# ESSM (Electronic system for student management)

This project is about creating an electronic system for setting grades from professors to students. It’s called ESSM and stands for Electronic System for Student Management. The project has used the local environment i.e., Ganache. The first part of the work was about defining and developing finalization of the smart contract. Smart contract is coded in Remix – Etherum IDE. The idea stands for some professors to teach some relevant subjects, and to be able to assign grades to the students who follow that subject.

This document shows the project description and anyone who wants to try to run the code, should follow these steps:

1. Open Ganache, create a new workspace and don’t forget to increase the gass limit.
2. Copy the first four addresses and then paste in the smart contract constructor to the part where professors are added manually.
3. Adjust the environment of the smart contract that will be deployed in Ganache. A screenshot of a phone

   Description automatically generated with medium confidence
4. Deploy smart contract.
5. Copy the smart contract address and than paste at ESSM/src/utils/contract.js at contractAddress.
6. Open console and write “npm install”, than write “npm run dev”, than click the link to open the website.

The relationship between the professor, the subject, and the student is defined by structures (struct). Therefore, there are five different structures in the smart contract code: Professor, Student, Grade, Course and Subject. Regarding the functions, the main function is the setGrade function, which function makes it possible for the professor to set the grade for the respective student. In the setGrade function, it has been made possible that only the professor with the address that has been deployed in the contract can set grades, and not the other professors. Also, the grade must be between 5 and 10. I have manually added some professors, some students, some subjects, some courses, and some grades in the constructor. At first glance, it seems to be a very static project (and this is mainly done for the sack of demonstration), but there are some other functions that make it possible to add professors, students, courses, subjects, and grades in a dynamic form. Other get-functions in the code make it possible to create dynamic drop downdrop-down lists found in the form to add any element. While other set-functions make it possible to add the elements we want. For the front-end I worked in Visual Studio Code with react, HTML and CSS. The connection between the functions of the smart contract and the front-end has been made with Web3 package. The project in front-end is powered by Vite.

The four addresses that I used for professors, were the first four addresses in my Ganache.

Text

Description automatically generatedTable

Description automatically generated

A screenshot of a computer

Description automatically generated

You can set the grade by clicking on the set button.

Graphical user interface, text, application, Word

Description automatically generated

At the admin part, you can add the other elements. I have decided to choose the last address of Ganache to have access as an admin. First, make sure that the last address of your Ganache is the same as this address in the picture:

Text

Description automatically generated

Replace the last address of your Ganache with this address found in the file: RegisterCourse, RegisterGrade, RegisterProfessor, RegisterStudent, and RegisterSubject.jsx. After that, re-deploy the contract with the last address of your Ganache. Copy the smart contract address than and then paste at ESSM/src/utils/contract.js at contractAddress. Now you will be able to manipulate the forms created in the Admin section.

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. Data in the blockchain is immutable and transparent. The only way to change something is through another transaction, which is recorded in the Ledger and can be traced based on the timestamp. Its transparency makes it much more reliable and useful. Since my project requires transparency, I used blockchain technology to rely on it as the best solution compare to other technologeies, i.g., centralized or cloud computing. Once the grade is assigned by the professor to the student, the student's parents can see the transaction made and there is no room for fraud. This is one of the many reasons I used blockchain technology for this project and therefore I like to continue using it in the future.