if (process.env.NODE\_ENV !== 'production') {

    require('dotenv').config()

  }

  const express = require('express')

  const app = express()

  const bcrypt = require('bcrypt')

  const passport = require('passport')

  const flash = require('express-flash')

  const session = require('express-session')

  const methodOverride = require('method-override')

  const initializePassport = require('./passport-config')

  initializePassport(

    passport,

    email => users.find(user => user.email === email),

    id => users.find(user => user.id === id)

  )

  const users = []

  app.set('view-engine', 'ejs')

  app.use(express.urlencoded({ extended: false }))

  app.use(flash())

  app.use(session({

    secret: process.env.SESSION\_SECRET,

    resave: false,

    saveUninitialized: false

  }))

  app.use(passport.initialize())

  app.use(passport.session())

  app.use(methodOverride('\_method'))

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

    res.render('index.ejs', { name: req.user.name })

  })

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

    res.render('login.ejs')

  })

  app.post('/login', checkNotAuthenticated, passport.authenticate('local', {

    successRedirect: '/',

    failureRedirect: '/login',

    failureFlash: true

  }))

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

    res.render('register.ejs')

  })

  app.post('/register', checkNotAuthenticated, async (req, res) => {

    try {

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

      users.push({

        id: Date.now().toString(),

        name: req.body.name,

        email: req.body.email,

        password: hashedPassword

      })

      res.redirect('/login')

    } catch {

      res.redirect('/register')

    }

  })

  app.delete('/logout', (req, res) => {

    req.logOut()

    res.redirect('/login')

  })

  function checkAuthenticated(req, res, next) {

    if (req.isAuthenticated()) {

      return next()

    }

    res.redirect('/login')

  }

  function checkNotAuthenticated(req, res, next) {

    if (req.isAuthenticated()) {

      return res.redirect('/')

    }

    next()

  }

  app.listen(3000)