QUESTO FILE POTREMMO USARLO ANCHE PER LA FUTURA DOCUMENTAZIONE, MA PER ORA L’HO FATTO PER FAR CAPIRE COSA HO FATTO, COSICCHÉ SARÀ PIÙ FACILE COLLEGARE LE VARIE COSE

(MOTIVO PER IL QUALE NON È MOLTO APPROFONDITA)

FILE USER.JS presente nella cartella db

Serve per gestire l’autenticazione e l’autorizzazione degli utenti

La password è salvata in modalità hashata per una questione di sicurezza

“\_\_v”:0, //collegamento per mongodb, continua a tenere traccia delle versioni del documento

“cookieJWToken”//per l’autenticazione(opzionale, infatti in quelli dopo non c’è)

server.js 🡪 serve per la configurazione del Express così da collegarsi a mongodb, gestisce le richieste http, utilizza middleware per la sicurezza e gestione degli errori.

CARTELLA SERVER contiene:

* Middleware -FILE corsAcControl.js🡪gestisce il controllo degli accessi cors

per le varie richieste, come Origins che contiene i domini che sono autorizzati a farne richiesta e gestione delle credenziali

errorHandle.js🡪 serve a gestire gli errori che si

verificano nell’app express

logEvents.js🡪 gestisce la registrazione e il logging degli

eventi

veryJWT.js🡪 l’ho creato per “”proteggere”” le varie vie dell’app che richiedono l’autenticazione

* Config -FILE Origins.js🡪 elenco di origini autorizzate, con il

corsAcControl.js, che si trova nella cartella middlware, andrà a consentire o negare gli accessi

CorsOptions.js🡪 controlla l’origine delle richieste in

base al contenuto di Origins, invia le credenziali necessarie per l’autenticazione dell’utente

db.js🡪 si connette a mongodb

relesList.js🡪 serve a definire i ruoli all’interno

dell’app

* Routes🡪 l’ho creato per l’organizzazione delle varie componenti

cartella api contiene:

index.js

logoutR.js

refreshR.js

registrerR.js

routes.js

setupR.js

cartella components:

VideoRow.jsx

* Controllers FILE: Controller.js🡪 gestisce l’operazione di login, verificando

l’autenticazione delle credenziali, garantendo un’implementazione sicura

logout.js🡪 gestisce il logout

refresh.js🡪 garantisce il continuo accesso alle app senza

dover riaccedere continuamente

Registrer.js🡪 registrazione dei nuovi utenti

userController.js🡪 gestisce gli utenti, come tipo

l’eliminazione dell’account e cose

simili

la cartella models: User.js🡪 definisce il modello user che

poi verrà utilizzato in

mongo tramite l’ultima riga del file