NETFLIX FULL STACK

Build Comprises of:

* Backend
* Front end
* Admin Panel

1. About Backend:

Develop backend using nodejs, express and mongodb for database alongwith firebase to store files.

Dependencies:

yarn add express mongoose nodemon dotenv jsonwebtoken morgan crypto-js

1. About Frontend:

Frontend is build using reactjs and scss along with material ui designing tool

Repo Link:

1. About Admin Panel:

It is also build using reactjs

Repo Link:

App Link

SIGNUP AND LOGIN API:

const express = require('express');

const router = express.Router();

const CryptoJS = require('crypto-js')

const jwt = require('jsonwebtoken')

const User = require('../schema/User');

//Register

router.post('/register', (req, res, next)=>{

    const newUser = new User({

        username: req.body.username,

        email: req.body.email,

        password: CryptoJS.AES.encrypt(req.body.password, process.env.SECRET\_KEY).toString(),

    });

    newUser

    .save()

    .then(result=>{

        res.status(201).json(result)

    })

    .catch(error => {

        res.status(500).json(error)

    })

})

//login

router.post('/login', (req, res)=>{

    User.findOne({email: req.body.email})

        .exec()

        .then(user => {

            if (!user){

                return res.status(401).json({

                    message: 'Auth Failed'

                })

            }

            //Decrypting password for comparision

            const bytes = CryptoJS.AES.decrypt(user.password, process.env.SECRET\_KEY);

            const originalPassword = bytes.toString(CryptoJS.enc.Utf8);

            if(originalPassword !== req.body.password){

                return res.status(401).json({

                    message: 'Please Enter Correct Password'

                })

            }else{

                const accessToken = jwt.sign(

                    {id: user.\_id, isAdmin: user.isAdmin},

                    process.env.SECRET\_KEY, {expiresIn: '1h'}

                    )

                //destructure to avoid getting password and \_\_v in response

                const {password, \_\_v, ...info} = user.\_doc;

                res.status(200).json({...info, accessToken})

            }

        })

        .catch(error=>{

            res.status(500).json(user)

        })

})

module.exports = router