1. redux
2. rrd v5
3. files upload
4. secure mails – gmail
5. connect nodejs with react
6. register
   1. with picture
   2. confirm email
7. 2 step auth
   1. email + password
   2. 6 chars
8. forget password

nodejs:

* how to start prod project: npm start
* how to start a dev project: npm run dev

react:

* how to start prod project: npm run build
* how to start a dev project: npm start

Register - nodejs:

1. create route = “/register”
2. create register file
3. import all modules
4. validate user input
   1. if failed
   2. response errors from joy with status 400
5. find user by email to check if email exists
   1. if exists
      1. response errors “the user already exists” with status 400
   2. else (not exists)
      1. encrypt the password
      2. store user data with encrypted password to db
      3. (Do not store the plain text password only the encrypted password)
      4. send confirmation email
         1. generate random number
         2. create url and send it to mail
      5. send “user created successfully” with status 201
6. when user access the url then update db

Confirm register:

1. create route /api/:email/:secretkey
2. user input validation
3. find user by email to check if email exists
   1. if not exists
      1. response error with status 400
   2. else – the email exists
      1. check secretkey
         1. if they no match
            1. response error with status 400
         2. else – they match
            1. update db
            2. response msg ok with status 200

Login – nodejs:

1. create route = “/login”
2. create login file
3. import all modules
4. validate user input
   1. if failed
   2. response errors from joy with status 400
5. find user by email to check if email exists
   1. if not exists
      1. response error “invalid email/password” with status 400
   2. if exists (else)
      1. decode password and compare
         1. if not
            1. response error “invalid email/password” with status 400
         2. else (password correct)
            1. create token
            2. response token