Admin Module

Admin

package com.Springbootpace.SpringBoot.Entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String email;

private String password;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

// Getters and setters

}

AdminRepository

package com.Springbootpace.SpringBoot.Repositary;

import java.util.Optional;

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

import com.Springbootpace.SpringBoot.Entity.Admin;

public interface AdminRepository extends JpaRepository<Admin, Long> {

Optional<Admin> findByEmail(String email);

}

AdminService

package com.Springbootpace.SpringBoot.Service;

import com.Springbootpace.SpringBoot.Entity.Admin;

public interface AdminService {

boolean validateLogin(String email, String password);

Admin findByEmail(String email);

}

AdminServiceImpl

package com.Springbootpace.SpringBoot.Service;

import java.util.Optional;

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

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.stereotype.Service;

import com.Springbootpace.SpringBoot.Entity.Admin;

import com.Springbootpace.SpringBoot.Repositary.AdminRepository;

@Service

public class AdminServiceImpl implements AdminService {

@Autowired

private AdminRepository adminRepository;

@Autowired

private PasswordEncoder passwordEncoder;

@Override

public boolean validateLogin(String email, String password) {

Optional<Admin> admin = adminRepository.findByEmail(email);

return admin.isPresent() && passwordEncoder.matches(password, admin.get().getPassword());

}

@Override

public Admin findByEmail(String email) {

return adminRepository.findByEmail(email).orElse(null);

}

}

AdminController

package com.Springbootpace.SpringBoot.Controller;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.web.bind.annotation.\*;

import com.Springbootpace.SpringBoot.Entity.\*;

import com.Springbootpace.SpringBoot.Service.\*;

import com.Springbootpace.SpringBoot.dto.LoginRequest;

@RestController

@RequestMapping("/api/admin")

@CrossOrigin(origins = "\*")

public class AdminController {

@Autowired

private AdminService adminService;

@Autowired

private HotelBookingService hotelService;

@Autowired

private CabBookingService cabService;

@Autowired

private FlightBookingService flightService;

@Autowired

private TouristBookingService touristService;

@Autowired

private UserService userService;

@Autowired

private ContactMessageService contactMessageService;

@Autowired

private PasswordEncoder passwordEncoder;

@PostMapping("/login")

public ResponseEntity<Map<String, String>> login(@RequestBody LoginRequest request) {

Admin admin = adminService.findByEmail(request.getEmail());

Map<String, String> response = new HashMap<>();

if (admin != null && passwordEncoder.matches(request.getPassword(), admin.getPassword())) {

response.put("message", "Login Successful");

response.put("role", "ADMIN");

response.put("email", admin.getEmail());

return ResponseEntity.ok(response);

} else {

response.put("message", "Invalid credentials");

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body(response);

}

}

*@GetMapping*("/summary")

public ResponseEntity<Map<String, Long>> getBookingSummary() {

Map<String, Long> summary = new HashMap<>();

summary.put("hotels", hotelService.getAllBookings().stream().count());

summary.put("flights", flightService.getAllBookings().stream().count());

summary.put("cabs", cabService.getAllBookings().stream().count());

summary.put("touristPackages", touristService.getAllBookings().stream().count());

summary.put("users", userService.getAllUsers().stream().count());

return ResponseEntity.*ok*(summary);

}

@GetMapping("/users")

public List<User> getAllUsers() {

return userService.getAllUsers();

}

@GetMapping("/hotels")

public List<HotelBooking> getAllHotelBookings() {

return hotelService.getAllBookings();

}

@GetMapping("/cabs")

public List<CabBooking> getAllCabBookings() {

return cabService.getAllBookings();

}

@GetMapping("/flights")

public List<FlightBooking> getAllFlightBookings() {

return flightService.getAllBookings();

}

@GetMapping("/tourist")

public List<TouristBooking> getAllTouristBookings() {

return touristService.getAllBookings();

}

@GetMapping("/contact")

public List<ContactMessage> getAllContactMessages() {

return contactMessageService.getAllMessages();

}

}