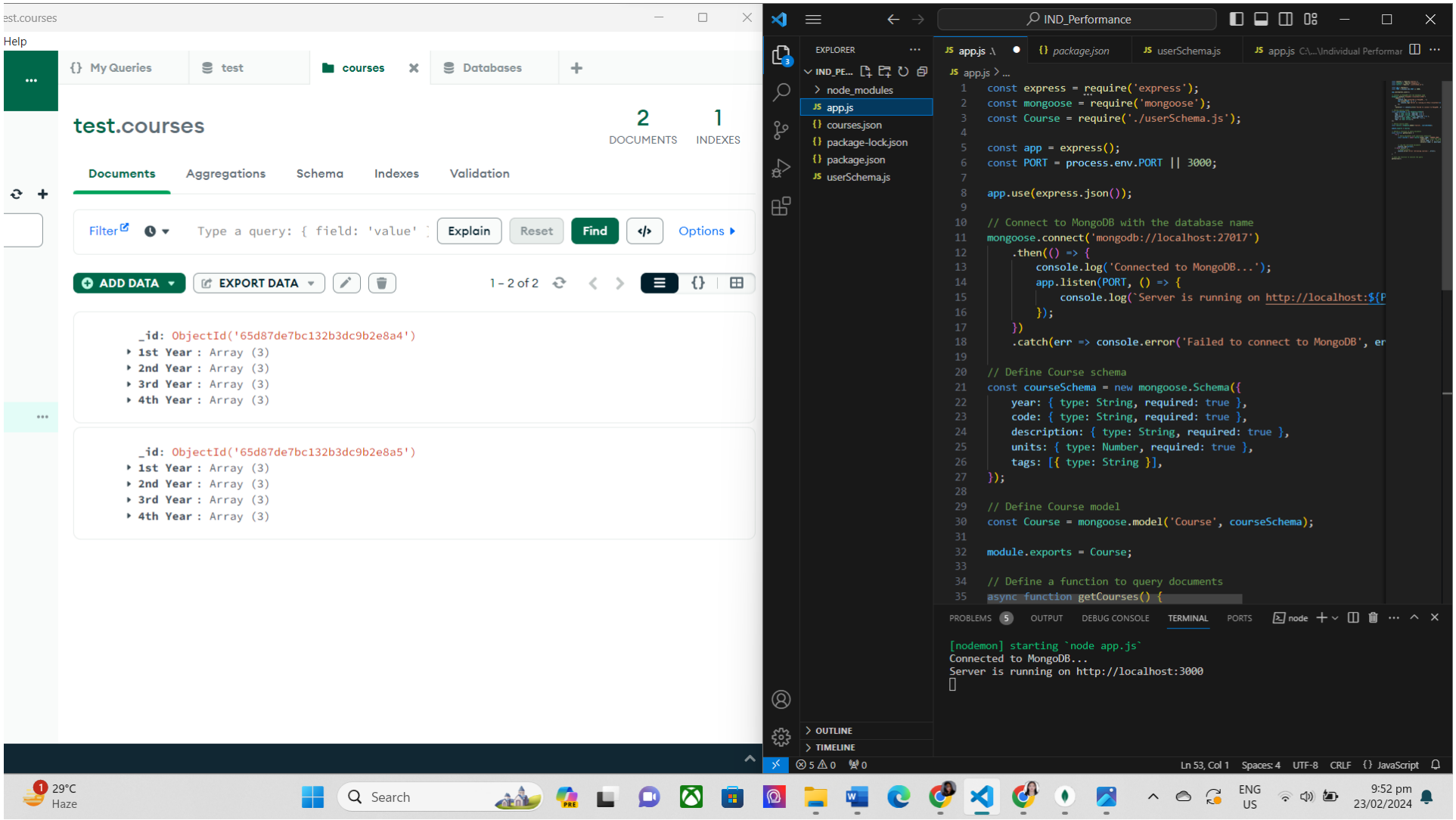
test.courses
Help
My Queries
test
courses
Databases
test.courses
2
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' }
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 2 of 2
_id: ObjectId('65d87de7bc132b3dc9b2e8a4')
1st Year : Array (3)
2nd Year : Array (3)
3rd Year : Array (3)
4th Year : Array (3)
_id: ObjectId('65d87de7bc132b3dc9b2e8a5')
1st Year : Array (3)
2nd Year : Array (3)
3rd Year : Array (3)
4th Year : Array (3)
IND_Performance
EXPLORER
IND_PE...
node_modules
app.js
courses.json
package-lock.json
package.json
userSchema.js
app.js
package.json
userSchema.js
app.js C:\...\Individual Performar
app.js > ...
const express = require('express');
const mongoose = require('mongoose');
const Course = require('./userSchema.js');
const app = express();
const PORT = process.env.PORT || 3000;
app.use(express.json());
// Connect to MongoDB with the database name
mongoose.connect('mongodb://localhost:27017')
.then(() => {
console.log('Connected to MongoDB...');
app.listen(PORT, () => {
console.log(`Server is running on http://localhost:${P
});
})
.catch(err => console.error('Failed to connect to MongoDB', er
// Define Course schema
const courseSchema = new mongoose.Schema({
year: { type: String, required: true },
code: { type: String, required: true },
description: { type: String, required: true },
units: { type: Number, required: true },
tags: [{ type: String }],
});
// Define Course model
const Course = mongoose.model('Course', courseSchema);
module.exports = Course;
// Define a function to query documents
async function getCourses() {
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
[nodemon] starting `node app.js`
Connected to MongoDB...
Server is running on http://localhost:3000
OUTLINE
TIMELINE
Ln 53, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
29°C
Haze
Search
ENG
US
9:52 pm
23/02/2024

1. **/courses/backend** - Retrieves all published backend courses, sorted alphabetically by name, and selects the name and specialization of each course.


```
[
    {
        "name": "Advanced Backend Development",
        "specialization": "Backend Development"
    },
    {
        "name": "Backend Architecture and Design",
        "specialization": "Backend Development"
    },
    {
        "name": "Node.js Fundamentals",
        "specialization": "Backend Development"
    }
]
```


2. **/courses/bsis** - Retrieves all published BSIS courses from the curriculum.


```
[
    {
        "year": "2022",
        "code": "BSIS101",
        "description": "Introduction to Information Systems",
        "units": 3,
        "tags": ["intro", "bsis"],
        "published": true
    },
    {
        "year": "2022",
        "code": "BSIS201",
        "description": "Database Management Systems",
        "units": 3,
        "tags": ["database", "bsis"],
        "published": true
    }
]
```


3. **/courses/bsit** - Retrieves all published BSIT courses from the curriculum.


```
[
    {
        "year": "2022",
        "code": "BSIT101",
        "description": "Introduction to Information Technology",
        "units": 3,
        "tags": ["intro", "bsit"],
        "published": true
    },
    {
        "year": "2022",
        "code": "BSIT202",
        "description": "Web Development Fundamentals",
        "units": 3,
        "tags": ["webdev", "bsit"],
        "published": true
    }
]
```

Challenges Faced and Solutions Implemented:

In setting up the backend API with MongoDB, I encountered several challenges due to my unfamiliarity with MongoDB. Connecting to the database posed initial difficulties, requiring extra effort to grasp the connection process. However, through research and experimentation, I managed to establish a successful connection.

One crucial aspect was data validation, ensuring that retrieved data matched the expected format and met specified criteria. Implementing data validation procedures helped maintain data integrity and accuracy throughout the application.

During testing, uncertainty arose regarding the correctness of the output. Verifying whether the endpoints returned the expected results was a significant concern. Nonetheless, thorough testing using various scenarios and tools helped validate the functionality of the implemented endpoints.

While retrieving all data was straightforward, sorting the courses proved challenging, especially concerning organizing them by their respective courses. Additionally, categorizing MongoDB output by year presented another hurdle, as I grappled with the complexities of structuring the code to achieve the desired output.

Throughout the process, I encountered parts of the code that were difficult to comprehend, leading to uncertainty about their placement and functionality. However, persistent effort and referring to documentation and resources aided in navigating and understanding these sections.

Overall, despite the challenges faced, I successfully managed to set up the backend API, retrieve data, and implement basic functionalities. Moving forward, I plan to deepen my understanding of MongoDB and improve my proficiency in handling similar tasks effectively.