const express = require('express');

const app = express();

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

// Register user

[app.post](http://app.post/)('/register', async (req, res) => {

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

  // Validate input

  if (!email || !password || !confirmPassword) {

    return res.status(400).send({ message: 'Please fill in all fields' });

  }

  // Hash password

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

  // Create user

  const user = await User.create({ email, password: hashedPassword });

  // Generate token

  const token = jwt.sign({ userId: [user.id](http://user.id/) }, process.env.SECRET\_KEY, {

    expiresIn: '1h',

  });

  res.send({ userId: [user.id](http://user.id/), email: user.email, token });

});

// Login user

[app.post](http://app.post/)('/login', async (req, res) => {

  const { email, password } = req.body;

  // Validate input

  if (!email || !password) {

    return res.status(400).send({ message: 'Please fill in all fields' });

  }

  // Find user

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

  if (!user) {

    return res.status(401).send({ message: 'Invalid email or password' });

  }

  // Compare passwords

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

  if (!isValidPassword) {

    return res.status(401).send({ message: 'Invalid email or password' });

  }

  // Generate token

  const token = jwt.sign({ userId: [user.id](http://user.id/) }, process.env.SECRET\_KEY, {

    expiresIn: '1h',

  });

  res.send({ userId: [user.id](http://user.id/), email: user.email, token });

});