const express = require('express');

const router = express.Router();

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

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

// Signup

router.post('/signup', async (req, res) => {

try {

const { email, password, name } = req.body;

const existing = await User.findOne({ email });

if (existing) return res.status(400).json({ error: 'Email already in use' });

const hashed = await bcrypt.hash(password, 10);

const user = new User({ email, password: hashed, name });

await user.save();

const token = jwt.sign({ id: user.\_id }, process.env.JWT\_SECRET, { expiresIn: '7d' });

res.json({ token, user: { id: user.\_id, email, name } });

} catch (err) {

res.status(500).json({ error: 'Signup failed' });

}

});

// Login

router.post('/login', async (req, res) => {

try {

const { email, password } = req.body;

const user = await User.findOne({ email });

if (!user) return res.status(400).json({ error: 'Invalid credentials' });

const valid = await bcrypt.compare(password, user.password);

if (!valid) return res.status(400).json({ error: 'Invalid credentials' });

const token = jwt.sign({ id: user.\_id }, process.env.JWT\_SECRET, { expiresIn: '7d' });

res.json({ token, user: { id: user.\_id, email, name: user.name } });

} catch (err) {

res.status(500).json({ error: 'Login failed' });

}

});

module.exports = router;