```
const express = require('express');
const router = express.Router();
const fs = require('fs');
const jwt = require('jsonwebtoken');
const bcrypt = require('bcryptjs');
const config = require('../config');

const adminDb = config.adminDb;

// Admin login
router.post('/login', (req, res) => {
  const { email, password } = req.body;
  if (!fs.existsSync(adminDb)) return res.status(500).json({ success: false, message: 'Admin not configured' });
  const admin = JSON.parse(fs.readFileSync(adminDb));
  if (admin.email !== email) return res.status(401).json({ success: false, message: 'Unauthorized' });
  if (!bcrypt.compareSync(password, admin.passwordHash)) return res.status(401).json({ success: false, message: 'Unauthorized' });

  const token = jwt.sign({ email }, process.env.JWT_SECRET || 'dev-secret', { expiresIn: '8h' });
  res.json({ success: true, token });
});

// Middleware to require token
```

```
function verify(req, res, next) {
  const h = req.headers.authorization;
  if (!h) return res.status(401).json({ success: false, message: 'Missing token' });
  const token = h.split(' ')[1];
  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET || 'dev-secret');
    req.admin = decoded;
    next();
  } catch (err) {
    return res.status(401).json({ success: false, message: 'Invalid token' });
  }
}

// Orders list
router.get('/orders', verify, (req, res) => {
  const ordersFile = config.dbPath;
  let orders = [];
  if (fs.existsSync(ordersFile)) orders = JSON.parse(fs.readFileSync(ordersFile));
  res.json({ success: true, orders });
});

module.exports = router;
```