For exercise 4.

"start": "NODE\_ENV=production node index.js",

    "dev": "NODE\_ENV=development nodemon index.js",

    "test": "NODE\_ENV=test jest --verbose --runInBand"

First of all we changed the scrips file from the package.json .we changed the start,dev and test adding –verbose –runInBand where the execution of the test takes places one after another.

(npm install --save-dev cross-env )=> because the NODE\_ENV scrips we write might not be working in the windows there fore install npm install –save-dev cross\_env and make sure to change it in the scrips. Also make sure that installed cross-env is not installed in the dev dependency. Run the command npm I cross-env -P and move it to dev dependency.the database configuration file is in the .config files of utils . change the MONGO\_URI format into the ternary operation .

Where const MONGODB\_URI =

  process.env.NODE\_ENV === "test"

    ? process.env.TEST\_MONGODB\_URI

    : process.env.MONGODB\_URI;

Now also make changes in the .env file where we have put the link of the DB.

npm install --save-dev supertest . Insatllation of the super test to make sure that we can run all the test cases even without running the npm run dev .. after that add anpther test case condition in the test file and make the test to get run and passed .

made the datas with initialBlog list and added the datas .added few more test cases and made it passed.

Added the sync property in the blogrouter.post and get instead of the promises. After .then function added the await function

Installation of jsonwebtoken : npm install jsonwebtoken

Make a file named user.js and copy the code from the study materials.hashing is the one way where data can be encrypted but it’s a one way . datas that has been encrypted cannot be decrypted back.

In the blog.js set the reference of the user for setting the scema .

Then install the bycrypt to create one way hash. Npm install bycrypt is the command . make a file name users.js in the controller and import it in the app.js file.