DOCTOR’S APPLICATION

- ## Login Page:

Login Controller:

```
package com.upendra.Doctor_App_TeleMed.doctLogin;

import java.util.Map;

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

@RestController
@RequestMapping("/login")
public class LoginController {
	@Autowired
	private LoginService loginService;

	@PostMapping(value = "/otpvalidate", consumes = "multipart/form-data",
produces = "application/json")
	public Map<String, Object> otpValidation(@RequestParam String mobile) {
		return loginService.otpValidation(mobile);
	}

}
```

Login Repository:

```
package com.upendra.Doctor_App_TeleMed.doctLogin;

import java.text.ParseException;
import java.util.Random;

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

@Repository
public class LoginRepository {

	@Autowired
	private LoginSmsService loginSmsService;

	public String otpValidation(String mobile) throws ParseException {
		StringBuffer otp = new StringBuffer();
		Random random = new Random();
		for (int i = 0; i < 4; i++) {
			otp.append(String.valueOf(random.nextInt(9)));
		}
```

```
        loginSmsService.send(otp.toString(), mobile);
        return otp.toString();
    }

}
```

Login Service:

```
package com.upendra.Doctor_App_TeleMed.doctLogin;

import java.util.LinkedHashMap;
import java.util.Map;

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

@Service
public class LoginService {
    @Autowired
    private LoginRepository loginRepository;

    public Map<String, Object> otpValidation(String mobile) {
        Map<String, Object> m = new LinkedHashMap<>();
        try {
            m.put("flag", true);
            m.put("message", loginRepository.otpValidation(mobile));
            m.put("status", 1);
            m.put("time", 2);
        } catch (Exception e) {
            m.put("flag", false);
            m.put("message", "Error, while sending OTP");
            m.put("status", 0);
        }
        return m;
    }

}
```

Login SMS Service:

```
package com.upendra.Doctor_App_TeleMed.doctLogin;

import java.text.ParseException;

import org.springframework.stereotype.Component;

import com.twilio.Twilio;
import com.twilio.rest.api.v2010.account.Message;
import com.twilio.type.PhoneNumber;

@Component
public class LoginSmsService {
	private final String ACCOUNT_SID=
"AC290c1ee9f1ef272604a9d025311d94ff";
	private final String AUTH_TOKEN =
"47c877584801e0bb9577a682ed3bba4e";
	private final String FROM_NUMBER = "+12545664510";

	public void send(String otp,String mobile)throws ParseException{
		Twilio.init(ACCOUNT_SID,AUTH_TOKEN);
		String msg="OTP for Telemedicine Doctors App Login is: " +otp;
		Message.creator(new PhoneNumber("+91"+mobile),new
PhoneNumber(FROM_NUMBER),msg).create();
	}


}
```

LIST OF APPOINTMENTS API:

Controller:

```
package com.upendra.Doctor_App_TeleMed.Appointments;


import java.util.Map;


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

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;


@RestController

@RequestMapping("/DoctorsApp")

public class AppController {

	@Autowired

	private AppService service;

	@PostMapping(value = "/Appointments", consumes =
"multipart/form-data", produces = "application/json")
```

```
		public Map<String, Object> doctorsAppointments(@RequestParam int doctor_id) {
			return service.doctorsAppointments(doctor_id);
		}

	}
```

Repository:

```
	package com.upendra.Doctor_App_TeleMed.Appointments;

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

	import org.springframework.beans.factory.annotation.Autowired;
	import org.springframework.jdbc.core.JdbcTemplate;
	import org.springframework.stereotype.Repository;

	@Repository
	public class AppRepo {
		@Autowired
		private JdbcTemplate jdbcTemplate;

		public Map<String, Object> Appointments(int doctor_id) {
			Map<String, Object> m1 = new HashMap<>();
			List<Map<String, Object>> appoint = Listofpatients(doctor_id);
			m1.put("appointments", appoint);
			return m1;
		}

		public List<Map<String, Object>> Listofpatients(int doctor_id) {
```

```
		StringBuffer query = new StringBuffer();
		query.append(" select a.name NAME,a.age AGE,a.gender GENDER,b.appointment_session  APPOINTMENTSESSION ");
		query.append(" from tbl_patient_register a ");
		query.append(" left join tbl_appointments b on a.patient_id=b.patient_id ");
		query.append(" left join tbl_doctor_register c on b.doctor_id=c.doctor_id ");
		query.append(" where appointment_session>now() and c.doctor_id=? ");
		return jdbcTemplate.queryForList(query.toString(), new Object[] { doctor_id });
	}


}
```

Service:

```
package com.upendra.Doctor_App_TeleMed.Appointments;


import java.util.LinkedHashMap;
import java.util.Map;


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


@Service
public class AppService {
	@Autowired
	private AppRepo appRepo;


	public Map<String, Object> doctorsAppointments(int doctor_id) {
		Map<String, Object> m = new LinkedHashMap<>();
```

```
                try {

                        Map<String, Object> m2 =
        appRepo.Appointments(doctor_id);
                        m.put("flag", true);
                        m.put("status", 1);
                        m.put("patients", m2);

                } catch (Exception e) {
                        e.printStackTrace();
                        m.put("flag", false);
                        m.put("status", 0);
                        m.put("message", "Error, fetching doctor info ");

                }
                return m;
        }
}
```

PATIENT DETAILS AND MEDICAL HISTORY API:

Controller:

```
        package com.upendra.Doctor_App_TeleMed.patient_detailsandmed_History;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
@RequestMapping("/patients")
public class Pat_detandMed_historyController {

    @Autowired
    private Pat_detandMed_historyService service;

    @PostMapping(value = "/medicalhistory", consumes = "multipart/form-data", produces = "application/json")
    public Map<String, Object> DoctorsandmedInfo(@RequestParam int patient_id) {
        System.out.println("came to controller"+patient_id);
        return service.DoctorsandmedInfo(patient_id);
    }
}
```

Repository:

```
        package com.upendra.Doctor_App_TeleMed.patient_detailsandmed_History;

    import java.io.IOException;
    import java.nio.file.Files;
    import java.nio.file.Path;
    import java.nio.file.Paths;
    import java.util.HashMap;
    import java.util.List;
    import java.util.Map;

    import org.springframework.beans.factory.annotation.Autowired;
    import org.springframework.jdbc.core.JdbcTemplate;
    import org.springframework.stereotype.Repository;
```

```
@Repository
public class Pat_detandMed_historyRepo {
    @Autowired
    private JdbcTemplate jdbcTemplate;

    public Map<String, Object> totaldata(int patient_id) {
        Map<String, Object> m1 = new HashMap<>();
        Map<String, Object> patientDetails = patientDetails(patient_id);
        List<CompleteDoctorDetails> prevAppointments = Listofdoctors(patient_id);

        m1.put("patientDetails", patientDetails);
        m1.put("doctors", prevAppointments);

        return m1;
    }

    public List<CompleteDoctorDetails> Listofdoctors(int patient_id) {
        StringBuffer query = new StringBuffer();
        query.append(" select a.name ,d.name dept_name,a.img_path ");
        query.append(" from tbl_doctor_register a ");
        query.append(" left join tbl_appointments b on a.doctor_id=b.doctor_id ");
        query.append(" left join tbl_patient_register c on b.patient_id=c.patient_id ");
        query.append(" left join tbl_departments d on a.dept_id=d.id ");
        query.append(" where appointment_session<now() and c.patient_id="+patient_id);
        return jdbcTemplate.query(query.toString(), (rs, rowNum) -> {
            CompleteDoctorDetails dc = new CompleteDoctorDetails();
            dc.setName(rs.getString("name"));
```

```
                dc.setDept_name(rs.getString("dept_name"));
                String filePath = rs.getString("img_path");
                System.out.println(filePath);
                if (filePath != null) {
                    Path path = Paths.get(filePath);
                    System.out.println(path+"came");
                    try {
                        byte[] data = Files.readAllBytes(path);
                        dc.setImg_path(data);
                    } catch (IOException e) {
                        e.printStackTrace();
                        throw new RuntimeException();
                    }
                }
                return dc;

            });
        }

        public Map<String, Object> patientDetails(int patient_id) {
            StringBuffer query = new StringBuffer();
            query.append(" select NAME,AGE,GENDER,HEIGHT,WEIGHT ");
            query.append(" from tbl_patient_register ");
            query.append(" where patient_id=? ");
            return jdbcTemplate.queryForMap(query.toString(), new Object[] { patient_id });
        }


    }
```

Service:

```
package com.upendra.Doctor_App_TeleMed.patient_detailsandmed_History;

import java.util.LinkedHashMap;
import java.util.Map;

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

@Service
public class Pat_detandMed_historyService {
	@Autowired
	private Pat_detandMed_historyRepo repo;

	public Map<String, Object> DoctorsandmedInfo(int patient_id) {

		Map<String, Object> m = new LinkedHashMap<>();
		try {
			Map<String, Object> m2 =repo.totaldata(patient_id);

			m.put("flag", true);
			m.put("status", 1);
			m.put("data", m2);
		} catch (Exception e) {
			e.printStackTrace();
			m.put("flag", false);
			m.put("status", 0);
			m.put("message", "something went wrong please look into it ");
```

```
            }

            return m;

        }


    }
```

Model:

```
package com.upendra.Doctor_App_TeleMed.patient_detailsandmed_History;

public class CompleteDoctorDetails {

    private String name;
    private String dept_name;
    private byte[] img_path;

    public byte[] getImg_path() {
        return img_path;
    }

    public void setImg_path(byte[] img_path) {
        this.img_path = img_path;
    }

    public String getName() {
        return name;
    }

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

    public String getDept_name() {
        return dept_name;
    }

    public void setDept_name(String dept_name) {
        this.dept_name = dept_name;
    }

    @Override
    public String toString() {
        return "CompleteDoctorDetails [ name=" + name + ", dept_name=" +
dept_name + ", img_path=" + img_path + "]";

    }

}
```

SCIENTIFIC LITERATURE:

Controller:

```
package com.upendra.Doctor_App_TeleMed.Scientific_Literature;
```

```
import java.util.Map;


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


@RestController
@RequestMapping("/Scientific")
public class LiteratureController {
    @Autowired
    private LiteratureService service;


    @PostMapping("/Literature")
    public Object fetchallLiteratures() {
        return service.fetchallLiteratures();
    }
    @PostMapping(value = "/SingleLiterature", consumes = "multipart/form-data", produces = "application/json")
    public Map<String, Object> SingleLiterature(@RequestParam int id) {
        return service.SingleLiterature(id);
    }
}
```

Repository:

```
package com.upendra.Doctor_App_TeleMed.Scientific_Literature;


import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

import java.util.List;

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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;


@Repository

public class LiteratureRepository {

    @Autowired

    private JdbcTemplate jdbcTemplate;


    public List<Literature> fetchallLiteratures() {

        String sql = "select subject,post_date,img from tbl_scientific_literature ";

        return jdbcTemplate.query(sql.toString(), (rs, rowNum) -> {

            Literature e = new Literature();

            e.setSubject(rs.getString("subject"));

            e.setPost_date(rs.getString("post_date"));


            String filePath = rs.getString("img");

            if (filePath != null) {

                Path path = Paths.get(filePath);


                try {

                    byte[] data = Files.readAllBytes(path);

                    e.setImg(data);

                } catch (IOException r) {

                    r.printStackTrace();

                    throw new RuntimeException();

                }
                }
                return e;
            });

        }

        public SingleLiterature SingleLiterature(int id) {
            StringBuffer query = new StringBuffer();
            query.append(" select
SUBJECT,DESCRIPTION,IMG,POST_DATE,POST_TIME ");
            query.append(" from tbl_scientific_literature ");
            query.append(" where id=" + id);

            return jdbcTemplate.queryForObject(query.toString(), (rs, rowNum) ->
{
                SingleLiterature sl = new SingleLiterature();
                sl.setSubject(rs.getString("subject"));
                sl.setDescription(rs.getString("description"));
                sl.setPost_date(rs.getString("post_date"));
                sl.setPost_time(rs.getString("post_time"));

                String filePath = rs.getString("img");
                if (filePath != null) {
                    Path path = Paths.get(filePath);
                    System.out.println(path + "came");
                    try {
                        byte[] data = Files.readAllBytes(path);
                        sl.setImg(data);
                    } catch (IOException e) {
                        e.printStackTrace();
                    throw new RuntimeException();
                }
            }
            return sl;
        });
    }

}
```

Service:

```
package com.upendra.Doctor_App_TeleMed.Scientific_Literature;

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

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

@Service
public class LiteratureService {
    @Autowired
    private LiteratureRepository repo;

    public Object fetchallLiteratures() {
        Map<String, Object> m9 = new HashMap<>();
        List<Literature> ListOfLiteratues = repo.fetchallLiteratures();
        try {

            m9.put("Flag", true);
```

```
                m9.put("Status", 1);
                m9.put("result", ListOfLiteratues);
        } catch (Exception e) {
                e.printStackTrace();

                m9.put("Flag", false);
                m9.put("Status", 0);
                m9.put("result", "Exception occurs ,PLease look into it");

        }
        return m9;
}

public Map<String, Object> SingleLiterature(int id) {

        Map<String, Object> li = new HashMap<>();
        try {
                li.put("Flag", true);
                li.put("Status", 1);
                li.put("Result", repo.SingleLiterature(id));
        } catch (Exception e) {
                e.printStackTrace();
                li.put("Flag", false);
                li.put("Status", 0);
                li.put("Result", "Exception occurs please look into it");

        }

        return li;
```

```
		}

	}
```

Literature model

```
	package com.upendra.Doctor_App_TeleMed.Scientific_Literature;

public class Literature {

	private String subject;

	private String post_date;

	private byte[] img;

	public String getSubject() {
		return subject;
	}

	public void setSubject(String subject) {
		this.subject = subject;
	}

	public String getPost_date() {
		return post_date;
	}

	public void setPost_date(String post_date) {
		this.post_date = post_date;
	}

	public byte[] getImg() {
		return img;
	}

	public void setImg(byte[] img) {
		this.img = img;
	}

	@Override
	public String toString() {
		return "Literature [subject=" + subject + ", post_date=" + post_date
+ " img=" + img + "]";
	}

}
```

Single Literature model:

```
package com.upendra.Doctor_App_TeleMed.Scientific_Literature;



public class SingleLiterature {
	private String subject;
	private String description;
	private String post_date;
```

```
        private String post_time;
        private byte[] img;
        public String getSubject() {
                return subject;
        }
        public void setSubject(String subject) {
                this.subject = subject;
        }
        public String getDescription() {
                return description;
        }
        public void setDescription(String description) {
                this.description = description;
        }
        public String getPost_date() {
                return post_date;
        }
        public void setPost_date(String post_date) {
                this.post_date = post_date;
        }
        public String getPost_time() {
                return post_time;
        }
        public void setPost_time(String post_time) {
                this.post_time = post_time;
        }
        public byte[] getImg() {
                return img;
        }
        public void setImg(byte[] img) {
                this.img = img;
        }
        @Override
        public String toString() {
                return "SingleLiterature [subject=" + subject + ", description=" +
description + ", post_date=" + post_date
                          + ", post_time=" + post_time + ", img=" + img + "]";
        }


}
```

DOCTOR DETAILS:

Controller:

```
package com.upendra.Doctor_App_TeleMed.DoctorsProfile;



import java.util.Map;



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

@RestController
@RequestMapping("/doctors")
public class DoctorsprofileController {
    @Autowired
    private DoctorsProfileService service;
    @PostMapping(value = "/Profile", consumes = "multipart/form-data", produces = "application/json")
    public Map<String, Object> DoctorsProfile(@RequestParam int doctor_id) {
        return service.DoctorsProfile(doctor_id);
    }



}
```

Repository:

```
    package com.upendra.Doctor_App_TeleMed.DoctorsProfile;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

@Repository
```

```
    public class DoctorsProfileRepo {
        @Autowired
        private JdbcTemplate jdbcTemplate;

        public Profile Doctorsprofiledata(int doctor_id) {

            StringBuffer query = new StringBuffer();
            query.append(" select a.name,c.name
dept_name,count(*)patients,avg(b.rating) rating,a.img_path ");
            query.append(" from tbl_doctor_register a ");
            query.append(" left join tbl_ratings b on a.doctor_id=b.doctor_id ");
            query.append(" left join tbl_departments c on a.dept_id=c.id ");
            query.append(" where a.doctor_id=" + doctor_id);
            query.append(" group by b.doctor_id ");
            return jdbcTemplate.queryForObject(query.toString(), (rs, rowNum) ->
{
                Profile p = new Profile();
                p.setName(rs.getString("name"));
                p.setDept_name(rs.getString("dept_name"));
                p.setPatients(rs.getString("patients"));
                p.setRating(rs.getString("rating"));

                String filePath = rs.getString("img_path");
                if (filePath != null) {
                    Path path = Paths.get(filePath);
                    try {
                        byte[] data = Files.readAllBytes(path);
                        p.setImg(data);
                    } catch (IOException e) {
                        e.printStackTrace();
                        throw new RuntimeException();
                    }
                }
                return p;
            });
        }
    }
```

Service:

```
    package com.upendra.Doctor_App_TeleMed.DoctorsProfile;

import java.util.LinkedHashMap;
import java.util.Map;

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

@Service
public class DoctorsProfileService {
    @Autowired
    private DoctorsProfileRepo repo;

    public Map<String, Object> DoctorsProfile(int doctor_id) {
        Map<String, Object> m = new LinkedHashMap<>();
        try {

            Profile m2 = repo.Doctorsprofiledata(doctor_id);
            m.put("flag", true);
            m.put("status", 1);
            m.put("Result", m2);
        } catch (Exception e) {
            e.printStackTrace();
            m.put("flag", false);
            m.put("status", 0);
            m.put("Result", "Doctors Profile Not found ");

        }
        return m;
    }
}
```

Model:

```
package com.upendra.Doctor_App_TeleMed.DoctorsProfile;


public class Profile {
	private String name;
	private String dept_name;
	private String patients;
	private String rating;
	private byte[] img;

	public String getName() {
		return name;
	}

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

	public String getDept_name() {
		return dept_name;
	}

	public void setDept_name(String dept_name) {
		this.dept_name = dept_name;
	}

	public String getPatients() {
		return patients;
	}

	public void setPatients(String patients) {
		this.patients = patients;
	}
	public String getRating() {
		return rating;
	}

	public void setRating(String rating) {
		this.rating = rating;
	}

	public byte[] getImg() {
		return img;
	}

	public void setImg(byte[] img) {
		this.img = img;
	}

	@Override
	public String toString() {
		return "Profile [name=" + name + ", dept_name=" + dept_name + ",
patients=" + patients + ", rating=" + rating
				+ ", img=" + img + "]";
	}

}
```

RATING AND REVIEWS API

Controller:

```
		package com.upendra.Doctor_App_TeleMed.RatingandReviews;


	import java.util.Map;


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

	import org.springframework.web.bind.annotation.PostMapping;

	import org.springframework.web.bind.annotation.RequestMapping;

	import org.springframework.web.bind.annotation.RequestParam;

	import org.springframework.web.bind.annotation.RestController;


	@RestController

	@RequestMapping("/Rating")

	public class RatingsandReviewsController {

		@Autowired
```

```
        private RatingsandReviewsService service;

        @PostMapping(value = "/reviews", consumes = "multipart/form-data", produces = "application/json")

        public Map<String, Object> RatingsandReviews(@RequestParam int doctor_id) {

            return service.RatingsandReviews(doctor_id);

        }


    }
```

Repository:

```
        package com.upendra.Doctor_App_TeleMed.RatingandReviews;


    import java.util.HashMap;

    import java.util.List;

    import java.util.Map;


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

    import org.springframework.jdbc.core.JdbcTemplate;

    import org.springframework.stereotype.Repository;


    @Repository

    public class RatingsandReviewsRepository {

        @Autowired

        private JdbcTemplate jdbcTemplate;


        public Map<String, Object> firstpart(int doctor_id) {

            StringBuffer query = new StringBuffer();

            query.append(" select count(*)patients,avg(b.rating) rating,count(*) comments  ");

            query.append(" from tbl_doctor_register a ");
            query.append(" left join tbl_ratings b on a.doctor_id=b.doctor_id ");
            query.append(" left join tbl_departments c on a.dept_id=c.id ");
            query.append(" left join tbl_patient_register d on
b.patient_id=d.patient_id ");
            query.append(" where a.doctor_id=? ");

            return jdbcTemplate.queryForMap(query.toString(), new Object[] {
doctor_id });
        }

        public Map<String, Object> mainpart(int doctor_id) {
            Map<String, Object> m = new HashMap<>();

            Map<String, Object> m1 = firstpart(doctor_id);
            Map<String, Object> m2 = secondpart(doctor_id);
            List<Map<String, Object>> m3 = thirdpart(doctor_id);
            m.put("detailsofpatients", m1);
            m.put("reviews", m2);
            m.put("patientreviews", m3);

            return m;
        }

        public Map<String, Object> secondpart(int doctor_id) {
            StringBuffer query = new StringBuffer();
            query.append(" select count(*) patients,avg(a.rating) rating ");
            query.append(" from tbl_ratings a ");
            query.append(" left join tbl_patient_register b on
a.patient_id=b.patient_id ");
            query.append(" where a.doctor_id=? ");
            return jdbcTemplate.queryForMap(query.toString(), new Object[] { doctor_id });
        }

        public List<Map<String, Object>> thirdpart(int doctor_id) {
            StringBuffer query = new StringBuffer();
            query.append(" select b.name,a.comments,if(date(a.lup_date)='0000-00-00','',date(a.lup_date)) DATE,a.RATING ");
            query.append(" from tbl_ratings a ");
            query.append(" left join tbl_patient_register b on a.patient_id=b.patient_id ");
            query.append(" where a.doctor_id=? ");
            return jdbcTemplate.queryForList(query.toString(), new Object[] { doctor_id });
        }

    }
```

Service:

```
package com.upendra.Doctor_App_TeleMed.RatingandReviews;

import java.util.LinkedHashMap;
import java.util.Map;

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

@Service
public class RatingsandReviewsService {
    @Autowired
    private RatingsandReviewsRepository repo;
```

```
        public Map<String, Object> RatingsandReviews(int doctor_id) {
            Map<String, Object> m = new LinkedHashMap<>();
            try {

                Map<String, Object> m1 = repo.mainpart(doctor_id);
                m.put("flag", true);
                m.put("status", 1);
                m.put("reviews", m1);

            } catch (Exception e) {
                e.printStackTrace();
                m.put("flag", false);
                m.put("status", 0);
                m.put("Result", "not fetched,please check it once ");

            }
            return m;
        }
    }
```

DOCTORS FETCH PROFILE:

Controller:

```
        package com.upendra.Doctor_App_TeleMed.fetchProfile;

import java.util.Map;

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

@RestController
@RequestMapping("/fetch")
public class FetchProfileController {
	@Autowired
	private FetchProfileService profileService;

	@PostMapping(value = "/profile", consumes = "multipart/form-data", produces = "application/json")
	public Map<String, Object> fetchdata(@RequestParam String mobile) {
		return profileService.fetchdata(mobile);
	}
}
```

Repository:

```
	package com.upendra.Doctor_App_TeleMed.fetchProfile;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

@Repository
public class FetchProfileRepository {
	@Autowired
```

```
    private JdbcTemplate jdbcTemplate;


    public FetchProfile fetchrepo(String mobile) {
        StringBuffer query = new StringBuffer();
        query.append(" select a.doctor_id id,a.name,b.about,c.name 
dept_name,a.email,a.mobile,a.img_path ");
        query.append(" from tbl_doctor_register a ");
        query.append(" left join tbl_about_doctor b on a.doctor_id=b.doctor_id ");
        query.append(" left join tbl_departments c on a.dept_id=c.id ");
        query.append(" where a.mobile="+ mobile);


        return jdbcTemplate.queryForObject(query.toString(), (rs, rowNum) -> {
            FetchProfile fp = new FetchProfile();
            fp.setName(rs.getString("name"));
            fp.setAbout(rs.getString("about"));
            fp.setDept_name(rs.getString("dept_name"));
            fp.setEmail(rs.getString("email"));
            fp.setMobile(rs.getString("mobile"));


            String filePath = rs.getString("img_path");
            System.out.println(filePath+"repo");
            if (filePath != null) {
                Path path = Paths.get(filePath);
                try {
                    byte[] data = Files.readAllBytes(path);
                    fp.setImg(data);
                } catch (IOException e) {
                    e.printStackTrace();
                    System.out.println("exception");
                    throw new RuntimeException();
                }
            }
            return fp;
        });
    }
}
```

Service:

```
    package com.upendra.Doctor_App_TeleMed.fetchProfile;

import java.util.LinkedHashMap;
import java.util.Map;

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

@Service
public class FetchProfileService {
    @Autowired
    private FetchProfileRepository repo;

    public Map<String, Object> fetchdata(String mobile) {
        Map<String, Object> fetchmap = new LinkedHashMap<>();
        FetchProfile det = repo.fetchrepo(mobile);
        System.out.println(det);
        try {
            fetchmap.put("flag", true);
            fetchmap.put("message", "fetched Successfully");
            fetchmap.put("status", 1);
            fetchmap.put("Result", det);
        } catch (Exception e) {
            e.printStackTrace();
            fetchmap.put("flag", false);
            fetchmap.put("message", "unable to fetch the data please check");
            fetchmap.put("status", 0);
        }
        return fetchmap;
    }
}
```

Model:

```
    package com.upendra.Doctor_App_TeleMed.fetchProfile;

public class FetchProfile {

    private String name;
    private String about;
    private String dept_name;
    private String email;
    private String mobile;
    private byte[] img;

    public String getName() {
        return name;
    }

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

    public String getAbout() {
        return about;
    }

    public void setAbout(String about) {
        this.about = about;
    }

    public String getDept_name() {
        return dept_name;
    }

    public void setDept_name(String dept_name) {
        this.dept_name = dept_name;
    }

    public String getEmail() {
        return email;
    }
```

```
	public void setEmail(String email) {
		this.email = email;
	}

	public String getMobile() {
		return mobile;
	}

	public void setMobile(String mobile) {
		this.mobile = mobile;
	}

	public byte[] getImg() {
		return img;
	}

	public void setImg(byte[] img) {
		this.img = img;
	}

	@Override
	public String toString() {
		return "FetchProfile [name=" + name + ", about=" + about + ", 
dept_name=" + dept_name + ", email=" + email
				+ ", mobile=" + mobile + ", img=" + img + "]";
	}

}
```

DOCTORS UPDATE PROFILE:

Controller:

```
package com.upendra.Doctor_App_TeleMed.updateProfile;


import java.util.Map;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.validation.annotation.Validated;
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 org.springframework.web.multipart.MultipartFile;
```

```
@RestController
@RequestMapping("/Update")
public class UpdateProfileController {
	@Autowired
	private UpdateProfileService service;

	@PostMapping(value = "/updateprofile", consumes = "multipart/form-data",
produces = "application/json")
	public Map<String, Object> profileUpdate(@RequestParam int id, @RequestParam
String name,
			@RequestParam String email, @RequestParam String about,
		 @RequestParam MultipartFile img) {
		return service.profileUpdate(new
Doctor_details().setId(id).setEmail(email).setAbout(about).setName(name), img);
	}
	/*
	 * @PostMapping(value = "/profile", consumes = "application/json", produces =
	 * "application/json") public Map<String, Object>
	 * UpdateProfile(@Validated @RequestBody Map<String, Object> profileUpdate) {
	 * return service.UpdateProfile(profileUpdate); }
	 */

	@PostMapping(value = "/fetch", consumes = "multipart/form-data", produces =
"application/json")
	public Map<String, Object> fetchProfile(@RequestParam String mobile) {
		return service.fetchProfile(mobile);
	}
}
```

Repository:

```
                package com.upendra.Doctor_App_TeleMed.updateProfile;


import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import org.springframework.web.multipart.MultipartFile;


@Repository

public class UpdateProfileRepository {

        @Autowired

        private JdbcTemplate jdbcTemplate;


        @Transactional(rollbackFor = Throwable.class)

        public int profileUpdate(Doctor_details personalDetails, MultipartFile img)

                        throws IllegalStateException, IOException {

                String fileName = img.getOriginalFilename();

                String filePath = "C:\\Doctors\\DoctorPics\\";

                String query = " update telemedicine.tbl_doctor_register a join
telemedicine.tbl_about_doctor b on a.doctor_id=b.doctor_id set
a.name=?,b.about=?,a.email=?,a.img_path=? where a.doctor_id=? ";

                int rowsUp = jdbcTemplate.update(query,
                                        new Object[] { personalDetails.getName(),
personalDetails.getAbout(),
                                                            personalDetails.getEmail(), filePath + fileName,
personalDetails.getId() });
                    File path = new File(filePath);
                    if (path.exists()) {
                              img.transferTo(new File(filePath + fileName));
                    } else {
                              if (path.mkdir() == true) {
                                        img.transferTo(new File(filePath + fileName));
                              } else {
                                        throw new RuntimeException();
                              }
                    }
                    return rowsUp;
          }

          /*
           * public int UpdateProfile(Map<String, Object> profileUpdate) { StringBuffer
           * query = new StringBuffer();
           * query.append(" update telemedicine.tbl_doctor_register a "); query.append(
           * " join telemedicine.tbl_about_doctor b on a.doctor_id=b.doctor_id set
a.name=?,b.about=?,a.mobile=?,email=? "
           * ); query.append(" where a.doctor_id=? "); return
           * jdbcTemplate.update(query.toString(), new Object[] {
           * profileUpdate.get("name"), profileUpdate.get("about"),
           * profileUpdate.get("mobile"), profileUpdate.get("email"),
           * profileUpdate.get("doctor_id") }); }
           */
```

```
    public Doctor_details fetchProfile(String mobile) {
        StringBuffer query = new StringBuffer();
        query.append(" select a.doctor_id id,a.name,b.about,c.name
dept_name,a.email,a.mobile,a.img_path ");
        query.append(" from tbl_doctor_register a ");
        query.append(" left join tbl_about_doctor b on a.doctor_id=b.doctor_id ");
        query.append(" left join tbl_departments c on a.dept_id=c.id ");
        query.append(" where a.mobile="+mobile);
        return jdbcTemplate.queryForObject(query.toString(), (rs, rowNum) -> {
            Doctor_details pf = new Doctor_details();
            pf.setId(rs.getInt("id"));
            pf.setName(rs.getString("name"));
            pf.setAbout(rs.getString("about"));
            pf.setDept_name(rs.getString("dept_name"));
            pf.setEmail(rs.getString("email"));
            pf.setMobile(rs.getString("mobile"));



            String filePath = rs.getString("img_path");
            System.out.println(filePath);
            if (filePath != null) {
                Path path = Paths.get(filePath);
                try {
                    byte[] data = Files.readAllBytes(path);
                    pf.setImg(data);
                    System.out.println(data+"repo");
                } catch (IOException e) {
                    e.printStackTrace();
                    throw new RuntimeException();
                }
            }
            return pf;
        });

    }

}
```

Service:

```
        package com.upendra.Doctor_App_TeleMed.updateProfile;

import java.util.LinkedHashMap;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

@Service
public class UpdateProfileService {
    @Autowired
    private UpdateProfileRepository repo;

    public Map<String, Object> profileUpdate(Doctor_details personalDetails, 
MultipartFile img) {
        Map<String, Object> m = new LinkedHashMap<>();
        try {
            int rowsUpdated = repo.profileUpdate(personalDetails, img);
            System.out.println(rowsUpdated);
            if (rowsUpdated > 1) {
                m.put("flag", true);
                m.put("message", "Profile Updated Successfully");
                m.put("status", 1);

            } else {
                m.put("flag", false);
                m.put("message", "profile not updated");
                m.put("status", 0);

            }
        } catch (Exception e) {
            e.printStackTrace();
            m.put("flag", false);
            m.put("message", "Error, while updating profile");
            m.put("status", 0);

        }
        return m;
    }

    /*
     * public Map<String, Object> UpdateProfile(Map<String, Object> profileUpdate) {
     * Map<String, Object> m = new LinkedHashMap<>(); try { int a =
     * repo.UpdateProfile(profileUpdate); System.out.println(a); if (a > 1) {
     * m.put("flag", true); m.put("status", 1); m.put("message", "update success");
     * } else {
     *
     * m.put("flag", false); m.put("status", 0); m.put("message",
     * "failed to update "); } } catch (Exception e) { e.printStackTrace();
     * m.put("flag", false); m.put("status", 0); m.put("message",
 * "Exception caught  update failed"); } return m; }
 */

public Map<String, Object> fetchProfile(String mobile) {
        Map<String, Object> m = new LinkedHashMap<>();
        try {

                m.put("Result", repo.fetchProfile(mobile));
                        m.put("flag", true);
                        m.put("status", 1);
                        m.put("message", "fetched success");
                } catch(

Exception e)
{
                e.printStackTrace();
                m.put("flag", false);
                m.put("status", 0);
                m.put("message", "Exception caught failed to fetch the data");
        }return m;
}
}
```

Model:

```
        package com.upendra.Doctor_App_TeleMed.updateProfile;

public class Doctor_details {
	private int id;
	private String name;
	private String mobile;
	private String email;
	private String about;
	private String dept_name;

	public String getDept_name() {
		return dept_name;
```

```
}

public Doctor_details setDept_name(String dept_name) {
	this.dept_name = dept_name;
	return this;
}

private byte[] img;

public int getId() {
	return id;
}

public Doctor_details setId(int id) {
	this.id = id;
	return this;
}

public String getName() {
	return name;
}

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

public String getMobile() {
	return mobile;
}

public Doctor_details setMobile(String mobile) {
	this.mobile = mobile;
	return this;
}

public String getEmail() {
	return email;
}

public Doctor_details setEmail(String email) {
	this.email = email;
	return this;
}

public String getAbout() {
	return about;
}

public Doctor_details setAbout(String about) {
	this.about = about;
	return this;
}

public byte[] getImg() {
	return img;
}

public Doctor_details setImg(byte[] img) {
		this.img = img;
		return this;
	}

	@Override
	public String toString() {
		return "Patient_Details [id=" + id + ",dept_name=" + dept_name + ",
name=" + name + ", mobile=" + mobile
				+ ", email=" + email + ", about=" + about + ", img=" +
img + "]";
	}

}
```

HOLIDAYS OF DOCTOR API:

Controller:

```
		package com.upendra.Doctor_App_TeleMed.holidaysofdoctor;


import java.util.Map;


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

import org.springframework.validation.annotation.Validated;

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;


@RestController

@RequestMapping("Holidays")

public class HolidaysofDoctorController {

	@Autowired

	private HolidaysofDoctorService service;

	@PostMapping(value = "/Doctor", consumes = "multipart/form-data", produces =
"application/json")

	public Map<String, Object> HolidaysDoct(@RequestParam int doctor_id) {
        return service.HolidaysDoct(doctor_id);

    }

    @PostMapping(value = "/days", consumes = "application/json", produces =
"application/json")

    public Map<String, Object> noofdays(@Validated @RequestBody
Map<String,Object> days) {

        return service.noofdays(days);

    }


}
```

Repository:

```
        package com.upendra.Doctor_App_TeleMed.holidaysofdoctor;


import java.util.Map;


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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;


@Repository

public class HolidaysofDoctorRepository {

    @Autowired

    private JdbcTemplate jdbcTemplate;


    public int firstpart(Map<String,Object> days) {

        String query = " insert into
tbl_doctor_holidays(doctor_id,from_date,to_date,comment) values(?,?,?,?)";


        return jdbcTemplate.update(query, new Object[] {
days.get("doctor_id"),days.get("to_date"),days.get("from_date"),days.get("comment") });

    }
    public Map<String, Object> secondpart(int doctor_id) {

        StringBuffer query = new StringBuffer();

        query.append(" SELECT from_date ,to_date , DATEDIFF(to_date , from_date) Noofdays ");

        query.append(" from tbl_doctor_holidays");

        query.append(" where doctor_id=? ");


        return jdbcTemplate.queryForMap(query.toString(), new Object[] { doctor_id });

    }


}
```

Service:

```
        package com.upendra.Doctor_App_TeleMed.holidaysofdoctor;


import java.util.LinkedHashMap;

import java.util.Map;


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

import org.springframework.stereotype.Service;


@Service

public class HolidaysofDoctorService {

    @Autowired

    private HolidaysofDoctorRepository repo;
```

```
public Map<String, Object> HolidaysDoct(int doctor_id) {
    Map<String, Object> m = new LinkedHashMap<>();
    try {

        Map<String, Object> m1 = repo.secondpart(doctor_id);
        m.put("flag", true);
        m.put("status", 1);

        m.put("Result", m1);

    } catch (Exception e) {
        e.printStackTrace();
        m.put("flag", false);
        m.put("status", 0);
        m.put("Result", "Something went wrong ,please check ");

    }
    return m;
}
public Map<String, Object> noofdays(Map<String, Object> days) {
    Map<String, Object> m = new LinkedHashMap<>();
    try {
        int rowsUp = repo.firstpart(days);
        System.out.println(rowsUp);
        if (rowsUp ==1) {
            m.put("flag", true);
            m.put("status", 1);
            m.put("message", " uploaded success");
        } else {
                m.put("flag", false);
                m.put("status", 0);
                m.put("message", " failed to load");
            }
        } catch (Exception e) {
            e.printStackTrace();
            m.put("flag", false);
            m.put("status", 0);
            m.put("message", "Exception caught Feedback uploaded failed");
        }
        return m;
    }



}
    A
```