BT Assign 4

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract StudentData {

    // Structure to represent a student

    struct Student {

        uint256 studentId;

        string name;

        uint256 age;

        string Branch; // Changed "subject" to "Branch"

    }

    // Array to store student records

    Student[] public students;

    // Fallback function - Accept Ether (for educational purposes only)

    receive() external payable {

        // Simple fallback function that accepts Ether and emits an event

        emit Received(msg.sender, msg.value);

    }

    event Received(address indexed sender, uint256 value);

    // Function to add a new student

    function addStudent(

        uint256 \_studentId,

        string memory \_name,

        uint256 \_age,

        string memory \_Branch

    ) public {

        Student memory newStudent = Student({

            studentId: \_studentId,

            name: \_name,

            age: \_age,

            Branch: \_Branch

        });

        students.push(newStudent);

    }

    // Function to get the total number of students

    function getStudentCount() public view returns (uint256) {

        return students.length;

    }

    // Function to get student details by index

    function getStudentDetails(uint256 index) public view returns (

        uint256 studentId,

        string memory name,

        uint256 age,

        string memory Branch

    ) {

        require(index < students.length, "Invalid index");

        Student storage student = students[index];

        return (student.studentId, student.name, student.age, student.Branch);

    }

}