Environment Setup

Java:

In order to run the code in this course you'll need to have Java 17 or higher installed.

About Maven:

All of the projects for this class will be based on maven (http://maven.apache.org/). Maven is a utility to organize and build a project (similar to what IDE's do, but without the GUI part).

Maven uses an XML file called pom.xml (Project Object Model) that keeps track of where your source code is, where your class files should go, what jar files (libraries) your project depends on, and even where to download these libraries from.

It is good to install the maven commandline tools on your path. I believe IntelliJ may come bundled with Maven, but having the tools on your path gives you a lot of extra flexibility. Other IDEs (like Visual Studio Code) require the commanline tools (see installation below).

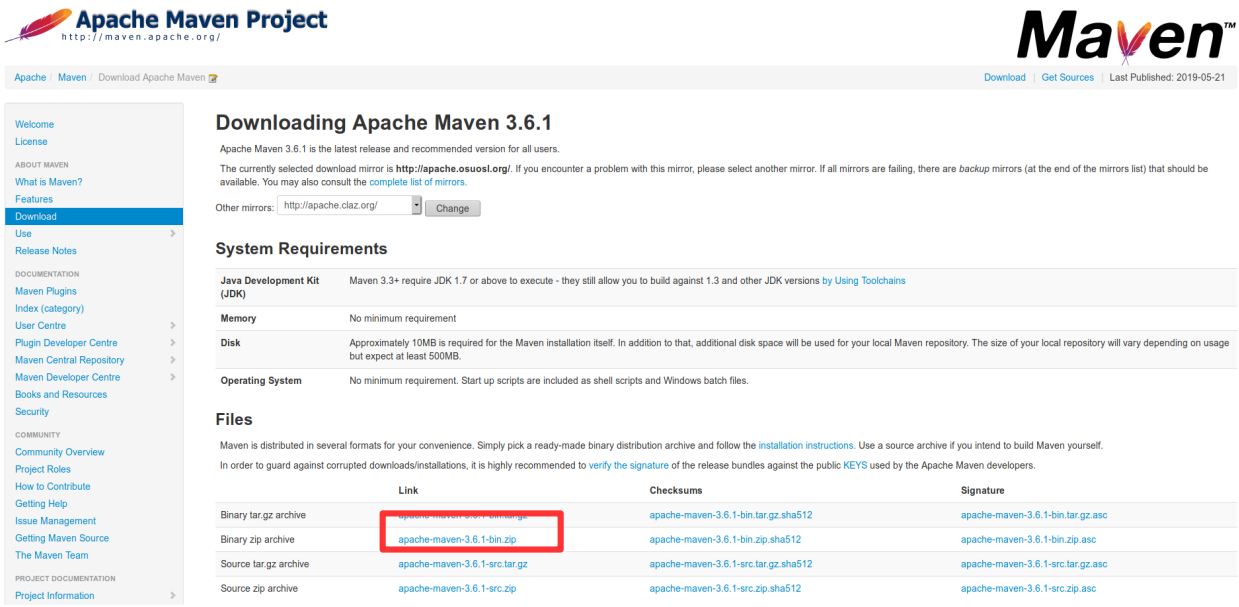
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>edu.mum.cs544</groupId>
    <artifactId>exercise02_1</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>exercise02_1</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.1.7.Final</version>
        </dependency>
        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <version>8.0.32</version>
        </dependency>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.20.0</version>
        </dependency>
    </dependencies>
</project>
```

Basic project naming, no need to specify directories, uses defaults

Jar files that we need for the project

Installing Maven:

Go to: http://maven.apache.org/download.cgi and download the binary zip archive.



Unpack it wherever you want (say: C:\Program Files\apache-maven\) and then add the bin directory (**`C:\Program Files\apache-maven\bin\`**) to your path.

Very important: if you don't know what you're doing (have never worked with the path before) then only add to the PATH value, do not overwrite or remove parts of it!

For a guide on how to add something to your path on Windows 10 see:
https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/

Integrated Development Environment:

I'm happy for students to use whatever IDE they want. Nevertheless people sometimes just want to follow along with screenshots, and to that extend I will illustrate certain actions with IntelliJ and Visual Studio code.

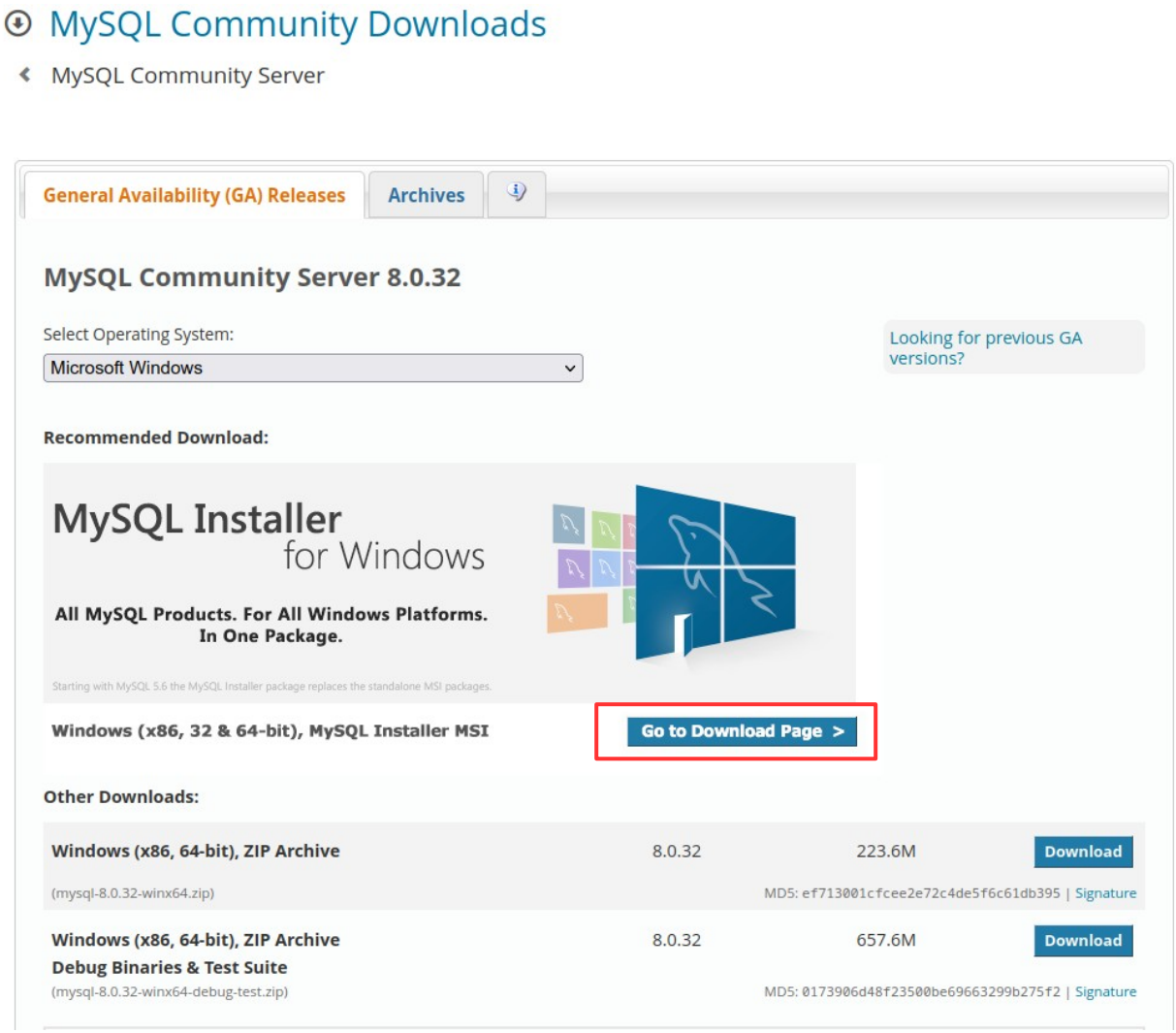
This does not mean you have to use either of IntelliJ or Visual Studio Code, as there always are ways to achieve the same thing with a different IDE as well.

This will first time I'm using IntelliJ for this course (June 2023) you may very well have suggestions for me on how to use it, please speak up if you see me struggling or doing something weird with IntelliJ.

MySQL Database:

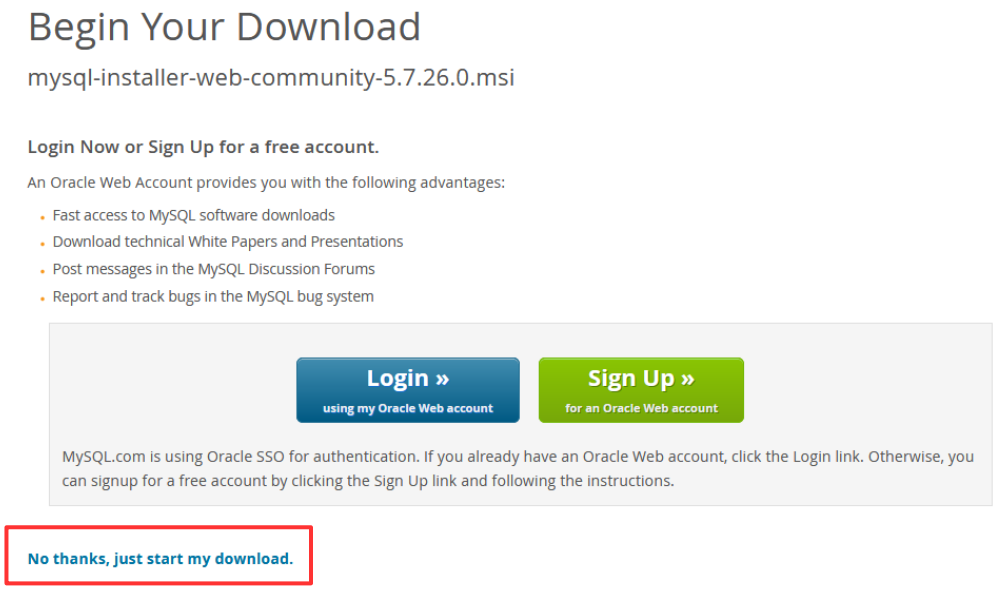
Many of the exercises in this course (especially those related to Hibernate) require a database. We will use MySQL since it is free and relatively widely used in the industry. You can download MySQL from any of the following locations:

The MySQL windows installer, which will run MySQL on startup found at:
https://dev.mysql.com/downloads/mysql/



The latest version is 8.0.32, be sure to download the Windows Installer, the zip version takes a lot more steps to setup.

Once you click download it will try to tell you that you should login or sign up, but you can simply click on the small "No Thanks" further down to get your download

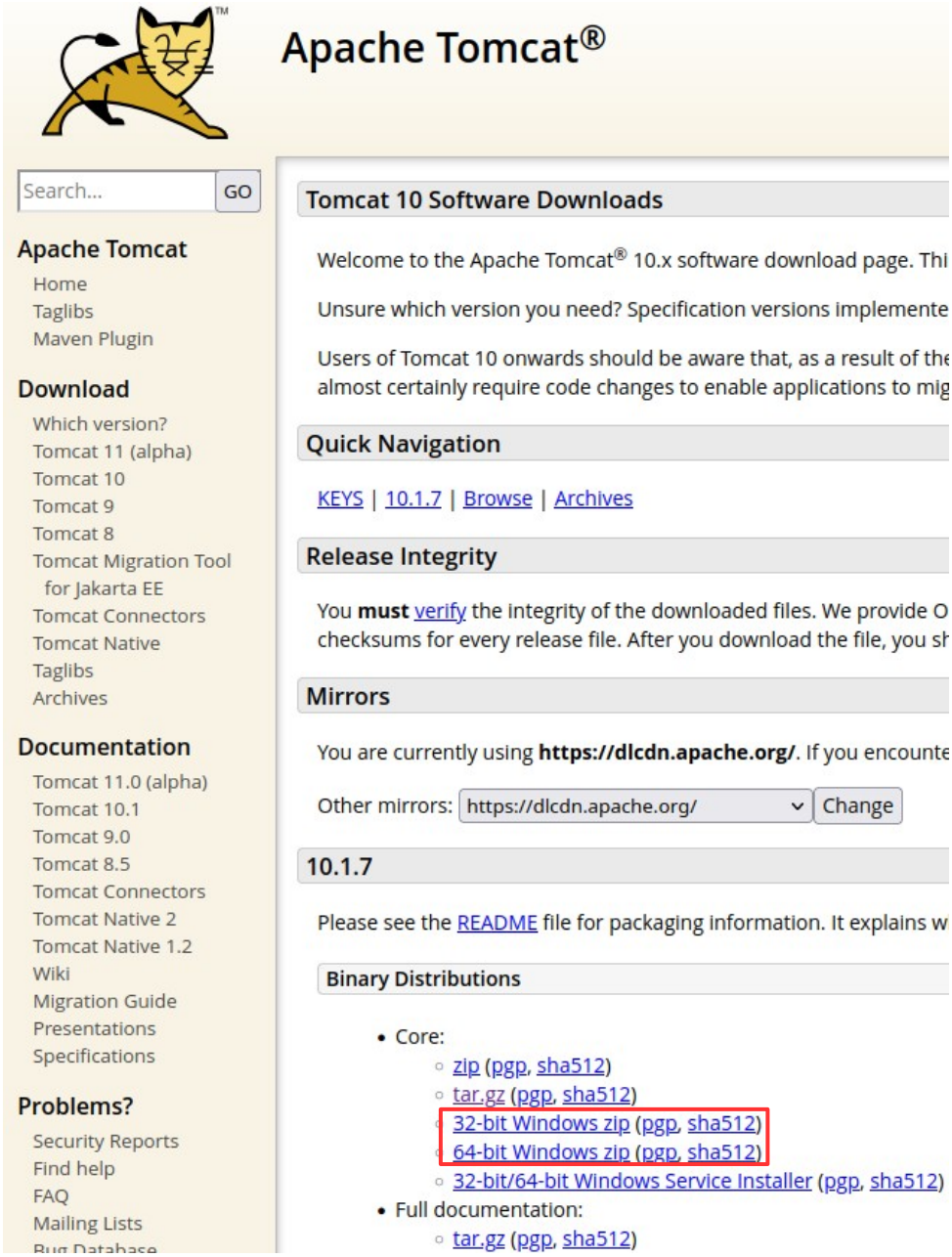


When installing MySQL it may complain about missing the Visual C++ Redistributable package 2013. Download and install it from: https://www.microsoft.com/en-us/download/details.aspx?id=40784

Important: while installing MySQL it will ask you to provide a root password. Chose something that you will remember! Or just go for something stupid like: **root** (that's what I did, although it's terrible for security – don't do this if you have sensitive data)

Web Server / Apache Tomcat:

I recommend that you use version 10 of Tomcat. You can donload it at:
https://tomcat.apache.org/download-10.cgi

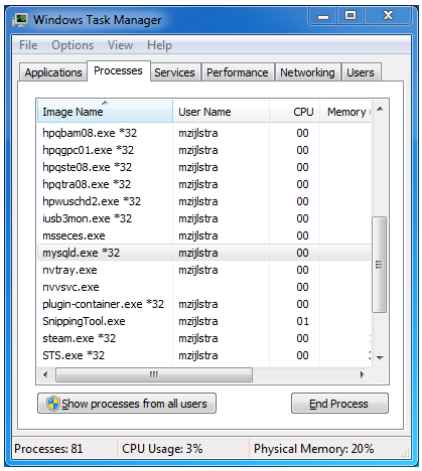


You'll most likely need the 64bit windows zip. Important: when you extract these files they need to be in a path without spaces in it! In other words something like C:\tomcat\ is okay but C:\Program Files\tomcat\ is **not okay** (because of the space between program and files).

W1D1 Exercises

Course Overview 1 – Does it run?

The Database:



The main objective of this exercise is to make sure that your development environment is working.

You can check to see if the MySQL server is running by looking in the task manager's process list for mysqld.exe. If it's not running then something went wrong with the MySQL installation process.

You can then connect with the MySQL Command Line Client (should be in your start menu after installation). It opens a command line window that right away asks for your password (the root password you made earlier).

Once connected enter the following SQL command to create the database for this course:
`create database cs544;`

The Exercise:

Download the W1D1-Course_Overview_1.zip file from the Sakia assignment page and extract it to: **`C:\CS544\exercises\`**.

Go to src/main/resources/application.properties and change the database password.

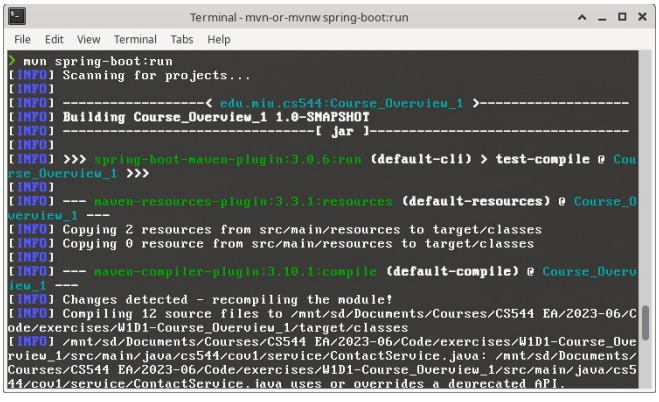
I will first describe how to run the exercise using only Maven (no IDE) and will then show the steps (with screenshots) on how to do it with both IntelliJ and Visual Studio Code.

Maven (no IDE):

Open a terminal / command line window (cmd.exe, or inside your IDE) and change your directory to **`C:\CS544\exercises\W1D1-Course_Overview_1\`** .

Then enter the command: **`mvn spring-boot:run`**

This will create a lot of output that looks something like this.

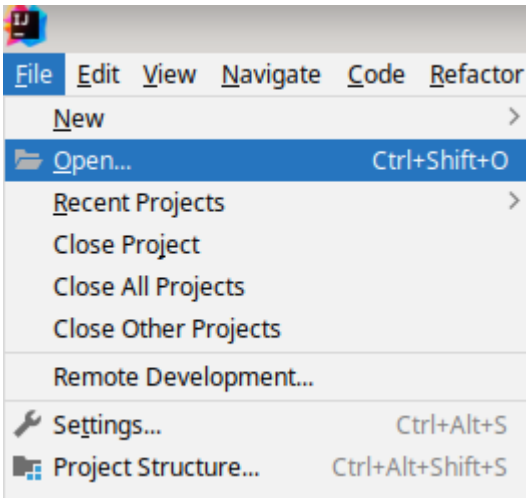


With the latest version of spring boot it's also giving errors about the cs544.email and cs544.phone tables not existing yet. But these are not fatal errors (everything still works).

If you followed these instructions (instead of IntelliJ or Visual Studio Code) you can go to the "Checking the Application" section down below to test the application itself.

IntelliJ

Open the project in IntelliJ with **File → Open** and select the exercise folder.



It may ask if you want to import it as a Maven project, although I believe it auto-detects it.

It will definitely auto-detect that it is a Spring Boot project, and where the Spring boot executable class is. You can simply click the run button to have it start.

Once it's running go to "Checking the Application" below.

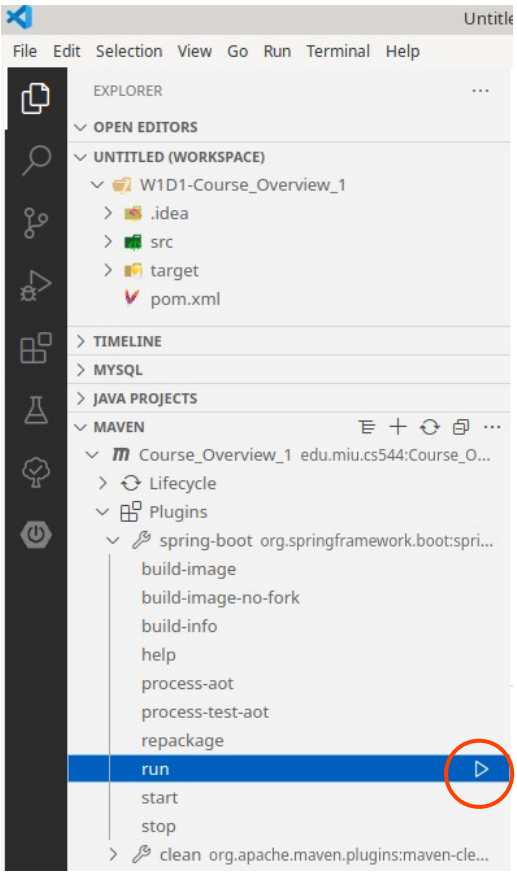
Visual Studio Code

Be sure to install at least the **Maven for Java** extension (by microsoft) I would actually recommend all of the following extensions for this course:

- **Extension Pack for Java (by Microsoft)**
- **Spring Boot Extension Pack (by VMware)**
- **Community Server Connectors (by Redhat, to start/stopTomcat)**
- **XML (by Redhat)**
- **MySQL (by Jun Han)**

In the menu go to **File → Add Folder to Workspace** and then select the exercise folder.

Then click on the run button in the **Maven section → Course_Overview_1→ Plugins→ spring-boot→ run**

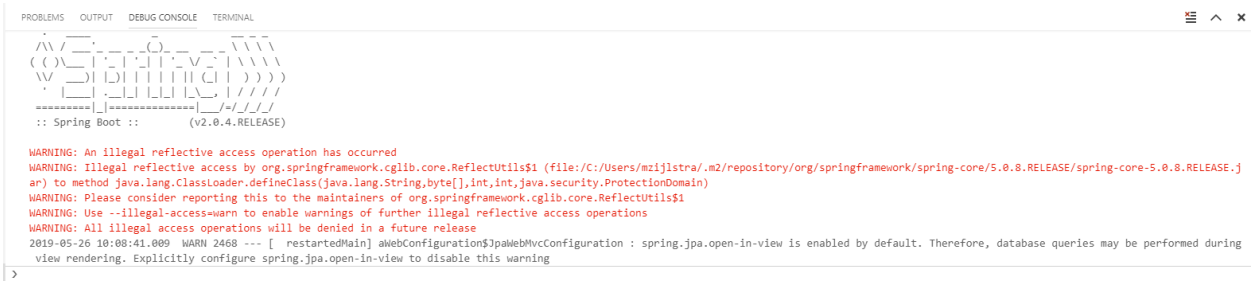


Checking the Application:

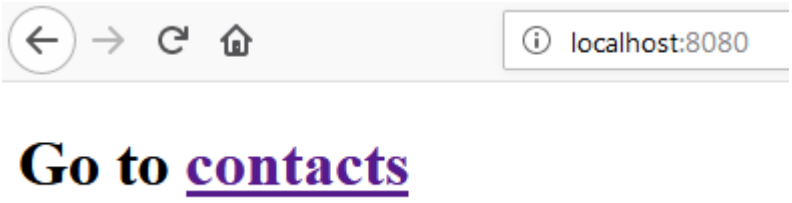
Most Spring Boot projects (including this one) have an embedded Tomcat web-server which should startup on port 8080 when the project starts.

It will connect with the CS544 database that we previously created.

Once the project has started you should see output something along the lines of what is shown in the screenshot below. Again note that the latest version also gives some SQL exceptions.



Now that it has started you can use your browser to connect to its webserver at: http://localhost:8080/ This should look something like:



Clicking on the link should take you to a login page, where you can use the username: **admin** and password: **admin**

Once logged in it will show a list of contacts (empty) and the possibility of adding a contact. Add a contact **Test 123** and clicking on the **Add** button

Once the contact has been added you should see it on the web page, and we should also be able to see it in the database.

On the MySQL console execute:

```
USE CS544;
SELECT * FROM contact;
```

It should have the contact you added in the database.

When you're done, you can write the first part of your report on Sakai. You can even submit it already as you have unlimited re-submits.

Also write this report if you are unable to make it work, the most important part is that I know what's going on.

Please write in the textbox on Sakai (do not write a separate file that you attach)

How to Submit:

Please write a brief report on Sakai about your experience with the exercise. Please use the textbox on the site (do not provide it as an attachment). Your report should describe how long it took, and what kind of problems you faced (if any). For example:

> Hi Professor, the first assignment it took me about half an hour, I accidentally forgot to change the database password, but I figured it out after reading the errors.

You can already submit this part of your report (even though there is still another exercise) as you have unlimited re-submits.

See the next page for the second exercise for today.

Course Overview 2 – Does it run?

Purpose:

In this exercise we're going to run almost the same application as the previous exercise, but instead of using Spring-Boot we'll use a pure Spring/Hibernate application on an external Tomcat web server (non-embedded).

Be sure to shut down the Spring Boot application from the previous exercise. If you forget to stop it port 8080 will still be in use, which the external Tomcat server also needs.

The Exercise (no IDE):

Most of the following steps can also be done inside your IDE, but it's good to go through the process manually once as well to better understand what's going on.

Update the database password inside the java configuration inside the project at:

`src\main\java\cs544\cov2\config\Config.java`

Download the code for this exercise from the Sakai Assignment, and extract it to your exercises folder (C:\CS544\exercises\).

Open the terminal and go to **`C:\CS544\exercises\W1D1-Course_Overview_2\`**

Execute the following command: **`mvn package`**

This will create a .war file inside the target directory. Web Archive (.war) files are essentially just zip files containing our compiled project and a deployment descriptor telling it how to run on a servlet container (Tomcat).

To make things simpler rename the created file:

`C:\CS544\exercises\W1D1-Course_Overview_2\target\Course_Overview_2-1.0-SNAPSHOT.war`

To:

`C:\CS544\exercises\W1D1-Course_Overview_2\target\COV2.war`

Next let's start the Tomcat server by running:

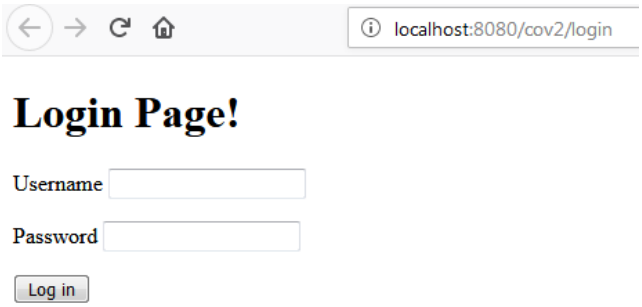
`C:\tomcat\apache-tomcat-10\bin\startup.bat`

Going to http://localhost:8080 should show you a Tomcat status page.

Then copy the COV2.WAR file to: **`C:\tomcat\apache-tomcat-10\webapps\`**

Once it's copied Tomcat should automatically detect it, unzip it and make the application available at: http://localhost:8080/COV2/

Note that this is almost exactly the same link as the previous project, the big difference is that it's now running on a separate Tomcat server, and is a project / subdirectory on that server (which adds the COV2 at the end).



Once again you can use username **admin** and password **admin** to login. Then add a Contact and check the database to see that it has been properly inserted.

Once you've confirmed that everything is working you can shut down Tomcat by executing:

```
C:\tomcat\apache-tomcat-10\bin\shutdown.bat
```

How to Submit:

Update your report on Sakai about your experience with the exercise. Again telling me how long it took, and what kind of problems you faced (if any). For example:

The second assignment also took about half an hour, I didn't really face any problems.

Thanks!