// SPDX-License-Identifier: MIT

pragma solidity ^0.8.18;

contract StudentData {

struct Student {

uint id;

string name;

uint age;

}

Student[] public students;

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

// Function to add a new student

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

Student memory newStudent = Student({

id: \_id,

name: \_name,

age: \_age

});

students.push(newStudent);

emit StudentAdded(\_id, \_name, \_age);

}

// Function to get student details by index

function getStudent(uint index) public view returns (uint, string memory, uint) {

require(index < students.length, "Student not found.");

Student memory student = students[index];

return (student.id, student.name, student.age);

}

// Fallback function to handle unexpected calls

fallback() external {

// This function can be used to log or handle unexpected calls

}

}