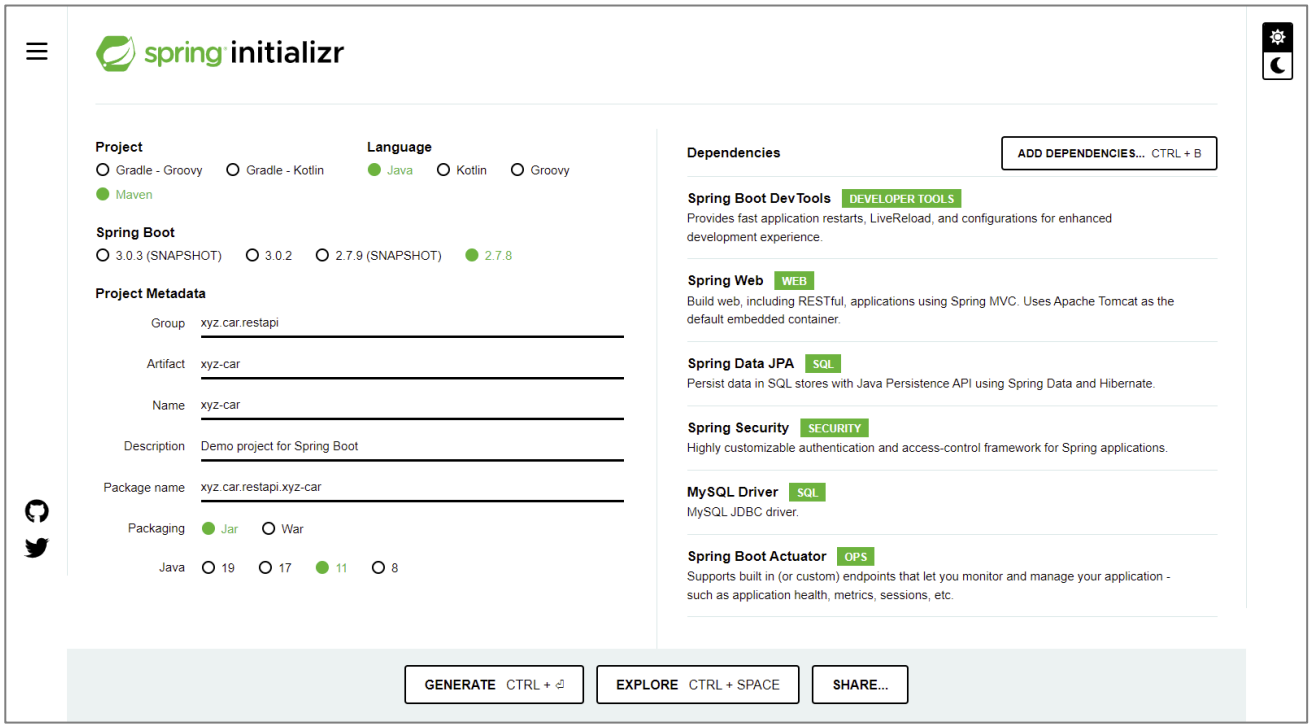
Assignment 1 **Chathushi Jayarathna**

Module No:	1	IU No:	1	Exercise No.	1

Learner Name :

Lab Assessment Statement	**Assignment 1 - Build an API** **Project Scenario,** The scope of this assignment is to develop an API for 'XYZ Cars Pte Ltd' as a website developer to develop a Used Car Sales portal. In this assignment, Users will be able to register in the portal using the Registration Page. Users of the portal can search for Cars using Make, Model, Registration & Price Range. Users will be able to view the Car information after searching them. The portal allow users to login, post Car for sale. These are the steps provided to build an API. 1. Create SpringBoot application Spring Initializr. Import the project in your IDE. 2. Create entities for your backend as per the scenario. 3. Create Repositories using JPA repositories. 4. Create service layer to access the data in your API 5. Create controller for your application using REST API 6. Test your API using POSTMAN tool. **Provide the source code of developed application.** **Provide screen capture of final result pages for API testing.**
Technical Environment	-JAVA, JPA, REST API
Guidelines	-
Duration	120 mins

1. Create SpringBoot application Spring Initializr. Import the project in your IDE.



2. Create entities for your backend as per the scenario.

UserAccount.java

```
package xyz.cars.restapi.entity;

import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;
import javax.validation.constraints.NotBlank;
import javax.validation.constraints.Size;

import com.fasterxml.jackson.annotation.JsonIgnore;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tb_user", uniqueConstraints = { @UniqueConstraint(columnNames =
"username") })
public class UserAccount {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id_user")
  private int idUser;

  @Column(nullable = false)
  @NotBlank(message = "Username is required")
  private String username;

  @Column(nullable = false)
  @Size(min = 6, message = "Password must be at least 6 characters long")
  private String password;

  @JsonIgnore
  @OneToMany(mappedBy = "user", cascade = CascadeType.ALL)
  private List<Car> cars;

public String getPassword() {
  // TODO Auto-generated method stub
  return null;
}

public String getUsername() {
  // TODO Auto-generated method stub
  return null;
}

public void setUsername(String username2) {
  // TODO Auto-generated method stub

}

public void setPassword(String encode) {
  // TODO Auto-generated method stub

}

}
```

Car.java

```java
package xyz.cars.restapi.entity;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.validation.constraints.Positive;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tb_car")
public class Car {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id_car")
  private int idCar;

  @Column(nullable = false)
  private String make;

  @Column(nullable = false)
  private String model;

  @Column(nullable = false)
  private String year;

  @Column(nullable = false)
  private int price;

  @ManyToOne(cascade = CascadeType.ALL)
  @JoinColumn(name = "id_user", nullable = false)
  private UserAccount user;

public int getIdCar() {
  // TODO Auto-generated method stub
  return 0;
}

public String getMake() {
  // TODO Auto-generated method stub
  return null;
}

public String getModel() {
  // TODO Auto-generated method stub
  return null;
}

public @Positive(message = "Price can't below 0 or Negative number") int getPrice() {
  // TODO Auto-generated method stub
  return 0;
}

public void setMake(Object make2) {
  // TODO Auto-generated method stub

}

public void setModel(Object model2) {
  // TODO Auto-generated method stub

}

public void setYear(Object year2) {
  // TODO Auto-generated method stub

}

public void setPrice(Object price2) {
  // TODO Auto-generated method stub

}

public void setUser(UserAccount user2) {
  // TODO Auto-generated method stub

}

}
```

3. Create Repositories using JPA repositories.

CarRepository.Java

```
package xyz.cars.restapi.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import xyz.cars.restapi.entity.Car;

public interface CarRepository extends JpaRepository<Car, Integer> {
  @Query(value = "SELECT * FROM tb_car "
      + "WHERE make LIKE '%' :keyword '%' "
      + "OR model LIKE '%' :keyword '%' "
      + "OR year LIKE '%' :keyword '%' ", nativeQuery = true)
  List<Car> searchCarByKeyword(@Param("keyword") String keyword);

  @Query(value = "SELECT * FROM tb_car "
      + "WHERE price >= :min AND price <= :max", nativeQuery = true)
  List<Car> searchCarByPriceRange(@Param("min") int min, @Param("max") int max);
}
```

UserRepository.Java

```
package xyz.cars.restapi.repository;

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

import xyz.cars.restapi.entity.UserAccount;

public interface UserAccountRepository extends JpaRepository<UserAccount, Integer> {

  public UserAccount findByUsername(String username);
}
```

4. Create service layer to access the data in your API

CarService.Java

```
package xyz.cars.restapi.service;

import java.util.List;

import xyz.cars.restapi.entity.Car;
import xyz.cars.restapi.models.CarDto;

public interface CarService {

  public List<Car> listCar() throws Exception;

  public Car getCarById(int idCar) throws Exception;

  public Car saveCarPost(CarDto carDto) throws Exception;

  public List<Car> searchByKeyword(String keyword) throws Exception;

  public List<Car> searchByPriceRange(int min, int max) throws Exception;
}
```

CarService.Java

```
package xyz.cars.restapi.service;

import java.util.List;

import xyz.cars.restapi.entity.UserAccount;

public interface UserService {

  List<UserAccount> listUser() throws Exception;

  UserAccount getUserById(int idUser) throws Exception;

  UserAccount addUser(UserAccount user) throws Exception;
}
```

CarServiceImpl.Java

```java
package xyz.cars.restapi.service;

import java.util.List;

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

import xyz.cars.restapi.entity.Car;
import xyz.cars.restapi.entity.UserAccount;
import xyz.cars.restapi.models.CarDto;
import xyz.cars.restapi.repository.CarRepository;

@Service
public class CarServiceImpl implements CarService {

  @Autowired
  private CarRepository carRepo;
  @Autowired
  private UserService userService;

  @Override
  public Car getCarById(int idCar) throws Exception {
    Car car = carRepo.findById(idCar).get();
    return car;
  }

  @Override
  public List<Car> listCar() throws Exception {
    List<Car> listCar = carRepo.findAll();
    return listCar;
  }

  @Override
  public Car saveCarPost(CarDto carDto) throws Exception {
    Car newCar = new Car();
    UserAccount user = userService.getUserById(carDto.getIdUser());

    newCar.setMake(carDto.getMake());
    newCar.setModel(carDto.getModel());
    newCar.setYear(carDto.getYear());
    newCar.setPrice(carDto.getPrice());
    newCar.setUser(user);

    carRepo.save(newCar);

    return newCar;
  }

  @Override
  public List<Car> searchByKeyword(String keyword) throws Exception {
    List<Car> listCar = carRepo.searchCarByKeyword(keyword);

    return listCar;
  }

  @Override
  public List<Car> searchByPriceRange(int min, int max) throws Exception {
    List<Car> listCar = carRepo.searchCarByPriceRange(min, max);

    return listCar;
  }

}
```

UserServiceImpl.Java

```
@Service
public class UserServiceImpl implements UserService {

  @Autowired
  private UserAccountRepository userRepo;

  @Autowired
  private PasswordEncoder passwordEncoder;

  @Override
  public UserAccount getUserById(int idUser) {
    UserAccount user = userRepo.findById(idUser).get();
    return user;
  }

  @Override
  public List<UserAccount> listUser() {
    List<UserAccount> listUser = userRepo.findAll();
    return listUser;
  }

  @Override
  public UserAccount addUser(UserAccount user) throws Exception {
    UserAccount newUser = new UserAccount();

    newUser.setUsername(user.getUsername());
    newUser.setPassword(passwordEncoder.encode(user.getPassword()));
    userRepo.save(newUser);

    return newUser;
  }
}
```

5. Create controller for your application using REST API

CarController.Java

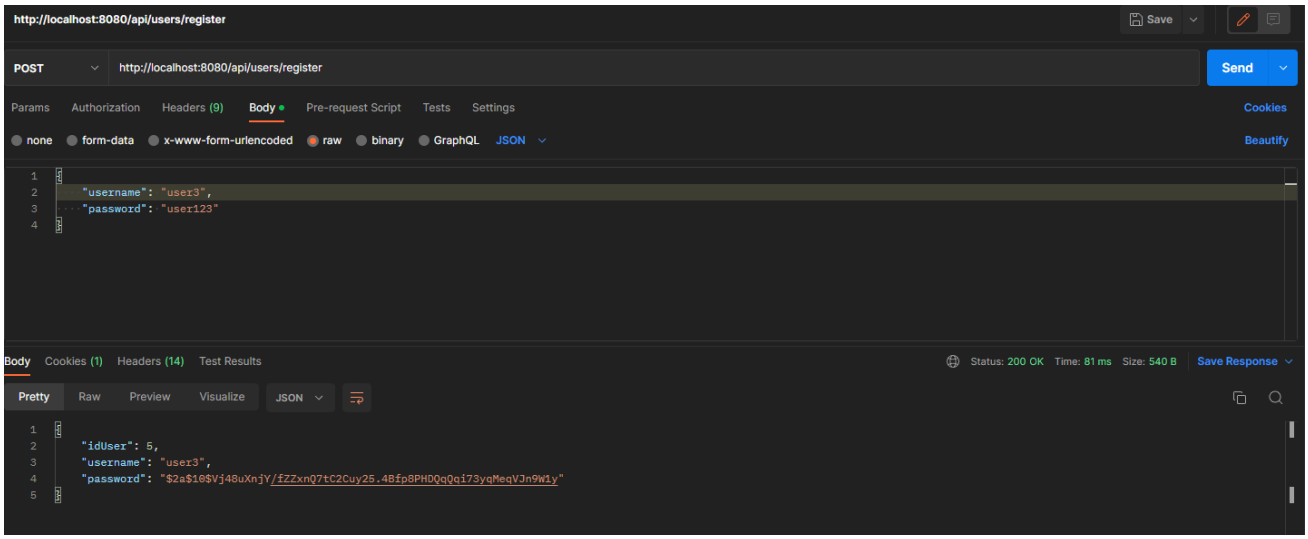
```
package xyz.cars.restapi.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.ui.Model;
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.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import xyz.cars.restapi.entity.Car;
import xyz.cars.restapi.models.CarDto;
import xyz.cars.restapi.service.CarService;

@RestController
@RequestMapping("/api/cars")
public class CarController {

  @Autowired
  private CarService carService;

  // List Car
  @GetMapping("")
  public List<Car> listCar() throws Exception {
    List<Car> listCar = carService.listCar();
    return listCar;
  }

  // Car Detail
  @GetMapping("/{idCar}")
  public Car carDetail(@PathVariable("idCar") int idCar) throws Exception {
    Car carDetail = carService.getCarById(idCar);
    return carDetail;
  }

  // Post Car
  @PostMapping("/post")
  public Car postCar(@RequestBody CarDto carDto) throws Exception {
    Car newCar = carService.saveCarPost(carDto);
    return newCar;
  }

  // Search By Keyword
  @GetMapping(value = "", params = "keyword")
  public List<Car> searchCar(@RequestParam("keyword") String keyword, Model model)
throws Exception {
    List<Car> searchCar = carService.searchByKeyword(keyword);
    return searchCar;
  }

  // Search By Price Range
  @GetMapping(value = "", params = { "min", "max" })
  public List<Car> searchCarByPriceRange(@RequestParam("min") int min,
@RequestParam("max") int max, Model model)
      throws Exception {
```

UserController.java

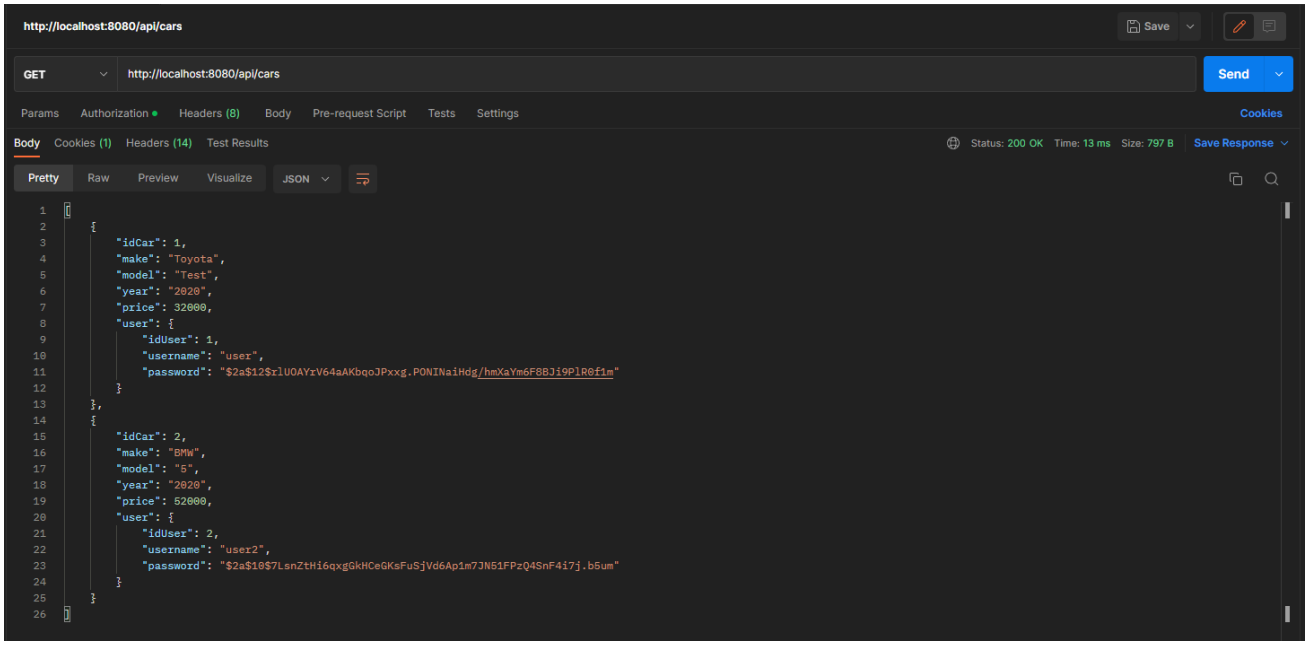
```
package xyz.cars.restapi.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
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.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import xyz.cars.restapi.entity.UserAccount;
import xyz.cars.restapi.service.UserService;

@RestController
@RequestMapping("/api/users")
public class UserController {

  @Autowired
  private UserService userService;

  @GetMapping("/")
  public List<UserAccount> listUser() throws Exception {
    List<UserAccount> listUser = userService.listUser();
    return listUser;
  }

  @GetMapping("/{idUser}")
  public UserAccount getUser(@PathVariable("idUser") int idUser) throws Exception {
    UserAccount user = userService.getUserById(idUser);
    return user;
  }

  @PostMapping("/register")
  public UserAccount addUser(@RequestBody UserAccount user) throws Exception {
    UserAccount newUser = userService.addUser(user);
    return newUser;
  }
}
```

6. Test your API using the POSTMAN tool.

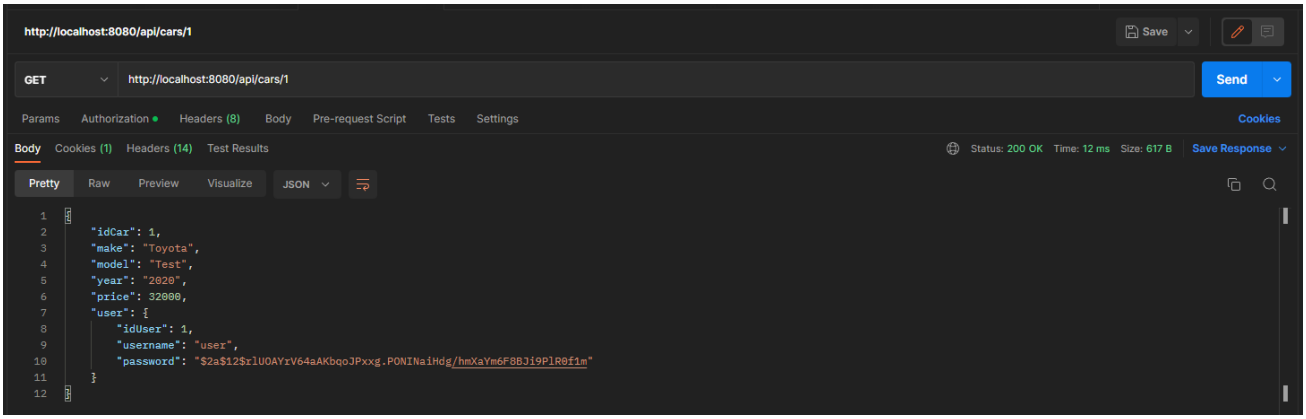
Register



Car List



Car Detail



Assignment 1 **Chathushi Jayarathna**

Post Car