# Java\_MVC project with MySQL :: Sakila.

1. **Project overview:** -

Java project with MySQL: Sakila contains the MVC architecture flow with all the defined DAO, Service Controller layers as per spring application standard. This project has all the CRUD features that allow users to:-

* 1. Add Customer
  2. Edit Customer
  3. Delete Customer
  4. Fetch/List Customer.

Project have all the OOPS concept with Inheritance, Data access layer, Configuration layer, Service layer (which server the basic CRUD functionality to controller) etc.

Project contains the build version tool as a Maven and utilizing the feature of latest build version 3. User can utilize all the features from UI (built on JSP containing JSTL and HTML code). All the features and project currently running on Tomcat 7 container (as a web server). As project structure built on Maven, so this project can be easily migrated to any of the other Web-server as per the client requirement.

Project utilizes the spring framework feature for Controller, Config and View binding as per enterprise standard. Spring is vast framework so spring people divide the whole spring in to different modules; they are designed in such a way that no module is dependent to other module, except spring core module.

Project has a feature of Entity relationship model based on Hibernate framework and the data binding controller with MySQL: Sakila database. The entire Config file, properties file needs to load from Web initializer, from where the scanning and project init work.

Features of CRUD are simply based on one controller – Customer Controller, and thus can be utilizing for more advance feature on Customer entity itself.

1. **OOPS concept cover :-**
   1. Abstraction – Featured in CustomerDaoIMPL class
   2. Inheritance – Featured in CustomerService layer
   3. Encapsulation – Featured in Customer Model class
   4. Polymorphism – Featured in CustomerController class
2. **Framework used in project-** 
   1. Hibernate framework – 2.1
   2. Spring 4.3 version

Feature enable with OOPS concept with the entire basic layer architecture standard. Like as Config,   
Controller, Service, JSP/View layer etc.

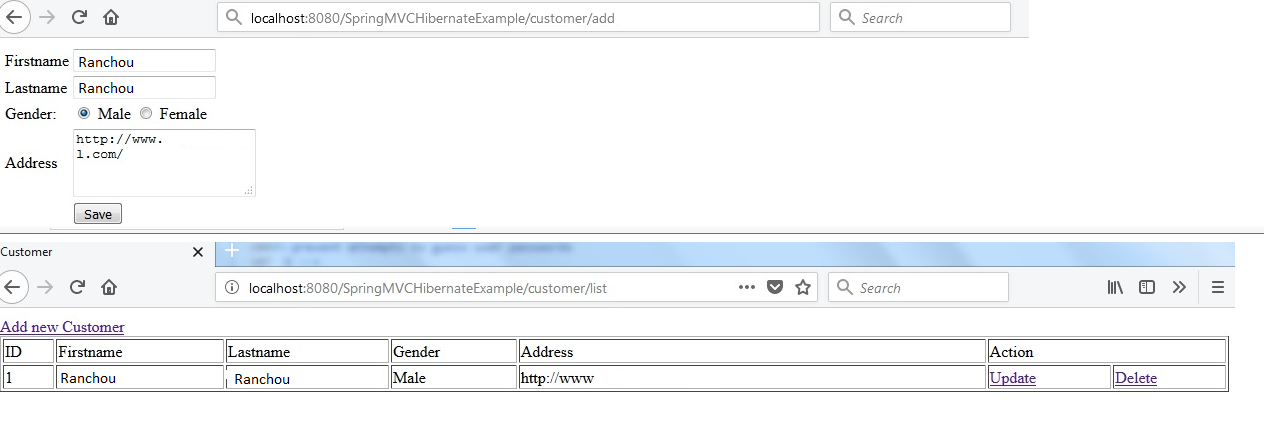
1. **Tools used in project**

Build tool – Maven version 3.1

JDK version – 1.8

Eclipse – IDE.

1. **Live project ruining on Tomcat 7 container – using localhost:8080**



**Thank You**