

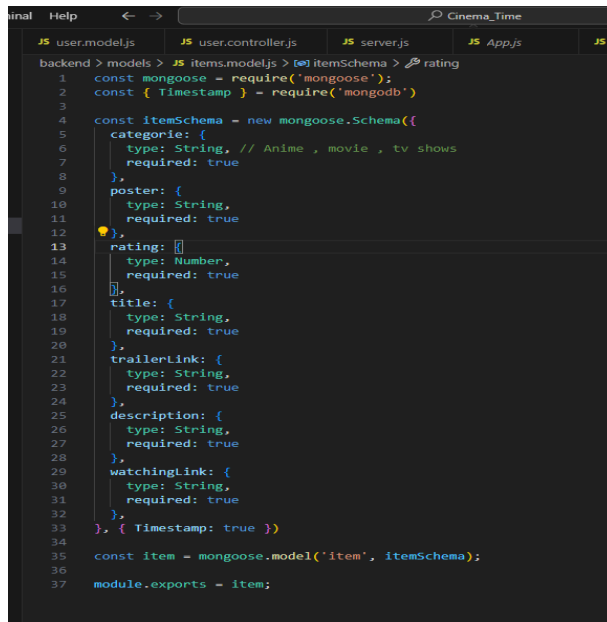
# DATABASE SCHEMA DOCUMENTATION

## 1. Item Schema:

### ➤ Description:

The Item schema represents movie, anime, or TV show entries in the Cinema Time database. It includes details such as category, poster image URL, rating, title, trailer link, description, and watching link.

### ➤ schema



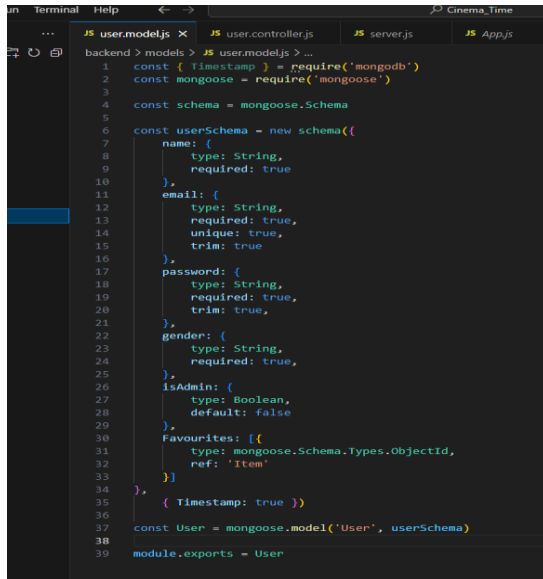
```
backend > models > JS items.model.js > itemSchema > rating
1 const mongoose = require('mongoose');
2 const { Timestamp } = require('mongodb')
3
4 const itemSchema = new mongoose.Schema({
5   categorie: {
6     type: String, // Anime , movie , tv shows
7     required: true
8   },
9   poster: {
10    type: String,
11    required: true
12  },
13   rating: {
14     type: Number,
15     required: true
16   },
17   title: {
18     type: String,
19     required: true
20   },
21   trailerLink: {
22     type: String,
23     required: true
24   },
25   description: {
26     type: String,
27     required: true
28   },
29   watchingLink: {
30     type: String,
31     required: true
32   },
33 }, { Timestamp: true })
34
35 const item = mongoose.model('item', itemSchema);
36
37 module.exports = item;
```

## 2. user Schema:

### ➤ Description:

The User schema represents user accounts in the cinema Time database. It includes details such as name, email, password, gender, isAdmin (admin status), and a list of favorite items.

### ➤ Schema



```
1 const { Timestamp } = require('mongoose')
2 const mongoose = require('mongoose')
3
4 const schema = mongoose.Schema
5
6 const userSchema = new Schema({
7   name: {
8     type: String,
9     required: true
10  },
11  email: {
12    type: String,
13    required: true,
14    unique: true,
15    trim: true
16  },
17  password: {
18    type: String,
19    required: true,
20    trim: true,
21  },
22  gender: {
23    type: String,
24    required: true,
25  },
26  isAdmin: {
27    type: Boolean,
28    default: false
29  },
30  Favourites: [{
31    type: mongoose.Schema.Types.ObjectId,
32    ref: 'Item'
33  }],
34  createdAt: {
35    type: Timestamp,
36    default: Date.now(),
37    expires: '1y'
38  }
39 })
40
41 const User = mongoose.model('User', userSchema)
42 module.exports = User
```

## 3. Relationships:

- The User schema has a one-to-many relationship with the Item schema through the Favourites field. Each user can have multiple favorite items.

## 4. Constraints:

- The email field in the User schema is unique, ensuring that each user account has a unique email address.
- The required attribute is set for essential fields such as name, email, password, gender, category, poster, rating, title, trailer Link, description, and watching Link, ensuring that these fields must be provided when creating new entries.