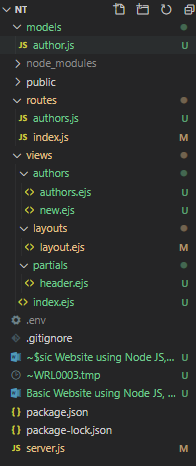
**Basic Website using Node JS, Express JS, MongoDB**



M – Models

V – Views

C – Controller (routes)

Models : author.js :- usually this is in singular form of our routes : authors.js

package.json

{

  "name": "NT",

  "version": "1.0.0",

  "description": "",

  "main": "server.js",

  "scripts": {

    "start": "node server.js",

    "devStart": "nodemon server.js"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "ejs": "^3.1.3",

    "express": "^4.17.1",

    "express-ejs-layouts": "^2.5.0",

    "kill-port": "^1.6.1",

    "mongoose": "^5.9.25"

  },

  "devDependencies": {

    "dotenv": "^8.2.0",

    "nodemon": "^2.0.4"

  }

}

server.js

if (process.env.NODE\_ENV !== 'production') {

  require('dotenv').load()

}

const express = require('express')

const app = express()

const expressLayouts = require('express-ejs-layouts')

const indexRouter = require('./routes/index')

app.set('view engine', 'ejs')

app.set('views', \_\_dirname + '/views')

app.set('layout', 'layouts/layout')

app.use(expressLayouts)

app.use(express.static('public'))

const mongoose = require('mongoose')

mongoose.connect(process.env.DATABASE\_URL, { useUnifiedTopology: true,  useNewUrlParser: true,})

const db = mongoose.connection

db.on('error', error => console.error(error))

db.once('open', () => console.log('Connected to Mongoose'))

app.use('/', indexRouter)

app.listen(process.env.PORT || 3000)

index.js in Routes folder (Controller)

const express = require('express')

const router = express.Router()

router.get('/', (req, res) => {

    res.render('first')

})

module.exports = router

<https://mybrary5403.herokuapp.com/>

<https://www.youtube.com/playlist?list=PLZlA0Gpn_vH8jbFkBjOuFjhxANC63OmXM>

<https://github.com/WebDevSimplified/Mybrary/tree/v1.0>

<https://dashboard.heroku.com/>

Express has no easy way to access variables..hence we have to install library

Cmd : npm i body-parser

While creating applications inside our ejs files, you will have variables that will sometimes be set and sometimes not set.

If your variable is not set then do the following

<% if(locals.errorMessage != null) {%>

<%= errorMessage %>

<% } %>

Here errorMessage was not defined ...so to check if its set or not we are using locals in an if statement.

JavaScript : Promises, Async..await

**m**

**m**