1. Import Mongoose and Connect

You can create Schema and use it for insert and update data

```
const mongoose = require("mongoose")
mongoose.connect("mongodb://0.0.0.0/my-data")
let employeeSchema =new mongoose.Schema({
    name:String,
    age: Number,
    department: String
})
```

2. Create functions for fetching data

```
let readdata = async () => {
    let employeeModel =
mongoose.model("employee",employeeSchema)
    let data = await employeeModel.find({})
    return data;
}
```

3. Create functions for inserting data

```
let saveData = async (name) => {
    let employeeModel =
mongoose.model("employee",employeeSchema)
    let response = await employeeModel.create({name:name})
    return response;
}
```

4. Create functions for deleting data

```
let deleteData = async (id) => {
    let employeeModel =
mongoose.model("employee",employeeSchema)
    let response = await employeeModel.deleteOne({_id: id})
    return response;
}
```

5. Create functions for deleting data

```
let updatedata = async (id,key,value) => {
    let data = {}
    data[key]=value
    let employeeModel = mongoose.model("employee",employeeSchema)
    let response = await employeeModel.updateOne({_id:id},{$set:data})
    return response;
}
```

Making pages for reading, updating, inserting and deleting data.

Also make a main page.

```
const express = require("express")
const path = require("path")
const mainPath = path.join(__dirname,"public")
const app = express()
app.use(express.static(mainPath))
app.set("view engine","ejs")
app.get("/data",(_,res)=>{
    readdata().then((data)=>{
        res.render("data",{data})
    })
})
app.get("/senddata",(req,res)=>{
    saveData(req.query.name).then((response)=>{
```

```
        res.send(response)
    })
})
app.get("/deletedata",(req,res)=>{
    deleteData(req.query.id).then((response)=>{
        res.send(response)
    })
})

app.get("/updatedata",(req,res)=>{
    updatedata(req.query.id,req.query.name,req.query.value).the
n((response)=>{
        res.send(response)
    })
})

app.listen(4000)
```