

# **REGIONAL TRANSPORT OFFICE MANAGEMENT SYSTEM USING BLOCKCHAIN**

## **A Project Report**

Submitted in the partial fulfillment of the requirements for the  
award of the degree of

**Bachelor of Technology**

**in**

**Department of Computer Science and Engineering**

**By**

180030029      P.AKHIL

180030039      V.YASHITHA

180031071      D.YASWANTH

180031097      B.CHANDRIKA RANI

Under the supervision of

**Mr. MADHAVARAPU CHANDAN** (Asst professor)



**Department of Computer Science and Engineering**

K L E F, Green Fields,

Vaddeswaram- 522502, Guntur(Dist), Andhra Pradesh, India.

November, 2021





## **DECLARATION**

The Project Report entitled “REGIONAL TRANSPORT OFFICE MANAGEMENT SYSTEM USING BLOCKCHAIN” is a record of bonafide work of P.Akhil-180030029, V.Yashitha-180030039, D.Yaswanth-180031071, B. Chandrika Rani-180031097, submitted in partial fulfillment for the award of B.Tech in Computer Science and Science to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

<b>ID NO</b>	<b>NAME</b>
180030029	P.AKHIL
180030039	V.YASHITHA
180031071	D.YASWANTH
180031097	B.CHANDRIKA RANI

## **CERTIFICATE**

This is to certify that the (Term Paper/Project) Report “REGIONAL TRANSPORT OFFICE MANAGEMENT SYSTEM USING BLOCKCHAIN” is being submitted by P.Akhil-180030029, V.Yashitha-180030039, D.Yaswanth-180031071, B.Chandrika Rani-180031097 submitted in partial fulfillment for the award of B.Tech in Computer Science and Science to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute.

**Signature of the Supervisor**

Mr. MADHAVARAPU CHANDAN

**Signature of the HOD**

**Signature of the External Examiner**

## **ACKNOWLEDGEMENT**

Our sincere thanks to **Mr. MADHAVARAPU CHANDAN** in the research for her outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Mr. MADHAVARAPU CHANDAN** Faculty for the project course in Computer Science and Engineering Department for providing us with adequate planning and support and the means by which we can complete this.

We express our gratitude to **Mr. V. HARI KIRAN**, Head of the Department for computer science and Engineering for providing us with adequate facilities, ways, and means by which we can complete this Research work.

We would like to place on record the deep sense of gratitude to the honorable Vice-Chancellor, KL University for providing the necessary facilities to carry the work.

Last but not the least, we thank all the Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our work.

<b>ID NO</b>	<b>NAME</b>
180030029	P.AKHIL
180030039	V.YASHITHA
180031071	D.YASWANTH
180031097	B.CHANDRIKA RAN

## **ABSTRACT**

As the population grows, so does the number of automobiles on the road is increasing. Keeping the valid license and vehicle papers is mandatory. It is very tedious to keep all this documents with us always and also it is tough and time consuming for traffic cops to review and authenticate these documents while also keeping track of the vehicles' challans. Therefore, we proposed a system in which the work of traffic police and driver (user) is simplified. In this system the driver will register to RTO services and the login credentials will be provided to driver to login. The driver will upload document details like RTO driving license etc. Therefore it is not necessary to him to carry hard copy driving license with him and also user can raise a complaint against the police. At RTO traffic police end the police will open the application and enter the vehicle number and the data from server will be fetched and documents details like Emission, Insurance etc, for verification will be shown. The traffic police will be able to generate the challan through the application and it will be linked to the vehicle and the driver has to pay the challan online. This application assists both drivers (users) and traffic cops.

## **TABLE OF CONTENTS**

<b>S.NO</b>	<b>TOPICS</b>	<b>PAGE NUMBERS</b>
1	Introduction	1-3
2	Literature Survey	4-5
3	Requirements	6
4	System Architecture	7-8
5	System Design	9-11
6	Implementation	12-19
7	Pseudo Code	20-42
8	Snapshots	43-60
9	Conclusion	61
10	References	62

# **CHAPTER 1: INTRODUCTION**

## **Overview**

RTO vehicle inspection is a web application to inspect vehicle documents like registration card, driving license, emission test and insurance copy through application. This application contains users or drivers registration by enters details and uploading the all vehicle details to application will generate unique personal id for all users. All users can login to the application using username and password to update the documents regularly.

Digital vehicle inspector application have admin or generally traffic police login to verify or check all user details with documents. This application is mostly useful for traffic police to check vehicle owner details in public premises and put the fain for false details or wrong vehicle owners or drivers without DL and also to catch vehicle thief. This application will send the fine notifications to individual owners.

## **Ethereum**

Ethereum is a Blockchain-based distributed computing platform established by Vitalik Buterin in 2013 and launched in 2015 after a successful online crowd sale. Furthermore, because Ethereum runs the decentralized application programming code, it is a programmable Blockchain, allowing users to create new applications. A smart contract is a programming code that allows money, data, and content to be exchanged using Blockchain technology by creating, deploying, and operating Decentralized Software applications.

## **Ethereum Virtual Machine (EVM)**

The Ethereum Virtual Machine (EVM) is a Turing complete programming that permits clients to send and run different projects written in different programming dialects (for instance, Solidity) and extraordinarily improves on the advancement of Blockchain applications. Since it is a sandbox climate, it is detached from the remainder of the principle organization. Therefore, each Ethereum network executes its own EVM execution utilizing similar directions. Ether, the Ethereum money, is utilized to pay for calculations in the EVM (ETH).

## **Transactions**

A transaction is a data package containing messages transmitted between Ethereum accounts which have been signed (the transaction sender to the recipient of the transaction). The transactions are validated by mining or, more precisely, by producing a signature with the transaction sender's private key. The Ethereum beneficiary location, the ETH gas value (cost of a unit of gas burned-through), the moved sum (greatest measure of gas to be used), and other optional data are even included in the transaction.

## **Ether and Gas**

Ether is a shaky cash of Ethereum paid by the trade sender for executing the code in an agreement set off by a message or an exchange and capacity and calculation. Besides, The execution cost, just as the expense of organization use, are communicated as far as gas. All the more explicitly, gas is a decent unit that evaluates the measure of register work needed by minors to play out a given activity. Therefore, clients use ETH to pay for gas. Sensors 2020, 20, 3928 5 of 27 are the Ethereum network hubs that acknowledge, proliferate, approve, and execute exchanges. Since various tasks require various measures of gas, as far as possible is the most extreme measure of gas that the sender will pay for the exchanges. There's significantly more data on ether and gas about Ether and Gas can be viewed as in.

## **Proof-of-Work (PoW):**

Verification of-Work addresses the first agreement calculation that decides the legitimacy of the exchange utilized in Blockchain to guarantee security. The expense of execution, just as the expense of organization utilization, is communicated in gas. To be more unequivocal, a gas is a decent unit that actions the measure of figure work needed by minors to do a job. Thus, clients can pay for gas with ETH. Through PoW, diggers take care of numerical issues by utilizing exceptional programming. Since various exercises request shifting measures of gas, as far as possible is the most extreme measure of gas that the sender will pay for the exchanges. The hash of the past block is consolidated into each new square. Consequently, the additional square turns into a piece of the chain that can be followed until the first or thereabouts called qualities is block.

## **Existing System:**

In a densely populated country like India, there are a lot of vehicles. It's a regular scenario for traffic cops to come across a random vehicle and begin verifying the vehicle's details. This might impede traffic, and if the cops aren't acting appropriately, users have no way to file a complaint on the moment, which can lead to disaster. With the advancement of technology, mobile applications and web-based systems have been available to speed up the process by which traffic cops can fine the vehicle owner or consumers can pay the fine online, albeit these methods are not always safe. However, there is currently no system in place that allows traffic cops to obtain all of the information on a specific vehicle. This prompted us to conceive and construct an application through which traffic cops may obtain all relevant information, including the vehicle's owner, registration, and insurance status, with only one click.

## **Proposed System:**

To overcome this problem we designed the web application in such a way that all the vehicle document details will be available on one click by entering the vehicle number due to this the traffic can be avoided as much as possible, and the user or driver can raise the complaint against the police on the spot through our application, and also the user or driver can pay the fine using our application securely to make sure the security of transaction we have used the blockchain concept. Police can also check whether the old penalties is remaining or not using fine view.

## **Problem Statement:**

Here, we are fostering a web application for RTO so here we give a concise portrayal of our task outline. In any case, we give unmistakable environment suggests the helpless customer can get to this site for their work reason related to RTO. First client needs to fill the enrollment structure with the goal that we give confirmation to him. Assuming client needs to pass his vehicle number then additionally it requires some investment in old framework. The supervisor is obliging approval reason similarly as it handles every one of the informational collection of RTO and manages all the communication. He has position to supported learning permit number, long-lasting permit number; pass the vehicle registration number, etc. User have to find the police station and then raise the complaint this takes time. To pay the fine they have to go to RTO office this also is waste.

## **CHAPTER 2: LITERATURE SURVEY**

### **Research of evaluation of Real Face Mask Detection**

In this section, We looked at many researchers' attempts to create and construct a technology-assisted solution for the convenience of traffic cops and RTOs. To begin, Avanish proposed a system in which a traffic police officer uses a Smart Card Reader to scan the driver's license. The license is essentially a smart card with an embedded micro module that includes a microprocessor and memory. As the officer scans the license, the officer's device's application displays the driver's information. Meanwhile, the system checks the driver's license, and a penalty of the appropriate amount is charged to the motorist based on the regulations that the officer has specified, and the driver's record is updated on the main server.

Limitations of this system are

- >> Additional cost incurred due to overhead of designing smart card based licenses.
- >> In densely populated country like India, it is not feasible to replace all regular licenses with new one.

User module, traffic module, and RTO admin module are all included in the Smart RTO online and android application. The user module in this system includes of forms that the user must fill out, such as license registration, automobile registration, and insurance forms. The traffic module primarily focuses on delivering information to traffic police officers, such as license checks, vehicle information checks, and insurance checks, as well as generating fines in the event that a driver breaks the regulations. In order to generate a license and vehicle number, the RTO administrator checks the information entered into the forms by the user. Data security procedures to prevent data leakage or tampering are not considered as a system limitation. The E-RTO management system includes vehicle registration and the storage of associated data about the vehicle and its owner, such as insurance, license plate number, and so on. It also involves vehicle authentication utilizing RFIDs provided by the RTO following successful verification. RTO is a high-level user with the ability to check and handle data provided by the user (vehicle owner/driver).

The architecture consists of three modules namely RTO module, Flex module and Court module. In this architecture, User licenses, vehicle information, vehicle insurance information, and vehicle emission information are all stored in the RTO module. The Traffic Police officer uses the Flex module to verify the information on the vehicle and the driver's license. On the same system, it also allows you to make fine amounts. Officer Id, Date, Vehicle Registration, List of Offenses, Fine Amount, and Payment Status are all included in the Court module. A. Choudhury discussed the number plate identification method based on horizontal and vertical edge processing and zone detection. This method has been shown to be an effective way to guess the vehicle's number from a number plate image captured from any angle or with any background.

Amruta proposes an RTO verification system in which the authors suggest that a centralized RTO database encompassing vehicle and driver information be maintained. The license number plate region will be retrieved from the recorded image using image segmentation techniques. And the technique used for the character recognition on number plate is Optical character recognition. The system is implemented and simulated using Matlab. Using feedback system, the extracted license plate number is then given to GSM modem for further SMS feedback system to the user and concerned authority.

## CHAPTER 3: REQUIREMENTS

The hardware and software requirements of the proposed system are as follows:

### Hardware Requirements

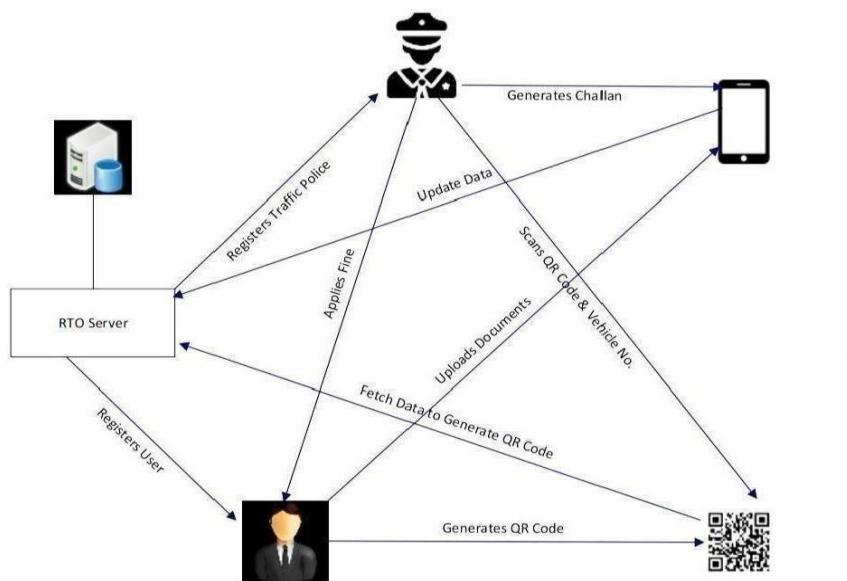
- **Processor:** Pentium –IV Speed-1.1Ghz
- **RAM:** 256MB
- **Hard disk:** 20GB
- **Mouse:** Two or Three button mouse

### Software Requirements

- **Operating System:** Windows 10, 64 –bit.
- **Front End :** HTML5, CSS, Bootstrap
- **Back End:** Python, Solidity
- **Tools:** Robo3T, Sublime, Ganache, Remix, Mongodb

## CHAPTER 4: SYSTEM ARCHITECTURE

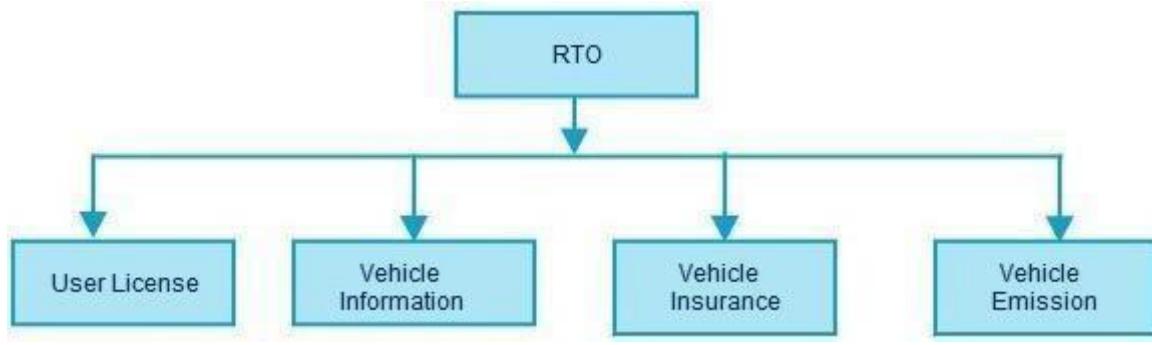
The system consists of three main modules i.e. admin, traffic police & user. The user registers at admin for the service and admin verifies the user and registers the user's information in RTO database. The traffic police are also registered by the admin. After registering the user logs in to the user app code for the license. The traffic police enter vehicle number to check the license and all vehicle documents. The traffic police determine the challan and apply it to the driver and the challan data will be updated to the RTO database. The admin has all the rights to view update and insert data into the RTO database.



We are proposing a model where we plan to offer better types of assistance through cells. The architecture mainly consisting three modules:

The License module incorporates all the data with respect to candidates individual subtleties like Name, DOB, Address, Contact no. Identity marks,

Photograph and furthermore permit subtleties like Type of permit, License status and Validity of the permit. It is likewise answerable for adding and overseeing permit related records to the information base.



Flex module: This module is utilized by the Traffic Police official. It is predominantly used to perform check of Vehicle and Driver License data.

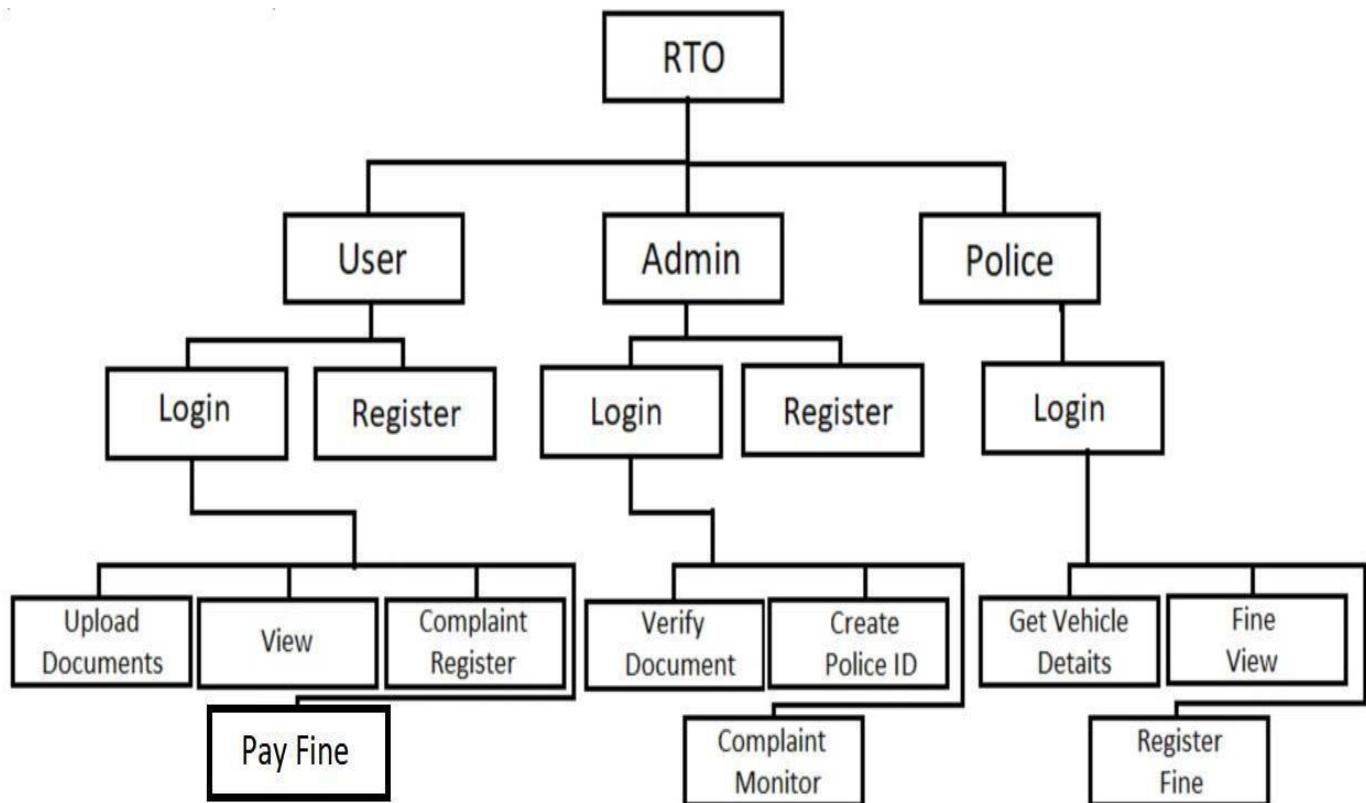
The court module: It contains data with respect to the officials ID, date, Vehicle Registration, rundown of offenses, Fine Amount, and Payment Status.

It likewise incorporates the outflow subtleties like Center Name, Emission Id and Validity. It is additionally liable for adding and overseeing discharge related records to the data set.

# CHAPTER 5: SYSTEM DESIGN

## Data Flow Diagram

Information stream outlines are utilized to graphically address the progression of information in a business data framework. DFD depicts the cycles that are engaged with a framework to move information from the contribution to the document stockpiling and reports age. The actual information stream graph portrays the execution of the intelligent dataflow.



This are the flow of our diagram:

- This data flow diagram has four levels at the first level RTO database will work.
- In the second level there are three main division.

## **1. User**

- In the third level user can register by giving personal credentials and login.
- Login have its own criteria at fourth level were user have to upload the vehicle document and view the document.
- User can raise a complaint against police if they have abused.
- User can also pay fine through our application.

## **2. Admin**

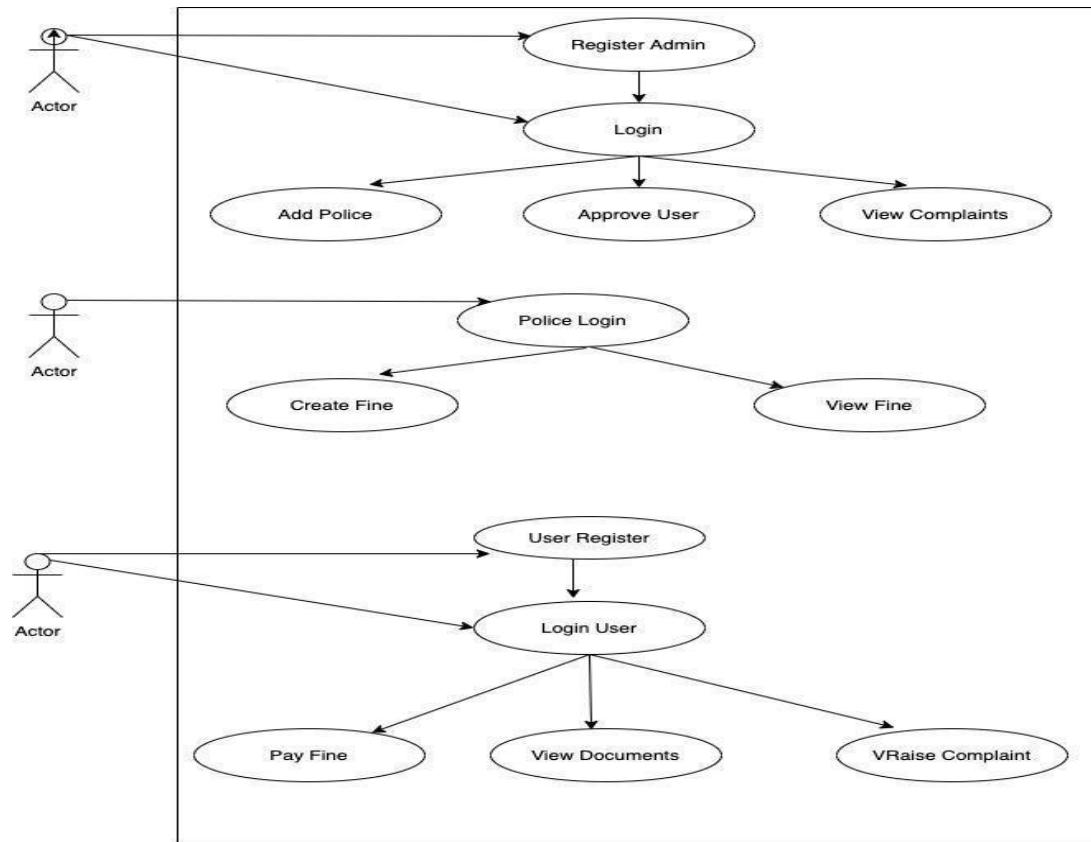
- In the third level Admin can register by giving personal credentials and login.
- Admin can verify the vehicle documents which has been uploaded by the user.
- Admin can view the complaints registered by the user.

## **3. Police**

- In the third level Police can login by the provided login credentials.
- Police can view the vehicle documents which has been verified by the RTO officer.
- Police can view the old fines which have not paid.
- Police can create a new fine.

## Use Case Diagram

A use case diagram is a low-level diagram which is used to illustrate the different ways in which the user can interact with the proposed system. It therefore provides the behavior of the system on the exterior. It does not specify the order in which the events occur.



The action performed at the admin level are admin can register himself and login, then he/she can create police login id ,approve the users vehicle document and view the complaint. Police can login using provided login credentials and create fine and view the fine.

# CHAPTER 6: IMPLEMENTATION

## Backend Implementation

### Python

Python is a high-level computer language for general-purpose programming that is interpreted. Python was created by Guido van Rossum and initially released in 1991. It has structures that allow for clear programming at both local and large sizes. Python has a dynamic type system and memory management that is automated. It features a big and extensive standard library and supports several programming paradigms, including object-oriented, imperative, functional, and procedural.

### Python Features

Python's features include:

- Easy to learn: Python has not many catchphrases, basic construction, and a plainly characterized one.
- Easy to read: Python code is all the more obviously characterized and apparent to the eyes.
  - Easy to maintain: Python's source code is fairly easy-to-maintain.
- Interactive Mode: Python has support for an intelligent mode which permits intuitive testing and troubleshooting of pieces of code.

### Uses of Python

Python is utilized by a huge number of developers and is utilized in many spots. At times just Python code is utilized for a program, yet more often than not it is utilized to do basic positions while one more programming language is utilized to accomplish more muddled undertakings. Its standard library is comprised of many capacities that accompany Python when it is installed. Some things that Python is often used for are:

- Web development
- Game programming
- Desktop GUIs
- Scientific programming
- Network programming.

## **Blockchain**

The blockchain technology has potential applications in various areas such as smart contracts, the Internet of Things, land registry, supply chain management, storing medical data, and identity management are all possible applications of blockchain technology. Despite the fact that there are over 6,000 active Blockchain software projects on GitHub, few software engineering academics have looked into these projects and their contributors.

According to the findings, the majority of BCS developers have prior experience with non-BCS development and are largely motivated by the goal of developing a decentralized financial system.

## **SOLIDITY REMIX**

Solidity is the main programming language for writing smart contracts for the ethereal blockchain. It is statically typed, and supports inheritance, libraries, and more! In short, it has all the capability that you need in order build industrial strength block chain applications. Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs that control how accounts behave in the ethereal state. The language of solidity is written in curly brackets.

Solidity is statically typed and, among other things, enables inheritance, libraries, and sophisticated user-defined types.

### Available Solidity Integrations

- Visual Studio Extension

Solidity plugin for Microsoft Visual Studio that includes the Solidity compiler.

- Package for Sublime Text — Solidity language syntax.
- Ether atom

Module for the Atom manager that highlights linguistic structure featuring, accumulation and a runtime environment.

- Vim Solidity

Plugin for the Vim editor providing syntax highlighting.

- Vim Syntactic

Plugin for the Vim editor providing compile checking.

**Discontinued:**

- Ethereal Studio

Specialized web IDE that also provides shell access to a complete Ethereal environment.

**Solidity Tools**

- sol graph

Picture Solidity control stream and feature potential security weaknesses.

- evmdis

EVM Disassembler that performs static examination on the bytecode to give a more significant level of deliberation than crude EVM activities.

- Doxity

Documentation Generator for Solidity.

With Solidity you can make contracts for utilizations like democratic, crowdfunding, blind closeouts, and multi-signature wallets.

**MongoDB**

MongoDB is an open-source record information base and driving NoSQL data set. This instructional exercise will give you incredible comprehension on MongoDB ideas expected to make and convey a profoundly adaptable and execution situated dataset.

MongoDB is a cross-stage, record situated data set that gives, elite execution, high accessibility, and simple adaptability. MongoDB works on concept of collection and document.

### **Database:**

Database is a physical container for collections. Every information base gets its own arrangement of records on the document framework. A single MongoDB server typically has multiple databases.

### **Collection:**

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields.

### **Document:**

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic construction implies that records in a similar assortment don't have to have similar arrangement of fields or design, and normal fields in an assortment's archives might hold various sorts of information.

### **Sample Document:**

Following model shows the archive design of a blog webpage, which is just a comma isolated key worth pair.

### **MongoDB - Advantages**

Any social information base has a commonplace outline plan that shows number of tables and the connection between these tables. While in MongoDB, there is no understanding of relationship.

#### Advantages of MongoDB over RDBMS

- **Schema less** – MongoDB is an archive data set in which one assortment holds various records. Number of fields, content and size of the record can vary starting with one report then onto the next.
- Structure of a single object is clear.
- No complex joins.
- Deep query- MongoDB upholds dynamic questions on records utilizing a report based inquiry language that is close to as amazing as SQL.
- Tuning

## **Smart contract**

A brilliant agreement is a decentralized application that executes business rationale in light of occasions. Brilliant agreement execution can bring about the trading of cash, conveyance of administrations, opening of content secured by computerized privileges the board or different kinds of information control, for example, changing the name on a land title. Savvy agreements can likewise be utilized to authorize security assurance by, for instance, working with the particular arrival of protection ensured information to meet a particular solicitation. There are an assortment of designs for how the projects supporting brilliant agreements are created, dispersed, oversaw and refreshed. In spite of the name, shrewd agreements are not lawfully authoritative agreements. Their primary capacity is to automatically execute business rationale that performs different errands, cycles or exchanges that have been modified into them to react to a given arrangement of conditions. Legitimate advances should be attempted to connect this execution to lawfully authoritative arrangements between parties.

## **How do smart contracts work?**

Business groups draw in with designers to build up their necessities for the savvy agreement's ideal conduct because of different occasions or conditions, which is the initial phase during the time spent making a brilliant agreement. Conditions like installment approval, shipment receipt, or a utility meter perusing limit are instances of straightforward occasions. More perplexing events, for example, deciding the worth of a subsidiary monetary instrument and finishing a subordinate arrangement, or consequently delivering a protection installment on account of an individual's passing or a catastrophic event, may be encoded utilizing more complex rationale.

The developers then create the logic and test it on a smart contract writing platform to check that it works as expected. After the application is written, it is passed on to a security review team. An interior master or an organization that works in verifying savvy contract security could be utilized. The agreement is then sent on a current square chain or other conveyed record framework whenever it has been approved.

The smart contract is arranged to tune in for occasion refreshes from a "prophet," which is successfully a cryptographically gotten streaming information source, whenever it has been sent. The brilliant agreement executes once it gets the fitting blend of occasions from at least one prophets.

### **Smart contract applications and block chain**

Blockchain is ideal for storing smart contracts because of the technology's security and immutability. On a shared ledger, smart contract data is encrypted, making it nearly impossible to lose the information recorded in the blocks. Another benefit of incorporating blockchain technology into smart contracts is its flexibility. Developers can store nearly any form of data in a blockchain, and they can choose from a wide range of transaction alternatives. Smart contracts built on the blockchain are making transactions and other corporate operations safer, efficient, and cost-effective, lowering transaction costs. Cook County, Illinois, for example, employed blockchain to construct a database to transfer and track property titles in 2016.

### **Smart contract advantages**

There are several potential business advantages from using smart contracts.

Cost efficiency. Smart contracts guarantee to mechanize business processes that length hierarchical limits. This can wipe out numerous functional costs and save assets, including the work force expected to screen the advancement of a perplexing cycle that executes in light of conditions that range organizations. Handling speed. Smart agreements can further develop the handling rate of business processes that stumble into numerous undertakings.

Smart contracts are executed automatically by the network, eliminating the need for a third party to oversee commercial transactions. It also offers automated transactions, which eliminate the possibility of human error and ensure contract execution accuracy.

## **Frontend implementation**

### **HTML AND CSS**

HTML represents Hyper Text Markup Language. It is utilized to configuration website pages utilizing a markup language. HTML is the mix of Hypertext and Markup language. Hypertext characterizes the connection between the website pages. A markup language is utilized to characterize the text report inside label which characterizes the construction of website pages. This language is utilized to comment on text so a machine can get it and control text appropriately. Most markup dialects are comprehensible. The language utilizes labels to characterize what control must be done on the text.

HTML and CSS are two of the most common Web page construction technologies. For a range of devices, HTML provides the page structure and CSS offers the layout. HTML and CSS, along with images and coding, are the foundations for creating Web pages and Web applications. Cascading Style Sheets is an acronym for Cascading Style Sheets. CSS specifies how HTML elements should appear on a screen, in print, or in other media. CSS helps you save time and effort. It has the ability to control the layout of numerous web pages at the same time. CSS files contain external stylesheets.

Cascading Style Sheets, is a simple design language intended to make the process of making web pages presentable easier. Styles can be applied to web pages using CSS. More crucially, CSS allows you to do so without having to worry about the HTML code that makes up each web page. CSS is simple to learn and understand, but it gives you a lot of power over how an HTML document looks.

### **WHY CSS?**

CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.

Easy Maintenance: To make a global change, simply alter the style, and all elements on all webpages will be automatically updated. Search Engines: Because CSS is a clean coding method, search engines will have an easier time "reading" its content. CSS offers a considerably larger set of attributes than HTML, therefore you may give your HTML page a far more professional appearance than HTML attributes.

## **Bootstrap:**

Bootstrap is a toolkit for building responsive websites and web applications that is free and open-source. It is the most widely used HTML, CSS, and JavaScript framework for creating mobile-first, responsive websites. It fixes a number of issues that we experienced previously, including cross-browser compatibility. Nowadays, webpages are optimised for all browsers (Internet Explorer, Firefox, and Chrome) and screen sizes (Desktop, Tablets, Phablets, and Phones). All due to Twitter's Mark Otto and Jacob Thornton, who created Bootstrap, which was eventually declared an open-source project.

Why Bootstrap?

Faster and Easier Web Development.

It creates Platform-independent web pages. It creates Responsive Web-pages.

It designed to be responsive to mobile devices too

## CHAPTER 7: PSEUDO CODE

### Mycontract.py

```
import json
from web3 import Web3

#[{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}, {"constant":true,"inputs":[],"name":"greet","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":true,"inputs":[],"name":"greeting","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"inputs":[],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]

# Set up web3 connection with Ganache
ganache_url = "http://127.0.0.1:7545"
web3 = Web3(HTTPProvider(ganache_url))

web3.eth.defaultAccount = web3.eth.accounts[0]

abi =
json.loads('[{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}, {"constant":true,"inputs":[],"name":"greet","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":true,"inputs":[],"name":"greeting","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"inputs":[],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]')

address = web3.toChecksumAddress("0x85e5869CF8117044dA9a32A2f657442A1f2Cf6b6")
contract = web3.eth.contract(address=address, abi=abi)
print(contract.functions.greet().call())

tx_hash = contract.functions.setGreeting('contactOne').transact()

#print(tx_hash)
web3.eth.waitForTransactionReceipt(tx_hash)
print('updated greeting: {}'.format(
contract.functions.greet().call()))
```

## App.py

```
from bson.json_util import dumps
from werkzeug.utils import secure_filename

import os
import os.path

import pymongo
import json, ast
import os
import uuid
import requests
import random
#import urllib2
import datetime
#import urllib.request
from urllib.request import urlopen
from bson import json_util

from web3 import Web3
import smtplib

from flask_cors import CORS

ganache_url = "http://127.0.0.1:7545"
web3 = Web3(HTTPProvider(ganache_url))

web3.eth.defaultAccount = web3.eth.accounts[0]

abi =
json.loads('[{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}, {"constant":true,"inputs":[],"name":"greet","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":true,"inputs":[],"name":"greeting","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":false,"inputs":[],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]')

#var greeterContract = new
web3.eth.Contract([{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}, {"constant":true,"inputs":[],"name":"greet","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":true,"inputs":[],"name":"greeting","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {"constant":false,"inputs":[],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]);
```

```

address = web3.toChecksumAddress("0x85e5869CF8117044dA9a32A2f657442A1f2Cf6b6")
account_2 = '0x8c1080cB33976a0Cd49132129cDDbDa256C5ac8c' # Fill me in Admin account

app = Flask(__name__)
CORS(app)

myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["BlockChain"]
userColl = mydb["User"]
rtoColl = mydb["RTO"]
policeColl = mydb["Police"]
fineColl = mydb["Fine"]
cocontractColl = mydb["Contract"]
vechileColl = mydb["Vechile"]
complaintColl = mydb["Complaint"]

@app.route('/user/details',methods=['POST'])
def userDetails():
    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    myquery = { "userID" : raw["userID"] }
    print(myquery)
    mydoc = userColl.find_one(myquery, { "_id":0 })
    print(mydoc)
    if mydoc is not None:
        resp = { "success" : True, "message":"success","userObj": mydoc }
    else:
        return { "success" : False, "message":"Sorry no user found" }
    return resp

@app.route('/user/login',methods=['POST'])
def userLogin():
    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    userName = raw["userName"]
    password = raw["password"]
    print("here 1")
    print(userName)
    print(password)
    myquery = { "userName" : userName, "password": password,"approval": "ADMINAPPROVED" }
    print(myquery)

```

```

mydoc = userColl.find_one(myquery, {"_id":0})
print(mydoc)
if mydoc is not None:
    resp = {"success" : True, "message":"success","userObj": mydoc}
else:
    return {"success" : False, "message":"Sorry no user found"}
return resp

@app.route('/rto/login',methods=['POST'])
def rtoLogin():
    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    userName = raw["userName"]
    password = raw["password"]
    print("here 1")
    print(userName)
    print(password)
    myquery = { "userName" : userName, "password": password}
    print(myquery)
    mydoc = rtoColl.find_one(myquery, {"_id":0})
    print(mydoc)
    if mydoc is not None:
        resp = {"success" : True, "message":"success","userObj": mydoc}
    else:
        return {"success" : False, "message":"Sorry no user found"}
    return resp

@app.route('/police/login',methods=['POST'])
def policeLogin():

    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    userName = raw["userName"]
    password = raw["password"]
    print("here 1")
    print(userName)
    print(password)
    myquery = { "userName" : userName, "password": password}
    print(myquery)
    mydoc = policeColl.find_one(myquery, {"_id":0})
    print(mydoc)
    if mydoc is not None:
        resp = {"success" : True, "message":"success","userObj": mydoc}

```

```

else:
    return {"success" : False, "message":"Sorry no user found"}
return resp

@app.route('/user/search',methods=['POST'])
def search():

    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    mobile = raw["mobile"]
    myquery = { "mobile" : mobile}
    print(myquery)
    mydoc = userColl.find_one(myquery, {"_id":0})
    print(mydoc)
    if mydoc is not None:
        resp = {"success" : True, "message":"success","userObj": mydoc}
    else:
        return {"success" : False, "message":"Sorry no user found"}

    return resp

```

```

@app.route('/admin/user/search',methods=['POST'])
def adminSearched():

    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    mobile = raw["mobile"]
    myquery = { "mobile" : mobile}
    print(myquery)
    mydoc = userColl.find_one(myquery, {"_id":0})
    print(mydoc)
    if mydoc is not None:
        resp = {"success" : True, "message":"success","userObj": mydoc}
    else:
        return {"success" : False, "message":"Sorry no user found"}

    return resp

```

```

@app.route('/user/register',methods=['POST'])
def registernew():
    data = request.get_json()

```

```

print(data)
raw = ast.literal_eval(json.dumps(data))
# gggg=int(datetime.datetime.now().strftime("%s")) * 1000
dataUserID = str(uuid.uuid4())

userColl.insert_one({ "userID": dataUserID,"userName": raw["userName"],"password": raw["password"],"approval": "PENDING", "emailID": raw["emailID"],"mobile": raw["mobile"],"address": raw["address"] , "privatekey": raw["privatekey"] , "accountID": raw["accountID"] })
return {"success" : True, "userID": dataUserID}

@app.route('/admin/register',methods=['POST'])
def adminregister():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
dataUserID = str(uuid.uuid4())

rtoColl.insert_one({ "rtoID": dataUserID,"userName": raw["userName"],"password": raw["password"]})
return {"success" : True, "userID": dataUserID}

@app.route('/police/register',methods=['POST'])
def policeRegister():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
dataUserID = str(uuid.uuid4())

policeColl.insert_one({ "policeID": dataUserID,"userName": raw["userName"],"password": raw["password"]})
return {"success" : True, "userID": dataUserID}

@app.route('/user/raisecomplaint',methods=['POST'])
def userRaiseComplaint():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
dataUserID = str(uuid.uuid4())
complaintColl.insert_one({ "complaintID": dataUserID,"complaintDesc": raw["complaintDesc"]})
return {"success" : True}

@app.route('/complaint/list',methods=['GET'])
def userList():

```

```

mydoc = complaintColl.find()
resp = dumps(mydoc)
return resp

@app.route('/user/userID',methods=['POST'])
def findUserbyUserID():

    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    userID = raw["userID"]
    print("here 1")
    print(userID)
    myquery = { "userID" : userID}
    mydoc = userColl.find_one(myquery, {"_id":0})

    if mydoc is not None:
        resp = { "success" : True, "message":"success","userObj":mydoc}
    else:
        return { "success" : False, "message":"Sorry no user found"}
    return resp

@app.route('/user/getfinedetails',methods=['POST'])
def getFineDetails():

    print("Hello world")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    userID = raw["userID"]

    print("here 1")
    print(userID)

    myquery = { "userID" : userID, "fineStatus": "UNPAID" }
    mydoc = fineColl.find_one(myquery, {"_id":0})

    if mydoc is not None:
        resp = { "success" : True, "message":"success","userObj":mydoc}
    else:
        return { "success" : False, "message":"Sorry no user found"}

    return resp

```

```

@app.route('/user/payfine',methods=['POST'])
def payFine():
    print("*****Payfine*****")
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    myquery = { "userID" : raw["userID"], "fineStatus": "UNPAID" }
    mydoc = fineColl.find_one(myquery, {"_id":0})

    if mydoc is None:
        print("ddds")
        return{ "success" : False, "message":"success","userObj":mydoc}
    #gggg=int(datetime.datetime.now().strftime("%s")) * 1000
    web3.eth.defaultAccount = web3.eth.accounts[0]

dataUserID = str(uuid.uuid4())
contract = web3.eth.contract(address=address, abi=abi)
contractID = contract.functions.greet().call()
print(contractID)
cocontractDoc = cocontractColl.find().sort([('_id', -1)]).limit(1)
print("hhhh")
dbContractID =""
for doc2 in cocontractDoc:
    print(doc2)
dbContractID = doc2.get('contractID')
print(dbContractID)
print("kkkk")

if dbContractID == contractID:
    print("correct")
tx_hash = contract.functions.setGreeting(dataUserID).transact()
cocontractColl.insert_one({"contractID":dataUserID})
    #return { "success" : True}
else:
    print("incorrect")
    #return { "success" : False}
nonce = web3.eth.getTransactionCount(raw["accountID"])
print("fdsfsfsdfsfdsfsfsfsd")
print(raw["fineAmount"])
tx = {'nonce': nonce,'to': account_2,'value': web3.toWei(raw["fineAmount"], 'ether'),'gas': 2000000,'gasPrice': web3.toWei('50', 'gwei'), }

signed_tx =web3.eth.account.signTransaction(tx, raw["privatekey"])

```

```

print(web3.toHex(tx_hash))

# fineColl.insert_one({ "userID": raw["userID"], "fineID": dataFineID, "userName": raw["userName"],
"emailID": raw["emailID"], "mobile": raw["mobile"], "address": raw["address"] , "accountID":
raw["accountID"] , "fineAmount": raw["fineAmount"] , "dataTime": gggg})
myquery = { "userID": raw["userID"] , "fineStatus": "UNPAID"}

print("raw")
print(myquery)
print(raw["userID"])
print("rawclose")
newvalues = { "$set": { "fineStatus": "PAID" } }
fineColl.update_one(myquery, newvalues)
return { "success" : True}

@app.route('/register/contract',methods=['POST'])
def registerContract():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    # gggg=int(datetime.datetime.now().strftime("%s")) * 1000
    dataUserID = str(uuid.uuid4())
    contract = web3.eth.contract(address=address, abi=abi)
    contractID = contract.functions.greet().call()

    print(contractID)

cocontractDoc = cocontractColl.find().sort([('_id', -1)]).limit(1)
    print("hhhh")

dbContractID = ""

for doc2 in cocontractDoc:
    print(doc2)
    dbContractID = doc2.get('contractID')
    print(dbContractID)
    print("kkkk")
    if dbContractID == contractID:
        print("correct")
    tx_hash = contract.functions.setGreeting(dataUserID).transact()
    cocontractColl.insert_one({ "contractID":dataUserID })
    return { "success" : True}
else:
    print("incorrect")
    return { "success" : False}

```

```

# json_docs.append(json_doc)

docs_list = list(coontractDoc)
# userColl.insert_one({ "userID": dataUserID,"userName": raw["userName"],"password": raw["password"], "emailID": raw["emailID"], "mobile": raw["mobile"], "address": raw["address"] , "private_key": raw["private_key"] , "accountID": raw["accountID"] , "dataTime": gggg})
return { "success" : True, "userID": "kk" }

@app.route('/create/fine',methods=['POST'])
def createFine():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    # gggg=int(datetime.datetime.now().strftime("%s")) * 1000
    dataFineID = str(uuid.uuid4())

    fineColl.insert_one({ "userID": raw["userID"], "fineStatus": "UNPAID", "fineID": dataFineID, "userName": raw["userName"], "fineReason": raw["fineReason"], "emailID": raw["emailID"], "mobile": raw["mobile"], "address": raw["address"] , "accountID": raw["accountID"] , "fineAmount": raw["fineAmount"] })
    fromaddr = 'shivakumarshivu1212@gmail.com'
    toaddrs = 'shivakumarshivu1212@gmail.com'
    b = int(data["fineAmount"])
    b=b*100
    string = ""
    string = str(b)
    var = "Fine Has been registered due to Traffic Rules Violation\nUser name: "+ data["userName"]+ "\n"+ "Fine Reason: "+data["fineReason"] + "\n"+ "FineAmount:" + string
    msg = var
    username = 'shivakumarshivu1212@gmail.com'
    password = 'thankstogmail'

    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login(username, password)
    server.sendmail(fromaddr, toaddrs, msg)
    server.quit()

    return { "success" : True, "userID": dataFineID}

""""@app.route('/create/fine',methods=['POST'])
def createFine():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    # gggg=int(datetime.datetime.now().strftime("%s")) * 1000

```

```

dataFineID = str(uuid.uuid4())

fineColl.insert_one({ "userID": raw["userID"], "fineStatus": "UNPAID", "fineID": dataFineID, "userName": raw["userName"], "fineReason": raw["fineReason"], "emailID": raw["emailID"], "mobile": raw["mobile"], "address": raw["address"], "accountID": raw["accountID"], "fineAmount": raw["fineAmount"] })
fromaddr = 'shivakumarshivu1212@gmail.com'
toaddrs = 'shivakumarshivu1212@gmail.com'
msg = 'Fine created'
username = 'shivakumarshivu1212@gmail.com'
password = 'thankstogmail'
server = smtplib.SMTP('smtp.gmail.com', 587)
server.ehlo()
server.starttls()
server.login(username, password)
server.sendmail(fromaddr, toaddrs, msg)
server.quit()
return {"success": True, "userID": dataFineID}"""

```

```

@app.route('/user/listfine', methods=['GET'])
def listFine():
    mydoc = fineColl.find()
    resp = dumps(mydoc)
    return resp

```

```

@app.route('/uploader', methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        f.save(secure_filename(f.filename))
        return 'file uploaded successfully'

```

```

@app.route('/user/admin/approve', methods=['POST'])
def adminApproveUser():
    data = request.get_json()
    print(data)
    raw = ast.literal_eval(json.dumps(data))
    myquery = { "userID": raw["userID"] }
    #before inserting check user is already registered or not
    newvalues = { "$set": { "approval": "ADMINAPPROVED" } }
    userColl.update_one(myquery, newvalues)
    return {"success": True}

```

```

@app.route('/user/addvechicle', methods=['POST'])
def addvechicle():

```

```

data = request.get_json()
# print(data)
raw = ast.literal_eval(json.dumps(data))
Number = raw["Number"]
#print(Number)
#dataUserID = str(uuid.uuid4())
vechileColl.insert_one({ "Number": raw["Number"], "Name": raw["Name"], "Gmail": raw["Gmail"], "Rcnumber": raw["Rcnumber"], "Insurance": raw["Insurance"], "Emession": raw["Emession"], "Gmail": raw["Gmail"], "phone": raw["phone"] })
mydoc = [i for i in vechileColl.find({ "Number": Number })]
resp = dumps(mydoc)
a=(raw["Insurance"])
b=a.split("-")
x=int(b[0])
y=int(b[1])
z=int(b[2])
res = x+y+z
print(res)
return { "success" : True, "Number": Number }
@app.route('/user/vechicleview',methods=['POST'])
def vechicleview():
    data = request.get_json()
    # print(data)
    raw = ast.literal_eval(json.dumps(data))
    Number = raw["Number"]
    #print(Number)
    dataUserID = str(uuid.uuid4())

    #vechileColl.insert_one({ "userID": dataUserID, "Number": raw["Number"], "Name": raw["Name"], "Gmail": raw["Gmail"], "Rcnumber": raw["Rcnumber"], "Insurance": raw["Insurance"], "Emession": raw["Emession"], "Gmail": raw["Gmail"] })

    mydoc = [i for i in vechileColl.find({ "Number": Number })]
    resp = dumps(mydoc)
    print("*****")
    return resp

@app.route('/user/checkvalidaty',methods=['POST'])
def checkvalidaty():
    mydoc = vechileColl.find()
    resp = dumps(mydoc)
    res = json.loads(resp)
    print(type(res))
    #print(res[1])
    return resp
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=1111, threaded=True)

```

## Greeter.sol

```
pragma solidity^0.4.21;
contract Greeter
{
    string public greeting;
    function Greeter() public
    {
        greeting = 'Hello';
    }
    function setGreeting(string _greeting) public
    {
        greeting = _greeting;
    }
    function greet() view public returns(string)
    {
        return greeting;
    }
}
```

## Main.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link href='http://fonts.googleapis.com/css?family=Parisienne' rel='stylesheet' type='text/css'>
<!-- CSS Files -->
<link rel="stylesheet" type="text/css" media="screen" href="css/style.css">
<link rel="stylesheet" type="text/css" media="screen" href="menu/css/simple_menu.css">
<!-- Contact Form -->
<link href="contact-form/css/style.css" media="screen" rel="stylesheet" type="text/css">
<link href="contact-form/css/uniform.css" media="screen" rel="stylesheet" type="text/css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1; }

input[type=text], input[type=password] {
    width: 50%;
    padding: 12px 20px;
    margin: 8px 0;
    align: center;
```

```
vertical-align: center;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}

button {
background-color: #04AA6D;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 50%;
}
button:hover {
opacity: 0.8;
}
.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}
img.avatar {
width: 20%;
border-radius: 50%;
}
.container {text-align: center;
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}
.cancelbtn {
width: 100%;
}
}
</style>
```

```

</head>
</style>
</head>
<body>
<div class="w3-container">
<marquee bgcolor="yellow"> <h2><b><i>RTO VEHICLE MANAGEMENT
SYSTEM</i></b></h2></marquee>
</div>
<div class="w3-content w3-section" style="max-width:500px">




</div>
<script>
var myIndex = 0;
carousel();
function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 2500);
}
</script>
<h2 class="w3-center" align="center">Welcome to RTO</h2>
<div class="imgcontainer">
</div>
<div class="container">
<a href="login.html"> <button type="submit">User Login</button></a>
<a href="adminlogin.html"> <button type="submit">Admin Login</button></a>
<a href="policelogin.html"> <button type="submit">Police Login</button></a>
</form>
</body>
</html>

```

## Login.html

```
<!doctype html>
<html>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<head>
  <title>RTO</title>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="main.html">RTO</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="register.html">Register</a></li>

      <li><a href="https://transport.karnataka.gov.in/english">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </div>
</nav>
<div class="container">
<center>
<h1>USER LOGIN</h1></center>

<form name="myform">
  <div class="form-group">
    <label for="usr">Name:</label>
    <input type="text" id="userName" class="form-control" id="usr" name="userName">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" id="password" class="form-control" id="pwd" name="password">
  </div>

</form>
  <div><button type="button" onclick="ONSubmitButton()" class="btn btn-primary btn-block" id="startbutton">LOGIN</button></div>

</div>
<script>
```

```

function ONSubmitButton()
{
var x = document.forms["myform"]["name"].value;
var name=document.myform.name.value;
if (!/^a-zA-Z]*$/g.test(document.myform.name.value)||x==" ")
{
    alert("Enter Valid user Name");
    document.myform.userName.focus();
    return false;
}
else if(password.length<6)
{
    document.myform.password.focus();
    alert("Password must be at least 6 characters long.");

    return false;
}
if( $("#userName").val().length == 0 || $("#password").val().length == 0 )
{
    alert("Empty")
}
else{
    var data = {
        "userName": $("#userName").val(),
        "password": $("#password").val()

    $.ajax({
        url : 'http://127.0.0.1:1111/user/login',
        type: "POST",
        dataType: "json",
        data: JSON.stringify (data),
        contentType: 'application/json',
        cache: false,
        timeout: 5000,
        success: function(data) {
            if(data.success == true)
            {
                console.log(data)
                console.log(data.userObj.accountID);
                localStorage.setItem("accountID", data.userObj.accountID);
                localStorage.setItem("address", data.userObj.address);
                localStorage.setItem("emailID", data.userObj.emailID);
                localStorage.setItem("mobile", data.userObj.mobile);
                localStorage.setItem("password", data.userObj.password);
                localStorage.setItem("privatekey",data.userObj.privatekey);
                localStorage.setItem("userID", data.userObj.userID);
                localStorage.setItem("userName", data.userObj.userName);
                localStorage.setItem("rcDoc", data.userObj.rcDoc);
            }
        }
    })
}
}

```

```
localStorage.setItem("dlDoc", data.userObj.dlDoc);
localStorage.setItem("insuranceDoc", data.userObj.insuranceDoc);
localStorage.setItem("emmissonDoc", data.userObj.emmissonDoc);

console.log("eeeeee")
console.log(localStorage.getItem("accountID"))
console.log("ffffff")
alert("success")
window.location.href = 'usermain.html';
    }
else{
    alert("Admin has not yet approved the User or no User Found")
}
    }
});
}
}
</script>
</body>

</html>
```

## AdminLogin.html

```
<!doctype html>
<html>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<head>
  <style>
  </style>
  <title>RTO</title>
</head>

<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="main.html">RTO</a>
    </div>
    <ul class="nav navbar-nav">
      <li><a href="https://transport.karnataka.gov.in/english">About</a></li>
      <li><a href="https://transport.karnataka.gov.in">Contact</a></li>
    </ul>
  </div>
</nav>
  <div class="container">
<center>
<h1>RTO LOGIN</h1></center>
<form>
  <div class="form-group">
    <label for="usr">Name:</label>
    <input type="text" id="userName" class="form-control" id="usr">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" id="password" class="form-control" id="pwd">
  </div>

</form>
  <div><button type="button" onclick="ONSubmitButton()" class="btn btn-primary btn-block" id="startbutton">LOGIN</button></div>
</div>
```

```

<script>

function ONSubmitButton()
{
if( $("#userName").val().length == 0 || $("#password").val().length == 0 )
{
alert("Empty")
}
else{
    var data = {
    "userName": $("#userName").val(),
    "password": $("#password").val(),}

$.ajax({
url : 'http://127.0.0.1:1111/rto/login',
type: "POST",
dataType: "json",
data: JSON.stringify (data),
contentType: 'application/json',

cache: false,
timeout: 5000,

success: function(data) {
    if(data.success == true)
    {
        alert("success")
        window.location.href = 'adminDashboard.html';
    }
}
else{
    alert("Error")
}
})
}
}

</script>
</body>

</html>

```

## PoliceLogin.html

```
<!doctype html>
<html>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<head>
  <title>RTO</title>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="main.html">RTO</a>
    </div>
    <ul class="nav navbar-nav">
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </div>
</nav>
  <div class="container">
<center>
<h1>POLICE LOGIN</h1></center>
<form>
  <div class="form-group">
    <label for="usr">Name:</label>
    <input type="text" id="userName" class="form-control" id="usr">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" id="password" class="form-control" id="pwd">
  </div>

</form>
  <div><button type="button" onclick="ONSubmitButton()" class="btn btn-primary btn-block" id="startbutton">LOGIN</button></div>
</div>
  <script>
function ONSubmitButton()
{
  if( $("#userName").val().length == 0 || $("#password").val().length == 0 )
  {
```

```
        alert("Empty")
    }

else{

var data = {
    "userName": $("#userName").val(),
    "password": $("#password").val(),
}

$.ajax({
    url : 'http://127.0.0.1:1111/police/login',
    type: "POST",
    dataType: "json",
    data: JSON.stringify (data),
    contentType: 'application/json',

    cache: false,
    timeout: 5000,

    success: function(data) {
        if(data.success == true)
        {
            alert("success")
            window.location.href = 'view_vehcile_details.html';
        }
        else{
            alert("Error")
        }
    });
}
}

</script>
</body>

</html>
```

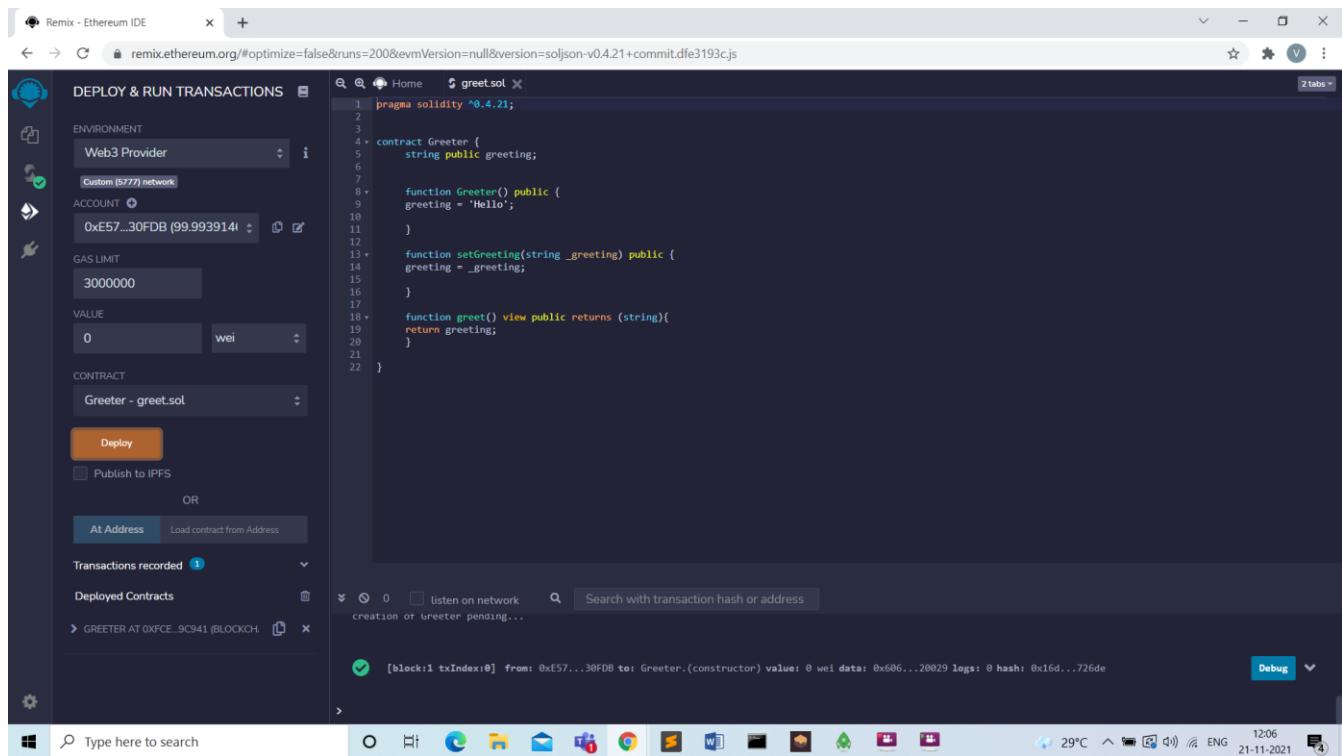
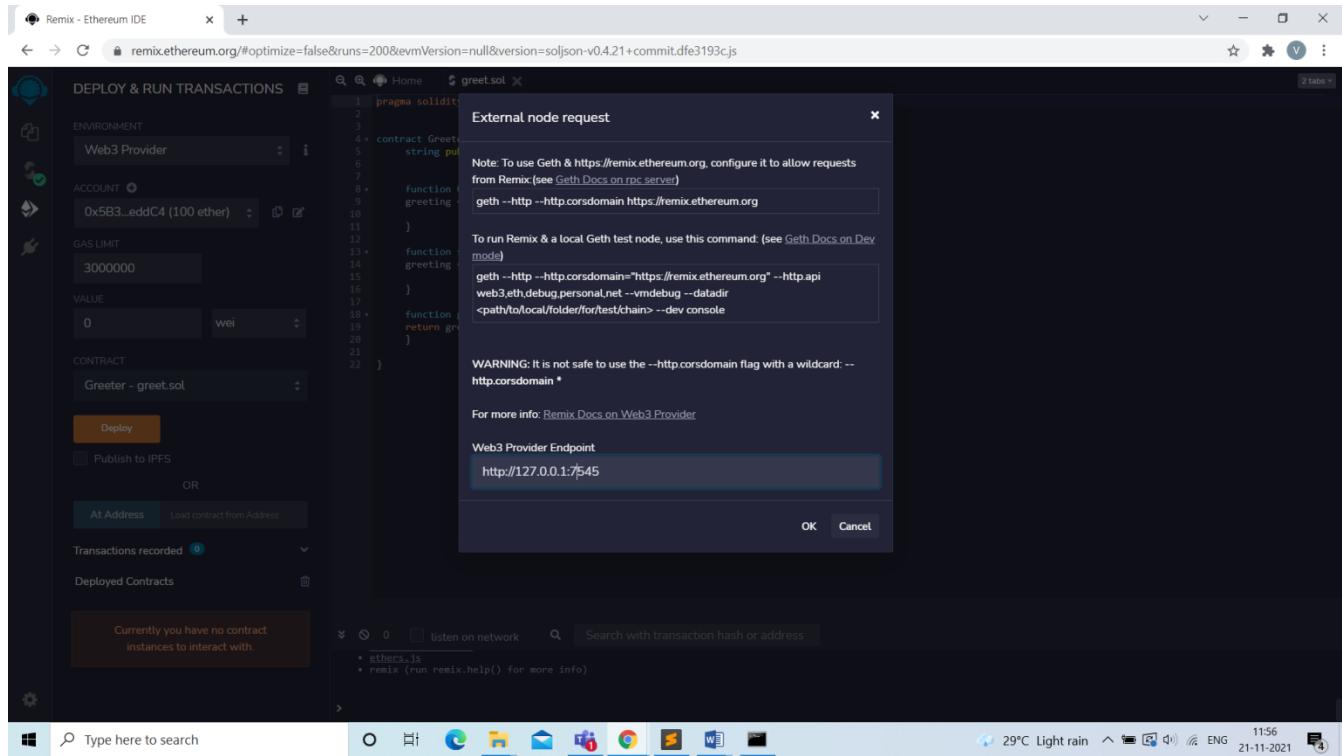
## SearchUser.html

```
<!doctype html>
<html>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<head>
  <title>RTO</title>
</head>

<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="main.html">RTO</a>
    </div>
    <ul class="nav navbar-nav">
      <li><a href="https://transport.karnataka.gov.in/english">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </div>
</nav>
  <div class="container">
<center>
<h1>SEARCH USER</h1></center>

<form>
  <div class="form-group">
    <label for="usr">SEARCH USER By Mobile Number:</label>
    <input type="text" id="mobile" class="form-control" id="usr">
  </div>
</form>
```

# CHAPTER 8: SNAPSHOTS



The screenshot shows the Remix Ethereum IDE interface. On the left, there's a sidebar for "DEPLOY & RUN TRANSACTIONS" with fields for "ENVIRONMENT" (Web3 Provider: Custom (5777) network), "ACCOUNT" (0xE57...30FDB (99.9)), "GAS LIMIT" (3000000), and "VALUE" (0). Below these are sections for "CONTRACT" (Greeter - greet.sol) and deployment options ("Deploy", "Publish to IPFS", "At Address"). A "Transactions recorded" section shows one entry: "GREETER AT 0xFCE...9C9". The main tab is titled "TRANSACTIONS" and displays a single transaction record:

TX HASH	0x37d33bccb1ce9cad331a8ff1abb6719aa70c7ce57226e48684b3e8bedb4cef7
FROM ADDRESS	0xE57C9136CD8e67d3E05E8C1035fFB5AeC2E30FDB
CREATED CONTRACT ADDRESS	0xFCEAD08A964586297eE7224f254590DEdD89c941
GAS USED	304267
VALUE	0

On the right, there's a "CONTRACT CREATION" button and a "Debug" dropdown. The status bar at the bottom shows system information: 29°C, ENG, 12:08, 21-11-2021.

This screenshot shows the same Remix interface but with a different focus. The "TRANSACTIONS" tab is still visible, but the main content area is now displaying a list of accounts under the "ACCOUNTS" tab. The sidebar remains the same, showing the deployment details for the Greeter contract.

MNEMONIC	enter nose purse aware rain liar swim artefact tonight hello lazy once	HD PATH	m/44'/60'/0'/0/account_index				
ADDRESS	0xE57C9136CD8e67d3E05E8C1035fFB5AeC2E30FDB	BALANCE	99.99 ETH	TX COUNT	1	INDEX	0
ADDRESS	0xB41208385b693Fa56fbE7F0A0eD7c11E0EEeC301	BALANCE	100.00 ETH	TX COUNT	0	INDEX	1
ADDRESS	0xe72d4C5d7e0FCa82b6d2BE54B910b5Ce7726f812	BALANCE	100.00 ETH	TX COUNT	0	INDEX	2
ADDRESS	0x9D92e8988C907560Ba153Ad75fA6b055478EB752	BALANCE	100.00 ETH	TX COUNT	0	INDEX	3
ADDRESS	0xcf2961B840aa67D96f5fdaA93302F180a1Ddf7b0	BALANCE	100.00 ETH	TX COUNT	0	INDEX	4
ADDRESS	0xe5C6777B0ffFe1d1DBDdFE9069672a982a307e109	BALANCE	100.00 ETH	TX COUNT	0	INDEX	5
ADDRESS	0xF7db0785D9D289A4dbDFF21C199fa2b3C89f170	BALANCE	100.00 ETH	TX COUNT	0	INDEX	6

The status bar at the bottom shows system information: 29°C, ENG, 12:08, 21-11-2021.

C:\Users\hp\Desktop\cryptonode\mycontract.py • (cryptonode) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
1 // app.py
2 <> approveUser.html
3 <> checkValidity.html
4 <> checkValidity.html.html
5 /* contract.py
6 <> createfine.html
7 <> date.py
8 greet.sol
9 <> images.html
10 latest.py
11 <> listcomplaint.html
12 <> login.html
13 logo.png
14 /* ma.py
15 <> main.html
16 /* mycontract.py
17 <> payfine.html
18 <> policeLogin.html
19 <> policeViewFine.html
20 <> register.html
21 <> registerAdmin.html
22 <> registerPolice.html
23 rto.png
24 rto1.png
25 rto2
26 rto3.png
27 rto4.png
28 <> SearchUser.html
29 <> test.html
30 /* test.py
31 <> untitled.py
32 <> userdetails.html
33 <> usermain.html
34 <> UserraiseComplaint.html
35 <> vehicle_details.html
36 <> view_vehicle_details.html
37 <> view_vehicle_details1.html
```

che checkValidity.html.html x | contract.py x | createfine.html x | date.py x | greet.sol x | images.html x | latest.py x | listcomplaint.html x | login.html x | ma.py x mycontract.py +

1 import json
2 from web3 import Web3
3
4 # [{"constant":false,"inputs":[{"name":"\_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable"}]
5
6 # Set up web3 connection with Ganache
7 ganache\_url = "http://127.0.0.1:7545"
8 web3 = Web3(HTTPProvider(ganache\_url))
9
10 web3.eth.defaultAccount = web3.eth.accounts[0]
11
12 abi = json.loads('[{"constant":false,"inputs":[{"name":"\_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable"}]')
13
14 address = web3.toChecksumAddress("0x1CEAD08A964586297e7224f254590DEdD09c941")
15
16
17 contract = web3.eth.contract(address=address, abi=abi)
18
19
20 print(contract.functions.greet().call())
21
22
23 tx\_hash = contract.functions.setGreeting('contactOne').transact()
24
25
26 #print(tx\_hash)
27
28 web3.eth.waitForTransactionReceipt(tx\_hash)
29
30
31 print('updated greeting: {}'.format(
32 contract.functions.greet().call()
33 ))

C:\Users\hp\Desktop\crytonode\mycontract.py (crytonode) - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
/* app.py
<> approveUser.html
<> checkValidity.html
<> checkValidity.html.html
/* contract.py
<> createfine.html
/* date.py
<greet.sol
<> images.html
/* latest.py
<> listcomplaint.html
<> login.html

/* mycontract.py
<> payfine.html
<> policelogin.html
<> policeviewfine.html
<> register.html
<> registeradmin.html
<> registerpolice.html





<> SearchUser.html
<> test.html
/* test.py
/* untitled.py
<> userdetails.html
<> usermain.html
<> UserraiseComplaint.html
<> vehicle_details.html
<> view_vehicle_details.html
<> view_vehicle_details1.html

1 import json
2 from web3 import Web3
3
4 [{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable"}]
5
6 [ex] Command Prompt
7 Microsoft Windows [Version 10.0.19043.1348]
8 Copyright © Microsoft Corporation. All rights reserved.
10 W C:\Users\hp>cd desktop
11
12 aC:\Users\hp\Desktop>cd crytonode
13
14 ac:Users\hp\Desktop\crytonode>python app.py
15 * Serving Flask app "app" (lazy loading)
16 * Environment: production
17 ** WARNING: This is a development server. Do not use it in a production deployment.
18 ** Use a production WSGI server instead.
19 * Debug mode: off
20 * Running on all addresses...
21 * WARNING: This is a development server. Do not use it in a production deployment.
22
23 tC:\Users\hp\Desktop\crytonode>python mycontract.py
24 python: can't open file 'C:\\\\users\\\\hp\\\\Desktop\\\\crytonode\\\\mycontract.py': [Errno 2] No such file or directory
25
26 #C:\Users\hp\Desktop\crytonode>python mycontract.py
27 Hello
28 updated greeting: contactOne
29
30 C:\Users\hp\Desktop\crytonode>
31
32 c
33

```

42 characters selected

Type here to search

Tab Size: 4 Python

30°C 12:14 21-11-2021

C:\Users\hp\Desktop\crytonode\mycontract.py (crytonode) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ganache

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES				
CURRENT BLOCK 2	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MUIRGLEACIER	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	WORKSPACE QUICKSTART	SAVE	SWITCH	⚙️

**TX HASH**  
**0x79d6cb3b6cef899f395a5522537d3909b4deef74f4ced802966911708836d983**

FROM ADDRESS  
0xE57C9136CD8e67d3E05E8C1035FFB5AeC2E30FDB

TO CONTRACT ADDRESS  
0xFCEAD08A964586297eE7224f254590DEdD09c941

GAS USED  
28920

VALUE  
0

**TX HASH**  
**0x37d33bccb1ce9cad331a8ff1abb67119aa70c7ce57226e48684b3e8bedb4cef7**

FROM ADDRESS  
0xE57C9136CD8e67d3E05E8C1035FFB5AeC2E30FDB

CREATED CONTRACT ADDRESS  
0xFCEAD08A964586297eE7224f254590DEdD09c941

GAS USED  
304267

VALUE  
0

Size: 4 Python

30°C 12:15 21-11-2021

The screenshot shows a Sublime Text window with the following details:

- File Explorer:** On the left, it lists files and folders related to a project named "mycontract". The "mycontract.py" file is currently selected.
- Ganache Interface:** The main window displays the Ganache interface. Key information shown includes:
  - CURRENT BLOCK:** 2
  - GAS PRICE:** 20000000000
  - GAS LIMIT:** 6721975
  - HARDFORK:** MUIRGLACIER
  - NETWORK ID:** 5777
  - RPC SERVER:** HTTP://127.0.0.1:7545
  - MINING STATUS:** AUTOMINING
- Transactions Tab:** Shows a single transaction record:
  - TX:** 0x79d6cb3b6cef899f395a5522537d3909b4deef74f4ced802966911708836d983
  - SENDER ADDRESS:** 0xE57C9136CD8e67d3E05E8C1035fFB5AeC2E30FDB
  - TO CONTRACT ADDRESS:** 0xFCEAD08A964586297eE7224f254590DEdD09c941
  - CONTRACT CALL:** (button)
  - Value:** 0.00 ETH
  - GAS USED:** 28920
  - GAS PRICE:** 20000000000
  - GAS LIMIT:** 130322
  - MINED IN BLOCK:** 2
- Events Tab:** Shows the message "NO EVENTS".
- System Tray:** At the bottom right, it shows the date (21-11-2021), time (12:15), and system status (30°C, ENG).

C:\Users\hp\Desktop\cryptonode\app.py • (cryptonode) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- cryptonode
  - adminDashboard.html
  - adminlogin.html
  - app.py
  - approveUser.html
  - checkValidity.html
  - checkValidity.html
  - contract.py
  - createfine.html
  - date.py
  - greet.sol
  - images.html
  - latest.py
  - listcomplaint.html
  - login.html
  - logo.png
  - map.py
  - main.html
  - mycontract.py
  - payfine.html
  - policelogin.html
  - policeviewfine.html
  - register.html
  - registeradmin.html
  - registerpolice.html
  - rto.png
  - rto1.png
  - rto2
  - rto3.png
  - rto4.png
  - SearchUser.html
  - test.html
  - test.py
  - untitled.py
  - userdetails.html
  - username.html
  - usercomplaint.html

adminDashboard.html app.py greet.sol mycontract.py

```
21 from web3 import Web3
22 import smtplib
23
24 from flask_cors import CORS
25
26 ganache_url = "http://127.0.0.1:7545"
27 web3 = Web3(HTTPProvider(ganache_url))
28
29
30 web3.eth.defaultAccount = web3.eth.accounts[0]
31
32 abi = json.loads('["{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreeting","outputs":[],"payable":false,"stateMutability":"nonpayable"}]')
33
34 #var greeterContract = new web3.eth.Contract([{"constant":false,"inputs":[{"name":"_greeting","type":"string"}],"name":"setGreetin
35
36
37 address = web3.toChecksumAddress("0xFCEAD08A964586297eF7224f254590fEd09c941")
38
39
40
41 account_2 = '0xC1A6e970a3B561F7d2a88d7c0001E36ea5dC08F2' # Fill me in
42
43
44 app = Flask(__name__)
45 CORS(app)
46
47
48 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
49
50 mydb = myclient["BlockChain"]
51
52 userColl = mydb["User"]
53
54 rtoColl = mydb["RTO"]
55
56 policeColl = mydb["Police"]
57
58 fineColl = mydb["Fine"]
59
60 contractColl = mydb["Contract"]
```

Spaces: 4      30°C      ENG      12:17      21-11-2021

C:\Users\hp\Desktop\crytonode\app.py (crytonode) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- crytonode
  - adminDashboard.html
  - adminlogin.html
  - /\* app.py
  - <> approveUser.html
  - <> checkValidity.html
  - <> checkValidity.html
  - /\* contract.py
  - <> createfine.html
  - /\* date.py
  - greet.sol
  - <> images.html
  - /\* latest.py
  - <> listcomplaint.html
  - <> login.html
  - logo.png
  - ma.py
  - <> main.html
  - /\* mycontract.py
  - <> payfine.html
  - <> policeLogin.html
  - <> policeviewfine.html
  - <> register.html
  - <> registerAdmin.html
  - <> registerPolice.html
  - rto.png
  - rto1.png
  - rto2
  - rto3.png
  - rto4.png
  - <> SearchUser.html
  - <> test.html
  - /\* test.py
  - <> userdetails.html
  - <> usermain.html

42 characters selected

adminDashboard.html app.py greet.sol mycontract.py

```

21 from web3 import Web3
22 import smtplib
23
24 from flask_cors import CORS
25
26
27 # Command Prompt - python app.py
28
29 # WARNING: This is a development server. Do not use it in a production deployment.
30 # Use a production WSGI server instead.
31 # Debug mode: off
32 # Running on all addresses.
33 # WARNING: This is a development server. Do not use it in a production deployment.
34 # Running on http://192.168.0.116:1111/ (Press CTRL+C to quit)
35
36 C:\Users\hp\Desktop\crytonode>python mycontract.py
37 C:\Users\hp\Desktop\crytonode>python mycontract.py
38 Hello
39 updated greeting: contactOne
40
41 C:\Users\hp\Desktop\crytonode>python app.py
42 * Serving Flask app 'app' (lazy loading)
43 * Environment: production
44 # WARNING: This is a development server. Do not use it in a production deployment.
45 # Use a production WSGI server instead.
46 # Debug mode: off
47 # Running on all addresses.
48 # WARNING: This is a development server. Do not use it in a production deployment.
49 # Running on http://192.168.0.116:1111/ (Press CTRL+C to quit)
50
51
52
53
54
55
56 policeColl = mydb["Police"]
57
58 fineColl = mydb["Fine"]
59
60 contractColl = mydb["Contract"]

```

Spaces: 4 Python

Type here to search

30°C 12:18 ENG 21-11-2021

Report.pdf main.html

RTO VEHICLE MANAGEMENT SYSTEM

Government of Andhra Pradesh Transport Department

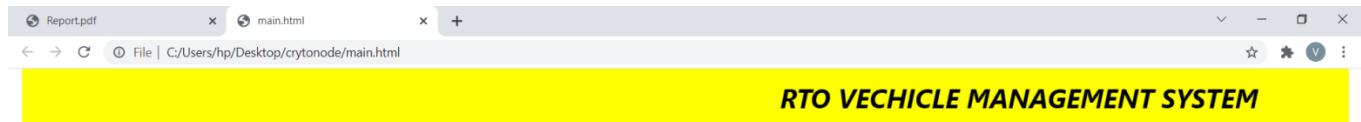
Welcome to RTO

User Login

Admin Login

Police Login

30°C 12:20 ENG 21-11-2021

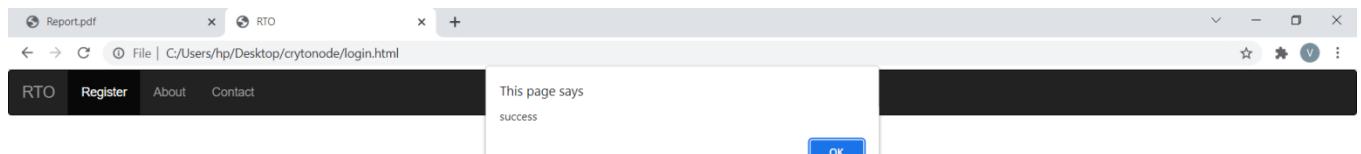


Welcome to RTO

User Login

Admin Login

Police Login



Report.pdf view\_vehcile\_details1.html

RTO About Contact

Government of Andhra Pradesh  
Transport Department

AP-05-6442

Search

Vechicle Details

Vechicle Number	Owner Name	RC Number	Insurance	Emission	Phone Number
AP-05-6442	Chandrika	AC45678	2021-11-17	2021-12-30	8466068349

Type here to search

30°C 12:22 21-11-2021

Report.pdf RTO

RTO About Contact

Government of Andhra Pradesh  
Transport Department

Complaint:

I am Chandrika from Vijayawada, due to miss communication the machavaram police M.Ravi Traffic police doesn't have good communication with the citizens.

REGISTER Complaint

Type here to search

30°C 12:24 21-11-2021

This page says  
success

OK

Transport Department

Complaint:

I am Chanrika from Vijayawada, due to miss communication the machavaram police M.Ravi Traffic police doesn't have good communication with the citizens.

REGISTER Complaint



This page says  
No fine

OK

Transport Department

PAY FINE

No Data

USER New PAY FINE

USER DETAILS



Report.pdf RTO

← → C File | C:/Users/hp/Desktop/crytonode/userdetails.html

RTO Register Login Raise Complaint About Contact



PAY FINE

User Name:Yashitha

Address:abc

Mobile:7382244899

Email:vilasagarapuyashitha.24@gmail.com

Account ID:HIJK9876A

Private Key:abcde

Type here to search

Report.pdf RTO

← → C File | C:/Users/hp/Desktop/crytonode/adminlogin.html

RTO About Contact



## RTO LOGIN

Name:

Yashitha

Password:

\*\*\*\*\*

OK

LOGIN

Type here to search

The screenshot shows a web browser window with the following details:

- Title Bar:** Report.pdf, RTO, File | C:/Users/hp/Desktop/crytonode/adminDashboard.html
- Header:** RTO, About, Contact
- Logo:** Government of Andhra Pradesh Transport Department
- Section:** SEARCH USER
- Text:** SEARCH USER By Mobile Number: [Input Field]
- Buttons:** SEARCH USER, ADD POLICE, VIEW COMPLAINTS, Add VEHICLE DETAILS, LOG OUT

The screenshot shows a web browser window with the following details:

- Title Bar:** Report.pdf, RTO, File | C:/Users/hp/Desktop/crytonode/adminDashboard.html
- Header:** RTO, About, Contact
- Message Box:** This page says  
Found User  
OK
- Logo:** Transport Department
- Section:** SEARCH USER
- Text:** SEARCH USER By Mobile Number: 7382244899
- Buttons:** SEARCH USER, ADD POLICE, VIEW COMPLAINTS, Add VEHICLE DETAILS, LOG OUT

Report.pdf RTO

← → C File | C:/Users/hp/Desktop/crytonode/approveUser.html

RTO About Contact

 Government of Andhra Pradesh  
Transport Department

USER DETAILS

User Name: Yashitha

Address: abc

Mobile: 7382244899

Approve User

Type here to search

30°C 12:27 21-11-2021 ENG

Report.pdf Vechicle\_details

← → C File | C:/Users/hp/Desktop/crytonode/vechicle\_details.html

RTO About Contact

 Government of Andhra Pradesh  
Transport Department

Vechicle Number:  
AP-06-2445

Owner Name:  
Akhil

RC Number:  
BC4567

Insurance valid Up To:  
21-11-2021

Emission valid Up to:  
31-12-2021

Gmail:  
180030029cse@gmail.com

Phone Numner:  
9346906256

REGISTER

Type here to search

30°C 12:29 21-11-2021 ENG

The screenshot shows a web browser window with two tabs open: "Report.pdf" and "listcomplaint.html". The main content area displays the "Government of Andhra Pradesh Transport Department" logo and the heading "Complaint List". A green header bar contains the text "Complaint Detail". Below it, a message reads: "I am Chanrika from Vijayawada, due to miss communication the machavaram police M.Ravi Traffic police doesn't have good communication with the citizens."

The screenshot shows a web browser window with two tabs open: "Report.pdf" and "RTO". The main content area displays the "Transport Department" logo and the heading "POLICE LOGIN". It includes fields for "Name" (Yashitha) and "Password" (\*\*\*\*\*), and a "LOGIN" button. A modal dialog box is overlaid on the page, stating "This page says success" with an "OK" button.

Report.pdf view\_vehcile\_details.html

RTO About Contact



AP-06-2445

SEARCH

CREATE FINE

LOG OUT

## Vechicle Details

Vechicle Number	Owner Name	RC Number	Insurance	Emession	Phone Number
AP-06-2445	Akhil	BC4567	2021-11-21	2021-12-31	9346906256

Type here to search 30°C 12:31 ENG 21-11-2021

Report.pdf Vehcile\_details

RTO About Contact

This page says  
success

OK

Vechicle Number:

AP-09-4567

Owner Name:

Yaswanth

RC Number:

DF6789

Insurance valid Up To:

25-11-2021

Emession valid Up to:

31-12-2021

Gmail:

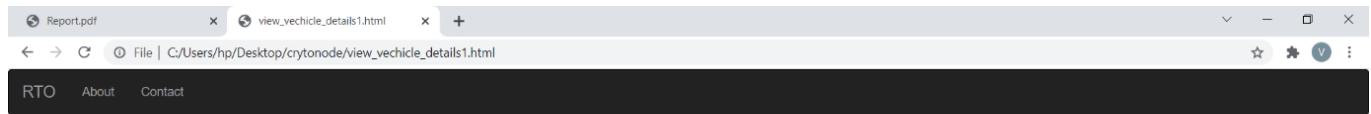
180031097cse@gmail.com

Phone Numner:

7989595459

REGISTER

Type here to search 30°C 12:37 21-11-2021



AP-09-4567

Search

### Vechicle Details

Vechicle Number	Owner Name	RC Number	Insurance	Emission	Phone Number
AP-09-4567	Yaswanth	DF6789	2021-11-25	2021-12-31	7989595459



The screenshot shows the Robo 3T application interface for managing MongoDB databases. On the left, a sidebar lists connections and database structures. The main area shows a connection to 'localhost:27017' with the 'Complaint' collection selected. One document is displayed in the details pane:

Key	Value	Type
{ 1 }	ObjectId("6199ed39e0bcd188f6e08aa") ↳ _id ↳ complaintID ↳ complaintDesc	Object Object String String

At the bottom, a Windows taskbar is visible with the search bar, pinned icons for File Explorer, Mail, and Edge, and system status indicators for weather (30°C), date (21-11-2021), and time (12:32).

Robo 3T - 1.4

File View Options Window Help

New Connection (4)

- System
  - > admin
  - > local
- BlockChain
  - > Complaint
  - > Fine
  - > Police
  - > RTO
  - > User
  - > Vechile
  - > Functions
  - > Users
- > config

localhost:27017 BlockChain

db.getCollection('Fine').find({})

Key	Value	Type
(1) ObjectId("6193ffc4e0c43bc0de8ade47")	{ 11 fields }	Object
"_id"	ObjectId("6193ffc4e0c43bc0de8ade47")	ObjectID
"userID"	544ac1ce-eeff-4359-b81c-cc50d013c102	String
"fineStatus"	PAID	String
"fineID"	7aac1089-4c00-4d78-a3ec-b28ad5c4b830	String
"userName"	Chandrika	String
"fineReason"	Overspeed	String
"emailID"	180031097cse@gmail.com	String
"mobile"	8466068349	String
"address"	ighhd	String
"accountID"	0xC410fcd6f73e707A2c59330C9951d31AE562f1	String
"fineAmount"	5	String

Logs Type here to search 30°C ENG 12:33 21-11-2021

Robo 3T - 1.4

File View Options Window Help

New Connection (4)

- System
  - > admin
  - > local
- BlockChain
  - > Complaint
  - > Fine
  - > Police
  - > RTO
  - > User
  - > Vechile
  - > Functions
  - > Users
- > config

localhost:27017 BlockChain

db.getCollection('Police').find({})

Key	Value	Type
(1) ObjectId("6193fa52e55f9a1b37d11375")	{ 4 fields }	Object
"_id"	ObjectId("6193fa52e55f9a1b37d11375")	ObjectID
"policeID"	e3c18f7b-1e91-4668-87f5-c8518f05a381	String
"userName"	Yashita	String
"password"	*123Yyy	String

Logs Type here to search 30°C ENG 12:33 21-11-2021

Robo 3T - 1.4

File View Options Window Help

New Connection (4)

- System
  - > admin
  - > local
- BlockChain
  - > Collections (6)
    - Complaint
    - Fine
    - Police
    - RTO
    - User
    - Vechile
  - > Functions
  - > Users
- > config

localhost:27017 BlockChain

db.getCollection('RTO').find({})

Key	Value	Type
(1) ObjectId("6193f924e55f9a1b37d11374")	{ 4 fields } <ul style="list-style-type: none"> <li>_id</li> <li>rtoID</li> <li>userName</li> <li>password</li> </ul>	Object
	ObjectId("6193f924e55f9a1b37d11374") 3e7b3438-dfbe-4e18-b164-d3ee4488a346	Object
	Yashitha	String
	*123YYyy	String

Logs

Type here to search

Windows Taskbar: 30°C, ENG, 12:33, 21-11-2021

Robo 3T - 1.4

File View Options Window Help

New Connection (4)

- System
  - > admin
  - > local
- BlockChain
  - > Collections (6)
    - Complaint
    - Fine
    - Police
    - RTO
    - User
    - Vechile
  - > Functions
  - > Users
- > config

localhost:27017 BlockChain

db.getCollection('User').find({})

Key	Value	Type
(1) ObjectId("6193f707e55f9a1b37d11373")	{ 10 fields } <ul style="list-style-type: none"> <li>_id</li> <li>userID</li> <li>userName</li> <li>password</li> <li>approval</li> <li>emailID</li> <li>mobile</li> <li>address</li> <li>privatekey</li> <li>accountID</li> </ul>	Object
	ObjectId("6193f707e55f9a1b37d11373") 71192c8b-6318-4cf1-8768-e1653ddd3a22	Object
	Yashitha	String
	*123YYyy	String
	ADMINAPPROVED	String
	vilasagarapu.yashitha2@gmail.com	String
	7382244899	String
	abc	String
	abcde	String
	H1JK9876A	String
(2) ObjectId("6193fc24e0c43bc0de8ade45")	{ 10 fields } <ul style="list-style-type: none"> <li>_id</li> <li>userID</li> <li>userName</li> <li>password</li> <li>approval</li> <li>emailID</li> <li>mobile</li> <li>address</li> <li>privatekey</li> <li>accountID</li> </ul>	Object
	ObjectId("6193fc24e0c43bc0de8ade45") 544ac1ce-eeff-4359-b81c-cc50d013c102	Object
	Chandrika	String
	chandrika	String
	ADMINAPPROVED	String
	180031097cse@gmail.com	String
	8466068349	String
	ighhd	String
	2477729a1661b5859bab69ff114e4a22829a5dfc1413ed33a89ac50e1248a0d4	String
	0xC410fccd6f73e707A2c59330C9951d31AE562f1	String

Logs

Type here to search

Windows Taskbar: 30°C, ENG, 12:34, 21-11-2021

Robo 3T - 1.4

File View Options Window Help

New Connection (4) Welcome db.getCollection("Fine").find() db.getCollection("Complaint").find() db.getCollection("Fine").find() db.getCollection("Police").find() db.getCollection("RTO").find() db.getCollection("User").find() db.getCollection("Vechile").find()

System New Connection localhost:27017 BlockChain db.getCollection('Vechile').find()

BlockChain

- Collections (6)
  - Complaint
  - Fine
  - Police
  - User
  - Vechile
  - Functions
- config

Key Value Type

(1) ObjectId("6193fe86e0c43bc0de8ade46")	{ 8 fields }	Object
_id	ObjectId("6193fe86e0c43bc0de8ade46")	Objectid
Number	AP-05-6442	String
Name	Chandrika	String
Gmail	180031097cse@gmail.com	String
Rcnumber	AC45678	String
Insurance	2021-11-17	String
Emession	2021-12-30	String
phone	8466068349	String

(2) ObjectId("6199ee47e0bcd1f188f6e08ab")	{ 8 fields }	Object
_id	ObjectId("6199ee47e0bcd1f188f6e08ab")	Objectid
Number	AP-06-2445	String
Name	Akhil	String
Gmail	180030029cse@gmail.com	String
Rcnumber	BC4567	String
Insurance	2021-11-21	String
Emession	2021-12-31	String
phone	9346906256	String

(3) ObjectId("6199ee8ae0bcd1f188f6e08ac")	{ 8 fields }	Object
---	--------------	--------

Logs Type here to search

30°C 12:34 ENG 21-11-2021

## **CHAPTER 9: CONCLUSION**

The blockchain technology offers a solution to most current issues. Blockchain innovation may play a huge part in the improvement of numerous administration issues. This will be particularly evident with additional execution of blockchain innovation in production of brilliant agreements. Such kind of agreements could digitize business benefits and further develop hidden business process. In future, blockchain is expected to bring about a paradigm shift in transport security. It will make the transaction verification process more efficient, and help achieve a quick delivery of services through mutual trust among participants.

Furthermore, to enable digital empowerment of residents by providing them with the decentralized options. Enable e-Signing of documents and make them available electronically and online minimize the use of physical documents.

Guarantee validness of the e-records and along these lines take out use of phony archives. Secure admittance to Government gave reports through an online interface and versatile application for inhabitants. Lessen managerial overhead of Government divisions and offices and make it simple for the inhabitants to get administrations.

## CHAPTER 10: REFERENCES

- [1] A Secure Used Car Trading System based on Block chain Technology by Cho Cho Htet, Department of computer engineering and information technology Mandalay, Myanmar karyancd2012@gmail.com.May Htet, Department of computer engineering and information technology Mandalay, Myanmar mayhtet@iuj.ac.jp`Link: <https://dl.acm.org/doi/10.1145/3366030.3366112>
- [2] G. Foroglou and A.-L. Tsilidou, "Further applications of the block chain," 2015.
- [3] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in Proceedings of IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858.
- [4] B. W. Akins, J. L. Chapman, and J. M. Gordon, "A whole new world: Income tax considerations of the bitcoin economy," 2013. [Online]. Available: <https://ssrn.com/abstract=2394738>  
Yan Lin, Senior Member, IEEE, Gary A. Jordan, Mark O. Sanford, Jinxiang Zhu, Member, IEEE, and William H. Babcock, "Economic Analysis of Establishing Regional Transmission Organization and Standard Market Design in the Southeast", IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 21, NO. 4, NOVEMBER 2006.
- [5] Juszakiewicz, "The use of Adobe Flex in combination with Java EE technology on the example of ticket booking system", in CAD Systems in Microelectronics (CADSM), 2011, pp. 317 - 320
- [6] Wan-Mi Chen, Yu-Cheng Chen, "Web design and implementation for remote control", in Intelligent Control and Automation (WCICA), 2012, pp. 920 - 924
- [7] Xiaosheng Yu, Yichang, China Cai Yi, "Design and Implementation of the Website Based on PHP & MYSQL", in E-Product E-Service and Entertainment (ICEEEE), 2010, pp. 1 – 4

# PLAGIARISM REPORT

PP Report - Plagiarism Checker Free x + paperpass.net/report

PaperPass.net Upload Report Account English Log out

The time it takes to process a paper depends on its length. Normally, the plagiarism check report will be completed within an hour.

Title	State	Similarity	Report	Submit Date
REGIONAL TRANSPORT OFFICE MANAGEMENT SYS...	Completed	9%	<a href="#">View Report</a>	2021-11-23 20:29

[delete](#)

**Warning:** The system only keeps the report within 100 days. Please download your report as soon as possible.

LegalStuff Privacy Policy  
User Agreement  
Refund Policy  
Copyright ©2021 PaperPass.net

Contact Us services@paperpass.net

Type here to search 26°C 20:43 23-11-2021

PP Report - Plagiarism Checker Free x PP REGIONAL TRANSPORT OFFICE | + view.paperpass.net/report/619d01ce35df1a2jm/

PaperPass.net Report Assessment Original Overall Similarity : 9% English

**Basic Information**

Title	REGIONAL TRANSPORT OFFICE MANAGEMENT SYSTEM USING BLOCKCHAIN
Time	2021-11-23 20:29:29 ( +05:30 Kolkata )
Words	9097

**Overall Similarity**

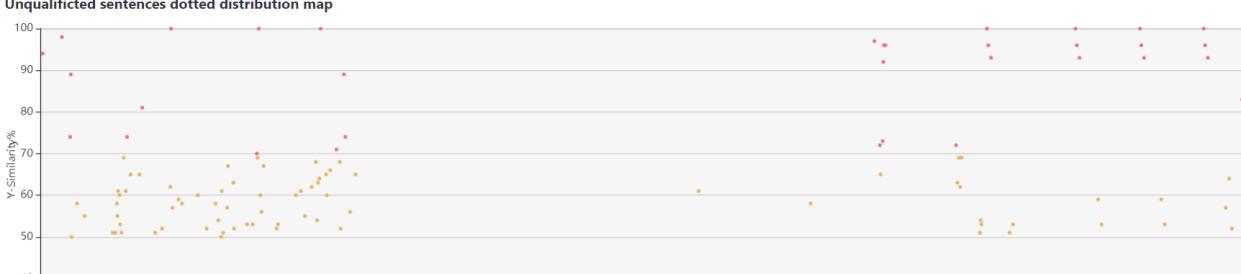
Overall Similarity	9%
Similarity degrees in local database	7%
Similarity degrees in internet database	5%

**Similarity Calculation Formula**  
 $(\text{Sentence1 similarity} + \text{Sentence2 similarity} + \dots) / n$   
Sentence similarity scale from 0.0~1.0 Green sentence represent similarity-free

**Sentences similarity distribution**



**Unqualified sentences dotted distribution map**



Type here to search 26°C 20:44 23-11-2021

Report - Plagiarism Checker Free x REGIONAL TRANSPORT OFFICE x +

view.paperpass.net/report/619d01ce35df1a2jm/

PaperPass Instant Report Assessment Original Overall Similarity : 9% English

**Similar resources on local database**

1. 2.3% Details Source : International Journal of Numerical Modelling: Electronic Networks, Devices and Fields Author : E. Bhaskaran Priyanka  
Title : A smart-integrated IoT module for intelligent transportation in oil industry
2. 2.2% Details Source : IET Software Author : Gabriel Sebastián Rivera Title : Model-based approach to develop learning exercises in language-learning applications
3. 1.6% Details Source : Microsystem Technologies Author : Marco Lee Title : Design and development of a friendly user interface for building construction traceability system
4. 1.4% Details Source : Research Papers Faculty of Materials Science and Technology Slovak University of Technology Author : Michal Kopček  
Title : Embedded PLC Webserver and Possibilities of its Utilization
5. 1.1% Details Source : SSRN Electronic Journal Author : Tyler Burleigh Title : How to Screen Out VPS and International Respondents Using Qualtrics: A Protocol
6. 1.1% Details Source : Advances in Methods and Practices in Psychological Science Author : Stephanie Permut Title : TaskMaster: A Tool for Determining When Subjects Are on Task
7. 1.0% Details Source : Network Security Author : Aditya K Sood Title : Abusing Glype proxies: attacks, exploits and defences
8. 0.9% Details Source : Journal of Information Technology & Software Engineering Author : Jaya Gupta Title : Server Side Protection against Cross Site Request Forgery using CSRF Gateway
9. 0.9% Details Source : Synthesis Lectures on Mobile and Pervasive Computing Author : Sumi Helal Title : Mobile Platforms and Development Environments
10. 0.9% Details Source : Advanced Materials Research Author : Hai Bo Wang Title : Building Mobile News Aggregation Application with Yahoo Pipes
11. 0.8% Details Source : International conference KNOWLEDGE-BASED ORGANIZATION Author : Luminita Giurgiu Title : Responsive Web Design Techniques
12. 0.8% Details Source : Electrical Engineering in Japan Author : Toru Watanabe Title : A CTI System to Support the Seamless Workplace and Its Implementation
13. 0.8% Details Source : International Journal of Autonomic Computing Author : Chandrakala Arya Title : Content extraction from news web pages using tag tree
14. 0.7% Details Source : Empirical Software Engineering Author : Mouna Hammoudi Title : RETRACTED ARTICLE: Facilitating debugging of web applications through recording reduction
15. 0.7% Details Source : Journal of Hospital Librarianship Author : Jason Bengtson Title : Scaling Into the Future With Smartlinks
16. 0.7% Details Source : Library Hi Tech News Author : Dan Lou Title : Two fast prototypes of web-based augmented reality enhancement for books
17. 0.6% Details Source : Advanced Materials Research Author : Cai Feng Cao Title : Research and Design of Open and Autonomous Learning System Based on HTML5
18. 0.6% Details Source : Technical Services Quarterly Author : Hong Ma Title : Tech Services on the Web: jQuery Mobile jquerymobile.com
19. 0.6% Details Source : Sensors Author : Rateb Jabbar Title : Blockchain for the Internet of Vehicles: A Decentralized IoT Solution for Vehicles Communication Using Ethereum
20. 0.6% Details Source : Réseaux Author : Samuel Goyet Title : Outils d'écriture du web et industrie du texte

Type here to search

26°C 21:22 23-11-2021 ENG

Report - Plagiarism Checker Free x REGIONAL TRANSPORT OFFICE x +

view.paperpass.net/report/619d01ce35df1a2jm/

PaperPass Instant Report Assessment Original Overall Similarity : 9% English

**Similar resources on local database**

11. 0.8% Details Source : International conference KNOWLEDGE-BASED ORGANIZATION Author : Luminita Giurgiu Title : Responsive Web Design Techniques
12. 0.8% Details Source : Electrical Engineering in Japan Author : Toru Watanabe Title : A CTI System to Support the Seamless Workplace and Its Implementation
13. 0.8% Details Source : International Journal of Autonomic Computing Author : Chandrakala Arya Title : Content extraction from news web pages using tag tree
14. 0.7% Details Source : Empirical Software Engineering Author : Mouna Hammoudi Title : RETRACTED ARTICLE: Facilitating debugging of web applications through recording reduction
15. 0.7% Details Source : Journal of Hospital Librarianship Author : Jason Bengtson Title : Scaling Into the Future With Smartlinks
16. 0.7% Details Source : Library Hi Tech News Author : Dan Lou Title : Two fast prototypes of web-based augmented reality enhancement for books
17. 0.6% Details Source : Advanced Materials Research Author : Cai Feng Cao Title : Research and Design of Open and Autonomous Learning System Based on HTML5
18. 0.6% Details Source : Technical Services Quarterly Author : Hong Ma Title : Tech Services on the Web: jQuery Mobile jquerymobile.com
19. 0.6% Details Source : Sensors Author : Rateb Jabbar Title : Blockchain for the Internet of Vehicles: A Decentralized IoT Solution for Vehicles Communication Using Ethereum
20. 0.6% Details Source : Réseaux Author : Samuel Goyet Title : Outils d'écriture du web et industrie du texte
21. 0.6% Details Source : Journal of Chemical Information and Modeling Author : Luc Patiny Title : ChemCalc: A Building Block for Tomorrow's Chemical Infrastructure
22. 0.5% Details Source : Procedia Computer Science Author : Yogi Udjaja  
Title : EKSPANPIXEL BLADSY STRANICA: Performance Efficiency Improvement of Making Front-End Website Using Computer Aided Software Engineering Tool
23. 0.5% Details Source : Information Technology and Libraries Author : Rick Bennett Title : assignFAST: An Autosuggest based tool for FAST Subject Assignment
24. 0.5% Details Source : FEMS Yeast Research Author : Minoska Valli Title : A subcellular proteome atlas of the yeast Komagataella phaffii

**Similar resources on internet**

1. 0.7% Details Source : Internet Title : jQuery Get Started - W3Schools
2. 0.7% Details Source : Internet Title : Hosted Libraries | Google Developers
3. 0.7% Details Source : Internet Title : What is a smart contract and how does it work?
4. 0.5% Details Source : Internet Title : Example 1: jquery cdn google - Newbedev

Type here to search

26°C 21:22 23-11-2021 ENG