Full stack practice 3

const express = require('express');

const app = express();

const PORT = 3000;

app.use(express.json());

let seats = {};

const TOTAL\_SEATS = 10;

for (let i = 1; i <= TOTAL\_SEATS; i++) {

seats[i] = { status: 'available', lockExpires: null, lockedBy: null };

}

function cleanupLocks() {

const now = new Date();

for (let seat in seats) {

if (seats[seat].status === 'locked' && seats[seat].lockExpires <= now) {

seats[seat] = { status: 'available', lockExpires: null, lockedBy: null };

}

}

}

app.get('/seats', (req, res) => {

cleanupLocks();

res.json({ seats });

});

app.post('/seats/lock', (req, res) => {

cleanupLocks();

const { seatNumber, user } = req.body;

if (!seatNumber || !user || !seats[seatNumber]) {

return res.status(400).json({ success: false, message: 'Invalid seat number or user' });

}

const seat = seats[seatNumber];

if (seat.status === 'booked') {

return res.status(400).json({ success: false, message: 'Seat already booked' });

}

if (seat.status === 'locked') {

return res.status(400).json({ success: false, message: `Seat is locked by ${seat.lockedBy}` });

}

const lockDuration = 60 \* 1000;

seat.status = 'locked';

seat.lockExpires = new Date(Date.now() + lockDuration);

seat.lockedBy = user;

res.json({ success: true, message: `Seat ${seatNumber} locked for ${user} for 1 minute` });

});

app.post('/seats/confirm', (req, res) => {

cleanupLocks();

const { seatNumber, user } = req.body;

if (!seatNumber || !user || !seats[seatNumber]) {

return res.status(400).json({ success: false, message: 'Invalid seat number or user' });

}

const seat = seats[seatNumber];

if (seat.status !== 'locked' || seat.lockedBy !== user) {

return res.status(400).json({ success: false, message: 'Seat not locked by you or lock expired' });

}

seat.status = 'booked';

seat.lockExpires = null;

seat.lockedBy = null;

res.json({ success: true, message: `Seat ${seatNumber} booked successfully by ${user}` });

});

// Start server

app.listen(PORT, () => {

console.log(`Ticket Booking API running at http://localhost:${PORT}`);

});