MVC SQL CRUD Rubric

Criteria	1 – Below	2 – Approaching	3 - Meeting	4 - Exceeding
HTML / JSP	• Syntactically incorrect HTML markup	• Page layout	• Content is contained within div's and is spaced and positioned using CSS	• HTML5 elements used where appropriate
CSS	• No CSS is present • Inline styles present • Internal stylesheets are used	• External CSS sheet is used, but very few modifications have been made to style the site	• Classes and ids are used to target style rules • CSS is normalized	• Attention has clearly been paid to organizing CSS and making it as reusable as possible
Spring	• The project is missing necessary dependencies • The spring framework is not configured correctly	• Spring dependencies are present in the pom.xml	• Web.xml and -servlet.xml files are configured properly	• A bean is used to create a DAO object on application startup
MVC	• The Model, View, Controller pattern is not implemented	• The application uses the MVC pattern to retrieve, manipulate and display data	• Database actions are encapsulated in a DAO • Classes map relational data to objects	• All data interchange between components is in the form of application objects
JSTL	• JSTL is not present, embedded Java Scriplets are used instead	• JSTL is used exclusively to write Response Parameter values	• JSTL is used to iterate through arrays of data, write values dynamically to html	• JSTL is used for conditionals and to track iteration through loops
SQL	• No SQL functionality implemented	• SELECT/INSERT/DELETE functionality are present	• Full CRUD was implemented	• Input is sanitized • Transactions are used and errors are rolled back.

JDBC	● JDBC is not configured correctly	● JDBC is configured	● SQL Exceptions are handled. ● All JDBC resources are closed appropriately	● JDBC objects and fields are properly scoped.
Aesthetic	● No attempt was made to create an attractive, modern aesthetic	● N/A	● Site has a consistent and deliberate approach to web design	● Interface is clean, intuitive, and modern
Object Oriented Programming	● Code is largely procedural ● Existing classes do not follow Object Oriented principles	● Classes exist which adhere to the Single Responsibility principle ● Extraneous specialty methods are implemented	● Object-Relational Mapping (ORM) is used to represent database records as objects. ● Getters and setters are implemented as needed.	● Polymorphism is utilized to represent like objects ● All getters and setters are implemented. ● Multiple constructors, including no-arg, are implemented.