Full Stack

Caltech | Center for Technology & Management Education

Post Graduate Program in Full Stack Web Development

simplilearn

Full Stack

Caltech | Center for Technology & Management Education

Implement Frameworks the DevOps Way

simplilearn

Create a Java Application on a Spring Framework

You Already Know

Course(s):

Spring 5.0 Core Training

Recap

- Explain spring training
  - Spring definition
  - Spring setup
  - Beans creation

- Explain spring in depth
  - Spring MVC architecture
  - Reactive application with spring

A Day in the Life of a Full Stack Developer

Due to company's frequent change in goals and based on Joe's skills, he is added to another sprint.

Joe has to create a Spring MVC web application that will retrieve users based on their user ID. The retrieved user data will then be edited in a form and updated in the database using Hibernate.

In this lesson, we will learn how to solve this real-world scenario to help Joe complete his task effectively and quickly.

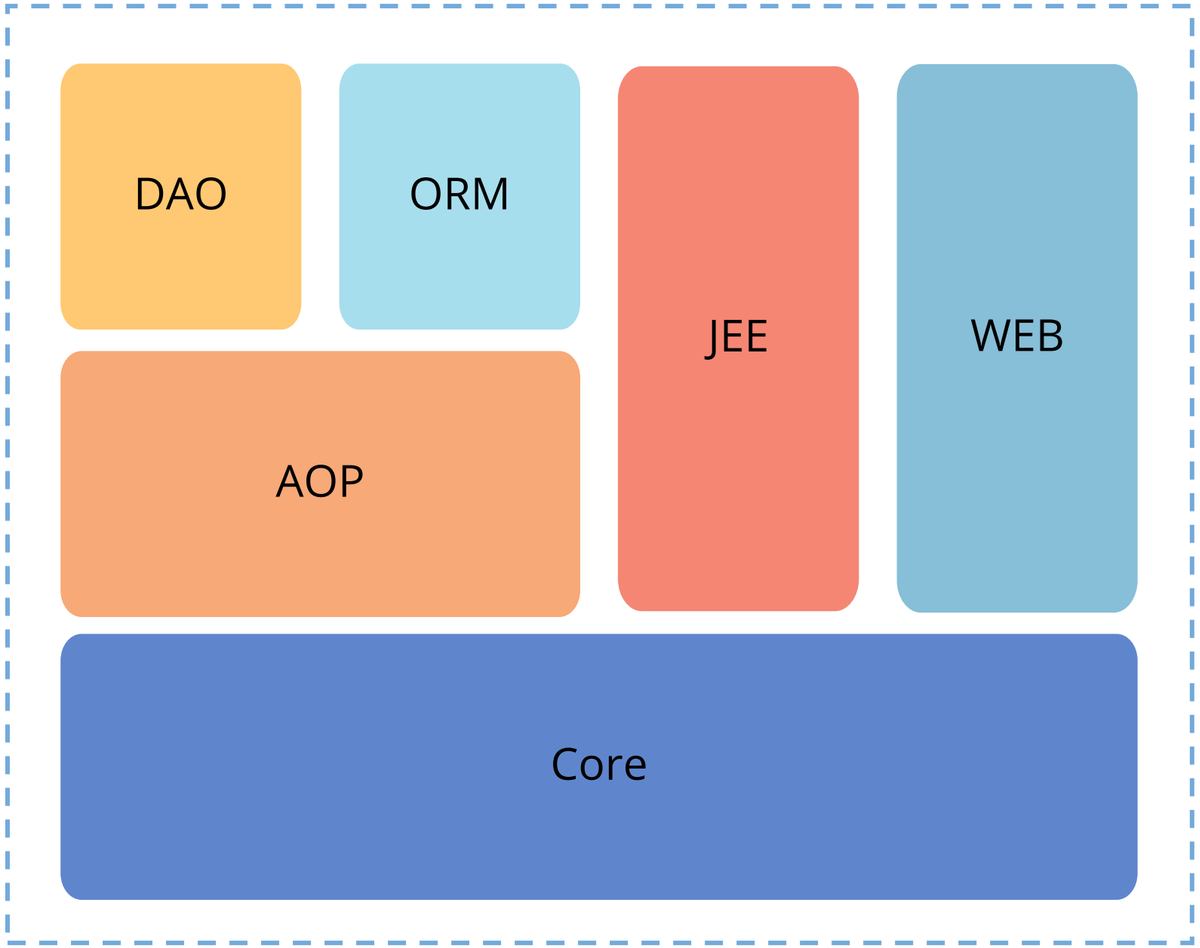
Learning Objectives

By the end of this lesson, you will be able to:

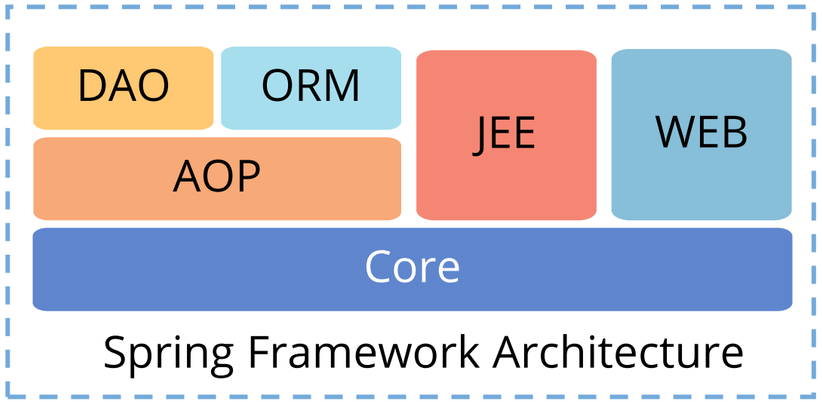
- Define Java frameworks and Spring frameworks
- Set up an Environment for Java frameworks
- Execute custom and default event handling
- Implement JDBC frameworks
- Explain Spring MVC frameworks

Spring Frameworks: Overview

Spring Framework Architecture



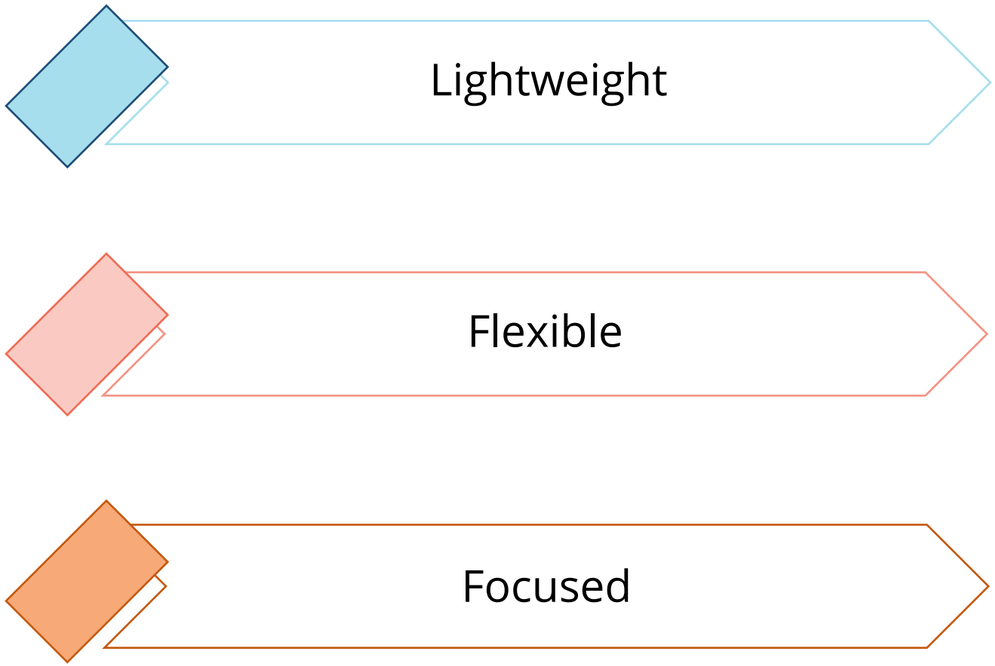
Spring Framework Architecture



Spring Framework Architecture

Core Module	Provides the Dependency Injection (DI) feature
AOP Module	Allows developers to define method-interceptors and point cuts
DAO Module	Provides an abstraction layer to the low-level tasks
ORM Module	Supports integrations with popular object relational mapping tools like Hibernate
JEE Module	Provides support for JMX, JCA, EJB, JMS, and so on
Web Module	Has its own MVC and also integrates well with the popular MVC frameworks like Struts

Spring Framework Architecture: Advantages



Assisted Practice

Environmental Setup

Duration: 30 min.

Problem Statement:

Set up a Spring framework environment.

Assisted Practice: Guidelines

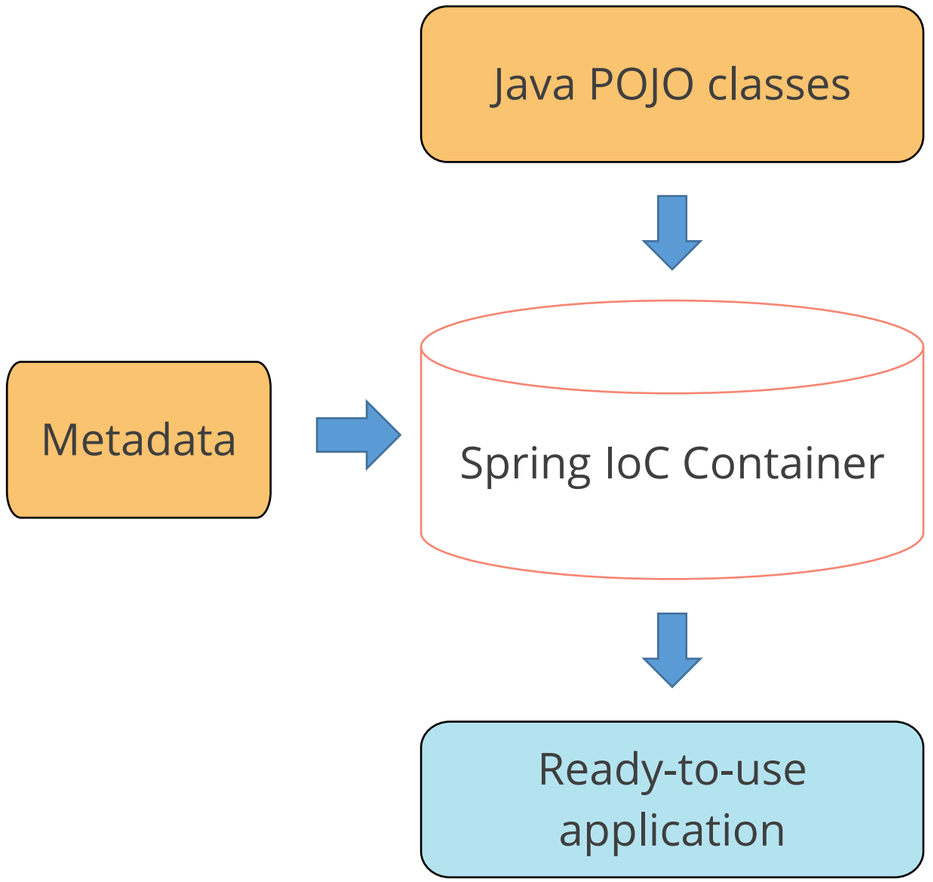
Steps to set up Spring framework environment:

1. Create a Maven project that is web enabled
2. Create pom.xml for including the required components
3. Build the project
4. Push the code to GitHub repositories

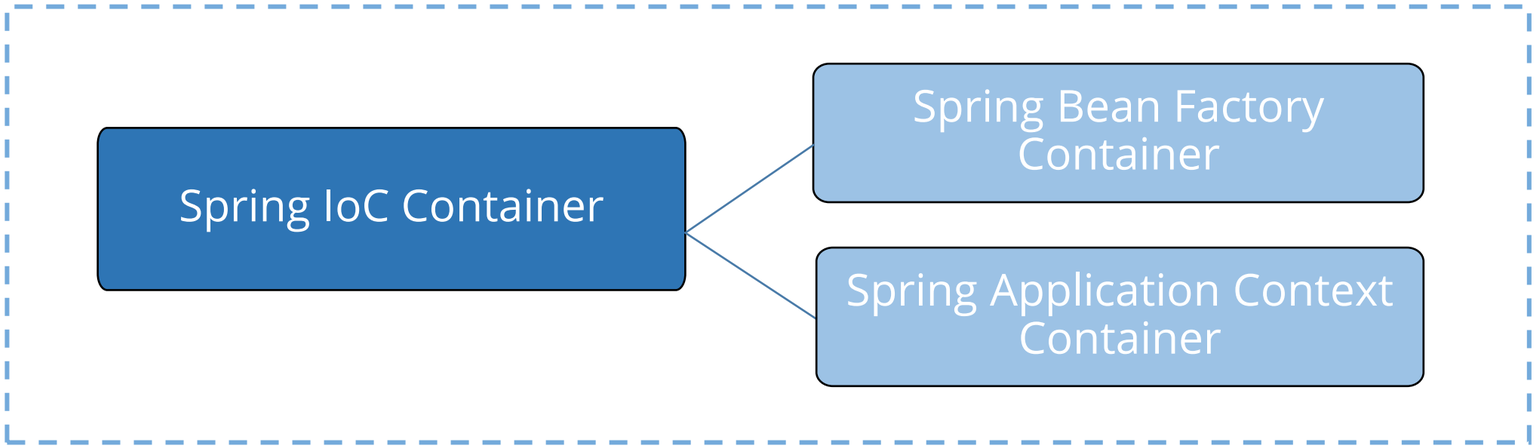
Spring Containers

Spring Container

The Spring container forms the core of the Spring Framework. The container instantiates, configures, and assembles the dependencies between the objects.



Spring IoC Containers: Types



Spring Bean Factory Container → Provides basic support of DI and is defined by org.springframework.beans.factory.BeanFactory interface

Spring Bean Factory Container → Adds more enterprise-specific functionalities and is defined by org.springframework.context.ApplicationContext interface

Uses of Spring Containers

- Spring Containers: Core of Spring Frameworks
- Create objects
- Configure objects
- Wire objects together
- Manage objects' life cycles

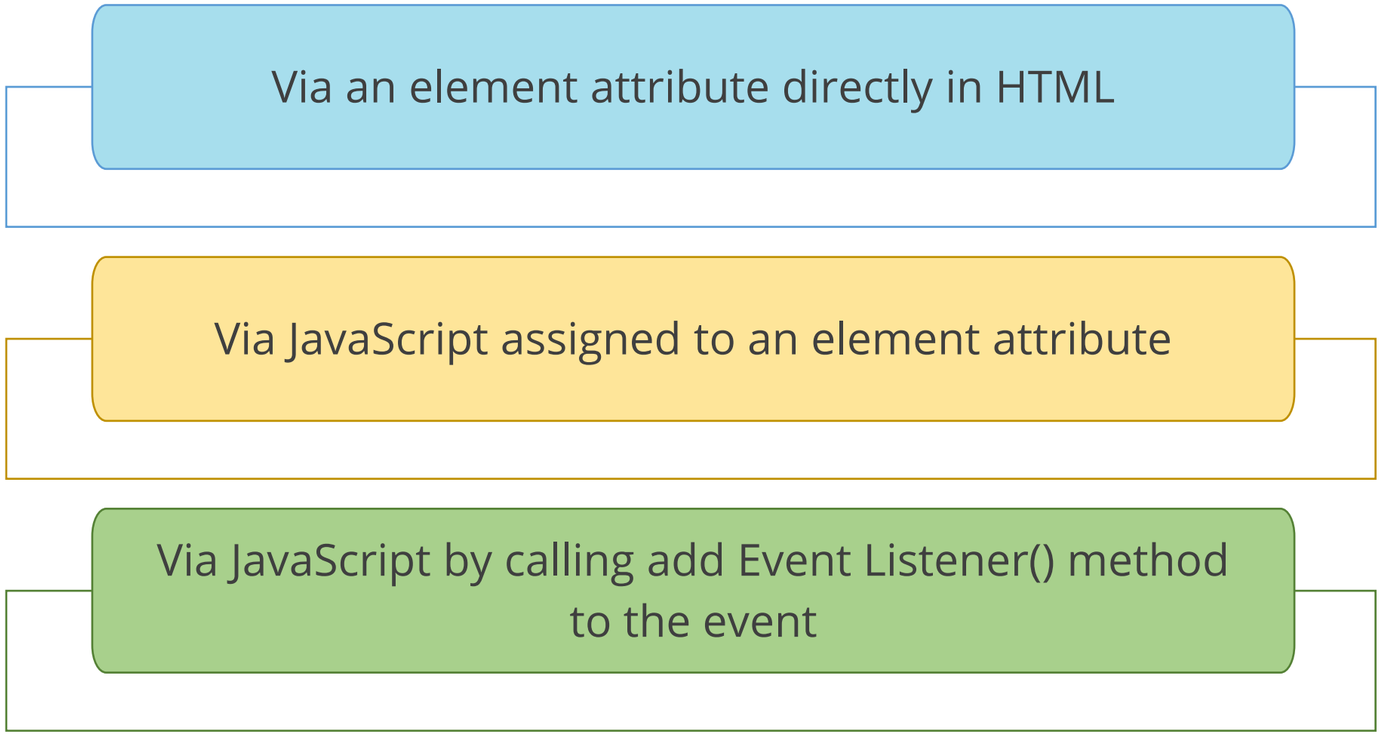
Default and Custom Event Handling

Event Handling

Event	Action involving a GUI object such as clicking a button
Event Handling	Mechanism controlling the event and deciding the result of an event
Delegation-Based Event Model	Event Handling concept used in Java
Objects Used to Implement Event Handling	Event source and Event listener objects

Types of Event Handling

An event handler can be assigned in the following ways:



Custom Event Handling in Java

Custom event handling in Java can be achieved by using custom listeners to manage callbacks in the code.

- Defines an interface
- Sets up a listener member variable
- Allows owner pass in a listener
- Triggers events on the defined listener

Assisted Practice

Default and Custom Event Handling

Duration: 60 min.

Problem Statement:

Create a Java program to demonstrate default and custom event handling.

Assisted Practice: Guidelines

Steps to demonstrate event handling:

1. Create a Maven project that is web enabled
2. Create pom.xml for including the required components
3. Create bean classes (StartEventHandler, StopEventHandler, CustomEvent, CustomEventPublisher, CustomEventListener)
4. Create a Controller class MainController
5. Create the dispatcher servlet main-servlet.xml
6. Create the views (index.jsp and customEvent.jsp)
7. Configure web.xml
8. Build, publish, and start the project
9. Run the project
10. Push the code to GitHub repositories

JDBC Framework

Spring JDBC Framework

The Spring JDBC framework provides a quick approach toward accessing the database without having to deal with the low-level tasks in the process. JDBC template class is used to perform all the database operations.

The **JDBC Template** class is the most popular and classic approach. It:

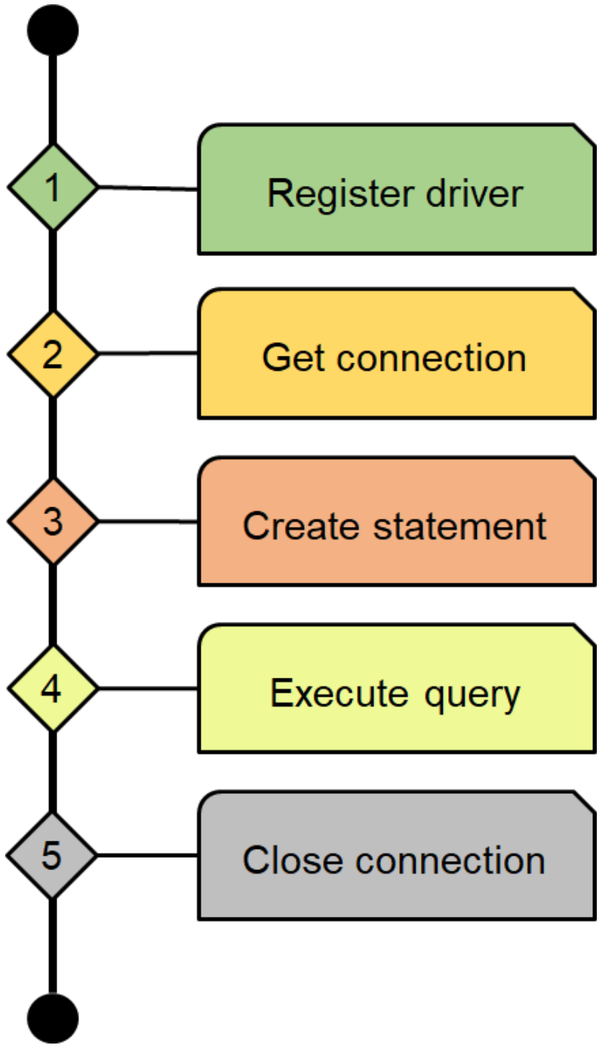
- Executes SQL queries
- Updates statements
- Stores procedure calls
- Performs iteration over result sets
- Extracts returned parameter values

Advantages of Spring JDBC Framework

Spring JDBC template:

- Provides automatic cleaning up of the resources
- Converts standard JDBC SQLExceptions into RuntimeExceptions
- Provides flexibility to the programmer

Steps to Operate JDBC Framework



Assisted Practice

JDBC Framework

Duration: 35 min.

Problem Statement:

Create a Java program to demonstrate JDBC and Spring Integration

Assisted Practice: Guidelines

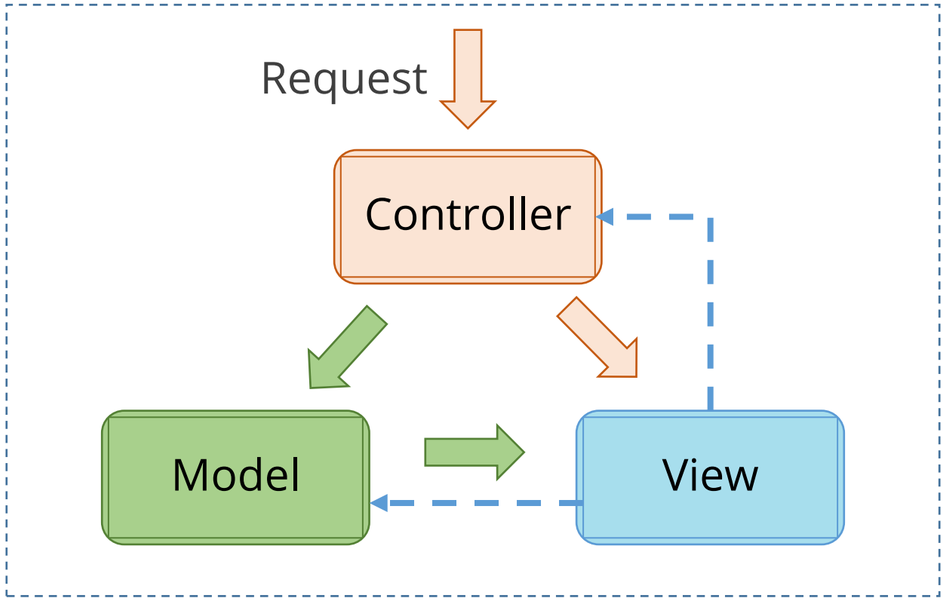
Steps to demonstrate JDBC framework functionality:

1. Create a Maven project that is web enabled
2. Create pom.xml for including the required components
3. Create a table (eproduct) in the database and fill it with sample data
4. Create an entity class (Eproduct)
5. Create a Controller class MainController
6. Create the dispatcher servlet main-servlet.xml
7. Create the views (index.jsp and listProducts.jsp)
8. Configure web.xml
9. Build, publish, and start the project
10. Run the project
11. Push the code to GitHub repositories

Spring MVC Framework

Spring MVC Framework

The Spring framework has its own MVC architecture that can be used to develop flexible and loosely coupled web applications.



Model — Encapsulates the application data

View — Renders the model data and generates output

Controller — Processes user requests, builds appropriate models, and passes them to the view for rendering

Spring MVC Components

The MVC pattern separates input logic, business logic, and UI logic and provides a loose coupling between these elements.

- Model
  - Domain-specific representation of the data on which the application works

- View
  - Provides multiple GUI components to interact with the model
  - Updates the controller with the updates to be made to the Model

- Controller
  - Updates the Model through views per user requirement
  - Interprets the user commands into common ones for the Model

Key Takeaways

- Spring framework is a Java platform to provide comprehensive infrastructure for developing Java applications.
- Spring framework is lightweight, flexible, and focused.
- Spring container creates objects and wires them together, configures them, and manages their life cycle from creation till destruction.
- The spring JDBC framework is used to provide a quick approach toward accessing the database without having to deal with the low-level tasks in the process.

Lesson-End Project

Duration: 30 min.

Searching for a Specific User and Updating the User Information

Problem Statement:

Create a Spring MVC web application that will retrieve users based on their user ID. The retrieved user data will then be edited in a form and updated in the database. The entire database processing has to be done using Hibernate. Front pages will be made in JSP.

Before the next Class

Course:

- Master Hibernate and JPA with Spring Boot in 100 steps
- DevOps: CI/CD with Jenkins pipelines, Maven, Gradle

You should be able to:

- Explain Spring Boot
- Demonstrate Hibernate and JPA integration
- Explain Jenkins
- Install and configure Jenkins
- Demonstrate basic builds on jenkins
- Explain CI/CD pipeline