

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\Backe**

```
package com.physio.backend;

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

@SpringBootApplication
public class BackendApplication {

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

}
```

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\config**

```
package com.physio.backend.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .csrf(csrf -> csrf.disable())
            .authorizeHttpRequests(auth -> auth
                .requestMatchers(
                    "/api/auth/**",
                    "/api/users/**",
                    "/api/services/**",
                    "/api/blogs/**",
                    "/api/appointments/**",
                    "/api/patient-journey/**",
                    "/api/activityLogs/**",
                    "/api/securityLogs/**",
                    "/api/passwordChangeRequests/**",
                    "/api/patients/**"
                )
                .permitAll()
                .anyRequest().authenticated()
            )
            .httpBasic(); // Optional

        return http.build();
    }
}
```

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;

import com.physio.backend.model.ActivityLog;
import com.physio.backend.service.ActivityLogService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/activityLogs")
@CrossOrigin(origins = "**")
public class ActivityLogController {

    @Autowired
    private ActivityLogService service;

    @GetMapping
```

```

    public List<ActivityLog> getAll() {
        return service.getAll();
    }

    @PostMapping
    public ActivityLog create(@RequestBody ActivityLog log) {
        log.setTimestamp( java.time.LocalDateTime.now());
        return service.create(log);
    }

    @DeleteMapping("/{id}")
    public void delete(@PathVariable Long id) {
        service.delete(id);
    }
}

```

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```

import com.physio.backend.model.Appointment;
import com.physio.backend.service.AppointmentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/appointments")
@CrossOrigin(origins = "**")
public class AppointmentController {

    @Autowired
    private AppointmentService service;

    @GetMapping
    public List<Appointment> getAll() {
        return service.getAll();
    }

    @GetMapping("/role/{role}")
    public List<Appointment> getByRole(@PathVariable String role) {
        return service.getByRole(role.toLowerCase());
    }

    @PostMapping
    public Appointment create(@RequestBody Appointment appointment) {
        return service.create(appointment);
    }

    @PutMapping("/{id}")
    public Appointment update(@PathVariable Long id, @RequestBody Appointment appointment) {
        return service.update(id, appointment);
    }

    @DeleteMapping("/{id}")
    public void delete(@PathVariable Long id) {
        service.delete(id);
    }

    @GetMapping("/role/{role}/completed")
    public List<Appointment> getCompletedAppointments(@PathVariable String role) {
        return service.getCompletedByRole(role.toLowerCase());
    }
}

```

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```

import com.physio.backend.model.User;
import com.physio.backend.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

```

```

import java.util.Map;

@RestController
@RequestMapping("/api/auth")
@CrossOrigin(origins = "**") // Allow frontend access
public class AuthController {

    @Autowired
    private UserService userService;

    @PostMapping("/login")
    public Object login(@RequestBody Map<String, String> credentials) {
        String email = credentials.get("email");
        String password = credentials.get("password");

        return userService.authenticate(email, password)
            .map(user -> Map.of(
                "id", user.getId(),
                "name", user.getName(),
                "email", user.getEmail(),
                "role", user.getRole()
            ))
            .orElseGet(() -> Map.of("error", "Invalid credentials"));
    }
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```

package com.physio.backend.controller;

import com.physio.backend.model.Blog;
import com.physio.backend.service.BlogService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/blogs")
@CrossOrigin(origins = "**")
public class BlogController {

    @Autowired
    private BlogService blogService;

    @GetMapping
    public List<Blog> getAll() {
        return blogService.getAll();
    }

    @PostMapping
    public Blog create(@RequestBody Blog blog) {
        return blogService.create(blog);
    }

    @PutMapping("/{id}")
    public Blog update(@PathVariable Long id, @RequestBody Blog blog) {
        return blogService.update(id, blog);
    }

    @DeleteMapping("/{id}")
    public void delete(@PathVariable Long id) {
        blogService.delete(id);
    }
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```

package com.physio.backend.controller;

import java.time.LocalDateTime;
import java.util.List;
import java.util.Map;

```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PatchMapping;
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 com.physio.backend.model.PasswordChangeRequest;
import com.physio.backend.service.PasswordChangeRequestService;

@RestController
@RequestMapping("/api/passwordChangeRequests")
@CrossOrigin("*")
public class PasswordChangeRequestController {

    @Autowired
    private PasswordChangeRequestService service;

    @GetMapping
    public List<PasswordChangeRequest> getAll() {
        return service.getAll();
    }

    @PostMapping
    public PasswordChangeRequest create(@RequestBody PasswordChangeRequest req) {
        req.setStatus("Pending");
        req.setRequestedAt(LocalDateTime.now());
        return service.save(req);
    }

    @PatchMapping("/{id}")
    public PasswordChangeRequest update(@PathVariable Long id, @RequestBody Map<String, String> body) {
        return service.updateStatus(id, body.get("status"));
    }
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\controller**  
package com.physio.backend.controller;

```

import com.physio.backend.model.Patient;
import com.physio.backend.service.PatientService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/api/patients")
@CrossOrigin(origins = "*")
public class PatientController {

    @Autowired
    private PatientService service;

    @GetMapping("/withReports")
    public List<Patient> getWithReports() {
        return service.getAllWithReports();
    }

    @PostMapping
    public Patient create(@RequestBody Patient p) {
        return service.create(p);
    }

    @GetMapping
    public List<Patient> getAll() {
        return service.getAll();
    }
}

```

```

// ? NEW: Fetch patient by MRN
@GetMapping(params = "mrn")
public ResponseEntity<Patient> getByMrn(@RequestParam String mrn) {
    return service.getByMrn(mrn)
        .map(ResponseEntity::ok)
        .orElse(ResponseEntity.notFound().build());
}

// ? NEW: Patch patient (assignments, charts)
@PatchMapping("/{id}")
public Patient patchPatient(@PathVariable Long id, @RequestBody Map<String, Object> updates) {
    return service.patch(id, updates);
}
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```
import com.physio.backend.dto.PatientJourneyDTO;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
```

```
import java.util.HashMap;
import java.util.Map;
```

```
@RestController
@RequestMapping("/api/patient-journey")
@CrossOrigin(origins = "**")
public class PatientJourneyController {
```

```

    @GetMapping("/{mrn}")
    public ResponseEntity<?> getPatientJourney(@PathVariable String mrn) {
        // Mock DB logic ? Replace this with real DB queries later
        if (!mrn.equalsIgnoreCase("MRN123456")) {
            return ResponseEntity.notFound().build();
        }

        PatientJourneyDTO dto = new PatientJourneyDTO();
        dto.setMrn("MRN123456");
        dto.setName("John Doe");

        Map<String, Boolean> statusMap = new HashMap<>();
        statusMap.put("registration", true);
        statusMap.put("consultation", true);
        statusMap.put("labReport", true);
        statusMap.put("prescription", false);
        dto.setStatus(statusMap);

        dto.setLabReportUrl("/pdfs/MRN123456_labreport.pdf");
        dto.setPrescriptionUrl("/prescriptions/MRN123456_prescription.pdf");

        return ResponseEntity.ok(dto);
    }
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```
import com.physio.backend.model.SecurityLog;
import com.physio.backend.service.SecurityLogService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
```

```
import java.time.LocalDateTime;
import java.util.List;
```

```
@RestController
@RequestMapping("/api/securityLogs")
@CrossOrigin(origins = "**")
public class SecurityLogController {
```

```
    @Autowired
```

```

private SecurityLogService service;

@GetMapping
public List<SecurityLog> getAll() {
    return service.getAll();
}

@PostMapping
public SecurityLog create(@RequestBody SecurityLog log) {
    log.setTimestamp(LocalDateTime.now());
    return service.save(log);
}
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```

import com.physio.backend.model.Service;
import com.physio.backend.service.ServiceService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

```

```
import java.util.List;
```

```

@RestController
@RequestMapping("/api/services")
@CrossOrigin(origins = "**")
public class ServiceController {

    @Autowired
    private ServiceService serviceService;

    @GetMapping
    public List<Service> getAll() {
        return serviceService.getAll();
    }

    @PostMapping
    public Service addService(@RequestBody Service service) {
        return serviceService.save(service);
    }

    @PutMapping("/{id}")
    public Service updateService(@PathVariable Long id, @RequestBody Service service) {
        return serviceService.update(id, service);
    }

    @DeleteMapping("/{id}")
    public void deleteService(@PathVariable Long id) {
        serviceService.delete(id);
    }
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\contr**

```
package com.physio.backend.controller;
```

```

import com.physio.backend.model.User;
import com.physio.backend.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

```

```
import java.util.*;
```

```

@RestController
@RequestMapping("/api/users")
@CrossOrigin(origins = "**") // Allow frontend access
public class UserController {

    @Autowired
    private UserService userService;

    @GetMapping
    public List<User> getAllUsers() {

```

```

        return userService.getAll();
    }

    @PostMapping
    public User createUser(@RequestBody User user) {
        return userService.create(user);
    }

    @PutMapping("/{id}")
    public User updateUser(@PathVariable Long id, @RequestBody User updatedUser) {
        return userService.update(id, updatedUser);
    }

    @DeleteMapping("/{id}")
    public void deleteUser(@PathVariable Long id) {
        userService.delete(id);
    }

    @PatchMapping("/{id}")
    public User patchUser(@PathVariable Long id, @RequestBody Map<String, Object> updates) {
        return userService.patch(id, updates);
    }
}

```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\dto\PatientJourneyDTO.java**

```
package com.physio.backend.dto;
```

```
import java.util.Map;
```

```

public class PatientJourneyDTO {
    private String mrn;
    private String name;
    private Map<String, Boolean> status; // e.g., registration, consultation, labReport, prescription
    private String labReportUrl;
    private String prescriptionUrl;

    // Getters and Setters
    public String getMrn() {
        return mrn;
    }
    public void setMrn(String mrn) {
        this.mrn = mrn;
    }

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

    public Map<String, Boolean> getStatus() {
        return status;
    }
    public void setStatus(Map<String, Boolean> status) {
        this.status = status;
    }

    public String getLabReportUrl() {
        return labReportUrl;
    }
    public void setLabReportUrl(String labReportUrl) {
        this.labReportUrl = labReportUrl;
    }

    public String getPrescriptionUrl() {
        return prescriptionUrl;
    }
    public void setPrescriptionUrl(String prescriptionUrl) {
        this.prescriptionUrl = prescriptionUrl;
    }
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\mode**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDateTime;
```

```
@Entity
@Table(name = "activity_logs")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class ActivityLog {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private LocalDateTime timestamp;
    private String actor;
    private String mrn;
    private String action;
}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\mode**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDate;
import java.time.LocalTime;
```

```
@Entity
@Table(name = "appointments")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Appointment {

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

    private String patientName;
    private String mrn;
    private String appointmentType; // "Online" or "In-Person"
    private LocalDate date;
    private LocalTime time;
    private String notes;
    private String status; // "Pending", "Approved", "Completed"
    private String meetingLink; // only for online

    private String role; // e.g., "doctor", "counselor"
}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\mode**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDateTime;
```

```
@Entity
@Table(name = "blogs")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Blog {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```



```

    private Long id;

    private String title;
    private String smallDesc;

    @Column(length = 5000)
    private String longDesc;

    private String banner;

    private LocalDateTime date;
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\model**

```

package com.physio.backend.model;
import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDateTime;
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class PasswordChangeRequest {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String counselorName;
    private String email;
    private String reason;
    private String status;
    private LocalDateTime requestedAt;
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\model**

```

package com.physio.backend.model;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDate;
import java.time.LocalTime;
import java.util.Map;

@Entity
@Table(name = "patients")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Patient {

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

    private String name;
    private String mrn;

    @Column(name = "patient_condition")
    private String condition;

    private String reportPdfUrl;

    // ? Therapy/Diet/Nutrition chart mappings
    @ElementCollection
    @CollectionTable(name = "patient_charts", joinColumns = @JoinColumn(name = "patient_id"))
    @MapKeyColumn(name = "chart_index")
    @Column(name = "chart_data")
    private Map<String, String> charts;

    // ? Role assignments (physio/diet/nutrition)
    @ElementCollection

```

```

@CollectionTable(name = "patient_assignments", joinColumns = @JoinColumn(name = "patient_id"))
@MapKeyColumn(name = "role")
@Column(name = "assigned")
private Map<String, Boolean> assignedTo;

// ? Doctor's last visit data
private LocalDate lastVisitDate;
private LocalTime lastVisitTime;
private String summary;
private String doctorNotes;
private String recommendation;
private String prescription; // PDF URL (prescription link)

// ? Vitals stored as a key-value map
@ElementCollection
@CollectionTable(name = "patient_vitals", joinColumns = @JoinColumn(name = "patient_id"))
@MapKeyColumn(name = "vital_name")
@Column(name = "vital_value")
private Map<String, String> vitals;
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\model**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
```

```
import java.time.LocalDateTime;
```

```

@Entity
@Table(name = "security_logs")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class SecurityLog {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Long userId;
    private String action;
    private String actor;
    private LocalDateTime timestamp;
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\model**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
```

```
import java.util.List;
```

```

@Entity
@Table(name = "services")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Service {

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

    private String title;
    private String subtitle;
    private String banner;
    private String priceMonthly;
    private String priceQuarterly;
    private String priceYearly;
    private String rating;
}

```

```

        @Column(length = 2000)
        private String description;

        private String previewVideo;

        @ElementCollection
        private List<String> features;
    }

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\model**

```
package com.physio.backend.model;
```

```
import jakarta.persistence.*;
import lombok.*;
```

```

@Entity
@Table(name = "users")
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String email;
    private String password;
    private String role; // e.g., physio, counselor, etc.
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;
```

```
import com.physio.backend.model.ActivityLog;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface ActivityLogRepository extends JpaRepository<ActivityLog, Long> {
}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;
```

```
import com.physio.backend.model.Appointment;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import java.util.List;
```

```

public interface AppointmentRepository extends JpaRepository<Appointment, Long> {
    List<Appointment> findByRoleAndStatusIgnoreCase(String role, String status);

    List<Appointment> findByRole(String role);
}

```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;
```

```
import com.physio.backend.model.Blog;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface BlogRepository extends JpaRepository<Blog, Long> {
}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;
```

```
import com.physio.backend.model.PasswordChangeRequest;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface PasswordChangeRequestRepository extends JpaRepository<PasswordChangeRequest, Long> {
}
```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;

import com.physio.backend.model.Patient;
import org.springframework.data.jpa.repository.JpaRepository;

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

public interface PatientRepository extends JpaRepository<Patient, Long> {
    List<Patient> findByReportPdfUrlIsNotNull();
    Optional<Patient> findByMrn(String mrn);
}
```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;

import com.physio.backend.model.SecurityLog;
import org.springframework.data.jpa.repository.JpaRepository;

public interface SecurityLogRepository extends JpaRepository<SecurityLog, Long> {
}
```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;

import com.physio.backend.model.Service;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ServiceRepository extends JpaRepository<Service, Long> {
}
```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\repos**

```
package com.physio.backend.repository;

import com.physio.backend.model.User;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.Optional;

public interface UserRepository extends JpaRepository<User, Long> {
    Optional<User> findByEmail(String email);
}
```

**File: C:\Users\hnp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\servic**

```
package com.physio.backend.service;

import com.physio.backend.model.ActivityLog;
import com.physio.backend.repository.ActivityLogRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ActivityLogService {

    @Autowired
    private ActivityLogRepository repo;

    public List<ActivityLog> getAll() {
        return repo.findAll();
    }

    public ActivityLog create(ActivityLog log) {
        return repo.save(log);
    }

    public void delete(Long id) {
        repo.deleteById(id);
    }
}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```
package com.physio.backend.service;

import com.physio.backend.model.Appointment;
import com.physio.backend.repository.AppointmentRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class AppointmentService {

    @Autowired
    private AppointmentRepository repo;

    public List<Appointment> getAll() {
        return repo.findAll();
    }

    public List<Appointment> getByRole(String role) {
        return repo.findByRole(role);
    }

    public Appointment getById(Long id) {
        return repo.findById(id).orElseThrow();
    }

    public Appointment create(Appointment a) {
        return repo.save(a);
    }

    public Appointment update(Long id, Appointment updated) {
        updated.setId(id);
        return repo.save(updated);
    }

    public void delete(Long id) {
        repo.deleteById(id);
    }

    public List<Appointment> getCompletedByRole(String role) {
        return repo.findByRoleAndStatusIgnoreCase(role, "Completed");
    }

}
```

**File: C:\Users\hpn\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```
package com.physio.backend.service;

import com.physio.backend.model.Blog;
import com.physio.backend.repository.BlogRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.time.LocalDateTime;
import java.util.List;

@Service
public class BlogService {

    @Autowired
    private BlogRepository repo;

    public List<Blog> getAll() {
        return repo.findAll();
    }

    public Blog getById(Long id) {
        return repo.findById(id).orElseThrow();
    }

    public Blog create(Blog blog) {
        blog.setDate(LocalDateTime.now());
    }

}
```

```

        return repo.save(blog);
    }

    public Blog update(Long id, Blog updated) {
        updated.setId(id);
        return repo.save(updated);
    }

    public void delete(Long id) {
        repo.deleteById(id);
    }
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```
package com.physio.backend.service;
```

```
import com.physio.backend.model.PasswordChangeRequest;
import com.physio.backend.repository.PasswordChangeRequestRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.List;
```

```
@Service
```

```
public class PasswordChangeRequestService {
```

```
    @Autowired
```

```
    private PasswordChangeRequestRepository repo;
```

```
    public List<PasswordChangeRequest> getAll() {
        return repo.findAll();
    }
```

```
    public PasswordChangeRequest save(PasswordChangeRequest req) {
        return repo.save(req);
    }
```

```
    public PasswordChangeRequest updateStatus(Long id, String newStatus) {
        PasswordChangeRequest req = repo.findById(id).orElseThrow();
        req.setStatus(newStatus);
        return repo.save(req);
    }
}
```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```
package com.physio.backend.service;
```

```
import com.physio.backend.model.Patient;
import com.physio.backend.repository.PatientRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.*;
```

```
import java.time.LocalDate;
```

```
import java.time.LocalTime;
```

```
@Service
```

```
public class PatientService {
```

```
    @Autowired
```

```
    private PatientRepository repo;
```

```
    public List<Patient> getAllWithReports() {
        return repo.findByReportPdfUrlIsNotNull();
    }
```

```
    public Patient create(Patient p) {
        return repo.save(p);
    }
```

```
    public List<Patient> getAll() {
        return repo.findAll();
    }
}
```

```

// ? NEW: Get patient by MRN
public Optional<Patient> getByMrn(String mrn) {
    return repo.findByMrn(mrn);
}

// ? NEW: Patch patient for assignment and charts
public Patient patch(Long id, Map<String, Object> updates) {
    Patient patient = repo.findById(id)
        .orElseThrow(() -> new RuntimeException("Patient not found"));

    if (updates.containsKey("assignedTo")) {
        Map<String, Boolean> assignment = new HashMap<>();
        ((Map<?, ?>) updates.get("assignedTo")).forEach((k, v) ->
            assignment.put(String.valueOf(k), Boolean.parseBoolean(v.toString()))
        );
        patient.setAssignedTo(assignment);
    }

    if (updates.containsKey("charts")) {
        Map<String, String> chartMap = new HashMap<>();
        ((Map<?, ?>) updates.get("charts")).forEach((k, v) ->
            chartMap.put(String.valueOf(k), String.valueOf(v))
        );
        patient.setCharts(chartMap);
    }

    if (updates.containsKey("lastVisitDate")) {
        patient.setLastVisitDate(LocalDate.parse(updates.get("lastVisitDate").toString()));
    }

    if (updates.containsKey("lastVisitTime")) {
        patient.setLastVisitTime(LocalTime.parse(updates.get("lastVisitTime").toString()));
    }

    if (updates.containsKey("summary")) {
        patient.setSummary(updates.get("summary").toString());
    }

    if (updates.containsKey("doctorNotes")) {
        patient.setDoctorNotes(updates.get("doctorNotes").toString());
    }

    if (updates.containsKey("recommendation")) {
        patient.setRecommendation(updates.get("recommendation").toString());
    }

    if (updates.containsKey("prescription")) {
        patient.setPrescription(updates.get("prescription").toString());
    }

    if (updates.containsKey("vitals")) {
        Map<String, String> vitalsMap = new HashMap<>();
        ((Map<?, ?>) updates.get("vitals")).forEach((k, v) ->
            vitalsMap.put(String.valueOf(k), String.valueOf(v))
        );
        patient.setVitals(vitalsMap);
    }

    return repo.save(patient);
}
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```
package com.physio.backend.service;
```

```

import com.physio.backend.model.SecurityLog;
import com.physio.backend.repository.SecurityLogRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

```

```

@Service
public class SecurityLogService {

    @Autowired
    private SecurityLogRepository repo;

    public List<SecurityLog> getAll() {
        return repo.findAll();
    }

    public SecurityLog save(SecurityLog log) {
        return repo.save(log);
    }
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```

package com.physio.backend.service;

import com.physio.backend.model.Service;
import com.physio.backend.repository.ServiceRepository;
import org.springframework.beans.factory.annotation.Autowired;

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

@org.springframework.stereotype.Service
public class ServiceService {

    @Autowired
    private ServiceRepository repo;

    public List<Service> getAll() {
        return repo.findAll();
    }

    public Optional<Service> getById(Long id) {
        return repo.findById(id);
    }

    public Service save(Service service) {
        return repo.save(service);
    }

    public Service update(Long id, Service updatedService) {
        Service existing = repo.findById(id).orElseThrow();
        updatedService.setId(id);
        return repo.save(updatedService);
    }

    public void delete(Long id) {
        repo.deleteById(id);
    }
}

```

**File: C:\Users\hp\OneDrive\Desktop\Molsys Internship\physioo\backend\src\main\java\com\physio\backend\service**

```

package com.physio.backend.service;

import com.physio.backend.model.User;
import com.physio.backend.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

import java.util.Map;
import java.util.Optional;

@Service
public class UserService {

    @Autowired
    private UserRepository userRepository;

    public List<User> getAll() {

```



```

        return userRepository.findAll();
    }

    public User create(User user) {
        return userRepository.save(user);
    }
    // In UserService.java
    public User patch(Long id, Map<String, Object> updates) {
        User user = userRepository.findById(id).orElseThrow();
        updates.forEach((k, v) -> {
            switch (k) {
                case "role" -> user.setRole(v.toString());
                case "password" -> user.setPassword(v.toString());
            }
        });
        return userRepository.save(user);
    }

    public User update(Long id, User updatedUser) {
        return userRepository.findById(id)
            .map(existing -> {
                existing.setName(updatedUser.getName());
                existing.setEmail(updatedUser.getEmail());
                existing.setPassword(updatedUser.getPassword());
                existing.setRole(updatedUser.getRole());
                return userRepository.save(existing);
            })
            .orElseThrow(() -> new RuntimeException("User not found"));
    }

    public void delete(Long id) {
        userRepository.deleteById(id);
    }

    public Optional<User> authenticate(String email, String password) {
        return userRepository.findByEmail(email)
            .filter(user -> user.getPassword().equals(password));
    }
}

```

**File: C:\Users\hpl\OneDrive\Desktop\Molsys Internship\physioo\backend\src\test\java\com\physio\backend\Backer**

```
package com.physio.backend;
```

```
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
```

```
@SpringBootTest
class BackendApplicationTests {
```

```
    @Test
    void contextLoads() {
    }
}

```

```
}
```