const express = require('express');

const { Pool } = require('pg');

const app = express();

const PORT = process.env.PORT || 3000;

const pool = new Pool({

connectionString: process.env.DATABASE\_URL,

ssl: { rejectUnauthorized: false }

});

async function initDB(){

const client = await pool.connect();

try {

await client.query(`

CREATE TABLE IF NOT EXISTS tasks (

task\_id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

status VARCHAR(20) DEFAULT 'pending',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

`);

console.log('DB initialized');

} finally {

client.release();

}

}

app.use(express.static('public'));

app.use(express.json());

app.get('/api/tasks', async (req, res) => {

const result = await pool.query('SELECT \* FROM tasks ORDER BY task\_id DESC');

res.json(result.rows);

});

app.post('/api/tasks', async (req, res) => {

const { title } = req.body;

const result = await pool.query(

'INSERT INTO tasks (title) VALUES ($1) RETURNING \*', [title]

);

res.status(201).json(result.rows[0]);

});

app.put('/api/tasks/:id/toggle', async (req, res) => {

const id = req.params.id;

const result = await pool.query(

`UPDATE tasks SET status = CASE WHEN status='pending' THEN 'done' ELSE 'pending' END

WHERE task\_id=$1 RETURNING \*`, [id]

);

if (result.rowCount === 0) return res.status(404).send('Not found');

res.json(result.rows[0]);

});

initDB().catch(err => {

console.error('DB init error', err);

process.exit(1);

});

app.listen(PORT, () => console.log(`✅ Server running on port ${PORT}`));