13/10/2024, 23:08 Remix - Ethereum IDE

```
S Student.sol 1 X │ ○ scenario1.json
                             Home
          // SPDX-License-Identifier: MIT
               pragma solidity ^0.8.0;
               contract StudentData {
Q
                   // Structure to store student details
                   struct Student {
                       uint id;
                       string name;
                       uint age;
                       string course;
         10
         11
濂
         12
         13
                   // Array to store multiple students
         14
                   Student[] public students;
         15
                   // Mapping to track if a student with a specific ID exists
         16
         17
                   mapping(uint => bool) public studentExists;
         18
         19
                   \ensuremath{//} Event to notify when a student is added
         20
                   event StudentAdded(uint id, string name, uint age, string course);
         21
         22
                   // Function to add a student
                   function addStudent(uint _id, string memory _name, uint _age, string memory _course) public {
         23
                       require(!studentExists[_id], "Student with this ID already exists!");
         24
         25
         26
                       // Add new student to the array
                       students.push(Student(_id, _name, _age, _course));
         27
         28
                       studentExists[_id] = true;
         29
         30
                       // Emit event for added student
         31
                       emit StudentAdded(_id, _name, _age, _course);
          32
          33
                   // Fallback function to handle ether sent directly to contract
         34
         35
                   \verb|fallback()| external payable { } \verb| <math>\blacksquare  undefined gas
         36
                       \ensuremath{//} This fallback function allows the contract to receive Ether.
         37
          38
                   // Function to get student details by index
          39
         40
                   function getStudent(uint index) public view returns (uint, string memory, uint, string memory) {
                       require(index < students.length, "Index out of bounds");</pre>
         41
         42
                       Student memory s = students[index];
         43
                       return (s.id, s.name, s.age, s.course);
          45
                   // Function to get the total number of students
         46
                   function getStudentCount() public view returns (uint) { ■ 2440 gas
         47
         48
                       return students.length;
         49
          50
               }
         51
```

