**Steps Need to be done**

1. **Web Application**
2. To copy the client

* Go to the folder C:\inetpub\wwwroot and copy all the files and put it in the server’s similar path.

1. To copy the web api

* Go to inetmgr (IIS), and expand the default browser. There you will find the api. Right click on it and click on explorer. Copy the files inside that folder and paste it in a similar folder inside the server.
* Right click on Default directory and create an application, point this directory to that
* Edit the web.config file and change the sql server details

<connectionStrings>

<add name="qmEntities" connectionString="metadata=res://\*/QMDataModel.csdl|res://\*/QMDataModel.ssdl|res://\*/QMDataModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=**SystemServerName**;initial catalog=qm;persist security info=True;user id=**userID**;password=**password**;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

* Create a folder in d drive and set this in web.config file

<add key="FTPLocation" value="D:\ftp" />

1. **Windows** **Service**
2. Go to the folder UtilityService in D crive
3. Make a similar folder in the target machine and paste it there
4. Configure the areas in app.config file

<add key="FTPLocation" value="D:\ftp\" />

**For testing purpose put this configuration**

**==========================================**

<add key="FromDate" value="2013-2-21 13:47:45.000" />

<add key="ToDate" value="2013-2-28 13:55:24.000" />

<add key="WorkGroupID" value="" />

<add key="Interval" value="180000" />

**For live make the configuration as this**

<add key="FromDate" value="" />

<add key="ToDate" value="" />

<add key="WorkGroupID" value="" />

<add key="Interval" value="180000" />

1. Now take the cmd prompt in administrative mode, and go to the folder

C:\Windows\Microsoft.NET\Framework\v4.0.30319

And run the command, InstallUtil.exe "D:\UtilityService\UtilityService.exe”

1. In the windows, type services.msc, in that you can able to see the file UtilityService in the list. Right click on that start the service

**Things to be taken care**

IN SQL Server, SQL Server Authentication need to be configured

Please make the antivirus as disabled as this will stop the creation of excel reports in service

Run the scripts

USE [qm]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ReportsGenerated] Script Date: 10/05/2018 09:05:09 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ReportsGenerated](

[ID] [int] IDENTITY(1,1) NOT NULL,

[ReportGeneratedFileName] [varchar](max) NULL,

[CreatedOn] [datetime] NULL,

[CreatedBy] [varchar](max) NULL,

[MethodofCreation] [varchar](max) NULL,

[ReportGeneratedFullPath] [varchar](max) NULL,

[ReportLocation] [varchar](max) NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetAllWorkGroups] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[GetAllWorkGroups]

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

Select workgroupid, workgroupname from workgroup where status = 1

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetReport] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Mujeeb K S

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[GetReport]

@startdate datetime,

@endDate datetime,

@workgroupid varchar(max)

AS

BEGIN

with CTEGetReport (starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored)

as

(select distinct m.starttime, m.endtime, m.mediaid, m.dnis, m.ani, r.updateuserid,

r.percentscore, r.overallscore,

r.reviewdate, i.username, i.userroleid, i.usertypeid,

w.workgroupname, s.[description], s.[name], s.sequencenumber,

q.questiondescription, q.questionnumber, q.questiontext, q.responserequired,

q.questionadditionalpoint, q.autofailpoint, q.questionadditionalconditionpoint,

sr.weightedscore, s.[weight] as sectionWeight,

qr.responsetext, q.[weight] as questionWeight,

qt.questiontypedesc, qt.scored as questionScored

from media m inner join review r on m.mediaid = r.mediaid

inner join iqmuser i on i.userid = r.owneruserid

inner join workgroup\_iqmuser wi on wi.userid = r.owneruserid

inner join workgroup w on w.workgroupid = wi.workgroupid

inner join sectionresult sr on sr.reviewid = r.reviewid

inner join section s on s.sectionid = sr.sectionid

inner join question q on q.sectionid = s.sectionid

inner join questionresult qr on q.questionid = qr.questionid

inner join questiontype qt on qt.questiontypeid = q.questiontypeid

where (m.starttime BETWEEN @startdate AND @endDate) OR

(m.endtime BETWEEN @startdate AND @endDate) OR

(m.starttime <= @startdate AND m.endtime >= @endDate) and

w.workgroupid in ( @workgroupid ))

select starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored from CTEGetReport

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetReportcsvList] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[GetReportcsvList]

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

--SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT ID, ReportGeneratedFileName, CreatedOn, CreatedBy, MethodofCreation, ReportGeneratedFullPath, ReportLocation

FROM ReportsGenerated order by ID desc

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetReportDailyJob] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Mujeeb K S

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[GetReportDailyJob]

@startdate datetime,

@endDate datetime,

@workgroupid uniqueidentifier = null

AS

BEGIN

if (@workgroupid is null)

begin

with CTEGetReport (starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored)

as

(select distinct m.starttime, m.endtime, m.mediaid, m.dnis, m.ani, r.updateuserid,

r.percentscore, r.overallscore,

r.reviewdate, i.username, i.userroleid, i.usertypeid,

w.workgroupname, s.[description], s.[name], s.sequencenumber,

q.questiondescription, q.questionnumber, q.questiontext, q.responserequired,

q.questionadditionalpoint, q.autofailpoint, q.questionadditionalconditionpoint,

sr.weightedscore, s.[weight] as sectionWeight,

qr.responsetext, q.[weight] as questionWeight,

qt.questiontypedesc, qt.scored as questionScored

from media m inner join review r on m.mediaid = r.mediaid

inner join iqmuser i on i.userid = r.owneruserid

inner join workgroup\_iqmuser wi on wi.userid = r.owneruserid

inner join workgroup w on w.workgroupid = wi.workgroupid

inner join sectionresult sr on sr.reviewid = r.reviewid

inner join section s on s.sectionid = sr.sectionid

inner join question q on q.sectionid = s.sectionid

inner join questionresult qr on q.questionid = qr.questionid

inner join questiontype qt on qt.questiontypeid = q.questiontypeid

where (m.starttime BETWEEN @startdate AND @endDate) OR

(m.endtime BETWEEN @startdate AND @endDate) OR

(m.starttime <= @startdate AND m.endtime >= @endDate))

select starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored from CTEGetReport

end

else

begin

with CTEGetReport (starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored)

as

(select distinct m.starttime, m.endtime, m.mediaid, m.dnis, m.ani, r.updateuserid,

r.percentscore, r.overallscore,

r.reviewdate, i.username, i.userroleid, i.usertypeid,

w.workgroupname, s.[description], s.[name], s.sequencenumber,

q.questiondescription, q.questionnumber, q.questiontext, q.responserequired,

q.questionadditionalpoint, q.autofailpoint, q.questionadditionalconditionpoint,

sr.weightedscore, s.[weight] as sectionWeight,

qr.responsetext, q.[weight] as questionWeight,

qt.questiontypedesc, qt.scored as questionScored

from media m inner join review r on m.mediaid = r.mediaid

inner join iqmuser i on i.userid = r.owneruserid

inner join workgroup\_iqmuser wi on wi.userid = r.owneruserid

inner join workgroup w on w.workgroupid = wi.workgroupid

inner join sectionresult sr on sr.reviewid = r.reviewid

inner join section s on s.sectionid = sr.sectionid

inner join question q on q.sectionid = s.sectionid

inner join questionresult qr on q.questionid = qr.questionid

inner join questiontype qt on qt.questiontypeid = q.questiontypeid

where (m.starttime BETWEEN @startdate AND @endDate) OR

(m.endtime BETWEEN @startdate AND @endDate) OR

(m.starttime <= @startdate AND m.endtime >= @endDate) and

w.workgroupid in ( @workgroupid ))

select starttime, endtime, mediaid, dnis, ani, updateuserid, percentscore, overallscore, reviewdate, username,

userroleid, usertypeid, workgroupname, [description], [name], sequencenumber, questiondescription, questionnumber,

questiontext, responserequired, questionadditionalpoint, autofailpoint, questionadditionalconditionpoint, weightedscore,

sectionWeight, responsetext, questionWeight, questiontypedesc, questionScored from CTEGetReport

end

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[GetReportList] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

CREATE PROCEDURE [dbo].[GetReportList]

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

Select ID, ReportGenerated, CreatedOn, CreatedBy from ReportsGenerated order by ID desc

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[InsertReport] Script Date: 10/05/2018 09:05:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[InsertReport]

@ReportGeneratedFileName varchar(MAX),

@CreatedOn datetime,

@CreatedBy varchar(MAX),

@MethodofCreation varchar(MAX),

@ReportGeneratedFullPath varchar(MAX),

@ReportLocation varchar(MAX)

AS

BEGIN

--SET NOCOUNT ON;

INSERT INTO [dbo].[ReportsGenerated]

(

ReportGeneratedFileName,

CreatedOn,

CreatedBy,

MethodofCreation,

ReportGeneratedFullPath,

ReportLocation

)

VALUES (

@ReportGeneratedFileName,

@CreatedOn,

@CreatedBy,

@MethodofCreation,

@ReportGeneratedFullPath,

@ReportLocation

);

END

GO