// This (2.1 Student\_data) is a sample code for an application to insert university student data in blockchain. in programme 2.2, the employee will be able to verify the data from the blockchain.

pragma solidity ^0.4.19;

contract studentDetails{

//Struct is the composite datatype which can store collectively various data types

struct student {

string sname;

uint age;

string qualification;

string college;

address saddress;

}

address owner;

mapping (address => student) students ; //mapping is datatype which has key and value , here key is address and value is struct datatype ,students is variable

//defining the constuctor for the contract

function studentDetails(){

owner = msg.sender;

}

//modifier to check owner or not , it provides reusability to use the same modifier at multi functions in the same contract

modifier ifOwner() {

if (owner!=msg.sender){

throw;

}

R uī6īūñ \_;

}

function insertDetails(string \_sname,uint \_age,string \_qualification,string \_college) ifOwner {

students[msg.sender].saddress = msg.sender;

students[msg.sender].sname = \_sname;

students[msg.sender].age = \_age;

students[msg.sender].qualification = \_qualification;

students[msg.sender].college =\_college;

}

//function to update details , here we can notice we added modifer so that only owner can update the details

function updateDetails(uint \_age) ifOwner {

students[msg.sender].age = \_age;

}

//function to fetch the details of student which are in the blockchain, here also we use the mapping to retrieve the details

function getStudentDetails(address saddress) constant returns(string,uint,string,string){

return (students[saddress].sname, students[saddress].age,

students[saddress].qualification,students[saddress].college );

}

//function to validate the entered details are correct or not

function validateStudent(address valAddress) constant returns(bool){

if ((students[valAddress].saddress==valAddress)){

return true;

}

else {

return false;

}

}

}