Commands:-

npm init

npm i express

npm i cors

npm i mongoose

npm i jsonwebtoken

npm i cookie-parser

npm i dotenv

npm i bcryptjs

npm i nodemon

Notes

* In mongodb, for each new value added

There is a (unique identifier) generated automatically which is (\_id)

* On haveing 2 connected tables in mongodb ,

So here (msgByUserId) will hold the (\_id) of the other table row.

const messageSchema = new mongoose.Schema(

  {

    .....

msgByUserId: {

      type: mongoose.Schema.ObjectId,

      required: true,

      ref: "User",

    },

  },

  {

    timestamps: true,

  }

);

* In the previous case, If I want to return a nested object of data in 1st table with data of 2nd table inside the key that hold the (\_id) value
* MessageModel.findOne({ \_id: messageId })
* .populate("msgByUserId")
* .then((message) => {
* console.log(message);
* });

Ex:- this returns

{

    \_id: "message123",

    text: "Hello!",

    msgByUserId: { // Replaced with actual User document

      \_id: "user456",

      name: "John Doe",

      email: "john@example.com",

    }

  }

* Here I retrieve data again from the database

Then send it to the user.

(not send the updated data I made, because may any error happens while updating)

const updateUser = await UserModel.updateOne(

  { \_id: user.\_id },

  { name, profile\_pic }

);

// Get data from DB after updating

const userInfomation = await UserModel.findById(user.\_id);

return response.json({

  message: "user update successfully",

  data: userInfomation,

// Send data I got from DB after update (not updateUser)

  success: true,

});

* Here is a (regular expression) that used for search.
* const query = new RegExp(search, "i", "g");

1. search => text for search in data set or in DB
2. i => Case insensitive
3. g => Global (search in all data set, not only take the first match you got, but return all matches)

* Z
* Z
* Z