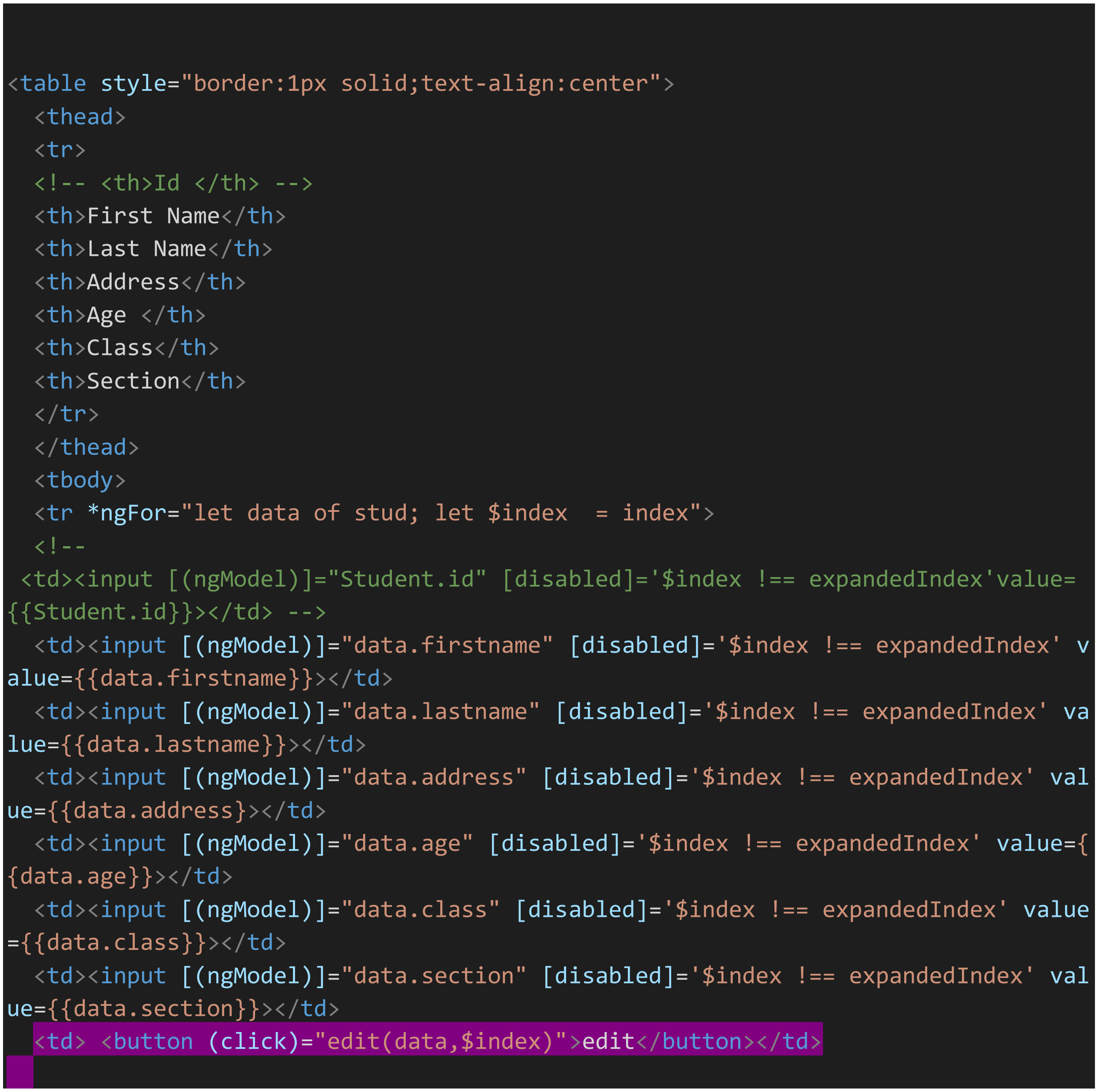
FRONT END

Creating an API using put method

Step:1

Update **view-student.html file** by adding the below code

view-student.html



```
<table style="border:1px solid;text-align:center">
  <thead>
  <tr>
  <!-- <th>Id </th> -->
  <th>First Name</th>
  <th>Last Name</th>
  <th>Address</th>
  <th>Age </th>
  <th>Class</th>
  <th>Section</th>
  </tr>
  </thead>
  <tbody>
  <tr *ngFor="let data of stud; let $index  = index">
  <!--
 <td><input [(ngModel)]="Student.id" [disabled]='$index !== expandedIndex'value=
{{Student.id}}></td> -->
  <td><input [(ngModel)]="data.firstname" [disabled]='$index !== expandedIndex' v
alue={{data.firstname}}></td>
  <td><input [(ngModel)]="data.lastname" [disabled]='$index !== expandedIndex' va
lue={{data.lastname}}></td>
  <td><input [(ngModel)]="data.address" [disabled]='$index !== expandedIndex' val
ue={{data.address}}></td>
  <td><input [(ngModel)]="data.age" [disabled]='$index !== expandedIndex' value={
{data.age}}></td>
  <td><input [(ngModel)]="data.class" [disabled]='$index !== expandedIndex' value
={{data.class}}></td>
  <td><input [(ngModel)]="data.section" [disabled]='$index !== expandedIndex' val
ue={{data.section}}></td>
  <td> <button (click)="edit(data,$index)">edit</button></td>
  <td><button (click)="save(data)">save</button></td>
  </tr>

  </tbody>
  </table>
```

Step:2

Update the view-student.ts file as below and write a promise in save()

view-student.ts

```
import { Component, OnInit } from '@angular/core';
import { ApiService } from '../api.service';
import { Student } from '../create-student/student';
@Component({
  selector: 'app-view-students',
  templateUrl: './view-students.component.html',
  styleUrls: ['./view-students.component.css']
})
export class ViewStudentsComponent implements OnInit {
  stud
  val:boolean = true ;
  expandedIndex: any
  constructor(private service: ApiService) { }

  ngOnInit() {
    console.log('hello');
   this.getAllStudents();
  }
//gets data from db
  getAllStudents () {
    console.log('hi');
    this.service.getstudents().subscribe((res: any) => {
      this.stud = res
      console.log(res);
    }, err => {
      console.log(err);
    })
  }
//edit the data
    public edit(data, index) {
```

```
    this.expandedIndex = index;
  }
  //saves edited data
  public save(data) {
     this.service.add(data).subscribe((data) => {
       console.log(data)
       this.val = true
     })
    }
  }
```

Step:3

Add an edit API by using put method as below to the existing service file

api.service.ts

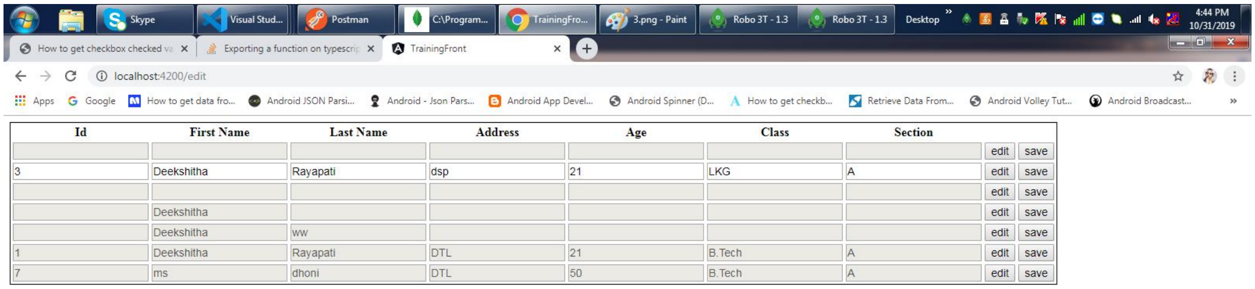
```
public add(data){

  return this._http.put(this.rootUrl+"edit/edit", data , {
    observe: 'body'
 });
}
```

Step:4 run the project

The UI part of the above code is

1.This is the screen we get, when we click on edit button that particular row can only be enable to update and while clicking on save the updated value will be stored in database



Id	First Name	Last Name	Address	Age	Class	Section		
							edit	save
3	Deekshitha	Rayapati	dsp	21	LKG	A	edit	save
							edit	save
	Deekshitha						edit	save
	Deekshitha	ww					edit	save
1	Deekshitha	Rayapati	DTL	21	B.Tech	A	edit	save
7	ms	dhoni	DTL	50	B.Tech	A	edit	save

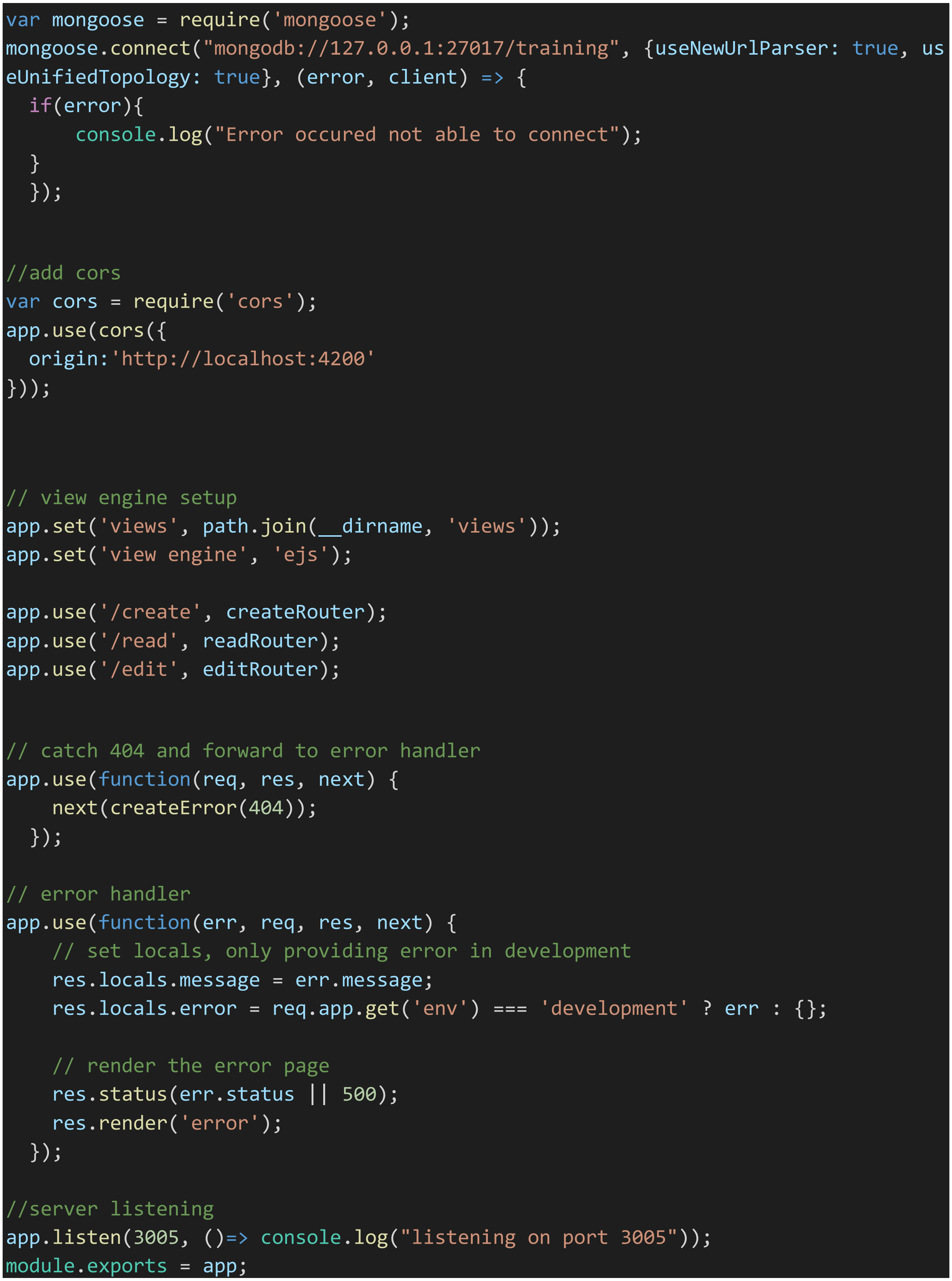
BACK END

Step 1: update/ replace the below code in index.js file

```
var express = require('express');
var path = require('path');

var createRouter = require('./routes/create');
var readRouter = require('./routes/read');
var editRouter = require('./routes/edit');
var bodyParser = require('body-parser')
var app = express();
app.use(bodyParser.urlencoded({ extended: false }))

// parse application/json
app.use(bodyParser.json())
// add mongoose
```



```
var mongoose = require('mongoose');
mongoose.connect("mongodb://127.0.0.1:27017/training", {useNewUrlParser: true, us
eUnifiedTopology: true}, (error, client) => {
  if(error){
      console.log("Error occured not able to connect");
  }
  });


//add cors
var cors = require('cors');
app.use(cors({
  origin:'http://localhost:4200'
}));



// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use('/create', createRouter);
app.use('/read', readRouter);
app.use('/edit', editRouter);


// catch 404 and forward to error handler
app.use(function(req, res, next) {
    next(createError(404));
  });

// error handler
app.use(function(err, req, res, next) {
    // set locals, only providing error in development
    res.locals.message = err.message;
    res.locals.error = req.app.get('env') === 'development' ? err : {};

    // render the error page
    res.status(err.status || 500);
    res.render('error');
  });

//server listening
app.listen(3005, ()=> console.log("listening on port 3005"));
module.exports = app;
```

Step 2: create a file in routes folder with name edit.js and place the below code in that file

```
var express = require('express');
var router = express.Router();

var ReadStudent = require('../models/C-stud');

router.put('/edit', (req, res) => {
    console.log(req.body)
      if (!req.body._id){
        return res.status(400).send(`No record with given id : ${req.body._id}`);
      }
       var data=new ReadStudent()
        var data = {
        //  id: req.body.id,
          firstName: req.body.firstName,
          lastName: req.body.lastName,
          address: req.body.address,
          age : req.body.age,
          class: req.body.class,
          section: req.body.section
        };
        ReadStudent.updateOne({"_id":req.body._id}, { $set: data }, { new: true }, (err, doc) => {
            if (!err)
            { res.send(doc);
                console.log(doc)
            }
            else { console.log('Error in tour Update :' + JSON.stringify(err, undefined, 2)); }
        });
});


module.exports = router;
```

Step 3: start the server by using **node index.js** **command in terminal**