**Stocks**

**Steps to begin :**

1. Install latest Netbeans with the latest SDK available on : <https://netbeans.org> .
2. Install mamp the substitute of wamp for Mac (including mySql Server and phpMyAdmin).
3. Create the database (Expenses) from <http://localhost:8888/MAMP/index.php?page=phpmyadmin&language=English> and the tables Users , Operations , Categories.
4. Create a new project in netbeans (Categories : Java Web , Project : Web Application) choosing GlassFish Server 4.1 as server , Java EE 7 Web as Java EE Version and JavaServer Faces as Framework .
5. Select PrimeFaces as JSF component suites to set up for your project .
6. In the Services menu in netbeans we need to create a new database connection to create a connection between our project and the database already created (Username : root , Password : root (default config) , port : 8889 (mySql port on mac) , host : localhost) .
7. Add a new server : GlassFish in netbeans and run it to start developing your website

**Coding :**

We need first to create entity classes from database , when you clicked this option you need to create a new datasource based on the connection created on step 6 and write any JNDI name desired. When this datasource is selected all the tables created in phpMyAdmin will be shown in netbeans so the user would select the tables needed to complete his website then on the next button you should enter a package name to finish this point.

Then based on the created entity classes you should create JSF Pages from entity classes . Then create the login page and login bean .

To user the j security check some modification must be done in the GlassFish administration console as following :

* The JDBC resources could be created from the console or from netbeans as we did and the JDBC connection pool also .
* When you open the GlassFish administration console on the left menu ,expand the configuration point then expand the server\_config and then the security point and click on the Realms.
* You need to create a new Realms ,write any Realm Name and use

com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm as a Class name and for the JAAS context you could type any name but keep it in mind , some fields are mandatory such as the JNDI which refers to the JDBC resource already created in the project it’s called jdbc/Expenses , User table , user name column and the password column as expected refer to the tables and column created and the group table and group name column in my case are the Users and the column “flag” in the database that indicates if the logged in user is a simple user or an admin.

* And also I used the algorithm AES to encrypt the data sent to the server and make the password unreadable for any sniffing.
* Back to the netbeans in the web.xml file ,security tab some modification must be made : As for the login configuration the form check button must be chosen and the login page and error page are entered to determine the welcome page and the page to redirect to in case of password error , the realm Name is the one created in the glass fish administration console.
* For the security roles a role must be created for each type of user (in this project 2 : admin, user) and the role name must be similar to the data in the Users table , flag column so that the code relate the two and differ between a user and an other .
* Now the final step in the web.xml is to create the Security constraints for the security roles . The url pattern in this step indicates the pages accessible by a specific user and not the other depending on the resource name or it’s flag in the database.
* In the glassfish-web.xml if not exist create it , in the security tab security roles should be added , but for most developers if these steps are followed these security roles will be created automatically when finishing the previous step.

I pushed my project on Github 3 times :

* The first time I created a simple login page , assign it as the startup page in web.xml attribute name welcome-file. I didn't use j security check the first time but instead I created a simple select from the users’ table to retrieve the password and compare it with the input password when the user click the login button. If the password in the DB match the typed password the user will be redirected to the operations page else an alert will be shown that the password is incorrect .
* The second time I implemented the j security check which provides more security and let the JSF pages verify on their on the username and password and redirect to faces/operations/List.xhtml if they match and to a alternative page or an error page if not.
* The last time I added the iReport by installing the required plugin from netbeans website and added some libraries (available in my Github project) to extract the report of the operations.

**Finally :**

This website contains an admin with flag : “1” that can control all the data in all the tables and users that can only add their operations and are unable to modify or create any fields that are related to others so at the end the admin can see all the operations done and print extract a report of the family or company expenses .

Note : Some modifications were made on the JSF pages created from entity classes to show the username instead of the userId and the operation name instead of the operation id (<h:outputText value=“#{item.categoryid.categoryName}”/> .categoryName returns the name not the id as the item.categoryid do)