Adding the search capabilities on products and middleware for validation
Assignment Solution

Relevel
by Unacademy

1. In our mini project built till session3 let's add custom filters as mentioned below
- Get books with given title
- Get books after given published date

Step 1: Modify Books Controller to accept queries in /books api:

```
const Books = require("../models/BookModel");
const sequelize = require("sequelize");
const Op = sequelize.Op;

/**
 * Get a list of all the Books
 */
const getAllBooks = async (req, res) => {

    const bookName = req.query.name;
    const publishedAfter = req.query.publishedAfter;
    let promise;

    if ( bookName && publishedAfter ) {
        const publishedAfterDate = new Date(publishedAfter);
        console.log(publishedAfterDate)
        promise = Books.findAll({
            where: {
                title: {
                    [Op.substring]: bookName
                },
                published: {
                    [Op.gt]: publishedAfter
                }
            }
        });
    }
    else if ( publishedAfter ) {
        const publishedAfterDate = new Date(publishedAfter);
        console.log(publishedAfterDate)
        promise = Books.findAll({
            where: {
                published: {
                    [Op.gt]: publishedAfterDate
                }
            }
        });
    }
else if ( bookName ) {
        promise = Books.findAll({
            where: {
                title: {
                    [Op.substring]: bookName
                }
            }
        });
    } else {
        promise = Books.findAll();
    }

    promise
        .then((books) => {
            res.send(books);
        })
        .catch(() => {
            console.error(error.message);
            res.status(500).json({message: error.message})
        })
}

module.exports = { getAllBooks, getBookById, addBook, updateBookById,
deleteBookById }
```

2. Also create a middleware to validate request instead and check for empty title here

Step 1: Create requestValidator.js file in middlewares folder

```
const Books = require("../models/BookModel");

const validateBookRequest = (req, res, next) => {
   /**
    * Validation of the request body
    */
   if (!req.body.title) {
       res.status(400).send({
           message: "Title of the book can't be empty !"
       })
       return;
   }
   next();
}
module.exports = { validateBookRequest }
```

Step 2: Update the routes accordingly

```
const express = require('express');
const router = express.Router();
const bookController = require("../controllers/BookController");
const { validateBookRequest } = require("../middlewares/requestValidator");

//Route for the POST request to add a book
router.post("/", validateBookRequest, bookController.addBook);

//Route for the GET request to fetch all the books details
router.get("/", bookController.getAllBooks);

//Route for the GET request to fetch a book based on the id
router.get("/:id", bookController.getBookById);

//Route for the PUT request to update a book based on the id
router.put("/:id", validateBookRequest, bookController.updateBookById);

//Route for the DELETE request to delete a book based on the id
router.delete("/:id", bookController.deleteBookById);

module.exports = router;
```

For reference checkout the github repo link here.