(Validations and Sanatization)

Till now we haven't put any checks in our schema go to user model and add strict checks if it doesn't met we will not save the data

you can add.

```
Const userSchema = new Mongoose.Schema({

    firstname : {
    type : String,
    require : true, ← it will make this data as mandatory.

    },
    lastname : {
     type : String,

    },

    emailid : {
      type : String,
      require : true,
      unique : true, ← it will set data unique.
    },
    gender : {
      type : String
    }

},{});
```

To get array of data from user we do

type : [String] ← it can be helpfull when taking data such as Skills, Hobby, etc.

to get or set default value we use

default : "default value"

To keep our email address data in lower case. we can use this string

Lowercase : true. ←
trim : true ← to trim all the white space. from

Date :
min : date, create a validator
max : date, validator.

Match : regexp, create validator.
minLength :
maxLength :

if String = minLength
if number = min

How to create a custom Validation

~~validate (value) {~~
validate (value {
# if (["male", "female", "others"].include(value)){
throw new Error("Gender data is not valid");
};

↑ when we are creating new object

By default this validation is only work only the time of creation, it will not work at patch, we have to enable it for patch or update.

we have to give = runValidators: true in the API

How to add when user has been register.

V1

⇒ { timeStamp: true } pass another argument in Schema.

its good and easy way

↑ it gives us created at and updated at

or

createdAt: {
type: Date
}

{All validators we have used} in this learning

type: String / number
minLength: 3
maxLength: 15
required: true
trim: true
min: 15
max: 100
lowercase: true
uppercase: true
default: "default value"
unique: "true"
validate(value) {
if (!['male', 'female', 'other'].includes(value)) {
throw new Error("enter valid data");
}
}

Pass as second argument.
{timestamp: true}

skills: {
type: [String],
max: 10,
}

API Level Validation (Data Sanitization)

API validation, suppose we have data now in Database & but at the time of login I don't need every data, I just need username, password or email id and password.

and ~~soft~~ suppose we also want to not allow user to change email id ∈ for security reason

How to add this check.

```
Const Allowed_Update = ["Photo", "About", "gender", "age",
                        "Skills"];

const isUpdateAllowed = object.keys(data).every((k) => {
    return Allowed_update.includes(k) });

if (!isUpdateAllowed) {
    throw new Error("update not allowed");
```

Sanitization

Important thing to do.

Should even user id is not allowed.

```
Const allowedUpdate = ["user_id", "age", "gender", "About",
                       "photo"];
const isallowedupdate = object.keys(data).every((k) =>
        allowedupdate.includes(k)
    );
if (!isallowedupdate) {
    throw new Error("update not allowed").
    };
```

We will pass userid via params.

in url

{userid = req.params?.userid}

and in app.post ("/user/:userid", (req, res) => {
use this.

~~or~~ : Skill ~~up~~ validation in database level.

```
skills : { type : [String]
validate(value) {
    if(value.length > 10) {
        throw new Error("skill should be not more
                         than 10")
    }
}
```

Email validator (Still we have some bug)
like if we pass anything or string on
& ~~user~~. emailid it will save it.

it is tough, so we will take npm library
[validator.library]

use this ⇐ it's easy.

Install Validator.

npm i validator.

We will first input this and we will do this at db level, validation.

• then

```
validate(value) {
if (!validator.isEmail(value)) {
    throw new Error("in valid email address"
                    + value);
```

You can use validator library it will move validation bit much easier.
Read Doc.

NEVER TRUST

Req. body

Sanatize and Validate everything.