# **Employee management Login and CRUD operations**

# **Table of Contents**

Create an Employee management system which allows to do following operations:	2		
System & technology used:	2		
Below are brief implementation details:  Business logic	2		
		Screenshots:	4

# **Employee management Login and CRUD operations**

# Create an Employee management system which allows to do following operations:

- 1. Login Employee.
- 2. Add more Employee.
- 3. Edit Employee details.
- 4. Delete Employee records.
- 5. Validations.

## System & technology used:

- 1. Operating System: Mac OSX High Sierra.
- 2. Eclipse Mars.
- 3. Tomcat 7
- 4. Java 8
- 5. JSF 2.x
- 6. Spring 4.x.x
- 7. Hibernate 4.x.x
- 8. Primefaces 5.x.x
- 9. JPA2
- 10. H2 database (in-memory) (also MySQL configurations available currently commented out)
- 11. pom.xml file contains all the dependencies (as it's a maven project)

## Below are brief implementation details:

- 1. To keep the task minimal, validations are kept minimal and no integration with other thirst party libraries are done; for example, spring security is not implemented, instead a simple Filter is used to check for loggedIn session attribute.
- 2. Also one Employee (not part of database record) is initially used as –

Username: admin Password: admin

Any other username-password combination will give error message for the first time. Once admin adds other employee, those can be used to log in too.

- 3. After logged in, User can add new Employee (minimal validations are done), edit records and delete. Duplicate checks are done. Password encrypted using md5 hash algorithm.
- 4. Once logged out, session will expire and any newly employee created can log in using their credentials.

# **Business** logic

For login, general user checks are done in bean level only. The architecture follows basic mvc pattern. Structure = Presentation Layer (JSF) → Service Layer(Spring) → Dao Layer (Hibernate)

CDI annotations are used

The service layer contains Transactional delegations via spring

The Dao layer does the persisting operation.

#### How to run:

- 1. On the root of the folder open terminal and type: mvn clean install (the project will build a employeemanagement .war file, incase maven in not available, a prebuild war file is attached)
- 2. Put the war file in tomcat7 or greater webapp folder.
- 3. Start tomcat server
- 4. Open Chrome → Type in the URL <a href="http://localhost:8080/employeemanagement/">http://localhost:8080/employeemanagement/</a>

Note:

The same demo is available online and is being committed into Github now:

Github: https://github.com/imdadareeph/employeemanagement

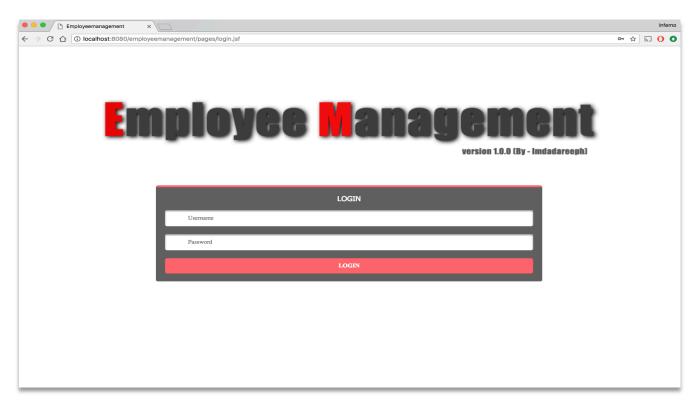
Live demo:

http://employeemanagement.j.layershift.co.uk/

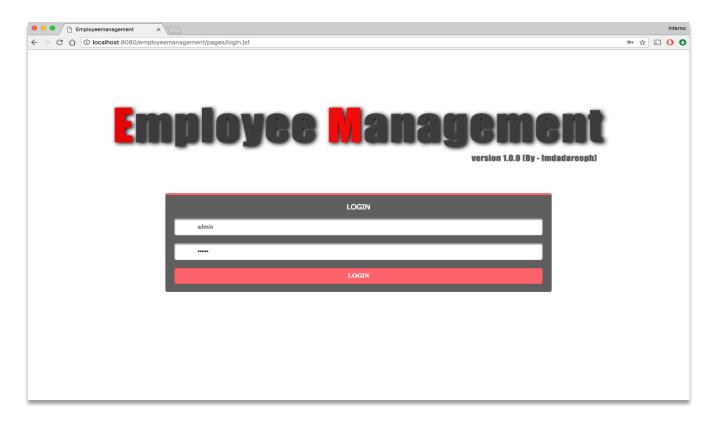
Screenshots are on next page.

### **Screenshots:**

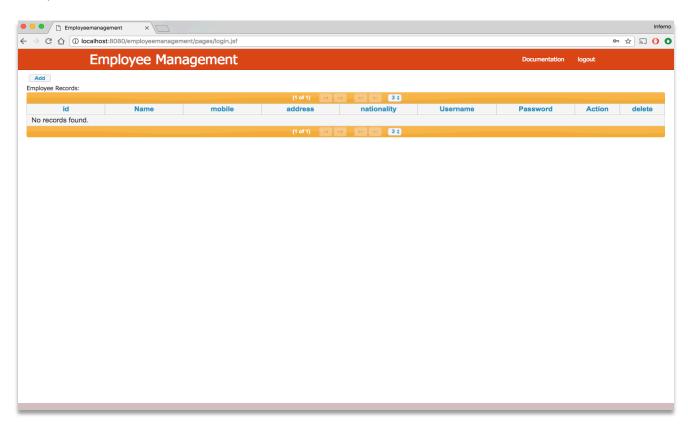
Open Chrome → Type in the URL <a href="http://localhost:8080/employeemanagement/">http://localhost:8080/employeemanagement/</a>



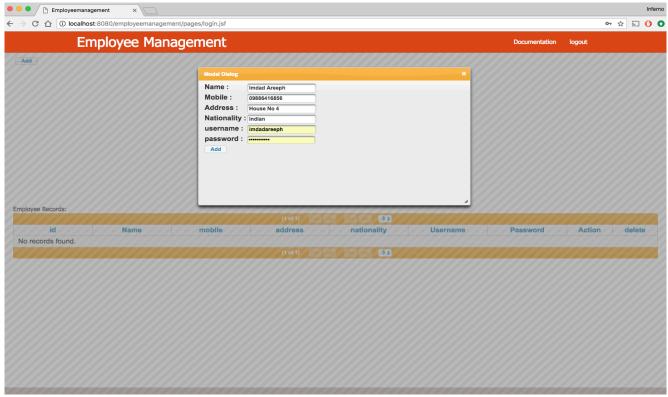
Give User credentials as User ID: admin and password: admin



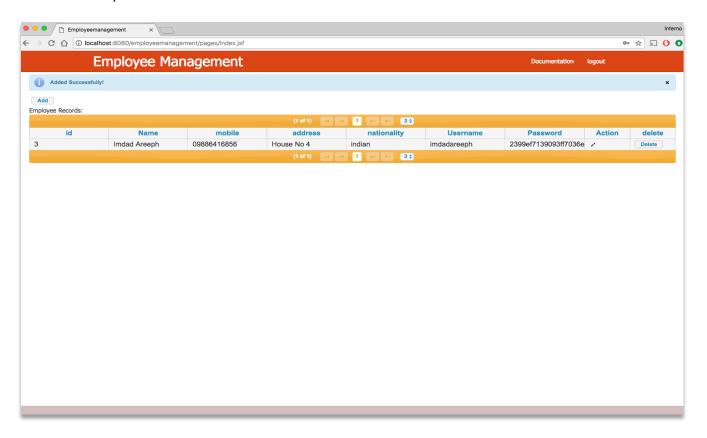
### Initially there will be no records

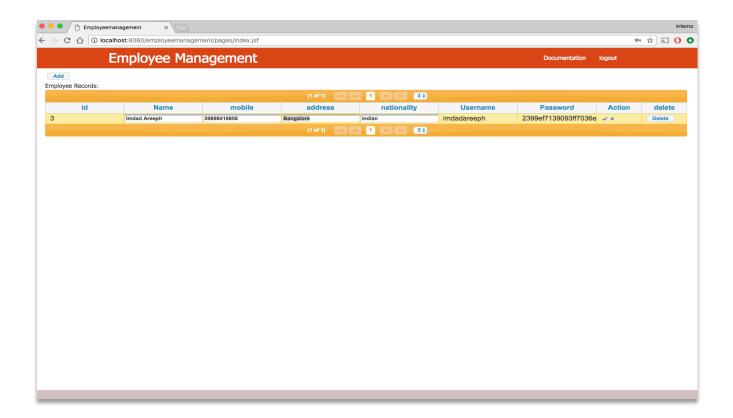


Add new Employee



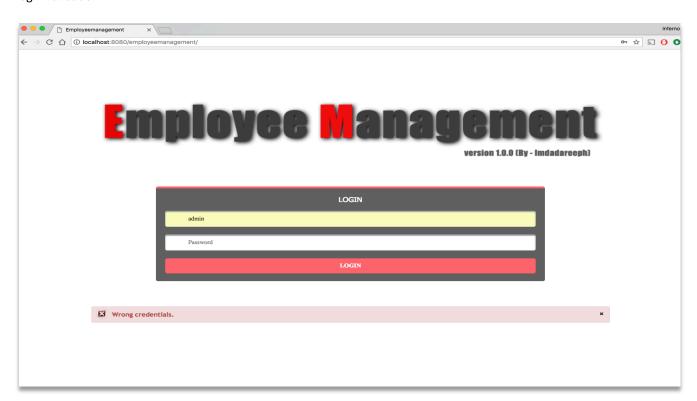
#### Added successfully





#### Validations:

#### Login validation



Edit validation

