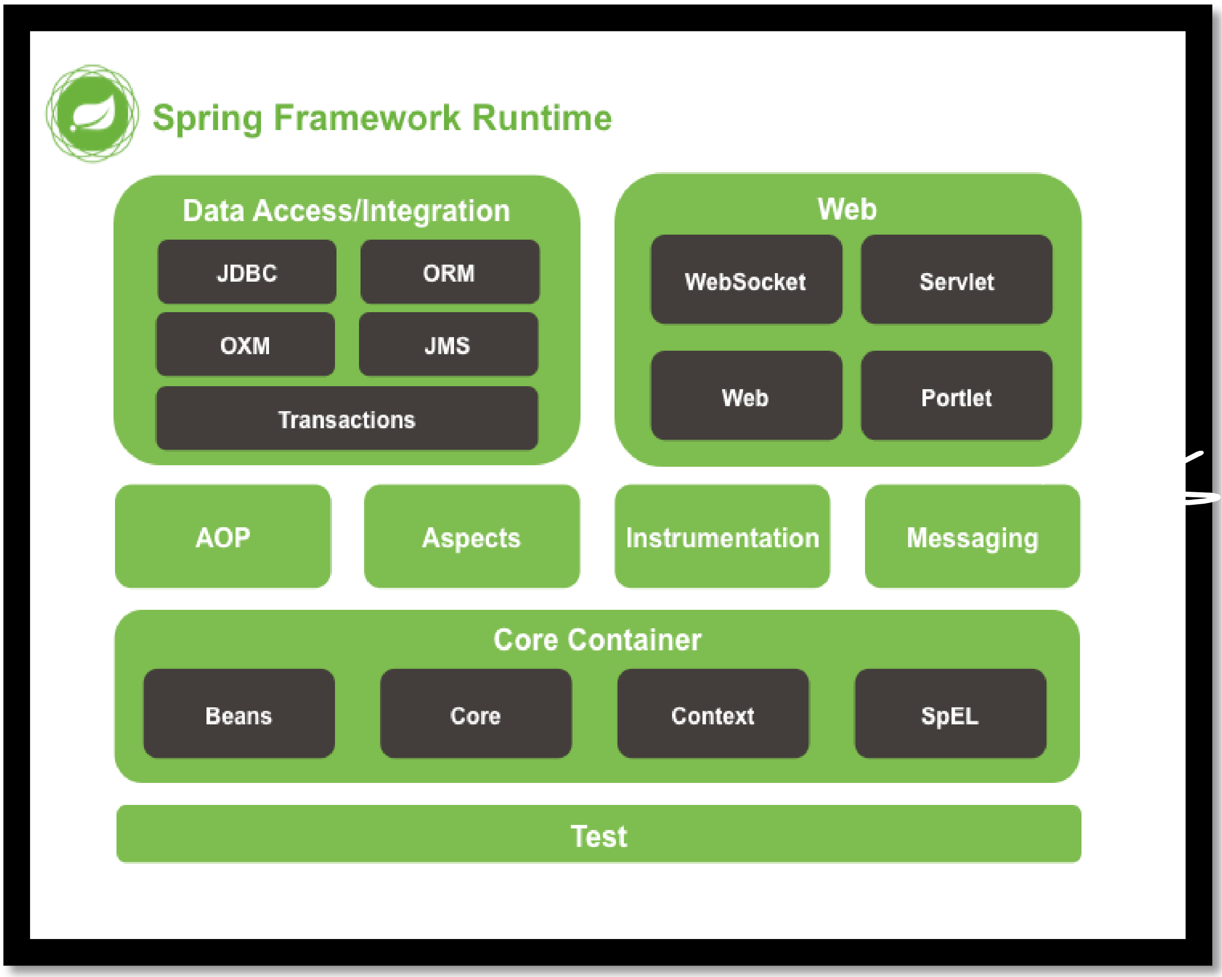
LAB13(B.Bhagyasri)

1.Explain the architecture of Spring Framework ?

Framework Modules

The Spring Framework consists of features organized into about 20 modules. These modules are grouped into Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, Messaging, and Test, as shown in the following diagram.



The Spring Framework is a comprehensive and modular framework for building enterprise-level applications in Java. It provides a wide range of features and functionalities that facilitate the development of robust, scalable, and maintainable applications. The architecture of the Spring Framework can be broken down into various parts:

1.Core Container: The core container provides the fundamental building blocks for the Spring Framework. It includes the following modules:

- **Beans**: This module is responsible for creating, managing, and wiring beans. Beans are the objects that form the backbone of an application and are managed by the Spring IoC container.
- **Core:** This module provides the basic functionalities of the Spring Framework, including the IoC (Inversion of Control) container. It includes the BeanFactory interface for managing beans. The Context (spring-context) module builds on the solid base provided by the Core and Beans modules: it is a means to access objects in a framework-style manner that is similar to a JNDI registry. The Context module inherits its features from the Beans module and adds support for internationalization (using, for example, resource bundles), event propagation, resource loading, and the transparent creation of contexts by, for example, a Servlet container. The Context module also supports Java EE features such as EJB, JMX, and basic remoting. The ApplicationContext interface is the focal point of the Context module. spring-context-support provides support for integrating common third-party libraries into a Spring application context, in particular for caching (EhCache, JCache) and scheduling (CommonJ, Quartz).
- The spring-expression module provides a powerful Expression Language for querying and manipulating an object graph at runtime. It is an extension of the unified expression language (unified EL) as specified in the JSP 2.1 specification. The language supports setting and getting property values, property assignment, method invocation, accessing the content of arrays, collections and indexers, logical and arithmetic operators, named variables, and retrieval of objects by name from Spring's IoC container. It also supports list projection and selection as well as common list aggregations.

- **Context:** This module builds on the core and beans modules, providing more advanced application context functionalities. It introduces features like internationalization, event propagation, resource loading, and application lifecycle management**.**
- **Expression Language (SpEL):** SpEL is a powerful expression language that allows for querying and manipulating objects at runtime. It is widely used in Spring configuration files.

2.Data Access/Integration: The data access and integration modules of Spring provide abstractions for working with databases, messaging systems, and other data sources.

- **JDBC (Java Database Connectivity):** Simplifies database access and error handling while providing better integration with Spring-managed transactions.
- **ORM (Object-Relational Mapping):** Provides integration with popular ORM frameworks like Hibernate, JPA, and JDO, making it easier to work with databases using object-oriented principles.
- **Transaction Management:** Offers consistent and flexible transaction management across various transaction APIs, including programmatic and declarative approaches.
- **JMS (Java Messaging Service):** Provides support for working with messaging systems, enabling communication between distributed components.

- **Spring Data:** Offers a set of high-level abstractions and repositories to simplify data access operations and eliminate boilerplate code.

3.Web Layer: The web layer modules of Spring facilitate the development of web applications and APIs.

- **Web:** Provides features for creating web applications, including multipart file uploading, initialization, and other web-related functionalities.
- **Web MVC:** Implements the Model-View-Controller (MVC) design pattern for building flexible and easily testable web applications.
- **Web WebSocket:** Offers support for WebSocket-based communication, enabling real-time interaction between clients and servers.

4.AOP (Aspect-Oriented Programming): The AOP module supports aspect-oriented programming, which allows you to separate cross-cutting concerns from the core application logic.

5.Instrumentation: This module provides class instrumentation support and integrates with third-party profilers to monitor application performance.

6.Messaging: The messaging module supports messaging-based communication, helping build scalable and flexible applications using message-driven architecture.Spring Framework 4 includes a spring-messaging module with key abstractions from the Spring Integration project such as Message, MessageChannel, MessageHandler, and others to serve as a foundation for messaging-based applications. The module also includes a set of annotations for mapping messages to methods, similar to the Spring MVC annotation based programming model.

❖ **Test:** The test module includes classes and utilities for writing unit and integration tests using JUnit or TestNG.

❖ **Security:** The security module offers comprehensive security features, including authentication, authorization, and secure communication.

❖ **Remoting:** This module supports remote method invocation, allowing objects to be accessed remotely through various protocols like RMI, HTTP, and more.

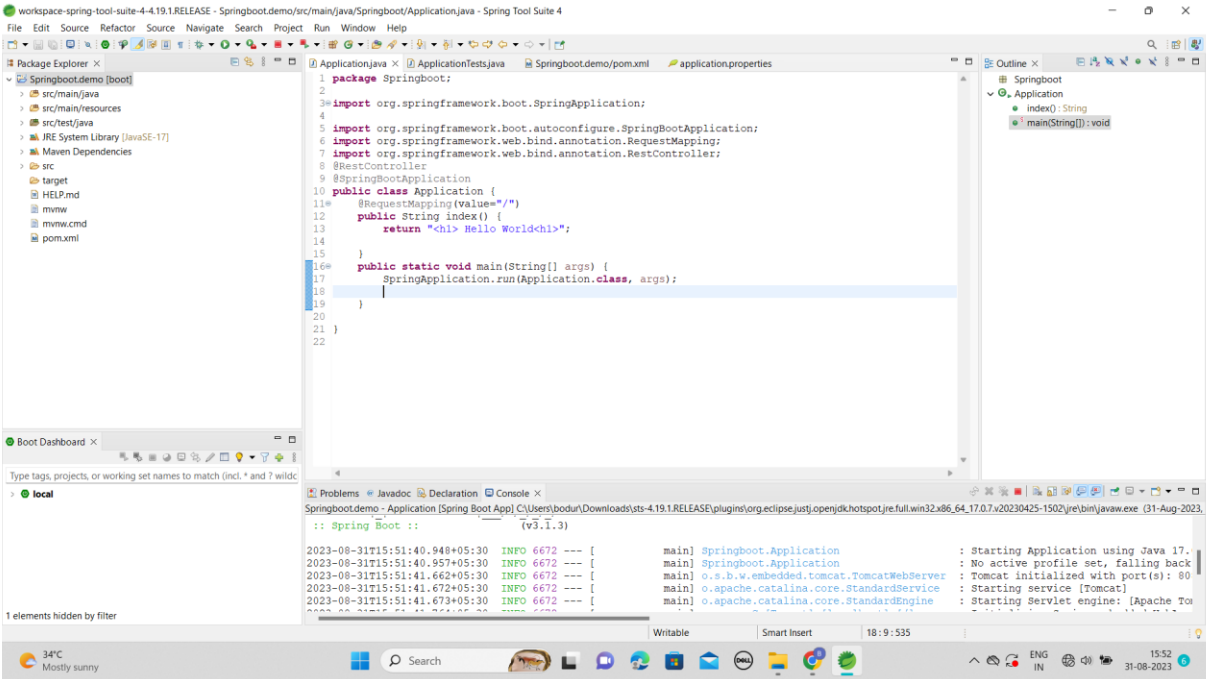
❖ **Aspects and Instrumentation:** These modules enable the integration of AOP and class instrumentation into your applications.

❖ **Languages:** Spring provides support for different languages and technologies like Groovy, Kotlin, and dynamic languages using the Spring Expression Language (SpEL).

The Spring Framework's modular architecture allows you to use the parts that are relevant to your application's requirements. Its extensive features and libraries simplify many aspects of application development, promoting clean and maintainable code. The core concept of Inversion of Control (IoC) and Dependency Injection (DI) is central to Spring's design, which promotes loose coupling and easier testing.

2. Create a Simple spring boot application to print hello world message to user in browser when the spring spoot app is up on your server.

Program:



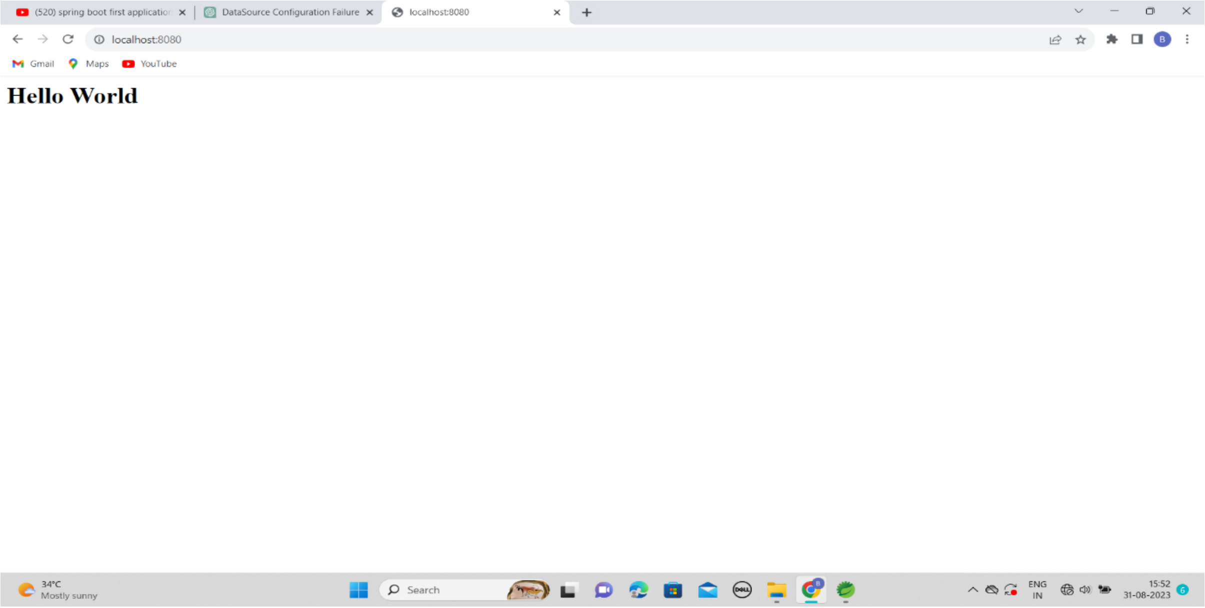
workspace-spring-tool-suite-4-4.19.1.RELEASE - Springboot.demo/src/main/java/Springboot/Application.java - Spring Tool Suite 4

```java
package Springboot;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
@SpringBootApplication
public class Application {
    @RequestMapping(value="/")
    public String index() {
        return "<h1> Hello World<h1>";

    }
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
        
    }

}

```

Springboot.demo - Application [Spring Boot App] C:\Users\bodur\Downloads\sts-4.19.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (31-Aug-2023,

```
 :: Spring Boot ::                (v3.1.3)

2023-08-31T15:51:40.948+05:30  INFO 6672 --- [           main] Springboot.Application                   : Starting Application using Java 17.
2023-08-31T15:51:40.957+05:30  INFO 6672 --- [           main] Springboot.Application                   : No active profile set, falling back
2023-08-31T15:51:41.662+05:30  INFO 6672 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 80
2023-08-31T15:51:41.672+05:30  INFO 6672 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-08-31T15:51:41.673+05:30  INFO 6672 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache To
```

Output:



localhost:8080

Hello World