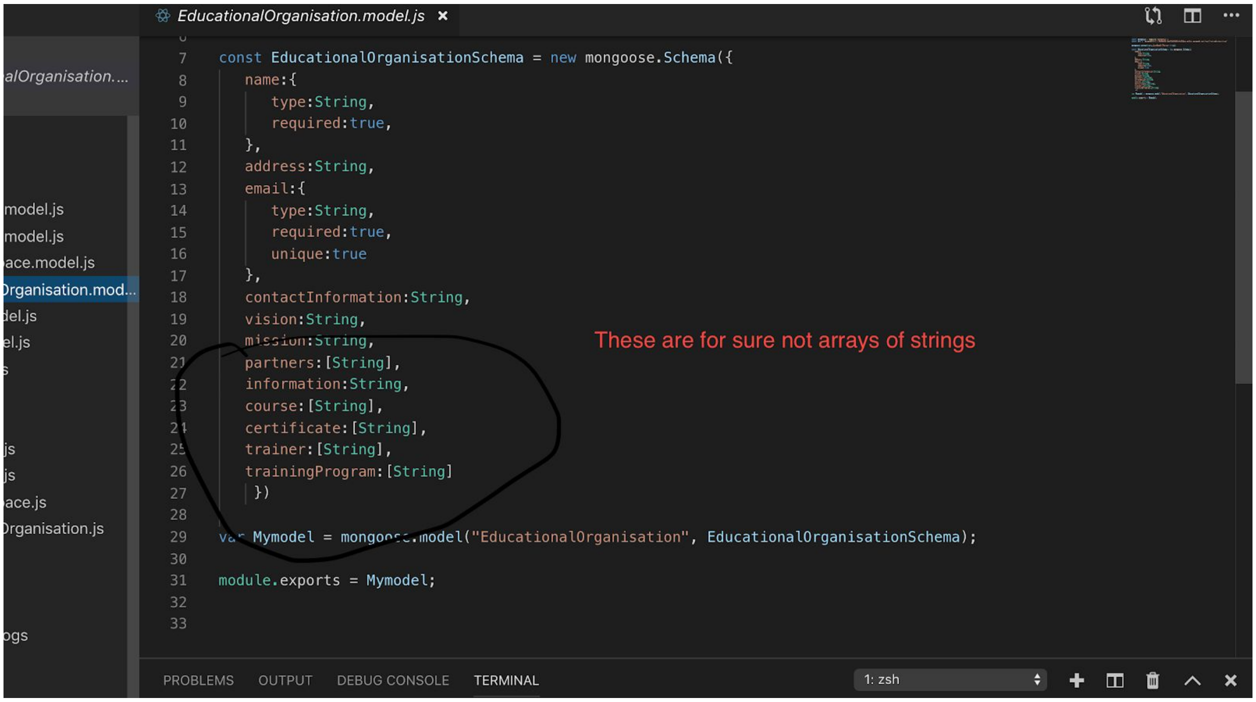
Sprint Backlog	Completeness	Estimation Report	Best practises (Naming conventions, indentations, code structure, etc..)	Integration on master (Individual)
1- The backlog itself (0 or 1) 1- The fair allocation among team members (0 or 1) 2/2	0 - Less than 30% of the sprint backlog 1 - 30-80% 2 - 80-100% 1/3	1 - Team velocity (0 none, 0.5 fake, 1 realistic) 1 - Sprint burndown chart (0 none, 0.5 fake, 1 realistic) 2/2	0 - Spaghetti code 1- Adhered to the main structure and naming conventions 2- Clean code 0.5/1	0 - Code not integrated 2 - Code integrated 2/2

Comments:
Overall I like the structure of your code,
HOWEVER:

1. Instead of using res.status(200) you can just use res.json
2. You handled it well, but think about an alternative for handling 500. I personally do not like announcing something that breaks the server
3. Why are you connecting inside every model??
4. Follow one unified naming convention, you naming schemes are all over the place.
5. Use ES6 Syntax of const and let instead of var
6. Too many unused directories like views and public config and init
7. Follow a uniform indentation or better yet use the same formatter/lintter
8. Your passwords are publicly visible!
9. Code does not run (mongo authentication error)

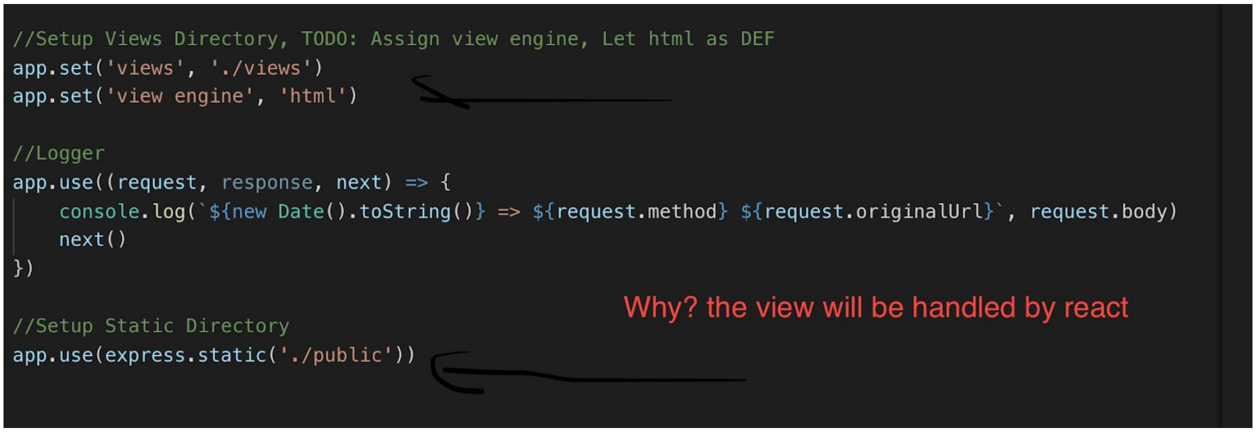


```
const EducationalOrganisationSchema = new mongoose.Schema({
   name:{
      type:String,
      required:true,
   },
   address:String,
   email:{
      type:String,
      required:true,
      unique:true
   },
   contactInformation:String,
   vision:String,
   mission:String,
   partners:[String],
   information:String,
   course:[String],
   certificate:[String],
   trainer:[String],
   trainingProgram:[String]
  })

var Mymodel = mongoose.model("EducationalOrganisation", EducationalOrganisationSchema);

module.exports = Mymodel;
```

These are for sure not arrays of strings

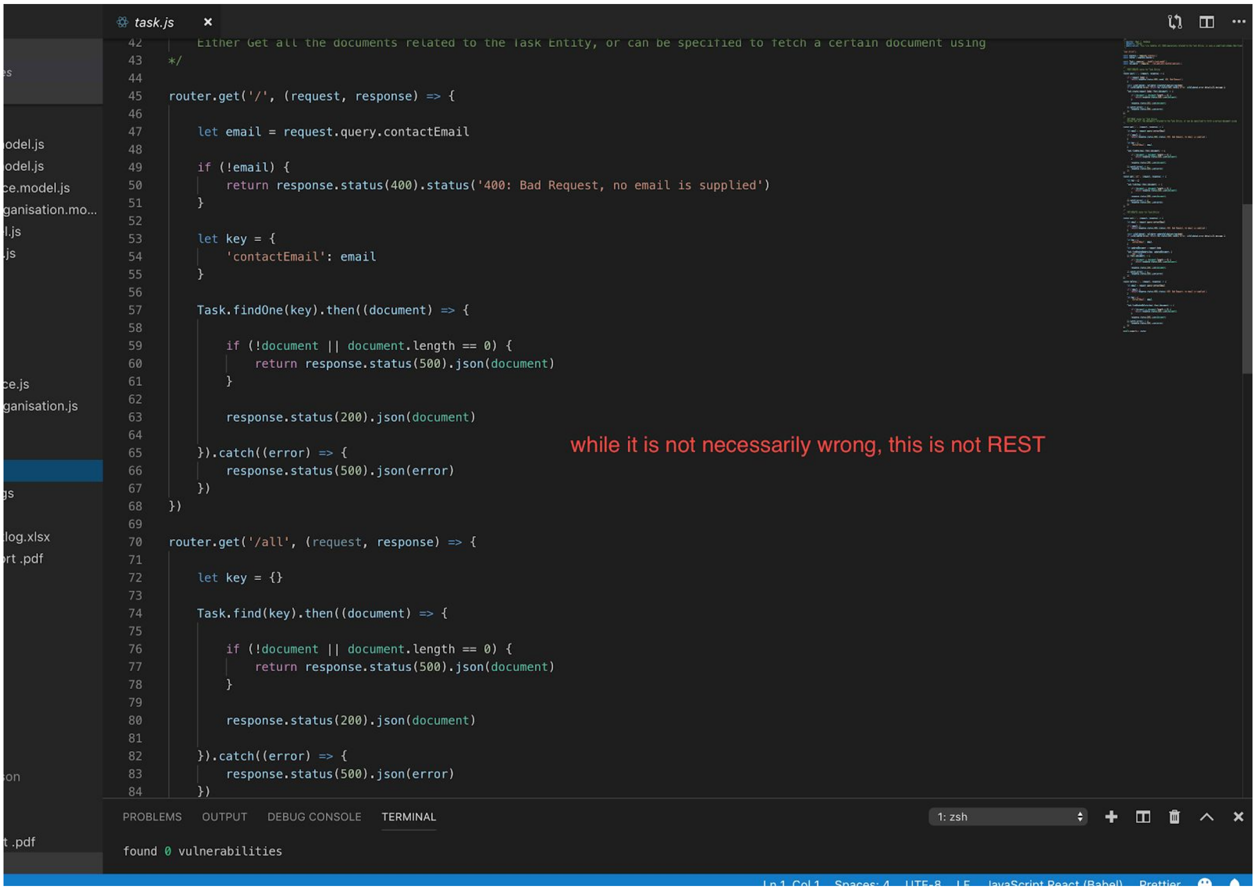


```
//Setup Views Directory, TODO: Assign view engine, Let html as DEF
app.set('views', './views')
app.set('view engine', 'html')

//Logger
app.use((request, response, next) => {
    console.log(`${new Date().toString()} => ${request.method} ${request.originalUrl}`, request.body)
    next()
})

//Setup Static Directory
app.use(express.static('./public'))
```

Why? the view will be handled by react



```
    Either Get all the documents related to the Task Entity, or can be specified to fetch a certain document using
*/

router.get('/', (request, response) => {

    let email = request.query.contactEmail

    if (!email) {
        return response.status(400).status('400: Bad Request, no email is supplied')
    }

    let key = {
        'contactEmail': email
    }

    Task.findOne(key).then((document) => {

        if (!document || document.length == 0) {
            return response.status(500).json(document)
        }

        response.status(200).json(document)

    }).catch((error) => {
        response.status(500).json(error)
    })
})

router.get('/all', (request, response) => {

    let key = {}

    Task.find(key).then((document) => {

        if (!document || document.length == 0) {
            return response.status(500).json(document)
        }

        response.status(200).json(document)

    }).catch((error) => {
        response.status(500).json(error)
    })
```

while it is not necessarily wrong, this is not REST

```
found 0 vulnerabilities
```