**Assignment : 05-Sep-2023**

**Middleware to check validity of JWT token**

const jwt=require("jsonwebtoken");

const RouteGuard=(req,res,next)=>{

try{

const isValid=jwt.verify(req.headers.authorization, "yoursecretkey");

if(isValid){

next();

}

else{

res.status.send({msg:"Unauthorized"});

}

}

catch(e){

res.status(500).send({msg:"Something went wrong"});

}

}

module.exports=RouteGuard;

**Middleware for form data validation**

const Validator=(req,res,next)=>{

const {username, password}=req.body;

if(username==null || password==null){

console.log("empty data");

res.status(400).send({msg:"empty data"});

}

else{

var mailformat= /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;

// username@domain.com

if(username.match(mailformat)){

next();

}

else{

console.log("invalid username or password");

res.status(400).send({msg: "username is invalid format"})

}

}

}

module.exports=Validator;

**Rate limiting middleware to allow limited number of request to specific route in given interval of time**

const setRateLimit = require("express-rate-limit");

// Rate limit middleware

const rateLimitMiddleware = setRateLimit({

windowMs: 60 \* 1000,

max: 5,

message: "You have exceeded your 5 requests per minute limit.",

headers: true,

});

module.exports = rateLimitMiddleware;

**Try File system operations to create, read, delete, update files.**

Create File:

Create file using writeFile():

fs.writeFile("./writefs.txt", "Hello", (err, file)=>{

if(err){

console.log(err);

}

console.log("Saved!");

});

Create file using open():

fs.open("./openfs.txt", "w", (err, file)=>{

if(err){

console.log(err);

}

console.log("Saved");

});

Create file using appendFile():

fs.appendFile("./appendfs.txt",

"file not exits, so craeted and data appened successfully using appendFile method",

(err)=>{

if(err){

console.log(err);

}

console.log("data appended successfully to the appendfs.tx t file");

});

Read File:

fs.readFile("./readfs.txt","utf8",(err, data)=>{

if(err){

console.log(err);

}

console.log(data);

});

Delete File:

fs.unlink("./writefs.txt", (err)=>{

if(err){

console.log(err);

}

console.log("File deleted!");

});

Update File:

Update file using appendFile:

fs.appendFile("./readfs.txt", "data appened successfully using appendFile method", (err,data)=>{

if(err){

console.log(err);

}

console.log("data appended successfully to the readfs.tx t file");

});

Update file using writeFile:

fs.writeFile("./writefs.txt", "Hi", (err, file)=>{

if(err){

console.log(err);

}

console.log("Saved!");

});