

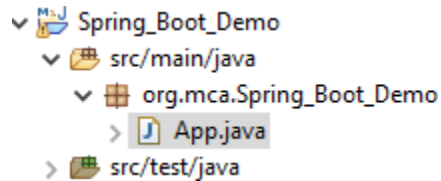
Program based on Assignment based Spring Boot and RESTful Web Services

Practical 10

Program no. 1 Write a program to create a simple Spring Boot application that prints a message.

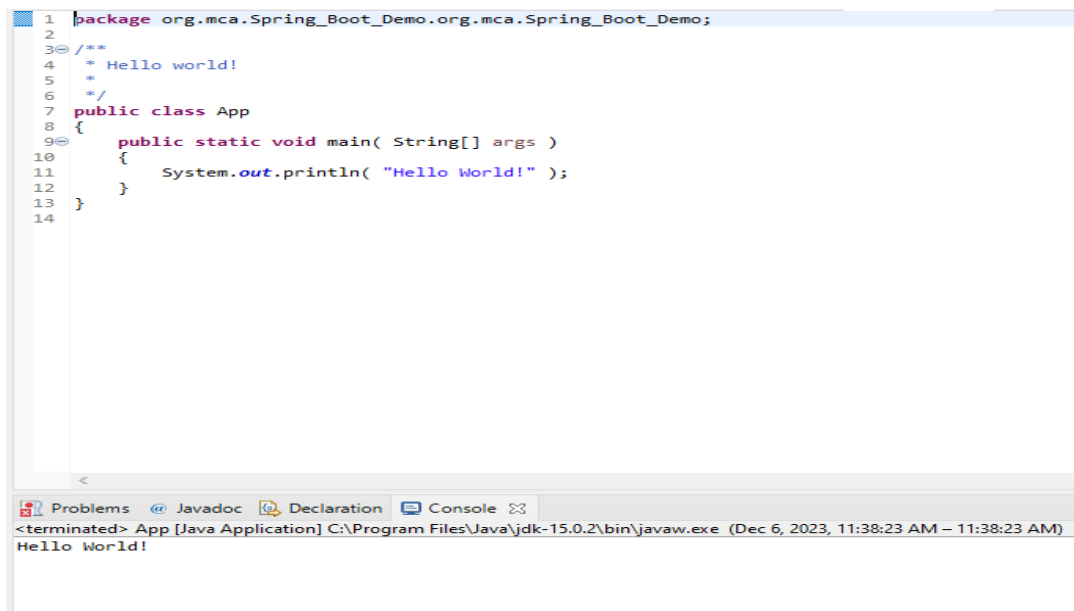
Line Of Code:

File-> new-> maven project-> default(internal -> Quick start-> 1.1) next->group_ID-> artifact_ID->go for project-> expand the project-> src-> main.JAVA-> org.mca.Spring_Boot_Demo->App.java(open)



```
package org.mca.Spring_Boot_Demo;
/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```

Output Screen:



Program based on Assignment based Spring Boot and RESTful Web Services

Program no. 2 Write a program to demonstrate RESTful Web Services with spring boot.

WelcomeController.java

```
package com.springboot.app;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
public class WelcomeController {
    @GetMapping("/welcome")
    public String welcome()
    {
        return "welcome to springboot dear";
    }
}
```

SpringBootApplication.java

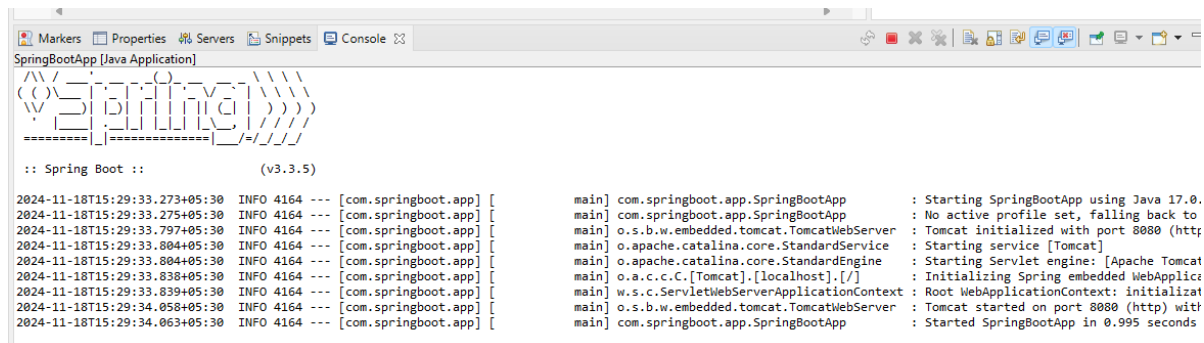
```
package com.springboot.app;

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

@SpringBootApplication
public class SpringBootApplication {

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

Output

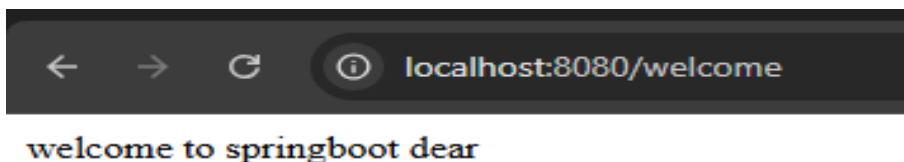


```
SpringBootApplication [Java Application]
:: Spring Boot :: (v3.3.5)

2024-11-18T15:29:33.273+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.275+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.797+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.804+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.804+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.838+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:33.839+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:34.058+05:30 INFO 4164 --- [com.springboot.app] [
2024-11-18T15:29:34.063+05:30 INFO 4164 --- [com.springboot.app] [

main] com.springboot.app.SpringBootApplication : Starting SpringBootApplication using Java 17.0.
main] com.springboot.app.SpringBootApplication : No active profile set, falling back to
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
main] o.apache.catalina.core.StandardEngine : Starting service [Tomcat]
main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplication
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializati
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with
main] com.springboot.app.SpringBootApplication : Started SpringBootApplication in 0.995 seconds
```

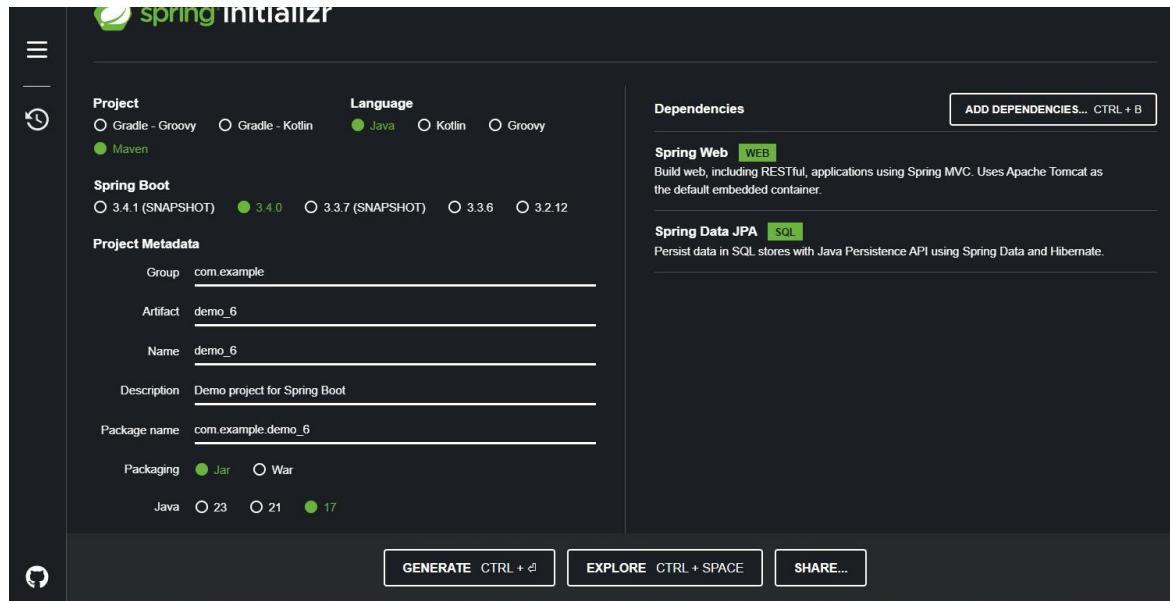
Navigate to <http://localhost:8080/welcome>



Program based on Assignment based Spring Boot and RESTful Web Services

Program no. 3 Write a program to demonstrate Database Connection with spring boot.

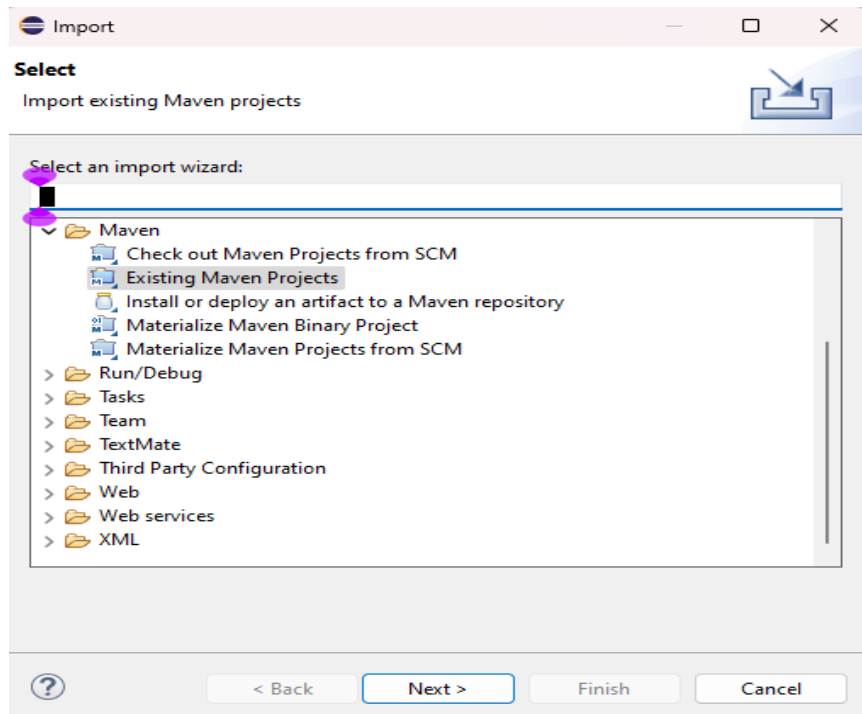
start.spring.io



The image shows the Spring Initializr web interface. It is a dark-themed form for generating a Spring project. The 'Project' section has radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', 'Java' (selected), 'Kotlin', and 'Groovy'. Below it, 'Maven' is selected. The 'Spring Boot' section has radio buttons for versions: '3.4.1 (SNAPSHOT)', '3.4.0' (selected), '3.3.7 (SNAPSHOT)', '3.3.6', and '3.2.12'. The 'Project Metadata' section contains text input fields for 'Group' (com.example), 'Artifact' (demo_6), 'Name' (demo_6), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo_6). There are also radio buttons for 'Packaging' ('Jar' selected, 'War' unselected) and 'Java' version ('23', '21', '17' selected). The 'Dependencies' section on the right has a button 'ADD DEPENDENCIES... CTRL + B' and two pre-selected dependencies: 'Spring Web' (WEB) and 'Spring Data JPA' (SQL). At the bottom are buttons for 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

On eclipse

File -> import-> Existing Maven Project



Roll no:39

Arif Shaikh

Program based on Assignment based Spring Boot and RESTful Web Services

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>3.4.0</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>demo_6</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>demo_6</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
  </scm>
  <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>
```

Program based on Assignment based Spring Boot and RESTful Web Services

```
        <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>
```

In src/main/resources

Application.properties

```
spring.application.name=demo_6
# PostgreSQL Database Configuration
spring.datasource.url=jdbc:postgresql://localhost:5433/postgres
spring.datasource.username=postgres
spring.datasource.password=abc
server.port=8081
spring.datasource.driver-class-name=org.postgresql.Driver
```

JPA Configuration

```
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Person.java

```
package com.example.demo_6;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
```

Program based on Assignment based Spring Boot and RESTful Web Services

```
@Entity
public class Person {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    // Getters and Setters
    public Long getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

Interface PersonRepository

package com.example.demo_6;
import org.springframework.data.jpa.repository.JpaRepository;
public interface PersonRepository extends JpaRepository<Person, Long> {
}
```

PersonController.java

```
package com.example.demo_6;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
public class PersonController {
```

Program based on Assignment based Spring Boot and RESTful Web Services

@Autowired

private PersonRepository personRepository;

@GetMapping("/persons")

```
public List<Person> getAllPersons() {  
    return personRepository.findAll();  
}
```

```
}
```

By default created

package com.example.demo_6;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

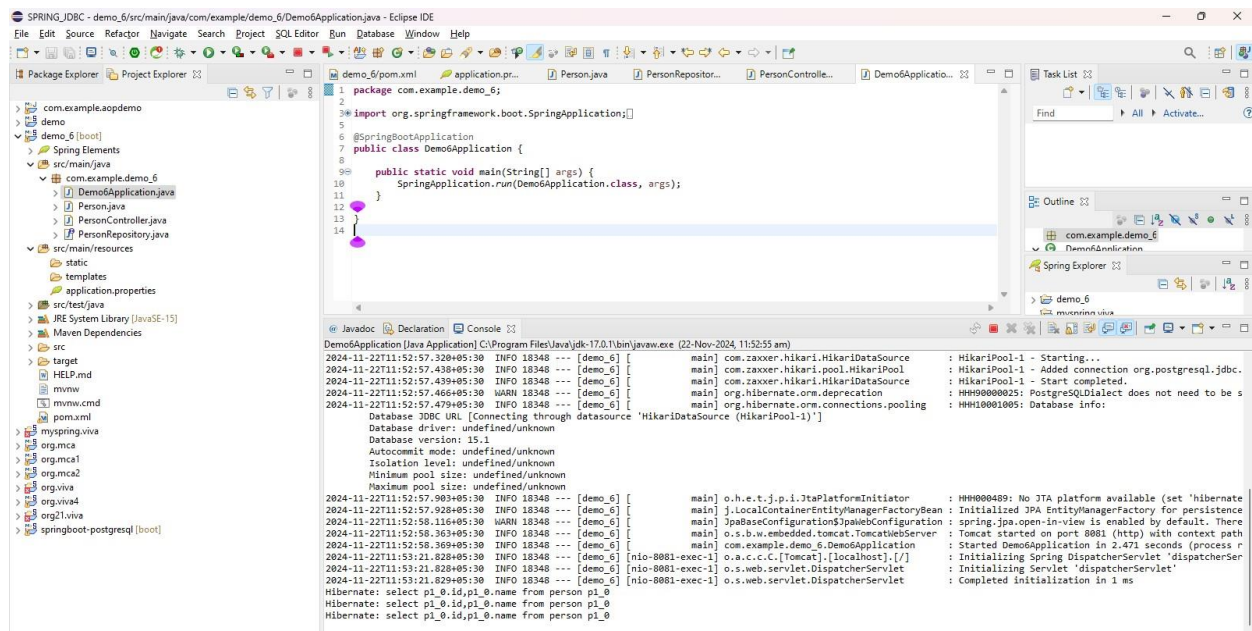
@SpringBootApplication

```
public class Demo6Application {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(Demo6Application.class, args);
```

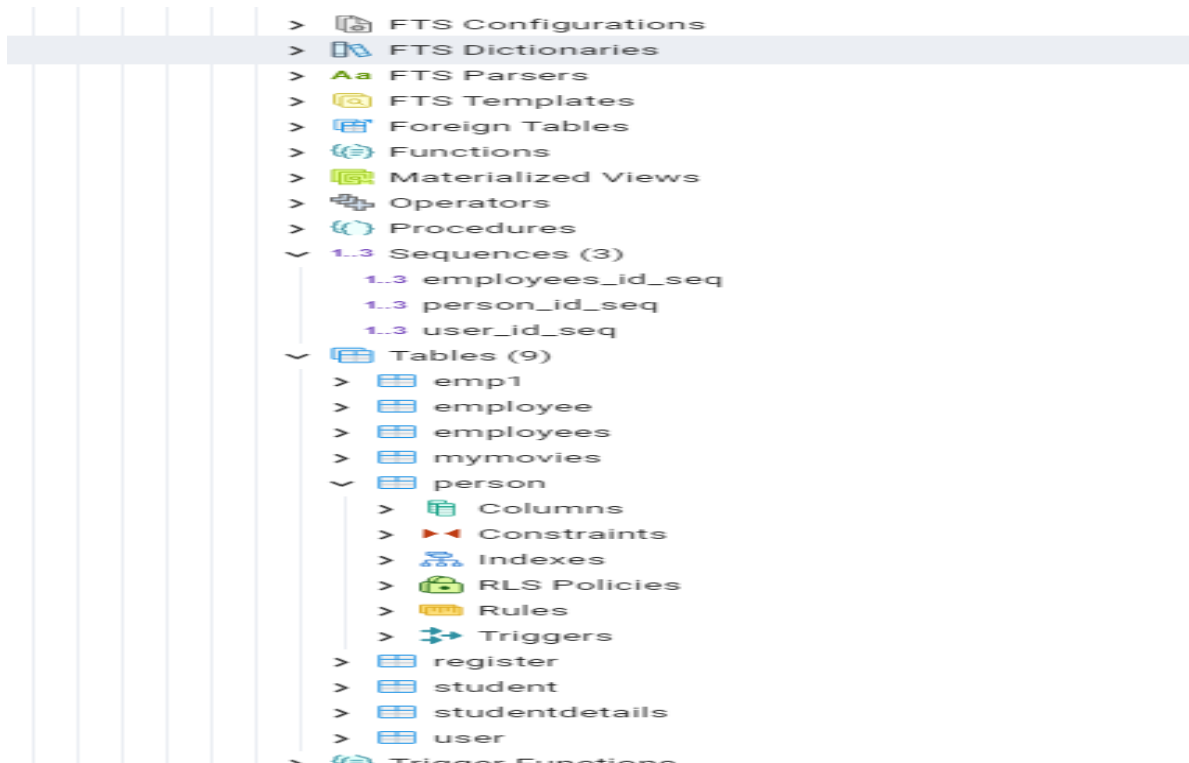
```
    } }
```



Program based on Assignment based Spring Boot and RESTful Web Services

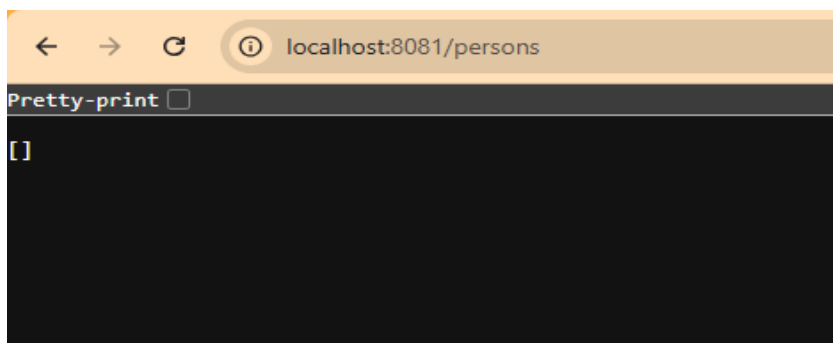
Run DemoApplication as JAVA Application.

Go to PgADMIN - > Run as Administrator -> postgresql 15-> databases->postgres-> Schemas-> tables -> check Person table is available.



On browser run

<http://localhost:8081/persons>



API is working Properly.

Program based on Assignment based Spring Boot and RESTful Web Services

Go back to PgADMIN and execute following Queries

```
INSERT INTO person (name) VALUES ('sonia');
```

```
INSERT INTO person (name) VALUES ('jasmine');
```

```
INSERT INTO person (name) VALUES ('John Doe');
```

```
INSERT INTO person (name) VALUES ('Jane Smith');
```

Now again go to browser and run following

Pretty-print ☒

```
[
  {
    "id": 1,
    "name": "sonia"
  },
  {
    "id": 2,
    "name": "jasmine"
  },
  {
    "id": 3,
    "name": "John Doe"
  },
  {
    "id": 4,
    "name": "Jane Smith"
  }
]
```