**Project Explanation**

**Overview**

The Virtual Classroom project is a web application designed to manage online classes, sessions, and student interactions. The project consists of a frontend application built with React and a backend server built with Express and MongoDB.

**Current Implementation**

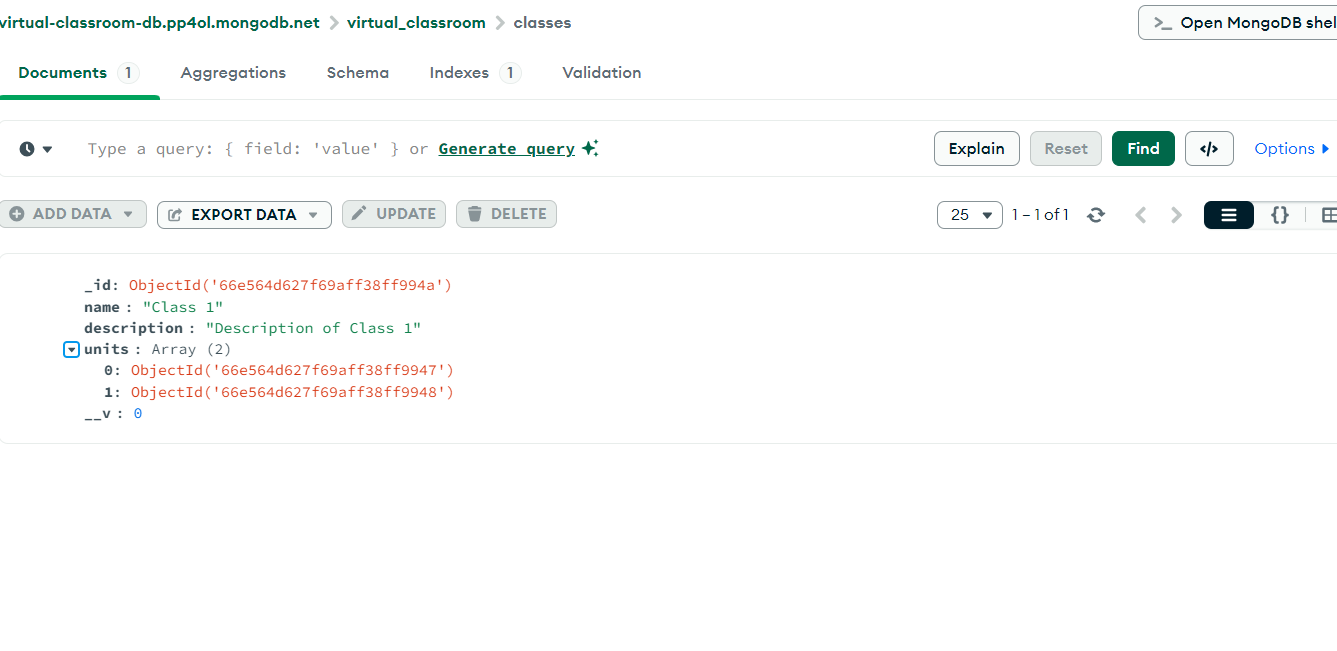
**1. Backend Setup**

**Database Connection**

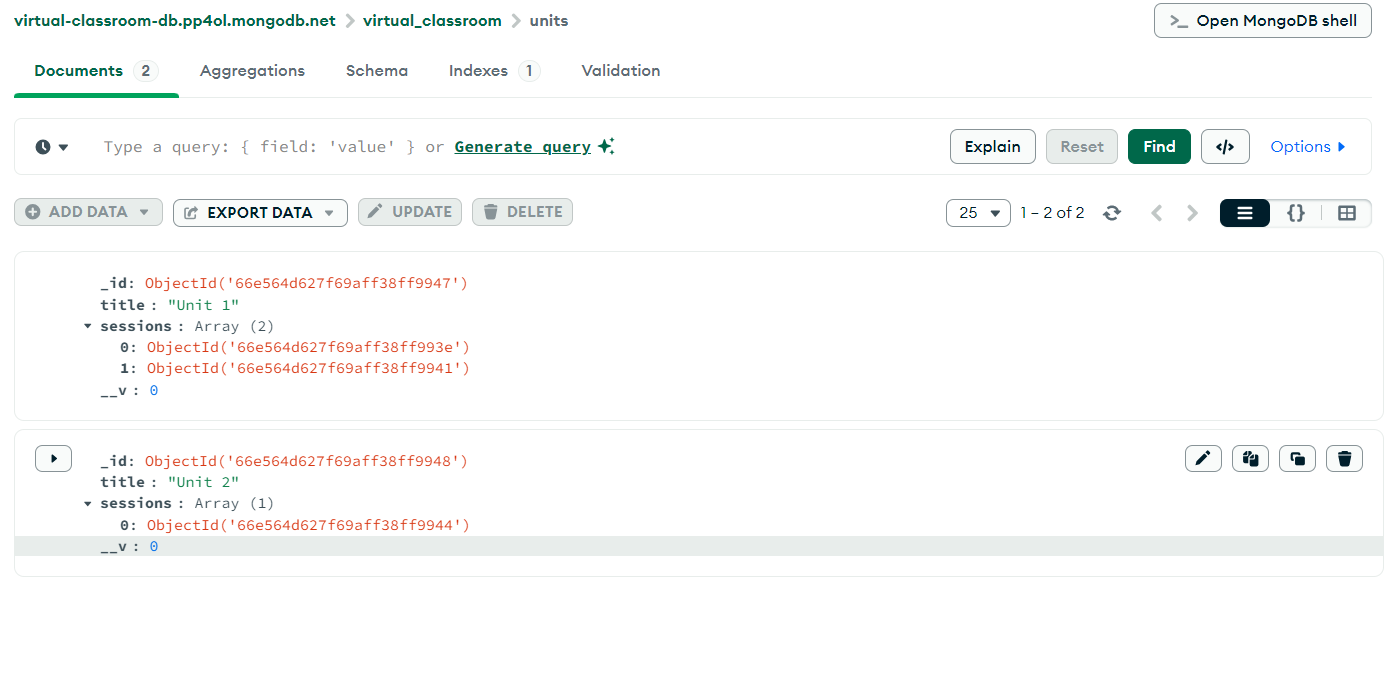
* **MongoDB Connection:** The backend server is configured to connect to MongoDB using Mongoose. The connection string is stored in the .env file under the MONGODB\_URI environment variable. This configuration allows the application to interact with a MongoDB database named virtual\_classroom.

**Models**

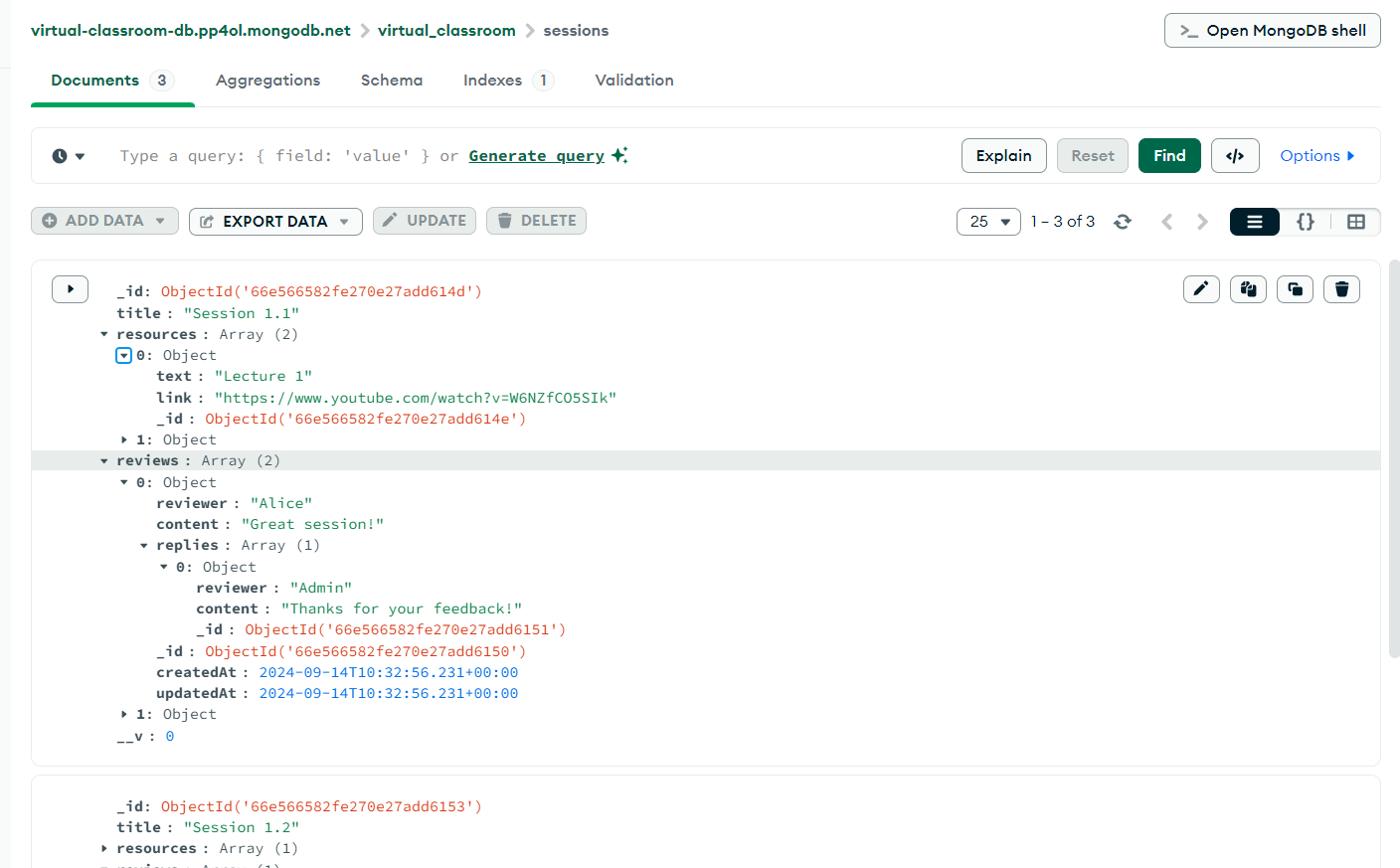
* **Class Model:** Defines the schema for a class in the virtual classroom, including attributes such as name, description, and other relevant details.



* **Unit Model:** Represents units within a class. Each unit can contain multiple sessions.

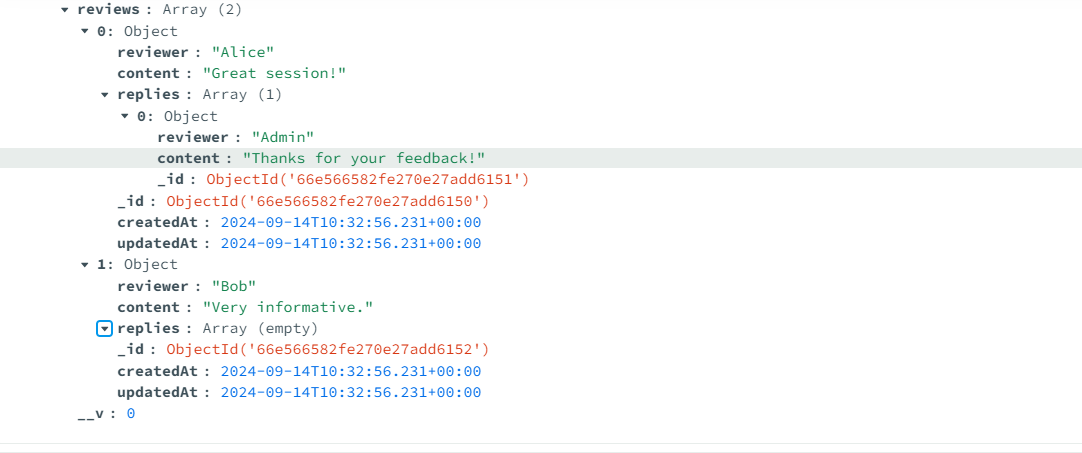


* **Session Model:** Defines sessions within units. Sessions include an array of resources, each with a text and link. This model also supports a review system with nested replies, allowing users to leave reviews and respond to them.



**Review System**

* **Nested Reviews:** The Session model has been updated to include a review system where each session can have multiple reviews. Reviews can also have nested replies, making it easy for users to interact and provide feedback on sessions.



**2. Frontend Setup**

**Folder Structure**

* **frontend/**: Contains the React application files. This includes components, pages, and other frontend logic.

**Frontend Features**

* **Basic Display:** The current implementation includes a basic page displaying the list of classes. The frontend is set up to interact with the backend server, but additional features such as displaying units, sessions, and reviews are yet to be fully integrated.

**3. Setup and Configuration**

**Backend**

* **app.js**: The main entry point for the backend application. It sets up the Express server and connects to the MongoDB database.
* **config/db.js**: Contains the database connection logic using Mongoose.

**Frontend**

* **React Application:** The frontend is configured with a basic structure. You should be able to see the "Classes" page when navigating to http://localhost:3000.

**4. Errors and Troubleshooting**

* **Module Not Found Errors:** These errors typically occur when required Node.js core modules are missing in the webpack configuration. This issue is related to the transition from webpack 4 to webpack 5, which no longer includes polyfills for Node.js core modules by default.

**Next Steps**

1. **Frontend Integration:** Integrate additional features into the frontend, such as displaying units, sessions, and handling user reviews.
2. **Polishing and Testing:** Complete the development of the frontend components, test the application thoroughly, and resolve any remaining issues.
3. **Deployment:** Prepare the application for deployment by configuring production settings and ensuring that both frontend and backend are ready for a live environment.