# Blood Donor Management System: RESTful API for Donor Information

# OVERVIEW

Build a simple Rest API application called Donors. This application manages blood donors information and allows the users to Add a new donor, update existing donor information, view existing donors and delete a donor information from the application.

# FUNCTIONAL REQUIREMENTS

* Build an application for getting student details.
* Have to create a folder named “Controller” inside the springapp in src.
* Inside the Controller create a java file named “ApiController.java”.
* Next have to create another folder inside the springapp named as

“Models”.

* Inside Models, you should create the java file named “Donor.java”.
* And create Repository folder and inside that create "DonorRepo.java".

**Create 5 variables**

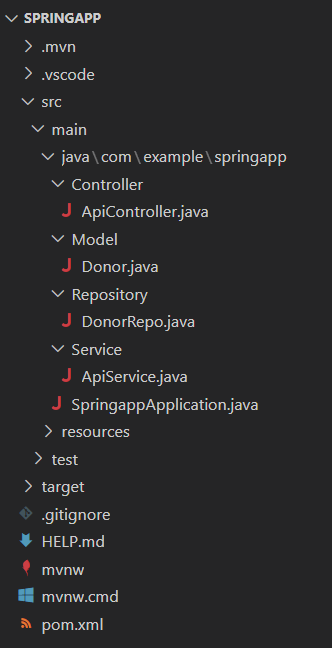
* Id - int
* Name- string
* Age – int
* Address – string
* Bloodgroup – String

as well as create getters and setters and constructors for the corresponding

variables. Finally, create the "Service" folder, and inside that create the

"ApiService.java" file.

The project structure looks like this image



**Core Platform**

OpenJDK 11

**API:**

POST - "/" --> true/false

GET - "/{id}" --> List of donor object GET - "/" --> List of donor object PUT - “/{id}” --> donor object DELETE - “/{id}” --> true/false

GET - "/donors/age/{age}" --> Returns a list of donors of a specific age

GET - "/donors/bloodGroup/{bloodGroup}" --> Returns a list of donors with a specific blood group.

GET - "/donors/ageRange" --> This endpoint accepts two request parameters: minAge and maxAge to define the age rangeand It returns a list of donors within a specified age range.

**Note:**

Copy and paste it into the **application.properties** file.

**spring.jpa.hibernate.ddl-auto=update**

**spring.datasource.url=jdbc:mysql://localhost/donor?createDatabaseIfNotExi st=true**

**spring.datasource.username=root spring.datasource.password=examly**

**spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver spring.jpa.show-sql= true**

**spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect**

**API endpoint:**

8080

**Platform Guidelines:**

To run the command use Terminal in the platform.

**Spring Boot:**

To start/run the application **'mvn spring-boot:run'**