Technisch ontwerp

Naam:	Brighton van Rouendal
Leerlingnummer:	2079016
Datum:	31-05-2022
Versie:	v1.0

Inhoudsopgave

Technische specificaties

De database wordt gemaakt in MongoDB en wordt mee gecommuniceerd via Mongoose (NodeJS module). MongoDB is een NoSQL database dat soort van JSON gebruikt om de documenten te behouden. Al is MongoDB een NoSQL database heb ik nog wel een relatie gemaakt tussen schema's.

(Praat over mongoose (misschien server))

Deze relaties zijn one-to-many tot nu toe. Tegens is dat niet helemaal waar de relaties worden gemaakt door een array in de parent schema met alle id’s van children en de children hebben de id van de parent in hun schema.

Er zijn ook een paar pre validations bij toevoeging of toepassing. Deze zijn voor het sanitise van de html zodat gebruikers geen mogelijke gevaarlijke html/javascript kunnen sturen naar de server en verdere gebruikers. Der is tevens ook een pre validations voor het maken van een zogenaamde slug, een slug is de titel/naam van een item zo veranderd dat het kan worden gebruikt in de url voor de webpagina. Op de manier ik ik het doe wordt de objectid van het item voor de titel/naam geplakt zodat er meerdere boeken of chapters met dezelfde naam kunnen komen omdat de objectid altijd unique is.

De relaties liggen zo: De User kan meerdere Boeken hebben maar één Boek kan niet meerdere Users hebben. Dit is hetzelfde voor Boeken en Chapters dus, Een Boek kan meerdere Chapters hebben maar een Chapter kan maar één boek hebben.
Dit is tot nu toe de database schema en relaties later zouden reviews en comments komen voor boeken en chapters in die order. met dezelfde soort relatie dat hierboven is aangegeven met de toevoeging voor een link naar een User.

Data models (Schema's)

Boeken schema

```
const bookSchema = new mongoose.Schema(
    {
        title: {
            type: String,
            required: true,
        },
        description: {
            type: String,
            maxlength: 500,
        },
        chapters: [
            {
                type: mongoose.Schema.Types.ObjectId,
                ref: 'Chapter',
            },
        ],
        published: {
            type: Boolean,
            default: false,
        },
        slug: {
            type: String,
            required: true,
            unique: true,
        },
        sanitizedHtml: {
            type: String,
            required: true,
        },
    },
    { timestamps: true, collection: 'books' }
);
```

Chapters schema

```
const chapterSchema = new mongoose.Schema(
    {
        title: {
            type: String,
            required: true,
        },
        text: {
            type: String,
            required: true,
            maxlength: 1000000,
        },
        book_id: {
            type: mongoose.Schema.Types.ObjectId,
            ref: 'Book',
        },
        published: {
            type: Boolean,
            default: false,
        },
        slug: {
            type: String,
            required: true,
            unique: true,
        },
        sanitizedHtml: {
            type: String,
            required: true,
        },
    },
    { timestamps: true, collection: 'chapters' }
);
```

User schema

```
const userSchema = new mongoose.Schema(
    {
        nick_name: {
            type: String,
            required: true,
            unique: true,
            minlength: 4,
            maxlength: 24,
        },
        e_mail: {
            type: String,
            required: true,
            unique: true,
        },
        password: {
            type: String,
            required: true,
        },
        dob: {
            type: Date,
            required: true,
        },
    },
    { timestamps: true, collection: 'users' }
);
```