**DOJO LEARNINGS**

|  |  |
| --- | --- |
| Before DOJO - GSS | After DOJO- GSS |
| 1. Manually Build and Dependency Injection process. | 1. MAVEN build tool is Integrated with GSS.   **Uses:**   1. Build Automation 2. Automation in dependency injection of Libraries. 3. Provided facility to add Test Cases – JUNIT |
| 1. No Formatted Design. Using legacy approach to develop product. | 1. MVC Design pattern introduced using SPRING BOOT.   **Uses: -**   1. Modularized the code in between layers. 2. No dependency between the classes and objects. 3. Redundancy of code is reduced. 4. Has features like cache mechanism, Connection pooling, and Data Template pattern which reduce the lot of effects like garbage collection handling, exception handling..etc. |
| 1. No ways to improve Performance Java code | 1. Performance Tips: 2. Connection Pooling (which can help to manage dB connections). 3. Cache mechanism (which helps to read data from dB once and store it at server memory for the first request. And for other requests it will then read it from server memory. |
| 1. No CI practices | 1. CI practices achieving using GIT and Jenkins. 2. Helps to monitor build process by whole team whenever new check in of code to GIT. 3. Once code is check in to git, then it will start executing test cases and then it will build the code. 4. If anything wrong it will intimate the team about the build status. |
| 1. Frontend design with JSP’s | 1. JSP’s Replaced with Thymeleaf templates. 2. integrated with Spring 3. Thymeleaf as a templating engine especially in that its implementation is not a part of application server but can be included in your WEB-INF/lib 4. Runs fast. |
| 1. No Testing Coverage for JAVA code. | 1. Unit & Integration Testing is added to JAVA code. 2. Helps to deliver good product. 3. Also future code changes will be monitored by the Unit test cases. So that no code break up will be there. 4. Able to monitor how much code is covered by with testing. |
| 1. Deployment was manual | 1. Continuous deployment (CD) practices. 2. Reduces the time for deployment. 3. Also helps to understand the DEVOPS practices. |
| 1. Less understanding on Agile Practices. | 1. Helps to understand AGILE process clearly and practice on product development. |