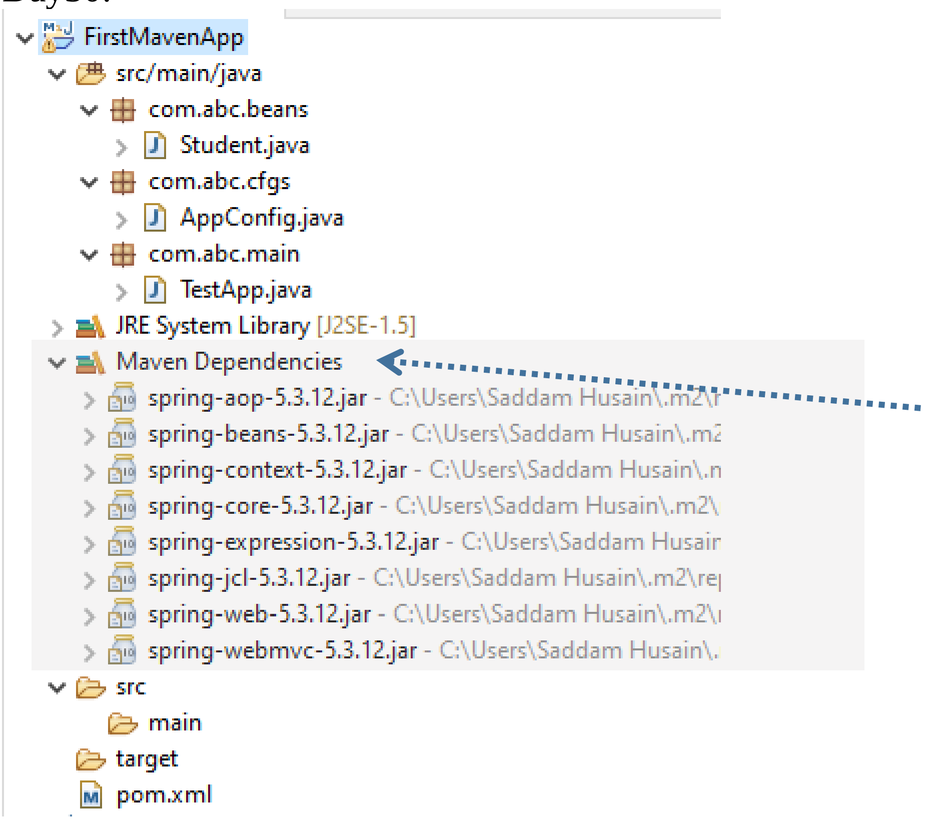
MAVEN PROJECT

Day30:-



```
FirstMavenApp
    src/main/java
        com.abc.beans
            Student.java
        com.abc.cfgs
            AppConfig.java
        com.abc.main
            TestApp.java
    JRE System Library [J2SE-1.5]
    Maven Dependencies
        spring-aop-5.3.12.jar - C:\Users\Saddam Husain\.m2\r
        spring-beans-5.3.12.jar - C:\Users\Saddam Husain\.m2
        spring-context-5.3.12.jar - C:\Users\Saddam Husain\.n
        spring-core-5.3.12.jar - C:\Users\Saddam Husain\.m2\
        spring-expression-5.3.12.jar - C:\Users\Saddam Husair
        spring-jcl-5.3.12.jar - C:\Users\Saddam Husain\.m2\re
        spring-web-5.3.12.jar - C:\Users\Saddam Husain\.m2\
        spring-webmvc-5.3.12.jar - C:\Users\Saddam Husain\.
    src
        main
    target
    pom.xml
```

Automatic inject by maven through pom.xml file

Pom.xml:-

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<groupId>com.abc</groupId>
	<artifactId>FirstMavenApp</artifactId>
	<version>1.0</version>

	<dependencies>

	<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
		<dependency>
			<groupId>org.springframework</groupId>
			<artifactId>spring-webmvc</artifactId>
			<version>5.3.12</version>
		</dependency>

	</dependencies>
</project>
```

AppConfig.java:-

(from now **pure java Annotation** use only no xml for dependencies).

```
package com.abc.cfgs;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

import com.abc.beans.Student;

@Configuration
@ComponentScan("com.abc.beans")
public class AppConfig {

        @Bean
        public Student student() {
                return new Student();
        }

}
```

Student.java:-

```
package com.abc.beans;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class Student {

        @Value("10")
        private Integer sid;

        @Value("sachin")
        private String sname;

        @Value("MI")
        private String saddress;

        @Value("sachin@gamil.com")
        private String semail;

        public Integer getSid() {
                return sid;
        }

        public void setSid(Integer sid) {
                this.sid = sid;
        }

        public String getSname() {
                return sname;
        }
```

```
	public void setSname(String sname) {
		this.sname = sname;
	}

	public String getSaddress() {
		return saddress;
	}

	public void setSaddress(String saddress) {
		this.saddress = saddress;
	}

	public String getSemail() {
		return semail;
	}

	public void setSemail(String semail) {
		this.semail = semail;
	}

	@Override
	public String toString() {

return "Student [sid=" + sid + ", sname=" + sname + ", saddress=" + saddress + ", 
semail=" + semail + "]";
	}

}
```

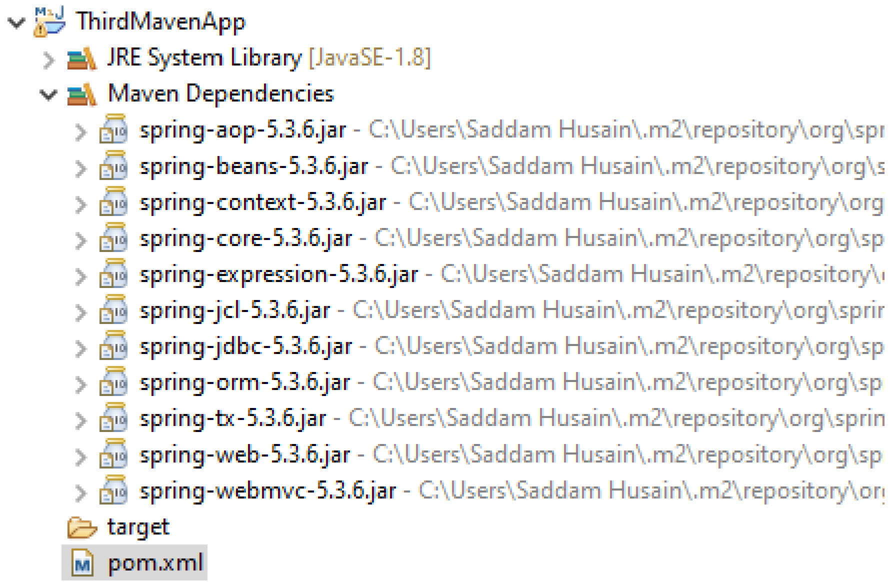
TestApp.java:-

```
package com.abc.main;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.context.support.AbstractApplicationContext;

import com.abc.beans.Student;
import com.abc.cfgs.AppConfig;

public class TestApp {

	public static void main(String[] args) {

ApplicationContext applicationContext = new AnnotationConfigApplicationContext(AppConfig.class);

		Object object = applicationContext.getBean("student");
		if (object instanceof Student) {
			Student student = (Student) object;
			System.out.println(student);
		}

		((AbstractApplicationContext) applicationContext).close();

	}

}
```

Output:-

```
Student [sid=10, sname=sachin, saddress=MI, semail=sachin@gamil.com]
```

/****************** Dependency management ***************************/



```
ThirdMavenApp
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
        spring-aop-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\spr
        spring-beans-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\s
        spring-context-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org
        spring-core-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sp
        spring-expression-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\
        spring-jcl-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sprir
        spring-jdbc-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sp
        spring-orm-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sp
        spring-tx-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sprin
        spring-web-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\org\sp
        spring-webmvc-5.3.6.jar - C:\Users\Saddam Husain\.m2\repository\or
    target
    pom.xml
```

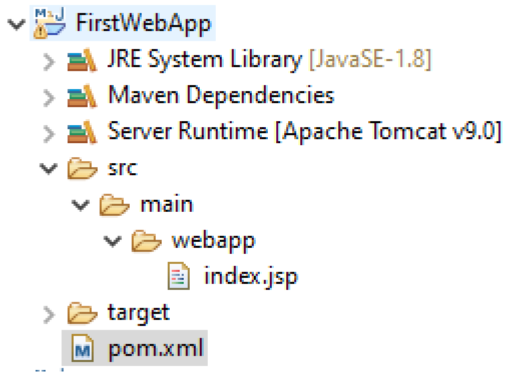
Pom.xml:-

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

      <modelVersion>4.0.0</modelVersion>
      <groupId>com.abc</groupId>
      <artifactId>ThirdMavenApp</artifactId>
      <version>1.2.3</version>

      <properties>
            <maven.compiler.source>1.8</maven.compiler.source>
            <maven.compiler.target>1.8</maven.compiler.target>
      </properties>

      <dependencyManagement>
            <dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-framework-bom -->
                  <dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-framework-bom</artifactId>
                        <version>5.3.6</version>
                        <type>pom</type>
                        <scope>import</scope>
                  </dependency>
            </dependencies>
      </dependencyManagement>
```

```
	<dependencies>
		<dependency>
			<groupId>org.springframework</groupId>
			<artifactId>spring-webmvc</artifactId>
		</dependency>
		
		<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
		<dependency>
			<groupId>org.springframework</groupId>
			<artifactId>spring-orm</artifactId>
		</dependency>
	
	</dependencies>


</project>
```

No Output.

/*********************web app using maven ************************/


FirstWebApp
JRE System Library [JavaSE-1.8]
Maven Dependencies
Server Runtime [Apache Tomcat v9.0]
src
main
webapp
index.jsp
target
pom.xml

Web.xml:-(deleted because we want to work with annotation only so in pom.xml we have to configure <failOnMissingWebXml>false</failOnMissingWebXml>

Pom.xml:-

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<groupId>com.abc</groupId>
	<artifactId>FirstWebApp</artifactId>
	<packaging>war</packaging>
	<version>0.0.1-SNAPSHOT</version>
	<name>FirstWebApp Maven Webapp</name>
	<url>http://maven.apache.org</url>
	
	<properties>
		<failOnMissingWebXml>false</failOnMissingWebXml>
		<maven.compiler.source>1.8</maven.compiler.source>
		<maven.compiler.target>1.8</maven.compiler.target>
	</properties>
```

```
    <dependencies>
        <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>3.1.0</version>
            <scope>provided</scope>
        </dependency>



        <!--
https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->
        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>javax.servlet.jsp-api</artifactId>
            <version>2.3.1</version>
            <scope>provided</scope>
        </dependency>

        <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.27</version>
        </dependency>



        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <finalName>FirstWebApp</finalName>
    </build>
</project>
```

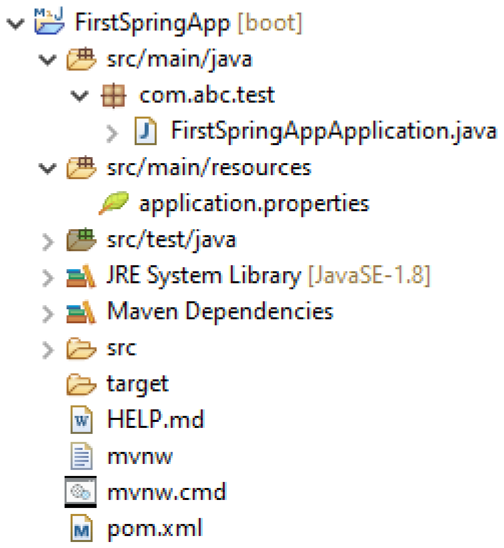
Index.jsp:-

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

No output code is just for understanding purpose we will develop this project in upcoming .

Day31:-

/************************ **SPRING BOOT** **************************/



```
FirstSpringApp [boot]
  src/main/java
    com.abc.test
      FirstSpringAppApplication.java
  src/main/resources
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Pom.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>

        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>2.5.6</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>

        <groupId>com.abc</groupId>
        <artifactId>FirstSpringApp</artifactId>
        <version>1.2.3</version>
        <name>FirstSpringApp</name>
        <description>Demo project for Spring Boot</description>

        <properties>
                <java.version>1.8</java.version>
        </properties>
        <dependencies>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-data-jpa</artifactId>
                </dependency>
```

```
		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<scope>runtime</scope>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>

	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

Application.properties:-

```
spring.datasource.driver-class-name = com.mysql.cj.jdbc.Driver
spring.datasource.url= jdbc:mysql:///abc
spring.datasource.username= root
spring.datasource.password= root123

spring.main.banner-mode  = console
spring.banner.location=classpath:com/abc/banner/myBanner.txt
```

FirstSpringAppApplication.java:-

```
package com.abc.test;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ConfigurableApplicationContext;

@SpringBootApplication
public class FirstSpringAppApplication {

	public static void main(String[] args) {

ApplicationContext applicationContext=SpringApplication.run(FirstSpringAppApplication.class, args);

	System.out.println(applicationContext.getClass().getName());

	((ConfigurableApplicationContext) applicationContext).close();
```

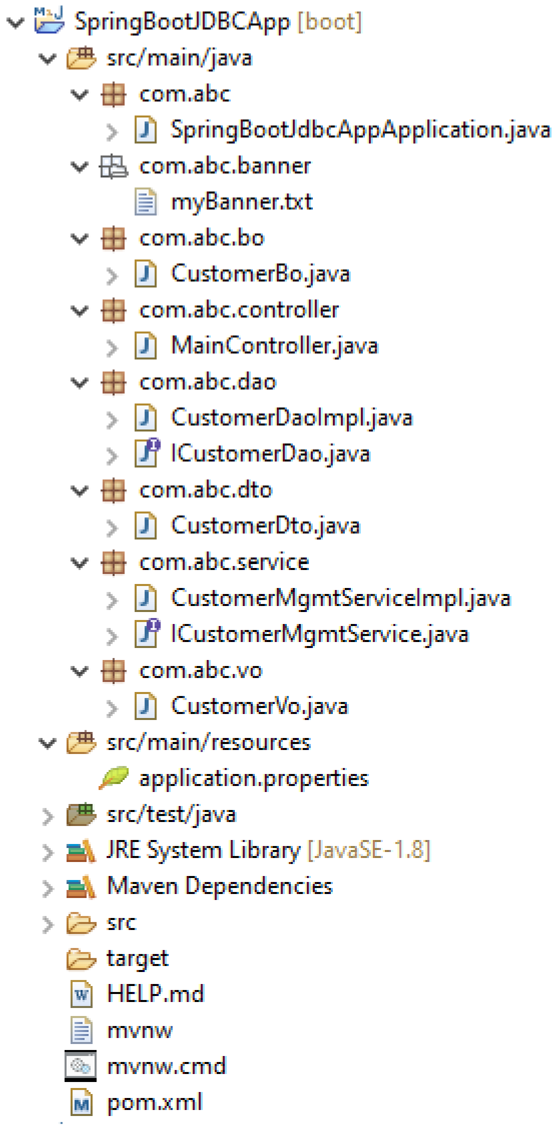
```
       }

}
```

Output:-

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.6)

2022-09-05 16:57:42.964  INFO 13084 --- [           main] com.abc.test.FirstSpringAppApplication   : Starting FirstSpringAppApplication using Java 11.0.15 on LAPTOP-M20K64TN
with PID 13084 (C:\Users\Saddam Husain\Desktop\spring\day16\FirstSpringApp\target\classes started by Saddam Husain in C:\Users\Saddam
Husain\Desktop\spring\day16\FirstSpringApp)
2022-09-05 16:57:42.975  INFO 13084 --- [           main] com.abc.test.FirstSpringAppApplication   : No active profile set, falling back to default profiles: default
2022-09-05 16:57:45.311  INFO 13084 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-09-05 16:57:45.393  INFO 13084 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 38 ms. Found 0 JPA repository
interfaces.
2022-09-05 16:57:47.186  INFO 13084 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2022-09-05 16:57:49.235  INFO 13084 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2022-09-05 16:57:49.580  INFO 13084 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-09-05 16:57:50.096  INFO 13084 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.32.Final
2022-09-05 16:57:50.892  INFO 13084 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2022-09-05 16:57:51.585  INFO 13084 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-09-05 16:57:52.609  INFO 13084 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation:
[org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-09-05 16:57:52.659  INFO 13084 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-09-05 16:57:53.051  INFO 13084 --- [           main] com.abc.test.FirstSpringAppApplication   : Started FirstSpringAppApplication in 12.271 seconds (JVM running for
14.181)
org.springframework.context.annotation.AnnotationConfigApplicationContext
2022-09-05 16:57:53.069  INFO 13084 --- [           main] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-09-05 16:57:53.078  INFO 13084 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2022-09-05 16:57:53.143  INFO 13084 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
```

DAY32

/********SPRING BOOT WITH DATA JPA MAVEN PROJECT**********/



```
SpringBootJDBCApp [boot]
  src/main/java
    com.abc
      SpringBootJdbcAppApplication.java
    com.abc.banner
      myBanner.txt
    com.abc.bo
      CustomerBo.java
    com.abc.controller
      MainController.java
    com.abc.dao
      CustomerDaoImpl.java
      ICustomerDao.java
    com.abc.dto
      CustomerDto.java
    com.abc.service
      CustomerMgmtServiceImpl.java
      ICustomerMgmtService.java
    com.abc.vo
      CustomerVo.java
  src/main/resources
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Pom.xml:- (auto generated)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>


	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>2.5.6</version>
		<relativePath /> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.abc</groupId>
	<artifactId>SpringBootJDBCApp</artifactId>
	<version>1.2.3</version>
	<name>SpringBootJDBCApp</name>
	<description>Demo project for Spring Boot</description>
	<properties>
		<java.version>1.8</java.version>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-jdbc</artifactId>
		</dependency>

	<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-dbcp2 -->
		<dependency>
			<groupId>org.apache.commons</groupId>
			<artifactId>commons-dbcp2</artifactId>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter</artifactId>
		</dependency>

		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<scope>runtime</scope>
		</dependency>

	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

myBanner.txt:-

```
 _______     _____  _____    ______    _________        _        ____    ____          _        _____      _____
|_   __ \   |_   _||_   _| .' ____ \  |  _   _  |      / \      |_   \  /   _|        / \      |_   _|    |_   _|
  | |__) |    | |    | |   | (___ \_| |_/ | | \_|     / _ \       |   \/   |         / _ \       | |        | |
  |  __ /     | '    ' |    _.____`.      | |        / ___ \      | |\  /| |        / ___ \      | |   _    | |
 _| |  \ \_    \ \__/ /    | \____) |    _| |_     _/ /   \ \_   _| |_\/_| |_     _/ /   \ \_   _| |__/ |  _| |_
|____| |___|    `.__.'      \______.'   |_____|   |____| |____| |_____||_____|   |____| |____| |________| |_____|
```

Application.properties:-

```
spring.datasource.driver-class-name = com.mysql.cj.jdbc.Driver
spring.datasource.url= jdbc:mysql:///abc
spring.datasource.username= root
spring.datasource.password= root123

spring.main.banner-mode  = console
spring.banner.location=classpath:com/abc/banner/myBanner.txt
```

CustomerVo.java:-

```java
package com.abc.vo;

public class CustomerVo {

	private String cname;
	private String caddr;
	private String pamt;
	private String time;
	private String rate;

	public CustomerVo() {
		System.out.println("");
	}

	public String getCname() {
		return cname;
	}

	public void setCname(String cname) {
		this.cname = cname;
	}

	public String getCaddr() {
		return caddr;
	}

	public void setCaddr(String caddr) {
		this.caddr = caddr;
	}

	public String getPamt() {
		return pamt;
	}

	public void setPamt(String pamt) {
		this.pamt = pamt;
	}

	public String getTime() {
		return time;
	}

	public void setTime(String time) {
		this.time = time;
	}
	public String getRate() {
		return rate;
	}

	public void setRate(String rate) {
		this.rate = rate;
	}

}
```

CustomerDto.java:-

```
package com.abc.dto;

public class CustomerDto {

	private String cname;
	private String caddr;
	private float pamt;
	private float time;
	private float rate;

	public CustomerDto() {
		System.out.println("");
	}

	public String getCname() {
		return cname;
	}

	public void setCname(String cname) {
		this.cname = cname;
	}

	public String getCaddr() {
		return caddr;
	}

	public void setCaddr(String caddr) {
		this.caddr = caddr;
	}

	public float getPamt() {
		return pamt;
	}

	public void setPamt(float pamt) {
		this.pamt = pamt;
	}

	public float getTime() {
		return time;
	}

	public void setTime(float time) {
		this.time = time;
	}

	public float getRate() {
		return rate;
	}
	public void setRate(float rate) {
		this.rate = rate;
	}

}
```

CustomerBo.java:-

```
package com.abc.bo;

public class CustomerBo {

	private String cname;
	private String caddr;
	private float pamt;
	private float intrAmt;

	public CustomerBo() {
		System.out.println("");
	}
	public String getCname() {
		return cname;
	}

	public void setCname(String cname) {
		this.cname = cname;
	}

	public String getCaddr() {
		return caddr;
	}

	public void setCaddr(String caddr) {
		this.caddr = caddr;
	}

	public float getPamt() {
		return pamt;
	}

	public void setPamt(float pamt) {
		this.pamt = pamt;
	}

	public float getIntrAmt() {
		return intrAmt;
	}

	public void setIntrAmt(float intrAmt) {
		this.intrAmt = intrAmt;
	}

}
```

ICustomerDao.java:-

```java
package com.abc.dao;

import com.abc.bo.CustomerBo;

public interface ICustomerDao {
	public int save(CustomerBo bo) throws Exception;

}
```

CustomerDaoImpl.java:-

```java
package com.abc.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.abc.bo.CustomerBo;

@Repository
public class CustomerDaoImpl implements ICustomerDao {

private static final String INSERT_QUERY =
                        "insert into customer(cname,caddr,pamt,intramt)values(?,?,?,?)";

@Autowired
private DataSource dataSource;// HikariCP best and by default supported by springboot

	@Override
	public int save(CustomerBo bo) throws Exception {

		Connection connection = null;
		connection = dataSource.getConnection();

		PreparedStatement pstmt = connection.prepareStatement(INSERT_QUERY);

		pstmt.setString(1, bo.getCname());
		pstmt.setString(2, bo.getCaddr());
		pstmt.setFloat(3, bo.getPamt());
		pstmt.setFloat(4, bo.getIntrAmt());

		int count = pstmt.executeUpdate();
		pstmt.close();
		connection.close();

		return count;
	}

}
```

ICustomerMgmtService.java:-

```
package com.abc.service;

import com.abc.dto.CustomerDto;

public interface ICustomerMgmtService {
	public String calculateInterestAmount(CustomerDto dto) throws Exception;
}
```

CustomerMgmtServiceImpl.java:-

```
package com.abc.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.abc.bo.CustomerBo;
import com.abc.dao.ICustomerDao;
import com.abc.dto.CustomerDto;

@Service
public class CustomerMgmtServiceImpl implements ICustomerMgmtService {

	@Autowired
	private ICustomerDao customerDao;

	@Override
	public String calculateInterestAmount(CustomerDto dto) throws Exception {

		CustomerBo bo = null;
		float intrAmt = 0.0f;

		intrAmt = (dto.getPamt() * dto.getTime() * dto.getRate()) / 100.0f;

		bo = new CustomerBo();
		bo.setCaddr(dto.getCaddr());
		bo.setCname(dto.getCname());
		bo.setPamt(dto.getPamt());
		bo.setIntrAmt(intrAmt);

		int count = customerDao.save(bo);

		if (count == 0) {
return "Customer registration failed:: " + dto.getPamt() + " interst:: " + dto.getRate();

		} else {

return "Customer registration succesfull:: " + dto.getPamt() + " interst:: " + dto.getRate();
		}

	}

}
```

MainController.java:-

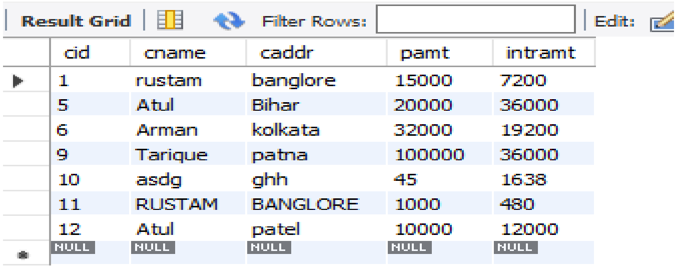
```
package com.abc.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import com.abc.dto.CustomerDto;
import com.abc.service.ICustomerMgmtService;
import com.abc.vo.CustomerVo;

@Controller("controller")
public final class MainController {

	@Autowired
	private ICustomerMgmtService customerMgmt;

	public String processCustomer(CustomerVo vo) throws Exception {
		CustomerDto dto = null;

		dto = new CustomerDto();

		dto.setCname(vo.getCname());
		dto.setCaddr(vo.getCaddr());
		dto.setPamt(Float.parseFloat(vo.getPamt()));
		dto.setRate(Float.parseFloat(vo.getRate()));
		dto.setTime(Float.parseFloat(vo.getTime()));

		String result = customerMgmt.calculateInterestAmount(dto);
		return result;

	}

}
```

SpringBootJdbcAppApplication.java:-

```
package com.abc;

import java.util.Scanner;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ConfigurableApplicationContext;

import com.abc.controller.MainController;
import com.abc.vo.CustomerVo;

@SpringBootApplication
public class SpringBootJdbcAppApplication {

	public static void main(String[] args) {

ApplicationContext applicationContext = SpringApplication.run(SpringBootJdbcAppApplication.class, args);
MainController controller = applicationContext.getBean("controller", MainController.class);
		System.out.println("Container started........");

		CustomerVo vo = new CustomerVo();
		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter the name:: ");
		vo.setCname(scanner.next());
		System.out.print("Enter the address:: ");
		vo.setCaddr(scanner.next());
		System.out.print("Enter the prinicipalAmt:: ");
		vo.setPamt(scanner.next());
		System.out.print("Enter the rate of interest:: ");
		vo.setRate(scanner.next());
		System.out.print("Enter the rate of Time period:: ");
		vo.setTime(scanner.next());

		try {
			String message = controller.processCustomer(vo);
			System.out.println(message);
		} catch (Exception e) {
			e.printStackTrace();
		} finally {
			scanner.close();
		}

		((ConfigurableApplicationContext) applicationContext).close();

	}

}
```

Output:-

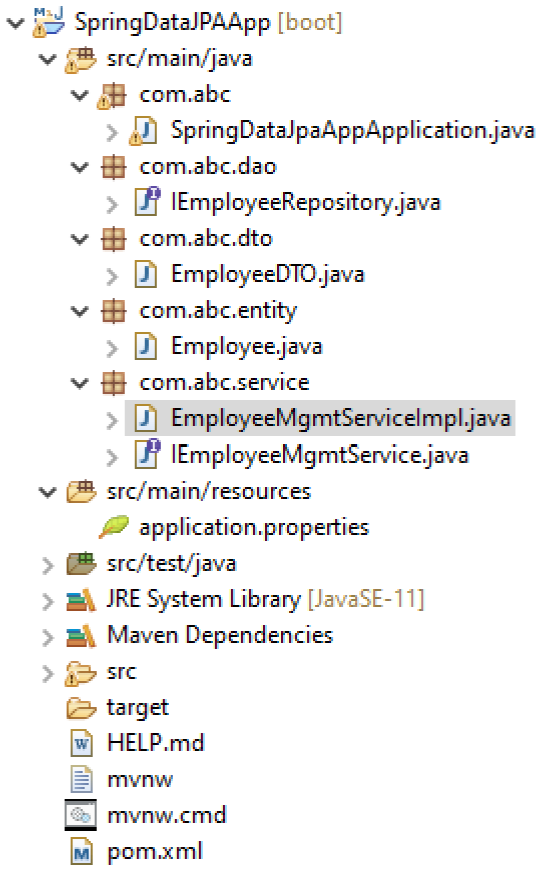
```
 _______     _____  _____   _______   _________        _        ____    ____        _        _____      _____ 
|_   __ \   |_   _||_   _| .'____ \  |  _   _  |      / \      |_   \  /   _|      / \      |_   _|    |_   _|
  | |__) |    | |    | |   | (___ \_| |_/ | | \_|     / _ \       |   \/   |       / _ \       | |        | |  
  |  __ /     | '    ' |    _.____`.      | |        / ___ \      | |\  /| |      / ___ \      | |   _    | |  
 _| |  \ \_    \ \__/ /    | \____) |    _| |_     _/ /   \ \_   _| |_\/_| |_   _/ /   \ \_   _| |__/ |  _| |_ 
|____| |___|    `.__.'      \______.'   |_____|   |____| |____| |_____||_____| |____| |____| |________| |_____|
2022-09-05 17:19:31.421  INFO 13280 --- [           main] com.abc.SpringBootJdbcAppApplication     : Starting
SpringBootJdbcAppApplication using Java 11.0.15 on LAPTOP-M20K64TN with PID 13280 (C:\Users\Saddam
Husain\Desktop\spring\day17\SpringBootJDBCApp\target\classes started by Saddam Husain in C:\Users\Saddam
Husain\Desktop\spring\day17\SpringBootJDBCApp)
2022-09-05 17:19:31.432  INFO 13280 --- [           main] com.abc.SpringBootJdbcAppApplication     : No
active profile set, falling back to default profiles: default
2022-09-05 17:19:34.205  INFO 13280 --- [           main] com.abc.SpringBootJdbcAppApplication     : Started
SpringBootJdbcAppApplication in 4.237 seconds (JVM running for 7.553)
Container started........

Enter the name:: Atul
Enter the address:: patel
Enter the prinicipalAmt:: 10000
Enter the rate of interest:: 5
Enter the rate of Time period:: 24


2022-09-05 17:20:06.320  INFO 13280 ---
[           main]
com.zaxxer.hikari.HikariDataSource       :
HikariPool-1 - Starting...
2022-09-05 17:20:07.703  INFO 13280 --- [           main] com.zaxxer.hikari.HikariDataSource       :
HikariPool-1 - Start completed.
Customer registration succesfull:: 10000.0 interst:: 5.0
2022-09-05 17:20:08.512  INFO 13280 --- [           main] com.zaxxer.hikari.HikariDataSource       :
HikariPool-1 - Shutdown initiated...
2022-09-05 17:20:08.551  INFO 13280 --- [           main] com.zaxxer.hikari.HikariDataSource       :
HikariPool-1 - Shutdown completed.
```



cid	cname	caddr	pamt	intramt
1	rustam	banglore	15000	7200
5	Atul	Bihar	20000	36000
6	Arman	kolkata	32000	19200
9	Tarique	patna	100000	36000
10	asdg	ghh	45	1638
11	RUSTAM	BANGLORE	1000	480
12	Atul	patel	10000	12000
NULL	NULL	NULL	NULL	NULL

Day33:-

Spring Data Jpa:-



```
SpringDataJPAApp [boot]
  src/main/java
    com.abc
      SpringDataJpaAppApplication.java
    com.abc.dao
      IEmployeeRepository.java
    com.abc.dto
      EmployeeDTO.java
    com.abc.entity
      Employee.java
    com.abc.service
      EmployeeMgmtServiceImpl.java
      IEmployeeMgmtService.java
  src/main/resources
    application.properties
  src/test/java
  JRE System Library [JavaSE-11]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Pom.xml(auto generated):-

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>2.5.6</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.abc</groupId>
	<artifactId>SpringDataJPAApp</artifactId>
	<version>1.2.3</version>
	<name>SpringDataJPAApp</name>
	<description>Demo project for Spring Boot</description>
	<properties>
		<java.version>11</java.version>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-jpa</artifactId>
		</dependency>
		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<scope>runtime</scope>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

Application.properties:-

```
#Datasource cfg
spring.datasource.url=jdbc:mysql:///abc
spring.datasource.username=root
spring.datasource.password= root123

#hibernate properties
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql= true
spring.jpa.properties.format_sql =true
spring.jpa.properties.dialect = org.hibernate.dialect.MySQL8Dialect.class
```

Employee.java:-

```
package com.abc.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "EMPLOYEE")
public class Employee {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private Integer eid;
```

```
    @Column(name = "ENAME", length = 15)
    private String ename;

    @Column(name = "EADD", length = 15)
    private String eadd;

    @Column(name = "ESALARY")
    private Float esalary;

    public Integer getEid() {
        return eid;
    }

    public void setEid(Integer eid) {
        this.eid = eid;
    }

    public String getEname() {
        return ename;
    }

    public void setEname(String ename) {
        this.ename = ename;
    }

    public String getEadd() {
        return eadd;
    }

    public void setEadd(String eadd) {
        this.eadd = eadd;
    }

    public Float getEsalary() {
        return esalary;
    }

    public void setEsalary(Float esalary) {
        this.esalary = esalary;
    }

    @Override
    public String toString() {

return "EmployeeDTO [eid=" + eid + ", ename=" + ename + ", eadd=" + eadd + ", esalary="
+ esalary + "]";
    }

}
```

EmployeeDto.java:-

```
package com.abc.dto;

public class EmployeeDTO {
```

```java
    private Integer eid;

    private String ename;

    private String eadd;

    private Float esalary;

    public EmployeeDTO(String ename, String eadd, Float esalary) {
        this.ename = ename;
        this.eadd = eadd;
        this.esalary = esalary;
    }

    public EmployeeDTO() {
    }

    public EmployeeDTO(Integer eid) {
        this.eid = eid;
    }

    public Integer getEid() {
        return eid;
    }

    public void setEid(Integer eid) {
        this.eid = eid;
    }

    public String getEname() {
        return ename;
    }

    public void setEname(String ename) {
        this.ename = ename;
    }

    public String getEadd() {
        return eadd;
    }

    public void setEadd(String eadd) {
        this.eadd = eadd;
    }

    public Float getEsalary() {
        return esalary;
    }

    public void setEsalary(Float esalary) {
        this.esalary = esalary;
    }

    @Override
    public String toString() {
return "EmployeeDTO [eid=" + eid + ", ename=" + ename + ", eadd=" + eadd + ", esalary="
+ esalary + "]";
    }

}
```

IEmployeeRepository.java:-

```
package com.abc.dao;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.abc.entity.Employee;

@Repository
public interface IEmployeeRepository extends CrudRepository<Employee, Integer> {

}
```

IEmployeeMgmtService.java:-

```
package com.abc.service;

import java.util.Optional;

import com.abc.dto.EmployeeDTO;
import com.abc.entity.Employee;

public interface IEmployeeMgmtService {

	public int registerEmployee(EmployeeDTO dto);

	public long getEmployeeCount();

	public Optional<EmployeeDTO> getEmployeeById(int id);

	public Iterable<EmployeeDTO> getAllEmployees();

	public Optional<Employee> removeEmployeeById(int id);
}
```

EmployeeMgmtServiceImpl.java:-

```
package com.abc.service;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

import org.springframework.beans.BeanUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.abc.dao.IEmployeeRepository;
import com.abc.dto.EmployeeDTO;
import com.abc.entity.Employee;
```

```
@Service("empService")
public class EmployeeMgmtServiceImpl implements IEmployeeMgmtService {

        @Autowired
        private IEmployeeRepository empRepo;

        @Override
        public int registerEmployee(EmployeeDTO dto) {

                Employee entity = new Employee();
                BeanUtils.copyProperties(dto, entity);
                Integer id = empRepo.save(entity).getEid();

                return id;
        }

        @Override
        public long getEmployeeCount() {
                return empRepo.count();
        }

        @Override
        public Optional<EmployeeDTO> getEmployeeById(int id) {

                Optional<Employee> optional = empRepo.findById(id);

                Optional<EmployeeDTO> optDTO = Optional.empty();

                if (optional.isPresent()) {

                        Employee entity = optional.get();

                        EmployeeDTO dto = new EmployeeDTO();
                        BeanUtils.copyProperties(entity, dto);

                        optDTO = Optional.of(dto);
                }
                return optDTO;
        }

        @Override
        public Iterable<EmployeeDTO> getAllEmployees() {

                Iterable<Employee> itEntities = empRepo.findAll();

                List<EmployeeDTO> iDTO = new ArrayList<EmployeeDTO>();

                itEntities.forEach(entity -> {

                        EmployeeDTO dto = new EmployeeDTO();
                        BeanUtils.copyProperties(entity, dto);
                        iDTO.add(dto);

                });

                return iDTO;
        }

        @Override
        public Optional<Employee> removeEmployeeById(int id) {
```

```
        Optional<Employee> optional = empRepo.findById(id);
        if (optional.isPresent()) {
            empRepo.delete(optional.get());
            return optional;
        } else {
            return optional;
        }

    }

}
```

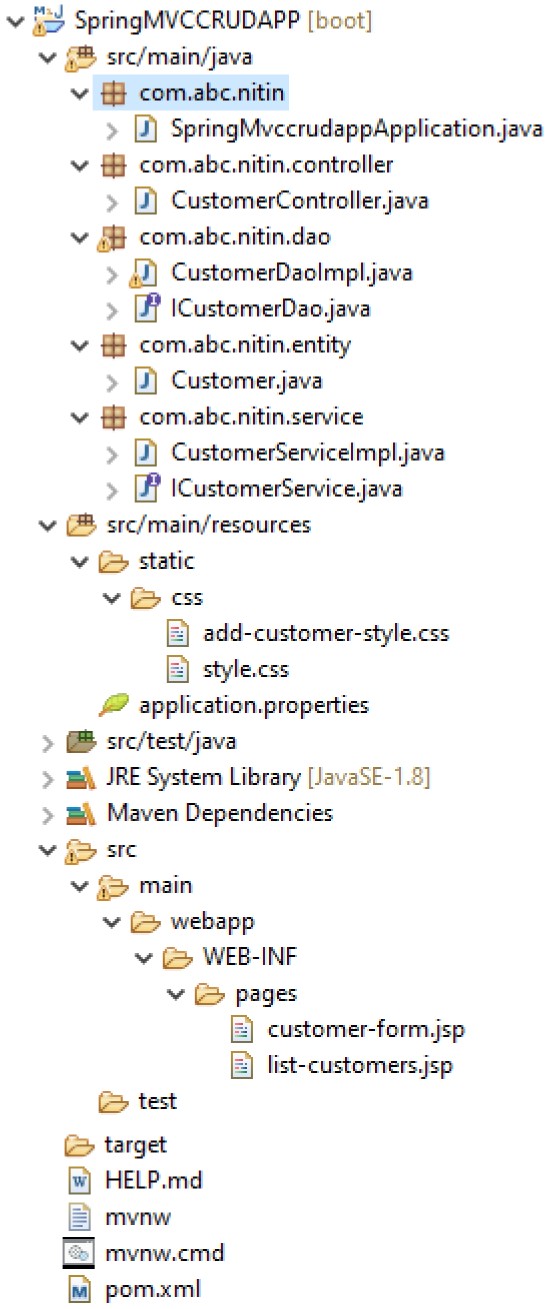
SpringDataJpaAppApplication.java:-

```
package com.abc;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ConfigurableApplicationContext;

import com.abc.service.IEmployeeMgmtService;

@SpringBootApplication
public class SpringDataJpaAppApplication {

    public static void main(String[] args) {
ApplicationContext applicationContext = SpringApplication.run(SpringDataJpaAppApplication.class, args);
IEmployeeMgmtService employeeMgmtService = applicationContext.getBean("empService", IEmployeeMgmtService.class);
        /*
          EmployeeDTO dto = new EmployeeDTO("dhoni", "CSK", 4567.8f);
          System.out.println("Employee object saved with the id:: " +
          employeeMgmtService.registerEmployee(dto));
         */

        /*
 System.out.println("No of records is :: "+employeeMgmtService.getEmployeeCount());
         */


//      Optional<EmployeeDTO> employee = employeeMgmtService.getEmployeeById(7);
//      if(employee.isPresent()) {
//          System.out.println(employee);
//      } else {
//          System.out.println("Record not found for the given id...");
//      }



          employeeMgmtService.getAllEmployees().forEach(System.out::println);
          System.out.println();
     employeeMgmtService.getAllEmployees().forEach(dto ->System.out.println(dto));
	//	System.out.println(employeeMgmtService.removeEmployeeById(8));


		((ConfigurableApplicationContext) applicationContext).close();
	}

}
```

Output:-

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.5.6)

2022-09-05 18:44:03.714  INFO 13964 --- [           main] com.abc.SpringDataJpaAppApplication      : Starting SpringDataJpaAppApplication using
2022-09-05 18:44:18.461  INFO 13964 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-09-05 18:44:20.065  INFO 13964 --- [           main] com.abc.SpringDataJpaAppApplication      : Started SpringDataJpaAppApplication in 17.803 seconds (JVM running for
20.787)

Hibernate: select employee0_.eid as eid1_0_, employee0_.eadd as eadd2_0_,
employee0_.ename as ename3_0_, employee0_.esalary as esalary4_0_ from employee
employee0_
EmployeeDTO [eid=1, ename=rustam, eadd=motihari, esalary=9000000.0]
EmployeeDTO [eid=2, ename=amir, eadd=delhi, esalary=200000.0]
EmployeeDTO [eid=3, ename=saddam, eadd=bihar, esalary=30000.0]
EmployeeDTO [eid=4, ename=atul, eadd=bhagalpur, esalary=50000.0]
EmployeeDTO [eid=7, ename=sachin, eadd=banglore, esalary=10000.0]

Hibernate: select employee0_.eid as eid1_0_, employee0_.eadd as eadd2_0_,
employee0_.ename as ename3_0_, employee0_.esalary as esalary4_0_ from employee
employee0_
EmployeeDTO [eid=1, ename=rustam, eadd=motihari, esalary=9000000.0]
EmployeeDTO [eid=2, ename=amir, eadd=delhi, esalary=200000.0]
EmployeeDTO [eid=3, ename=saddam, eadd=bihar, esalary=30000.0]
EmployeeDTO [eid=4, ename=atul, eadd=bhagalpur, esalary=50000.0]
EmployeeDTO [eid=7, ename=sachin, eadd=banglore, esalary=10000.0]

2022-09-05 18:44:21.631  INFO 13964 --- [           main] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
2022-09-05 18:44:21.638  INFO 13964 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2022-09-05 18:44:21.687  INFO 13964 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
```

Day34:-
Spring mvc:-



```
SpringMVCCRUDAPP [boot]
  src/main/java
    com.abc.nitin
      SpringMvccrudappApplication.java
    com.abc.nitin.controller
      CustomerController.java
    com.abc.nitin.dao
      CustomerDaoImpl.java
      ICustomerDao.java
    com.abc.nitin.entity
      Customer.java
    com.abc.nitin.service
      CustomerServiceImpl.java
      ICustomerService.java
  src/main/resources
    static
      css
        add-customer-style.css
        style.css
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
    main
      webapp
        WEB-INF
          pages
            customer-form.jsp
            list-customers.jsp
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Pom.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.6</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.abc</groupId>
    <artifactId>SpringMVCCRUDAPP</artifactId>
    <version>1.2.3</version>
    <name>SpringMVCCRUDAPP</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>

        <dependency>
            <groupId>org.apache.tomcat.embed</groupId>
            <artifactId>tomcat-embed-jasper</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
        </dependency>

    </dependencies>
```

```
        <build>
                <plugins>
                        <plugin>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-plugin</artifactId>
                        </plugin>
                </plugins>
        </build>

</project>
```

Pages.

List-Customer.jsp:-

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customers List</title>
<link type="text/css" rel="stylesheet" href="/css/style.css">

</head>
<body>

        <div id="wrapper">
                <div id="header">
                        <h2>Customer Relationship Manager</h2>
                </div>
        </div>

        <div id="container">
                <div id="content">

                        <!-- Button to provide save option for adding the customer -->
<input type="button" value="Add customer"onclick="window.location.href='showFormforAdd';
return false;"class="add-button">

                        <table>
                                <tr>
                                        <th>FirstName</th>
                                        <th>LastName</th>
                                        <th>Email</th>
                                        <th>Action</th>
                                </tr>

                <c:forEach var="tempCustomer" items="${customers }">
                   <c:url var="updateLink" value="/customer/showFormforUpdate">
                      <c:param name="customerId" value="${tempCustomer.id }" />
                   </c:url>
        <c:url var="deleteLink" value="/customer/deleteCustomer">
           <c:param name="customerId" value="${tempCustomer.id }" />
        </c:url>

                    <tr>
                        <td>${tempCustomer.firstName}</td>
                        <td>${tempCustomer.lastName}</td>
                        <td>${tempCustomer.email}</td>

                        <td><a href="${updateLink }">UPDATE</a></td>
<td><a href="${deleteLink }" onclick="if(!(confirm('Are you sure u want to delete a customer?'))) return false">DELETE</a></td>

                    </tr>
                </c:forEach>
            </table>

        </div>
    </div>

</body>
</html>
```

Customer-form.jsp:-

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customers List</title>
<link type="text/css" rel="stylesheet" href="/css/style.css">
<link type="text/css" rel="stylesheet"
    href="/css/add-customer-style.css">

</head>
<body>

    <div id="wrapper">
        <div id="header">
            <h2>Customer Relationship Manager</h2>
        </div>
    </div>

    <div id="container">
        <form:form action="saveCustomer" modelAttribute="customer"
            method="POST">

            <!-- sending this field also to the controller -->
            <form:hidden path="id" />
            <table>
                <tbody>
                    <tr>
                        <td><label>FirstName: </label></td>
                        <td><form:input path="firstName" /></td>
                    </tr>

                    <tr>
                        <td><label>LastName: </label></td>
                        <td><form:input path="lastName" /></td>
                    </tr>

                    <tr>
                        <td><label>Email: </label></td>
                        <td><form:input path="email" /></td>
                    </tr>

                    <tr>
                        <td></td>
                <td><input type="submit" class="save" value="save"></td>
                    </tr>
                </tbody>

            </table>


        </form:form>

        <div style=""></div>

        <p>
            <a href="/customer/list" />Back to list</a>
        </p>


    </div>

</body>
</html>
```

Css:-

Style.css:-

```
html, body{
    margin-left:15px; margin-right:15px;
    padding:0px;
    font-family:Verdana, Arial, Helvetica, sans-serif;
}

table {
    border-collapse:collapse;
    border-bottom:1px solid gray;
    font-family: Tahoma,Verdana,Segoe,sans-serif;
    width:72%;
}
```

```
th {
       border-bottom:1px solid gray;
       background:none repeat scroll 0 0 #09c332;
       padding:10px;
       color: #FFFFFF;
}

tr {
       border-top:1px solid gray;
       text-align:center;
}

tr:nth-child(even) {background: #FFFFFF}
tr:nth-child(odd) {background: #BBBBBB}

#wrapper {width: 100%; margin-top: 0px; }
#header {width: 70%; background: #09c332; margin-top: 0px; padding:15px 0px 15px 15px;}
#header h2 {width: 100%; margin:auto; color: #FFFFFF;}
#container {width: 100%; margin:auto}
#container h3 {color: #000;}
#container #content {margin-top: 20px;}

.add-button {
       border: 1px solid #666;
       border-radius: 5px;
       padding: 4px;
       font-size: 12px;
       font-weight: bold;
       width: 120px;
       padding: 5px 10px;

       margin-bottom: 15px;
       background: #cccccc;
}
```

Add-customer-style.css:-

```
form {
       margin-top: 10px;
}

label {
       font-size: 16px;
       width: 100px;
       display: block;
       text-align: right;
       margin-right: 10px;
       margin-top: 8px;
       margin-bottom: 8px;
}

input {
       width: 250px;
       border: 1px solid #666;
       border-radius: 5px;
       padding: 4px;
       font-size: 16px;
}
```

```
.save {
	font-weight: bold;
	width: 130px;
	padding: 5px 10px;
	margin-top: 30px;
	background: #cccccc;
}

table {
	border-style:none;
	width:50%;
}

tr:nth-child(even) {background: #FFFFFF}
tr:nth-child(odd) {background: #FFFFFF}

tr {
	border-style:none;
	text-align:left;
}
```

/**************************backend code ****************************/

Customer.java:-

```
package com.abc.nitin.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "customer")
public class Customer {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	@Column(name = "id")
	private Integer id;

	@Column(name = "firstname")
	private String firstName;

	@Column(name = "lastname")
	private String lastName;

	@Column(name = "email")
	private String email;

	public Integer getId() {
		return id;
	}
```

```
        public void setId(Integer id) {
                this.id = id;
        }

        public String getFirstName() {
                return firstName;
        }

        public void setFirstName(String firstName) {
                this.firstName = firstName;
        }

        public String getLastName() {
                return lastName;
        }

        public void setLastName(String lastName) {
                this.lastName = lastName;
        }

        public String getEmail() {
                return email;
        }

        public void setEmail(String email) {
                this.email = email;
        }

        @Override
        public String toString() {

return "Customer [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ",
email=" + email + "]";
        }

}
```

ICustomerDao.java:-

```
package com.abc.nitin.dao;

import java.util.List;

import com.abc.nitin.entity.Customer;

public interface ICustomerDao {

        public List<Customer> getCustomers();

        public void save(Customer customer);

        public Customer getCustomer(int id);

        public void deleteCustomer(int id);
}
```

CustomerDaoImpl.java:-

```java
package com.abc.nitin.dao;

import java.util.List;

import javax.persistence.EntityManager;

import org.hibernate.Session;
import org.hibernate.query.Query;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.abc.nitin.entity.Customer;

@Repository
public class CustomerDaoImpl implements ICustomerDao {

        @Autowired
        private EntityManager entityManager;

        @Override
        public List<Customer> getCustomers() {

                Session session = entityManager.unwrap(Session.class);
        Query<Customer> theQuery = session.createQuery("from Customer", Customer.class);

                List<Customer> customers = theQuery.getResultList();

                return customers;
        }

        @Override
        public void save(Customer customer) {
                Session session = entityManager.unwrap(Session.class);
                session.saveOrUpdate(customer);
        }

        @Override
        public Customer getCustomer(int id) {

                Session session = entityManager.unwrap(Session.class);

                Customer customer = session.get(Customer.class, id);
                return customer;
        }

        @Override
        public void deleteCustomer(int id) {
                Session session = entityManager.unwrap(Session.class);

        Query query = session.createQuery("delete from Customer where id=:customerId");
                query.setParameter("customerId", id);

                query.executeUpdate();

        }

}
```

IcustomerService.java:-

```
package com.abc.nitin.service;

import java.util.List;

import com.abc.nitin.entity.Customer;

public interface ICustomerService {

        public List<Customer> getCustomers();

        public void save(Customer customer);

        public Customer getCustomer(int id);

        public void deleteCustomer(int id);

}
```

CustomerServiceImpl.java:-

```
package com.abc.nitin.service;

import java.util.List;

import javax.transaction.Transactional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.abc.nitin.dao.ICustomerDao;
import com.abc.nitin.entity.Customer;

@Service
@Transactional
public class CustomerServiceImpl implements ICustomerService {

        @Autowired
        private ICustomerDao customerDao;

        @Override
        public List<Customer> getCustomers() {

                return customerDao.getCustomers();
        }

        @Override
        public void save(Customer customer) {
                customerDao.save(customer);

        }

        @Override
        public Customer getCustomer(int id) {
                return customerDao.getCustomer(id);
        }
        @Override
        public void deleteCustomer(int id) {

                customerDao.deleteCustomer(id);
        }

}
```

CustomerController.java:-

```
package com.abc.nitin.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.abc.nitin.entity.Customer;
import com.abc.nitin.service.ICustomerService;

@Controller
@RequestMapping("/customer")
public class CustomerController {

        @Autowired
        private ICustomerService customerService;



        @GetMapping("/list")
        public String showCustomers(Model model) {

                List<Customer> customers = customerService.getCustomers();

                model.addAttribute("customers", customers);

                return "list-customers";
        }




        @GetMapping("/showFormforAdd")
        public String showFormForAdd(Model model) {

                Customer customer = new Customer();

                model.addAttribute("customer", customer);

                return "customer-form";
        }
```

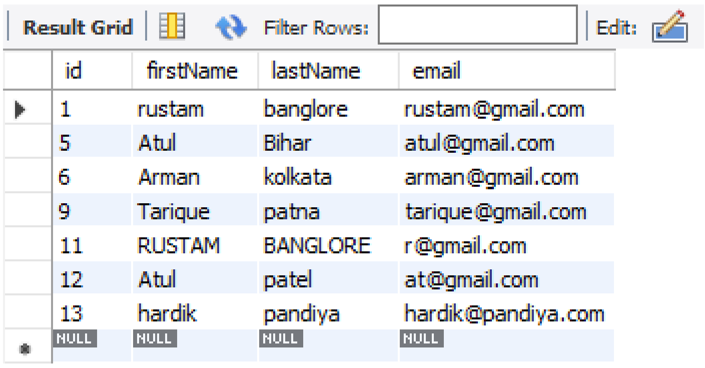
```
    @PostMapping("/saveCustomer")
    public String saveCustomer(@ModelAttribute("customer") Customer theCustomer) {

        customerService.save(theCustomer);

        return "redirect:/customer/list";

    }



@GetMapping("/showFormforUpdate")
public String showFormforUpdate(@RequestParam("customerId") int theId, Model model) {

        Customer customer = customerService.getCustomer(theId);

        model.addAttribute("customer", customer);

        return "customer-form";
    }




    @GetMapping("/deleteCustomer")
    public String deleteCustomer(@RequestParam("customerId") int theId) {

        customerService.deleteCustomer(theId);

        return "redirect:/customer/list";
    }



}
```

SpringMvccrudappApplication.java:-

```
package com.abc.nitin;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringMvccrudappApplication {

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

}
```

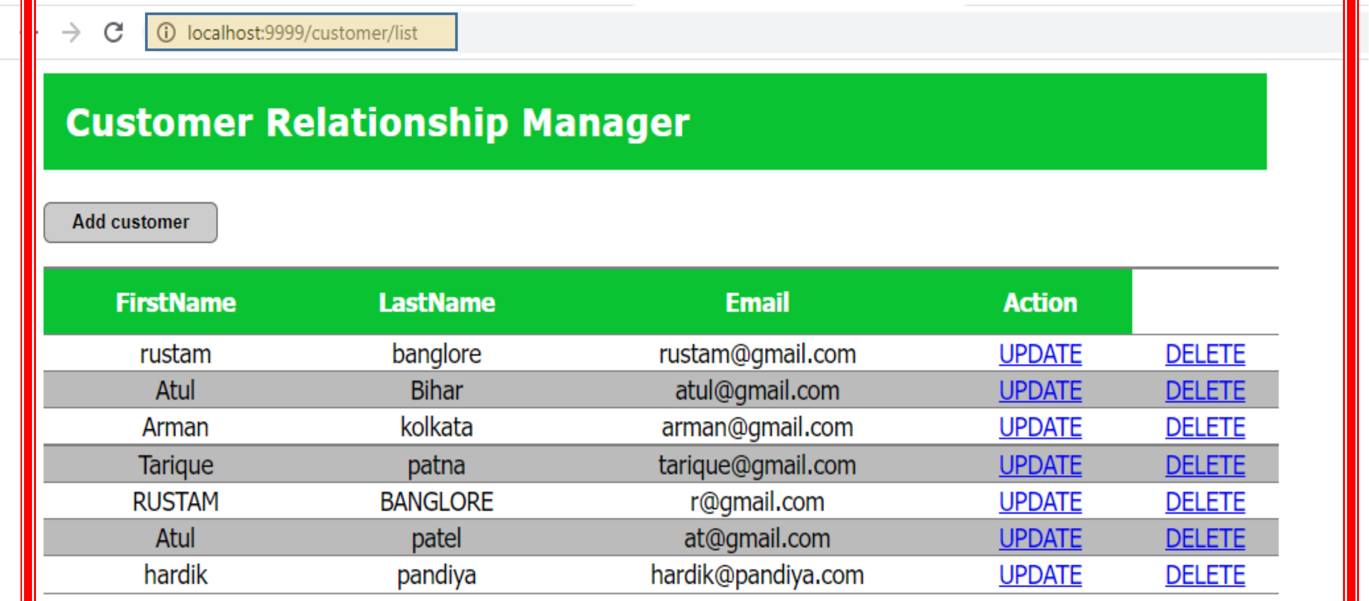
Output:-

In db:-



id	firstName	lastName	email
1	rustam	banglore	rustam@gmail.com
5	Atul	Bihar	atul@gmail.com
6	Arman	kolkata	arman@gmail.com
9	Tarique	patna	tarique@gmail.com
11	RUSTAM	BANGLORE	r@gmail.com
12	Atul	patel	at@gmail.com
13	hardik	pandiya	hardik@pandiya.com
NULL	NULL	NULL	NULL

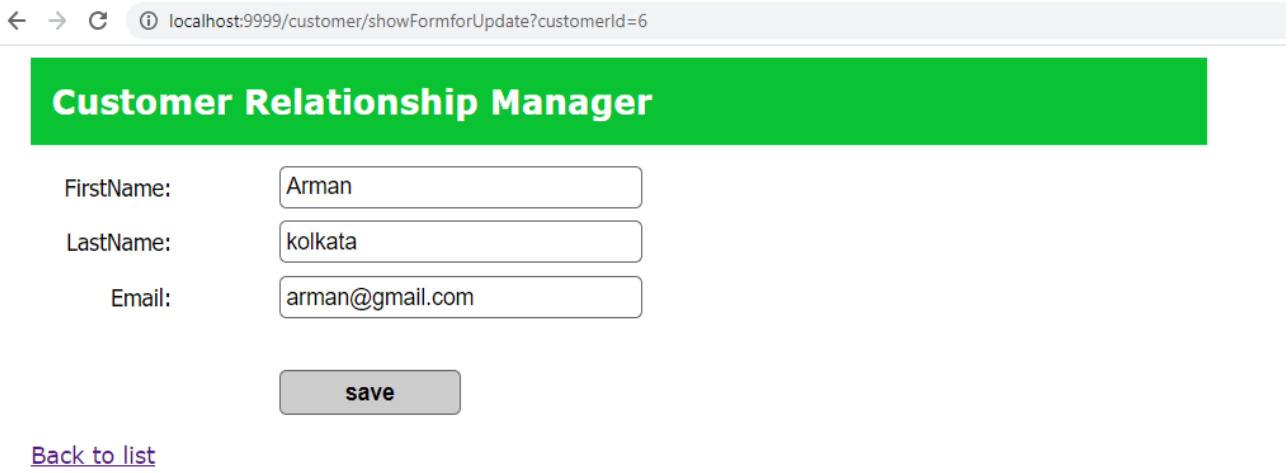
Inbrowser:-


localhost:9999/customer/list

Customer Relationship Manager

FirstName	LastName	Email	Action	
rustam	banglore	rustam@gmail.com	UPDATE	DELETE
Atul	Bihar	atul@gmail.com	UPDATE	DELETE
Arman	kolkata	arman@gmail.com	UPDATE	DELETE
Tarique	patna	tarique@gmail.com	UPDATE	DELETE
RUSTAM	BANGLORE	r@gmail.com	UPDATE	DELETE
Atul	patel	at@gmail.com	UPDATE	DELETE
hardik	pandiya	hardik@pandiya.com	UPDATE	DELETE

For clicking on update:-


localhost:9999/customer/showFormforUpdate?customerId=6

Customer Relationship Manager

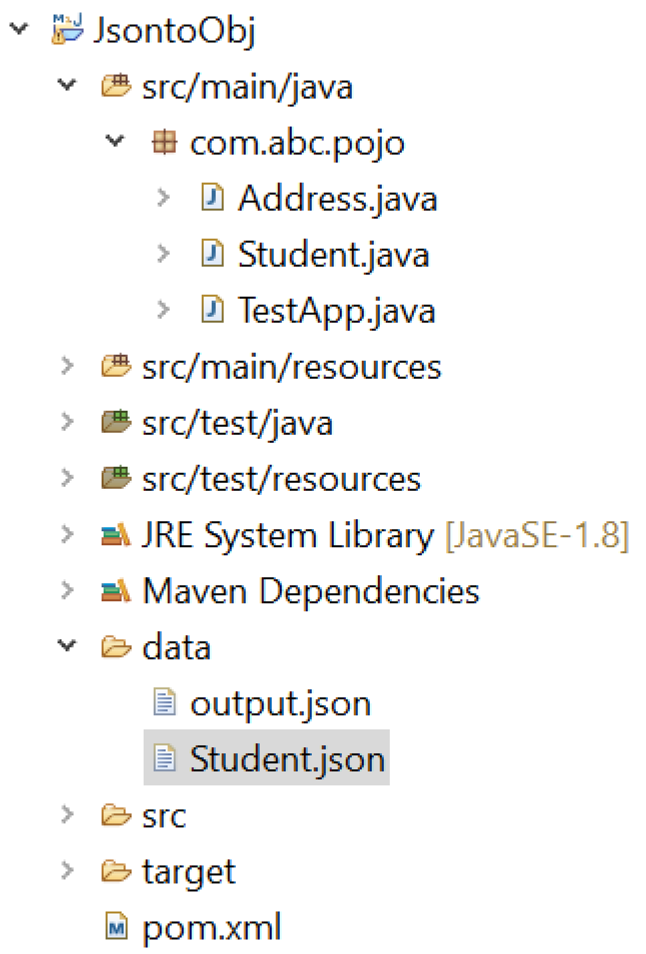
FirstName: Arman

LastName: kolkata

Email: arman@gmail.com

Back to list

/******************Json to object and vice versa*****************/



```
JsontoObj
    src/main/java
        com.abc.pojo
            Address.java
            Student.java
            TestApp.java
    src/main/resources
    src/test/java
    src/test/resources
    JRE System Library [JavaSE-1.8]
    Maven Dependencies
    data
        output.json
        Student.json
    src
    target
    pom.xml
```

Pom.xml:-

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.abc.nitin</groupId>
  <artifactId>JsontoObj</artifactId>
  <version>1.2.3</version>

    <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

 <dependencies>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.13.4</version>
</dependency>
```

```
</dependencies>
</project>
```

Student.json:-

```
{

"id":10,
"name": "rustam",
"age":23,
"country": "ind",

"address":{
    "city":"bandra",
    "state":"Bombay",
    "zip":"560034"
    },
    "isMarried":true,
    "team":["IND","MI","World|x"],
    "brands":["nokia","apple","samsung"]
 }
```

Address.java:-

```
package com.abc.pojo;

public class Address {

        @Override
        public String toString() {
        return "address [city=" + city + ", state=" + state + ", zip=" + zip + "]";
        }
        private String city;
        private String state;
        private String zip;
        public String getCity() {
                return city;
        }
        public void setCity(String city) {
                this.city = city;
        }
        public String getState() {
                return state;
        }
        public void setState(String state) {
                this.state = state;
        }
        public String getZip() {
                return zip;
        }
        public void setZip(String zip) {
                this.zip = zip;
        }


}
```

Student.java:-

```java
package com.abc.pojo;

import java.io.Serializable;
import java.util.Arrays;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;

@JsonIgnoreProperties(ignoreUnknown=true)
public class Student implements Serializable {

        /**
         *
         */
        private static final long serialVersionUID = 1L;
        private Integer id;
        private String name;
        private Integer age;
        private String country;
        private Boolean isMarried;
        private String[] team;
        private Address address;
        public Boolean getIsMarried() {
                return isMarried;
        }
        public void setIsMarried(Boolean isMarried) {
                this.isMarried = isMarried;
        }
        public String[] getTeam() {
                return team;
        }
        public void setTeam(String[] team) {
                this.team = team;
        }
        public Address getAddress() {
                return address;
        }
        public void setAddress(Address address) {
                this.address = address;
        }

        public Integer getId() {
                return id;
        }
        public void setId(Integer id) {
                this.id = id;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public Integer getAge() {
                return age;
        }
        public void setAge(Integer age) {
                this.age = age;
        }
```

```
    @Override
    public String toString() {

return "Student [id=" + id + ", name=" + name + ", age=" + age + ", country=" + country
+ ", isMarried="+ isMarried + ", team=" + Arrays.toString(team) + ", address=" +
address + "]";
    }
    public String getCountry() {
        return country;
    }
    public void setCountry(String country) {
        this.country = country;
    }


}
```

TestApp.java:-

```
package com.abc.pojo;

import java.io.File;
import java.io.IOException;

import com.fasterxml.jackson.core.exc.StreamReadException;
import com.fasterxml.jackson.databind.DatabindException;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.SerializationFeature;

public class TestApp {

    public static void main(String[] args) {


        //create an object for ObjectMapper(from JacksonApi)

        ObjectMapper mapper=new ObjectMapper();

        try {
            //read json file and convert to java pojo
    Student student=mapper.readValue(new File("data/Student.json"), Student.class);//
            System.out.println(student); //deserialization
            System.out.println();

        mapper.enable(SerializationFeature.INDENT_OUTPUT);
        mapper.writeValue(new File("data/output.json"), student); //serialization
        } catch (StreamReadException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (DatabindException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

}
```

Output:-

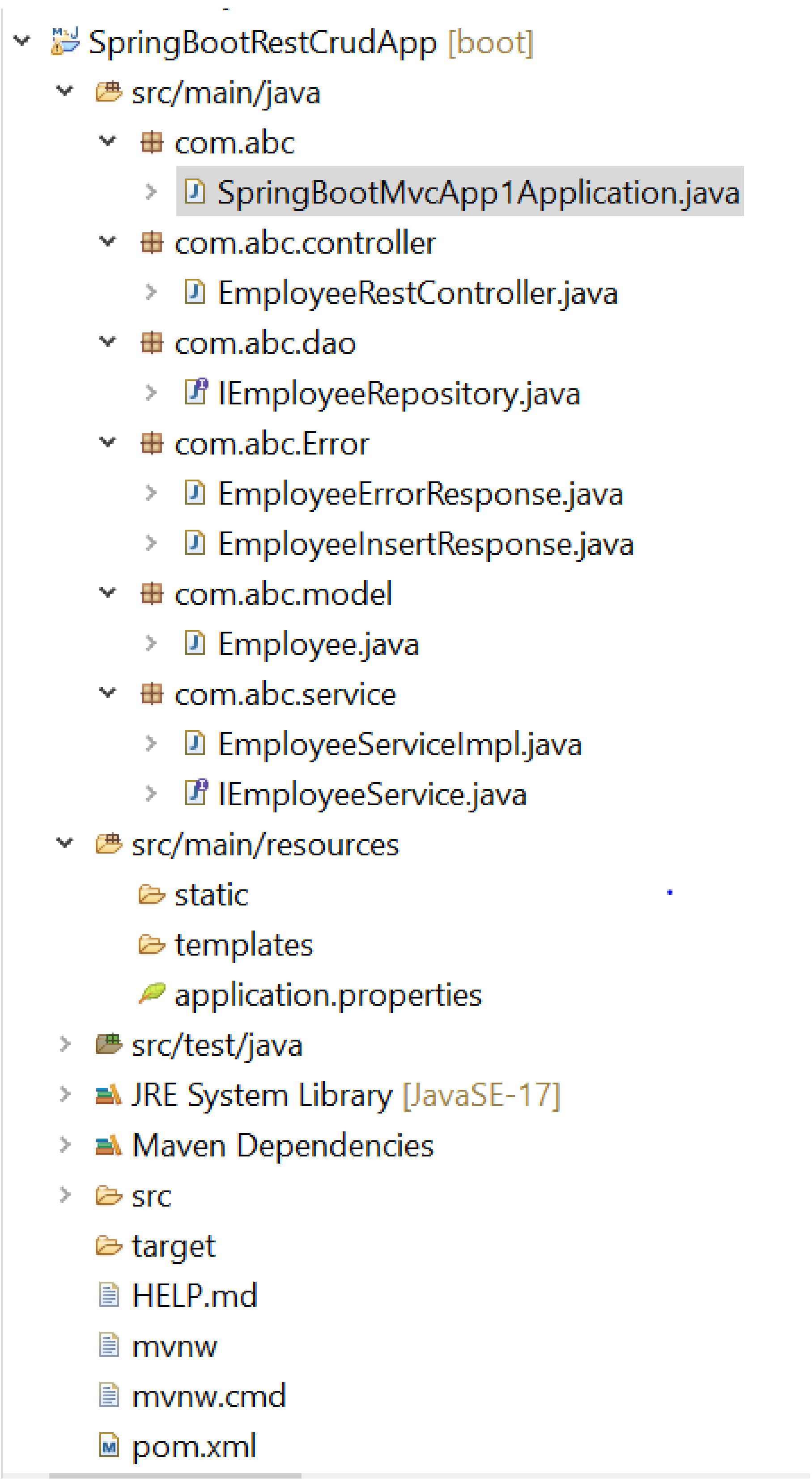
InObject:-

```
Student [id=10, name=rustam, age=23, country=ind, isMarried=true, team=[IND, MI,
World|x], address=address [city=bandra, state=Bombay, zip=560034]]
```

INOutput.json:-

```
{
  "id" : 10,
  "name" : "rustam",
  "age" : 23,
  "country" : "ind",
  "isMarried" : true,
  "team" : [ "IND", "MI", "World|x" ],
  "address" : {
    "city" : "bandra",
    "state" : "Bombay",
    "zip" : "560034"
  }
}
```

/************* **spring Rest with Data jpa Crud project** *****************/



```
SpringBootRestCrudApp [boot]
    src/main/java
        com.abc
            SpringBootMvcApp1Application.java
        com.abc.controller
            EmployeeRestController.java
        com.abc.dao
            IEmployeeRepository.java
        com.abc.Error
            EmployeeErrorResponse.java
            EmployeeInsertResponse.java
        com.abc.model
            Employee.java
        com.abc.service
            EmployeeServiceImpl.java
            IEmployeeService.java
    src/main/resources
        static
        templates
        application.properties
    src/test/java
    JRE System Library [JavaSE-17]
    Maven Dependencies
    src
    target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

Pom.xml:-

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.3</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.abc.rustam</groupId>
    <artifactId>SpringRestMVCCRUDApp</artifactId>
    <version>1.2.3</version>
    <name>SpringBootMVCApp-1</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

Application.properties:-

```
spring.datasource.url=jdbc:mysql:///abc
spring.datasource.username=root
spring.datasource.password=root123

  server.port=9999
```

Employee.java:-

```java
package com.abc.model;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="employee")
public class Employee implements Serializable{

	/**
	 * 
	 */
	private static final long serialVersionUID = 1L;

	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Column(name="id")
	private Integer id;

	@Column(name="first_name" ,length=20)
	private String first_name;

	@Column(name="last_name" ,length=20)
	private String last_name;

	@Column(name="salary")
	private Integer salary;

	public Integer getId() {
		return id;
	}

	public void setId(Integer id) {
		this.id = id;
	}

	public String getFirst_name() {
		return first_name;
	}
```

```
    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }

    public String getLast_name() {
        return last_name;
    }

    public void setLast_name(String last_name) {
        this.last_name = last_name;
    }

    public Integer getSalary() {
        return salary;
    }

    public void setSalary(Integer salary) {
        this.salary = salary;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", first_name=" + first_name + ",
last_name=" + last_name + ", salary=" + salary
                        + "]";
    }


}
```

IemployeeRepository.java:-

```
package com.abc.dao;

import org.springframework.data.jpa.repository.JpaRepository;

import com.abc.model.Employee;

public interface IEmployeeRepository extends JpaRepository<Employee, Integer> {

}
```

IemployeeService.java:-

```
package com.abc.service;

import java.util.List;

import com.abc.model.Employee;

public interface IEmployeeService {

    public List<Employee> findAll(); //getting all record
```

```
        public void save( Employee employee);   //saving new record

        public Employee getEmployee(Integer id); //getting single record

        public void delete(Integer id);   //deleting a record
}
```

EmployeeServiceImpl.java:-

```
package com.abc.service;

import java.util.List;

import javax.transaction.Transactional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.abc.dao.IEmployeeRepository;
import com.abc.model.Employee;

@Service
@Transactional
public class EmployeeServiceImpl implements IEmployeeService {


        @Autowired
        private IEmployeeRepository repository;

        @Override
        public List<Employee> findAll() {

                return repository.findAll();
        }

        @Override
        public void save(Employee employee) {

                repository.save(employee);

        }

        @Override
        public Employee getEmployee(Integer id) {

                return repository.findById(id).get();
        }

        @Override
        public void delete(Integer id) {

      repository.deleteById(id);
        }

}
```

EmployeeRestController.java:-

```
package com.abc.controller;

import java.util.List;
import java.util.NoSuchElementException;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.abc.Error.EmployeeErrorResponse;
import com.abc.Error.EmployeeInsertResponse;
import com.abc.model.Employee;
import com.abc.service.IEmployeeService;

@RestController
public class EmployeeRestController {

        @Autowired
        private IEmployeeService service;



        @GetMapping("api/employees")
        public List<Employee> findAll() {

                return service.findAll();
        }





        @GetMapping("api/employees/{id}")
        public ResponseEntity<?> getEmployee(@PathVariable Integer id) {

                try {
                        Employee employee = service.getEmployee(id);
                        return new ResponseEntity<Employee>(employee,HttpStatus.FOUND);
                } catch (NoSuchElementException e) {

                        EmployeeErrorResponse errorresponse=new EmployeeErrorResponse();
        errorresponse.setErrorcode(HttpStatus.NOT_FOUND.value());
        errorresponse.setErrorMessage("Employee not found for the give id :: " +id);

                        return new ResponseEntity<>(errorresponse,HttpStatus.NOT_FOUND);

                }
        }
```

```
    @PostMapping("/api/employees")
    public ResponseEntity<?> save(@RequestBody Employee employee) {
        service.save(employee);
        EmployeeInsertResponse response= new EmployeeInsertResponse();

response.setMessage("Recorded inserted suceesfully with status code ::" + HttpStatus.OK.value());
        return new ResponseEntity<>(response,HttpStatus.OK);

    }


    @DeleteMapping("api/employees/{id}")
    public ResponseEntity<?> delete(@PathVariable Integer id) {

        try {
            service.getEmployee(id);
            service.delete(id);

            EmployeeErrorResponse errorresponse=new EmployeeErrorResponse();

errorresponse.setErrorMessage("Employee deleted successfully for a given id ::"+id);
            return new ResponseEntity<>(errorresponse,HttpStatus.OK);
        } catch (NoSuchElementException e) {

            EmployeeErrorResponse errorresponse=new EmployeeErrorResponse();
    errorresponse.setErrorcode(HttpStatus.NOT_FOUND.value());
    errorresponse.setErrorMessage("Employee not found for the give id :: " +id);

            return new ResponseEntity<>(errorresponse,HttpStatus.NOT_FOUND);

        }
    }



    @PutMapping("api/employees/{id}")
     public ResponseEntity<?> update(@RequestBody Employee employee ,@PathVariable
Integer id){

        try {
            Employee existingemployee=service.getEmployee(id);
            System.out.println(existingemployee);
            service.save(employee);

            EmployeeErrorResponse errorresponse=new EmployeeErrorResponse();

errorresponse.setErrorMessage("Employee record updated successfully for a id ::"+id);
            return new ResponseEntity<>(errorresponse,HttpStatus.OK);
        } catch (NoSuchElementException e) {

            EmployeeErrorResponse errorresponse=new EmployeeErrorResponse();
    errorresponse.setErrorcode(HttpStatus.NOT_FOUND.value());
    errorresponse.setErrorMessage("Employee not found for the give id :: " +id);

            return new ResponseEntity<>(errorresponse,HttpStatus.NOT_FOUND);

        }
    }
}
```

EmployeeErrorResponse.java:-

```
package com.abc.Error;

public class EmployeeErrorResponse {

	private Integer errorcode;
	private String errorMessage;
	public Integer getErrorcode() {
		return errorcode;
	}
	public void setErrorcode(Integer errorcode) {
		this.errorcode = errorcode;
	}
	public String getErrorMessage() {
		return errorMessage;
	}
	public void setErrorMessage(String errorMessage) {
		this.errorMessage = errorMessage;
	}
}
```

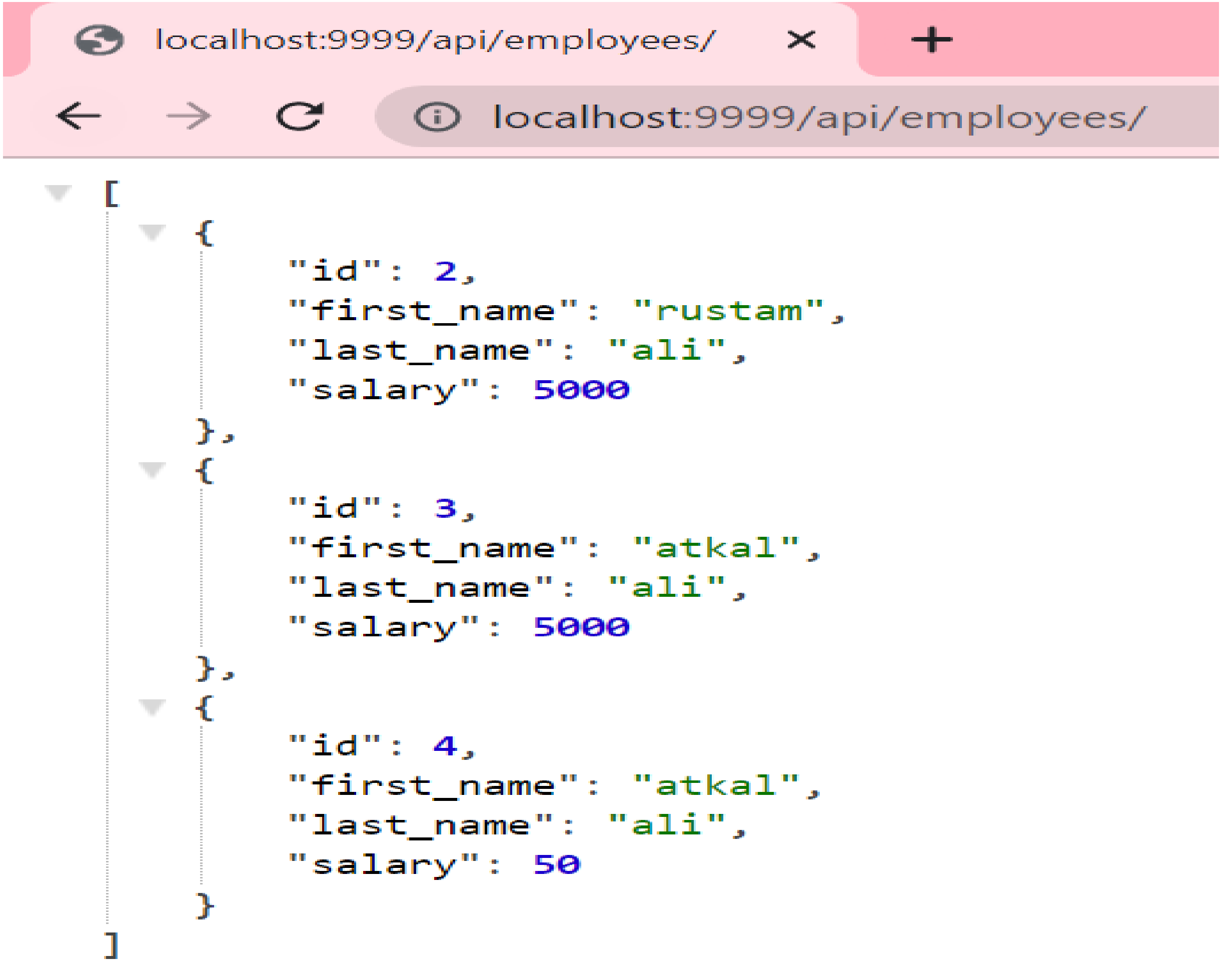
EmployeeInsertResponse.java:-

```
package com.abc.Error;

public class EmployeeInsertResponse {

	private String message;

	public String getMessage() {
		return message;
	}

	public void setMessage(String message) {
		this.message = message;
	}
}
```

SpringBootMvcApp1Application.java:-

```
package com.abc;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringBootMvcApp1Application {

	public static void main(String[] args) {
		SpringApplication.run(SpringBootMvcApp1Application.class, args);
	}

}
```

Output(also use postman tool to hit api):-



```
[
    {
        "id": 2,
        "first_name": "rustam",
        "last_name": "ali",
        "salary": 5000
    },
    {
        "id": 3,
        "first_name": "atkal",
        "last_name": "ali",
        "salary": 5000
    },
    {
        "id": 4,
        "first_name": "atkal",
        "last_name": "ali",
        "salary": 50
    }
]
```