Block chain practical no 3.

1.Create blank folder named bankdaap on d drive

2.Open vscode

3. Click on open folder and select blank folder created on d drive (Import folder)(must have installed nodejs)

4.Open vs terminle and check version of node js using **npn -v**

5.Check version of node using **node -v**

6. Split terminle

7.On first terminle install ganache and web3 using npm install ganache cliweb3@1.2.6 (if ganache is installed proprly in folder node_modules and json files will be available) (for installation of ganache server if you recieved error execute command no.8 otherwise 9)

8.Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

9 Now keep working ganache server on first terminle using .\node_modules\.bin\ganache-cli

10.open second terminle and install http server and solidity compiler using npm install -g http-server npm install solc@0.6.4 start http server using http-server it will written you three adresses check any one.. UI will be visible .. Also add solidity extension.. for further process of scripting stop http server using ctrl+c

11.click node on second terminle.. you will recieve java script trminle write node under node 12.write on terminle Web3=require('web3')

13.now web3 = new Web3("http://localhost:8545")

14.web3.eth.getAccounts(console.log) 15.bytecode=fs.readfilesync('bank_sol

Bank.sol

```
pragma solidity ^0.6.4;
contract Bank {
   uint256 public balance;
   constructor () public {
      balance = 0;
   }
   function getBalance() public view returns (uint256) {
      return balance;
   }
   function withdrawal( uint256 amount ) public  {
      require ( balance >= amount, "Insufficient Balance");
      balance = balance - amount;
   }
   function deposit (uint256 amount) public {
      balance = balance + amount;
   }
}
```

Index.html

```
<html>
   <style>
      .button{
         background-color: #f74c08;
         border: none;
         color: white;
         padding: 15px 25px;
         text-align: center;
         font-size: 16px;
         cursor: pointer;
      }
      .button:hover{
         background-color: #f74c08;
      }
      input{
         border: rgb(207, 207, 207);
         border-radius: 5px;
         background-color: rgb(207, 207, 207);
         height: 40px;
         width: 300px;
         cursor: pointer;
      }
   </style>
   <body>
      <center>
         <h2>Blockchain Bank Application</h2>
```

```
            <input type = 'text' id = 'amount'>
            <p id = "balance"></p>
            <button id = "deposit" class = "button">Deposit</button>
            <button id = "withdraw" class = "button">Withdraw</button>
        </center>

        <script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js@1.0.0-beta.37/dist/web3.min.js"></script>
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>

               <script>
            var contract;

            $(document).ready(function()
            {
                // web3 provider from ganache-cli
                web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
                // smart contract address
                var address = "0x09D650cd1b1aA2FF227365C92c9a66FB76c2e0d1";
                var abi = [{"inputs":[],"stateMutability":"nonpayable","type":"constructor"},{"inputs":[],"name":"balance","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"deposit","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"getBalance","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"withdrawal","outputs":[],"stateMutability":"nonpayable","type":"function"}]

                contract = new web3.eth.Contract(abi, address);
```

```
    contract.methods.getBalance().call().then(function(balance)

    {

        $('#balance').html(balance);

    })

})


$('#deposit').click(function()

    {

        var amt = 0;

        amt = parseInt($('#amount').val());

        // Fetching the various address of web3

        web3.eth.getAccounts().then(function(accounts)

        {

            var acc = accounts[0]

            console.log("acc: " + accounts[0]);

            return contract.methods.deposit(amt).send({ from : acc });

        }).then(function(tx)

        {

            console.log(tx);

        }).catch(function(tx)

        {

            console.log(tx);

        })


})


$('#withdraw').click(function()
        {
          var amt = 0;
          amt = parseInt($('#amount').val());
          // Fetching the various address of web3
          web3.eth.getAccounts().then(function(accounts)
          {
            var acc = accounts[0]
            console.log("acc: " + accounts[0]);
            return contract.methods.withdrawal(amt).send({ from : acc });
          }).then(function(tx)
          {
            console.log(tx);
          }).catch(function(tx)
          {
            console.log(tx);
          })
      })

    </script>
  </body>
</html>
```