The Modules:  
1. Express  
2. Mysql2  
3. Pug  
4. Body parser  
5. Csv  
6. Csvtojson  
  
The Routs:  
app.get('/', [ start]);

app.get('/drop', [ DBCreation.DropUsersTable,DBCreation.DropTeachersTable]);

app.get('/create', [ DBCreation.CreateUsers, DBCreation.CreateTeachers]);

app.get('/show', [ DBCreation.ShowUsersTable,DBCreation.ShowTeachersTable]);

app.get('/insert', [ DBCreation.InsertDataToUsers, DBCreation.InsertDataToTeachers]);

The Syntax:  
var SQL = require('./db');

const path = require('path');

 const csv =require('csvtojson');

const CreateTeachers = (req,res,next)=> {

    var Q1 = "CREATE TABLE `teachers` (  `id` int NOT NULL,  `fullname` varchar(50) NOT NULL,  `subject` varchar(70) DEFAULT NULL,  `city` varchar(70) DEFAULT NULL,  `price` int DEFAULT NULL,  `experience` int DEFAULT NULL,  `averagerank` double DEFAULT NULL,  `tel` varchar(70) DEFAULT NULL,  PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;"

    console.log("teachers table creation");

    SQL.query(Q1,(err,mySQLres)=>{

        if (err) {

            console.log("error ", err);

            res.status(400).send({message: "error in creating table"});

            return;

        }

        console.log('created teachers table');

        next();

    })

};

const CreateUsers = (req,res,next)=> {

    var Q2 = "CREATE TABLE `users` (`username` varchar(255) NOT NULL,`password` varchar(50) NOT NULL,`fullname` varchar(70) DEFAULT NULL,PRIMARY KEY (`username`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;"

    console.log("users table creation");

    SQL.query(Q2,(err,mySQLres)=>{

        if (err) {

            console.log("error ", err);

            res.status(400).send({message: "error in creating table"});

            return;

        }

        console.log('created users table');

        next();

    })

};

const InsertDataToUsers = (req,res,next)=>{

    var Q3 = "INSERT INTO users SET ?";

    const csvFilePath= path.join(\_\_dirname, "users.csv");

    csv()

    .fromFile(csvFilePath)

    .then((jsonObj)=>{

    jsonObj.forEach(element => {

        var NewEntry = {

            "username": element.username,

            "password": element.password,

            "fullname": element.fullname

        }

        SQL.query(Q3, NewEntry, (err,mysqlres)=>{

            if (err) {

                console.log("error in inserting data", err);

            }

        });

    });

    })

    next();

};

const InsertDataToTeachers = (req,res,next)=>{

    var Q4 = "INSERT INTO teachers SET ?";

    const csvFilePath= path.join(\_\_dirname, "teachers.csv");

    csv()

    .fromFile(csvFilePath)

    .then((jsonObj)=>{

    jsonObj.forEach(element => {

        var NewEntry = {

            "id": element.id,

            "fullname" : element.fullname,

            "subject" : element.subject,

            "city" : element.city,

            "averagerank" : element.averagerank,

            "price" : element.price,

            "tel" : element.tel,

            "experience" : element.experience

        }

        SQL.query(Q4, NewEntry, (err,mysqlres)=>{

            if (err) {

                console.log("error in inserting data", err);

            }

        });

    });

    })

    next();

};

const ShowUsersTable = (req,res,next)=>{

    var Q6 = "SELECT \* FROM users";

    SQL.query(Q6, (err, mySQLres)=>{

        if (err) {

            console.log("error in showing table users", err);

            res.send("error in showing table users");

            return;

        }

        console.log(mySQLres,"showing users table");

        next();

    })};

    const ShowTeachersTable = (req,res,next)=>{

        var Q5 = "SELECT \* FROM teachers";

        SQL.query(Q5, (err, mySQLres)=>{

            if (err) {

                console.log("error in showing table Teachers", err);

                res.send("error in showing table Teachers");

                return;

            }

            console.log(mySQLres,"showing Teachers table");

            next();

        })};

const DropUsersTable = (req, res, next)=>{

    var Q7 = "DROP TABLE users";

    SQL.query(Q7, (err, mySQLres)=>{

        if (err) {

            console.log("error in droping table users", err);

            res.status(400).send({message: "error om dropping table users" + err});

            return;

        }

        console.log("users table drpped");

        next();

    })

};

const DropTeachersTable = (req, res, next)=>{

    var Q8 = "DROP TABLE teachers";

    SQL.query(Q8, (err, mySQLres)=>{

        if (err) {

            console.log("error in droping table Teachers ", err);

            res.status(400).send({message: "error om dropping table Teachers" + err});

            return;

        }

        console.log("Teachers table drpped");

        next();

    })

}

module.exports = {

    CreateUsers,

    CreateTeachers,

    InsertDataToUsers,

     InsertDataToTeachers,

     ShowUsersTable,

     ShowTeachersTable,

     DropUsersTable,

     DropTeachersTable

 };