# Project: Smart Contract for Banking Application

**Problem Statement:** Write a simple bank smart contract in solidity that allows users to do the following:

* Deposit money into their account
* Withdraw money from their account
* Check balance

After a contract is created, deploy the contract on Ropsten network.

**Steps to Perform:**

Step 1**:** Open remix IDE for writing Solidity smart contract

Step 2**:** Write the following code in Remix IDE

pragma solidity ^0.4.18;

contract SimpleBank

{

mapping (address => uint) private balances;

address public owner;

event LogDepositMade(address accountAddress, uint amount);

function SimpleBank() public {

owner = msg.sender;

}

function deposit() public payable returns (uint) {

require((balances[msg.sender] + msg.value) >= balances[msg.sender]);

balances[msg.sender] += msg.value;

return balances[msg.sender];

}

function withdraw(uint withdrawAmount) public returns (uint remainingBal) {

require(withdrawAmount <= balances[msg.sender]);

balances[msg.sender] -= withdrawAmount;

msg.sender.transfer(withdrawAmount);

return balances[msg.sender];

}

function balance() view public returns (uint) {

return balances[msg.sender];

}}

Step 3**:** Open Metamask extension and connect to Ropsten test network

Step 4**:** Create an account in the Ropsten network

Step 5**:** Request free ether from <https://faucet.metamask.io/>

Step 6**:** Click on run

Step 7**:** Select Injected Web3 Ropsten under environment tab

Step 8**:** Click oncreate button

Step 9**:** Click on submit in Metamask

Step 10**:** Click on Metamask extension

Step 11**:** Click on the contract deployment

Step 12**:** Review your transaction details on Etherscan page