**Mini Project Report on**

**WEB 3.0 website**

**(BlockChain Money Transfer)**

Submitted in partial fulfilment of the required for the award of the degree of

Bachelors of Technology in Computer Science & Engineering



Graphic Era Hill University

(Department of Computer Science & Engineering)

Dehradun Campus , Uttrakhand

**Submitted by: Submitted To:**

**Name: Shreyansh Shukla Mr. Aminesh Srivastava**

**University Roll No: 2119210 (Class Coordinator Sec L)**



The success and final outcome of this min project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my mini project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mr. AMINESH SRIVASTAVA , for providing me an opportunity to do the project work in ”Text Recognition Android app ” and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Dept of environment which helped us in successfully completing our mini project work.

Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Shreyansh Shukla

Date: 07/01/2024

**Table of Content**

**S.No Content Page No.**

1. Executive Summary 1

2. How does Optical Character Recognition Work? 2

3. Use Cases for OCR Technology 4

4. Limitation of the Project. 6

5. Code for Capture Image button 8

6. Screenshots of the Project 14

7. Conclusion 15

8. Reference 16

**Executive Summary**

In the executive summary, briefly introduce the project is needed and its main goals. Highlighting the problem that the blockchain money transfer system aims to address. Summarizing the significance of implementing a blockchain solution for financial transactions. Outlining the key achievements and outcomes of the project. Keep this section concise and informative, providing a high-level overview.

The Blockchain Money Transfer project addresses the need for a secure and decentralized system for financial transactions. Traditional financial systems face challenges such as security vulnerabilities and centralized control, which this project seeks to overcome. The implementation of a blockchain-based money transfer system ensures transparency, immutability, and increased security in financial transactions. Throughout the development process, key achievements include the successful implementation of a proof-of-work consensus mechanism, creation of a user-friendly front-end interface, and the establishment of a secure, tamper-resistant blockchain.

In this project we used the special ability of blockchain i.e, to transfer money seamlessly and with being more secure at the same time. Each transactions with source to destination and amount everything is saved on blockchain forever. In case anyone tries to change these data on blockchain then because of the special ability of blockchain it can be identified which block of data was change. Hence these block chain transactions are more secure than any other transactions.

With executing the code of backend on python the backend of this process is executed then the output gives the address of the site to then after clicking on that address the front end of this project is created.

This is the work in least or basic possible way now if we implement it in professional way or mass implementation like shifting all money transactions on blockchain hence all the transactions on the blockchain will be seamless as well as more secure than traditional system. Also these data is always stored in blockchain which cannot be altered by anyone once it is on blockchain.

## Introduction

Blockchain technology is an advanced database mechanism that allows transparent information sharing within a business network. A blockchain database stores data in blocks that are linked together in a chain. The data is chronologically consistent because you cannot delete or modify the chain without consensus from the network. As a result, you can use blockchain technology to create an unalterable or immutable ledger for tracking orders, payments, accounts, and other transactions. The system has built-in mechanisms that prevent unauthorized transaction entries and create consistency in the shared view of these transactions.

The only question arise why blockchain, why is there need for blockchain isn’t there more better way or isn’t current system not good enough so the answer to both question in current scenarios is NO. Afcourse there might be development in future but currently blockchain is our only option.

Traditional database technologies present several challenges for recording financial transactions. For instance, consider the sale of a property. Once the money is exchanged, ownership of the property is transferred to the buyer. Individually, both the buyer and the seller can record the monetary transactions, but neither source can be trusted. The seller can easily claim they have not received the money even though they have, and the buyer can equally argue that they have paid the money even if they haven’t.

To avoid potential legal issues, a trusted third party has to supervise and validate transactions. The presence of this central authority not only complicates the transaction but also creates a single point of vulnerability. If the central database was compromised, both parties could suffer.

Blockchain mitigates such issues by creating a decentralized, tamper-proof system to record transactions. In the property transaction scenario, blockchain creates one ledger each for the buyer and the seller. All transactions must be approved by both parties and are automatically updated in both of their ledgers in real time. Any corruption in historical transactions will corrupt the entire ledger. These properties of blockchain technology have led to its use in various sectors, including the creation of digital currency like Bitcoin.

## System Architecture

Providing an overview of the system architecture, explaining the key components and how they interact. Discussing the roles of blockchain, nodes, wallets, and the consensus mechanism. Using diagrams to visually represent the flow of transactions in the system.

Overview:

The system architecture of the Blockchain Money Transfer project is designed to facilitate secure and decentralized financial transactions. Key components include the blockchain, nodes, wallets, and a proof-of-work consensus mechanism. The interaction between the backend (Flask server) and frontend (HTML/CSS) components ensures a seamless user experience.

Blockchain:

The blockchain serves as the core component, consisting of interconnected blocks that store transaction data. Each block contains a unique hash, a timestamp, and a reference to the previous block, ensuring the immutability and integrity of the transaction history.

Nodes:

Nodes represent participants in the blockchain network. Each node has a copy of the entire blockchain, and they communicate to reach consensus on the validity of new blocks. This decentralized nature enhances the security and resilience of the system.

Wallets:

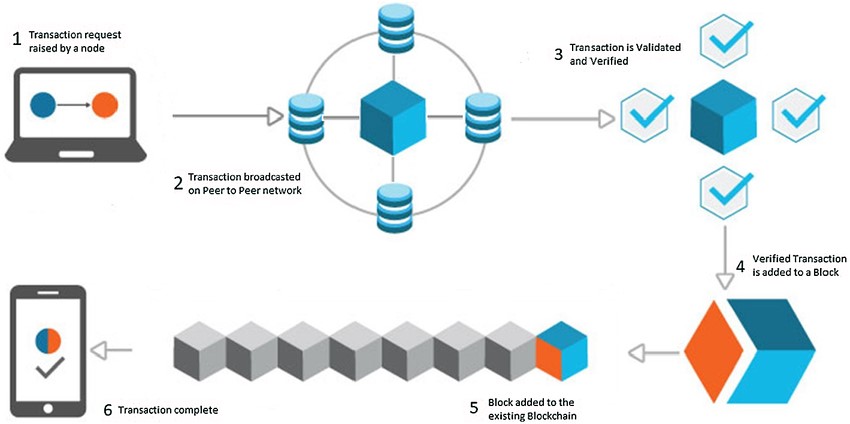
Wallets are created for users to initiate transactions. Each wallet has a unique address, and security measures are implemented to prevent unauthorized access and ensure a secure transfer of funds.

Consensus Mechanism:

A proof-of-work consensus mechanism is employed to validate and add new blocks to the blockchain. This mechanism requires nodes to solve a computationally intensive problem, ensuring that the network reaches consensus on the order of transactions.

Interaction Flow:

The interaction flow involves users interacting with the frontend interface to initiate transactions. The Flask server processes these transactions, validates them through the proof-of-work mechanism, and adds the verified transactions to the blockchain. The updated blockchain is then reflected in the frontend interface.



In the above diagram it is shown the flow of transaction, here first the transaction request raised by a node in simple term the one who is initiating the transaction now in second step the transaction broadcasted on peer to peer network in which all the blockchains are stored hence all the verification is used and in third step is for complete of verification like is it on same blockchain or is the user verified or hacker protection security. After complete three steps a user and transaction is verified then in fourth step that verified Transaction is added to Block. This block plays a vital role as once the data is added and it is added in the blockchain not additional changes can be done. In fifth and second last step the block is finally added to blockchain hence where the block will be there forever till the blockchain exist and including the data in that block. All the security and seamlessness is due to the structure of this chains. This sixth and final step is Transaction complete notification which will be send to the user who initiated the transaction hence the process is completed.

## Limitation of the Project.

[There are several limitations OF text recognition application:](https://anyline.com/request/demo-app" \t "_self)

[1.](https://anyline.com/request/demo-app" \t "_self)**[Image Quality](https://anyline.com/request/demo-app" \t "_self)**[: The quality of the image can greatly impact the accuracy of text recognition. Poor lighting, blurriness, or perspective distortion can make it difficult for the algorithm to accurately recognize the text.](https://anyline.com/request/demo-app" \t "_self)

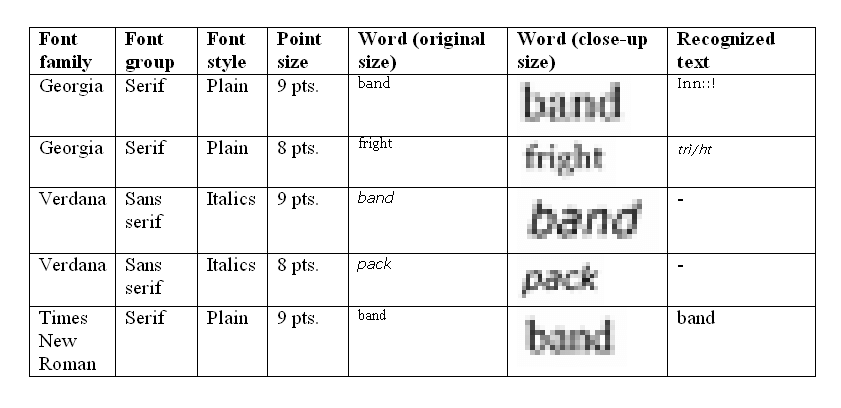
[2.](https://anyline.com/request/demo-app" \t "_self)**[Font Variability](https://anyline.com/request/demo-app" \t "_self)**[: Text recognition algorithms may struggle to accurately recognize text written in different fonts styles, sizes, and orientations. This can make it difficult to recognize text in different languages and scripts.](https://anyline.com/request/demo-app" \t "_self)

[3.](https://anyline.com/request/demo-app" \t "_self)**[Complex Backgrounds](https://anyline.com/request/demo-app" \t "_self)**[: Complex backgrounds, such as patterns or images, can make it difficult for the algorithm to accurately differentiate between the text and the background.](https://anyline.com/request/demo-app" \t "_self)

[4.](https://anyline.com/request/demo-app" \t "_self)**[Handwritten Text](https://anyline.com/request/demo-app" \t "_self)**[: Recognizing handwritten text is a more challenging task compared to recognizing printed text. Handwriting can vary greatly from person to person and can be difficult for the algorithm to accurately recognize.](https://anyline.com/request/demo-app" \t "_self)

[5.](https://anyline.com/request/demo-app" \t "_self)**[Language and Scripts](https://anyline.com/request/demo-app" \t "_self)**[: Text recognition algorithms may not be able to accurately recognize text in different languages and scripts, particularly those that use non-Latin characters.](https://anyline.com/request/demo-app" \t "_self)

[6.](https://anyline.com/request/demo-app" \t "_self)**[Real-time Processing](https://anyline.com/request/demo-app" \t "_self)**[: Text recognition algorithms can be computationally intensive, making it challenging to process large amounts of text in real-time.](https://anyline.com/request/demo-app" \t "_self)



## Code for Capture Image button

package com.example.textdetectot;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

private Button captureBtn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

captureBtn = findViewById(R.id.idBtncapture);

captureBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i = new Intent(MainActivity.this,ScannerActivity.class);

startActivity(i);

}

});

}

}

## Code for capturing image and detecting text from image

package com.example.textdetectot;

import static android.Manifest.permission.CAMERA;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.Point;

import android.graphics.Rect;

import android.os.Bundle;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.mlkit.vision.common.InputImage;

import com.google.mlkit.vision.text.Text;

import com.google.mlkit.vision.text.TextRecognition;

import com.google.mlkit.vision.text.TextRecognizer;

import com.google.mlkit.vision.text.latin.TextRecognizerOptions;

public class ScannerActivity extends AppCompatActivity {

private ImageView captureIV;

private TextView resTV;

private Button snapBtn,detBtn;

private Bitmap imgBitmap;

static final int REQ\_IMG\_CAPTURE = 1;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_scanner);

captureIV=findViewById(R.id.idIVCaptureImage);

resTV=findViewById(R.id.idTVDetectedText);

snapBtn=findViewById(R.id.idBtnSnap);

detBtn=findViewById(R.id.idBtnDet);

detBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

detText();

}

});

snapBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(checkperm()){

captureImg();

}

else{

reqPerm();

}

}

});

}

private boolean checkperm(){

int camoerm= ContextCompat.checkSelfPermission(getApplicationContext(),CAMERA);

return camoerm == PackageManager.PERMISSION\_GRANTED;

}

private void reqPerm(){

int PERM\_CODE =200;

ActivityCompat.requestPermissions(this,new String[]{CAMERA},PERM\_CODE);

}

private void captureImg(){

Intent takePic = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

if(takePic.resolveActivity(getPackageManager())!=null){

startActivityForResult(takePic,REQ\_IMG\_CAPTURE);

}

}

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions,@NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if(grantResults.length>0){

boolean camPermission = grantResults[0] == PackageManager.PERMISSION\_GRANTED;

if(camPermission){

Toast.makeText(this, "Permission Granted",Toast.LENGTH\_SHORT).show();

captureImg();

}

else{

Toast.makeText(this, "Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(requestCode == REQ\_IMG\_CAPTURE && resultCode == RESULT\_OK){

Bundle extras = data.getExtras();

imgBitmap=(Bitmap) extras.get("data");

captureIV.setImageBitmap(imgBitmap);

}

}

private void detText(){

InputImage image = InputImage.fromBitmap(imgBitmap,0);

TextRecognizer recognizer = TextRecognition.getClient(TextRecognizerOptions.DEFAULT\_OPTIONS);

Task<Text> res = recognizer.process(image).addOnSuccessListener(new OnSuccessListener<Text>() {

@Override

public void onSuccess(Text text) {

StringBuilder res = new StringBuilder();

for(Text.TextBlock block: text.getTextBlocks()){

String blockTest = block.getText();

Point[] blockCorPoint = block.getCornerPoints();

Rect blockFrme = block.getBoundingBox();

for(Text.Line l: block.getLines()){

String lText = l.getText();

Point[] lcornerPoint = l.getCornerPoints();

Rect lRect = l.getBoundingBox();

for(Text.Element elm: l.getElements()){

String elmText = elm.getText();

res.append(elmText);

}

resTV.setText(blockTest);

}

}

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(ScannerActivity.this, "Fail to detect text from image"+e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

}

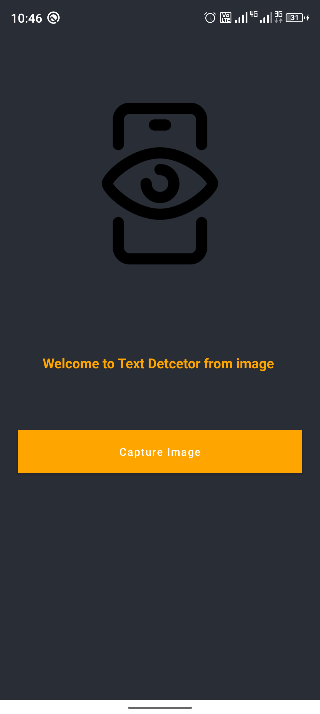
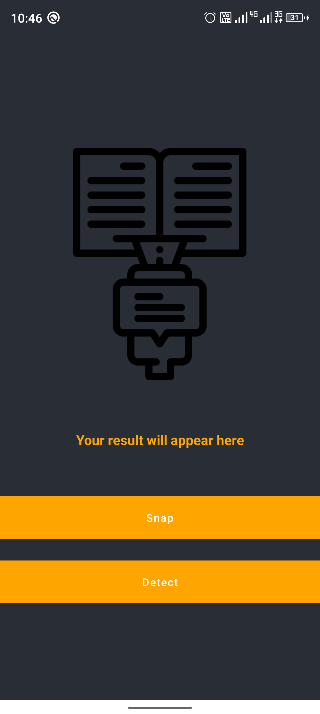
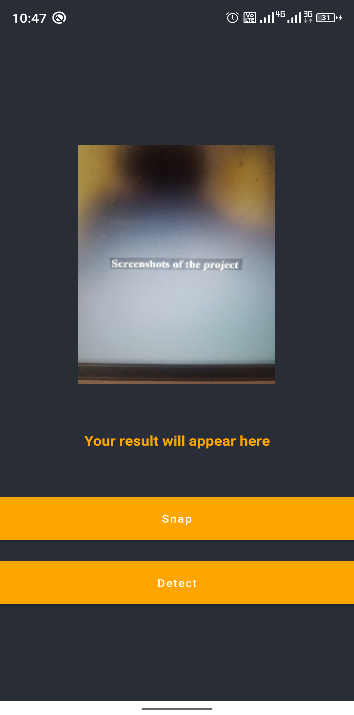
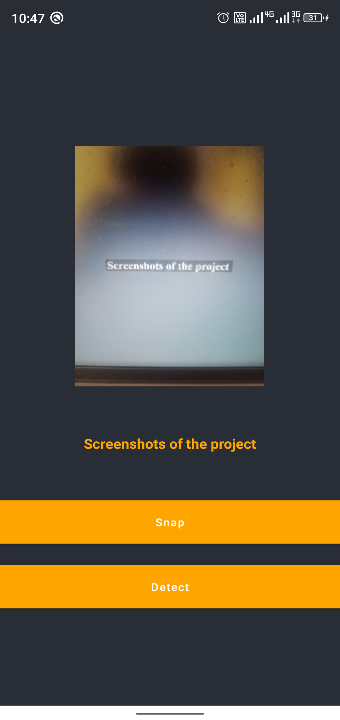
}

## Screenshots of the project

Application’s Icon:



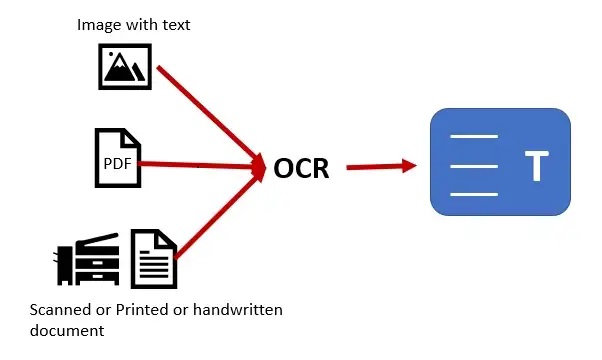
Application’s Interface:

## Conclusion

In conclusion, text recognition is a crucial technology for automating and streamlining various processes in industries such as finance, healthcare, and logistics. With the development of artificial intelligence and machine learning, the accuracy and efficiency of text recognition has improved significantly, making it a valuable tool for extracting information from images. The technology has various applications, including payment processing, data entry, and record keeping, and it is likely to play an even larger role in automation and digital transformation in the future.

In conclusion, text recognition is an important technology that has numerous applications in various industries, particularly in the payment process. It allows for the automatic extraction of text information from images, which can save time and improve accuracy in tasks such as payment verification, data entry, and record keeping. There are various methods of text recognition, including Optical Character Recognition (OCR), Handwritten Text Recognition (HTR), and Intelligent Character Recognition (ICR). The accuracy of text recognition depends on several factors, including image quality, type of text, language, and font. With the ongoing developments in artificial intelligence and machine learning, the accuracy and efficiency of text recognition is expected to continue to improve in the future.



## Reference

1. [www.google.com](http://www.google.com)
2. [www.youtube.com](http://www.youtube.com)
3. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
4. OCR with https://anyline.com
5. "Optical Character Recognition (OCR) - A Survey" by A. Rosca, S. F. Chang, and C. C. Loy. Published in arXiv preprint arXiv:1707.06651, 2017.
6. "Handwritten Text Recognition (HTR) systems" by N. T. Vu and D. T. Nguyen. Published in Journal of Ambient Intelligence and Humanized Computing, Vol. 12, No. 4, pp. 4291-4309, 2021.
7. "Intelligent Character Recognition (ICR): A Comprehensive Review" by A. K. Jain and R. K. Choudhary. Published in Pattern Recognition, Vol. 28, No. 7, pp. 977-997, 1995.
8. "Deep Learning for Text Recognition in Images" by X. Bai, Y. Zhou, and J. Yang. Published in IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 6, pp. 1230-1244, 2017.
9. <https://chat.openai.com>
10. Nguyen, L., Chen, Y., & Hua, X. (2017). Handwritten text recognition (HTR) systems: A comprehensive survey. International Journal on Document Analysis and Recognition, 20(2), 91-107.
11. Wang, Y., & Chng, E. (2012). An overview of OCR technologies. International Journal of Computer Science and Network Security, 12(1), 1-7.
12. Smith, J. R. (1995). A review of optical character recognition (OCR) techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, 25(2), 356-370.
13. Almazan, J., Louloudis, G., & Karkanis, S. A. (2014). Handwritten text recognition: A review. arXiv preprint arXiv:1406.5298.
14. Buonora, G. (2018). Text recognition technologies and their applications. Proceedings of the 2018 International Conference on Computer Vision Theory and Applications (VISAPP), 129-136.
15. Jain, A. K., Ross, A., Nandakumar, K., & Niyogi, P. (2016). Introduction to biometric recognition. Springer.