**MERN BOOTCAMP**

**Mango DB** - <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>

Run MangoDB - C:\Program Files\MongoDB\Server\5.0\bin\mongod.exe

**Express JS** - <https://expressjs.com/>

1) npm init => start server.

2) npm i express => install Express.

Before any project do this

// add all used packages as mentioned below.

const express = require("express");

const app = express();

const port = 3000

// create routes

app.get('/', (req, res) => {

  res.send('Hello World!')

})

app.listen(port, () => {

  console.log(`Example app listening on port ${port}`)

})

**Nodemon**

When we change code, we need to restart server, to prevent this we use nodemon (automatic reload)

npm install nodemon => install nodemon

update json file ->

"scripts": {

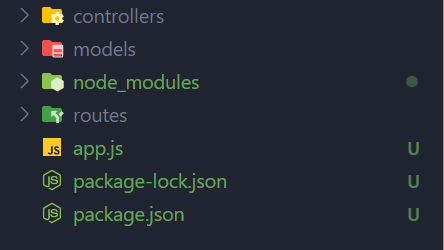
    "start": "nodemon index.js"

  },

To start server use -> npm start

**Backend Development**

Create projbackend folder and create folders as mentioned below



**Mongoose** – used to deal with mangodb and object modelling.

**Documentation** - <https://mongoosejs.com/docs/guide.html>

**Salt cryptography** - <https://en.wikipedia.org/wiki/Salt_(cryptography)>

It is used to store password in encrypted format.

Steps for backend –

1. Add required libraries.
2. Create Models (Mangoose Schema).
3. Export module as

module.exports = mongoose.model("User", userSchema);

[Virtuals](https://mongoosejs.com/docs/api.html#schema_Schema-virtual) are document properties that you can get and set but that do not get persisted to MongoDB. The getters are useful for formatting or combining fields, while setters are useful for de-composing a single value into multiple values for storage.

**UUID** – For the creation of [**RFC4122**](http://www.ietf.org/rfc/rfc4122.txt) UUIDs

<https://www.npmjs.com/package/uuid>

**Crypto**- used to encrypt password.

<https://www.npmjs.com/package/crypto-js>