```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract StudentData {
    struct Student {
        uint256 id;
        string name;
        uint256 age;
        mapping(string => uint256) marks; // Mapping to hold marks for each
subject
    }

    Student[] public students;

    fallback() external payable {
        revert("Fallback: Function does not exist");
    }

    receive() external payable {}

    // Function to add a student
    function addStudent(string memory _name, uint256 _age) public {
        uint256 studentId = students.length; // Assign ID based on current array
length
        students.push(); // Create space for the new student
        Student storage newStudent = students[studentId]; // Reference to the
new student
        newStudent.id = studentId;
        newStudent.name = _name;
        newStudent.age = _age;
    }

    // Function to set marks for a student
    function setMarks(uint256 _studentId, string memory _subject, uint256
_marks) public {
        require(_studentId < students.length, "Student ID does not exist."); //
Ensure ID is valid
        students[_studentId].marks[_subject] = _marks; // Set marks for the
specified subject
    }

    // Function to get marks for a specific subject
    function getMarks(uint256 _studentId, string memory _subject) public view
returns (uint256) {
        require(_studentId < students.length, "Student ID does not exist."); //
Ensure ID is valid
        return students[_studentId].marks[_subject]; // Return marks for the
specified subject
    }

    // Function to retrieve student details by ID
    function getStudent(uint256 _id) public view returns (uint256, string
memory, uint256) {
        require(_id < students.length, "Student ID does not exist."); // Ensure
ID is valid
        Student storage student = students[_id]; // Reference to the student in
storage
        return (student.id, student.name, student.age); // Return student
details
    }

    // Function to get the total number of students
    function getTotalStudents() public view returns (uint256) {
        return students.length;
    }
}
```