CODE:

```
// SPDX-License-Identifier: MIT pragma solidity

>=0.7.0; contract StudentManagement { struct

Student { uint256 stud_id; // Changed to uint256
for

positive IDs string Name; string Department,

}

Student[] private Students; // Made it private for
encapsulation

// Payable constructor to allow Ether transfer constructor()

payable {

// Optional: Initialize state if needed

}

function addStudent(uint256 stud_id, string memory Name, string memory
Department)  public {

Student memory stud = Student(stud_id, Name,
Department).

Students.push(stud);

} function getStudent(uint256 stud_id) public view returns (string memory,
string

memory) { for (uint256 i = 0; i < Students.length; i++) {  if
(Students[i].stud_id

stud_id) {
```

```
return (Students[i] Name,
Students[i].Department);

}

return ("Name Not Found", "Department Not Found");

}

// Receive function to handle plain Ether transfers receive()

external payable {

// Optional: Handle received Ether, e.g., logging, storing info,
etc.

}

// Fallback function fallback()

external payable {

// This can be used for specific handling if necessary

// Example: Adding a default
student


}
Students.push(Student(7, "XYZ",
"Mechanical"));


}
```

OUTPUT:

Deployed/Unpinned Contracta