Application data type

|  |  |
| --- | --- |
| User Schema |  |
| name | string |
| email | string |
| password | string |
| social | An array of social objects |
| documents | An array of file objects |
| cover images - up to 5 | An array of file objects |
| profile photo - 1 | File Object |
| background image - 1 | File Object |
| bioinfo | String |

|  |  |
| --- | --- |
| File Schema |  |
| File name | string |
| description | string |
| path | string |
| owner | object |
| tag | An array of objects |
| dateCreated | Date |
| dateModified | Date |
| Highlighted (no more than 9 files) | boolean |

|  |  |
| --- | --- |
| Tag Schema |  |
| id | uniqueID that is sortable (integer) |
| name | string |

Each user can have multiple files, but each file belongs to exactly one user.

const userSchema = new Schema({

name: {type : String, required: true},

email: {type : String, required: true, unique: true},

password: { type : String, required: true, minlength: 6},

social: [{ type: mongoose.Schema.Types.ObjectId, required: true, ref: 'Social'}],

documents: [{ type: mongoose.Schema.Types.ObjectId, required: true, ref: 'File'}],

bioinfo: { type: String, required: true},

officeAddress: { type: String, required: false},

mobile: { type: String, required: false},

semail: { type: String, required: true},

photos: [{ type: mongoose.Types.ObjectId, required: true, ref: 'Photos'}],

tags: [{ type: mongoose.Types.ObjectId, required: true, ref: 'Tag'}]

});

const fileSchema = new Schema({

name: {type : String, required: true}, // file name

description: {type : String, required: true},

path: {type : String, required: true},

owner: {type: mongoose.Schema.Types.ObjectId, required: true, ref: 'User'},

dateCreated: {type : Date, default: Date.now},

dateModified: {type : Date},

highlighted: {type: Boolean, required: true},

achivement: {type: Boolean, required: true},

institution: {type : String},

dateAchieved: {type : Date},

tags: [{type: mongoose.Schema.Types.ObjectId, ref: 'Tag'}]

});

**const tagSchema = new Schema({**

**name: {type : String, required: true}, // tag name**

**color: {type : String, required: true},**

**owner: { type: mongoose.Schema.Types.ObjectId, required: true, ref: 'User'}, // user object id, user has multiple tags, each tag has one user**

**documents: [{ type: mongoose.Schema.Types.ObjectId, required: true, ref: 'File'}] // many-to-many relationship, each tag can have multiple file objectid, each file can have multiple tags**

**});**

**Get method** {

api/manage: needs cover image, profile image and bio info.

api/home: needs cover image, profile image and bio info.

api/files: returns all files

api/files/:id return specific file.

api/bioinfo: returns bioinfo

api/coverimage

api/backgroundimage

api/users return a list of users in database.

}

**Post method** {

api/users/signup

* Name
* Password
* Email
* Documents
* Bioinfo
* semail

api/submit/type?=

-document

-coverimage

-backgroundimage

-bioinfo

Data -> file

api/users/login

* Email
* password

}

**Put method** {

api/bioinfo update bio info

api/changeUsername

api/changeEmail

}

**Patch method** {

api/manage/:fileid/update

}