```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract StudentData {
    struct Student {
        uint256 id;
        string name;
        uint8 age;
        string course;
    }

    Student[] private students;

    event StudentAdded(uint256 id, string name, string course);
    event FallbackCalled(address sender, uint value, string message);

    function addStudent(uint256 _id, string memory _name, uint8 _age, string memory _course)
public {
        students.push(Student(_id, _name, _age, _course));
        emit StudentAdded(_id, _name, _course);
    }

    function getStudent(uint256 index) public view returns (uint256, string memory, uint8, string
memory) {
        require(index < students.length, "Invalid index");
        Student memory s = students[index];
        return (s.id, s.name, s.age, s.course);
    }

    function getTotalStudents() public view returns (uint256) {
        return students.length;
    }
```

```
    fallback() external payable {
        emit FallbackCalled(msg.sender, msg.value, "Fallback function triggered!");
    }

    receive() external payable {
        emit FallbackCalled(msg.sender, msg.value, "Receive function triggered!");
    }
}
```

Step	Action	Details
1️⃣ **Open Remix IDE**	Visit https://remix.ethereum.org	Online Solidity IDE
2️⃣ **Create New File**	File Explorer → "New File" → name it StudentData.sol	Paste the code above
3️⃣ **Compile the Contract**	Go to **Solidity Compiler** tab → Compiler version **0.8.20** → EVM version **Cancun** or **London** → click **Compile StudentData.sol**	Green ✔ = successful compilation
4️⃣ **Deploy the Contract**	Go to **Deploy & Run Transactions** tab	
	Environment → **Remix VM (Cancun)**	Local blockchain simulation
	Account → choose first account	Default account with 100 ETH (fake)
	Gas Limit → 6000000	Safe gas limit
	Value → 0	No ETH needed for deployment
	Click **Deploy**	Contract is created successfully

Step	Action	Details
5️⃣ **Verify Deployment**	Contract appears under **Deployed Contracts**	You'll see STUDENTDATA AT 0x...

(a) Add Student

Field	Example Input	Explanation
_id	1	Student ID
_name	"Alice"	Student name (with quotes)
_age	20	Integer only
_course	"Blockchain"	Student course (with quotes)
Action	Click **addStudent**	Adds student data to array
Output	Terminal shows StudentAdded event log	

(b) Get Student

Field	Example Input	Explanation
index	0	First student in array
Action	Click **getStudent**	Reads struct details
Output	Returns (1, "Alice", 20, "Blockchain")	