ASSIGNMENT – 14

INPUT –

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.27;

contract StudentData {

    // Structure to store student information

    struct Student {

        uint id;

        string name;

        uint age;

    }

    // Array to store the list of students

    Student[] public students;

    // Event to log the addition of a new student

    event StudentAdded(uint id, string name, uint age);

    // Constructor: Now payable to accept Ether

    constructor() payable {

        // Constructor can accept Ether

    }

    // Function to add a student to the array

    function addStudent(uint \_id, string memory \_name, uint \_age) public {

        students.push(Student(\_id, \_name, \_age));

        emit StudentAdded(\_id, \_name, \_age); // Emit event when student is added

    }

    // Function to retrieve the number of students

    function getStudentCount() public view returns (uint) {

        return students.length;

    }

    // Receive function to accept plain Ether transactions

    receive() external payable {

        // Receive function handles Ether sent directly

    }

    // Fallback function to handle unexpected calls or Ether with data

    fallback() external payable {

        // Fallback function to receive Ether when calldata is present

    }

    // Function to retrieve contract balance if fallback or receive function receives Ether

    function getBalance() public view returns (uint) {

        return address(this).balance;

    }

}

OUTPUT –







