Index.ejs

|  |
| --- |
| const express = require('express')  const morgan = require('morgan')  const path = require('path')  const { emit } = require('process')  let sqlite3 = require('sqlite3').verbose()  let db = new sqlite3.Database('10\_1P')  const port = 3000  const app = express()  app.use(express.static('public\_html'))  app.use(morgan('common'))  //configuring express to use body-parser as middle-ware.  app.use(express.urlencoded({ extended: false }));  app.set('views', path.join(\_\_dirname, '\\public\\views'))  app.set('view engine', 'ejs')  app.get('/', (req, res) => {      res.render('index', {          title: 'dKin Membership'      })  })  app.post('/submitmembership', (req, res) => {      let body = req.body      console.log(body)      db.serialize(() => {          db.run('INSERT INTO USER(fname, sname, email, mobile, numcaps, favourite, comment) VALUES(?,?,?,?,?,?,?)', [body.firstname, body.surname, body.email, body.mobileNumber, body.inputNumCaps, body.capstyle, body.comments], function (err) {              if (err) {                  return console.log('insert failed, err: ', err.message)              }              console.log('insert success')          })      })      db.close      res.status(200).render('/')  })  app.get('/membershipdetails', (req, res) => {      db.serialize(() => {          db.all('SELECT \* FROM User', function (err, rows) {              if (err) {                  return console.log('query failed, error msg: ', err.message)              } else {                  console.log('query success.')                  console.log(JSON.stringify(rows))                  res.render('membershipdetails', {                      title: 'Ice Cream Review',                      rows: rows                  })              }          })      })      db.close;  })  app.use((req, res) => {      res.render('404', {          title: 'Ice Cream Review',          message: `no handler for request ${req.url}`,          url: req.url      })  })  app.use((err, req, res, next) => {      let status = err.status || 500      res.status(status).send(`error code ${status} , error msg ${err.toString()}`)  })  app.listen(port, () => {      console.log(`Web server running at: http://localhost:${port}`);      console.log(`Type Ctrl+C to shut down the web server`);  }) |