meil_report_new.sh   script

```
#! /bin/bash

set -x –v

filepath = $1

# Calculating real date of log file for future using

Year=$(stat $filepath |grep Modify|awk '{print $2}'|awk -F '-' '{print $1}')

daymonth =$(stat $filepath |grep Modify|awk '{print $2}'|awk -F '-' '{print $2"-"$3}')

if [ $daymonth == "01-01" ]

then

      year=`expr $year - 1`

fi

# Going to working directory of script

cd /opt/Chmail/jetty/logs/mail_report_new.dir

# Creating final csv file named containing date

filename=$(zcat $filepath|awk '{print """$year"'""-"$1"-"$2}'|head -1)

touch "log-$filename.csv"

# For better performance, we first search “postfix” pattern in log file and save these lines into a temporary file. Because all rows that have send and receive reports, containing postfix pattern.

zgrep postfix $filepath > postfix_tmpfile

# By using below commands, we can extract all “from” patterns that contained “time stamp of send time” and “queue number of email” and  “sender address” in a comma-separated format and save these lines to a temporary file

grep "postfix/qmgr" postfix_tmpfile|grep from|awk '{print $1"\t"$2"\t"$3","$6","$7","$9}'|sed 's/\://3'|sed 's/from=<//g'|sed 's/>,//g'|sed 's/nrcpt=//g'|awk -F',' '{system("echo "$0";date -d \""$1" """$year"'"\" \"+%s\"")}'|sed 'N;s/\n/,/'|awk -F "," '{print $NF","$2","$3}'|grep -v ",$"|sort -u -t ',' -k2,2 >senders_tmp

# By using below commands, we can extract all “to” patterns that contained “time stamp of receive time” and “queue number of email” and “recipient address” and “DSN number” and “comments” in a comma-separated format and save these lines to a temporary file

grep "dsn=" postfix_tmpfile|awk -F "status=" '{print $1" "$2}'|sed -e 's/ conn_use=[0-9]*, / /'|sed -e 's/orig_to=<.*>,//g'|awk '{s = ""; for (i = 13; i <= NF; i++) s = s $i " ";system("date -d \""$1" "$2" "$3" """$year"'"\" \"+%s\"");print "#B#"$6"#B#"$7"#B#"$11"#B#"s}'|sed 's/://1'|sed 'N;s/\n//'|sed
```

```
's/to=<//g'|sed 's/>,//g'|sed 's/,//g'|sed 's/#B#/,/g'|sed 's/dsn=//'|grep -Ev "from MTA\(\[127.0.0.1\]:10025\): 250 2.0.0 Ok: queued as|from MTA\(smtp:\[127.0.0.1\]:10025\): 250 2.0.0 Ok: queued as"|sort -t ',' -k 2 >delivery_tmp
```

# After creating two former files, we join these files based on “queue number” column and saving result into another temporary file

```
join -t ',' -j 2 -o 1.3,2.3,1.1,2.1,1.2,2.4,2.5 senders_tmp delivery_tmp > join_tmp
```

# Saving rows that “DSN number” of them is 2.x.x(successful send) or 5.x.x(rejected or bounced)   into final report file

```
grep -v "^.*,.*,.*,.*,.*,4.*" join_tmp > "log-$filename.csv"
```

# Unifying duplicate rows in differed file based on “recipient address” and “time stamp of send time” and “time stamp of receive time” and “queue number” columns and saving them to another file

```
awk -F"," '!seen[$2, $5]++' deferred_file | sort -t ',' -k 5  > deferred_file_uniqued
```

# Then compare old days differed mails with today’s recipients and add delivered or rejected mails (dsn=2.x.x , 5.x.x) to final report file

```
join -t ',' -1 5 -2 2 -o 1.1,1.2,1.3,2.1,1.4,2.4,2.5 deferred_file_uniqued delivery_tmp|grep -v "^.*,.*,.*,.*,.*,4.*" >> "log-$filename.csv"
```

# Then removing rows that their status became clear (dsn= 2.x.x or 5.x.x) and recreating differ file

```
join -t ',' -1 5 -2 2 -o 1.1,1.2,1.3,2.1,1.5,2.4,2.5 deferred_file_uniqued delivery_tmp|grep "^.*,.*,.*,.*,.*,4.*"|sort -t ',' -k 5 > deferred_file
```

# Bellow commands used for creating differed file that used in future days for delivery report

```
grep "^.*,.*,.*,.*,.*,4.*" join_tmp >> deferred_file
```

# At the end, removing sso.eblagh to sso.eblagh from final file and save remaining rows in another file that used by Log server

```
grep -Ev '^sso.eblagh@tamin.ir,sso.eblagh@tamin.ir,.*' "log-$filename.csv" > /opt/Chmail/jetty/logs/mail_report_mssql.dir/"log-$filename.csv"

###END###
```

Defining meil_report_new.sh   executable and crontab setting

```
Chmod +x /opt/Chmail/script/meil_report_new.sh

Crontab -e

        0 2 * * * /opt/Chmail/script/meil_report_new.sh /var/log/Chmail.log-$(date +\%Y\%m\%d).gz
```

mail_report_alldays_new.sh script

```
#! /bin/bash

set -x -v

cd /opt/Chmail/jetty /logs/mail_report_alldays.dir

for i in $(ls /var/log/Chmail.log-*);do

echo $i >> script.log

filepath="$i"

year=$(stat $filepath |grep Modify|awk '{print $2}'|awk -F '-' '{print $1}')

daymonth=$(stat $filepath |grep Modify|awk '{print $2}'|awk -F '-' '{print $2"-"$3}')

if [ $daymonth == "01-01" ]

then

        year=`expr $year - 1`

fi

filename=$(zcat $filepath|awk '{print "'"$year"'""-"$1"-"$2}'|head -1)

touch "log-$filename.csv"

zgrep postfix $filepath > postfix_tmpfile

grep "postfix/qmgr" postfix_tmpfile|grep from|awk '{print $1"\t"$2"\t"$3","$6","$7","$9}'|sed 's/\://3'|sed 's/from=<//g'|sed 's/>,//g'|sed 's/nrcpt=//g'|awk -F',' '{system("echo "$0";date -d \""$1" "'"$year"'"\" \"+%s\"")}'|sed 'N;s/\n/,/'|awk -F "," '{print $NF","$2","$3}'|grep -v ",$"|sort -t ',' -k 2 >senders_tmp

grep "dsn=" postfix_tmpfile|awk -F "status=" '{print $1" "$2}'|sed -e 's/ conn_use=[0-9]*, / /'|sed -e 's/orig_to=<.*>,//g'|awk '{s = ""; for (i = 13; i <= NF; i++) s = s $i " ";system("date -d \""$1" "$2" "$3" "'"$year"'"\" \"+%s\"");print "#B#"$6"#B#"$7"#B#"$11"#B#"s}'|sed 's/://1'|sed 'N;s/\n//'|sed 's/to=<//g'|sed 's/>,//g'|sed 's/,//g'|sed 's/#B#/,/g'|sed 's/dsn=//'|grep -Ev "from MTA\(\[127.0.0.1\]:10025\): 250 2.0.0 Ok: queued as"|sort -t ',' -k 2 >delivery_tmp

join -t ',' -j 2 -o 1.3,2.3,1.1,2.1,1.2,2.4,2.5 senders_tmp delivery_tmp > join_tmp

grep -v "^.*,.*,.*,.*,.*,4.*" join_tmp > "log-$filename.csv"

awk -F"," '!seen[$2, $5]++' deferred_file | sort -t ',' -k 5  > deferred_file_uniqued

join -t ',' -1 5 -2 2 -o 1.1,1.2,1.3,2.1,1.4,2.4,2.5 deferred_file_uniqued delivery_tmp|grep -v "^.*,.*,.*,.*,.*,4.*" >> "log-$filename.csv"

join -t ',' -1 5 -2 2 -o 1.1,1.2,1.3,2.1,1.5,2.4,2.5 deferred_file_uniqued delivery_tmp|grep "^.*,.*,.*,.*,.*,4.*"|sort -t ',' -k 5 > deferred_file
```

```
grep "^.*,.*,.*,.*,.*,4.*" join_tmp >>deferred_file

echo -e "It's done\n"

grep -Ev '^sso.eblagh@tamin.ir,sso.eblagh@tamin.ir,.*' "log-$filename.csv" > /opt/Chmail/jetty/logs/mail_report_mssql.dir/"log-$filename.csv"


done
```

MS SQL section

---- Query1---------------------------------------

```
CREATE DATABASE [tamin]            ➔ for creating database
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'tamin', FILENAME = N'E:\MSSQL\DATA\tamin.mdf' , SIZE = 8134656KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'tamin_log', FILENAME = N'E:\MSSQL\DATA\tamin_log.ldf' , SIZE = 13180928KB ,
MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
GO

------------------------------------
```

---- Query2---------------------------------------

```
USE [tamin]                  ➔ for creating table
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[log](
	[Srecipient] [nvarchar](200) NOT NULL,
	[Rrecipient] [nvarchar](200) NOT NULL,
	[Stime] [float] NULL,
	[Deliverytime] [float] NULL,
	[Qid] [nvarchar](100) NOT NULL,
	[Dsn] [nvarchar](60) NULL,
	[Comment] [nvarchar](2000) NULL,
PRIMARY KEY CLUSTERED
(
	[Srecipient] ASC,
	[Rrecipient] ASC,
	[Qid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

---- Query3--

```sql
USE [tamin]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[users](
	[user] [nvarchar](50) NULL,
	[password] [nvarchar](50) NULL
) ON [PRIMARY]
GO
```

---

import-logs.ps1 script

```powershell
# Definition of parameters

param (

	$localPath = "E:\DB-Sync\Logs\",    # Path of log files

	# Path of csv files in remote server ( mail server )

	$remotePath = "/opt/Chmail/jetty/logs/mail_report_mssql.dir/",

	$database = 'tamin',

	$csvPath = "E:\DB-Sync\csv-files\",    # Local path of csv files

	$server = 'maillog',

	$table = 'dbo.log',

	$logServer = 'mail-1.tamin.ir',

	$user = 'logtransporter',

	$pass = '0DyJmyOpGhChaBxW7X53',

	$serverfingerprint='ssh-rsa 2048 dc:20:41:fe:70:19:18:f6:39:30:5b:b9:82:ff:25:ce',

	$sshlogname = 'ssh' ,

	$errlogname = 'error' ,

	$mylogname = 'sqlerr',

	$mailserver = 'mail-1.tamin.ir',

	$mailfrom = 'taminlogger <taminloger@taminlog.local>',

	$mailto =  'log-acc@tamin.ir'

	)
```

```
try

{

# Turns script debugging features on and off, sets the trace level, and toggles strict mode.

       set-psdebug -off

       #set-psdebug -trace 2

# Variable for printing row number of csv file that any error occurred for it in sql import section . This number must become zero at the beginning of script.

       $n = 0

# The log files create with names that contained today date ; But CSV file that script is working with it , has a name contained yesterday date

       $date = Get-Date

       $logtime = $date.ToString('yyyy-MMM-d__hh')

       $date=$date.AddDays(-1)

       $d= $date.ToString('yyyy-MMM-d')

       $errorLog = $localPath + $errlogname + $d + '.log'

       $mylog = $localPath + $mylogname + $d + '.log'

       $file = "log-" + $d + ".csv"

       $path = $csvPath + $file

       $sshlog = $localPath + $sshlogname + $logtime + '.log'

# Appending path of csv files in local server to log files

       $path | Out-File -filepath $myLog –Append

       $path | Out-File -filepath  $errorLog -Append

# Mail Subject and Body

       $mailsubject = "import log status of " + $d

       $mailbody = "see attachments " + $logtime

 # Load WinSCP .NET assembly

 #The Add-Type cmdlet adds a Microsoft .NET Framework class in your Windows PowerShell session.

    Add-Type -Path "E:\DB-Sync\WinSCPnet.dll"

# Setup session options

# Defines information to allow an automatic connection and authentication of the session. It is used with the Session.Open and Session.ScanFingerprint methods.

# https://winscp.net/eng/docs/library_session

        $sessionOptions = New-Object WinSCP.SessionOptions -Property @{
```

```
        Protocol = [WinSCP.Protocol]::Sftp

        HostName = $logServer

        UserName = $user

        Password = $pass

        SshHostKeyFingerprint = $serverfingerprint

    }

 # Define information about WinScp session like debug level and ssh log path

    $session = New-Object WinSCP.Session -Property @{

       #DebugLogLevel="0"

       #DebugLogPath=$localPath + "ssh-debug.log"

       SessionLogPath= $sshlog


       }


# try-catch function : try { NonsenseString } catch { "An error occurred." }

    try

    {

        # Format timestamp

        #$stamp = $(Get-Date -Format "yyyyMMddHHmmss")

        # Opens the session

        $session.Open($sessionOptions)

        # Download the file and throw on any error

        $session.GetFiles(($remotePath + $file),($csvPath )).check()

    }

# if an error occurred , it’s logged to errorlog and mailed the log to admin and exit
from script

    catch {

       $_ | Out-File -filepath  $errorLog -Append

       $mailsubject = "Error" + $mailsubject

       send-mailmessage  -SmtpServer  $mailserver  -From  $mailfrom  -to  $mailto  -Subject
$mailsubject -BodyAsHtml -Body $mailbody -Attachments $errorLog

       exit 1

 }
```

```
# if file transferred correctly , close the session

       finally

    {

        # Disconnect, clean up

        $session.Dispose()

    }

# After transferring csv file from mail server to log server, we must import it to MS SQL
database for application use
```

# The **Import-Csv** cmdlet provides a way for you to read in data from a comma-separated values file (CSV) and then display that data in tabular format within the Windows PowerShell console. ForEach-Object Performs an operation against each item in a collection of input objects.

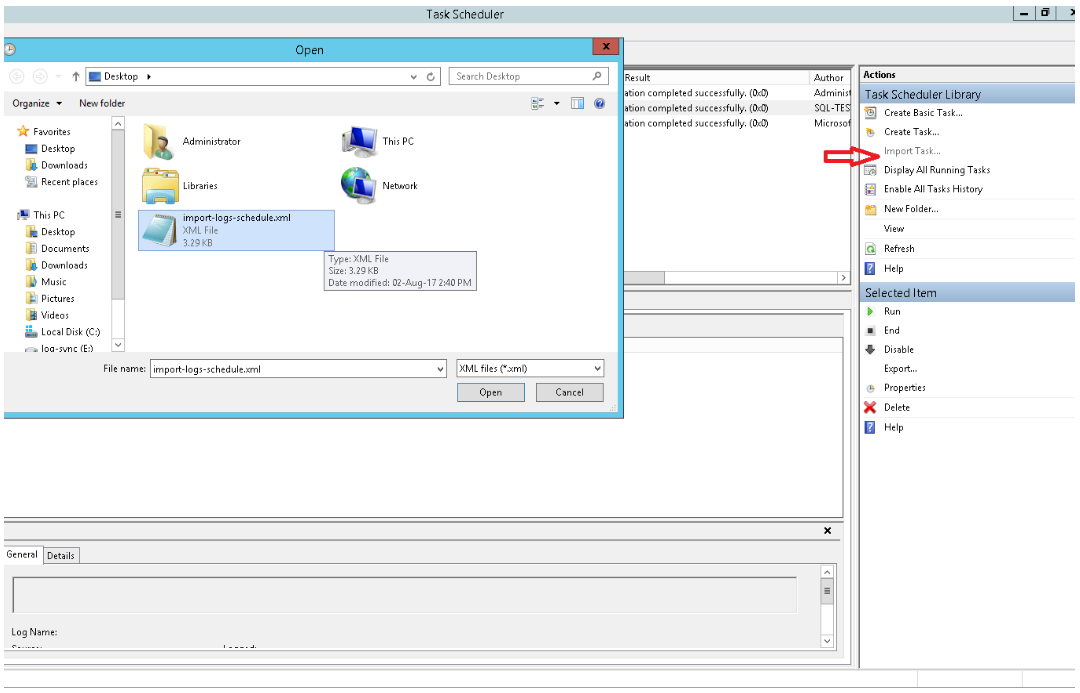
```
       Import-Csv -Path $path -Header Srecipient, Rrecipient, Stime, Deliverytime, Qid,
Dsn, Comment |  ForEach-Object {

       $n++      # row number

 # Replacing ‘ with “ in comment column for preventing sql import error

       $escaped = $_.Comment.Replace("'", "''")

try {
```

# The **Invoke-Sqlcmd** cmdlet lets you run your **sqlcmd** script files in a Windows PowerShell environment. Much of what you can do with **sqlcmd** can also be done using **Invoke-Sqlcmd**.

```
              Invoke-Sqlcmd  -ErrorVariable sqlerror  -Database $database -ServerInstance
$server          -Query          "insert          into          $table          VALUES
('$($_.Srecipient)','$($_.Rrecipient)','$($_.Stime)','$($_.Deliverytime)','$($_.Qid)','$(
$_.Dsn)','$escaped')"

# If any error occurred during importation of a row in database, print row number and
error to sql error log file

   if ($sqlerror) {

      $n | Out-File -filepath $myLog -Append

      $sqlerror | Out-File -filepath $myLog -Append

       }

}

# If a suddenly or general error occurred, it’s logged to error log file and mail this
file to admin and exit from script

catch {

       $_ | Out-File -filepath  $errorLog -Append

       $mailsubject = "Error" + $mailsubject
       send-mailmessage  -SmtpServer $mailserver   -From $mailfrom -to $mailto -Subject
$mailsubject -BodyAsHtml -Body $mailbody -Attachments $errorLog

       exit 1

 }

 }    # end of ForEach-Object ( importing sql section )

# If before sections passed correctly, all log files mailed to admin

       $attachfile = $mylog , $sshlog , $errorLog

       send-mailmessage  -SmtpServer $mailserver   -From $mailfrom -to $mailto -Subject
$mailsubject -BodyAsHtml -Body $mailbody -Attachments $attachfile

       #Write-Host $sqlerror



}

# If any error occurred in before sections , it’s logged to error log file and mailed it
to admin and exit from script

catch

{

    $_ | Out-File -filepath $errorLog -Append

    $mailsubject = "Error" + $mailsubject

   send-mailmessage  -SmtpServer $mailserver    -From $mailfrom  -to $mailto  -Subject
$mailsubject -BodyAsHtml -Body "see error" -Attachments $errorLog

       exit 1

}
```

Creating a task schedule for running powershell script

We create a xml file named import-logs-schedule.xml that contain settings of task scheduler . You can edit it and then import it to your log server :



Apache Tomcat Section

Download the latest version of Apache Tomcat for windows from below link:

http://tomcat.apache.org/

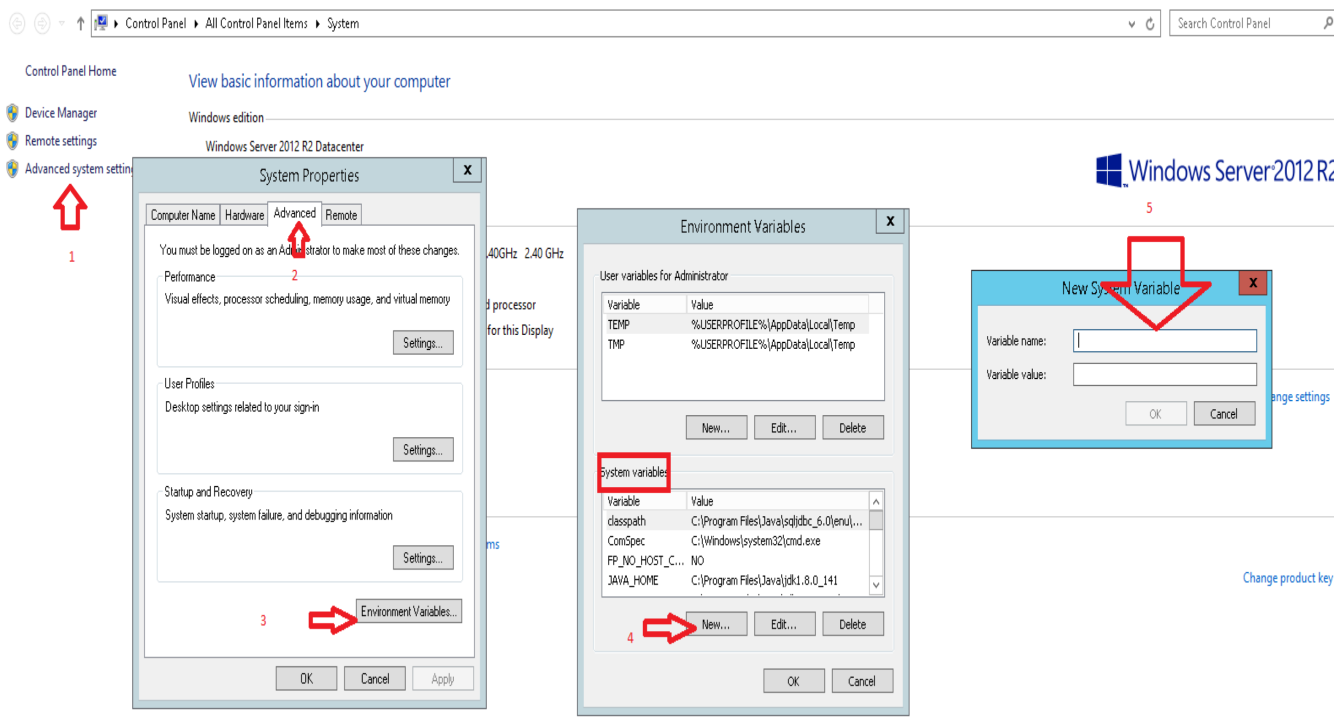
and install it !!

Download the latest version of JRE and install it :

```
md "C:\Program Files\Java" xcopy D:\Temp\jdk1.8.0_141*.* "C:\Program Files\Java" /s
```

Download jdbc driver and extract it into C:\Program Files\Java .

Then we must set system variables :



Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
classpath	C:\Program Files\Java\sqljdbc_6.0\enu\...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_141

1. Variable Name: JAVA_HOME

   Variable value: C:\Program Files\Java\jdk1.8.0_141

2. Variable Name: classpath
   Variable value: C:\Program Files\Java\sqljdbc_6.0\enu\sqljdbc4.jar

3. Variable Name: JRE_HOME
   Variable value: C:\Program Files\Java\jdk1.8.0_141\jre

Edit configuration files of tomcat for using certificate:

C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\ server.xml

```
<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="443" />

<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"

acceptorThreadCount="16"

SSLEnabled="true"      sslProtocol="TLSv1.2"   sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1"
scheme="https" secure="true"


ciphers="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,

               TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA"
```

```
allowTrace="false"

disableUploadTimeout="true"

enableLookups="false"

acceptCount="100"

keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat 8.5\keystore\tamin"

keystorePass="jnGm6O"

maxConnections="14000"

maxHeaderCount="20"

maxKeepAliveRequests="100"

maxPostSize="524288"

maxThreads="250"

maxHttpHeaderSize="2048"

pollerThreadCount="4"

server="KWS/1.0"

clientAuth="false"

connectionTimeout="10000"

compression="on"                                                    compressionMinSize="512"
compressableMimeType="text/css,text/html,text/javascript,text/plain,text/xml,application/
ecmascript,application/javascript,application/json,application/vnd.ms-
fontobject,application/x-ecmascript,application/xhr,text/xhr"

/>
```

C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\web.xml

Add below lines before </web-app>

```
<security-constraint>

 <web-resource-collection>

 <web-resource-name>Protected Context</web-resource-name>

 <url-pattern>/*</url-pattern>

 </web-resource-collection>

 <!-- auth-constraint goes here if you requre authentication -->

 <user-data-constraint>

 <transport-guarantee>CONFIDENTIAL</transport-guarantee>

 </user-data-constraint>

 </security-constraint>
```

After setting above configuration, copy .war file into “C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps” folder ; And then restart tomcat .

(windows logo) + R -> services.msc -> apache tomcat -> restart

Note that files of web application must be copy to the ROOT folder and restart tomcat again.

At the end, we can create another user for connection between database and web application :

```
USE [master]
GO

/* For security reasons the login is created disabled and with a random password. */
/****** Object:  Login [sa]    Script Date: 02-Aug-17 3:50:55 PM ******/
CREATE LOGIN [username] WITH PASSWORD=password', DEFAULT_DATABASE=[master],
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=ON
GO

ALTER LOGIN [username] DISABLE
GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [username]
GO
```

And then change related config file in web apps of tomcat:

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps\ROOT\WEB-INF\classes\Dbconfig.properties

```
#tamin db config

sql.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver

sql.url=jdbc:sqlserver://localhost:1433

sql.name=tamin

sql.username=username

sql.password=password
```