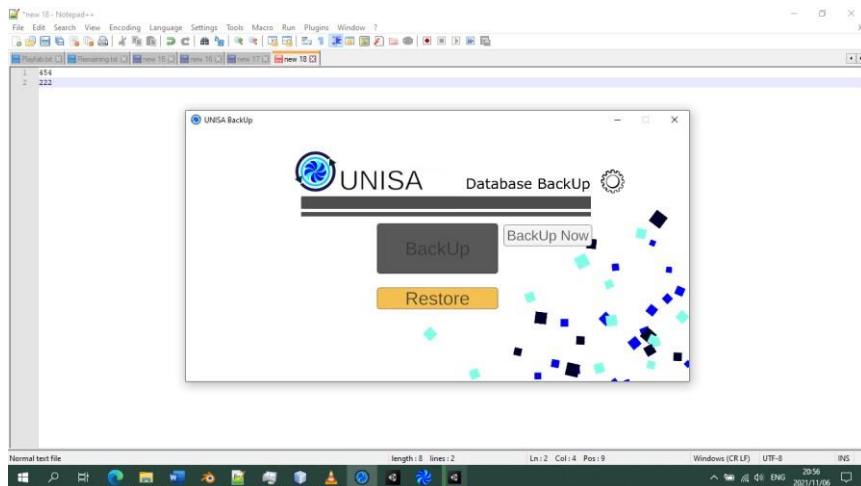
The user can also specify their database connection by entering their username, password, server and database name.



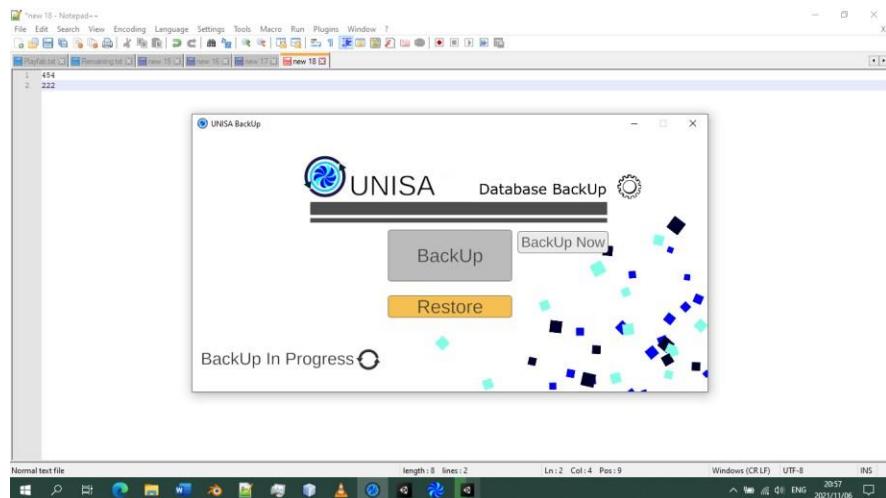
Backup

In the main menu, when the backup button is pressed, the backup procedure will be executed at the next specified time according to the frequency set.

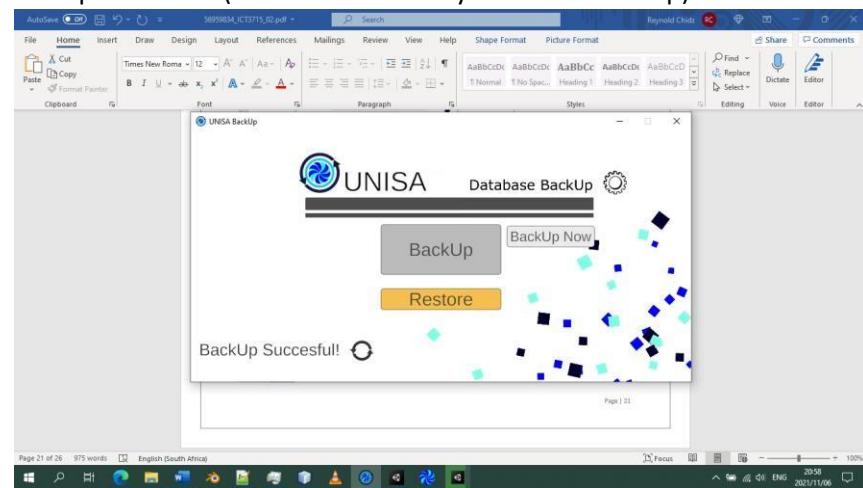
A new button will appear called 'BackUp Now' which will cause the backup procedure to be prematurely executed.



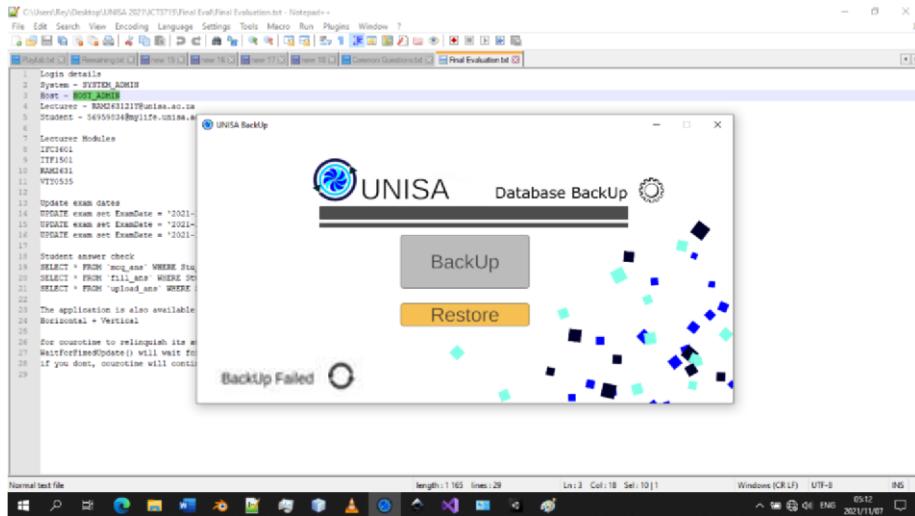
When the backup Now button is pressed, it will showcase messaging regarding the procedure: Backup in progress



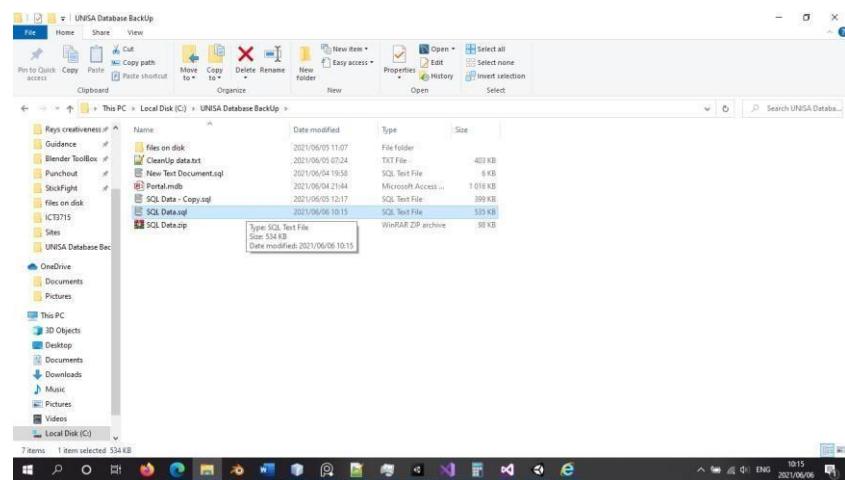
Backup Successful(when it successfully made the backup)



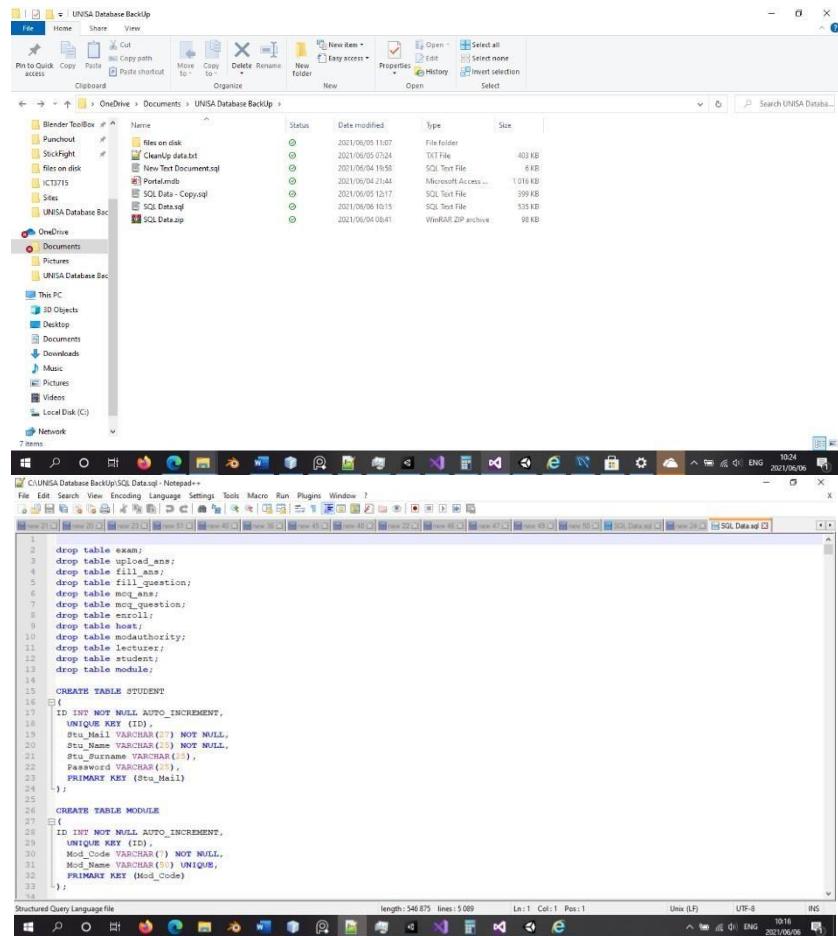
BackUp Failed(when user is offline or something is wrong with the connection to online drive)



When a backup is made, it is stored in specified storage locations online and offline. Local Backup

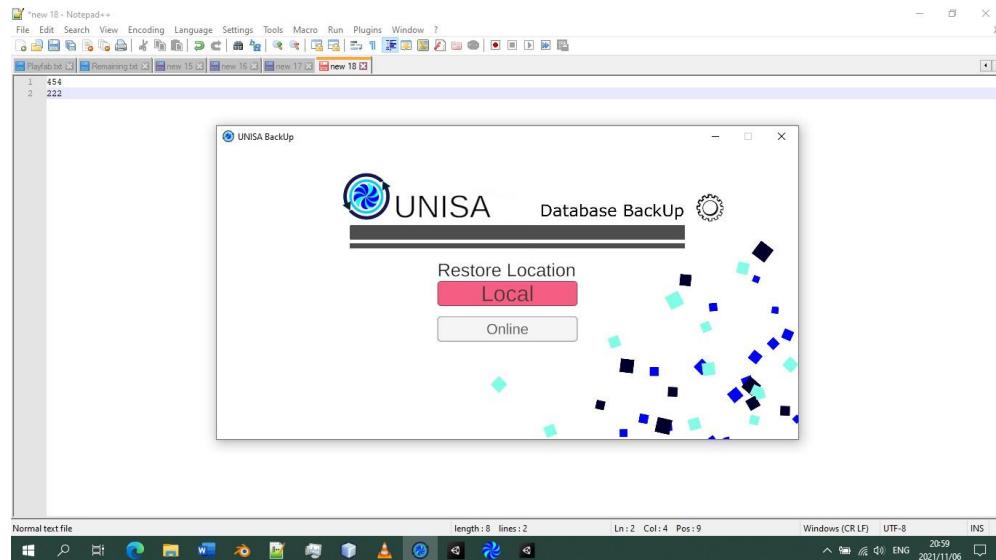


Online Backup



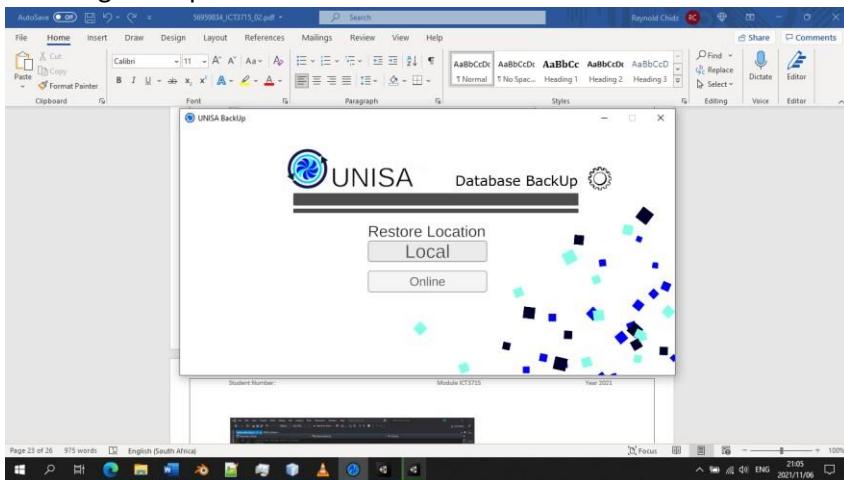
RESTORE

When the user clicks the restore button they are redirected to this page:

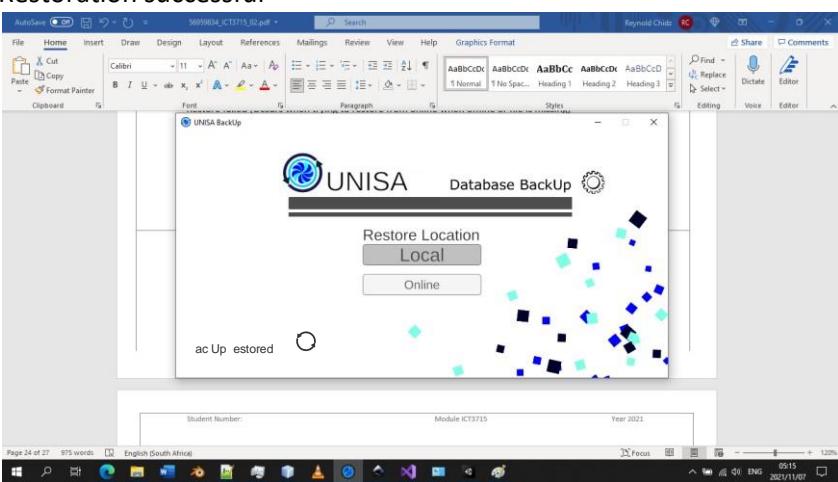


Options are given to restore from a local or online backup. When either of the buttons are clicked, these messages are displayed:

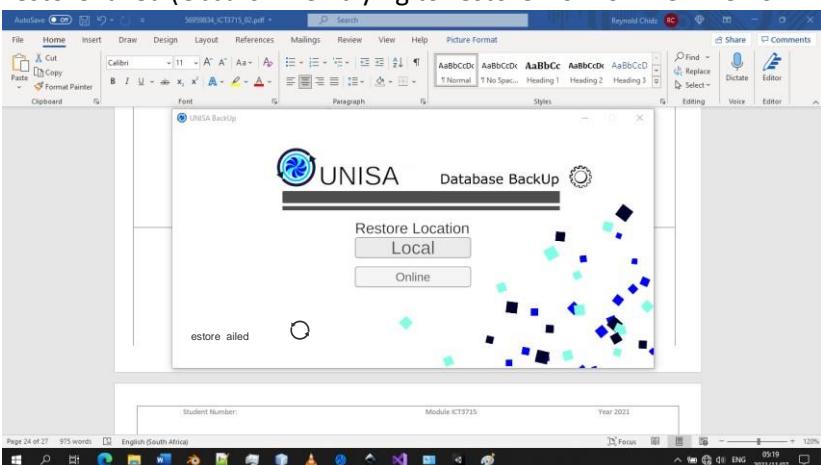
Restoring backup



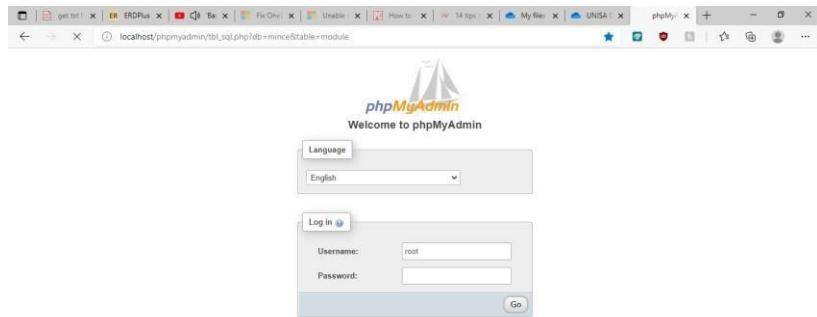
Restoration successful



Restore failed (Occurs when trying to restore from online when offline or file is missing)



Back and Restore Testing



The SQL data.sql file generated will automatically run during the restoration process. For these tests, the SQL script will be manually executed to showcase everything is in working order:

Data deliberately manipulated; records removed.

Table	Action	Rows	Type	Collation	Size	Overhead
boom	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 kB	-
enroll	Browse Structure Search Insert Empty Drop	1,283	InnoDB	utf8mb4_general_ci	272.0 kB	-
exam	Browse Structure Search Insert Empty Drop	1,000	InnoDB	utf8mb4_general_ci	144.0 kB	-
fill_ans	Browse Structure Search Insert Empty Drop	303	InnoDB	utf8mb4_general_ci	280.0 kB	-
fill_question	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 kB	-
host	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 kB	-
lecturer	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 kB	-
mcq_ans	Browse Structure Search Insert Empty Drop	413	InnoDB	utf8mb4_general_ci	144.0 kB	-
mcq_question	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 kB	-
modauthority	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 kB	-
module	Browse Structure Search Insert Empty Drop	1,000	InnoDB	utf8mb4_general_ci	128.0 kB	-
student	Browse Structure Search Insert Empty Drop	285	InnoDB	utf8mb4_general_ci	64.0 kB	-
upload_ans	Browse Structure Search Insert Empty Drop	505	InnoDB	utf8mb4_general_ci	160.0 kB	-
13 tables	Sum	4,864	InnoDB	utf8mb4_general_ci	1.3 MB	0.8

```

1 row affected. (Query took 0.3320 seconds.)
delete from enroll where id=1

1 row affected. (Query took 0.2832 seconds.)
delete from enroll where id=2

1 row affected. (Query took 0.1280 seconds.)
delete from enroll where id=3

1 row affected. (Query took 0.1190 seconds.)
delete from enroll where id=4

```

The SQL script generated by the backup app will now be manually imported.

The image contains three vertically stacked screenshots of the phpMyAdmin interface.

- Screenshot 1: Importing into the database "unisaportal".** This screen shows the import configuration for a SQL file named "Data.sql". It includes options for file compression, character set (utf8), and partial imports. A message at the bottom indicates the import was successful, executing 31 queries.
- Screenshot 2: Database structure for "unisaportal".** This screen displays the table structure for the "unisaportal" database. It lists 13 tables: boom, enroll, exam, fill_ans, fill_question, host, lecturer, mcq_ans, mcq_question, modauthority, module, student, upload_ans, and weather_records. Each table has its name, rows, type, size, and overhead details.
- Screenshot 3: Database structure for "unisaportal" (continued).** This is a continuation of the table structure view, showing the remaining tables: New, boom, enroll, exam, fill_ans, fill_question, host, lecturer, mcq_ans, mcq_question, modauthority, module, student, upload_ans, and weather_records.

TOTAL: 60

Submitting the Assignment.

Complete this template. Save the document as PDF, e.g. 12345678_ICT3715_01.pdf, (replace 1234568 with your student number).

Submit this PDF together with the Screen Captures on or before the due date.