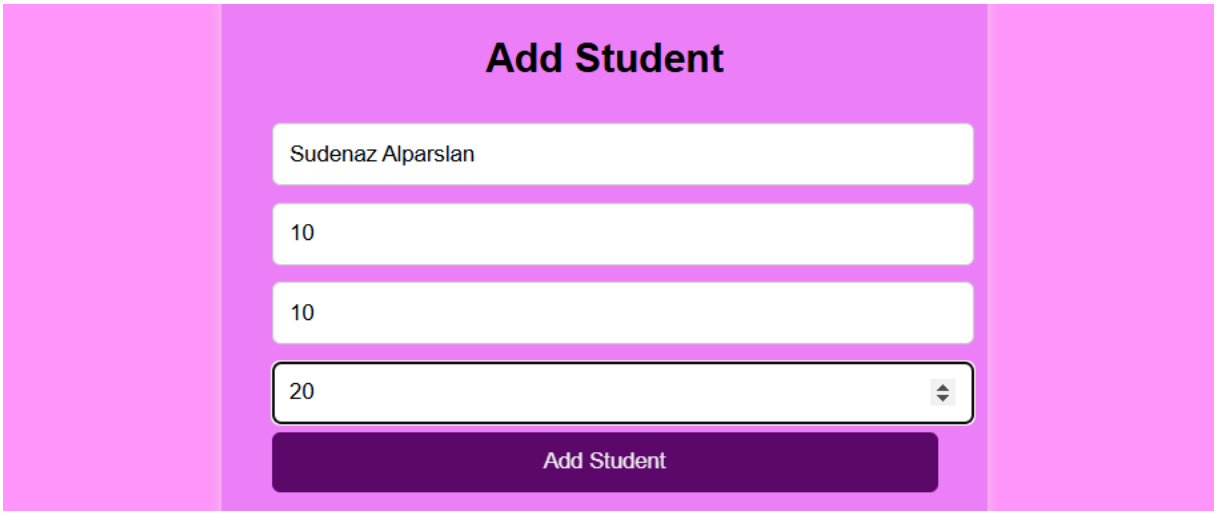
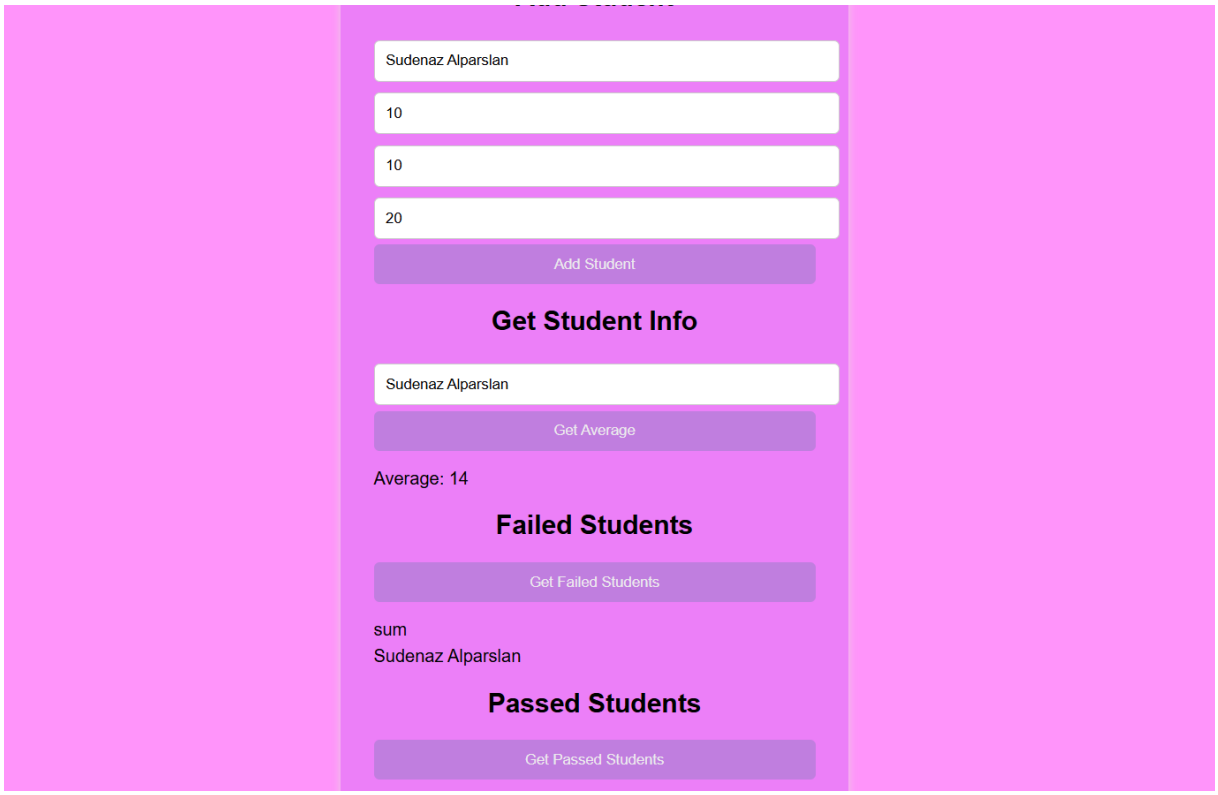
SÜMEYYE ALPARSLAN – 220504043

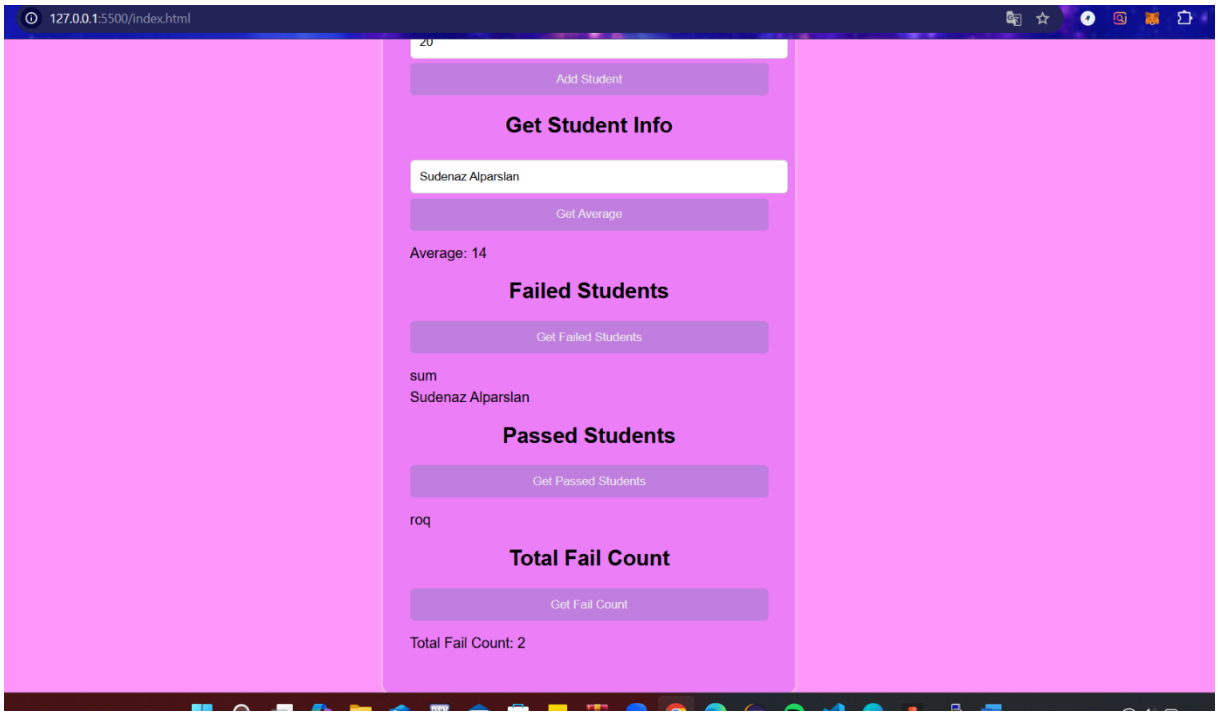
GRADE CALCULATOR:

The GradeCalculator contract allows for storing and calculating student grades. It defines a Student struct with fields for the student's name, midterm grades, final grade, average score. The addStudent function calculates the average score based on weighted midterm and final exam grades, and assigns a letter grade. If the average is below 50, the student is counted as failed, and the failCount is incremented.

Additionally, the contract provides functions to retrieve information about student performance. The getFailCount function returns the number of students who failed, while getFailedStudents and getPassedStudents return lists of names for students who failed or passed, respectively, based on their average scores.







I've connected to metamask and added some other students too. And it shows the ones who failed and the ones who passed & the Fail Counter which is how many people failed. Roq, Sum & and Sudenaz Alparslan are different students.

SOLIDITTY CODE:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.26;

contract GradeCalculator {
    struct Student {
        string name;
        uint mid1;
        uint mid2;
        uint finalGrade;
        uint average;
        string grade;
    }

    Student[] private students;
    uint private failCount = 0;

    function addStudent(string memory _name, uint _mid1, uint _mid2, uint _final) public {
        require(_mid1 >= 0 && _mid1 <= 100, "Midterm 1 grade must be between 0 and 100.");
        require(_mid2 >= 0 && _mid2 <= 100, "Midterm 2 grade must be between 0 and 100.");
        require(_final >= 0 && _final <= 100, "Final grade must be between 0 and 100.");

        uint average = (_mid1 * 30 / 100) + (_mid2 * 30 / 100) + (_final * 40 / 100);
        string memory grade = calculateGrade(average);

        if (average < 50) {
            failCount++;
        }

        students.push(Student(_name, _mid1, _mid2, _final, average, grade));
    }

    function getFailCount() public view returns (uint) {
        return failCount;
    }
```

```
    function getFailedStudents() public view returns (string[] memory) {
        string[] memory failedStudentNames = new string[](failCount);
        uint index = 0;

        for (uint i = 0; i < students.length; i++) {
            if (students[i].average < 50) {
                failedStudentNames[index] = students[i].name;
                index++;
            }
        }

        return failedStudentNames;
    }

    function getPassedStudents() public view returns (string[] memory) {
        uint passedCount = students.length - failCount;
        string[] memory passedStudentNames = new string[](passedCount);
        uint index = 0;

        for (uint i = 0; i < students.length; i++) {
            if (students[i].average >= 50) {
                passedStudentNames[index] = students[i].name;
                index++;
            }
        }

        return passedStudentNames;
    }


}
```

JAVASCRIPT CODE:

```
let web3;

let contract;

const contractAddress = "0xE2B9F63A18f38b2fB0CDC284bf0a9140A04653ea"; // Buraya kontrat
adresini yazın

const contractABI = [

    {

        "inputs": [

            {

                "internalType": "string",

                "name": "_name",

                "type": "string"

            },

            {

                "internalType": "uint256",

                "name": "_mid1",

                "type": "uint256"

            },

            {

                "internalType": "uint256",

                "name": "_mid2",

                "type": "uint256"

            },

            {

                "internalType": "uint256",

                "name": "_final",

                "type": "uint256"

            }

        ],

        "name": "addStudent",

        "outputs": [],

        "stateMutability": "nonpayable",

        "type": "function"

    },

    {

        "inputs": [],

        "name": "getFailCount",

        "outputs": [
```

```
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [],
    "name": "getFailedStudents",
    "outputs": [
        {
            "internalType": "string[]",
            "name": "",
            "type": "string[]"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [],
    "name": "getPassedStudents",
    "outputs": [
        {
            "internalType": "string[]",
            "name": "",
            "type": "string[]"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "string",
```

```
                "name": "_name",
                "type": "string"
            }
        ],
        "name": "getStudentInfo",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
];

window.onload = async () => {
    if (typeof window.ethereum !== "undefined") {
        web3 = new Web3(window.ethereum);
        await window.ethereum.enable(); // MetaMask'ı etkinleştir
        contract = new web3.eth.Contract(contractABI, contractAddress);
    } else {
        alert("MetaMask is not installed!");
    }
};

// Add Student
async function addStudent() {
    const name = document.getElementById("studentName").value;
    const mid1 = parseInt(document.getElementById("mid1").value);
    const mid2 = parseInt(document.getElementById("mid2").value);
    const final = parseInt(document.getElementById("finalGrade").value);
    const accounts = await web3.eth.getAccounts();

    await contract.methods.addStudent(name, mid1, mid2, final).send({ from: accounts[0] });
    alert("Student added successfully!");
}
```

```
// Get Student Info (Average)
async function getStudentInfo() {
    const studentName = document.getElementById("getStudentName").value;
    const average = await contract.methods.getStudentInfo(studentName).call();
    document.getElementById("studentInfoResult").innerText = `Average: ${average}`;
}

// Get Failed Students
async function getFailedStudents() {
    const failedStudents = await contract.methods.getFailedStudents().call();
    const failedList = document.getElementById("failedStudentsList");
    failedList.innerHTML = "";
    failedStudents.forEach(student => {
        const li = document.createElement("li");
        li.textContent = student;
        failedList.appendChild(li);
    });
}

// Get Passed Students
async function getPassedStudents() {
    const passedStudents = await contract.methods.getPassedStudents().call();
    const passedList = document.getElementById("passedStudentsList");
    passedList.innerHTML = "";
    passedStudents.forEach(student => {
        const li = document.createElement("li");
        li.textContent = student;
        passedList.appendChild(li);
    });
}

// Get Fail Count
async function getFailCount() {
    const failCount = await contract.methods.getFailCount().call();
    document.getElementById("failCount").innerText = `Total Fail Count: ${failCount}`;
}
```

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Grade Calculator</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Grade Calculator</h1>
        <div>
            <h2>Add Student</h2>
            <input type="text" id="studentName" placeholder="Student Name">
            <input type="number" id="mid1" placeholder="Midterm 1 (0-100)">
            <input type="number" id="mid2" placeholder="Midterm 2 (0-100)">
            <input type="number" id="finalGrade" placeholder="Final Grade (0-100)">
            <button onclick="addStudent()">Add Student</button>
        </div>

        <div>
            <h2>Get Student Info</h2>
            <input type="text" id="getStudentName" placeholder="Enter Student Name">
            <button onclick="getStudentInfo()">Get Average</button>
            <p id="studentInfoResult"></p>
        </div>

        <div>
            <h2>Failed Students</h2>
            <button onclick="getFailedStudents()">Get Failed Students</button>
            <ul id="failedStudentsList"></ul>
        </div>

        <div>
            <h2>Passed Students</h2>
            <button onclick="getPassedStudents()">Get Passed Students</button>
            <ul id="passedStudentsList"></ul>
```

```
        </div>

        <div>
            <h2>Total Fail Count</h2>
            <button onclick="getFailCount()">Get Fail Count</button>
            <p id="failCount"></p>
        </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/web3@1.6.1/dist/web3.min.js"></script>
    <script src="app.js"></script>
</body>
</html>
```

CSS CODE:

```
body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background-color: hsl(303, 100%, 79%);
}

.container {
    background-color: #ec7ff8;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(232, 225, 225, 0.799);
    width: 400px;
}

h1, h2 {
    text-align: center;
}

input {
```

```
    width: 100%;
    padding: 10px;
    margin: 5px 0;
    border-radius: 5px;
    border: 1px solid #ccc;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #c07edf;
    color: rgb(237, 234, 238);
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #5b086a;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    margin: 5px 0;
}
```