# Image result for epita

# Identity Access Management

**Submitted By**: Sonali Narayan Ghadage

**Course:** Data Science & Analytics

**Spring 2018**

Table of contents:

|  |  |  |
| --- | --- | --- |
| Sr. no. | Index | Page No. |
| 1. | Subject Description | 3-4 |
| 2. | Subject Analysis | 4-6 |
| * Major Features: |
| * Create |
| * Insert |
| * Update |
| * Search |
| * Delete |
| * GUI (Graphical User Interface): |
| * Application feasibility |
| * Data description |
| * Expected results |
| * Scope of the application |
|  |
| 3. | Conception | 6-9 |
| * Chosen algorithms |
| * Data structures |
| * Global application flow |
| * Global schema or database |
|  |
| 4. | Configuration Instructions | 9-14 |
| 5. | Commented Screenshots | 14-21 |
| 6. | Bibliography | 21 |

Subject Description:

The Identity Management System is GUI based application which allows end user to manage and manipulate user identities. User can create, update, delete and search identities. This application is coded on POM (Project Object Model) architecture. It is an XML representation of a Maven project held in a file named pom.xml. Application runs on Tomcat server and uses MySQL database to store data.

Subject Analysis:

* Major Features:

**Create:**

This allows user to create identity using unique user\_id, name and email\_id.

**Insert:**

This allows user to insert identities randomly using any unique user\_id which is inserted into the table in an ascending order.

**Update:**

This allows user to edit or update name and email\_id of each existing unique identity.

**Delete:**

This allows user to delete existing identity.

**Search:**

This allows user to search existing identity using unique user\_id.

**Graphical User Interface:**

GUI provides user interface to the user to interact with the system user friendly and efficiently.

* Application Feasibility:

It’s a web application which has been made up of Java servlets; this gives web interface to the user. Due to web based application, cost of installation is low. This also helps to maintain and version the software at centralized point. Web applications are easier to extend than standalone PC application. This application uses structured database to store data, which is feasible to manage and restore the data. A project contains configuration files, the defect tracking system, the organization, the URL of where the project lives, the project's dependencies, and all of the other little pieces that come into play to give code life.

It is a one-stop-shop for all things concerning the project and that is nothing but project object model (POM). The POM contains all necessary information about a project, as well as configurations of plug-ins to be used during the build process. It is, effectively, the declarative manifestation of the "who", "what", and "where", while the build lifecycle is the "when" and "how".

* Data Description:

First of all, while providing authentication it takes username and password from the table located in identities database where user’s username and password is stored. Basically, this application gives variety of options to manipulate data such as create, update, delete and search. These actions are performed on three attributes, Name, Email and Unique Id. Unique Id is unique in overall application lifecycle, so no duplication of id throughout the application lifecycle.

While creation of identity, system asks three fields which are Id, Name and Email as when we are giving Id; it becomes unique and can't be duplicate. At the time of submission, system also validates that user\_id, System doesn’t allow user to create duplicate identity. There is no limitation on Name field; user can enter any alphanumeric value in this field. This is also applicable while updating the user information. Update operation works on the basis of unique id. System matches unique id to map data with database. However Search operation can search any identity based on its unique identity. To store data, system is using structured database, using MySQL.

* Expected Results:

As developed web application is to manage user identities in an organizational manner. So the expected output would be creation, deletion, updating, and searching of those unique identities stored in the database. Moreover GUI helps to manage those identities efficiently.

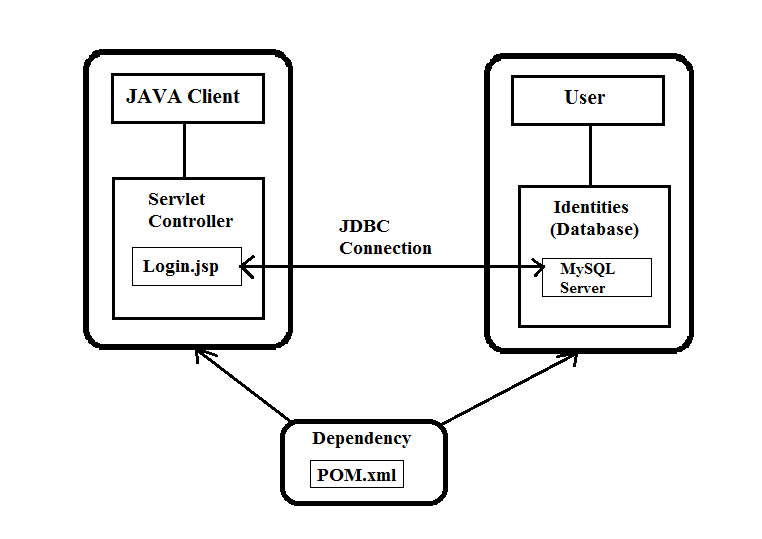
* Scope of the Application:

This application fulfills basic operation of managing data. Right now scope of the application is limited to create, update, delete and search operation. In future, system will filter user's data. Current system has only one credential access, which can be progresses in future development to many accesses. System doesn’t provide any admin and normal user for access control. In future development, users can emerge into two types such as admin and normal user.

Admin user would have full control on the system, whereas normal user would have limited control on data and system. Future system will also have feature to export data to Spreadsheet, and imported into system from spreadsheets. In future development, identities will have more attributes, which can be address, phone number, zip code, access control etc.

Conception:

The web application uses maven architecture which is based on POM. To manage and better organization of code and result business logic, UI and database have been kept separately this structure will allows easy modifications with respect to future scope.



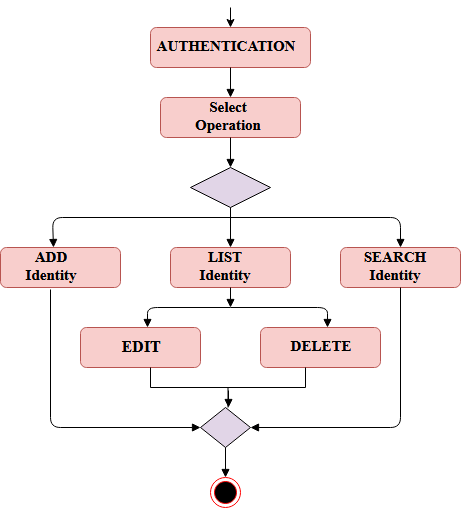
* Chosen Algorithm:

This web application is using no particular algorithm as such, but to store data MySql 5.1.44 is used. Based on the primary key user\_id, update, delete and search operations has been performed. So, this mapping is handled by unique id of identity.

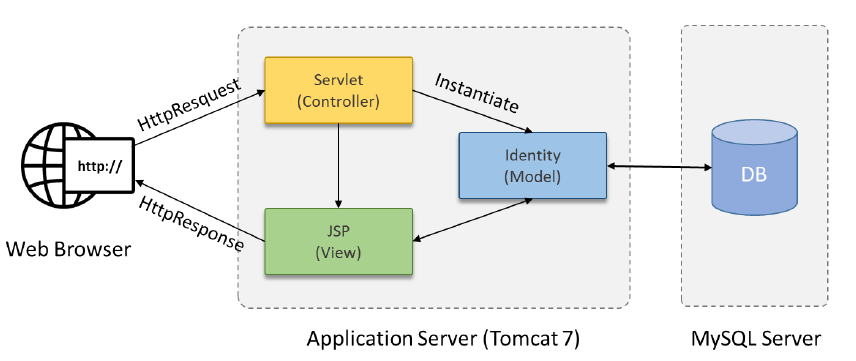
* Data Structures:

This application uses MySQL database to store data. As for development setup, system is using MySQL workbench and XAMPP Server to run MySQL service. SQL is providing structured way of data storage, therefore manipulation of data becomes easier for user. List and array data structures are also used to do operations on that data in java classes.

* Global Application Flow:



* Global Schema:



Configuration Instructions:

This application needs several configurations. Following are the requirements of the project.

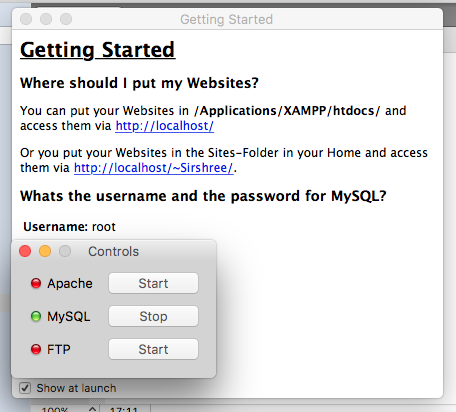
* 1. XAMPPServer (for MySQL)
  2. MySQL Workbench
  3. Maven configuration
  4. Apache Tomcat Server 8.0
  5. Setup environment variable
* XAMPPServer (for MySQL):

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.

XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well. In this web application MySQL is using XAMPP server for storing database.

This is the website (**https://www.apachefriends.org/download.html)** to download XAMPP server according to your computer configurations.

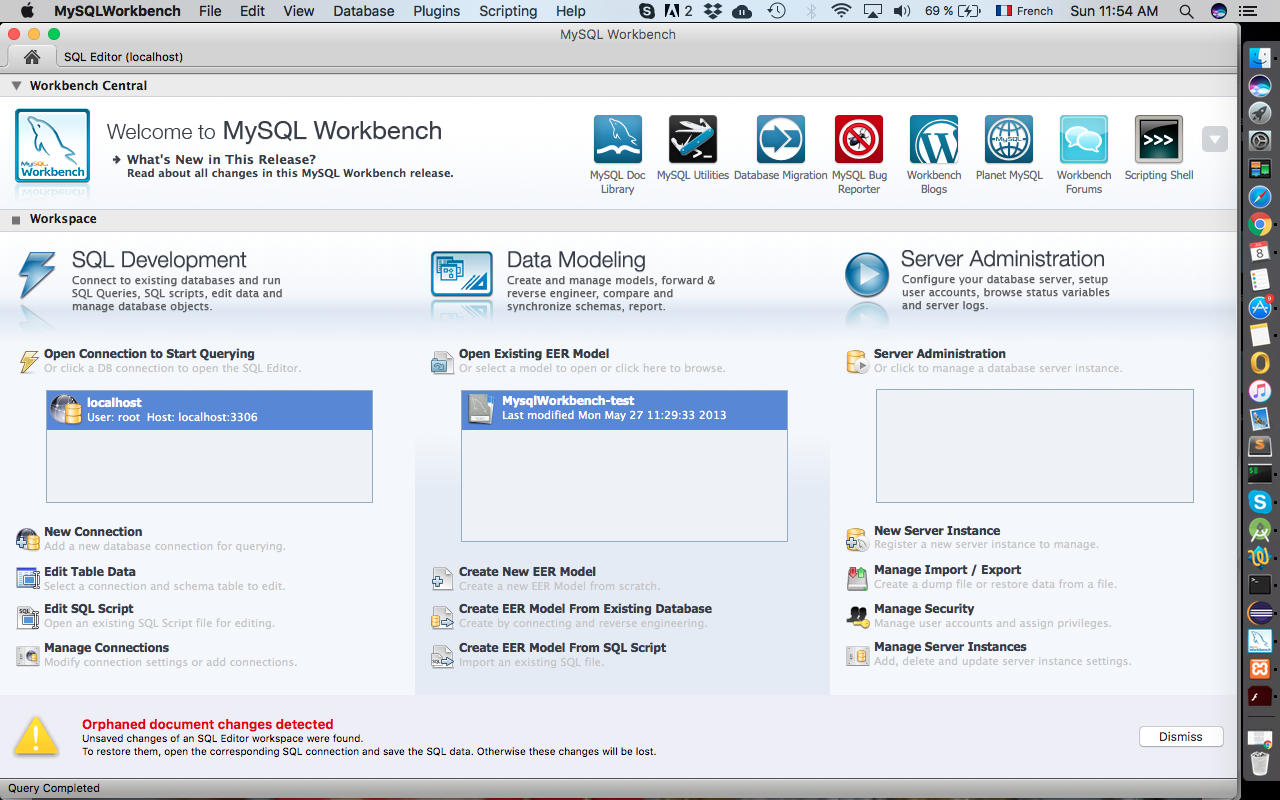
It will look like as following after installation.



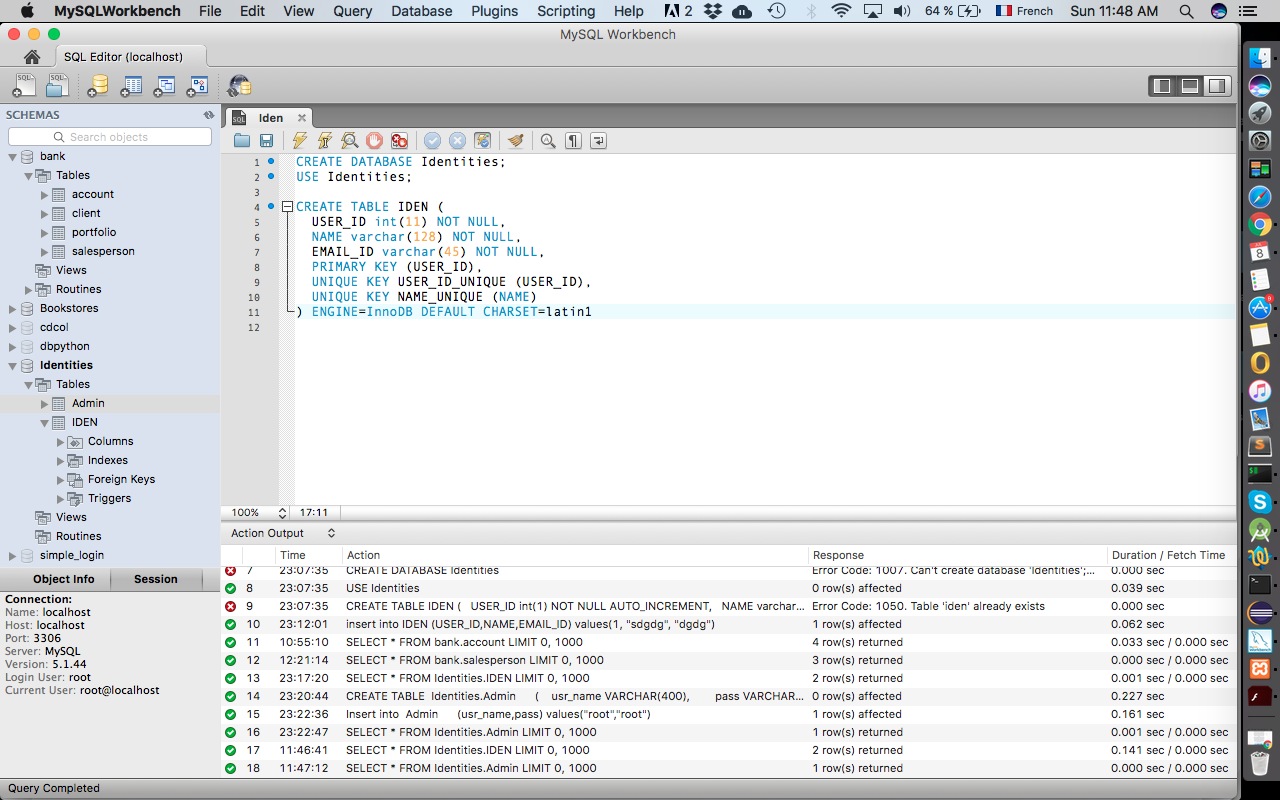
* MySQL Workbench:

XAMPP Server is controlled by MySQL Workbench. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more.

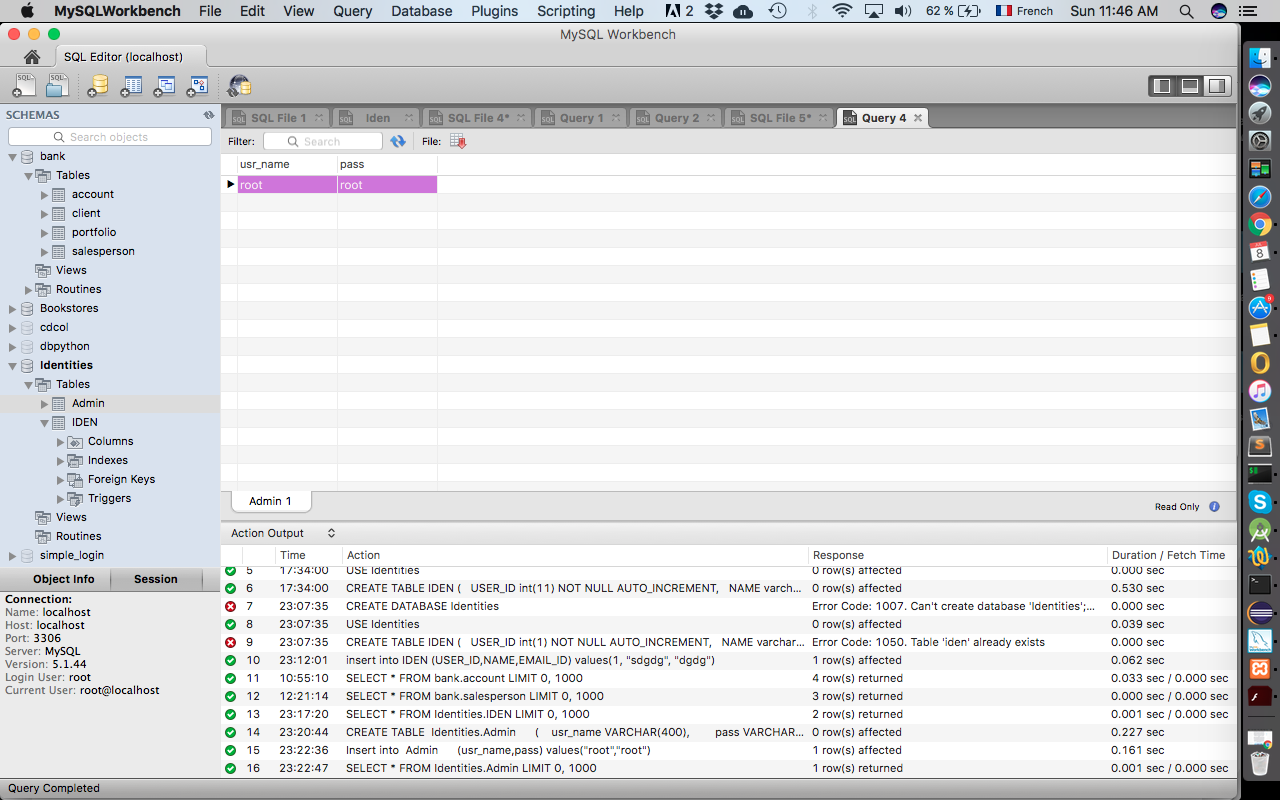
MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. You can download it from this website (<https://www.mysql.com/products/workbench/>)



Here XAMPP server is connected to the “localhost” having root “user”. After clicking on that following page will open where we can create a database.



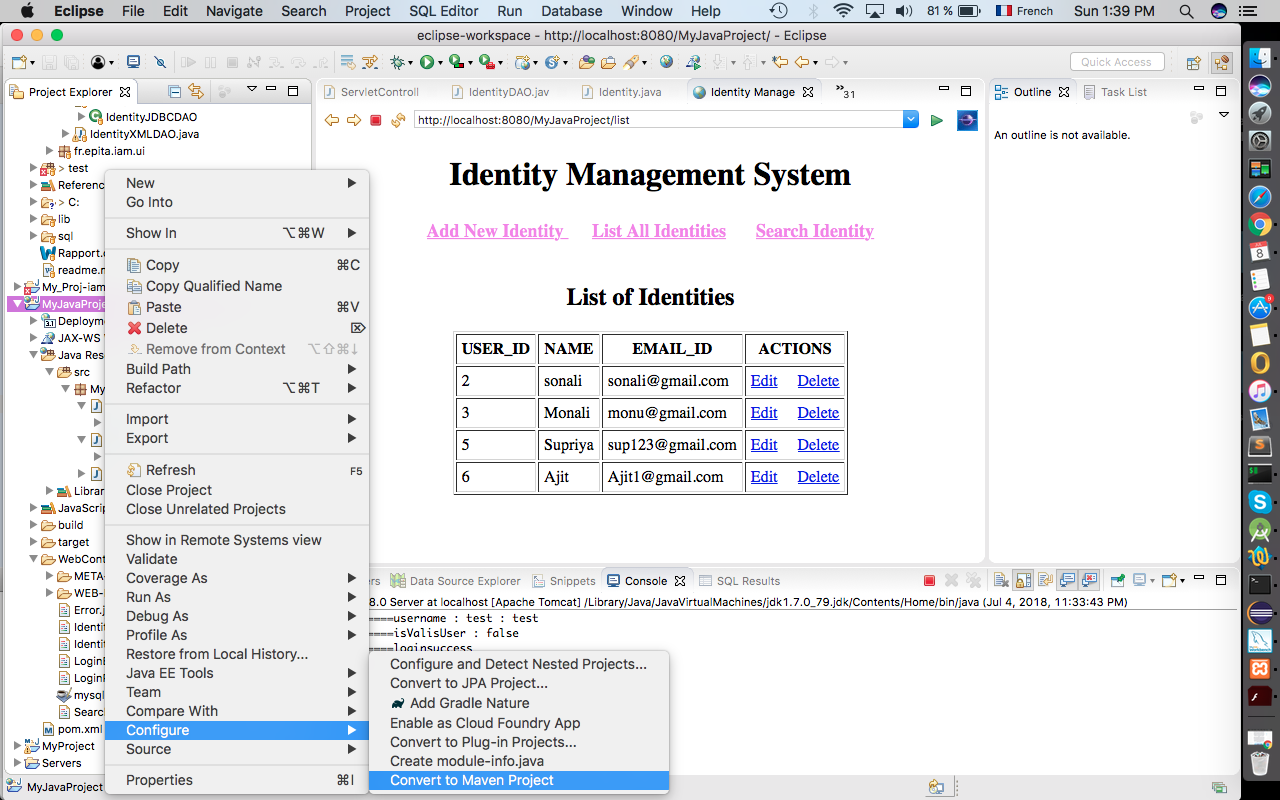
On the same database Admin table is also added to store username and password “root”.



* Maven configuration:

Then convert this project to a Maven project by right click on the project, select Configure Convert to Maven Project, as shown below.

The needed information create Maven POM file, such as group ID, artifact ID, etc. Then it will add the dependencies. You will find it in WEB CONTENT->pom.xml file.



* Apache Tomcat Server 8.0:

Apache tomcat server is used to host java application to make servlet and JSP runable.

Go to http://tomcat.apache.org/index.html and download Tomcat version 8.

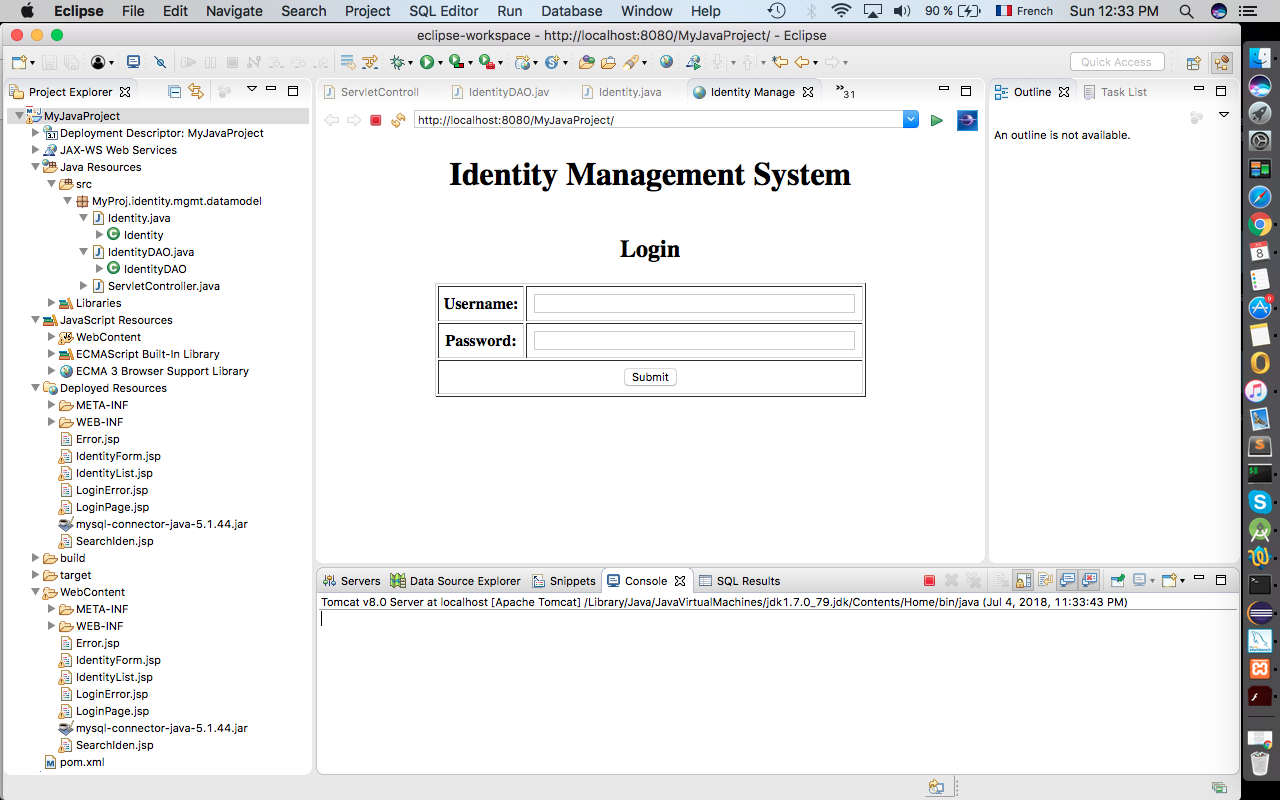
Install the executable file.

Open project in eclipse and right click on project. You will see menu, go to run as and select run on server.

There will be run on server window, Select Tomcat v8.0 Server. If you want to change server name, then you can do it by inputting new name in Server Name textbox. Now, click on Next.

There you will find new dialog, having two columns called Available and Configured. Select the project from Available and click on Add button. This will add project to Configured column.

Now, click on finish. Project has been published to server and ready to use. Put (http://localhost:8080/MyJavaProject/) login URL into your browser. You will see following screen.



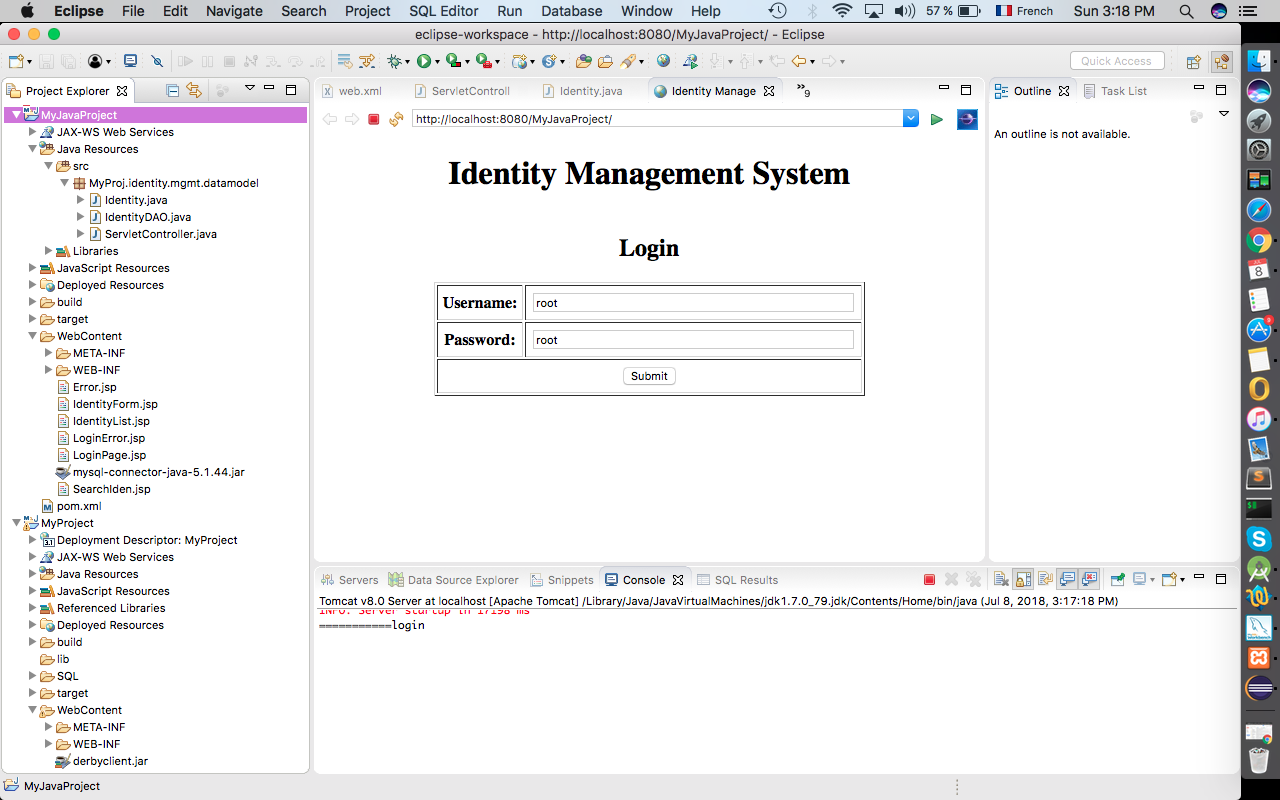
* Setup Environment Variable:

You will find this file in project at /MyJavaProject/WebContent/WEB-INF/web.xml

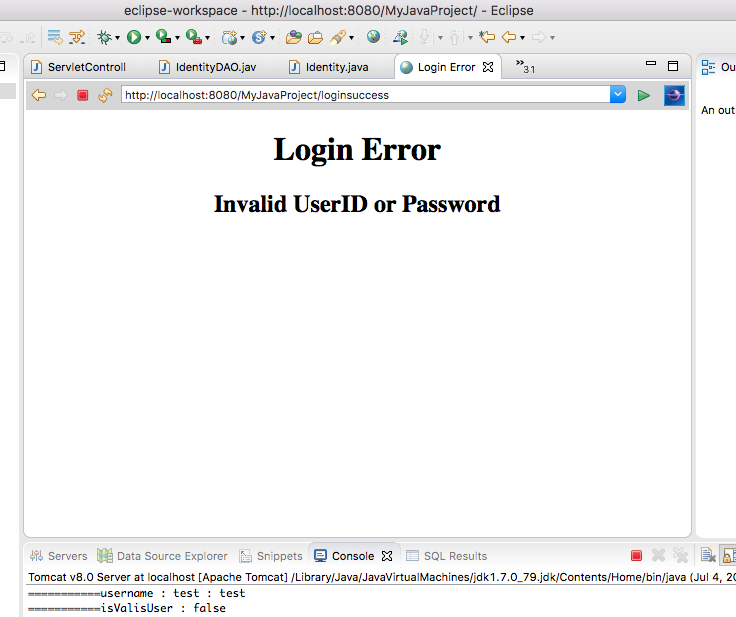
In web.xml you will find the environment variable of the whole project named as “ServletController”.

Commented Screenshots:

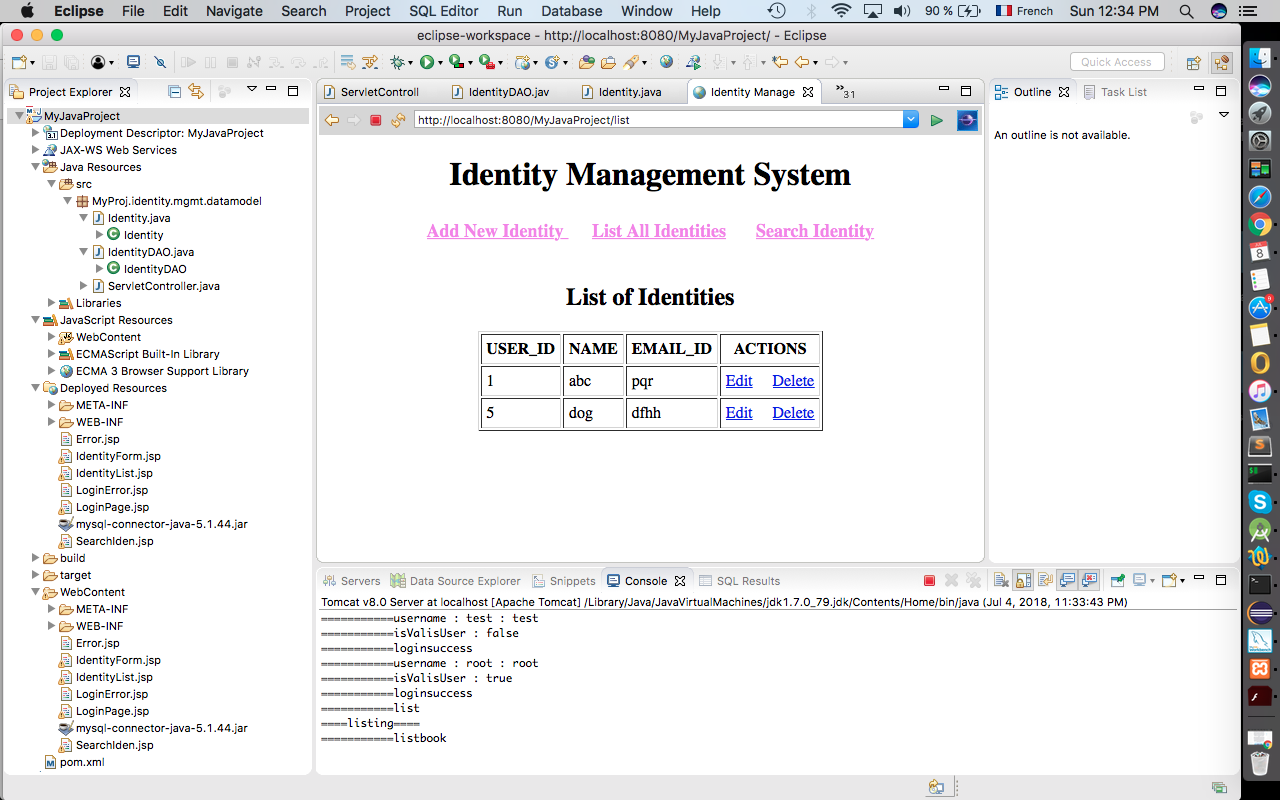
Following screenshots will show overall execution of this web project.



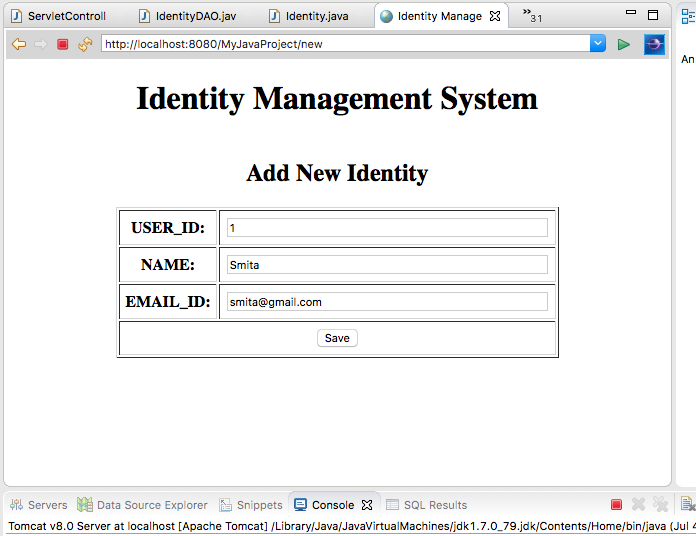
This is login page of web application. If the username or password is invalid then it will display following message.



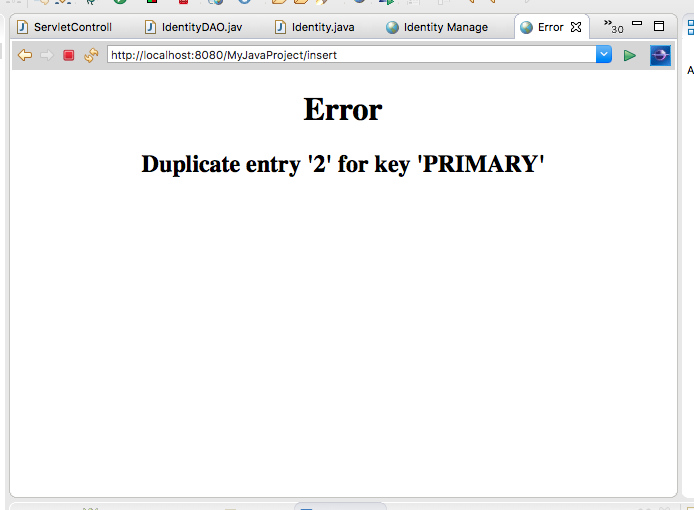
After successful login you will see the options like “Add new Identity”, “List of Identities”, “Search Identity” and you will see two hyperlinks named as “Edit” and “Delete”.



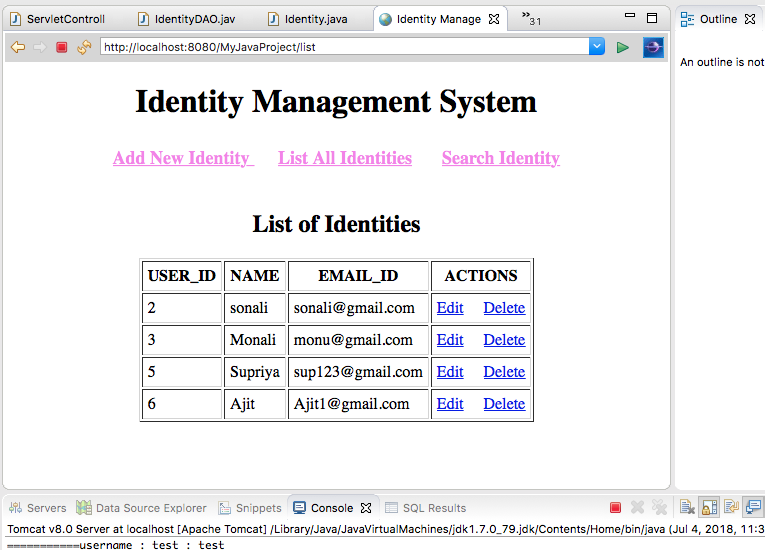
Now, after clicking on “Add New Identity” you will see following page to add new identity.



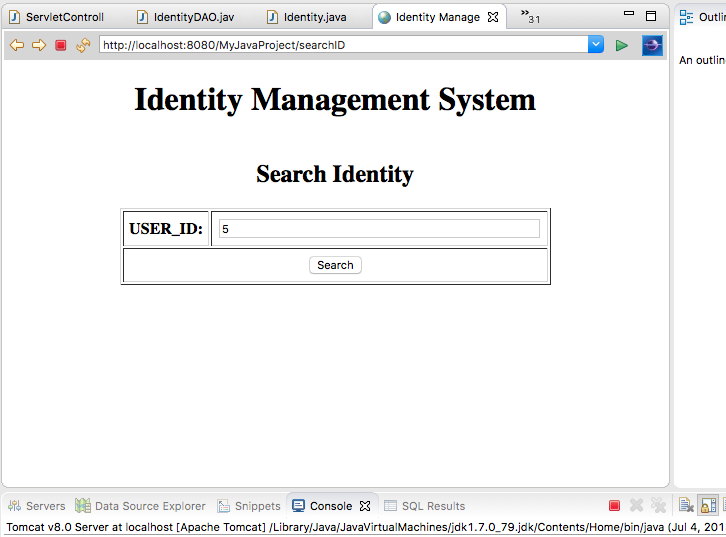
When you are adding the same identity of same user\_id then it will display the following error.



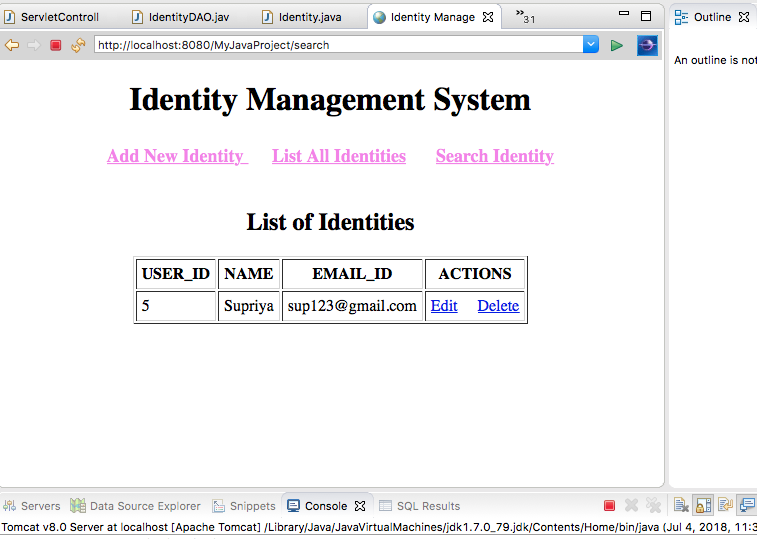
You can as many identities as you want. After adding multiple identities you will see the “List of Identities”.



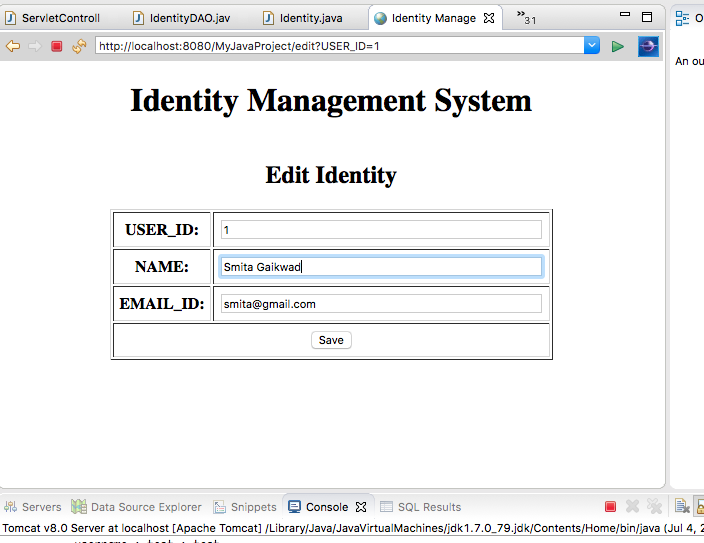
Now, by clicking on “Search Identity”, you can search any identity by giving user\_id to the application as shown in the following image.



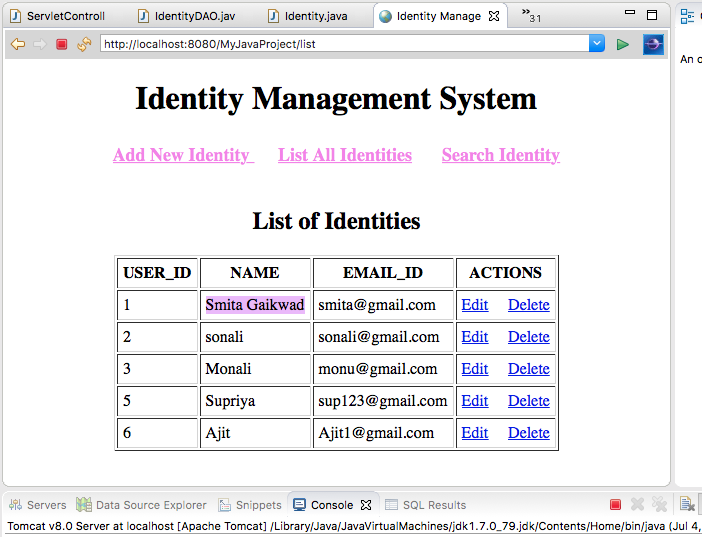
As given in the search identity the 5th identity will display on the screen.



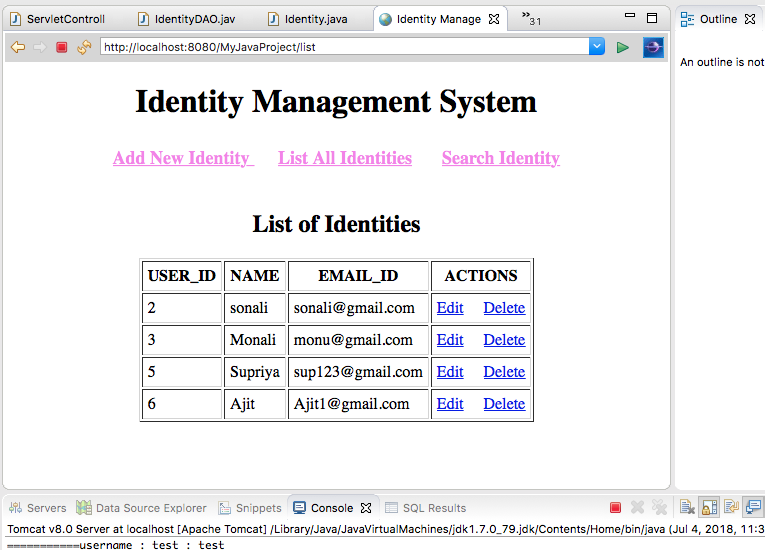
After search operation by clicking on “Edit” hyperlink, you can edit particular identity. In the following image I am editing the first identity.



Those modifications will reflect in the list.



Now, after clicking on “Delete” hyperlink, the identity will get delete. Here in the following image I am deleting first identity, and then the result will be, as shown in the following image.



Bibliography:

* [www.google.com](http://www.google.com)
* <https://www.mysql.com/products/workbench/>
* [**https://www.apachefriends.org/download.html**](https://www.apachefriends.org/download.html)
* <https://stackoverflow.com/>
* <http://tomcat.apache.org/>