Vanessa R. Octa **February 23, 2024**

BSI/T 3D

Introduction

This documentation provides an overview of the functionality and endpoints of the Express application with MongoDB integration for managing courses.

Getting Started

To run the application, ensure you have Node.js and MongoDB installed on your system. Then follow these steps:

1. Clone the repository.

2. Install dependencies using `npm install`.

3. Start the server with `npm start`.

Endpoints

1. Retrieve all BSIS courses

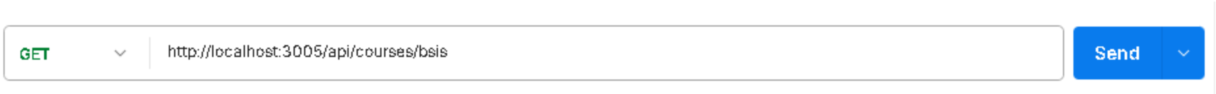
URL: `/api/courses/bsis`

Method:`GET`

Description: Retrieves all Bachelor of Science in Information Systems (BSIS) courses.

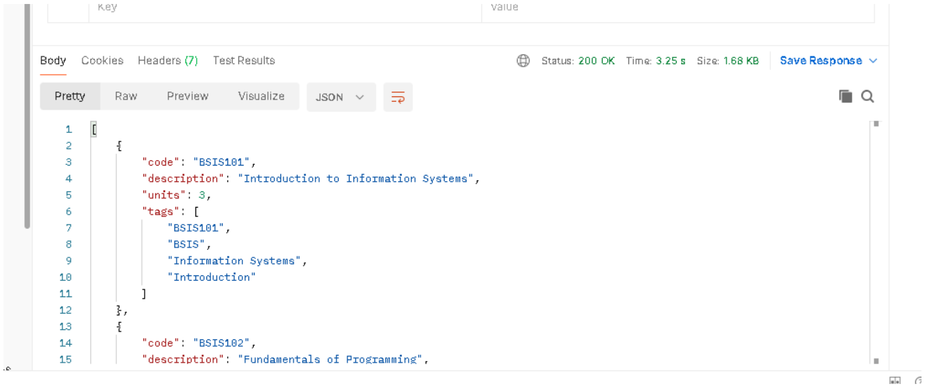
Response: An array of BSIS courses in JSON format.

Request:



http://localhost:3005/api/courses/bsis

Response:



Status: 200 OK Time: 3.25 s Size: 1.68 KB

```
[
    {
        "code": "BSIS101",
        "description": "Introduction to Information Systems",
        "units": 3,
        "tags": [
            "BSIS101",
            "BSIS",
            "Information Systems",
            "Introduction"
        ]
    },
    {
        "code": "BSIS102",
        "description": "Fundamentals of Programming",
```

2. Retrieve all BSIT courses

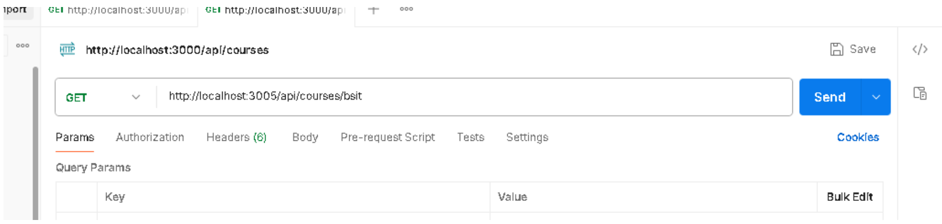
URL: `/api/courses/bsit`

Method: `GET`

Description: Retrieves all Bachelor of Science in Information Technology (BSIT) courses.

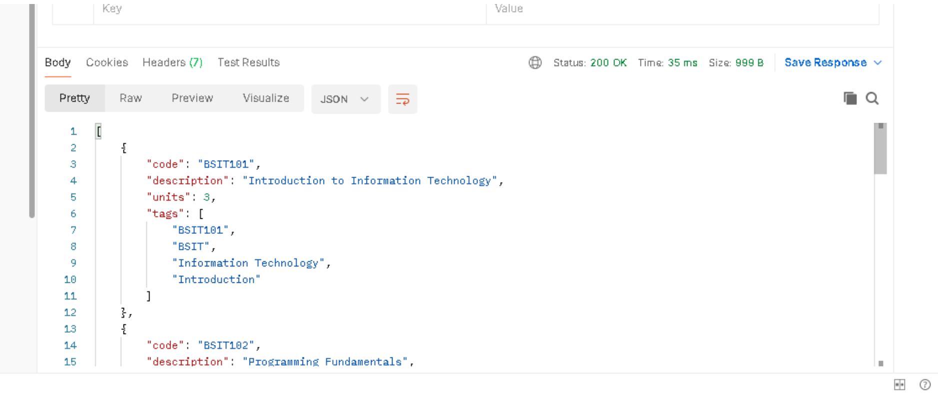
Response: An array of BSIT courses in JSON format.

Request:



GET http://localhost:3005/api/courses/bsit

Response:



Status: 200 OK Time: 35 ms Size: 999 B

```
[
    {
        "code": "BSIT101",
        "description": "Introduction to Information Technology",
        "units": 3,
        "tags": [
            "BSIT101",
            "BSIT",
            "Information Technology",
            "Introduction"
        ]
    },
    {
        "code": "BSIT102",
        "description": "Programming Fundamentals",
```

3. Retrieve all backend courses alphabetically

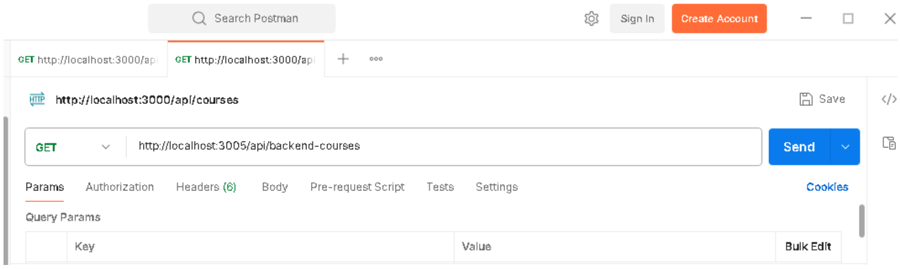
URL: `/api/backend-courses`

Method: `GET`

Description: Retrieves all backend courses sorted alphabetically by description.

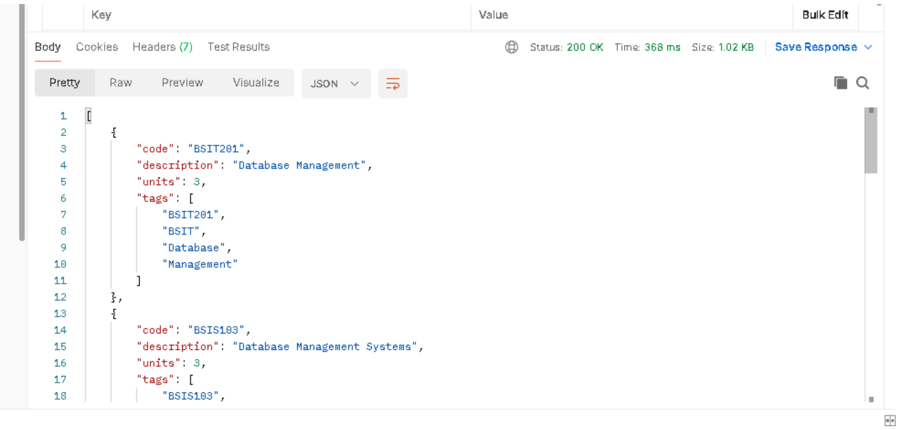
Response: An array of backend courses in JSON format.

Request:



GET http://localhost:3005/api/backend-courses

Response:



Status: 200 OK Time: 368 ms Size: 1.02 KB

```
[
    {
        "code": "BSIT201",
        "description": "Database Management",
        "units": 3,
        "tags": [
            "BSIT201",
            "BSIT",
            "Database",
            "Management"
        ]
    },
    {
        "code": "BSIS103",
        "description": "Database Management Systems",
        "units": 3,
        "tags": [
            "BSIS103",
```

4. Retrieve course details

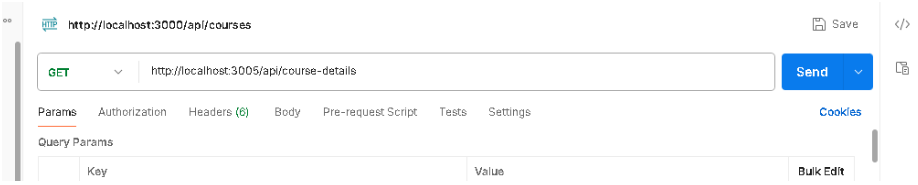
URL: `/api/course-details`

Method: `GET`

Description: Retrieves details of all courses including their name and specialization.

Response: An array of course details in JSON format.

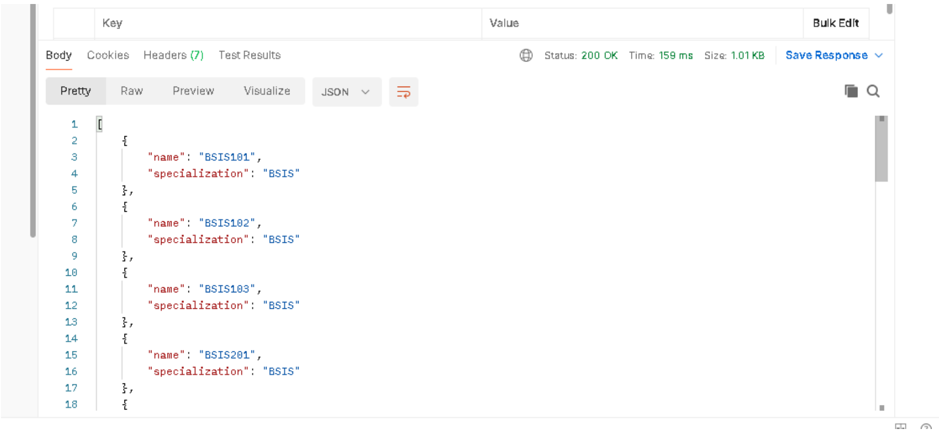
Request:



http://localhost:3000/api/courses

GET http://localhost:3005/api/course-details

Response:



Status: 200 OK Time: 159 ms Size: 1.01 KB

```
[
    {
        "name": "BSIS101",
        "specialization": "BSIS"
    },
    {
        "name": "BSIS102",
        "specialization": "BSIS"
    },
    {
        "name": "BSIS103",
        "specialization": "BSIS"
    },
    {
        "name": "BSIS201",
        "specialization": "BSIS"
    },
    {
```

5. Home

URL: `/`

Method: `GET`

Description: A simple route to test server connectivity.

Response: Returns 'Hello World!'

Data Models

Course

- **Fields:**

  - `description` (String, required): Description of the course.

  - `tags` (Array of Strings): Tags associated with the course.

Dependencies

Express: Fast, unopinionated, minimalist web framework for Node.js.

Mongoose: MongoDB object modeling tool designed to work in an asynchronous environment.

Database

The application connects to a local MongoDB instance named `mongo-test`.

Error Handling

In case of errors, the server returns a 500 Internal Server Error response along with an error message in JSON format.

Running the Server

The server listens on port 3005 by default. You can specify a different port using the `PORT` environment variable.

This documentation covers the endpoints, data models, dependencies, database connection, error handling, and how to run the server.