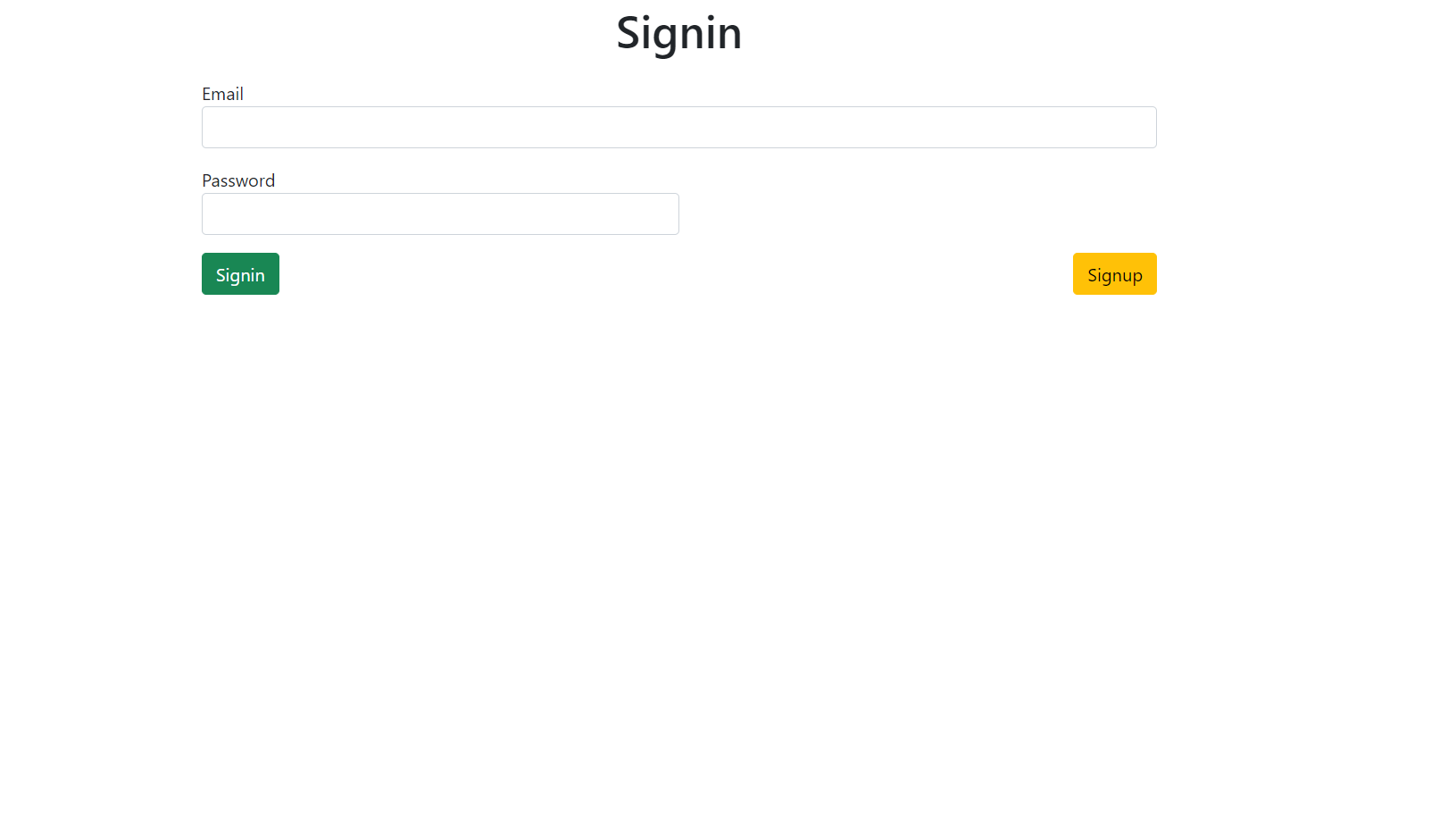
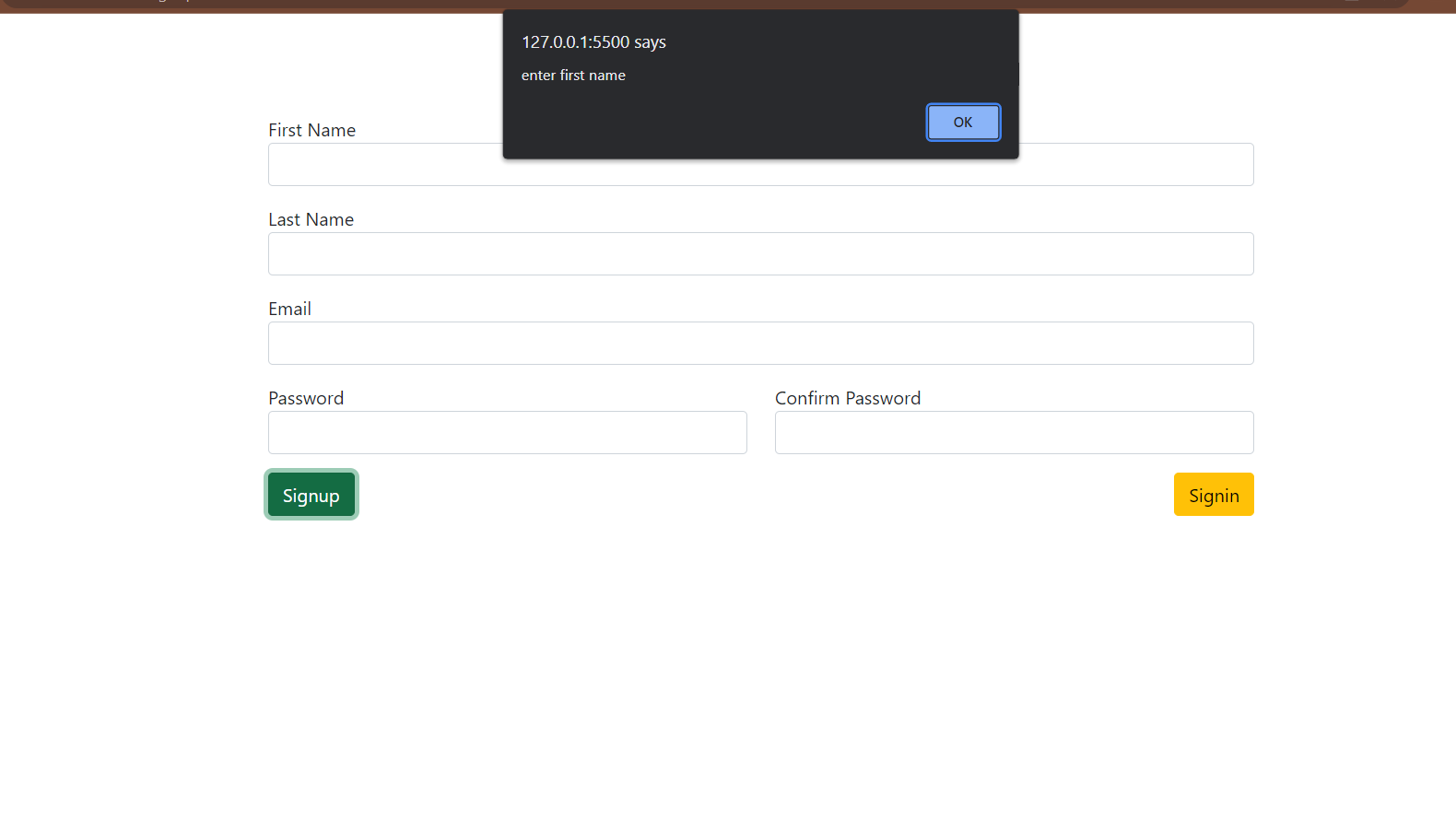
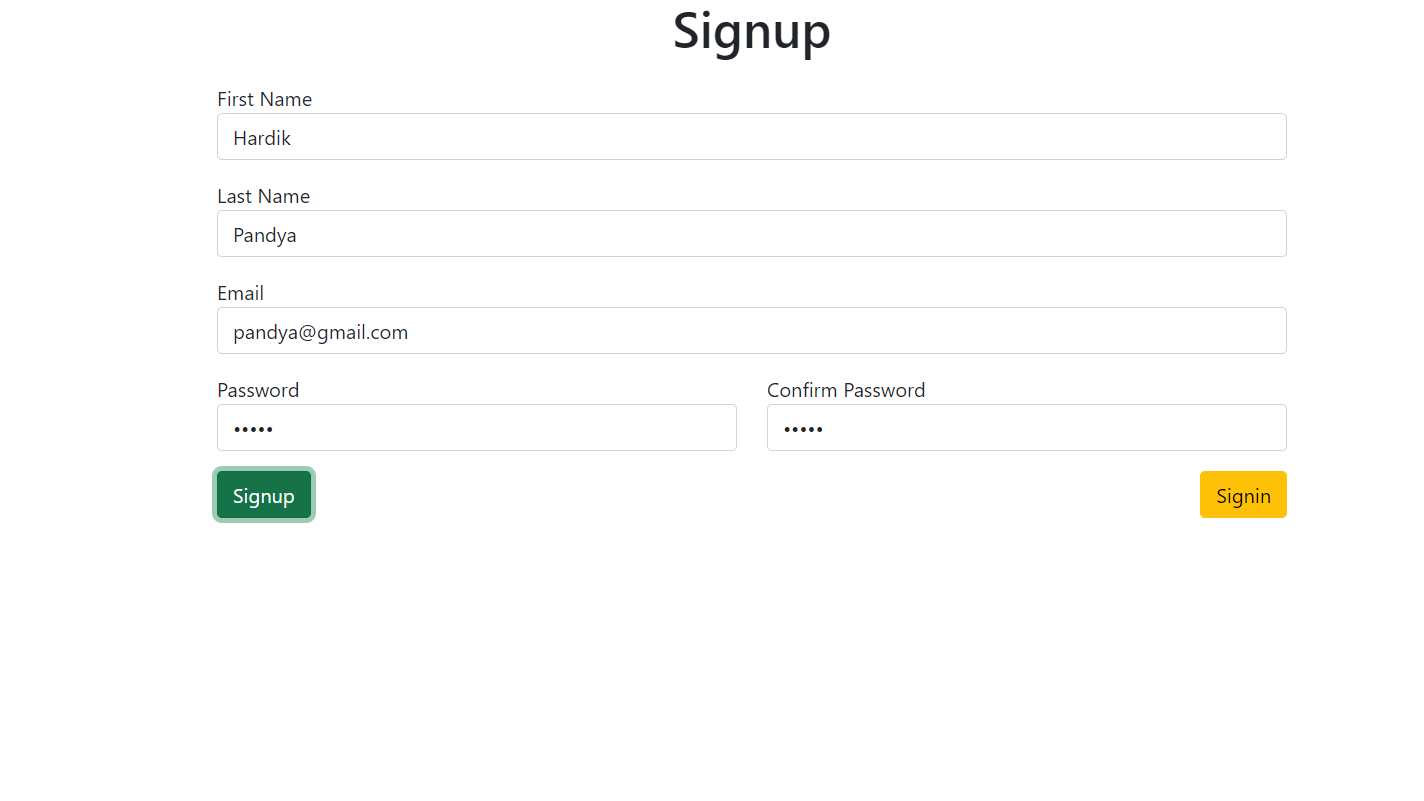
Step

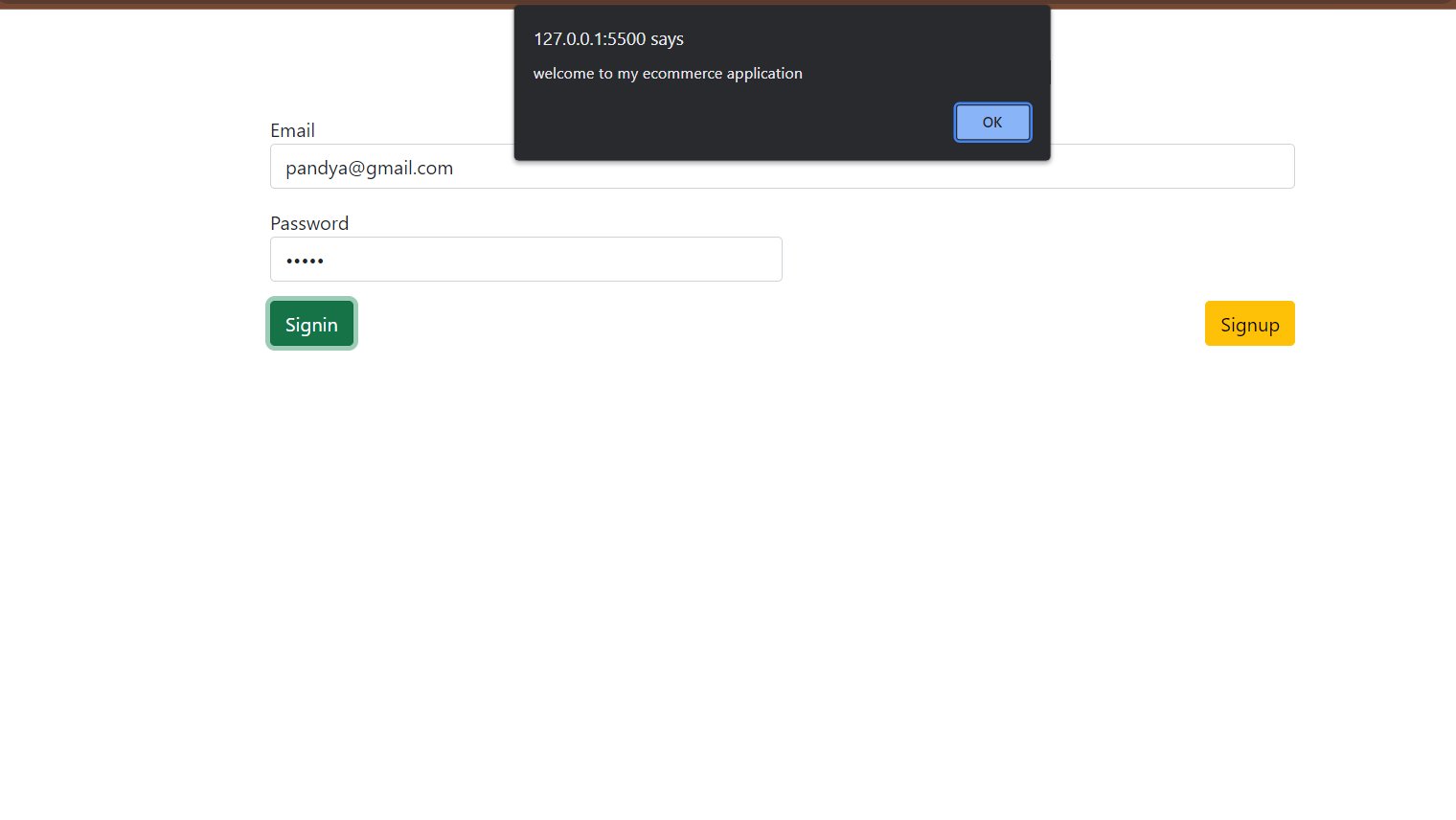
Page1



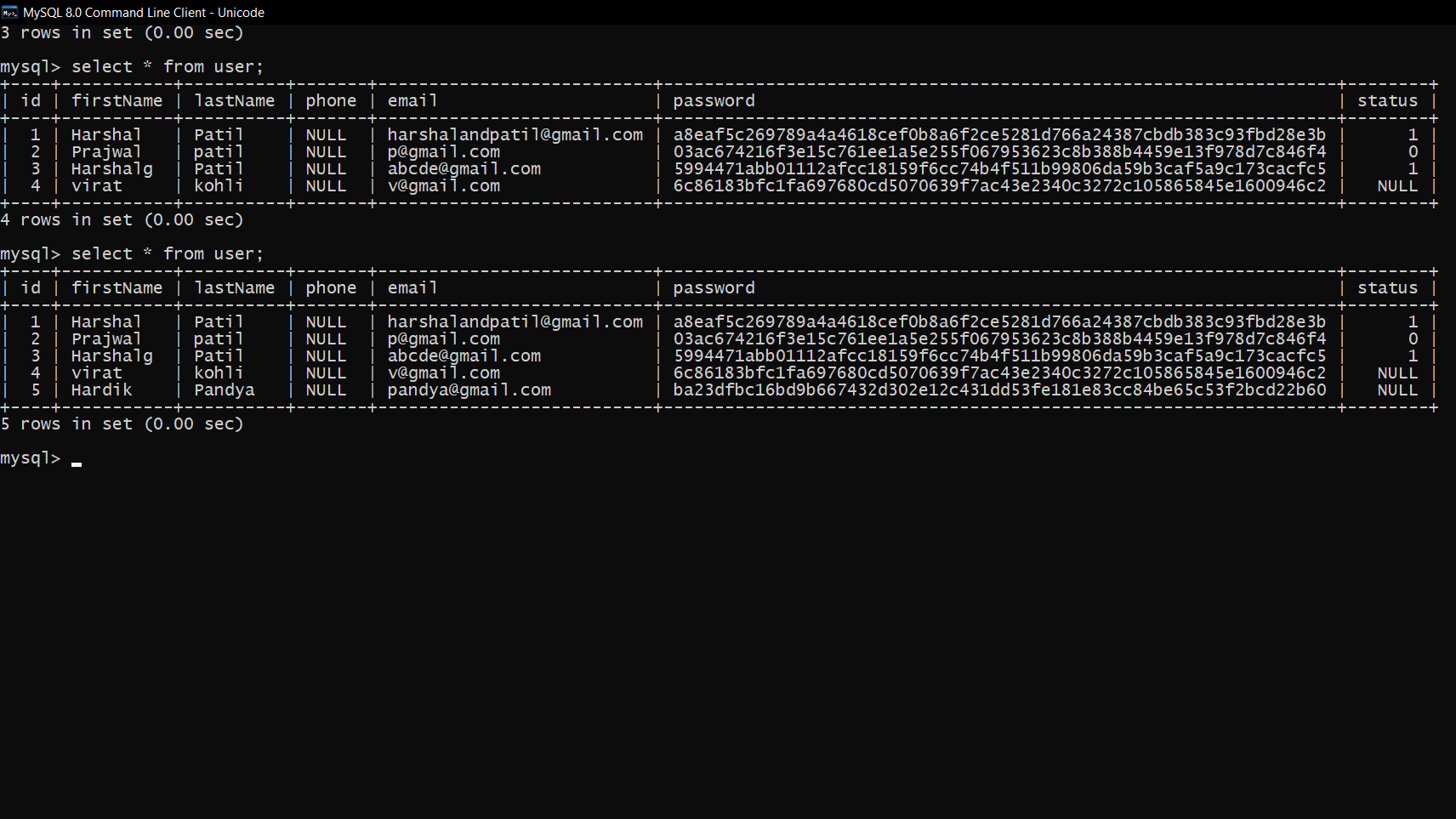
Page with validation functionality



Signup with user



Successfull Signed in with user



Database table with password security

Backend Server.js

const express = require("express");

const bodyParser = require("body-parser");

const jwt = require("jsonwebtoken");

const config = require("./config");

const cors = require("cors"); // npm install cors

// list of routers

const routerUser = require("./routes/user");

const app = express();

// enable frontend application to call the APIs

app.use(cors("\*"));

app.use(bodyParser.urlencoded());

app.use(bodyParser.json());

app.use((request, response, next) => {

  // skip checking the token for following APIs

  // signin and signup

  if (

    request.url == "/user/signin" ||

    request.url == "/user/signup" ||

    request.url.startsWith("/user/verify") ||

    request.url.startsWith("/user/status")

  ) {

    // skip checking the token

    next();

  } else {

    // get the token from headers

    const token = request.headers["token"];

    try {

      // verify if the token is original or intact

      const payload = jwt.verify(token, config.secret);

      // get id from the token

      // add the user id in the request object so that it can be used

      // in ever other APIs

      request.id = payload["id"];

      // call the next handler

      next();

    } catch (ex) {

      response.send({

        status: "error",

        error: "unauthorized access",

      });

    }

  }

});

// add routers

app.use(routerUser);

app.get("/", (request, response) => {

  response.send("welcome to ecommerce application");

});

app.listen(3000, "0.0.0.0", () => {

  console.log(`server started on port 3000`);

});

User.js

const express = require("express");

const db = require("../db");

const crypto = require("crypto-js");

const mailer = require("../mailer");

const jwt = require("jsonwebtoken");

const config = require("../config");

const utils = require("../utils");

const router = express.Router();

router.get("/user/profile", (request, response) => {

  const statement = `select firstName, lastName, email, phone from user where id = ${request.id}`;

  db.execute(statement, (error, data) => {

    response.send(utils.createResult(error, data));

  });

});

router.post("/user/signup", (request, response) => {

  const { firstName, lastName, email, password } = request.body;

  console.log(request.body);

  // encrypt the password

  const encryptedPassword = "" + crypto.SHA256(password);

  // by default every user will be non-verified

  const statement = `insert into user (firstName, lastName, email, password) values (

    '${firstName}', '${lastName}', '${email}', '${encryptedPassword}'

  )`;

  db.execute(statement, (error, data) => {

    const result = utils.createResult(error, data);

    if (!error) {

      // mailer.sendEmail(

      //   'signup.html',

      //   'welcome to ecommerce application',

      //   email,

      //   (error, info) => {

      //     response.send(result)

      //   }

      // )

      response.send(result);

    } else {

      response.send(result);

    }

  });

});

router.post("/user/signin", (request, response) => {

  const { email, password } = request.body;

  // encrypt the password

  const encryptedPassword = "" + crypto.SHA256(password);

  const statement = `select id, firstName, lastName, email, phone from user

      where email = '${email}' and password = '${encryptedPassword}'`;

  db.execute(statement, (error, users) => {

    const result = {

      status: "",

    };

    if (error != null) {

      // error while executing statement

      result["status"] = "error";

      result["error"] = error;

    } else {

      if (users.length == 0) {

        // user does not exist

        result["status"] = "error";

        result["error"] = "User does not exist";

      } else {

        const user = users[0];

        const payload = { id: user["id"] };

        const token = jwt.sign(payload, config.secret);

        result["status"] = "success";

        result["data"] = {

          token: token,

          firstName: user["firstName"],

          lastName: user["lastName"],

          email: user["email"],

          phone: user["phone"],

        };

      }

      response.send(result);

    }

  });

});

module.exports = router;

database file

create database mern;

use mern;

create table user (id integer primary key auto\_increment, firstName varchar(15), lastName varchar(15), phone varchar(12), email varchar(50), password varchar(100));

ALTER TABLE user ADD UNIQUE (email);

-- status

-- 0: non-verified, 1: active, 2: suspended

ALTER TABLE user ADD COLUMN status int(1);