

INT222

Dental Project

Name: Ravinda Yadav

Regis.No. 11708826

Roll.no: A20

Content:

Pages:

1. Homepage
2. About Us
3. News
4. Services
5. Contact Us
6. Admin ->
 1. Dashboard
 2. Appointments ->
 1. Book Appointment
 2. All Appointment
 3. Update Appointment
 4. Cancel Appointment
 3. Patients ->
 1. All patients
 2. Add patient
 3. Delete patient
 4. View Patient

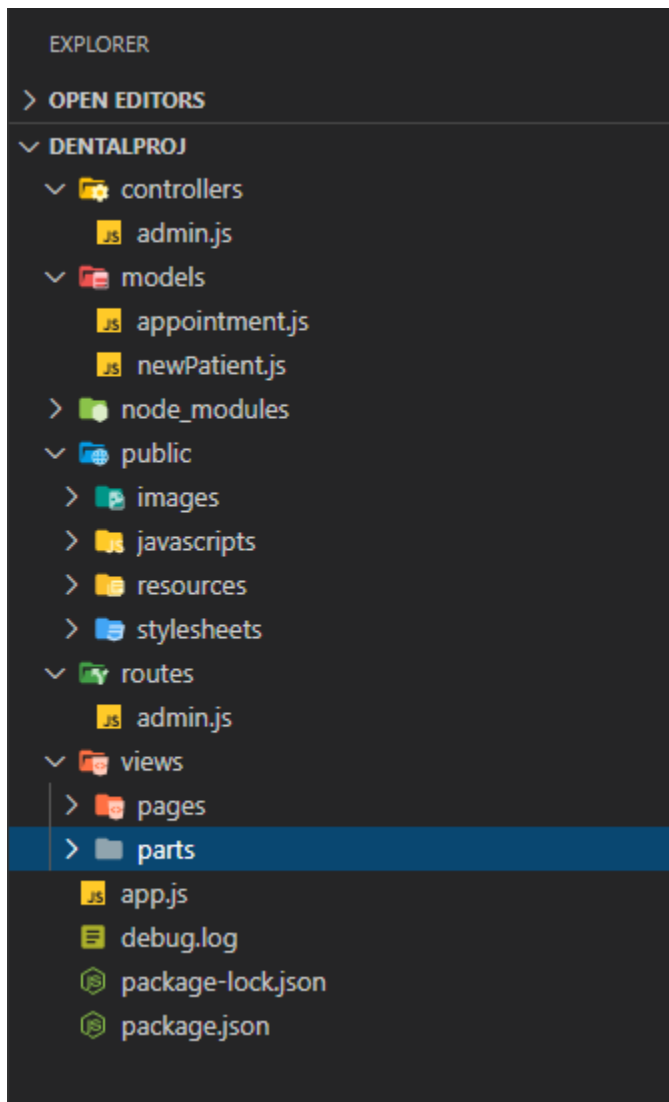
Backend Services Implemented:

1. Form created
2. Validation
3. MongoDB using mongoose
4. Create,delete,update all the crud on mongoose done
5. Get,Post,Delete routes implemented
6. Used routes style
7. MVC style
8. Express server is used
9. Front-end done using bootstrap,css and html

Github link of the project:

<https://github.com/Ravindrayadavrk1006/expressnodeproject/tree/master>

directory structure:



Some code:

Main app.js

```
const express=require('express');
const bodyParser=require('body-parser');
const mongoose=require('mongoose')
const controllers=require('./controllers/admin')
const adminRoutes=require('./routes/admin')
const port=3001
const app=express();
app.use(bodyParser.urlencoded({extended:true}));
app.use(bodyParser.json());
databaseName='OmDentals'
const url=`mongodb://localhost:27017/${databaseName}`
app.set('view engine','ejs');
app.use(express.static(__dirname+'/public'))
//loading the admin routes over the /admin
app.use('/admin',adminRoutes);
app.get('/',(req,res,next)=>{
    res.render('pages/index')
})
app.get('/aboutUs',(req,res)=>{
    res.render('pages/aboutus')
})
app.get('/news',(req,res)=>{
    res.render('pages/news')
})
app.get('/services',(req,res)=>{
    res.render('pages/services')
})
app.get('/contactUs',(req,res)=>{
    res.render('pages/contactUs')
})

//connecting to dbase and inside it connecting to the express server inside the m
mongoose promise
mongoose.connect(url,{
    useNewUrlParser:true,
    useUnifiedTopology:true,
})
.then((result)=>{
```

```

console.log('connected to databse');
app.listen(port,()=>{
  console.log(`application is running at localhost:${port}`)
})
})
.catch(err=>{
  console.log(err)
})

```

The controller file:

```

// const bodyParser=require('body-parser');
const Patient=require('../models/newPatient')
const bookAppointment=require('../models/appointment')
const {validationResult}=require('express-validator');
exports.addPatients=(req,res)=>{
  let tempObj={
    firstName:req.body.firstName,
    secondName:req.body.secondName,
    mobileNo:req.body.mobileNo,
    DOB:req.body.Dob,
    email:req.body.email,
    treatmentType:req.body.treatmentType,
    treatmentInfo:req.body.treatmentInfo
  }
  console.log(tempObj);
  const aPatient=new Patient(tempObj)
  aPatient
    .save()
    .then(result=>{
      console.log("data saved to the db");
      res.redirect('/admin')
      // res.render('\admin',{msg:"patient added succesfully SUCCESSFULLY!"
    })
  })
}
exports.bookAppointment=((req,res)=>{
  let tempObj=
  {
    firstName:req.body.firstName,
    secondName:req.body.secondName,

```

```

        dateTime:req.body.dateTime,
        email:req.body.email,
        mobileNo:req.body.mobileNo,
    }
    console.log(tempObj);
    let newAppointment=new bookAppointment(tempObj)
    newAppointment
        .save()
        .then(result=>{
            console.log("appointment booked");
            res.redirect('/admin')
        })
        .catch(err=>{
            if (err) throw Error;
            console.log(err);
        })

    // res.redirect('/admin',{msg:"appointment booked SUCCESSFULLY!"})
})
exports.allPatients=(req,res)=>{
    Patient.find({},(err,result)=>{
        if(err) throw new Error
        else
            console.log(result)
            res.render('pages/adminPages/allPatients',{patients:result})
    })
}
exports.getdeletePatient=(req,res)=>{
    res.render('pages/adminPages/deletePatient');
}
exports.deletePatient=(req,res)=>{
    var firstName=req.body.firstName;
    var secondName=req.body.secondName;
    Patient.deleteOne({firstName:firstName,secondName:secondName},(err,result)=>{
        if(err) throw new Error
        else
        {
            console.log("patient deleted")
            res.redirect('/admin')
        }
    })
}

exports.updatePatientInfo=(req,res)=>{

```

```

    var firstName=req.body.firstName;
    var secondName=req.body.secondName;
    Patient.findOne({firstName:firstName,secondName:secondName}),(err,result)=>{
        console.log(result);
        var oldTreatmentType=result.treatmentType;
        var oldTreatmentInfo=result.treatmentInfo;
        var newTreatmentType=req.body.treatmentType;
        var newTreatmentInfo=req.body.treatmentInfo;
        var treatmentType=newTreatmentType;
        var treatmentInfo=oldTreatmentInfo+ " " + newTreatmentInfo
        Patient.updateOne({firstName:firstName,secondName:secondName},{ $set:{treatmentInfo:treatmentInfo,treatmentType:treatmentType}},{upsert:true});
    })
    res.redirect('/admin');
}
exports.allAppointments=(req,res)=>{
    bookAppointment.find({},(err,result)=>{
        res.render('pages/adminPages/allAppointments',{appointments:result})
    })
}
exports.getcancelAppointment=(req,res)=>{
    res.render('pages/adminPages/cancelAppointment')
}
exports.cancelAppointment=(req,res)=>{
    var firstName=req.body.firstName;
    var secondName=req.body.secondName;
    bookAppointment.deleteMany({firstName:firstName,secondName:secondName}),(err,result)=>{
        if(err) throw new Error(err);
        console.log(result);
        res.redirect('/admin')
    })
}
exports.getupdateAppointment=(req,res)=>{
    res.render('pages/adminPages/updateAppointment')
}
exports.updateAppointment=(req,res)=>{
    var firstName=req.body.firstName;
    var secondName=req.body.secondName;
    var dateTime=req.body.dateTime;
    bookAppointment.updateOne({firstName:firstName,secondName:secondName},{ $set:{dateTime:dateTime}),(err,result)=>{
        if (err) throw new Error("cannot update the appointment")
        console.log("appointment updated with result=>",result);
    }
}

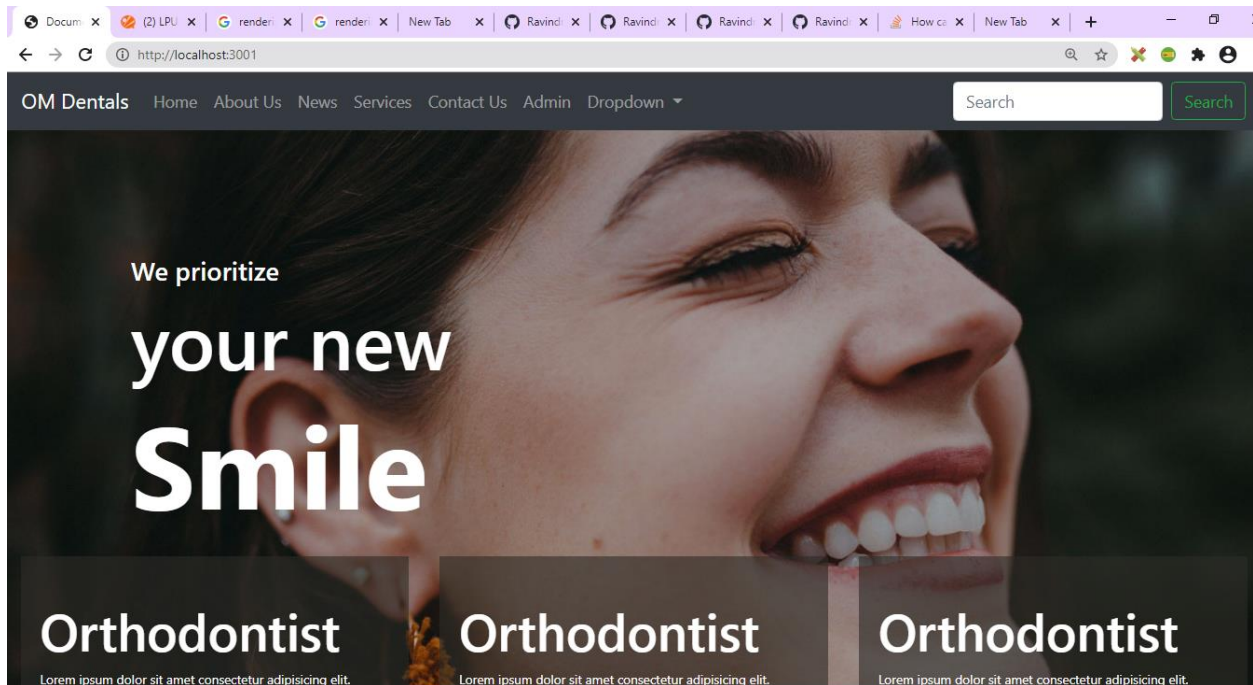
```

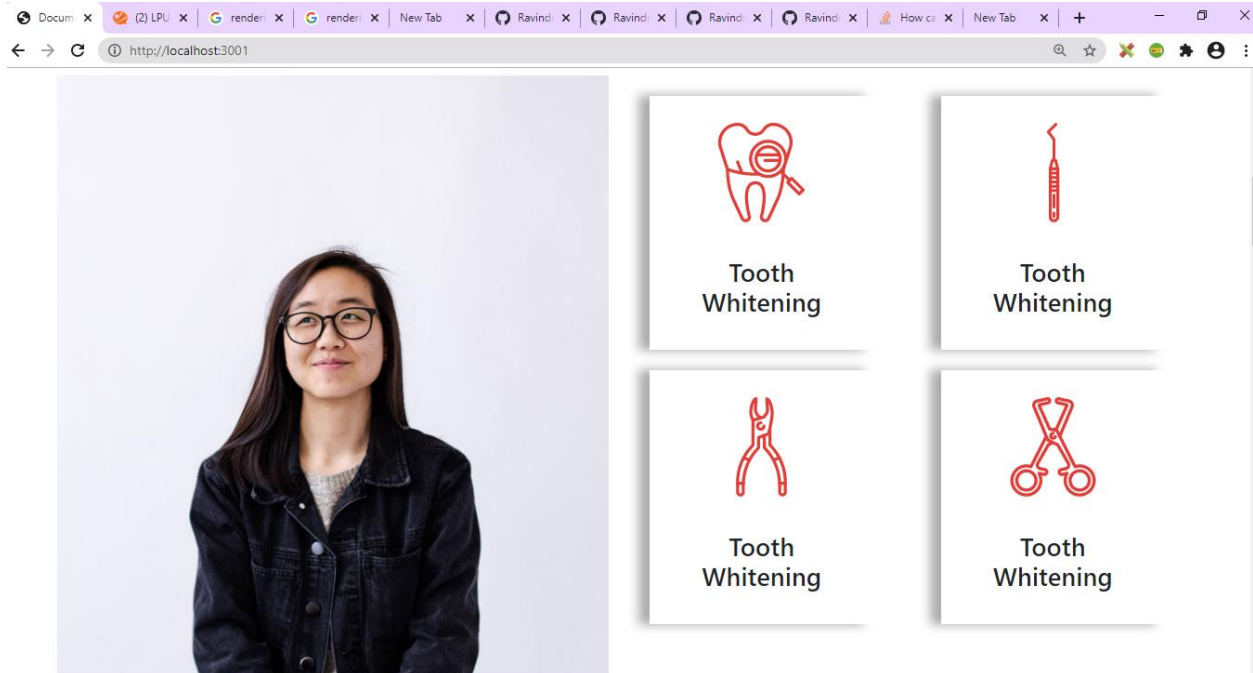
```

});
res.redirect('/admin');
}
exports.findPatient=(req,res)=>{
  var firstName=req.body.firstName;
  var secondName=req.body.secondName;
  Patient.findOne({firstName:firstName,secondName:secondName}),(err,result)=>{
    console.log(result);
    res.render(`pages/adminPages/showPatient`,{patient:result});
  })
}
exports.getfindPatient=(req,res)=>{
  res.render('pages/adminPages/findPatient')
}
exports.getupdatePatientInfo=(req,res)=>{
  res.render('pages/adminPages/updatePatientInfo');
}

```

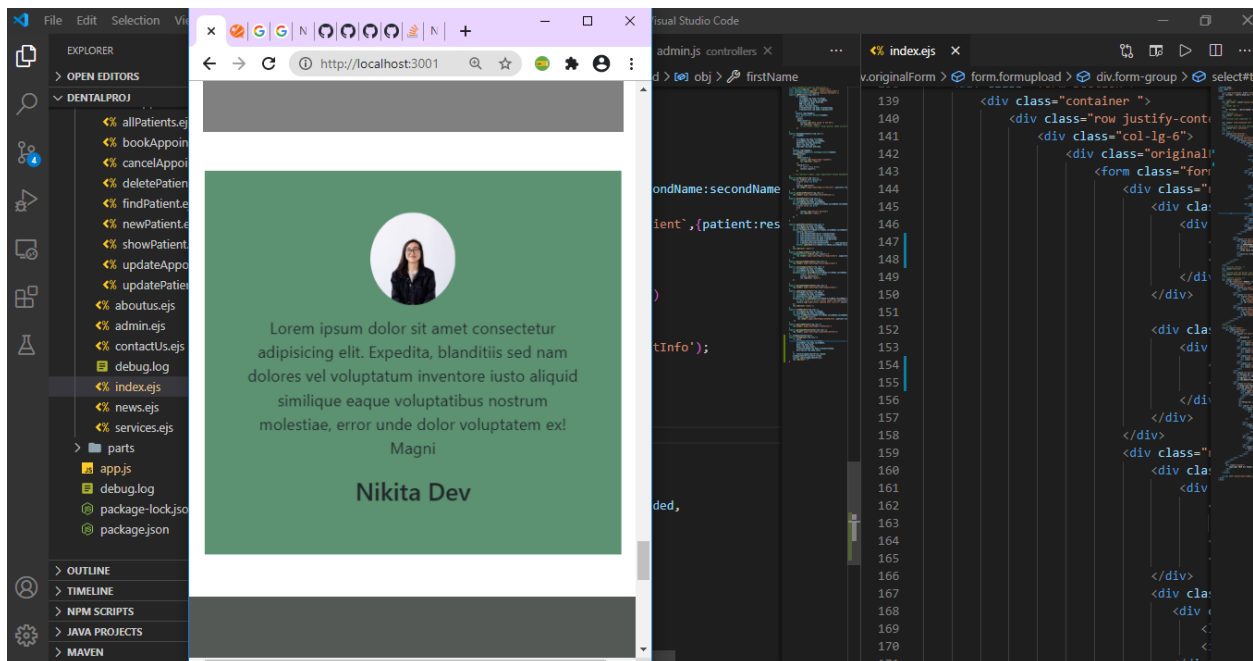
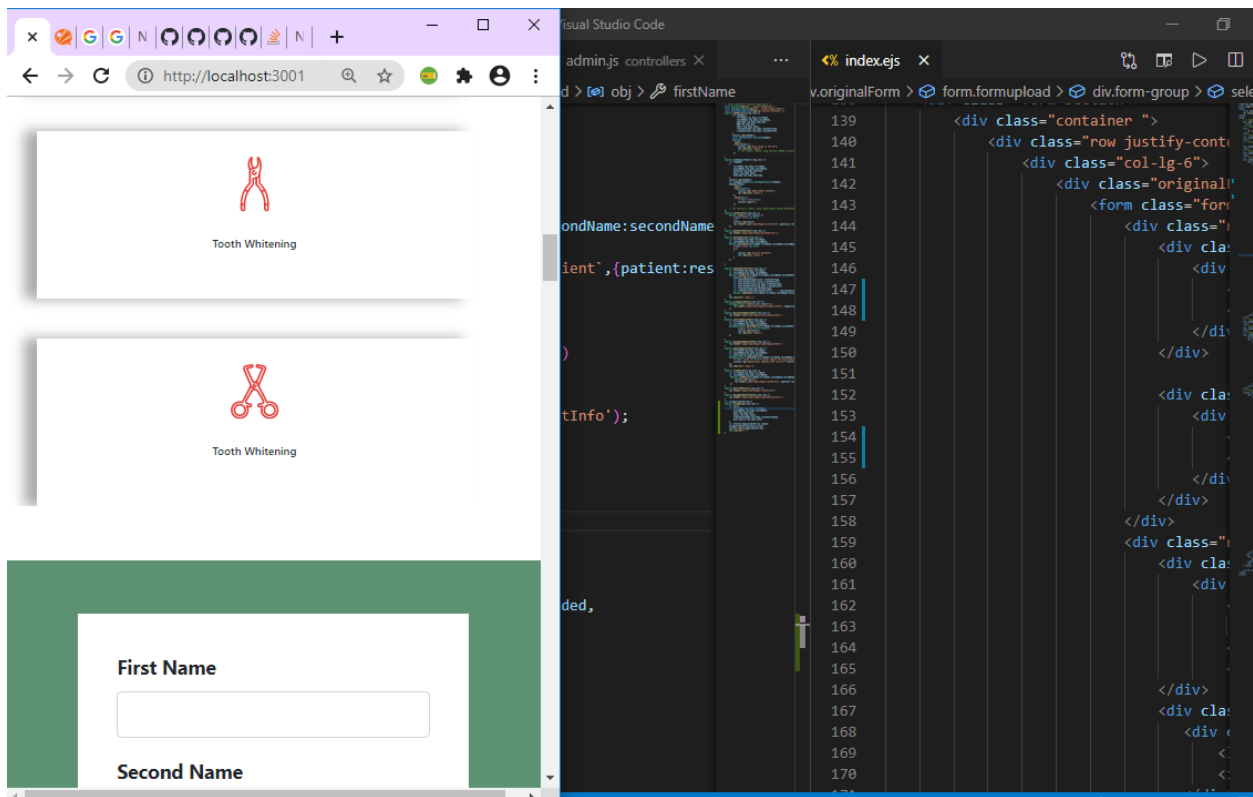
Some html views:





The Website is made fully responsive :

Some check:



Some pages of admin section and his functionalities:

Book Appointment

First Name

Second Name

Email

Mobile No

Date

Book

Cancel

Add patient :

Add Patient

First Name

Last Name

Mobile No.

Date of Birth

Email

Treatment Type

Area to focus during treatment

Find the patient

first name	second Name
<input type="text" value="rao"/>	<input type="text" value="yadav"/>
<input type="button" value="Show"/>	

All the pages are implemented and well working authentication has not been implemented except that almost all functionalities implemented