Mark Joseph R. Largo

BSI/T – 3F

Performance Test 1

Objective:

The performance test aims to evaluate your ability to retrieve, sort, and select backend courses data according to specified requirements.

Steps:

- Download and install the MongoDB Database tool
- Create a backend API endpoint for the provided course data.

```
//Retrieve all published backend courses and sort them alphabetically by their names
app.get("/api/CoursesSorted", async (req, res) => {
    try {
      const years = await courseSchema.find();
      let courses = [];
      years.forEach((year) => {
        ["1st Year", "2nd Year", "3rd Year", "4th Year"].forEach((yearKey) => {
          if (year[yearKey]) {
            courses.push(...year[yearKey]);
          }
        });
      });
      courses.sort((a, b) => a.description.localeCompare(b.description));
      res.json(courses);
    } catch (err) {
      res.status(500).json({ message: err.message });
    }
});
```

- This is the endpoint I used for retrieving backend courses alphabetically.

```
[
  {
    "_id": "65d883c7806d1af371f587eb",
    "code": "BSIT103",
    "description": "Computer Organization and Architecture",
    "units": 3,
    "tags": [
      "BSIT103",
      "BSIT",
      "Computer",
      "Organization",
      "Architecture"
    ]
  },
  {
    "_id": "65d883c7806d1af371f587e2",
    "code": "BSIS203",
    "description": "Data Structures and Algorithms",
    "units": 3,
    "tags": [
      "BSIS203",
      "BSIS",
      "Data",
      "Structures",
      "Algorithms"
    ]
  },
  {
    "_id": "65d883c7806d1af371f587ec",
    "code": "BSIT201",
    "description": "Database Management",
    "units": 3,
    "tags": [
      "BSIT201",
      "BSIT",
      "Database",
      "Management"
    ]
```


- This is the output of all the published backend courses sorted alphabetically.

```
//Select and extract the name and specialization of each course
app.get("/api/CoursesNameAndSpecialization", async (req, res) => {
    try {
      const years = await courseSchema.find();
      let courses = [];
      years.forEach((year) => {
        ["1st Year", "2nd Year", "3rd Year", "4th Year"].forEach((yearKey) => {
          if (year[yearKey]) {
            courses.push(...year[yearKey]);
          }
        });
      });
      const descriptionsAndTags = courses.map((course) => ({
        description: course.description,
        tags: course.tags,
      }));
      res.json(descriptionsAndTags);
    } catch (err) {
      res.status(500).json({ message: err.message });
    }
  });
```

- Here is the endpoint I used to Select and extract the name and specialization of each course.

```
[
  {
    "description": "Introduction to Information Systems",
    "tags": [
      "BSIS101",
      "BSIS",
      "Information Systems",
      "Introduction"
    ]
  },
  {
    "description": "Fundamentals of Programming",
    "tags": [
      "BSIS102",
      "BSIS",
      "Programming",
      "Fundamentals"
    ]
  },
  {
    "description": "Database Management Systems",
    "tags": [
      "BSIS103",
      "BSIS",
      "Database",
      "Management",
      "Systems"
    ]
  },
  {
    "description": "Systems Analysis and Design",
    "tags": [
      "BSIS201",
      "BSIS",
      "Systems",
      "Analysis",
      "Design"
    ]
  }
```


- This is the extracted name and specialization of each course.

```
//Retrieve all published BSIS and BSIT courses from the curriculum.

app.get("/api/CoursesPublished", async (req, res) => {
    try {
      const years = await courseSchema.find();
      let courses = [];
      years.forEach((year) => {
        ["1st Year", "2nd Year", "3rd Year", "4th Year"].forEach((yearKey) => {
          if (year[yearKey]) {
            courses.push(...year[yearKey]);
          }
        });
      });
      const descriptionsAndTags = courses
        .filter(
          (course) => course.tags.includes("BSIT") || course.tags.includes("BSIS")
        )
        .map((course) => ({
          description: course.description,
          tags: course.tags,
        }));
      res.json(descriptionsAndTags);
    } catch (err) {
      res.status(500).json({ message: err.message });
    }
  });

```

- This is the endpoint to retrieve all the published BSIS and BSIT courses.

```
[
  {
    "description": "Introduction to Information Systems",
    "tags": [
      "BSIS101",
      "BSIS",
      "Information Systems",
      "Introduction"
    ]
  },
  {
    "description": "Fundamentals of Programming",
    "tags": [
      "BSIS102",
      "BSIS",
      "Programming",
      "Fundamentals"
    ]
  },
  {
    "description": "Database Management Systems",
    "tags": [
      "BSIS103",
      "BSIS",
      "Database",
      "Management",
      "Systems"
    ]
  },
  {
    "description": "Systems Analysis and Design",
    "tags": [
      "BSIS201",
      "BSIS",
      "Systems",
      "Analysis",
      "Design"
    ]
```


- This is the output of the retrieved BSIS and BSIT courses.

While doing this activity, I encountered difficulties understanding MongoDB concepts and efficiently utilizing the database due to my limited knowledge. To overcome this, I dedicated time to self-learning through online resources. Additionally, comprehending MongoDB's sorting methods and implementing sorting logic in Express.js API endpoints posed challenges in creating functionality for alphabetically sorting data. I must say that I still have a lot of things to learn and improve, overcoming these obstacles not only improved my problem-solving skills but also expanded my knowledge about backend programming.