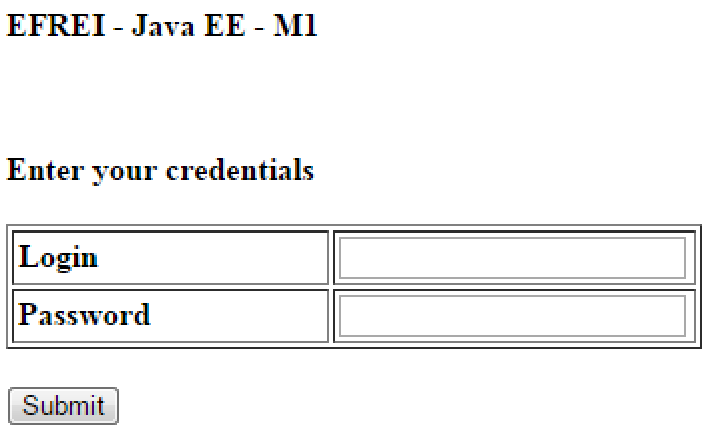
EFREI – M1 – Java EE Project

Goal: Implement a **Club directory.**

Welcome page

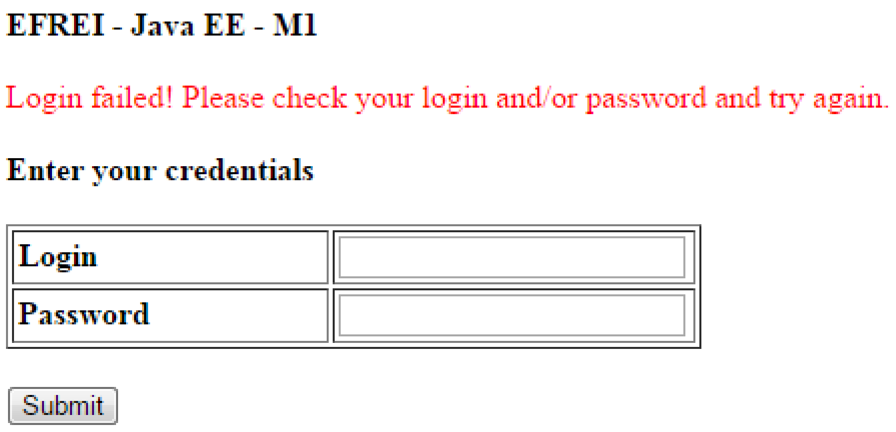

EFREI - Java EE - M1

Enter your credentials

Login	
Password	

Login : **admin**
Mot de passe : **efreijee**

Connection failed message

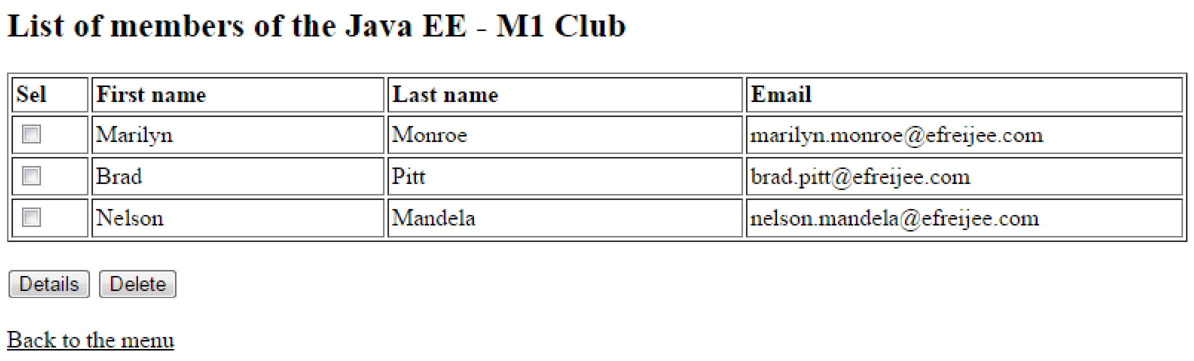

EFREI - Java EE - M1

Login failed! Please check your login and/or password and try again.

Enter your credentials

Login	
Password	

Members list screen



List of members of the Java EE - M1 Club

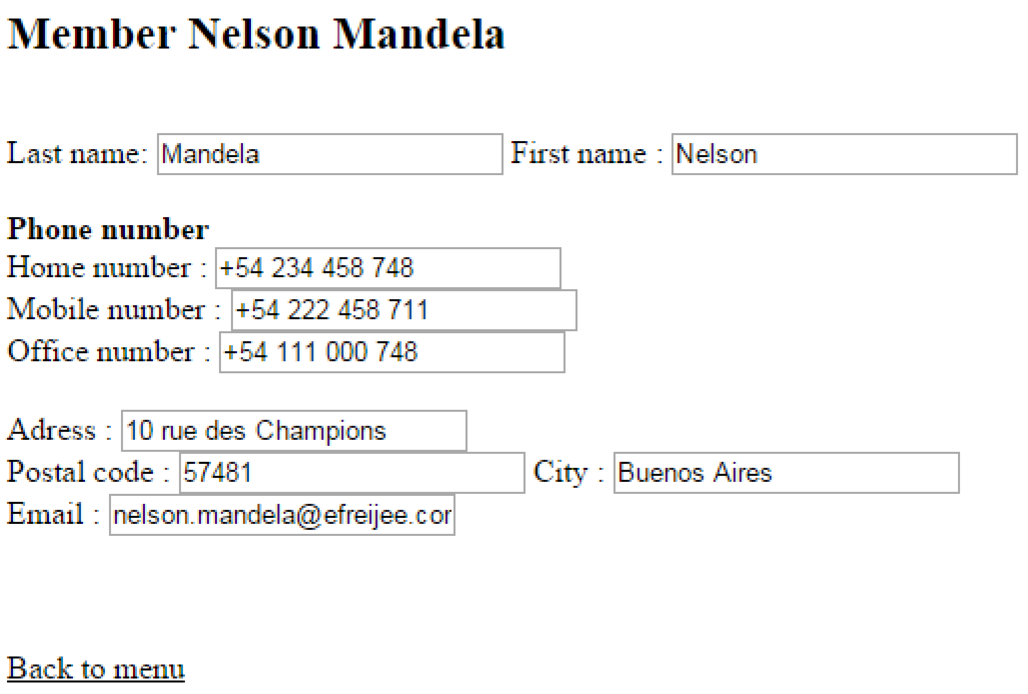
Sel	First name	Last name	Email
☐	Marilyn	Monroe	marilyn.monroe@efreijee.com
☐	Brad	Pitt	brad.pitt@efreijee.com
☐	Nelson	Mandela	nelson.mandela@efreijee.com

1. On this screen if you delete all the members the following message is displayed (in blue & bold):

   The Club has no member!

2. And in that case (empty directory), provide a link "***Add new members***" that will automatically refill the base with 5 brand new members!

Screen Member Details



Member Nelson Mandela

Last name: Mandela First name : Nelson

Phone number
Home number : +54 234 458 748
Mobile number : +54 222 458 711
Office number : +54 111 000 748

Adress : 10 rue des Champions
Postal code : 57481 City : Buenos Aires
Email : nelson.mandela@efreijee.cor

Instructions

General

- ✓ All String constants have to be declared at the beginning of the class and this way :
  `private static final String MSG_ERR_LOGIN_PWD = "Invalid credentials ";`
- ✓ No public attribute!
- ✓ DBMS to use : **Java DB** (Derby). Please add at least 4 rows in you database.
- ✓ Your code has to be well documented.
- ✓ 3 versions = 3 NetBeans projects = 1.zip file.
- ✓ Provide a readme.txt file
- ✓ Provide a SQL script and put it in /WEB-INF.

a) Version 1

1. No servlet. JSP only.
2. Java code in JSPs
3. Credentials are stored in a database.
4. The entry point of the application is ***index.jsp***
5. You may use a Java Bean ***BeanMember.java***

b) Version 2

1. Add a servlet ***Controller.java*** in a package named ***m1.jee.ctrl***
2. Replace as many scriptlets as possible with EL and JSTL syntax.
3. Create a class DBConnection.java that will contain code to connect to the database.
4. Create a class DBDisconnect.java that will contain code to disconnect from the DB and release all related resources.
5. Credentials are stored in the context.
6. Error messages are managed using a Map
7. The entry point of the application is ***Controller.java***

c) Version 3

1. Persistence will be managed with EJBs and JPA.
2. You will use a Web service to retrieve the credentials.

More instructions will be eventually given later.