web.config (inside root folder)

```
<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
		<connectionStrings>
			<add name="w10Stud022ConnectionString" connectionString="Data
Source=idedb2.hh.se;Initial Catalog=w10Stud022;Persist Security Info=True;User
ID=w10Stud022;Password=ylva01" providerName="System.Data.SqlClient"/>
			<!-- connectionstring to change the login function to work with
idedb2.hh.se-->
			<add name="myMembership" connectionString="Data
Source=idedb2.hh.se;Initial Catalog=aspnetdb;Persist Security Info=True;User ID=login;Password=login"
providerName="System.Data.SqlClient"/>
		</connectionStrings>
		<system.web>
			<compilation debug="true" targetFramework="4.0"/>
			<!-- Those rows is needed to handle roles -->
			<authentication mode="Forms"></authentication>
			<roleManager enabled="true"></roleManager>
			<membership>
				<providers>
					<!-- Cleans earlier provider name to
redefine it (Don't forget this one otherwise you cant specify the new AspNetSqlMembershipProvider) -->
					<clear/>
					<!-- this is copied from machine.config i
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config OBS!: remember to change the
connectionStringName to myMembership, otherwise its not used -->
					<add name="AspNetSqlMembershipProvider"
type="System.Web.Security.SqlMembershipProvider, System.Web, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" connectionStringName="myMembership" enablePasswordRetrieval="false"
enablePasswordReset="true" requiresQuestionAndAnswer="false" applicationName="/"
requiresUniqueEmail="false" passwordFormat="Hashed" maxInvalidPasswordAttempts="5"
minRequiredPasswordLength="6" minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10"
passwordStrengthRegularExpression=""/>
				</providers>
			</membership>
		</system.web>
  <location path="mail.aspx">
    <system.web>
<!--location tag: Allows all in the role admin to access the mail.aspx file (all other
users is denied access). Its only affects the specified file inside this location tag
(mail.aspx) -->
      <authorization>
        <allow roles="admin"/>
        <deny users="*"/>
      </authorization>
    </system.web>
  </location>
  <!—This is an example of how you can configure a g-mail account to send mail using asp.net. You need
to configure your gmail account to allow sending mail (activate IMAP) -->
		<system.net>
			<mailSettings>
				<smtp>
					<network enableSsl="true" port="587"
host="smtp.gmail.com" userName="jesper.hakerod@gmail.com" password="yourpassword"/>
				</smtp>
			</mailSettings>
		</system.net>
</configuration>
```

web.config (inside admin subfolder)

```
<?xml version="1.0"?>
<configuration>
    <system.web>
      <!-- Allows all in the role admin to access the content in "admin"-folder (all other users is denied access) You can create an admin role and set the role to admin manually by using the ASP.NET configuration tool (above the Solution Explorer)-->
      <authorization>
        <allow roles="admin"/>
        <deny users="*"/>
      </authorization>
    </system.web>
</configuration>
```

mail.aspx.cs (codebehind file for sending email)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net.Mail; // need to load those classes for sending mail

public partial class mail : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        // compare to what you made in laboratory work (creating an instance of the calculator-class), now you work with classes inside the framework.net instead. You need to load them in the top of this side to use them here (see using System.Net.Mail row)
        MailMessage myMail = new MailMessage();
        myMail.Subject = subject.Text;
        myMail.Body = message.Text;
        myMail.From = new MailAddress("jesper.hakerod@gmail.com", "Jesper Hakeröd");
        myMail.To.Add(new MailAddress("jesper.hakerod@gmail.com", "Jesper Hakeröd"));
        SmtpClient mySmtp = new SmtpClient();
        mySmtp.Send(myMail);
        // prints out a simple indication of that something been sent...
        response.Text="Thanks for your message!";
    }
}
```

Inside the database:

When you create tables you need to define a primary key to get an unique identifier for each row. You need a primary key to point at only one single row in case of updating or delete rows inside the database table (for instance by using a gridview and datasource in an aspx.net page).

To let the key count up automatically for each new row you add in a database table, you use the identity option on the primary key. Then it starts at 1 and counts up 1 for each new row you add inside that database table (first row receives 1, second row receives 2, third row receives 3, and so on).