**University of Mumbai**

**Practical Journal of**

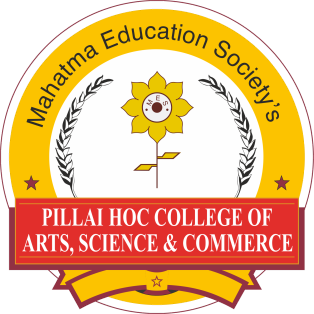
**Blockchain, Natural Language Processing & Deep Learning**

**M.Sc.(Information Technology) Part-II**

**Submitted by**

**Shaikh Obed S.A**

**Seat No: 1172743**



**DEPARTMENT OF INFORMATION TECHNOLOGY PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE, RASAYANI**

***(Affiliated to Mumbai University)***

**RASAYANI, 410207 MAHARASHTRA**

**2023-2024**

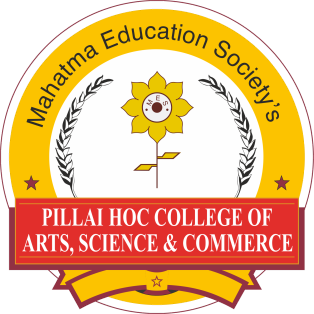
**Mahatma Education Society’s**

**Pillai Hoc College of Arts, Science & Commerce, Rasayani**

#### (Affiliated to Mumbai University)

**RASAYANI–MAHARASHTRA-410207**

### DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Blockchain** was done in computer lab of Mahatma Education Society’s Pillai HOC College of Arts, Science & Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

###### Exam Seat No: 1172743

**Subject In-Charge Coordinator**

###### External Examiner Principal

**Date: College Seal**

# BLOCKCHAIN

## INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **PracticalName** | **Page No.** |
| 1 | Write the following programs for Blockchainin Python:   1. Asimpleclientclassthatgeneratestheprivateandpublickeysby using the built-in Python RS Aalgorithm and test it. 2. A transaction class to send and receive money and test it. 3. Create multiple transactions and display them. 4. Create a blockchain, a genesis block and execute it. 5. Create a mining function and test it. 6. Add blocks to the miner and dump the blockchain. | 1 |
| 2 | Install and configure Go Ethereumand the Mistbrowser. Developand testa sample application. (Meta Mask & Remix) | 8 |
| 3 | Implement and demonstrate the use of the following in Solidity:   1. Variable,Operators,Loops,Decision Making,Strings,Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables. 2. Functions, Function Modifiers, View functions, Pure Functions, Fall back Function, Function Overloading, Mathematical functions, Cryptographic functions**.** | 20 |
| 4 | Implement and demonstrate the use of the following in Solidity:   1. Withdrawal Pattern, Restricted Access. 2. Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces. 3. Libraries,Assembly,Events,Error handling. | 45 |
| 5 | Write a program to demonstrate mining of Ether. | 64 |
| 6 | Demonstrate the running of the blockchain node. | 66 |
| 7 | Create your own blockchain and demonstrate its use. | 69 |

### PRACTICAL:1

###### Asimpleclientclassthatgeneratestheprivateandpublickeysbyusingthebuilt-inPythonRSA algorithm and test it.

importbinascii

importCrypto fromCrypto.PublicKeyimportRSA fromCrypto.SignatureimportPKCS1\_v1\_5

classClient:

def init (self): random=Crypto.Random.new().readself.\_private\_key= RSA.generate(1024, random) self.\_public\_key = self.\_private\_key.publickey()self.\_signer= PKCS1\_v1\_5.new(self.\_private\_key)

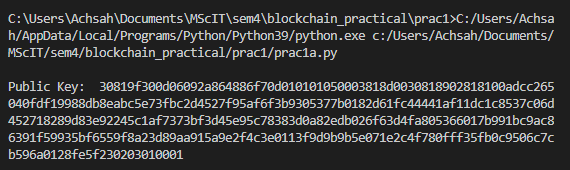
@property defidentity(self):return

binascii.hexlify(self.\_public\_key.exportKey(format="DER")).decode("ascii"

)

Dinesh=Client() print("\nPublicKey:",Dinesh.identity)

### Output:



###### PillaiHOCCollegeofArts,ScienceandCommerce Page|1

1. **Atransactionclassto sendand receivemoneyandtestit.**

importbinasciiimport collectionsimportdatetim e

fromclientimportClientfrom Crypto.Hash import SHA

fromCrypto.SignatureimportPKCS1\_v1\_5

classTransaction:

def init (self,sender,recipient,value):self.sender= sender

self.recipient=recipientself.value=value self.time = datetime.datetime.now()

defto\_dict(self):

identity="Genesis"ifself.sender="Genesis"elseself.sender.identity returncollections.OrderedDict(

{

"sender": identity, "recipient": self.recipient,"value": self.value, "time": self.time,

}

)

defsign\_transaction(self): private\_key=self.sender.\_private\_keysigner= PKCS1\_v1\_5.new(private\_key) h=SHA.new(str(self.to\_dict()).encode("utf8")) returnbinascii.hexlify(signer.sign(h)).decode("ascii")

Dinesh = Client() Ramesh=Client()

t = Transaction(Dinesh, Ramesh.identity, 5.0) print("\nTransaction Recipient:\n",t.recipient)#print("\nTransactionSender:\n",t.sender)

print("\nTransactionValue:\n",t.value)

signature=t.sign\_transaction() print("\nSignature:\n",signature)

### Output:



###### Createmultipletransactionsanddisplaythem.

fromclientimportClient fromtransaction\_classimportTransaction

Dinesh = Client() Ramesh=Client()

t = Transaction(Dinesh, Ramesh.identity, 5.0) print("\nTransaction Recipient:\n",t.recipient)#print("\nTransactionSender:\n",t.sender) print("\nTransaction Value:\n", t.value)

signature=t.sign\_transaction() print("\nSignature:\n",signature)

Dinesh = Client() Ramesh = Client() Seema=Client()Vijay

=Client()

t1=Transaction(Dinesh,Ramesh.identity,15.0)t1.sign\_transaction() transactions = [t1]

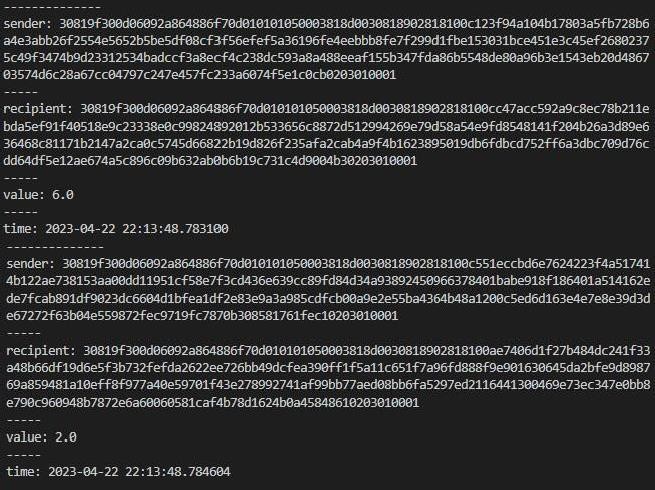
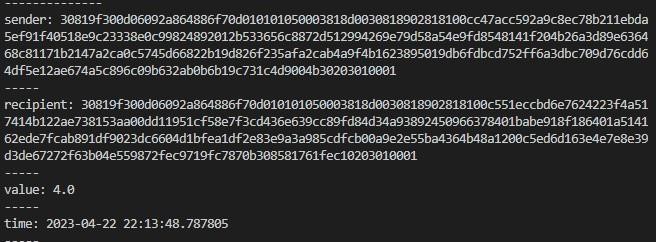
t2=Transaction(Dinesh,Seema.identity,6.0)

t2.sign\_transaction() transactions.append(t2) t3 = Transaction(Ramesh, Vijay.identity, 2.0) t3.sign\_transaction() transactions.append(t3) t4=Transaction(Seema,Ramesh.identity,4.0) t4.sign\_transaction() transactions.append(t4)

fortransactionintransactions:Transaction.display\_transaction(transaction)print("–

————————————–")

### Output:



###### Createablockchain,agenesisblockandexecuteit.

fromclientimportClient fromtransaction\_classimportTransaction

classBlock:

def init (self, client): self.verified\_transactions = [] self.previous\_block\_hash=""self.Nonce= "" self.client=client

defdump\_blockchain(blocks): print(f"\nNumberofblocksinthechain:{len(blocks)}")

fori,blockinenumerate(blocks):print(f"block#

{i}")

fortransactionin block.verified\_transactions:Transaction.display\_transaction(transaction)print("–

————————————–") print("=====================================")

Dinesh=Client() t0=Transaction("Genesis",Dinesh.identity(),500.0)

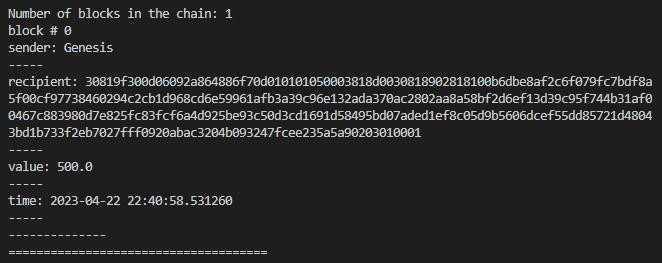
block0=Block(Dinesh)block0.previous\_block\_hash="" NONCE

= None

block0.verified\_transactions.append(t0)digest = hash(block0)

last\_block\_hash=digest

TPCoins = [block0] dump\_blockchain(TPCoins)



###### Createaminingfunctionandtestit.

importhashlib

defsha256(message): returnhashlib.sha256(message.encode("ascii")).hexdigest()

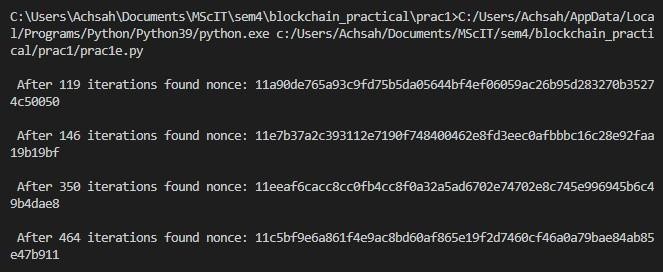
def mine(message, difficulty=1):assert difficulty÷1prefix="1"\*difficulty for i in range(1000):

digest=sha256(str(hash(message))+str(i))if digest.startswith(prefix):

print(f"after{str(i)}iterationsfoundnonce:{digest}")#returnprint(digest) mine("test

message", 2)

###### Output:



1. **Addblockstothemineranddumptheblockchain.**

importdatetime import hashlib

#Createaclasswithtwofunctions

classBlock:

def init (self,data,previous\_hash): self.timestamp=datetime.datetime.now(datetime.timezone.utc)self.data=data self.previous\_hash = previous\_hashself.hash =

self.calc\_hash()

defcalc\_hash(self):

sha=hashlib.sha256()

hash\_str=self.data.encode("utf-8")sha.update(hash\_str) returnsha.hexdigest()

#Instantiatetheclass blockchain=[Block("Firstblock","0")]

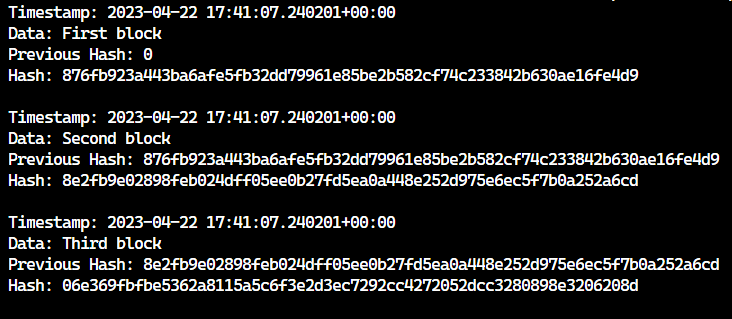
blockchain.append(Block("Second block", blockchain[0].hash))blockchain.append(Block("Third block", blockchain[1].hash))

# Dumping the blockchain forblockinblockchain:print(

f"Timestamp:{block.timestamp}\nData:{block.data}\nPreviousHash:

{block.previous\_hash}\nHash:{block.hash}\n")

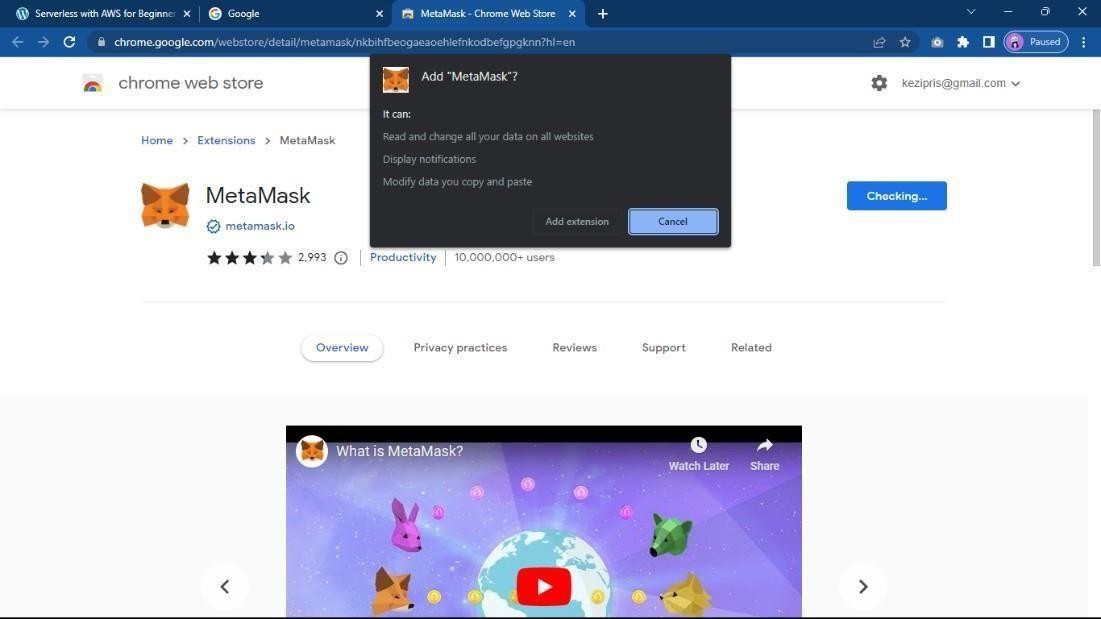
**Output:**

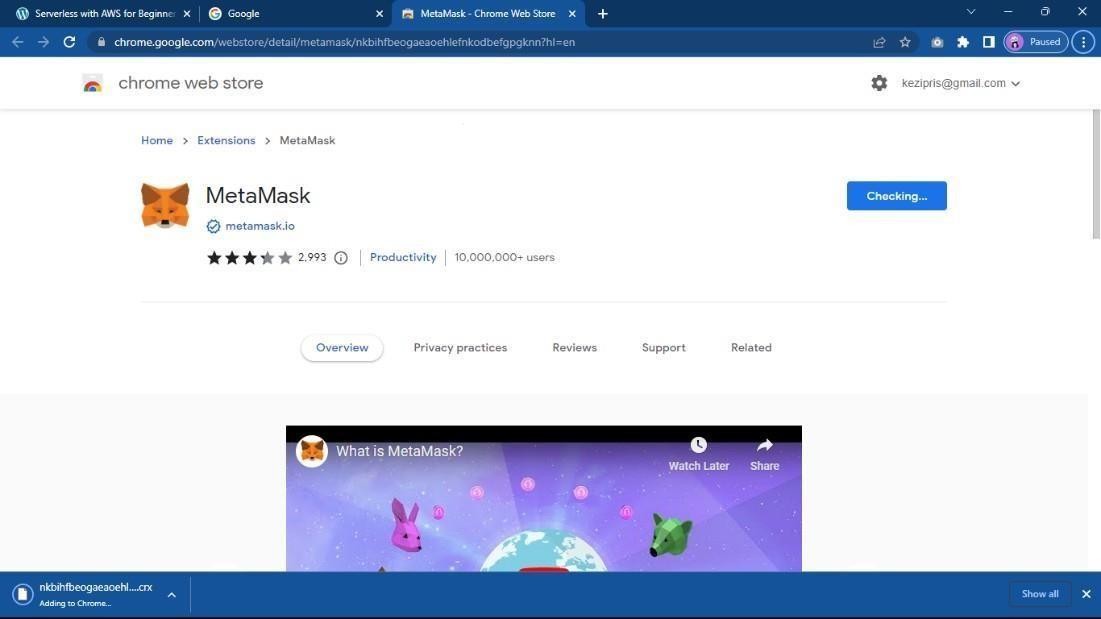


### PRACTICAL-2

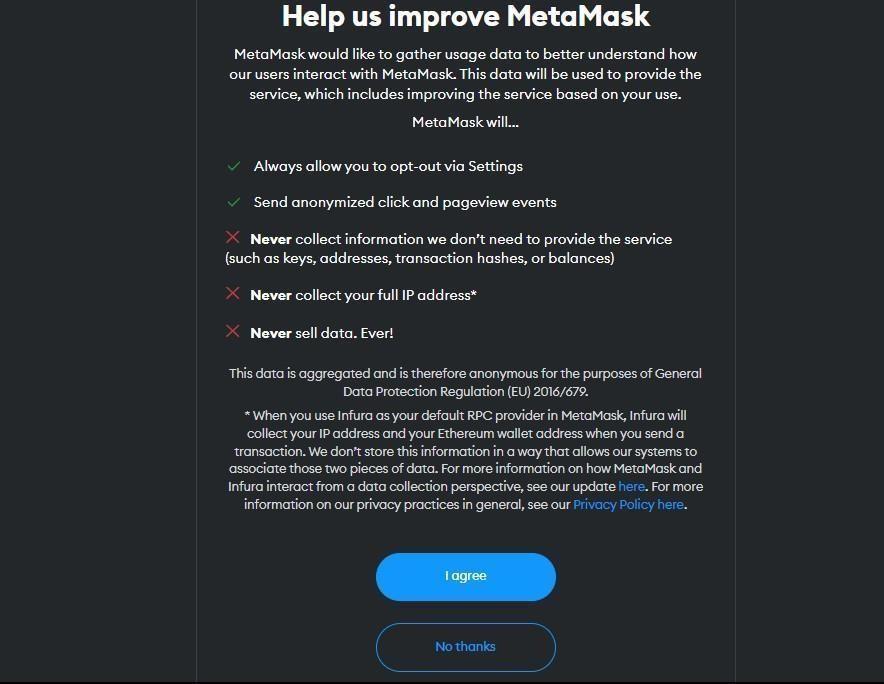
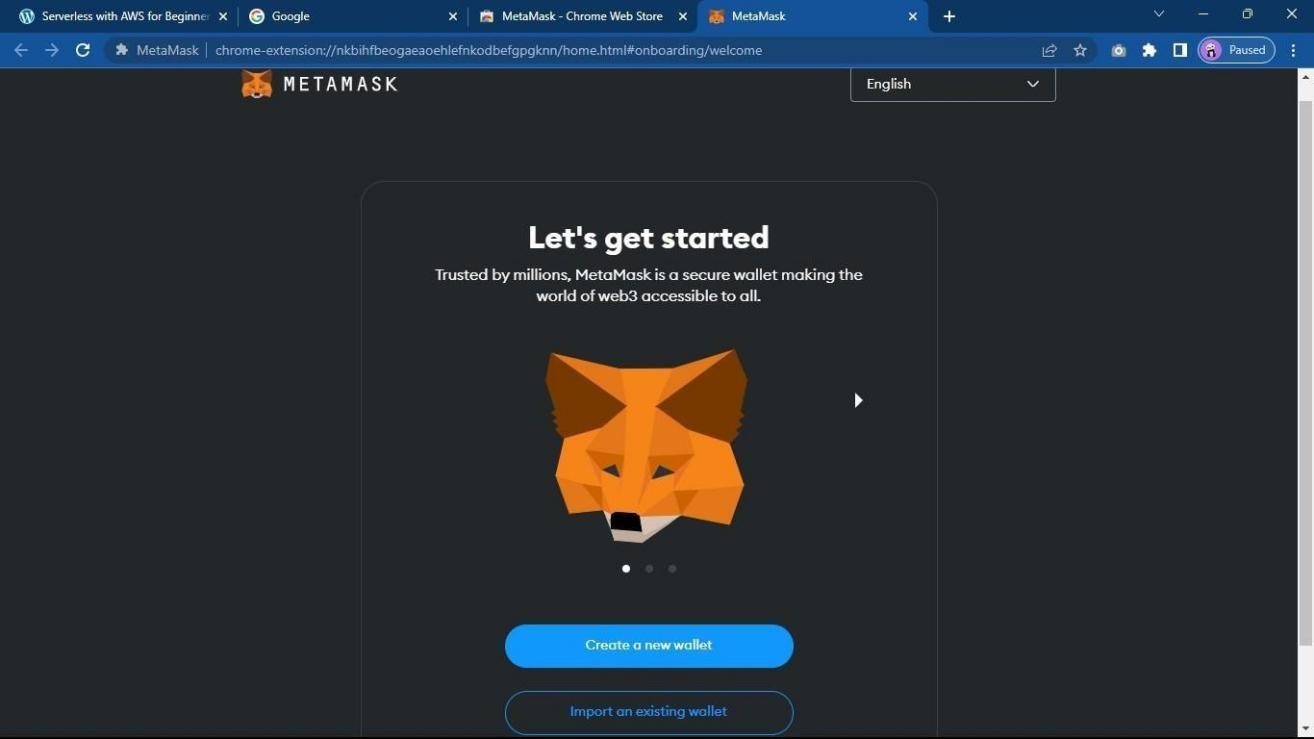
**Aim:Install and configure go Ethereum and themist browser.develop and test asample application (MetaMask& remix)**

**Step1->** InstallMetaMaskextensionforchromefromChromeWebStore

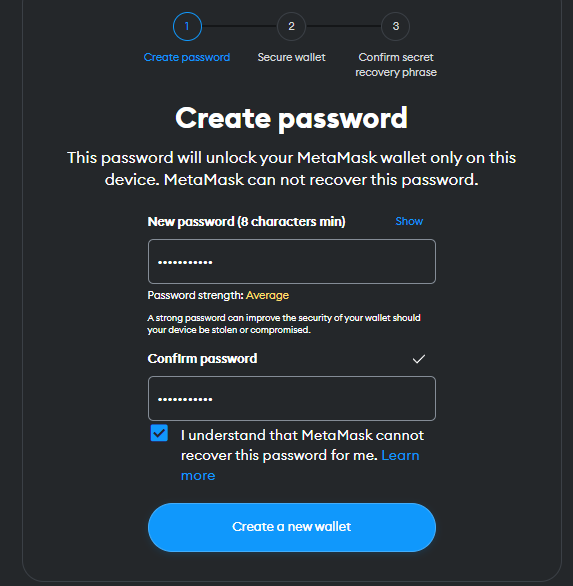


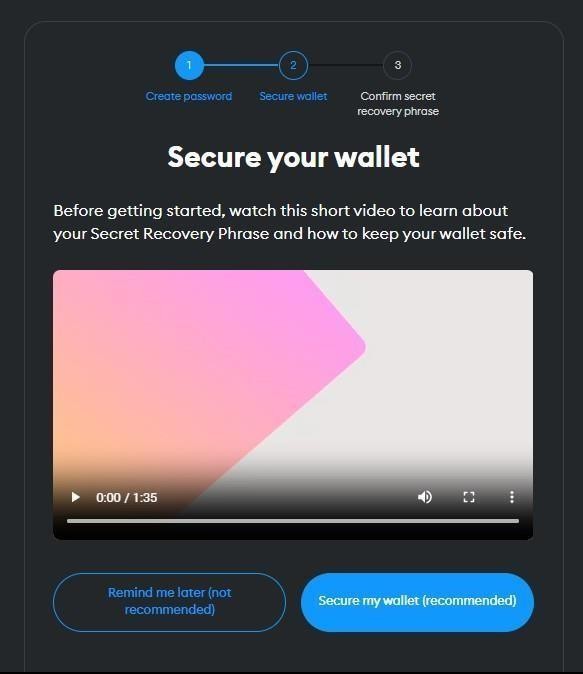


**Step 2->** Click on MetamaskExtension in Extensions. Below page will open in anewtab.Clickon Create a New Wallet. Click on I agree.



**Step 3->**Createapassword.Thispasswordcanbeusedonlyonthedeviceitwas created on. Create a Strong password and click on Create a new Wallet button

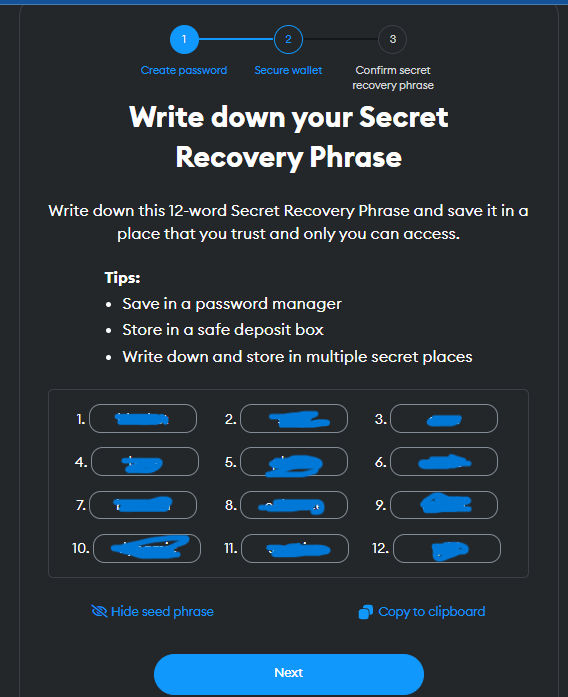




**Step4->** ClickonSecuremywalletbutton,followingwindowwillappear



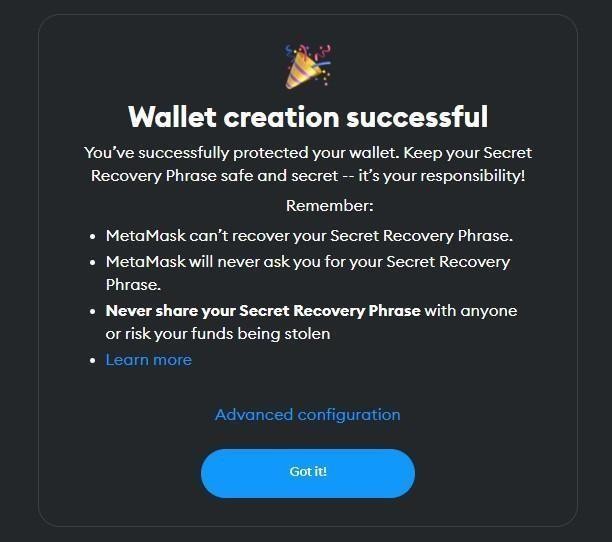
**Step5->** ClickonRevealSecretRecoveryPhrasebuttonandsavethewordsinthesamesequence



**Step6->** EntertherespectivewordsintheemptypositionsandclickConfirm.



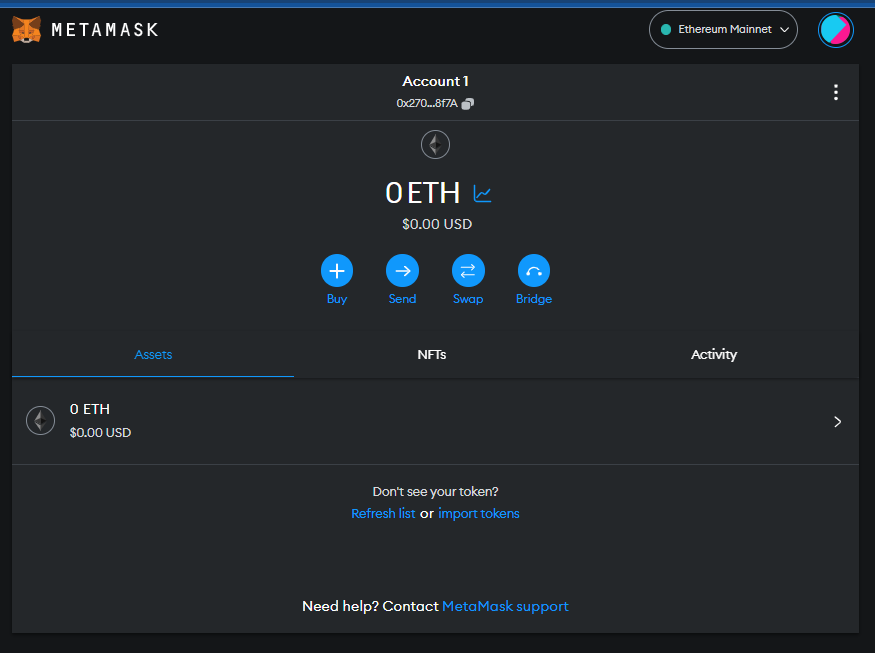
**Step7->** ClickGotit!



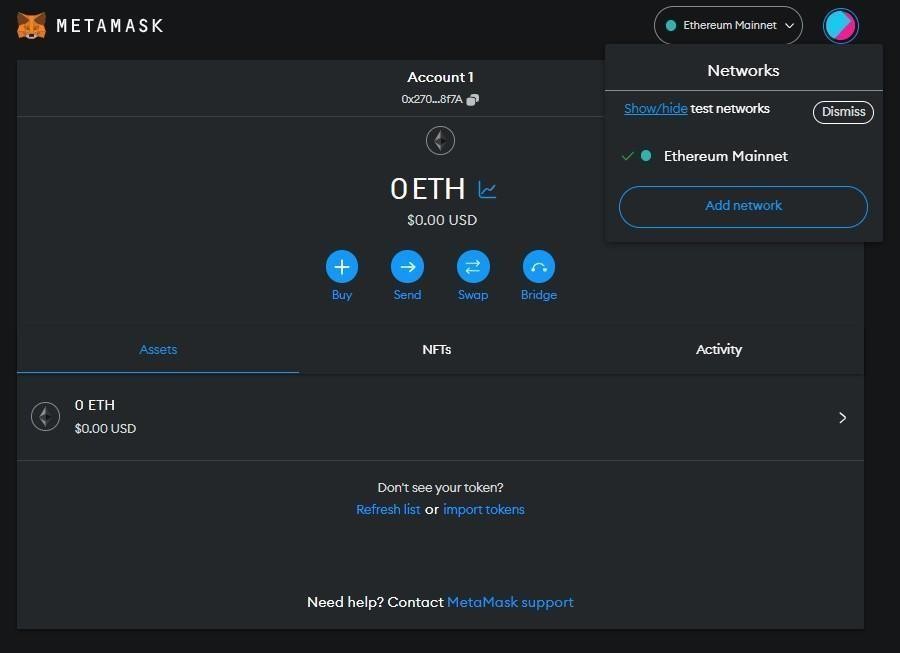
**Step8->** ClickonNext

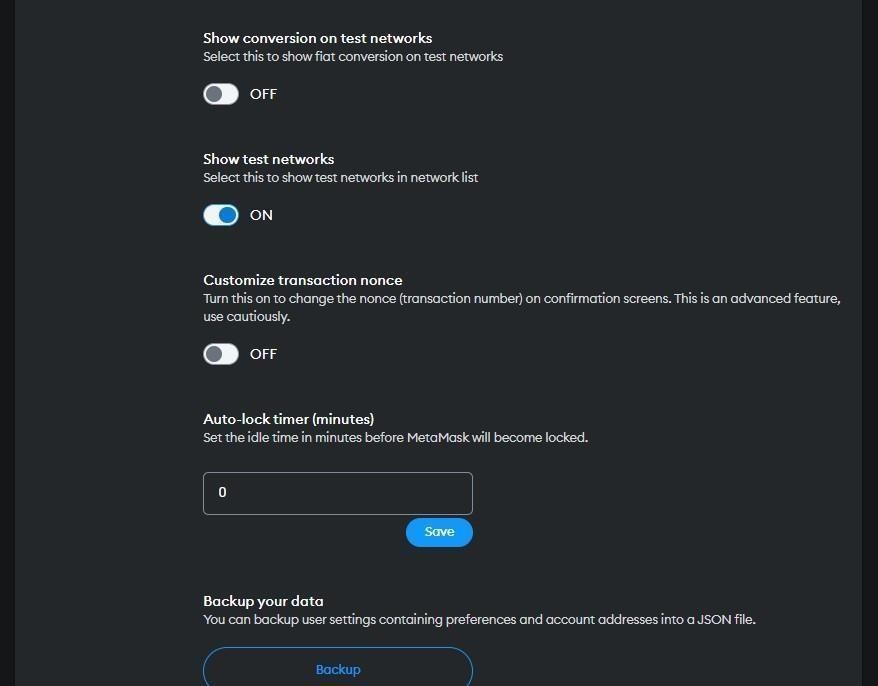


**Step9->** FollowingwillbetheDashboard

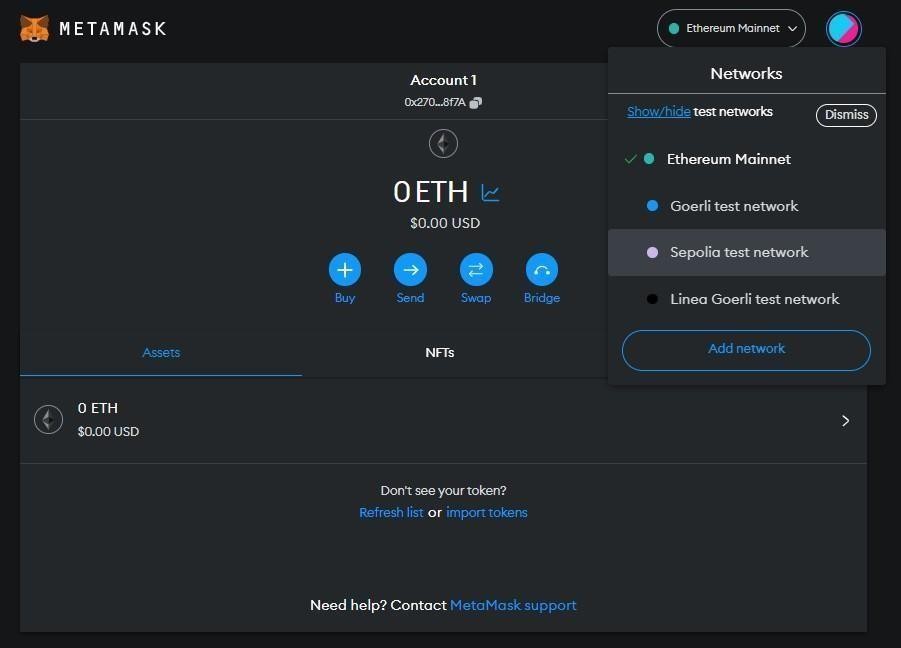


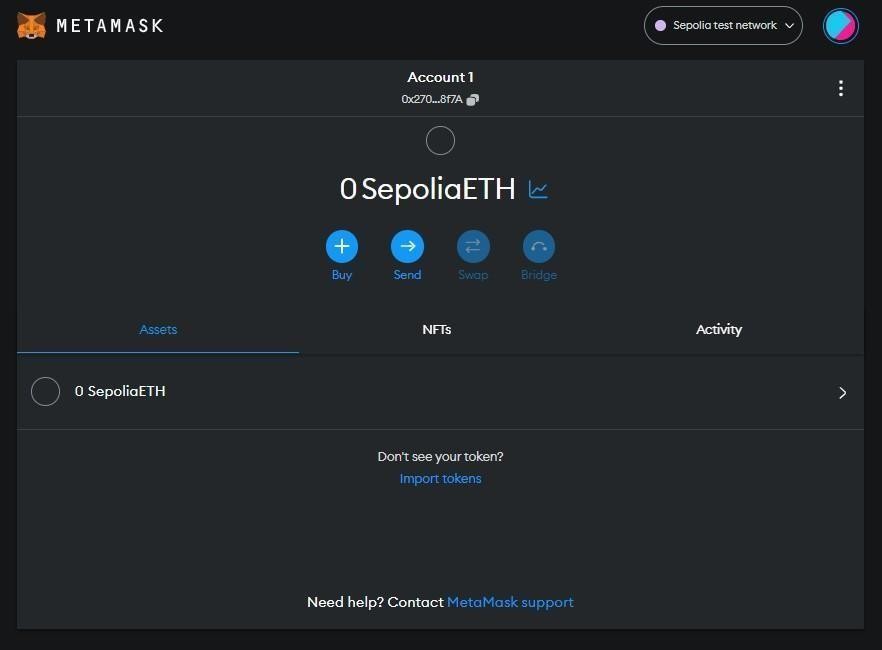
**Step10->**ClickonEthereumMainnet button.NextclickonShow/hidetestnetworks.



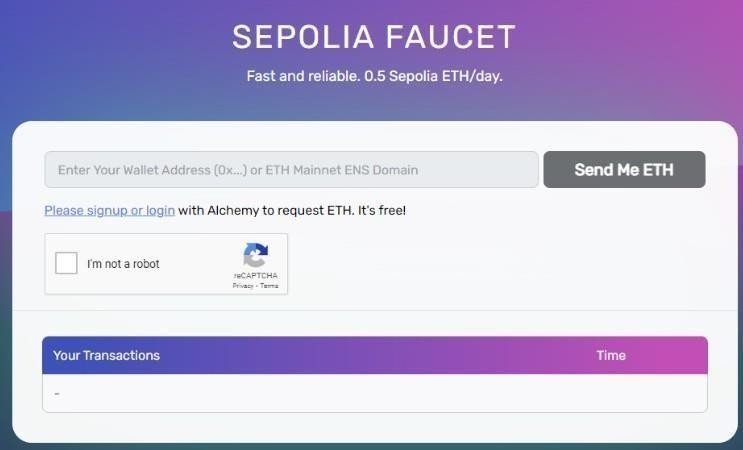


**Step 11->** Check if tesnets are shown byclickingonEtherumMainnetbutton.ClickonSepoliatest network.





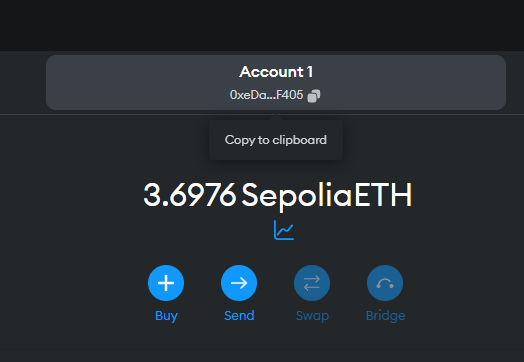
**Step12->**Goto<https://sepoliafaucet.com/>andClickonAlchemyLoginbutton.



**Step13->**Loginto agmailaccountin anotherbrowsertabandclickonSignin withGoogle



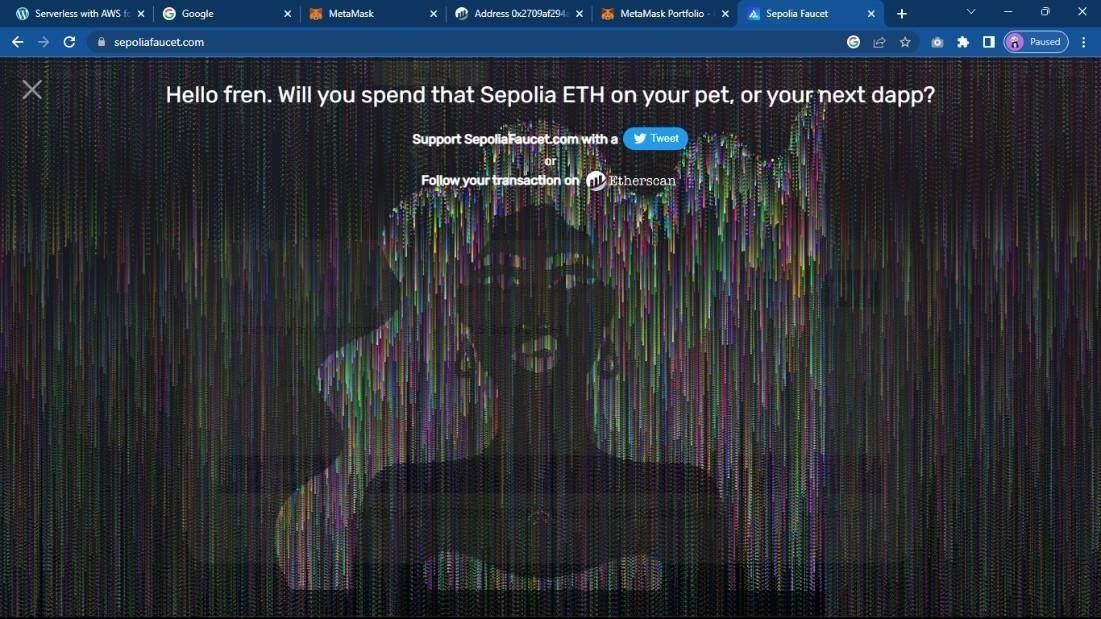
**Step14->**NowgotoMetaMaskandcopytheaccountaddress.



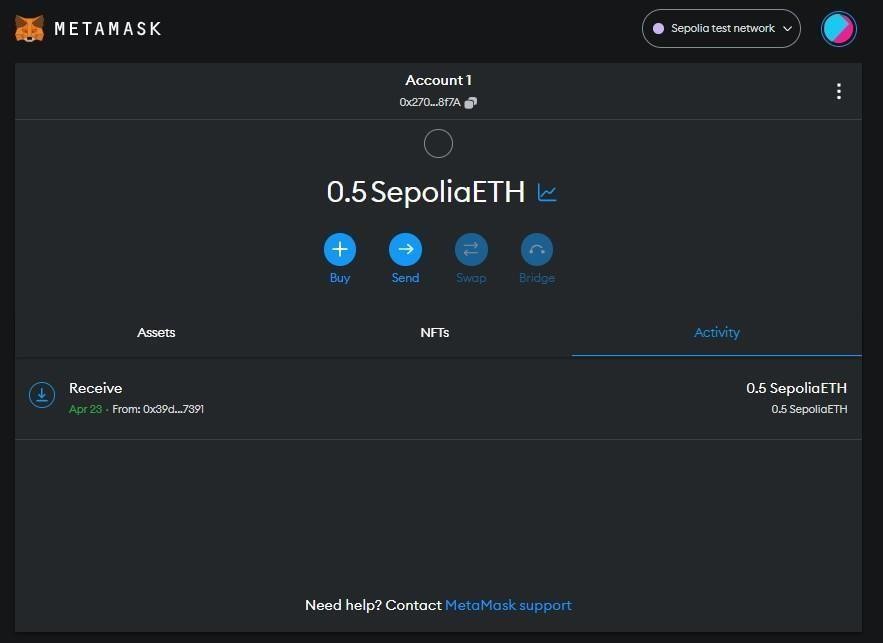
**Step15->**PastetheaddressandclickonSendMeETH.



**Step16->**YourETHtransferissuccesfull.Youshouldseeasimilaranimation.



**Step17->**CheckyourMetaMaskaccountforSepoliatestnetwork.0.5ETHwillbeadded.

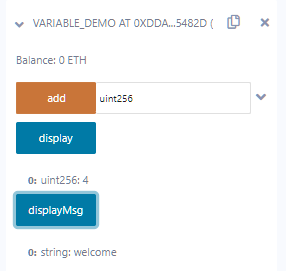


### PRACTICAL-3

**Aim:Implementanddemonstratetheuseofthefollowinginsolidity\**

* 1. TOEXECUTESOLIDITYSCRIPTSGOTO->[HTTPS://REMIX.ETHEREUM.ORG/](https://remix.ethereum.org/)
  2. OPEN CONTRACTS FOLDER AND STARTING WRITING SCRIPTS.THE SCRIPTS ARE COMPILED USING SOLIDITY COMPILER.
  3. THE FOLLOWING SCRIPTS WERE COMPILED USING 0.5.0+COMMIT.1D4F565A SOLIDITY COMPILER
  4. DEPLOYTHESCRIPTSTOEXECUTECODE

###### Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs,Mappings, Conversions, Ether Units, Special Variables

* 1. **Variable**

pragmasolidity^0.5.0;

contractvariable\_demo{uint256s um=4;

//statevariableu int256x;addressa

;

strings="welcome";

functionadd(uint256)public{uint256y= 2;//local variable sum=sum+x+y:sum=sum+x+y;

}

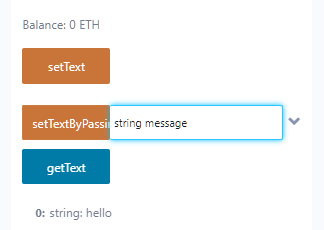
functiondisplay()publicviewreturns(uint256)

{returnsum;

}

Figure1-Displayingvariablevalue

###### Strings



pragmasolidity^0.5.0;

contractLearningStrings{stringt ext;

functiongetText()publicviewreturns(stringmemory){returntex t;

}

functionsetText()public{tex t="hello";

}

functionsetTextByPassing(stringmemorymessage)public{text=m essage;

}

}

Figure2-Beforesettingnewstringvalue

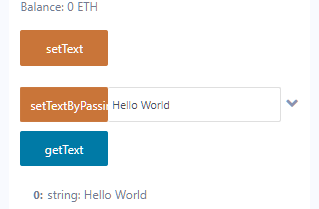
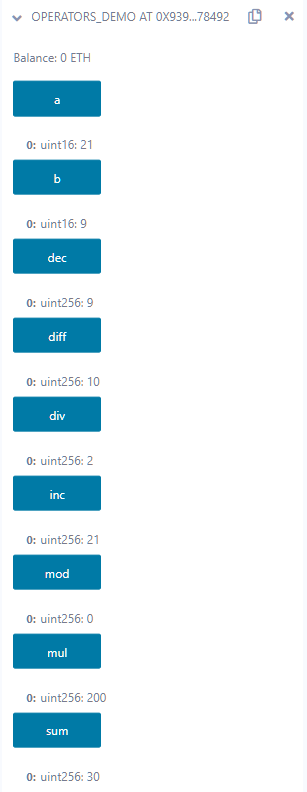


Figure3-Aftersettingstringvalue

###### Operators



pragmasolidity^0.5.0;

contractSolidityTest{uint 16publica=20;uint16pu blicb=10; uint256publicsum=a+b;uint256pub licdiff=a- b;uint256publicmul=a\*b;uint256p ublicdiv=a/b;uint256publicmod=a

%b;uint256publicdec= -- b;uint256publicinc= ++a;

}

Figure4-Alloperatorsofsoliditydisplayed

###### Array

pragmasolidity^0.5.0;contract arraydemo

{

//StaticArray uint[6]arr2=[10,20,30];

functiondispstaticarray()publicviewreturns(uint[6]memory)

{

returnarr2;

}

//DynamicArray uint x=5;

uint[] arr1; functionarrayDemo()public

{

while(x>0)

{

arr1.push(x);x=x- 1;

}

}

functiondispdynamicarray()publicviewreturns(uint[]memory)

{

returnarr1;

}

}

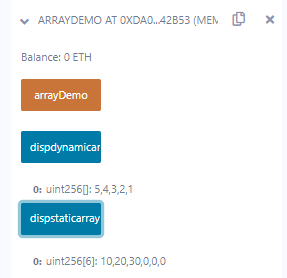


Figure5-Arraydisplayed

###### DecisionMaking

IfElse

pragmasolidity^0.5.0;contract ifelsedemo

{

uinti=10; functiondecision\_making()publicviewreturns(stringmemory)

{

if(i%2==0)

{

}

else

{

}

}

}

return"even";

return"Odd";



Figure6-Ifelseoutput

###### Loops

pragmasolidity^0.5.0;contract loopDemo

{

ForLoop

uint[] data;

functionforDemo()publicreturns(uint[]memory)

{

for(uinti=0;i<10;i++){ data.push(i);

}

returndata;

}

functiondisp()publicviewreturns(uint[]memory)

{

returndata;

}

}

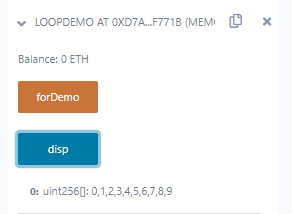


Figure7-Appendingvaluestoarrayusingforloop

pragmasolidity^0.5.0;contract whiledemo

WhileLoop

{

uint[]data;uintx=0;

functionwhileLoopDemo()public

{

while(x<5)

{

data.push(x); x=x+1;

}

}

functiondispwhileloop()publicviewreturns(uint[]memory)

{

returndata;

}

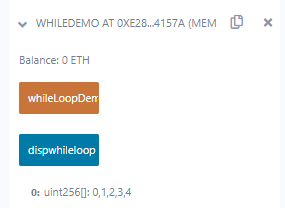
}

Figure8-Appendingvaluestoarrayusingwhileloop



Do

While

pragmasolidity^0.5.0;

//Creatingacontractcontra ctDoWhile{

//Declaringadynamicarrayuint256 []data;

//Declaringstatevariable uint8j=0;

//Definingfunctiontodemonstrate

//'Do-Whileloop' functionloop()publicreturns(uint256[]memory){do{

j++;

data.push(j);

}while(j<5);retur ndata;

}

functiondisplay()publicviewreturns(uint256[]memory){return data;

}

}

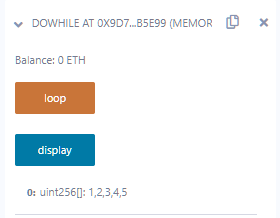
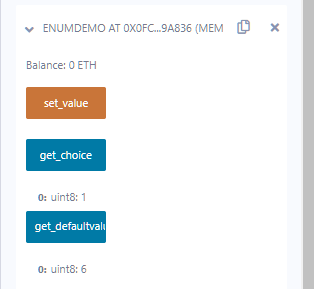


Figure9Appendingvaluestoarrayusingdowhileloop

###### Enums



pragmasolidity^0.5.0;

contractenumdemo{enumw eek\_days{

Monday,Tuesday

,Wednesday,Thu rsday,Friday,S aturday,Sunday

}

week\_days week;week\_dayschoice;

week\_daysconstantdefault\_value=week\_days.Sunday;

functionset\_value()public{choic e=week\_days.Tuesday;

}

functionget\_choice()publicviewreturns(week\_days){returncho ice;

}

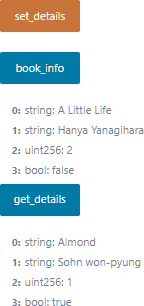
functionget\_defaultvalue()publicviewreturns(week\_days){returnd efault\_value;

}

}

Figure10-Accessingenumvalues

###### Structs



pragmasolidity^0.5.0;

contractstructdemo{struct Book{

stringname;string author;uint256id; boolavailability;

}

Bookbook2; Bookbook1=Book("ALittleLife","HanyaYanagihara",2,false);

functionset\_details()public{ book2=Book("Almond","Sohnwon-pyung",1,true);

}

functionbook\_info()p ublic

viewretur ns(

stringmemory,s tringmemory,ui nt256,

bool

)

{

return(book1.name,book1.author,book1.id,book1.availability);

}

functionget\_details()public viewreturn

s(

stringmemory,stringmemory,uint256,bool

)

{

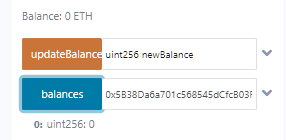
return(book2.name,book2.author,book2.id,book2.availability);

}

}

Figure 11-Structuredatatypeinsolidity

###### Mappings



pragmasolidity^0.5.0;

contractLedgerBalance{ mapping(address=>uint256)publicbalances;

functionupdateBalance(uint256newBalance)public{balances[ms g.sender] =newBalance;

}

}

contractUpdater{ functionupdateBalance()publicreturns(uint256)

{LedgerBalanceledgerBalance=newLedgerBalance();returnl edgerBalance.balances(address(this));

}

}

Figure12-Beforeupdatingbalance

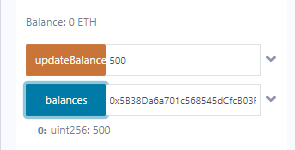


Figure13-Afterupdatingbalance

###### Conversions

//SPDX-License- Identifier:MITpragma solidity^0.8.0;

contractImplicitConversion{ functionadd()publicpurereturns(uint256){uint256a=10

;

uint256b=20;retur na+b;

}

}

contractExplicitConversion{ functionconvert()publicpurereturns(bytesmemory){stringmemo

rystr="Hello World"; bytesmemoryb=bytes(str);returnb

;

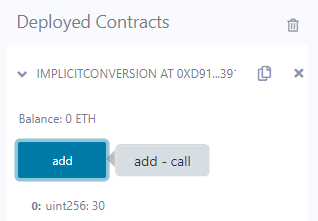
}

}

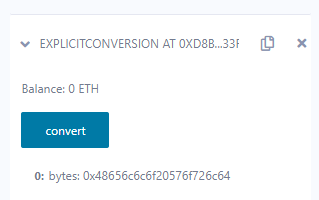
**Step1->** Deploybothcontracts



**Step2->** OpenImplicitConversionandclickonaddbuttontosumanddisplayvalue



**Step3->** OpenExplicitConversionandclickonconvertbutton



###### EtherUnits

//SPDX-License- Identifier:MITpragma solidity^0.8.0;

contractSolidityTest{ functionconvert\_Amount\_to\_Wei(uint256Amount)public

pure returns(uint256)

{

returnAmount\*1wei;

}

functionconvert\_Amount\_To\_Ether(uint256Amount)publi c

pure returns(uint256)

{

returnAmount\*1ether;

}

functionconvert\_Amount\_To\_Gwei(uint256Amount)public pure

returns(uint256)

{

returnAmount\*1gwei;

}

functionconvert\_seconds\_To\_mins(uint256\_seconds)pub lic

pure returns(uint256)

{

return\_seconds/60;

}

functionconvert\_seconds\_To\_Hours(uint256\_seconds)pu blic

pure returns(uint256)

{

return\_seconds/3600;

}

functionconvert\_Mins\_To\_Seconds(uint256\_mins)public pure

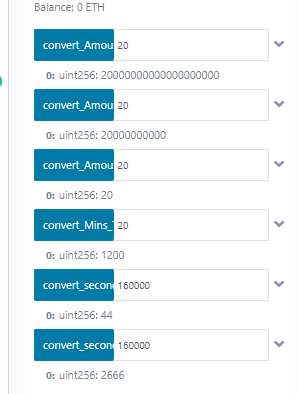
returns(uint256)

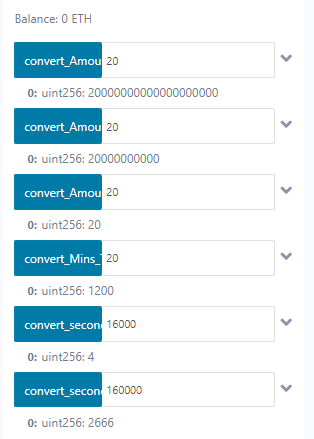
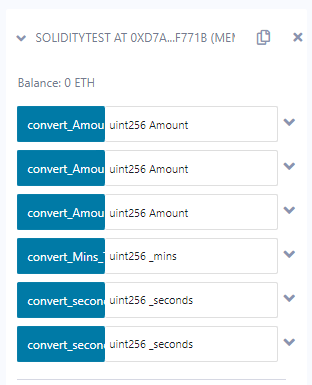
{

return\_mins\*60;

}

}



**Step1->** Providevaluestoeachfunctionandclickonthem

###### SpecialVariables

//SPDX-License- Identifier:MITpragma solidity^0.8.0;

contractSpecial\_Variables{mapping(addre ss=>uint256)rollNo;

functionsetRollNO(uint256\_myNumber)public{rollNo[msg.sende r] =\_myNumber;

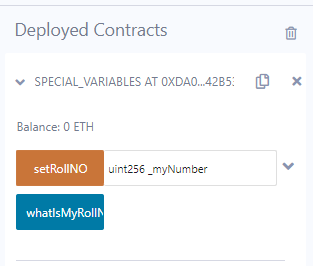
}

functionwhatIsMyRollNumber()publicviewreturns(uint256){returnr ollNo[msg.sender];

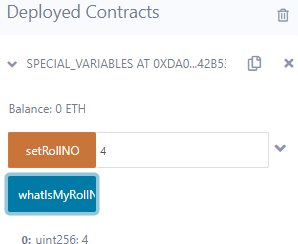
}

}

**Step1->** DeploycontractSpecialVariables

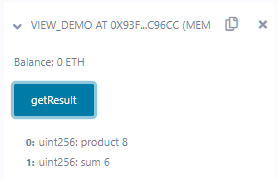


**Step 2->** Input a number for setRollNO function and click on it &whatIsMyRollNumber button



###### Functions,FunctionModifiers,Viewfunctions,PureFunctions,FallbackFunction,Function Overloading, Mathematical functions, Cryptographic functions

* 1. **ViewFunctions**



pragmasolidity^0.5.0;

contract

view\_demo{uint256 num1

=2;uint256 num2=4;

functiongetResult()publicviewreturns(uint256product,uint256sum)

{product=num1\*num2; sum=num1+num2;

}

Figure14-Viewfunctiondemo

###### PureFunctions

pragmasolidity^0.5.0;

contractpure\_demo{ functiongetResult()publicpurereturns(uint256product,uint256sum)

{uint256num1=2; uint256num2=4;product=n um1\*num2;sum=num1

+num2;

}

}

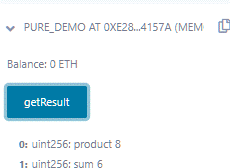


Figure15-Purefunctionoutput

###### MathematicalFunctions

pragmasolidity^0.5.0; contract Test{

functionCallAddMod()publicpurereturns(uint){return addmod(7,3,3);

}

functionCallMulMod()publicpurereturns(uint){return mulmod(7,3,3);

}

}

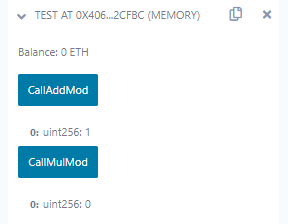


Figure16-Mathematicalfunctionsinsolidity

### CryptographicFunctions

pragmasolidity^0.5.0;contract Test{

functioncallKeccak256()publicpurereturns(bytes32result){return keccak256("BLOCKCHAIN");

}

functioncallsha256()publicpurereturns(bytes32result){return sha256("BLOCKCHAIN");

}

functioncallripemd()publicpurereturns(bytes20result){return ripemd160("BLOCKCHAIN");

}

}

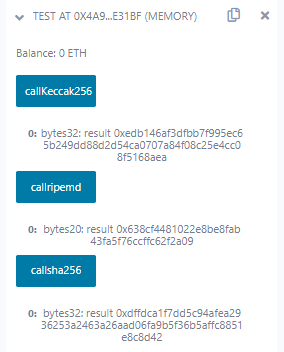


Figure17-Cryptographyalgorithmsinsolidity

###### Functions

// SPDX-License-Identifier: MITpragmasolidity>=0.4.22<0.9.0;

contractTest{

functionreturn\_example()pub lic

purereturn s(

uint256,uint2 56,uint256,st ringmemory

)

{

uint256num1=10;uint256n um2=16;

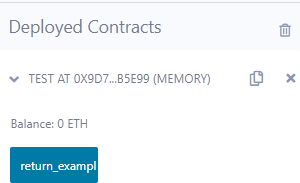
uint256sum=num1+num2;uint25 6prod=num1\*num2;uint256diff

=num2-num1; stringmemorymessage="Multiplereturnvalues";return(sum, prod,diff,message);

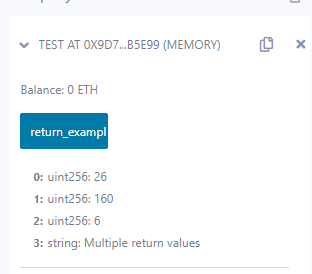
}

}

**Step1->** DeployTestContract



**Step2->** Clickonreturn\_examplebuttontodisplayallvalues



###### FallbackFunction

//SPDX-License- Identifier:MITpragma solidity^0.5.12;

contractA{

uint256n;

functionset(uint256value)external{n=va lue;

}

function()externalpayable{n=0

;

}

}

contractexample{ functioncallA(Aa)publicreturns(bool){

(boolsuccess,)=address(a).call(abi.encodeWithSignature("setter()"));require(success

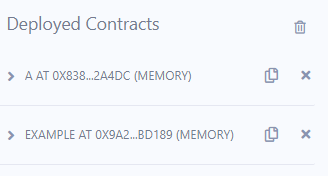
);

addresspayablepayableA=address(uint160(address(a)));return (payableA.send(2ether));

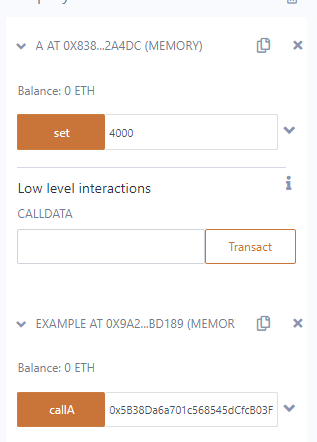
}

}

**Step1->** DeploybothA&examplecontracts



**Step2->** Providevaluestobothdeployedcontractsaccordingly(useanyaddress)



### FunctionOverloading

//SPDX-License- Identifier:MITpragma solidity^0.8.0;

contractOverloadingExample{ functionadd(uint256a,uint256b)publicpurereturns(uint256){returna+b;

}

functionadd(stringmemorya,stringmemoryb)public pure

returns(stringmemory)

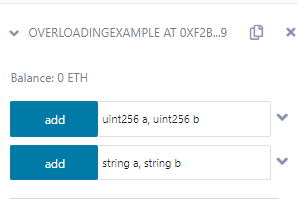
{

returnstring(abi.encodePacked(a,b));

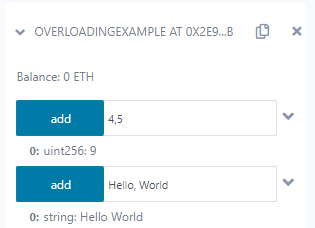
}

}

**Step1->** DeployOverloadingExamplecontract



**Step2->** Giveintegerandstringvaluestobothaddfunctionsasbelow



### Functionmodifiers

//SPDX-License- Identifier:MITpragma solidity^0.5.0;

contractExampleContract{ addresspublicowner=0x5B38Da6a701c568545dCfcB03FcB875f56beddC4;uint256pub liccounter;

modifieronlyowner(){ require(msg.sender==owner,"Onlythecontractownercancall");

\_;

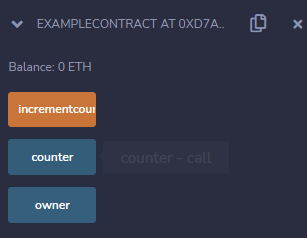
}

functionincrementcounter()publiconlyowner{counter++

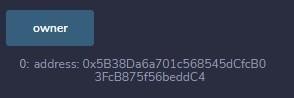
;

}

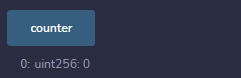
}



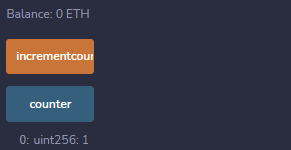
**Step1->** Clickonownerbutton



**Step2->** Clickoncounterbuttoninitiallyitis 0.



**Step 3->** Then click on increment counter button and again clickoncounterbutton,thecounterhas been increased



### PRACTICAL-4

**Aim:Implementanddemonstratetheuseofthefollowinginsolidity**

### WithdrawalPattern,RestrictedAccess

##### WithdrawalPattern

//SPDX-License- Identifier:MITpragma solidity0.8.18;

contractWithdrawalPattern{addre sspublicowner; uint256publiclockedbalance;uint256p ublicwithdrawablebalance;

constructor(){ owner=msg.sender;

}

modifieronlyowner(){ require(msg.sender==owner,"Onlytheownercancallthisfunction");

\_;

}

functiondeposit(uint256amount)public payable{require(amount>0,"Amountmustbegreaterthanzero");lo ckedbalance+=amount;

}

functionwithdraw(uint256amount)publicpayableonlyowner{require( amount<=withdrawablebalance,"Insufficient withdrawablebalance"

);

withdrawablebalance-

=amount;payable(msg.sender).transfer(amount);

}

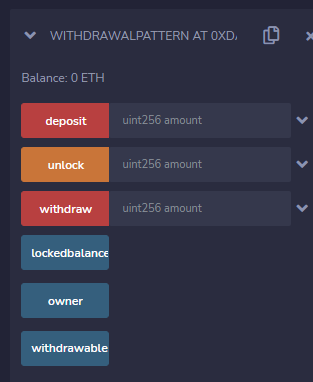
functionunlock(uint256amount)publiconlyowner{ require(amount<=lockedbalance,"Insufficientlockedbalance");lockedbal ance-=amount;

withdrawablebalance+=amount;

}

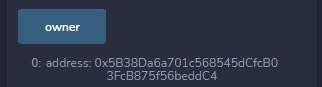
}

###### Output:

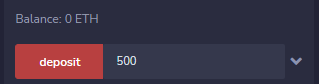


**Flowofexecution**

**Step1->** Clickonowner



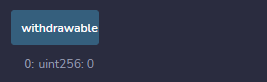
**Step2->** Enteranamountandclickondeposit



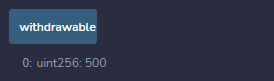
**Step3->** Clickonlockedbalancebuttontodisplaythelockedamountintheaccount



**Step4->** Clickonwithdrawablebalancebutton

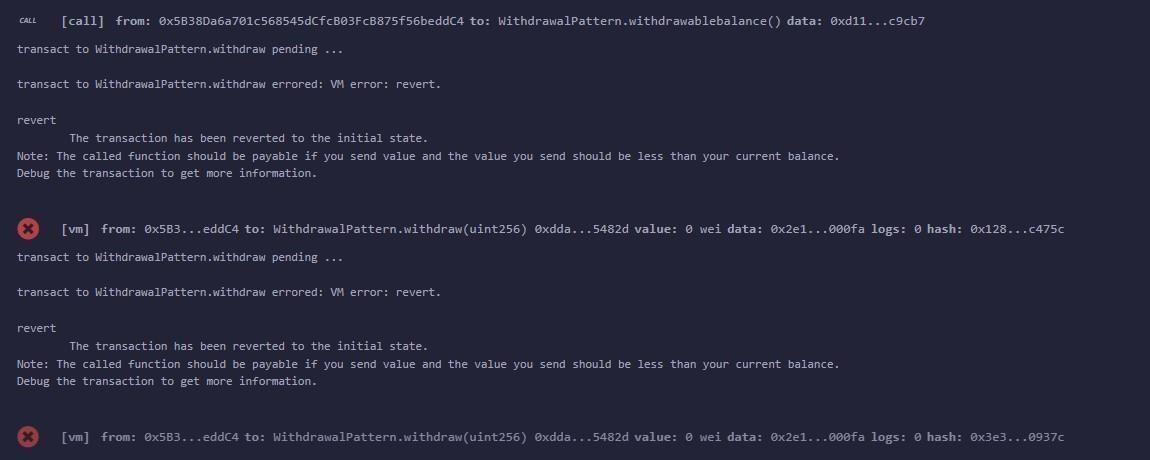


**Step 5->** Clickonunlockbuttonandenteranyamounttotransferamounttowithdrawable balance.Check locked balance and withdrawable balance.



**Step6->** EnteranyamountyouwanttowithdrawandClickthewithdrawbutton.

Youshouldgetanerrorandthetransactionshouldbereverted.



##### RestrictedAccess

//SPDX-License- Identifier:MITpragma solidity^0.8.18;

contractRestrictedAccess{ addresspublicowner=msg.sender; uint256publiccreationTime=block.timestamp;

modifieronlyBy(address\_account){ require(msg.sender==\_account,"Sendernotauthorized!");

\_;

}

modifieronlyAfter(uint256\_time){ require(block.timestamp>=\_time,"Functionwascalledtooearly!");

\_;

}

modifiercosts(uint256\_amount){ require(msg.value>=\_amount,"NotenoughEtherprovided!");

\_;

}

functionforceOwnerChange(address\_newOwner)public payablecosts(200

ether)

{

owner=\_newOwner;

}

functionchangeOwner(address\_owner)publiconlyBy(owner){owner=\_o wner;

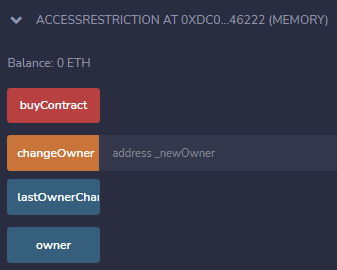
}

functiondisown()publiconlyBy(owner)onlyAfter(creationTime+3weeks){deleteowner;

}

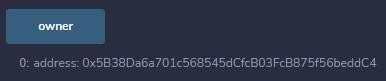
}

###### Output:

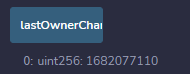


**Flowofexecution**

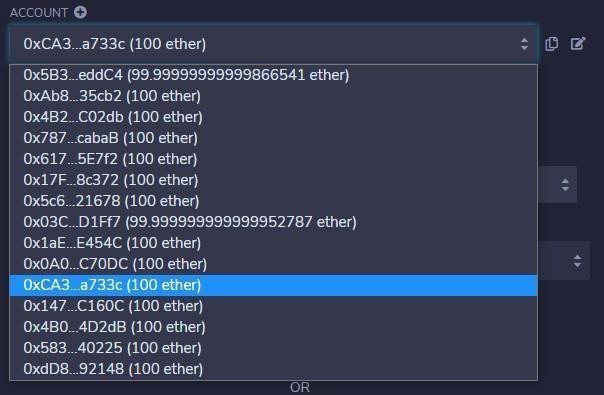
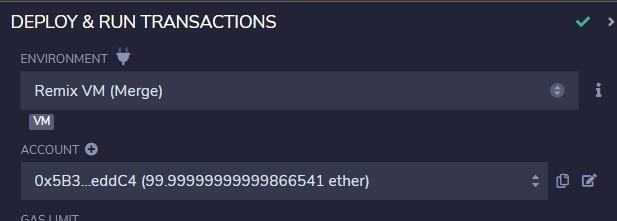
**Step1->** Clickonownertocreateanownerobject



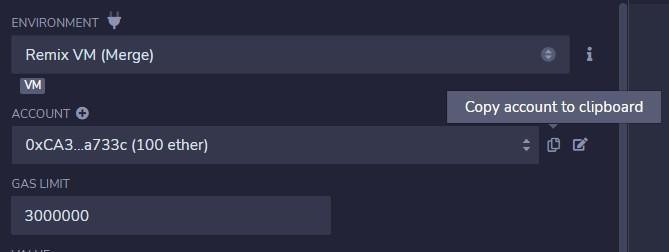
**Step2->** ClickonlastOwnerChangebutton



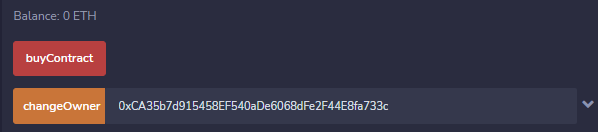
**Step3->** ChangetheaddressoftheaccountfromAccountdropdowninDeploytabofRemix IDE.



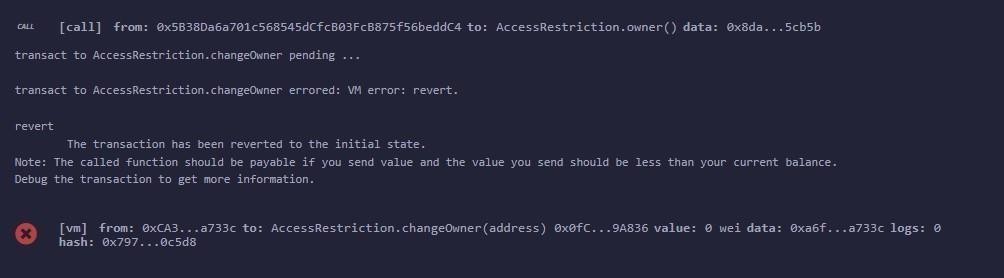
**Step4->** Copytheaddress



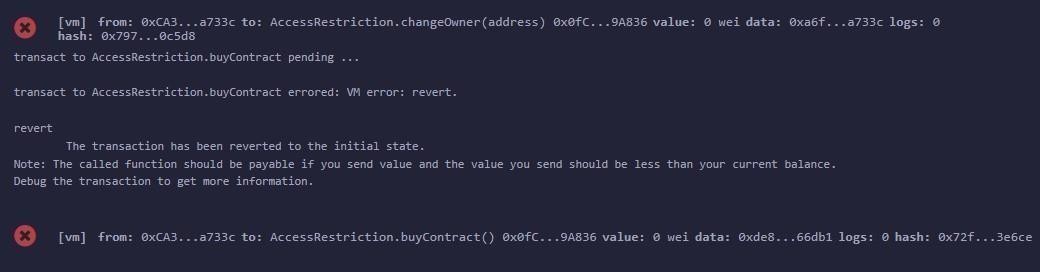
**Step5->** PastetheaddressinchangeOwnerinputandclickonchangeOwner.



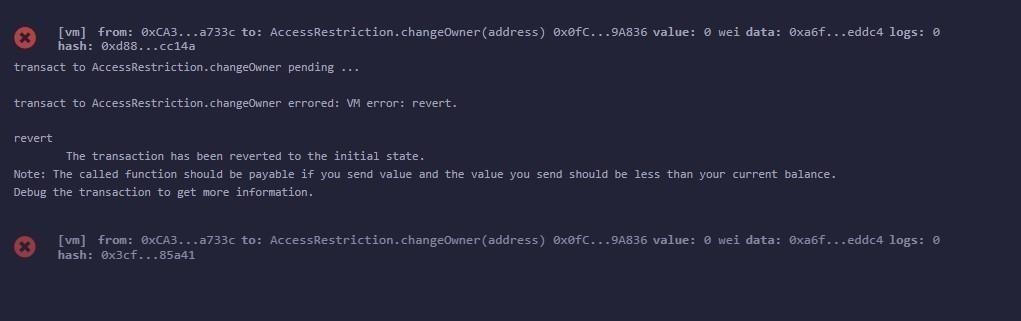
**Step6->** Youshouldgetanerrorasfollowing



**Step7->** Ifyouclickonbuycontractitshouldgiveanerrorasfollows

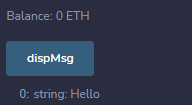


**Step 8->** Now,pastetheactualaddressoftheaccountinthechangeownerinputand click on changeowner



### Contracts,Inheritance,Constructors,AbstractContracts,Interfaces

* 1. Contract



Output

pragmasolidity^0.5.0;

contractContract\_demo{stringm essage="Hello";

functiondispMsg()publicviewreturns(stringmemory){returnmes sage;

}

}

##### Inheritance

}

functionshow\_value()publicviewreturns(uint256){returnc

c.getValue();

}

pragmasolidity>=0.4.22<0.6.0;

contractParent{uint256in ternalsum;

functionsetValue()external{uint256a= 10;

uint256b=20;sum=a

+b;

}

}

contractchildisParent{ functiongetValue()externalviewreturns(uint256){return

sum;

}

}

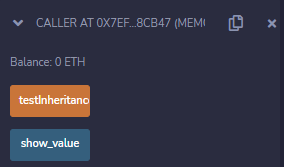
contractcaller{ childcc=newchild();

functiontestInheritance()publicreturns(uint256){ cc.setValue();

returncc.getValue();

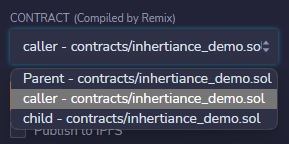
}

**Output:**

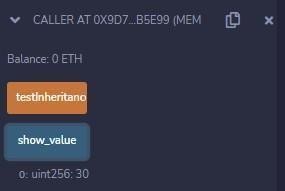


### Flowofexecution

**Step1->** SelectcallercontracttodeployinContractanddeploy



**Step2->** ClicktestInheritanceandthenclickonshow\_valuetoviewvalue



* 1. AbstractContracts

//SPDX-License- Identifier:MITpragma solidity^0.5.17;

contractCalculator{ functiongetResult()externalviewreturns(uint256);

}

contractTestisCalculator{constr uctor()public{}

functiongetResult()externalviewreturns(uint256){uint256a=1

;

uint256b=2; uint256result=a+b;retur nresult;

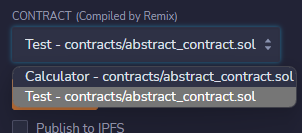
}

}

###### Outputs:

**Flowofexecution**

**Step1->** SelectTestcontractanddeploy

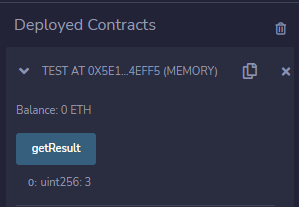


**Step2->** Thecontactwilldeployasbelow



Outputs

**Step3->** ClickongetResulttogetsumofa+b



1. Constructors



//SPDX-License- Identifier:MITpragma solidity^0.5.0;

// Creating a contractcontractconstructorEx ample{

stringstr;

constructor()public{ str="GeeksForGeeks";

}

functiongetValue()publicviewreturns(stringmemory){returnst r;

}

###### Flowofexecution

**Step1->** ClickongetValuetoprintstrin



##### Interfaces

pragmasolidity^0.5.0;

interfaceCalculator{ functiongetResult()externalviewreturns(uint);

}

contractTestisCalculator{constr uctor()public{}

functiongetResult()externalviewreturns(uint){uinta=1

;

uintb=2; uintresult=a+b;returnresu lt;

}

}

###### Outputs:

**Flowofexecution**

**Step1->** ClickongetResulttodisplaysum



1. **Libraries,Assembly,Events,Errorhandling.**

##### Libraries myLib.sol Code

}

//SPDX-License- Identifier:MITpragmasolidity>=0

.7.0<0.9.0;

librarymyMathLib{ functionsum(uint256a,uint256b)publicpurereturns(uint256){returna+b;

}

functionexponent(uint256a,uint256b)publicpurereturns(uint256){returna\*\*b

;

using}\_library.solCode

//SPDX-License- Identifier:MITpragmasolidity>=0

.7.0<0.9.0;

import"contracts/myLIB.sol";con tractUseLib{

functiongetsum(uint256x,uint256y)publicpurereturns(uint256){ returnmyMathLib.sum(x,y);

}

functiongetexponent(uint256x,uint256y)publicpurereturns(uint256){returnmyMathLi b.exponent(x,y);

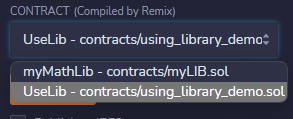
}

}

###### Outputs:

**Flowofexecution**

**Step1->** ChangecontracttoUseLibanddeploy.



**Step2->** Thedeployedcontractshouldbesameasbelow



**Step3->** Inputvaluestobothgetexponentandgetsumfunctionsasbelow



**Step4->** Executebothfunctions.Youwillgetbelowoutpu



##### Assembly

//SPDX-License-Identifier:GPL- 3.0pragma solidity>=0.4.16<0.9.0;

contractInlineAssembly{

//Definingfunction functionadd(uint256a)publicviewreturns(uint256b){assembly{

letc:=add(a,16)msto re(0x80,c)

{

letd:=add(sload(c),12)b := d

}

b:=add(b,c)

}

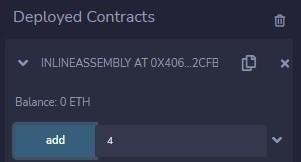
}

}



###### Flowofexecution

**Step1->** Inputanumberforaddfunction



**Step2->** Clickaddtooutputsum



##### Events

//SPDX-License- Identifier:MITpragma solidity^0.5.0;

// Creating a contractcontracteventExam ple{

//Declaringstatevariablesuint25 6publicvalue=0;

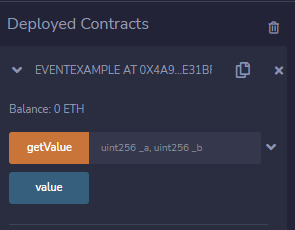
//Declaringanevent eventIncrement(addressowner);

//Definingafunctionforloggingevent functiongetValue(uint256\_a,uint256\_b)public{emitInc

rement(msg.sender); value=\_a+\_b;

}

###### Output



**Flowofexecution**

**Step1->** ProvidevaluestogetValuefunctionandclickonit.



**Step2->** Intheterminalcheckforlogs



##### ErrorHandling

contractErrorDemo{

functiongetSum(uint256a,uint256b)publicpurereturns(uint256){uint256sum=a+b;

//require(sum<255,"Invalid");assert(sum<255); returnsum;

}

}

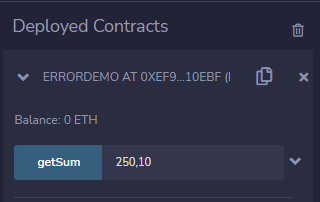
//SPDX-License-Identifier:MITpragma solidity^0.5.17;



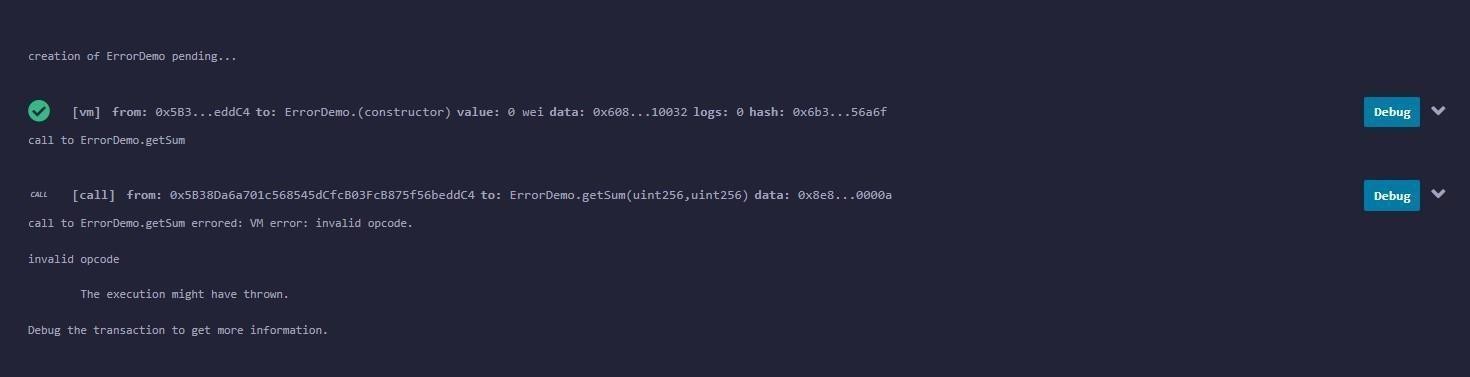
Output

###### Flowofexecution

**Step1->** ProvidesomevaluesandpressongetSum



**Step2->** Checkterminalpanel



### PRACTICAL-5

**Aim:Writeaprogramtodemonstrateminingofether**

constWeb3=require('web3');

constweb3=newWeb3(new Web3.providers.HttpProvider('http:127.0.0.1:7545'));ReplacewithyourGanacheHTTPprovider

asyncfunctionmine(){ constaccounts=awaitweb3.eth.getAccounts();constcoinbaseacc1= accounts[0];

constcoinbaseacc2=accounts[1]; console.log(`MiningetheronGanachewithcoinbaseaddress:

${coinbaseacc1}`);

while(true){try{

awaitweb3.eth.sendTransaction({from: coinbaseacc1, to:coinbaseacc2,value:

50,

});

console.log(`Minedanewblock!`);

}catch(err){console.error(err);

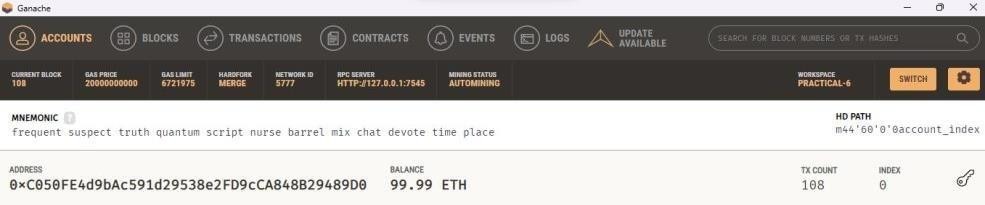
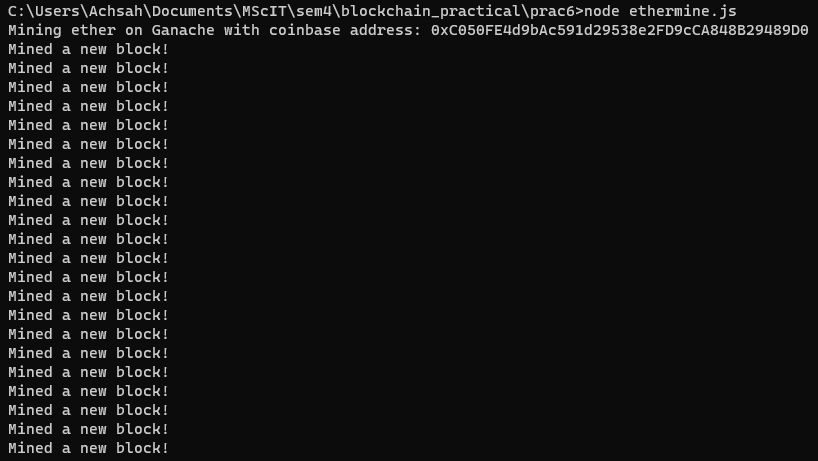
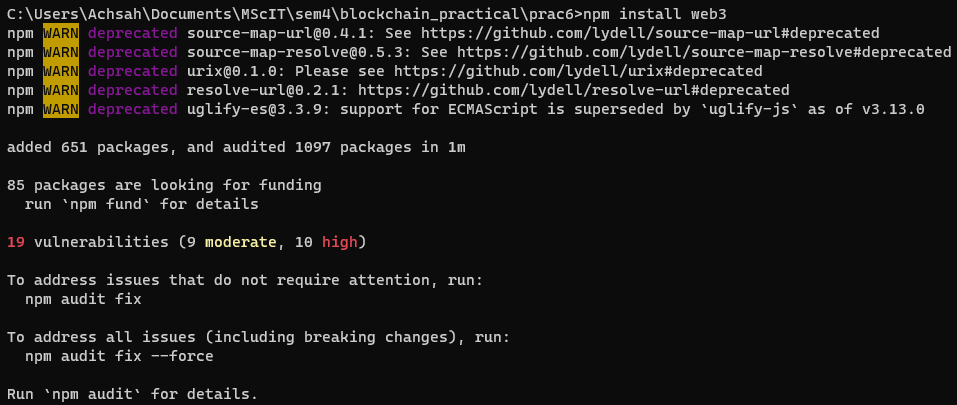
}

}

}

mine();

**Output:**



### PRACTICAL-6

**Aim:Demonstratetherunningoftheblockchainnode**

**Step1->**Create a folder named ethermine and a JSON file named genesis.jsonandwritethefollowinglines in it.

{

"config":{

"chainId":3792, "homesteadBlock":0, "eip150Block":0, "eip155Block":0, "eip158Block":0

},

"difficulty":"2000", "gasLimit":"2100000","alloc":

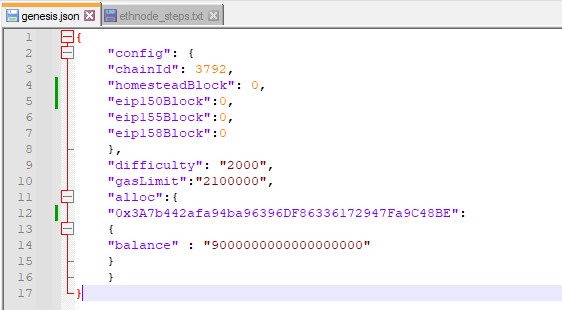
{

"0×0b6C4c81f58B8d692A7B46AD1e16a1147c25299F": {"balance": "9000000000000000000"

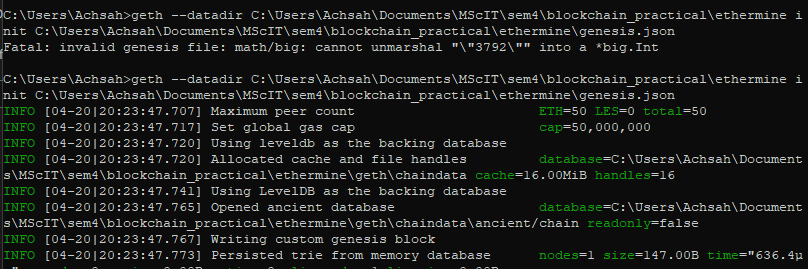
}

}

}



###### Step 2->Run command geth account new – datadirC:\Users\Achsah\Documents\MScIT\sem4\blockchain\_practical\etherminetestnet- blockchain

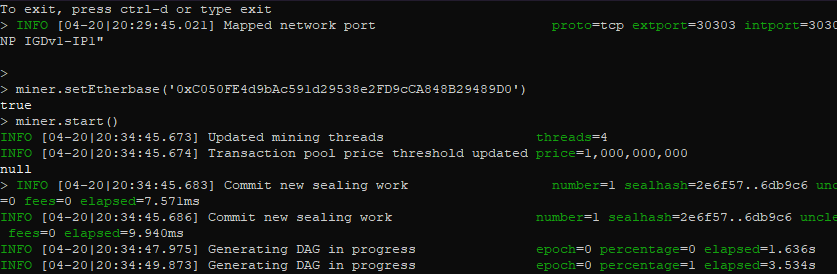
**Step 3->** Run command **geth account new -- datadirC:\Users\Achsah\Documents\MScIT\sem4\blockchain\_practical\eth ermine**

###### Step4->Runcommandgeth--identity"localB"--http--http.port"8280"--http.corsdomain"\*"-- http.api"db,eth,net,web3" –datadir "C:\Users\Achsah\Documents\MScIT\sem4\blockchain\_practical\ethermine" --port "30303" - nodiscover --networkid 5777 console. This command willenablegeth console.

**PillaiHOCCollegeofArts,ScienceandCommerce Page| 67**

**Step5->** Run the command

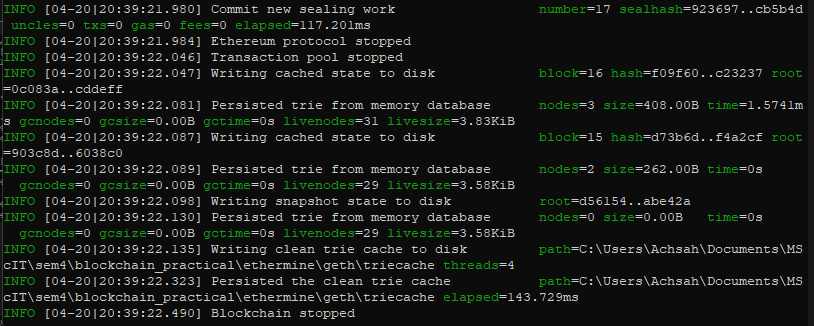
**miner.setEtherbase('0xC050FE4d9bAc591d29538e2FD9cCA848B29489D0’)**inthegethconsole **Step6->** Runthecommand**miner.start()**tostartmining



**Step7->** Belowscreenshotsaretheminingprocessesrunningonyourlocalmachine.



**Step8->** Tostoptheminingpress**Ctrl+D**



### PRACTICAL-7

**Aim:Createyourownblockchainanddemonstrateitsuse**

Createajavascriptfolderwiththefollowingcodeinanyfolderofyourchoice.

###### JavaScript Code

constSHA256=require("crypto-js/sha256");classBlock{ constructor(index,timestamp,data,previousHash=""){this.index=index;

this.timestamp = timestamp; this.data = data; this.previousHash=previousHash;this.hash= this.calculateHash();

}

calculateHash(){return SHA256(

this.index+this.previousHash+ this.timestamp + JSON.stringify(this.data)

).toString();

}

}

classBlockchain{ constructor() {

this.chain=[this.createGenesisBlock()];

}

createGenesisBlock(){ returnnewBlock(0,"21/04/2023","GenesisBlock","0");

}

getLatestBlock(){ returnthis.chain[this.chain.length-1];

}

addBlock(newBlock){ newBlock.previousHash=this.getLatestBlock().hash;

newBlock.hash=newBlock.calculateHash();this.chain.push(newBlock);

}

isChainValid(){ for(leti=1;i<this.chain.length;i+){constcurrentBlock= this.chain[i];

constpreviousBlock=this.chain[i-1];

if(currentBlock.hash currentBlock.calculateHash()){returnfalse;

}

if(currentBlock.previousHash previousBlock.hash){return false;

}

}

returntrue;

}

}

BlockchainImplementation

letmyCoin=newBlockchain(); myCoin.addBlock(newBlock(1,"22/04/2023",{amount:4}));myCoin.addBlock(newBlock(2,"22/04/2023",

{amount:8})); console.log('Isblockchainvalid?'+myCoin.isChainValid());console.log(JSON.stringify(myCoin,null, 4));

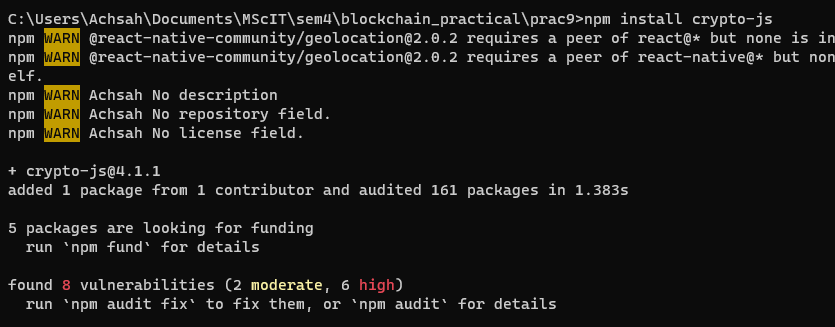
###### Flowofexecution

Output

**Step1->** Makesureyouhaveinstallednodejsinyoursystem



**Step2->**Weneed**crypto–js**nodemoduletomakeourownblockchain.Soinstallitas following



**Step3->** Runtheabovecodeincommandlineusingcommand:nodemain.js



**University of Mumbai**

**Practical Journal of**

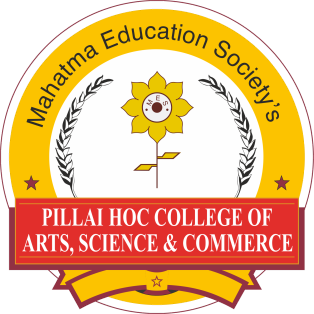
**Blockchain, Natural Language Processing & Deep Learning**

**M.Sc.(Information Technology) Part-II**

**Submitted by**

**Shaikh Obed S.A**

**Seat No: 1172743**



**DEPARTMENT OF INFORMATION TECHNOLOGY PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE, RASAYANI**

***(Affiliated to Mumbai University)* RASAYANI, 410207 MAHARASHTRA**

**2023-2024**

**Mahatma Education Society’s**

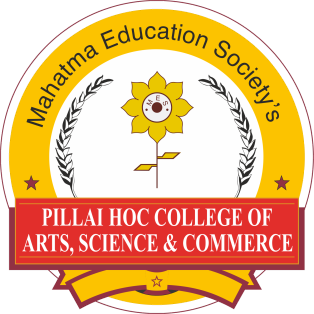
**Pillai Hoc College of Arts, Science & Commerce,**

### Rasayani

#### (Affiliated to Mumbai University)

**RASAYANI–MAHARASHTRA-410207**

### DEPARTMENT OF INFORMATIONTECHNOLOGY



**CERTIFICATE**

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Natural Language Processing** was done in computer lab of Mahatma Education Society’s Pillai HOC College of Arts, Science& Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

###### Exam Seat No: 1172743

**Subject In-Charge Coordinator**

###### External Examiner Principal

**Date: College Seal**

# Natural Language Processing

## INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Practical Name** | **Page No.** |
| 1 | 1. Install NLTK 2. Convert the given text to speech. 3. Convert audio file Speech to Text. | 1 |
| 2 | 1. Study of various Corpus–Brown, Inaugural, Reuters, udhr with various method like fields, raw, words, sents, categories. 2. Study Condition a frequency distributions 3. Studio f tagged corpora with methods like tagged sents, tagged\_words. 4. Map Words to Properties Using Python Dictionaries. | 6 |
| 3 | 1. Study of Word net Dictionary with methods assents,definitions, examples, antonyms 2. Study lemmas, hyponyms, hypernyms. 3. Write a program using python to find synonym andantonym of word "active" using Word net. | 11 |
| 4 | 1. Tokenization using Python’s split() function 2. Tokenization using Regular Expressions (Reg Ex) 3. Tokenization using NLTK 4. Tokenization using the spa cylibrary | 13 |
| 5 | Import NLPL ibraries for Indian Language sand perform word tokenizationin Hindi | 15 |
| 6 | Illustrate part of speech tagging.   1. Part of speech Tagging and chunking of user deined text. 2. Named Entity recognition of user defined text. 3. Named Entity recognition with diagram using NLTK corpus– Treebank | 16 |
| 7 | Finitestate automata   1. Define grammar using nltk. Analyze a sentence using the same. 2. Accept the input string with Regular expression of Finite Automaton: 101+. | 18 |

|  |  |  |
| --- | --- | --- |
|  | C.Accept the input string with Regular expression of FA: (a+b)\*bba. |  |
| 8 | Study Porter Stemmer, Lancaster Stemmer, Regexp Stemmer,Snowball Stemmer  Study Word Net Lemmatizer | 21 |
| 9 | Implement Naive Bayes classifier | 22 |
| 10 | 1. Speech tagging using nktl 2. B.Usage ofGiveandGaveinthePennTreebanksample 3. Probabilistic parser 4. Malt parsing: Parse a sentence and draw a tree using malt parsing. | 24 |
| 11 | 1. Multi word Expressionsin NLP 2. Normalized Web Distance and Word Similarity 3. Word Sense Disambiguation | 29 |

**1A]Install NLTK**

**PRACTICAL: 1**

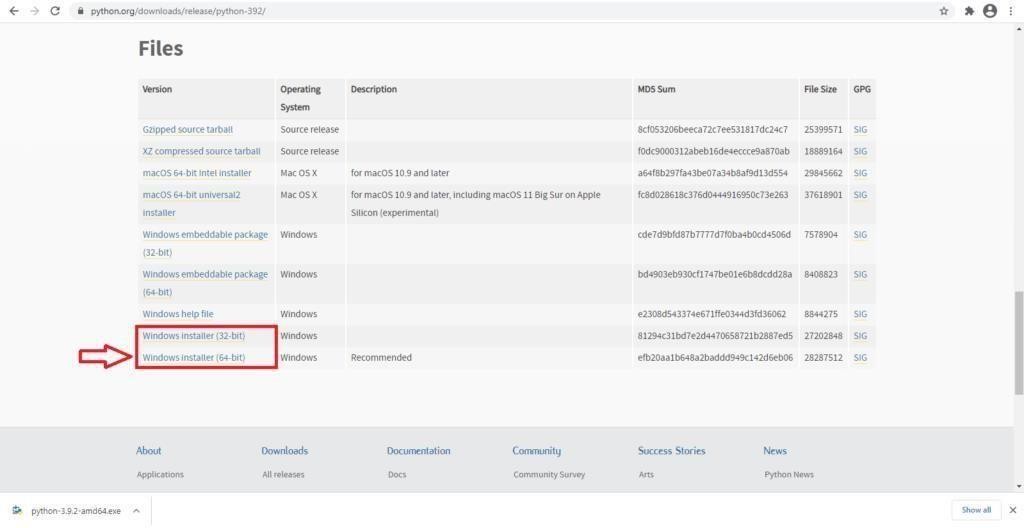
###### Python3.9.2InstallationonWindows InstallNLTK

**Python3.9.2InstallationonWindows**

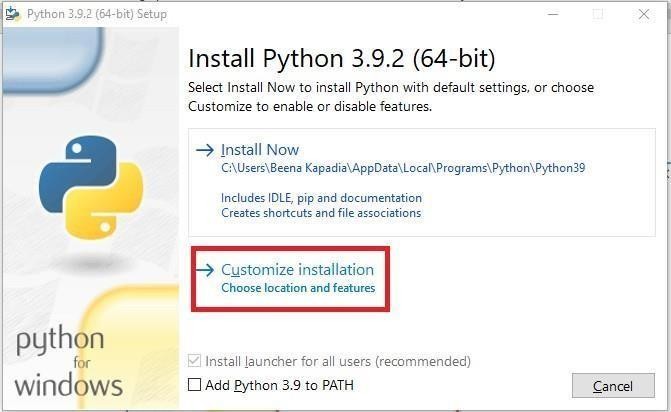
Step1)**Gotolink**https:/[/www.p](http://www.python.org/downloads/)y[thon.org/downloads/,](http://www.python.org/downloads/)**andselectthe latestversionforwindows.**



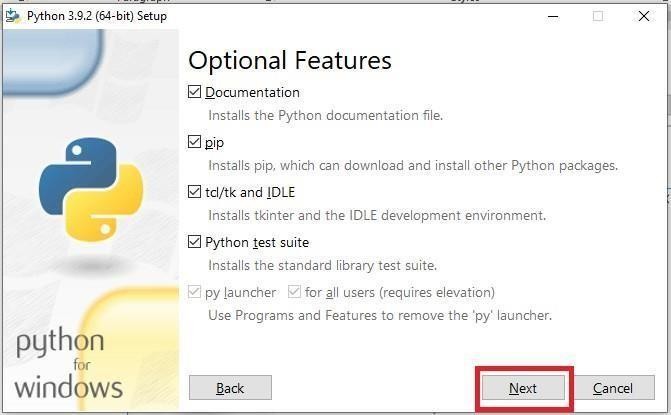
**Note**:Ifyoudon'twanttodownloadthelatestversion,youcanvisit thedownloadtab and seeallreleases.



**Step2)**ClickontheWindowsinstaller(64bit) **Step3)**SelectCustomizeInstallation

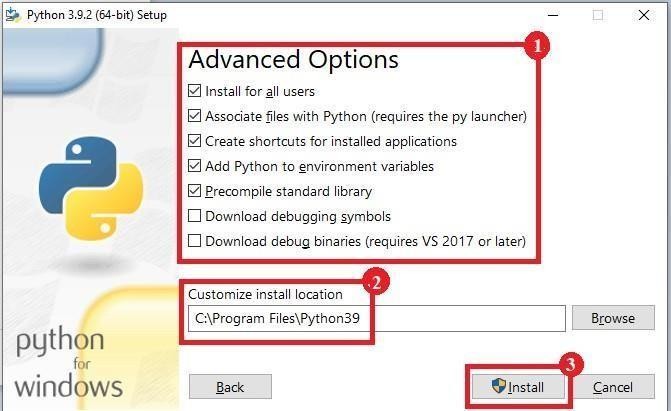


**Step4)**ClickNEXT



**Step5)** Innextscreen

1. Selecttheadvancedoptions
2. GiveaCustominstalllocation.Keepthedefaultfolderas c:\Programfiles\Python39
3. ClickInstall

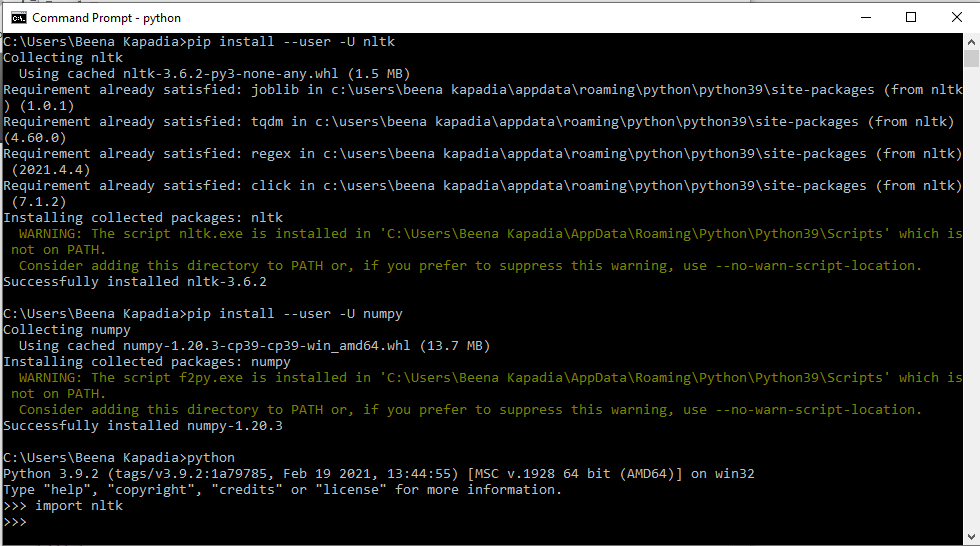


**Step6)**ClickClosebuttononceinstallisdone.

**Step7)open**commandpromptwindowandrunthefollowing commands:C:\Users\BeenaKapadia>pip install--upgradepip C:\Users\Beena Kapadia> pip install --user -U nltkC:\Users\BeenaKapadia>>pipinstall --user-U numpyC:\Users\BeenaKapadia>python

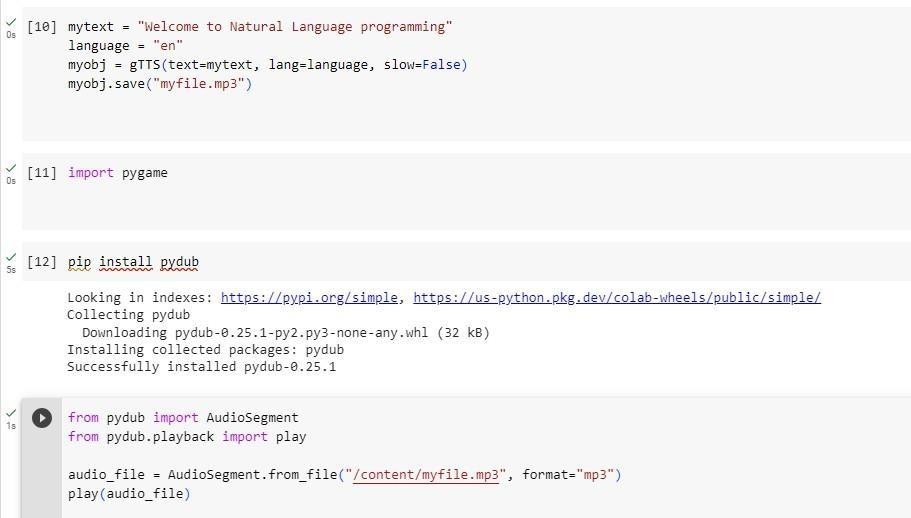
>>>importnltk

>>>



###### PillaiHOCCollegeof Arts,Science &Commerce Page |3

**1B]Convertthegiventextto speech.**



**1C]ConvertaudiofileSpeechto Text.**





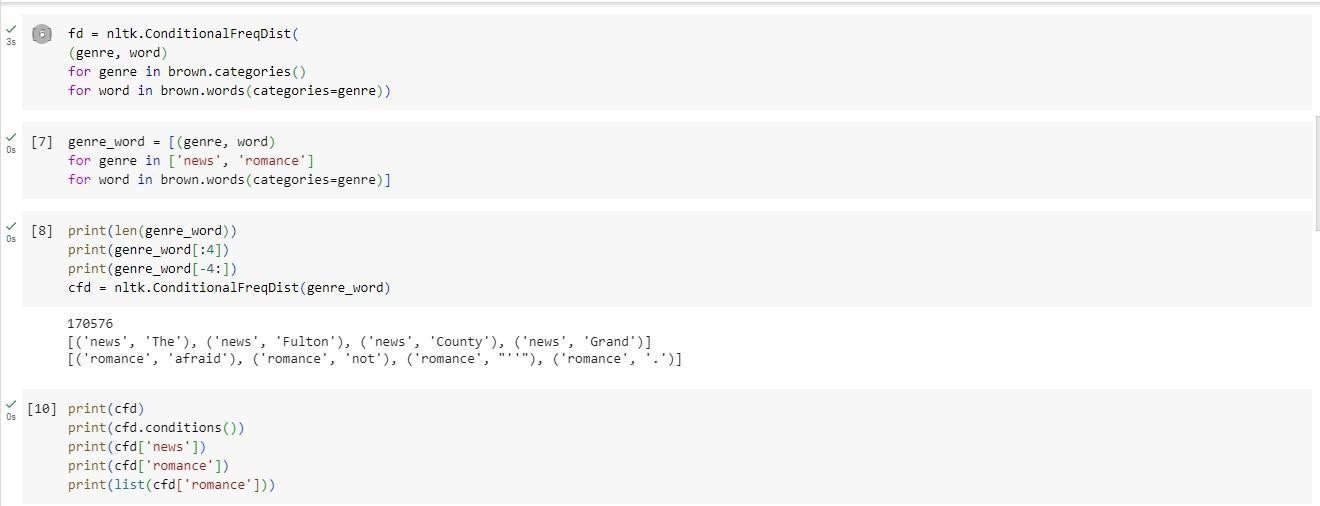
### PRACTICAL:2

###### 2A]StudyofvariousCorpus–Brown,Inaugural,Reuters,udhrwithvariousmethod likefilelds, raw,words, sents, categories.



**2C]StudyConditionalfrequency distributions**









###### 2D]Studyof taggedcorporawithmethodsliketagged\_sents,tagged\_words.



**2F]MapWordsto PropertiesUsingPython Dictionaries.**



### PRACTICAL:3

###### 3A]Studyof WordnetDictionarywith methodsassynsets,definitions,examples,antonyms



**3B]Studylemmas,hyponyms,hypernyms.**



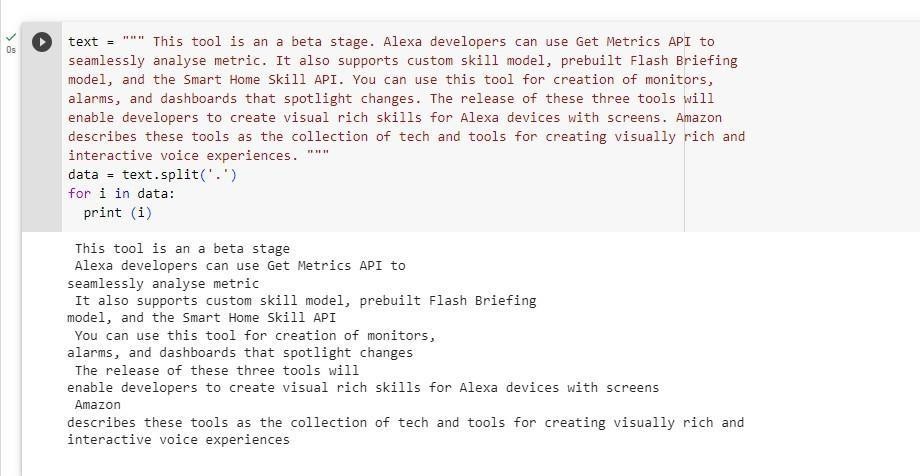


**3C]Writeaprogramusingpythontofindsynonymandantonymofword "active"usingWordnet.**



### PRACTICAL:4

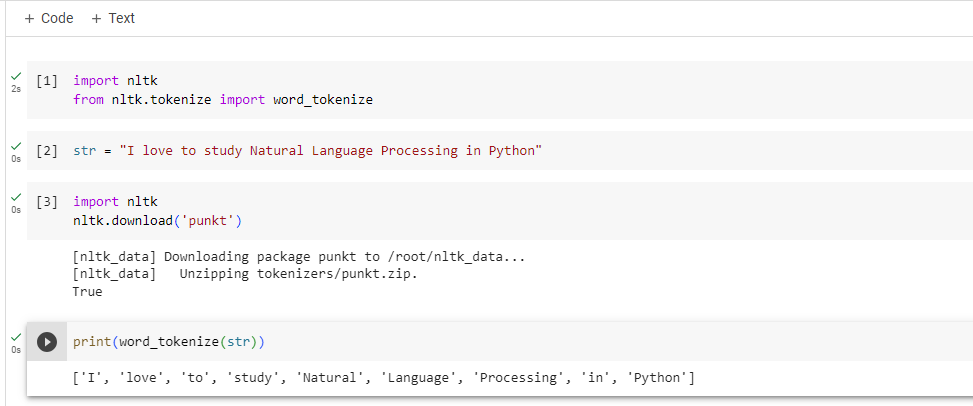
###### 4A]TokenizationusingPython’ssplit() function



**4B]TokenizationusingRegularExpressions(RegEx)**



###### 4C]Tokenization usingNLTK



**4D]Tokenizationusingthe spaCylibrary**



### PRACTICAL:5

###### ImportNLPLibrariesforIndianLanguagesandperform 5A] word tokenization in Hindi

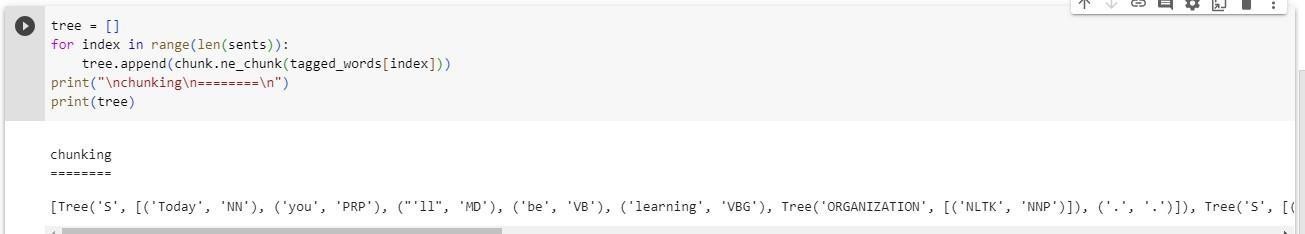
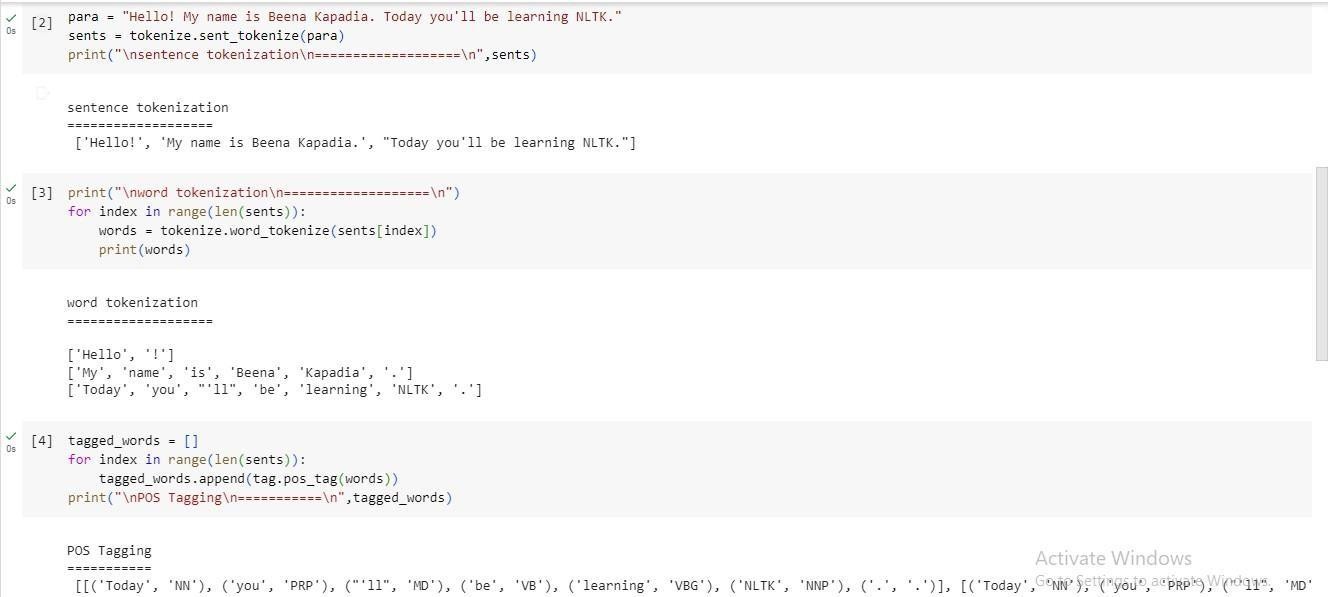


**Illustratepartofspeechtagging.**

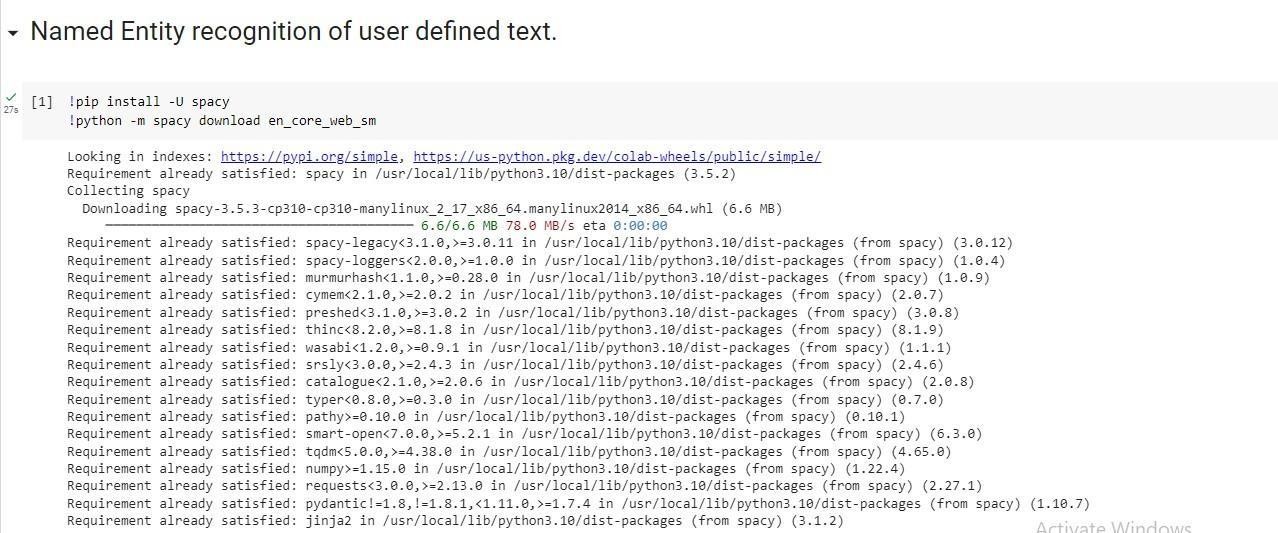
### PRACTICAL:6

###### 6A]PartofspeechTaggingandchunkingof userdefinedtext.





**6B]NamedEntityrecognitionofuserdefinedtext.**

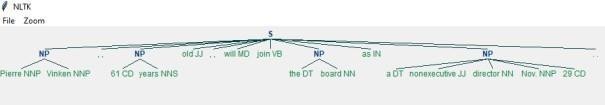


###### 6C]Named Entity recognitionwith diagramusingNLTKcorpus– Treebank

importnltknltk.download('treebank'

)

fromnltk.corpus import treebank\_chunktreebank\_chunk.tagged\_sents()[0] treebank\_chunk.chunked\_sents()[0] treebank\_chunk.chunked\_sents()[0].draw()



**Finitestate automata**

### PRACTICAL:7

###### 7A]Definegrammarusingnltk.Analyzeasentenceusingthesame.

importnltk fromnltkimporttokenize

grammar1=nltk.CFG.fromstring(""" S

-> VP

VP->VPNP NP->DetNP

Det->'that'

NP->singularNoun NP

-> 'flight'

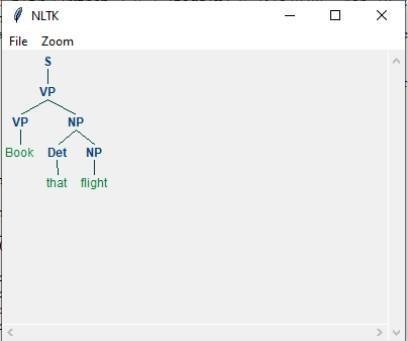
VP->'Book' """)

sentence="Bookthatflight" forindexin range(len(sentence)):

all\_tokens=tokenize.word\_tokenize(sentence) print(all\_tokens) parser=nltk.ChartParser(grammar1)

for tree in parser.parse(all\_tokens): print(tree)

tree.draw()



###### 7B]AccepttheinputstringwithRegularexpressionof Finite Automaton:101+.

defFA(s):

#ifthelengthislessthan3,itcan'tbeaccepted,soendtheprocess iflen(s) <

3:

return"Rejected"

#firstthreecharactersarefixed,checkingthemusingindex if s[0] == '1':

ifs[1]=='0':

ifs[2]=='1':

#Afterindex2,only"1"canappear,so break theprocess if anyothercharacter is

detected

fori inrange(3, len(s)):

ifs[i] !='1':

return"Rejected" return"Accepted"#ifallnestedifsaretrue return "Rejected"# else of 3rd if return"Rejected"#elseof2ndif

return "Rejected"# else of 1st if

inputs=['1','10101','101','10111','01010','100','','10111101','1011111']

fori in inputs:

print(FA(i))



###### 7C]AccepttheinputstringwithRegularexpressionof FA:(a+b)\*bba.

defFA(s):

size= 0

#scancompletestringandmakesurethatitcontainsonly'a'&'b' for i in s: ifi=='a'ori=='b': size

+= 1

else: return"Rejected"

#Aftercheckingthatitcontainsonly'a'&'b' # check its length; it should be at least 3

if size>=3:

#checkthelast3elements if s[size-3] == 'b':

ifs[size-2]=='b':

ifs[size-1]=='a': return"Accepted"#ifall3ifsaretrue return "Rejected"# else of 3rd if return "Rejected"# else of 2nd if return "Rejected"# else of 1st ifreturn "Rejected"

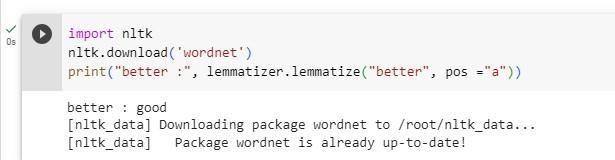
inputs=['bba','ababbba','abba','abb','baba','bbb',''] for i in inputs:

print(FA(i))



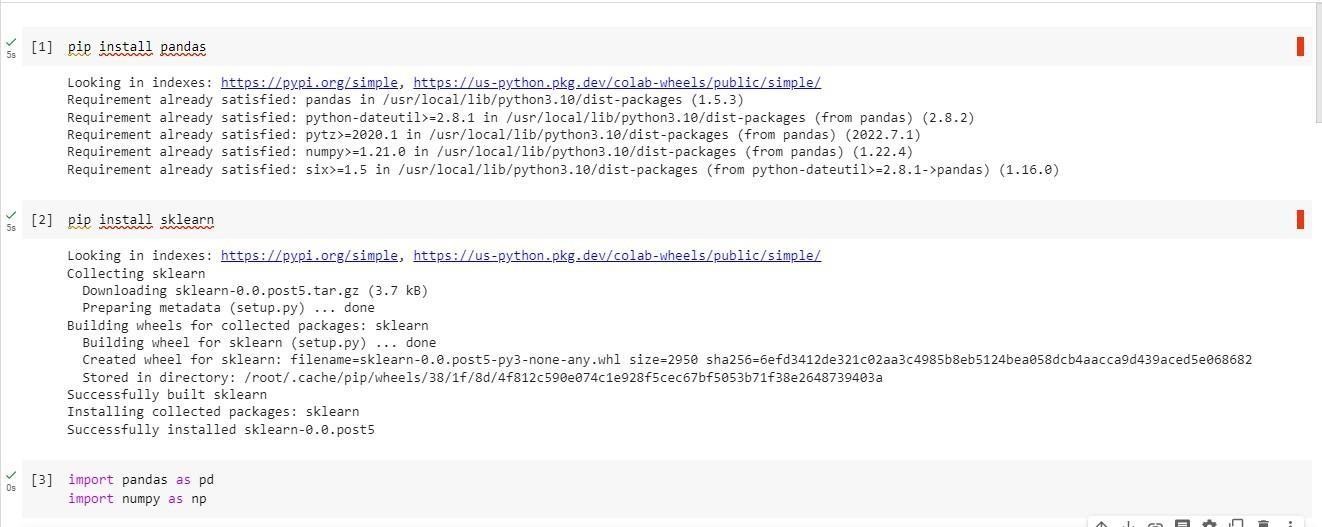
### PRACTICAL:8

###### StudyPorterStemmer,LancasterStemmer,RegexpStemmer,SnowballStemmer Study WordNetLemmatizer



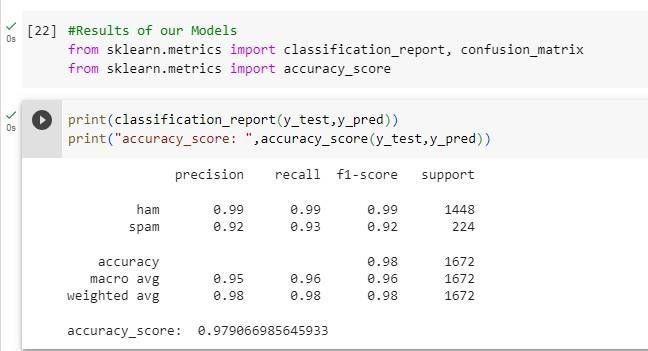
**ImplementNaiveBayes classifier**

### PRACTICAL:9



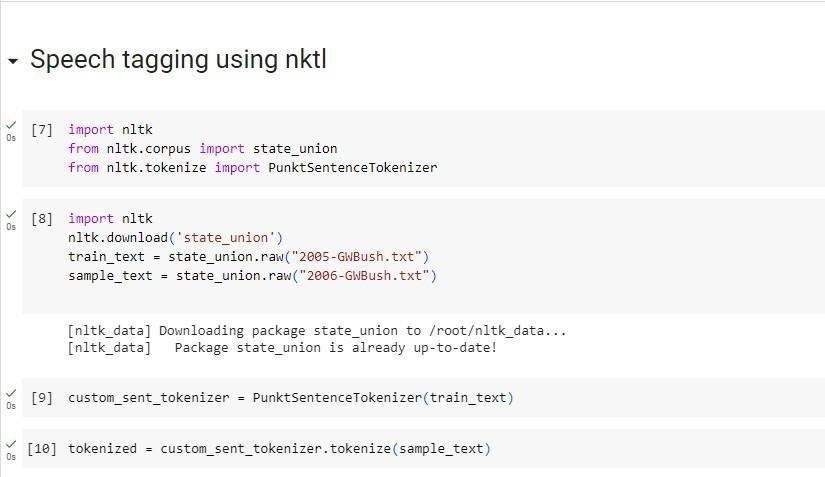


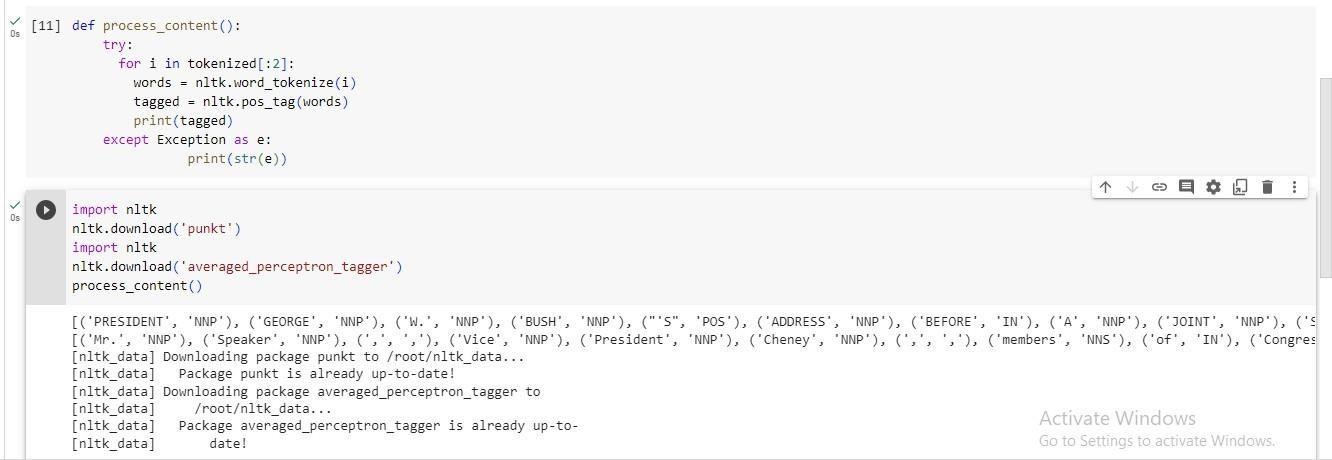




**10AII]Speech taggingusing nktl**

### PRACTICAL:10

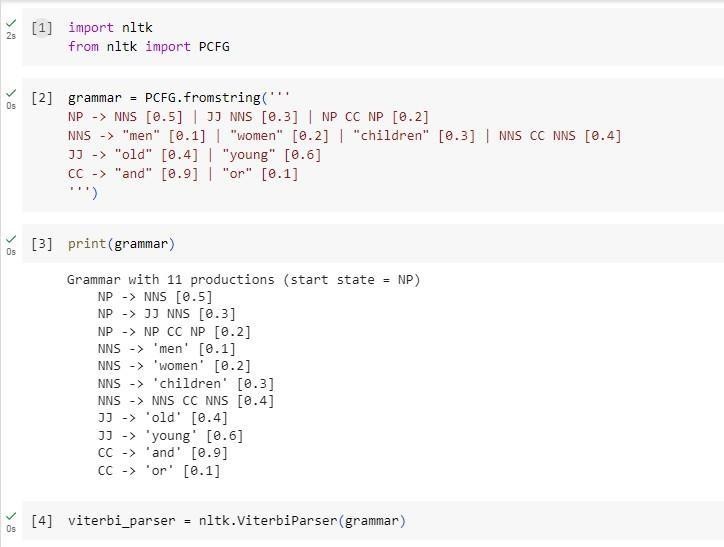




###### 10BI]Usageof Giveand Gavein the PennTreebanksample



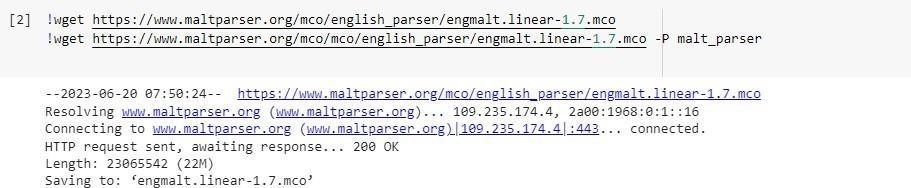
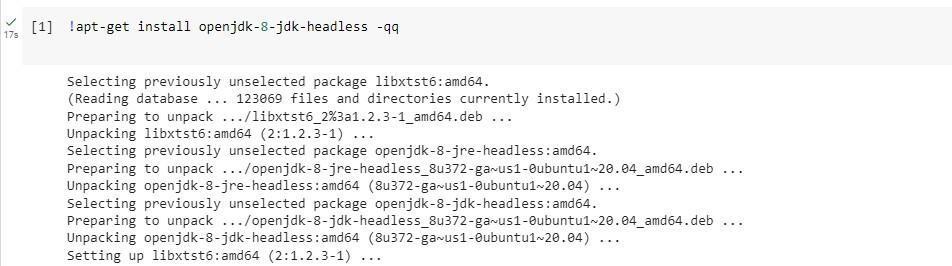
**10BII]probabilisticparser**





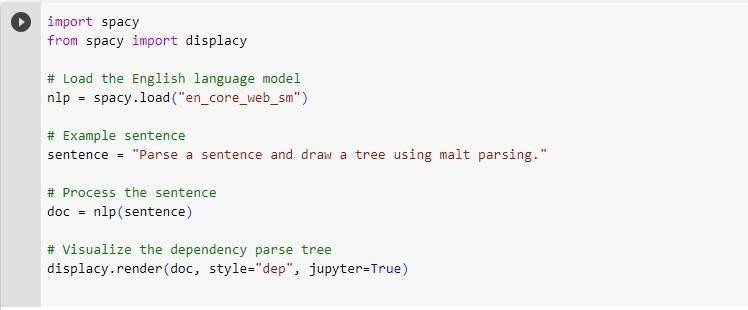
###### 10C]Malt parsing:

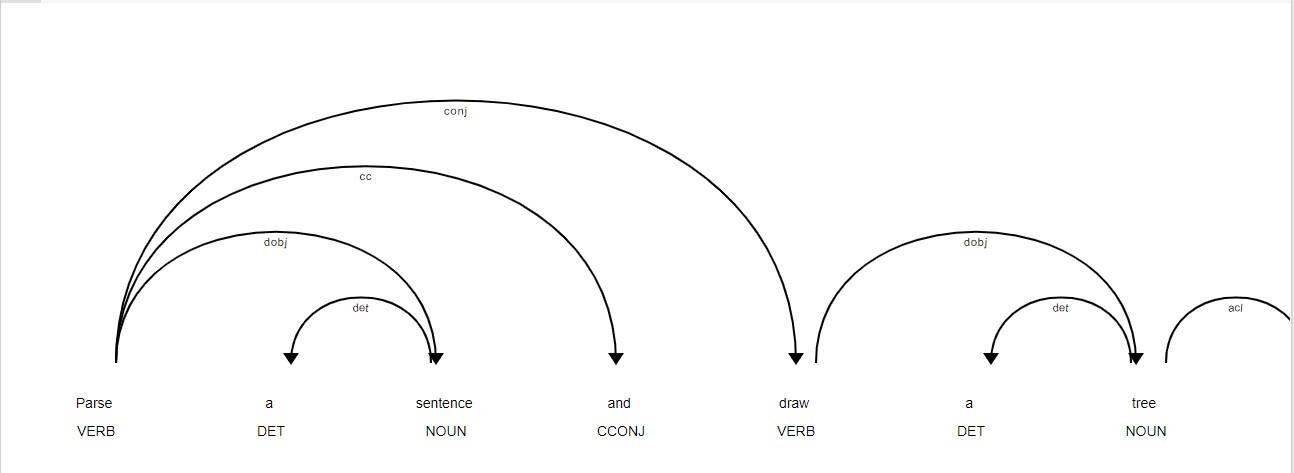
**Parseasentenceanddraw atreeusingmaltparsing.**











**11A]MultiwordExpressionsinNLP**

### PRACTICAL:11



###### 11B]NormalizedWeb DistanceandWordSimilarity

importnumpyasnp import re importtextdistanceimp ortsklearn

fromsklearn.clusterimportAgglomerativeClustering

texts =[

'Reliancesupermarket','Reliancehypermarket','Reliance','Reliance','Reliance', 'downtown', 'Reliance market', 'Mumbai','MumbaiHyper','Mumbaidxb','mumbai airport', 'k.mtrading','KMTrading','KMtrade', 'K.M.Trading', 'KM.Trading'

]

defnormalize(text):

"""Keeponlylower-casedtextandnumbers""" returnre.sub('[^a-z0-9]+', ' ', text.lower())

defgroup\_texts(texts, threshold=0.4): """Replaceeachtextwiththerepresentativeofitscluster"""

normalized\_texts = np.array([normalize(text) for text in texts]) distances = 1 - np.array([

[textdistance.jaro\_winkler(one,another)foroneinnormalized\_texts]

foranotherin normalized\_texts

])

clustering=AgglomerativeClustering( distance\_threshold=threshold, affinity="precomputed",linkage="complete",n\_clusters=None

).fit(distances) centers=dict()

forcluster\_idin set(clustering.labels\_): index=clustering.labels\_==cluster\_id

centrality = distances[:, index][index].sum(axis=1) centers[cluster\_id]=normalized\_texts[index][centrality.argmin()] return [centers[i] for i in clustering.labels\_]

print(group\_texts(texts))



**11C]WordSenseDisambiguation**



**University of Mumbai**

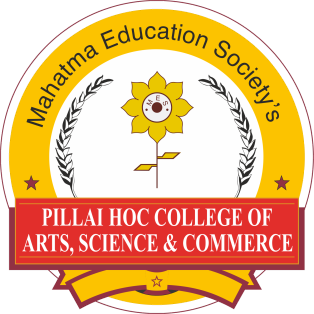
**Practical Journal of**

**Blockchain, Natural Language Processing & Deep Learning**

**M.Sc.(Information Technology) Part-II**

**Submittedby**

**Shaikh Obed S.A Seat No: 1172743**



**DEPARTMENT OF INFORMATION TECHNOLOGY PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE, RASAYANI**

***(Affiliated to Mumbai University)***

**RASAYANI, 410207 MAHARASHTRA**

2023-2024

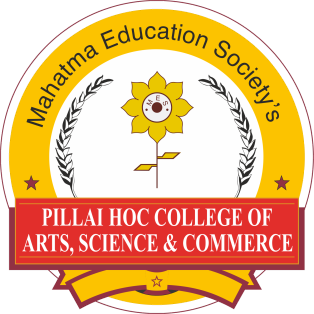
**Mahatma Education Society’s**

**Pillai Hoc College of Arts, Science & Commerce, Rasayani**

#### (Affiliated to Mumbai University)

**RASAYANI–MAHARASHTRA-410207**

### DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Deep Learning** was done in computer labof Mahatma Education Society’s Pillai HOC College of Arts, Science & Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

###### Exam Seat No: 1172743

**Subject In-Charge Coordinator**

###### External Examiner Principal

**Date: College Seal**

# Deep Learning

## INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **PracticalName** | **Page No.** |
| 1 | Write a Program to demonstrate following operations.   1. Create Vector, Matrix and Tensor 2. Multiplication of two:Vector, Matrix and Tensor 3. Addition of two:Vector,Matrix and Tensor 4. Multiply Matrix with Vector 5. Matrix Dot product and Matrix Inverse | 1 |
| 2 | Performing matrix multiplication and finding Eigen vectors andEigen values using Tensor Flow | 5 |
| 3 | Implementing deep neural network for performing binary classification task. | 6 |
| 4 | Using deep feed forward network with two hidden layers for performing classification and predicting the probability of class. | 7 |
| 5 | Evaluating feed for ward deep network for multiclass Classification using K-Fold cross-validation. | 9 |
| 6 | Implementation of convolution a neural network to predict numbers from number images. | 10 |
| 7 | Performing encoding and decoding of images using deep Auto encoder. | 12 |
| 8 | Denoising of images using Autoencoder. | 14 |

### PracticalNo:1 Aim:WriteaProgramtodemonstratefollowingoperations.

1. CreateVector,MatrixandTensor
2. Multiplicationoftwo:Vector,MatrixandTensor
3. Additionoftwo:Vector,MatrixandTensor
4. MultiplyMatrixwithVector
5. MatrixDotproductandMatrixInverse

### CreateVector,MatrixandTensor Code:

importnumpy as np importtensorflowastf

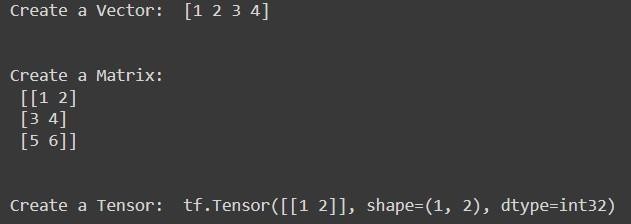
x=np.array([1,2,3,4]) print("CreateaVector:",x) print("\n")

A=np.array([[1,2],[3,4],[5,6]])

print("CreateaMatrix:\n",A) print("\n")

tensor\_A=tf.constant([[1,2]],dtype=tf.int32) print("Create a Tensor: ",tensor\_A) print("\n")

### Output:



1. **Multiplicationoftwo:Vector,MatrixandTensor Code:**

A=np.array([[1,2],[3,4],[5,6]])

print("A=",A)

print("\n")

B=np.array([[2,5],[7,4],[4,3]])

print("B=",B)

print("\n")

C=A\*B

print("MultiplicationoftwoMatrix:\n",C) print("\n")

x=np.array([1,2,3,4])

y=np.array([5,6,7,8])

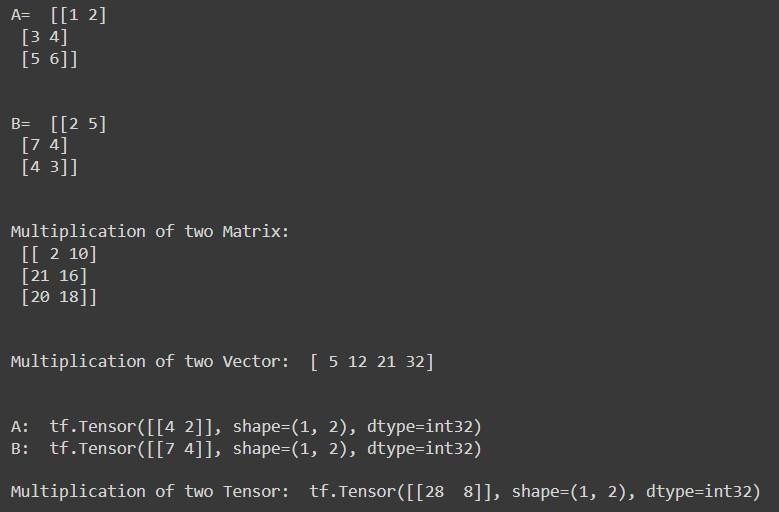
z=x\*y print("MultiplicationoftwoVector:",z) print("\n")

tensor\_A=tf.constant([[4,2]],dtype=tf.int32) print("A: ",tensor\_A)

tensor\_B=tf.constant([[7,4]],dtype=tf.int32) print("B: ",tensor\_B,"\n")

tensor\_multiply=tf.multiply(tensor\_A,tensor\_B) print("MultiplicationoftwoTensor:",tensor\_multiply) print("\n")

### Output:



1. **Additionoftwo:Vector,MatrixandTensor Code:**

x=np.array([1,2,3,4])

y=np.array([5,6,7,8])

z=x+y print("AdditionoftwoMatrix:",z) print("\n")

A=np.array([[1,2],[3,4],[5,6]])

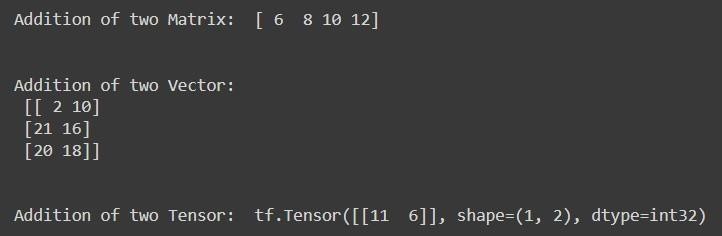
B=np.array([[2,5],[7,4],[4,3]])

C=A\*B

print("AdditionoftwoVector:\n",C) print("\n")

tensor\_add=tf.add(tensor\_A,tensor\_B) print("AdditionoftwoTensor:",tensor\_add) print("\n")

### Output:

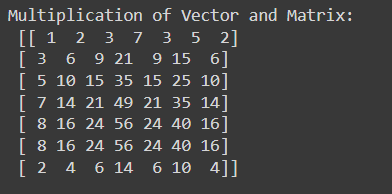


1. **MultiplyMatrixwithVector Code:**

x=np.array([1,2,3,7,3,5,2]) y=np.array([[1],[3],[5],[7],[8],[8],[2]])

c=x\*y print("MultiplicationofVectorandMatrix:\n",c) print("\n")

### Output:



1. **MatrixDotproductandMatrixInverse Code:**

U=[2,-3]

V=[1,3]

dotproduct=np.dot(U,V) print("Matrixdotproduct:",dotproduct) print("\n")

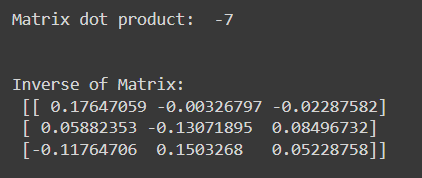
A=np.array([[6,1,1],

[4,-2,5],

[2,8,7]])

print("InverseofMatrix:\n",np.linalg.inv(A)) print("\n")

### Output:



**PracticalNo:2**

### Aim:PerformingmatrixmultiplicationandfindingEigenvectorsand Eigen values using TensorFlow

**Code:**

importtensorflowastf

x=tf.constant([1,2,3,4,5,6],shape=[2,3]) print(x) y=tf.constant([7,8,9,10,11,12],shape=[3,2]) print(y)

z=tf.matmul(x,y) print("\n")

print("Multiplyingthematrices:") print(z)

print("\n")

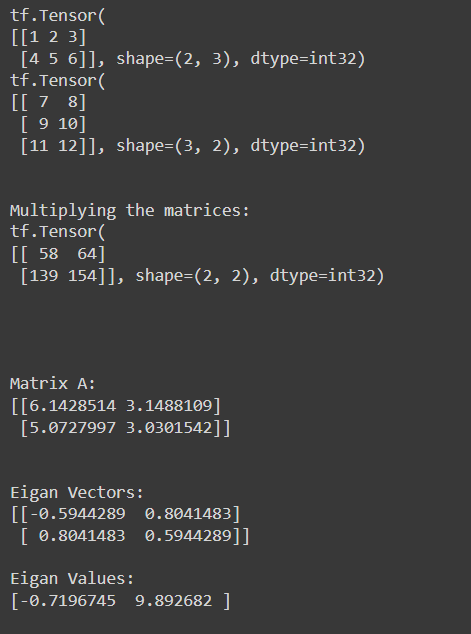
e\_matrix\_A=tf.random.uniform([2,2],minval=3,maxval=10,dtype=tf.float32,name="matrixA ")

print("\n") print("MatrixA:\n{}\n\n".format(e\_matrix\_A))

eigan\_values\_A,eigan\_vectors\_A=tf.linalg.eigh(e\_matrix\_A) print("Eigan Vectors: \n{}\n\nEigan Values:

\n{}\n\n".format(eigan\_vectors\_A,eigan\_values\_A))

### Output:



**PracticalNo:3**

### Aim:Implementingdeepneuralnetworkforperformingbinary classification task.

**Code:**

pip install scikeras frompandasimportread\_csv fromkeras.modelsimportSequential from keras.layers import Dense

fromscikeras.wrappersimportKerasClassifier fromsklearn.model\_selectionimportcross\_val\_score from sklearn.preprocessing import LabelEncoder fromsklearn.model\_selectionimportStratifiedKFoldd ataframe=read\_csv('sonar.csv',header=None) dataset=dataframe.values

X=dataset[:,0:60].astype(float) Y=dataset[:,60]

encoder=LabelEncoder() encoder.fit(Y) encoded\_y=encoder.transform(Y)

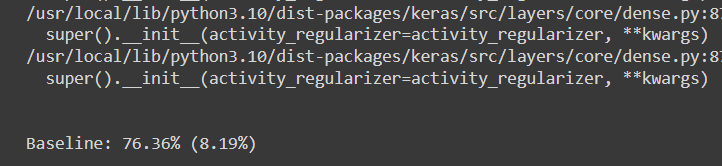
defcreate\_baseline(): model=Sequential()

model.add(Dense(60,input\_dim=60,activation='relu')) model.add(Dense(1,activation='sigmoid')) model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy']) return model

estimator=KerasClassifier(model=create\_baseline,epochs=10,batch\_size=5,verbose=0) KFold=StratifiedKFold(n\_splits=10,shuffle=True) results=cross\_val\_score(estimator,X,encoded\_y,cv=KFold)

print("\n") print("Baseline:%.2f%%(%.2f%%)"%(results.mean()\*100,results.std()\*100))

### Output:



**PracticalNo:4**

### Aim:Usingdeepfeedforwardnetworkwithtwohiddenlayersfor performing classification and predicting the probability of class.

**Code:**

importnumpy asnp fromsklearn.datasetsimportload\_wine

fromsklearn.preprocessingimportMinMaxScaler,OneHotEncoder from keras.layers import Dense, Input, concatenate, Dropout fromkeras.modelsimportModel fromtensorflow.kerasimportoptimizers

optimizers.RMSpropopti mizers.Adam

dataset=load\_wine() ensemble\_num = 10 bootstrap\_size=0.8 training\_size=0.8

num\_hidden\_neurosn=10 dropout= 0.25

epochs=100 batch=10 temp

= []

scaler = MinMaxScaler() one\_hot=OneHotEncoder() dataset['data']=scaler.fit\_transform(dataset['data'])

dataset['target']=one\_hot.fit\_transform(np.reshape(dataset['target'],(-1,1))).toarray() for i in range(len(dataset.data)):

temp.append([dataset['data'][i],np.array(dataset['target'][i])])

temp=np.array(temp,dtype=object) np.random.shuffle(temp) #holdouttrainingand teststop index

stop = int(training\_size\*len(dataset.data))

train\_X=np.array([xforxintemp[:stop,0]]) train\_Y=np.array([xforxintemp[:stop,1]]) test\_X = np.array([x for x in temp[stop:,0]]) test\_Y = np.array([x for x in temp[stop:,1]])

num\_hidden\_neurons=64

sub\_net\_outputs=[] sub\_net\_inputs = []

fori in range(ensemble\_num): #twohidden layersto keepit simple

#specifyinputshapetotheshapeofthetrainingset net\_input = Input(shape=(train\_X.shape[1],)) sub\_net\_inputs.append(net\_input) Y=Dense(num\_hidden\_neurons)(net\_input) Y

= Dense(num\_hidden\_neurons)(Y) Y=Dropout(dropout)(Y)

#sub\_netscontainstheoutputtensors sub\_net\_outputs.append(Y)

Y=concatenate(sub\_net\_outputs) Y=Dense(train\_Y[0].shape[0],activation='softmax')(Y) model

= Model(inputs=sub\_net\_inputs, outputs=Y) model.compile(optimizer='rmsprop',loss='categorical\_crossentropy') print('\n')

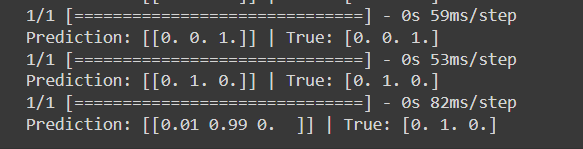
print("7\_AdityaHadap") print("Begintraining...")

model.fit([train\_X]\*ensemble\_num,train\_Y,validation\_data=[[test\_X]\*ensemble\_num, test\_Y], epochs=epochs, batch\_size=batch)

print("Training complete...") np.set\_printoptions(precision=2,suppress=True) for i in range(len(test\_X)):

print("Prediction:"+str(model.predict([test\_X[i].reshape(1,test\_X[i].shape[0])]\* ensemble\_num)) + " | True: " + str(test\_Y[i]))

### Output:



**PracticalNo:5**

### Aim:EvaluatingfeedforwarddeepnetworkformulticlassClassification using K- Fold cross-validation.

**Code:**

!pipinstallscikeras

!pipinstallnp\_utils

import pandas

# importnp\_utils fromkeras.modelsimportSequential from keras.layers import Dense

fromscikeras.wrappers import KerasClassifier fromtensorflow.keras.utilsimportto\_categorical # from scikeras import np\_utils fromsklearn.model\_selectionimportcross\_val\_score from sklearn.model\_selection import KFold fromsklearn.preprocessingimportLabelEncoder from sklearn.pipeline import Pipeline

dataframe=pandas.read\_csv("iris.data",header=None) dataset=dataframe.values X=dataset[:,0:4].astype(float)

Y=dataset[:,4]

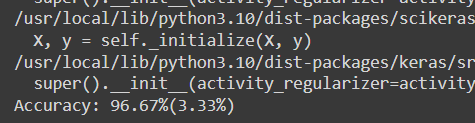
encoder=LabelEncoder()encoder.fit(Y) encoded\_Y=encoder.transform(Y) dummy\_y=to\_categorical(encoded\_Y)

defbaseline\_model(): model=Sequential()

model.add(Dense(8,input\_dim=4,activation='relu')) model.add(Dense(3,activation='softmax')) model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy']) return model

estimator=KerasClassifier(build\_fn=baseline\_model,epochs=200,batch\_size=5,verbose=0) kfold=KFold(n\_splits=10,shuffle=True) results=cross\_val\_score(estimator,X,dummy\_y,cv=kfold) print("Accuracy:%.2f%%(%.2f%%)"%(results.mean()\*100,results.std()\*100))

### Output:



**PracticalNo:6**

### Aim:Implementationofconvolutionalneuralnetworktopredictnumbers from number images.

**Code:**

fromkeras.datasetsimportmnist fromtensorflow.keras.utilsimportto\_categorical from keras.models import Sequential

fromkeras.layersimportDense,Conv2D,Flatten import matplotlib.pyplot as plt

importnumpy asnp (X\_train,Y\_train),(X\_test,Y\_test)= mnist.load\_data()

plt.imshow(X\_train[0]) plt.show

print(X\_train[0].shape)

X\_train=X\_train.reshape(60000,28,28,1) X\_test=X\_test.reshape(10000,28,28,1)

Y\_train=to\_categorical(Y\_train) Y\_test=to\_categorical(Y\_test) Y\_train[0]

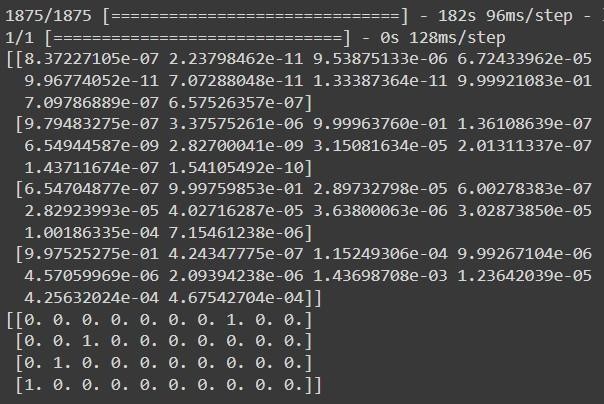
print(Y\_train[0])

model=Sequential() model.add(Conv2D(64,kernel\_size=3,activation='relu',input\_shape=(28,28,1))) model.add(Conv2D(32, kernel\_size=3, activation='relu')) model.add(Flatten())

model.add(Dense(10,activation='softmax')) model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy']) model.fit(X\_train, Y\_train, validation\_data=(X\_test, Y\_test), epochs=1)

print(model.predict(X\_test[:4])) print(Y\_test[:4])

### Output:



**PracticalNo:7**

### Aim:Performingencodinganddecodingofimagesusingdeepautoencoder. Code:

importkeras fromkerasimportlayers

fromkeras.datasetsimportmnist import numpy as np importmatplotlib.pyplotaspltencoding

\_dim=32 input\_img=keras.Input(shape=(784,))

encoded=layers.Dense(encoding\_dim,activation='relu')(input\_img) decoded=layers.Dense(784,activation='sigmoid')(encoded) autoencoder=keras.Model(input\_img,decoded)

encoder=keras.Model(input\_img,encoded) encoded\_input=keras.Input(shape=(encoding\_dim,)) decoder\_layer=autoencoder.layers[-1] decoder=keras.Model(encoded\_input,decoder\_layer(encoded\_input))

autoencoder.compile(optimizer='adam',loss='binary\_crossentropy')

(x\_train,\_),(x\_test,\_)=mnist.load\_data() x\_train=x\_train.astype('float32')/255. x\_test=x\_test.astype('float32')/25 x\_train=x\_train.reshape((len(x\_train),np.prod(x\_train.shape[1:]))) x\_test=x\_test.reshape((len(x\_test),np.prod(x\_test.shape[1:]))) print(x\_train.shape)

print(x\_test.shape)

autoencoder.fit(x\_train,x\_train,epochs=13,batch\_size=256,shuffle=True,validation\_data=(x\_t est,x\_test))

encoded\_imgs=encoder.predict(x\_test) decoded\_imgs=decoder.predict(encoded\_imgs)

n=10

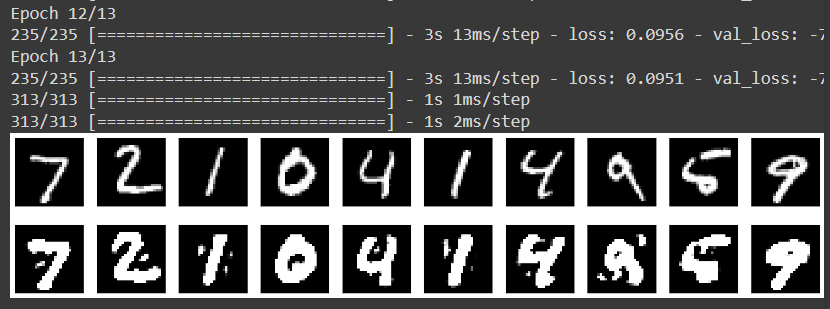
plt.figure(figsize=(20,4)) for i in range(n): ax=plt.subplot(2,n,i+1)

plt.imshow(x\_test[i].reshape(28,28)) plt.gray() ax.get\_xaxis().set\_visible(False) ax.get\_yaxis().set\_visible(False)

ax=plt.subplot(2,n,i+1+n) plt.imshow(decoded\_imgs[i].reshape(28,28)) plt.gray()

ax.get\_xaxis().set\_visible(False) ax.get\_yaxis().set\_visible(False) plt.show()

### Output:



**PracticalNo:8** **Aim:DenoisingofimagesusingAutoencoder.**

### Code:

importkeras fromkeras.datasetsimportmnist from keras import layers importnumpy asnp

fromkeras.callbacksimportTensorBoard import matplotlib.pyplot as plt

(X\_train,\_),(X\_test,\_)=mnist.load\_data()

X\_train = X\_train.astype('float32') / 255. X\_test= X\_test.astype('float32')/ 255.

X\_train=np.reshape(X\_train,(len(X\_train),28,28,1)) X\_test=np.reshape(X\_test,(len(X\_test),28,28,1))

noise\_factor=0.5

X\_train\_noisy = X\_train + noise\_factor \* \ np.random.normal(loc=0.0,scale=1.0,size=X\_train.shape)

X\_test\_noisy = X\_test + noise\_factor \* \ np.random.normal(loc=0.0,scale=1.0,size=X\_test.shape)

X\_train\_noisy=np.clip(X\_train\_noisy,0., 1.)

X\_test\_noisy=np.clip(X\_test\_noisy,0., 1.)

n=10

plt.figure(figsize=(20,2)) for i in range(1,n+1):

ax=plt.subplot(1,n,i) plt.imshow(X\_test\_noisy[i].reshape(28,28)) plt.gray()

ax.get\_xaxis().set\_visible(False) ax.get\_yaxis().set\_visible(False)

plt.show() input\_img=keras.Input(shape=(28,28,1))

x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(input\_img) x=layers.MaxPooling2D((2,2),padding='same')(x) x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(x) encoded=layers.MaxPooling2D((2,2),padding='same')(x)

x= layers.Conv2D(32,(3,3),activation='relu',padding='same')(encoded) x=layers.UpSampling2D((2,2))(x) x=layers.Conv2D(32,(3,3),activation='relu',padding='same')(x) x=layers.UpSampling2D((2,2))(x) decoded=layers.Conv2D(1,(3,3),activation='sigmoid',padding='same')(x)

autoencoder=keras.Model(input\_img,decoded) autoencoder.compile(optimizer='adam',loss='binary\_crossentropy') autoencoder.fit(X\_train\_noisy,X\_train,

epochs=3, batch\_size=328, shuffle=True,

validation\_data=(X\_test\_noisy,X\_test), callbacks=[TensorBoard(log\_dir='/tmp/tb',histogram\_freq=0,write\_graph=False)])

predictions=autoencoder.predict(X\_test\_noisy)

m=10

plt.figure(figsize=(20,2)) for i in range(1,m+1):

ax=plt.subplot(1,m,i) plt.imshow(predictions[i].reshape(28,28)) plt.gray()ax.get\_xaxis().set\_visible(False) ax.get\_yaxis().set\_visible(False)

plt.show()

### Output:

