**Police Complaint Management System using Blockchain Technology**

**1. INTRODUCTION:**

In India, complaints regarding offenses have to be registered under the law. There are two types of offenses i.e. cognizable and non-cognizable offenses [3]. Cognizable offenses include serious types of crimes like murder, theft, kidnapping, and rape, etc. As defined in Section 2 (c) of the Criminal Procedure Code 1973, in case of a cognizable offense, police can arrest the suspect without any warrant [6]. The assigned inspector can start the investigation process without any orders from the court. In the commission of any cognizable offense, the First information report aka FIR. is registered at the police station. Any individual can file an FIR. if he/she is a victim or has seen the offense being committed. FIR. details include the complainant’s name and address, date and time of location and facts of the incident, etc. Once the FIR is registered, chargesheet report is filed by the police officer. The complainant can apply for acquiring the chargesheet by submitting a letter under the Right to Information Act (RTI) and by paying a certain amount of fees to the court. Non-cognizable offenses include criminal activities like cheating or forgery etc. N.C. complaint aka non-cognizable complaint can be registered at the police station. It has a structure similar to FIR. As defined in Section 2(l) of Criminal Procedure court 1973, in case of non-cognizable offense, a police officer has no authority to arrest/investigate without a warrant [6]. The police officer has to obtain permission from the court/magistrate in order to start the investigation process. The crime rate i.e. crime per lakh is increasing at a rapid rate. More than 50 lakh cognizable crimes were registered in the year 2018 [1]. Due to increased criminal activities as well as the presence of corrupt police officials, they tend to refuse, avoid or detain the registration of FIR/NCR/Complaints which are the obstacles for the complainants to seek justice at the very beginning. According to a survey [2], 24% of people were unable to register their complaints and 9% of people said that the non-registration was because they were demanded to pay a bribe. Among the people who were able to register their complaints, 30% of the complainants didn’t receive an FIR copy. There is a need for a transparent system to eradicate corruption from the public systems. We aim to propose an online police complaint management system using blockchain technology for managing FIR’s and NCR’s in a decentralized manner in order to cater to problems involving denial of police officers to file complaints. Blockchain technology is based on a peer to peer network topology i.e. it is a distributed as well as a decentralized data structure that contains all the legal transactions in links of blocks. The first application of blockchain technology is bitcoin [5], which was proposed by Satoshi Nakamoto. The prime functionality of blockchain is to make sure that only valid blocks are entered into the chain i.e. the block should acquire minimum votes/consensus. The validation of a block is done with the help of consensus mechanisms like Proof of Work, Proof of State and Proof of Capacity, etc. Once a block of transactions is added to the network, it is computably infeasible to tamper a block. The Interplanetary file system(IPFS) is an algorithm which involves peer to peer network for storing as well as sharing files in a distributed manner. The mechanism uses content based addressing i.e. every file is hashed (based on content) and stored in a decentralized network. IPFS can be incorporated with blockchain to provide features like immutability, high reliability and throughput. Relevant to this context, we tend to provide a decentralized application which keeps a track of all the activities related to police complaints right from filing a complaint to submission of charge-sheet to the court. The use of blockchain technology ensures trust between the complainants and the police department. The system is not only secure from loss of data but also from brute force hacking or other types of malicious attacks.

**1.1 Objective of the project:**

The criminal activities in India are increasing at a rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing FIRs and NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be present in the police station to file a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. Thus, there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR filed by the police will be encrypted, stored in the IPFS and hash is added to the blockchain network. If the police decide not to file the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any chances of the FIR/NCR being tampered and going unnoticed.

**2. LITERATURESURVEY:**

**FIR SYSTEM USING BLOCK CHAIN TECHNOLOGY**

In police stations, there are records of crimes. Crime Records are unable to locate crimes and the offenders who committed them. To maintain the crime and criminal data under the current system, a FIR is used. It has less security and makes fraud simple to do. Each time, a manual update has been made to the record. This system's primary goal is to secure data utilising block chain technology. Using their authentication credentials, Crime Investigators can view the data form database. The reports, which are prepared by witnesses and police officers, are accessible to the investigator (writer). Investigators have the authority to edit data (i.e., update, remove, and so on), and this data aids investigators in speeding up their investigations and identifying offenders more quickly. Previous research has focused on the centralized handling of digital evidence, however if a centralized system server is breached, sensitive operational and investigation data may be exposed. As a result, there is a need to manage digital evidence and investigative information in a distributed system setting using block chain technology. Performance is reduced when massive amounts of data, such evidential films, are kept in a block chain because more data must be processed only once before being generated. As a result, we suggest three-tier block chain architecture, with hot and cold block chains for digital evidence. Information that changes regularly is stored on the hot block chain, whereas material that does not change, such as files, is saved in the cold blockchain. To assess the system, we compared the storage and inquiry processing performance of digital crime evidence across the multi-level block chain system's capacities.

# **e-Cops: An Online Crime Reporting and Management System for Riyadh City**

he idea draws motivation from the people who hesitate to visit the police station and personal belief of weak investigation and corruption and limited spreading of crime information [3].The usage of the internet is increased in every sector, so people find it easier to register a complaint online rather than visiting the police station, it is secure and possible to hide their identity if they want [1]. This application helps to create a bridge between normal people and the police department to share information and evidence [1] [2]. It is helpful to track and monitor the criminals around the state and country and maintain a complete record of criminal information

**E-Police System- FIR Registration and Tracking through Android Application**

We can see that technology has touched many spheres of our lives in India. There is technology in business, in education, in socializing and maintaining human relations, in purchasing, in agriculture, in banking, communication, and almost every part of our lives. This intrusion of technology has aided the work in all these sections, and has proved beneficial, and time and effort saving. The only major part of our society that still remains majorly devoid of this luxury is the Indian Police Department. The Indian Police Department has ever since remained manually driven for most of its routine chores. The officials have been adopting the basic fundamental methods of carrying out the proceedings with the traditional “pen and paper” method being highly prevalent. These traditional practices were comfortable in earlier days, when population was far less, and the crime rates were also comparably minimal. But in today’s India, when the evil elements of the society are in a boom and so many cases being registered every day, it has become a very tedious task to manage the case and all its related documents, manually. Digitization in Police department is the need of the hour. The traditional method of visiting a police station for registering a police complaint and getting updates needs to be replaced with an online process. Hence an E-police system is being developed which will collect complainant’s data through a mobile application, sends the information over to the Police department on their web portal, and in this way the entire interaction occurs online, with information exchanges over the application and the web portal.

# **Design and Implementation of an E-Policing System to Report Crimes in Nigeria**

Policing system in Nigeria, a country of about 160 million people is in a state of degradation, criminal offences such as theft, bribery, armed robbery, assassinations and corruption are highly enormous. At the centre of this transformation is the electronic policing (E-policing) system, which is an emerging Internet technology. E-policing practiced by many law enforcement organizations around the globe to improve law enforcement services provided to the community. The use of Internet technology and electronic-based systems seeks an infrastructure with initial costs that are high and fast application development. This paper is focused on designing and developing an E-policing system to track and control crime rate in Nigeria, and also presents a framework based on technology towards alleviating barriers of manual policing and provisions of how cloud computing principles and Internet technology could be adapted to Nigeria Police Force (NPF) present and future expectations

# **Proposed e-police system for enhancement of e-government services of Bangladesh**

E-government is the ICT based system of government service delivery for achieving good governance which is necessity for good and corruption free nation. E-police system is an e-government related service and it makes the communication process a possibility, a great success for modern era which increases the professional efficiency for the government police administration, so we can apply this system in Bangladesh. The aim of this paper is to upgrade the country's police administration to the world standard. The home ministry would be connected with the several police units of the city in a fibre-optic based metropolitan area network and a database will be setup for warrant notices, examining the finger prints using the latest electronic device etc. There have to be set up a 'Third Eye' software in the special branches of the police department so that it helps the police supervisors to monitor crime and criminal records. There have to be set up an electronic database and an interactive website which will contain daily press releases, supplement, list of top terrorists and criminals, lists of people under police custody and people injured in road or other accidents etc. In this paper we focus on the infrastructure of an e-police system as well as its steps, challenges of implementation and its necessity. For implementing the software we can use JAVA, PHP (especially Approver) and MySQL.

# **Police Complaint Management System using Blockchain Technology**

The criminal activities in India are increasing at a rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing FIRs and NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be present in the police station to ﬁle a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. hus there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR ﬁled by the police will be encrypted, store in the IPFS and hash is added to the blockchain network. If the police decide not to ﬁle the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any chances ofthe FIR/NCR being tampered and going unnoticed. The criminal activities in India are increasing at a rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing FIRs and NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be present in the police station to ﬁle a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. Thus there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR ﬁled by the police will be encrypted, stored in the IPFS and hash is added to the blockchain network. If the police decide not to ﬁle the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any chances of the FIR/NCR being tampered and going unnoticed. The criminal activities in India are increasing at a rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing FIRs and NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be present in the police station to ﬁle a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. Thus there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR ﬁled by the police will be encrypted, stored in the IPFS and hash is added to the blockchain network. If the police decide not to ﬁle the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any c ances of the FIR/NCR being tampered and going unnoticed. he criminal activities in India are increasing at a rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing FIRs and NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be presentin the police station to ﬁle a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. Thus there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR ﬁled by the police will be encrypted, stored in the IPFS and hash is added to the blockchain network. If the police decide not to ﬁle the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any chances of the FIR/NCR being tampered and going unnoticed.The criminal activities in India are increasing ate rapid rate. Many of these activities go unreported. Even after having an online portal for the police for storing Firesand NCRs, most of the FIRs are handwritten as a traditional practice. In most of the cases, the complainant has to be presenting the police station to ﬁle a cognizable offense. An effective system for e-governance was started in 2009 named Crime and Criminal Tracking Network and Systems (CCTNS) for the entire country. However, it is a centralized system for a particular state. Thus there is a need for a completely decentralized system for assuring that there is no central point of failure in the system and complaints are managed securely protected from unauthorized access. Our aim is to propose a blockchain-based solution to manage complaints against both cognizable and non-cognizable offenses. The FIR ﬁled by the police will be encrypted, stored in the IPFS and hash is added to the blockchain network. If the police decide not to ﬁle the FIR under pressure or deny receiving any complaint, then the complainant will have strong proof against him/her as the complaint along with its timestamp was stored on the blockchain network. Having all the records stored in an immutable database would remove any chances of the FIR/NCR being tampered and going unnoticed.

# **LDAvis: A method for visualizing and interpreting topics**

We present LDAvis, a web-based interactive visualization of topics estimated using Latent Dirichlet Allocation that is built using a combination of R and D3. Our visualization provides a global view of the topics (and how they differ from each other), while at the same time allowing for a deep inspection of the terms most highly associated with each individual topic. First, we propose a novel method for choosing which terms to present to a user to aid in the task of topic interpretation, in which we define the relevance of a term to a topic. Second, we present results from a user study that suggest that ranking terms purely by their probability under a topic is suboptimal for topic interpretation. Last, we describe LDAvis, our visualization system that allows users to flexibly explore topic-term relationships using relevance to better understand a fitted LDA model.

# **Being Active in Online Communications: Firm Responsiveness and Customer Engagement Behaviour**

This research investigates the behavioral effects of firms' online activeness in influencing customer engagement in word-of-mouth communications. Using a large-scale field dataset of hotel reviews and managerial responses, this study empirically examines firm responsiveness in relationship to community members' participation in the online review posting. Novel findings are reported that response volume and speed are important for effecting firm–customer interactions. This highlights a firm-leading influence on customers' word-of-mouth behaviour by identifying firm engagement as a motivational driver of customer engagement. It offers implications for researchers and practitioners with regard to social media marketing, in particular firm engaging in the online communication network by acting in an active and prompt manner

# **Online Fir Registration and Sos System**

Nowadays, technology has touched many citizens living in India and abroad. Technology is in industry, education, social interactions and preservation human relations, in sales, in agriculture and in banking, communication, and just about any aspect of our life. This technology interference has supported the work in many of these areas and has been helpful, saving time and effort. On the other hand, the rising in crime rates all across the globe is one of the serious concerns for all the governments. However, to tackle this problem, a vending teller machine-based FIR online registration system is proposed in this research work. This will assuredly help people in registering the crimes and will remove the fear of going to police station to lodge the complaint. Furthermore, this machine can be installed at various public places for the ease of access. This machine authenticates the user first and then allowed to lodge the complaint again the crime. In the proposed system, wavelet-based image processing has implemented for the better authentication of the person registering the complaint. The verification of the person is carried out by Haar cascade-based image processing technique, and it has been found that the results are 92% in real time. The further extension in this machine can be customized for in case of no action the report may be sent to the Chief Minister’s office.

# **Two-Level Blockchain System for Digital Crime Evidence Management**

Digital evidence, such as evidence from CCTV and event data recorders, is highly valuable in criminal investigations, and is used as definitive evidence in trials. However, there are risks when digital evidence obtained during the investigation of a case is managed through a physical hard disk drive until it is submitted to the court. Previous studies have focused on the integrated management of digital evidence in a centralized system, but if a centralized system server is attacked, major operations and investigation information may be leaked. Therefore, there is a need to reliably manage digital evidence and investigation information using blockchain technology in a distributed system environment. However, when large amounts of data—such as evidence videos—are stored in a blockchain, the data that must be processed only within one block before being created increase, causing performance degradation. Therefore, we propose a two-level blockchain system that separates digital evidence into hot and cold blockchains. In the criminal investigation process, information that frequently changes is stored in the hot blockchain, and unchanging data such as videos are stored in the cold blockchain. To evaluate the system, we measured the storage and inquiry processing performance of digital crime evidence videos according to the different capacities in the two-level blockchain system

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

In Earlier all criminal records were maintained in a register which can be easily manipulated and time-consuming process to transfer register data from one police station to other station. To overcome from above issue Online Criminal Tracking System was introduced which is managed by a single centralized server, if this server is down then criminal tracking services will get disturbed and internal employee of server management can be manipulated easily and those manipulation cannot be detected easily.

**DISADVANTAGE:**

* Less Secure

**3.2 Proposed System**

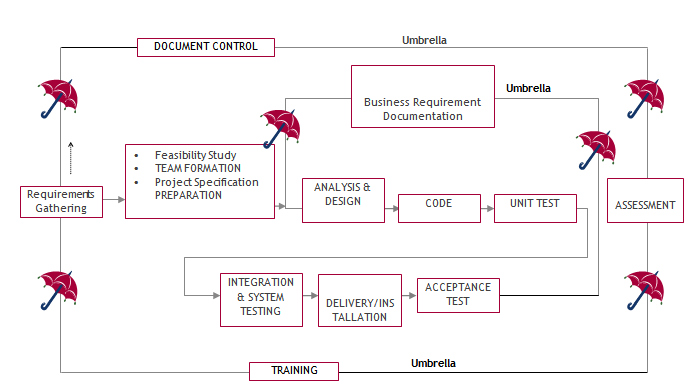
The proposed system is a decentralized platform for managing complaints with the help of various technologies like blockchain, IPFS, etc. The detailed architecture of the system is explained in the form of modules followed by the implementation of the system Like Security Module, Blockchain Module, Web/Mobile Interface and Implementation So by employing Blockchain technology we can overcome from all disadvantages faces during single centralized servers.

**ADVANTAGE:**

* More Secure

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

****

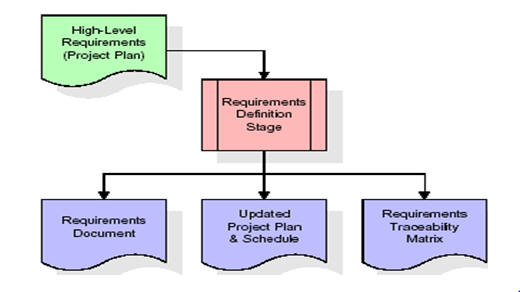
The requirements gathering process takes as its input SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering stage:**

the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

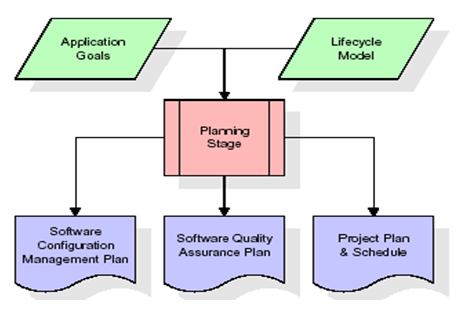
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

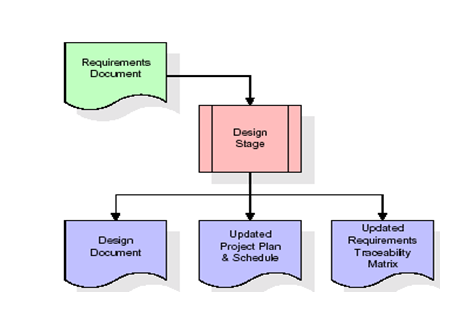
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

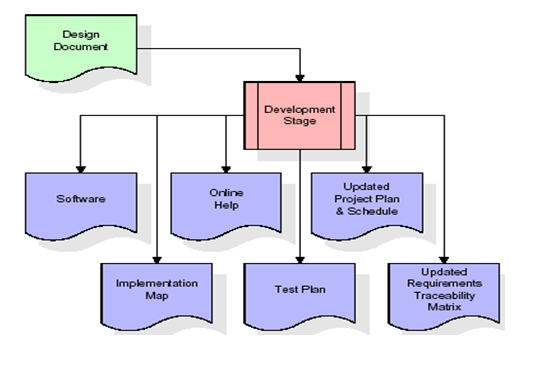
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

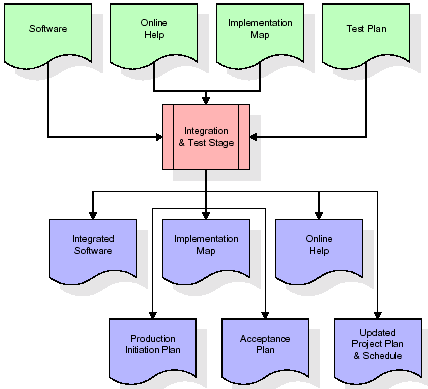
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

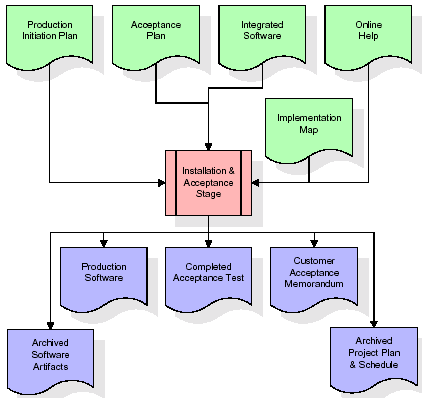


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would

Ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user-friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB (min)
* Hard Disk - 500 GB

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python (3.7.0)

**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake.



**Use case Diagram:**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

****

**Sequence Diagram:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



**Component Diagram:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**Deployment Diagram:**

A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.

Index

Police Login

Admin Login

Add New Police

Add Fir

Update Investigations

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1. Index 2. Successfully Index

3. Police Login 4. Successfully Police Login

5. Admin Login 6. Successfully Admin Login

7. Add New Police 8. Successfully Add New Police

9. Add Fir 10. Successfully Added FIR

11. Update Investigations 12. Successfully Update Investigations

**5. IMPLEMETATION**

**5.1 PYTHON**

\* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

\*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

\* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

\* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

* Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
* Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
* Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
* Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
* Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
* Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
* Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
* Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
* Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

**5.2 Installation**

To install Python on your computer, follow these basic steps:

* Step 1: Visit the Python website Go to the official Python website at <https://www.python.org/>.
* Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
* Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
* Step 4: Install Python 3.7.0 using Cmd.
* Step 5: Install the all libraries that required to run the project
* Step 6: Run

**5.3 Python Features:**

1. **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2. **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3. **Open Source:** Python is a free online programming language since it is open-source.
4. **Portable:** Python is portable because the same code may be used on several computer standard
5. **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6. **GUI:** It stands for GUI (Graphical User Interface)
7. **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

**5.4 Python GUI (Tkinter)**

* Python provides a wide range of options for GUI development (Graphical User Interfaces).
* Tkinter, the most widely used GUI technique, is used for all of them.
* The Tk GUI toolkit offered by Python is used with the conventional Python interface.
* Tkinter is the easiest and quickest way to write Python GUI programs.
* Using Tkinter, creating a GUI is simple.
* A part of Python's built-in library is Tkinter. The GUI programs were created.
* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

**5.5 Python IDLE**

* Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
* Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
* IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
* IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
* The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
* Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

* Open the command prompt (Windows) or terminal (macOS/Linux).
* Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
* Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

* Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
* Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
* Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
* Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
* Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
* Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
* Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

**5.6 Libraries**

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

**Libraries in Python offer various advantages:**

* Code Reusability:
* Efficiency:
* Collaboration
* Domain-Specific Functionality
* To use a Python library, you need to install it first.

There are some libraries following:

* **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

* It has functions for analysing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
* Pandas allow us to analyse big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

* **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
* NumPy stands for Numerical Python.
* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
* The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.
* **Matplotlib:**

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

* Plotting Functions
* Customization Options
* Multiple Interfaces
* Integration with NumPy and pandas
* Subplots and Figures:
* Saving and Exporting
* **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

* Easy-to-Use Interface:
* Broad Range of Algorithms:
* Data Pre-processing and Feature Engineering:
* Model Evaluation and Validation:
* Integration with NumPy and pandas:
* Robust Documentation and Community Support:
* **Keras:**

\* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

* User-Friendly API
* Multi-backend Support
* Wide Range of Neural Network Architectures
* Pre-trained Models and Transfer Learning:
* Easy Model Training and Evaluation:
* GPU Support:
* **h5py:**

\* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

* + HDF5 File Access
  + Dataset Handling:
  + Group Organization:
  + Attributes:
  + Compatibility with NumPy
  + Performance
* **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

* Neural Network Framework:
* Computational Graphs
* Automatic Differentiation
* GPU and TPU Support
* Distributed Computing
* Deployment Capabilities
* **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

* Cross-Platform Compatibility
* Simple and Easy-to-Use
* Widgets and Layout Management
* Event-Driven Programming
* Customization and Styling
* Integration with Other Libraries
* **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

* Text Processing
* Part-of-Speech Tagging
* Named Entity Recognition
* Chunking and Parsing
* Sentiment Analysis:
* WordNet Integration:
* **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

Here are some key features and functionalities of SciPy:

* Numerical Integration:
* Optimization and Root Finding
* Linear Algebra
* Signal and Image Processing
* Statistics

**5.2 Sample Code:**

**Views.py**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from django.core.files.storage import FileSystemStorage

import os

from datetime import date

import os

import json

from web3 import Web3, HTTPProvider

import ipfsApi

import os

from django.core.files.storage import FileSystemStorage

import pickle

from datetime import date

import pyaes, pbkdf2, binascii, os, secrets

import base64

import urllib, mimetypes

from django.http import HttpResponse

global details, username

details=''

global contract

api = ipfsApi.Client(host='http://127.0.0.1', port=5001)

def readDetails(contract\_type):

global details

details = ""

print(contract\_type+"======================")

blockchain\_address = 'http://127.0.0.1:9545' #Blokchain connection IP

web3 = Web3(HTTPProvider(blockchain\_address))

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

compiled\_contract\_path = 'Police.json' #Police contract code

deployed\_contract\_address = '0x1DD4fb45C1cdC8C3f32cbaA60464c8107D4D4058' #hash address to access Police contract

with open(compiled\_contract\_path) as file:

contract\_json = json.load(file) # load contract info as JSON

contract\_abi = contract\_json['abi'] # fetch contract's abi - necessary to call its functions

file.close()

contract = web3.eth.contract(address=deployed\_contract\_address, abi=contract\_abi) #now calling contract to access data

if contract\_type == 'addusers':

details = contract.functions.getUsers().call()

if contract\_type == 'complaints':

details = contract.functions.getComplaints().call()

if contract\_type == 'investigations':

details = contract.functions.getInvestigation().call()

print(details)

def saveDataBlockChain(currentData, contract\_type):

global details

global contract

details = ""

blockchain\_address = 'http://127.0.0.1:9545'

web3 = Web3(HTTPProvider(blockchain\_address))

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

compiled\_contract\_path = 'Police.json' #Police contract file

deployed\_contract\_address = '0x1DD4fb45C1cdC8C3f32cbaA60464c8107D4D4058' #contract address

with open(compiled\_contract\_path) as file:

contract\_json = json.load(file) # load contract info as JSON

contract\_abi = contract\_json['abi'] # fetch contract's abi - necessary to call its functions

file.close()

contract = web3.eth.contract(address=deployed\_contract\_address, abi=contract\_abi)

readDetails(contract\_type)

if contract\_type == 'addusers':

details+=currentData

msg = contract.functions.addUsers(details).transact()

tx\_receipt = web3.eth.waitForTransactionReceipt(msg)

if contract\_type == 'complaints':

details+=currentData

msg = contract.functions.addComplaints(details).transact()

tx\_receipt = web3.eth.waitForTransactionReceipt(msg)

if contract\_type == 'investigations':

details+=currentData

msg = contract.functions.addInvestigation(details).transact()

tx\_receipt = web3.eth.waitForTransactionReceipt(msg)

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def PoliceLogin(request):

if request.method == 'GET':

return render(request, 'PoliceLogin.html', {})

def AdminLogin(request):

if request.method == 'GET':

return render(request, 'AdminLogin.html', {})

def AddNewPolice(request):

if request.method == 'GET':

return render(request, 'AddNewPolice.html', {})

def AddFir(request):

if request.method == 'GET':

return render(request, 'AddFir.html', {})

def getKey(): #generating key with PBKDF2 for AES

password = "s3cr3t\*c0d3"

passwordSalt = '76895'

key = pbkdf2.PBKDF2(password, passwordSalt).read(32)

return key

def encrypt(plaintext): #AES data encryption

aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163637846056248223))

ciphertext = aes.encrypt(plaintext)

return ciphertext

def decrypt(enc): #AES data decryption

aes = pyaes.AESModeOfOperationCTR(getKey(), pyaes.Counter(31129547035000047302952433967654195398124239844566322884172163637846056248223))

decrypted = aes.decrypt(enc)

return decrypted

def UpdateInvestigations(request):

if request.method == 'GET':

global username

output = '<tr><td><font size="" color="black">Complaint&nbsp;ID</font></td><td><select name="t1">'

readDetails("complaints")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

output+='<option value="'+arr[0]+'">'+arr[0]+'</option>'

output += "</select></td></tr>"

context= {'data1': output}

return render(request, 'UpdateInvestigations.html', context)

def UpdateInvestigationsAction(request):

if request.method == 'POST':

global username

today = date.today()

complaint = request.POST.get('t1', False)

investigation = request.POST.get('t2', False)

today = date.today()

data = complaint+"#"+username+"#"+investigation+"#"+str(today)+"#\n"

saveDataBlockChain(data,"investigations")

output = "Investigation Details Submitted Under Complaint No : "+str(complaint)

context= {'data': output}

return render(request, 'UserScreen.html', context)

def ViewInvestigations(request):

if request.method == 'GET':

global username

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Complaint No</font></th>'

output+='<th><font size=3 color=black>Complaint Details</font></th>'

output+='<th><font size=3 color=black>Complainer Name</font></th>'

output+='<th><font size=3 color=black>Complainer Contact No</font></th>'

output+='<th><font size=3 color=black>Address</font></th>'

output+='<th><font size=3 color=black>Criminal Name</font></th>'

output+='<th><font size=3 color=black>Contact No</font></th>'

output+='<th><font size=3 color=black>Address</font></th>'

output+='<th><font size=3 color=black>Case Type</font></th>'

output+='<th><font size=3 color=black>Station Details</font></th>'

output+='<th><font size=3 color=black>IPFS Storage Hashcode</font></th>'

output+='<th><font size=3 color=black>Document Name</font></th>'

output+='<th><font size=3 color=black>Complaint Date</font></th>'

output+='<th><font size=3 color=black>Inspector Name</font></th>'

output+='<th><font size=3 color=black>Investigation Details</font></th>'

output+='<th><font size=3 color=black>Download Documents</font></th></tr>'

readDetails("investigations")

investigations = details.split("\n")

readDetails("complaints")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

output+='<tr><td><font size=3 color=black>'+arr[0]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[1]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[2]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[3]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[4]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[5]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[6]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[7]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[8]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[9]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[10]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[11]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[12]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[13]+'</font></td>'

invest = ""

for k in range(len(investigations)):

ar = investigations[k].split("#")

if ar[0] == arr[0]:

invest += ar[2]+"\n"

output+='<td><font size=3 color=black>'+invest+'</font></td>'

output+='<td><a href=\'DownloadAction?file='+arr[0]+'\'><font size=3 color=black>Click Here</font></a></td></tr>'

output += "</table><br/><br/><br/><br/>"

context= {'data': output}

return render(request, 'UserScreen.html', context)

def ViewReports(request):

if request.method == 'GET':

global username

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Complaint No</font></th>'

output+='<th><font size=3 color=black>Complaint Details</font></th>'

output+='<th><font size=3 color=black>Complainer Name</font></th>'

output+='<th><font size=3 color=black>Complainer Contact No</font></th>'

output+='<th><font size=3 color=black>Address</font></th>'

output+='<th><font size=3 color=black>Criminal Name</font></th>'

output+='<th><font size=3 color=black>Contact No</font></th>'

output+='<th><font size=3 color=black>Address</font></th>'

output+='<th><font size=3 color=black>Case Type</font></th>'

output+='<th><font size=3 color=black>Station Details</font></th>'

output+='<th><font size=3 color=black>IPFS Storage Hashcode</font></th>'

output+='<th><font size=3 color=black>Document Name</font></th>'

output+='<th><font size=3 color=black>Complaint Date</font></th>'

output+='<th><font size=3 color=black>Inspector Name</font></th>'

output+='<th><font size=3 color=black>Investigation Details</font></th>'

output+='<th><font size=3 color=black>Download Documents</font></th></tr>'

readDetails("investigations")

investigations = details.split("\n")

readDetails("complaints")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

output+='<tr><td><font size=3 color=black>'+arr[0]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[1]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[2]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[3]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[4]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[5]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[6]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[7]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[8]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[9]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[10]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[11]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[12]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[13]+'</font></td>'

invest = ""

for k in range(len(investigations)):

ar = investigations[k].split("#")

if ar[0] == arr[0]:

invest += ar[2]+"\n"

output+='<td><font size=3 color=black>'+invest+'</font></td>'

output+='<td><a href=\'DownloadAction?file='+arr[0]+'\'><font size=3 color=black>Click Here</font></a></td></tr>'

output += "</table><br/><br/><br/><br/>"

context= {'data': output}

return render(request, 'AdminScreen.html', context)

def AddFirAction(request):

if request.method == 'POST':

global username

today = date.today()

complaint = request.POST.get('t1', False)

complainer\_name = request.POST.get('t2', False)

complainer\_contact = request.POST.get('t3', False)

complainer\_address = request.POST.get('t4', False)

criminal\_name = request.POST.get('t5', False)

criminal\_contact = request.POST.get('t6', False)

criminal\_address = request.POST.get('t7', False)

case\_type = request.POST.get('t8', False)

station = request.POST.get('t9', False)

filedata = request.FILES['t10'].read()

filename = request.FILES['t10'].name

filedata = encrypt(filedata)

complaint\_id = 0

readDetails("complaints")

rows = details.split("\n")

if len(rows) == 0:

complaint\_id = 1

else:

complaint\_id = len(rows)

filedata = pickle.dumps(filedata)

hashcode = api.add\_pyobj(filedata)

data = str(complaint\_id)+"#"+complaint+"#"+complainer\_name+"#"+complainer\_contact+"#"+complainer\_address+"#"+criminal\_name+"#"+criminal\_contact+"#"

data += criminal\_address+"#"+case\_type+"#"+station+"#"+hashcode+"#"+filename+"#"+str(today)+"#"+username+"\n"

saveDataBlockChain(data,"complaints")

output = "Complaint Details saved in Blockchain with IPFS storage hashcode : "+hashcode+"<br/>Complaint No : "+str(complaint\_id)

context= {'data': output}

return render(request, 'UserScreen.html', context)

def DownloadAction(request):

if request.method == 'GET':

global username

complaint = request.GET['file']

fileName = ""

hashcode = ""

readDetails("complaints")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[0] == complaint:

fileName = arr[11]

hashcode = arr[10]

break

content = api.get\_pyobj(hashcode)

content = pickle.loads(content)

content = decrypt(content)

response = HttpResponse(content,content\_type="application/octet-stream")

response['Content-Disposition'] = "attachment; filename=%s" % fileName

return response

def ViewPolice(request):

if request.method == 'GET':

global username

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Username</font></th>'

output+='<th><font size=3 color=black>Passwor</font></th>'

output+='<th><font size=3 color=black>Police Station Contact No</font></th>'

output+='<th><font size=3 color=black>E-Mail ID</font></th>'

output+='<th><font size=3 color=black>Address</font></th></tr>'

readDetails("addusers")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

output+='<tr><td><font size=3 color=black>'+arr[0]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[1]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[2]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[3]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[4]+'</font></td></tr>'

output += "</table><br/><br/><br/><br/>"

context= {'data': output}

return render(request, 'AdminScreen.html', context)

def AddNewPoliceAction(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

contact = request.POST.get('contact', False)

email = request.POST.get('email', False)

address = request.POST.get('address', False)

record = 'none'

readDetails("addusers")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[1] == username:

record = "exists"

break

if record == 'none':

data = username+"#"+password+"#"+contact+"#"+email+"#"+address+"\n"

saveDataBlockChain(data,"addusers")

context= {'data':'Signup process completed and record saved in Blockchain'}

return render(request, 'AddNewPolice.html', context)

else:

context= {'data':username+'Username already exists'}

return render(request, 'AddNewPolice.html', context)

def PoliceLoginAction(request):

if request.method == 'POST':

global username

username = request.POST.get('username', False)

password = request.POST.get('password', False)

status = 'none'

readDetails("addusers")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[0] == username and arr[1] == password:

status = 'success'

break

if status == 'success':

context= {'data':"Welcome "+username}

return render(request, 'UserScreen.html', context)

else:

context= {'data':'Invalid login details'}

return render(request, 'PoliceLogin.html', context)

def AdminLoginAction(request):

if request.method == 'POST':

global username

username = request.POST.get('username', False)

password = request.POST.get('password', False)

if username == 'admin' and password == 'admin':

context= {'data':"Welcome "+username}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':'Invalid login details'}

return render(request, 'AdminLogin.html', context)

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

**System Testing**

Testing has become an System integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

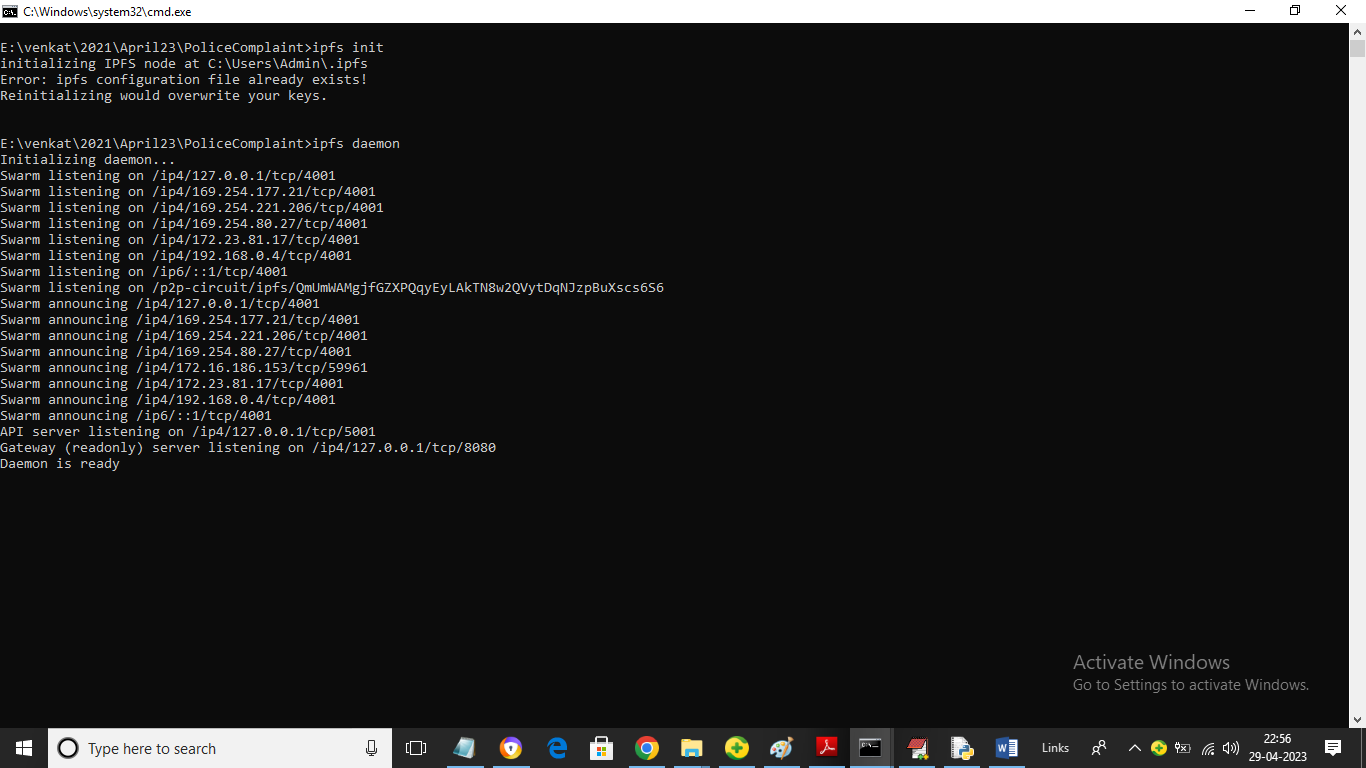
**Acceptance Testing**

When that user fined no major problems with its accuracy the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

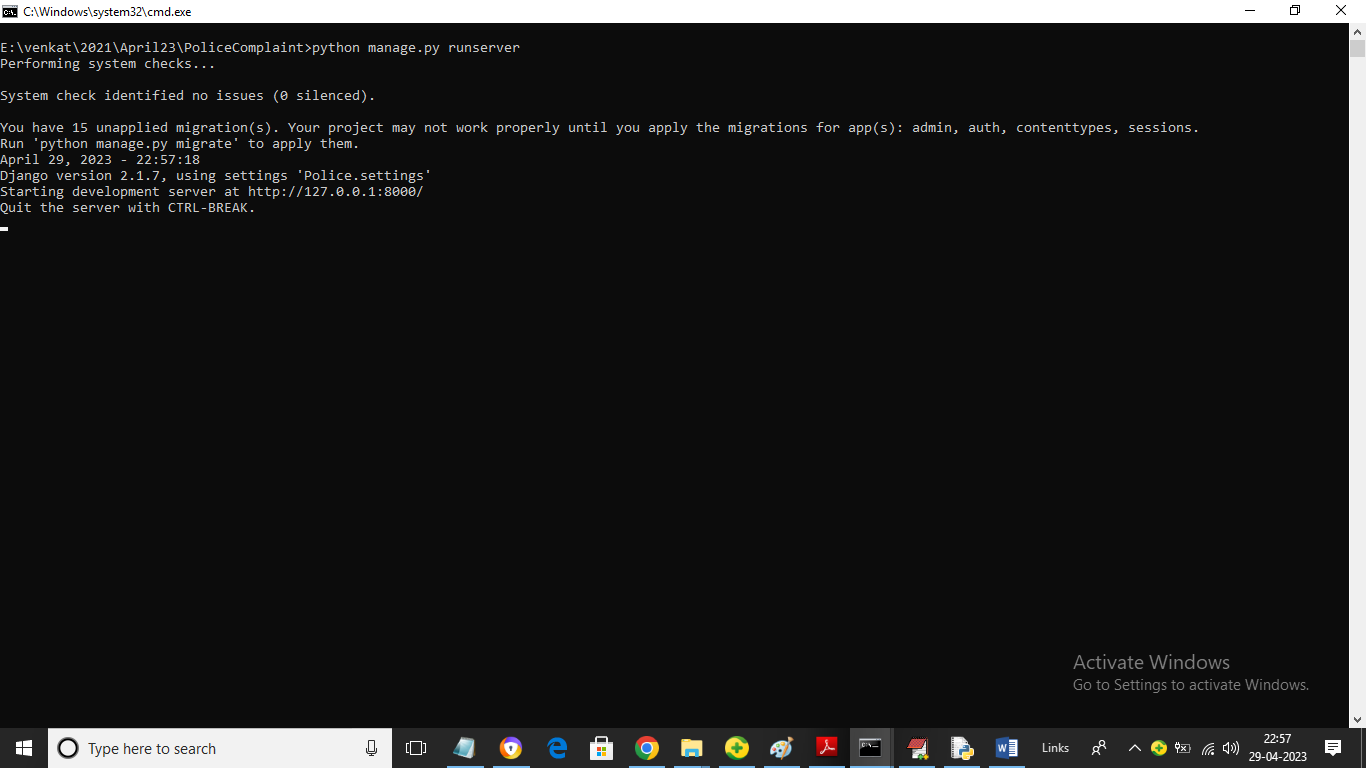
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test  Case  ID | Test  case  Name | Test Case  Desc. | Step | Expected | Actual | Test  Case  Status | Test  Case  Priority |
| 01 | Index | Verify The Index done or not | If The Index may not done | We cannot do the further operations | We Can do further operations | High | High |
| 02 | Police Login | Verify the  Police Login done or not | If the  Police Login May Not be done | We cannot do the further operations | We Can do further operations | High | High |
| 03 | Add New Police | Verify The Add New Police done or not | If The Add New Police may not done | We cannot do the further operations | We Can do further operations | High | High |
| 04 | Add FIR | Verify The Add FIR done or not | If The Add FIR may not done | We cannot do the further operations | We Can do further operations | High | High |
| 05 | Update Investigations | Verify Update Investigations Action done or not | If The Update Investigations Action may not done | We cannot do the further operations | We Can do further operations | High | High |

## 7. SCREEN SHOTS:

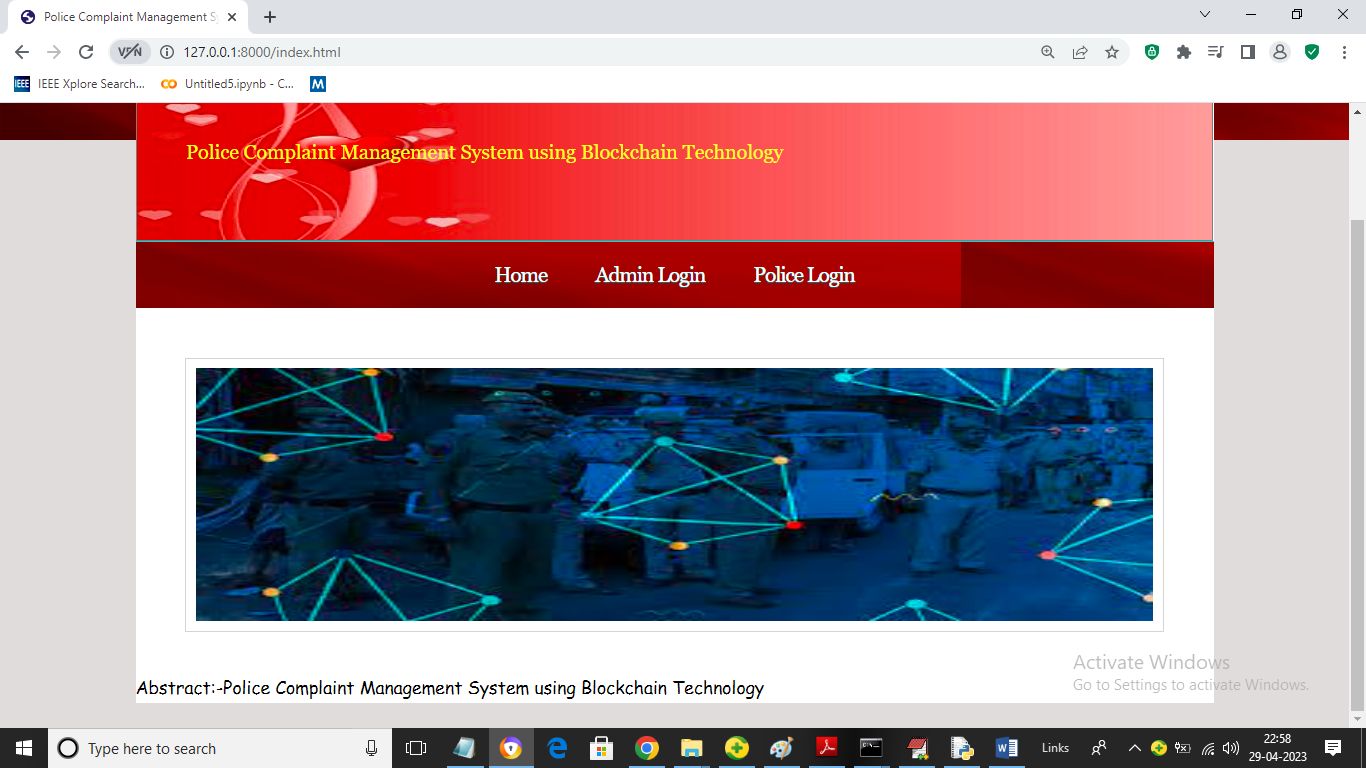
To run project double click on ‘Start\_IPFS.bat’ file to start IPFS server and get below output



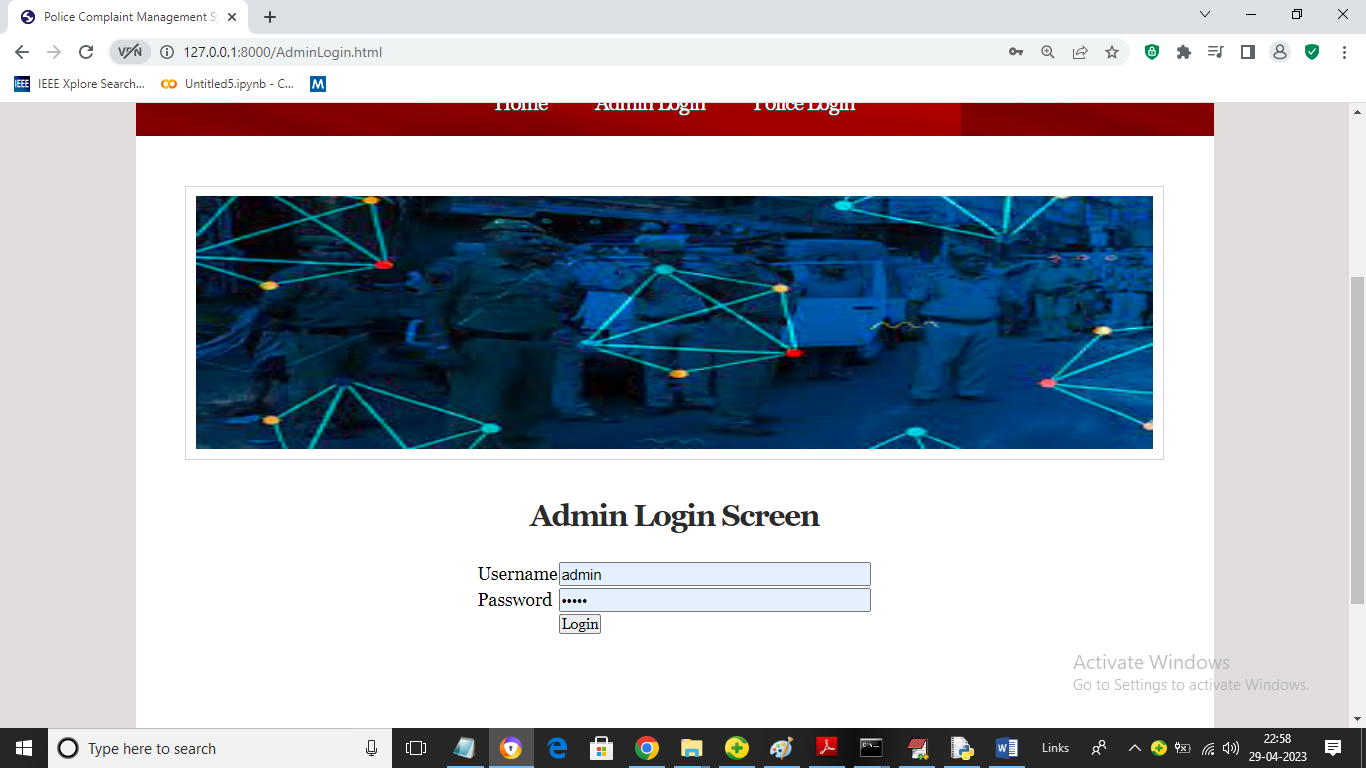
In above screen IPFS server started and now double click on ‘run.bat’ to start python WEB SERVER and get below page



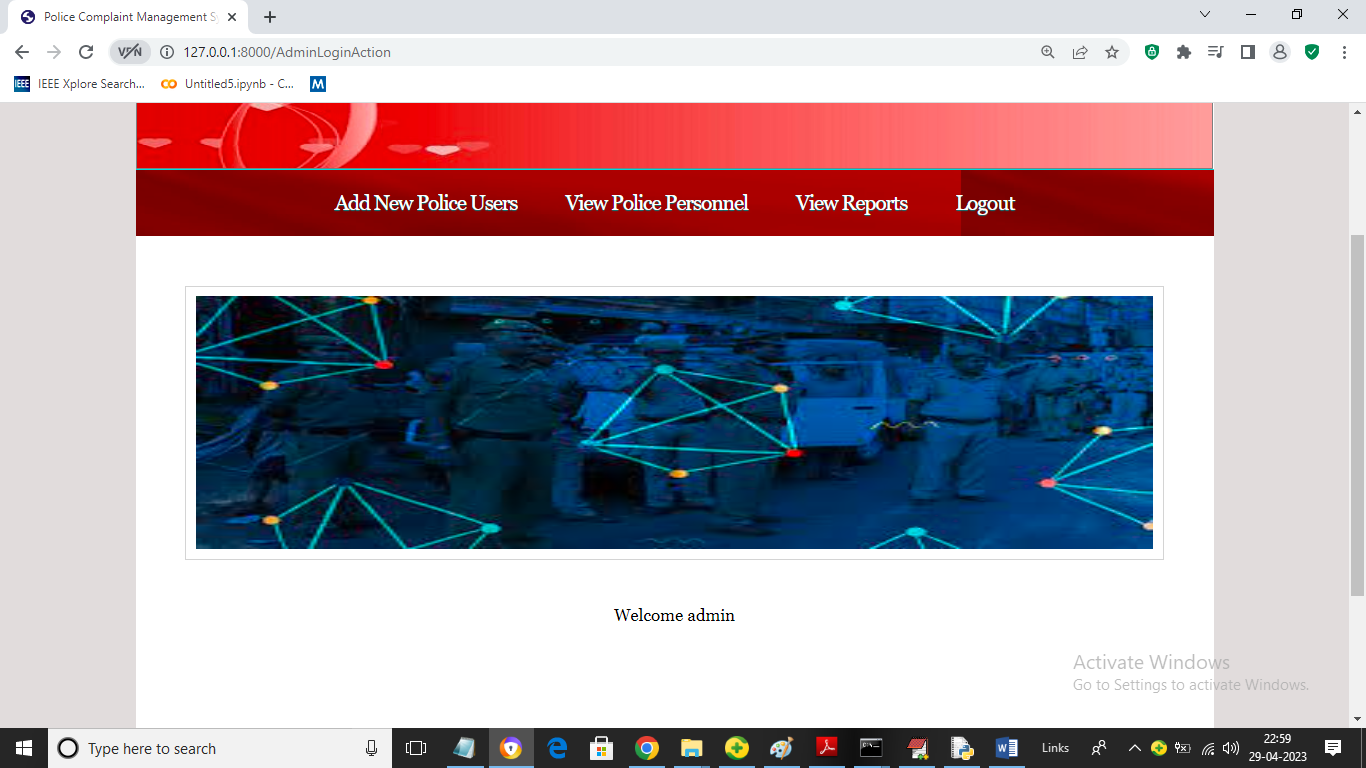
In above screen python web server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



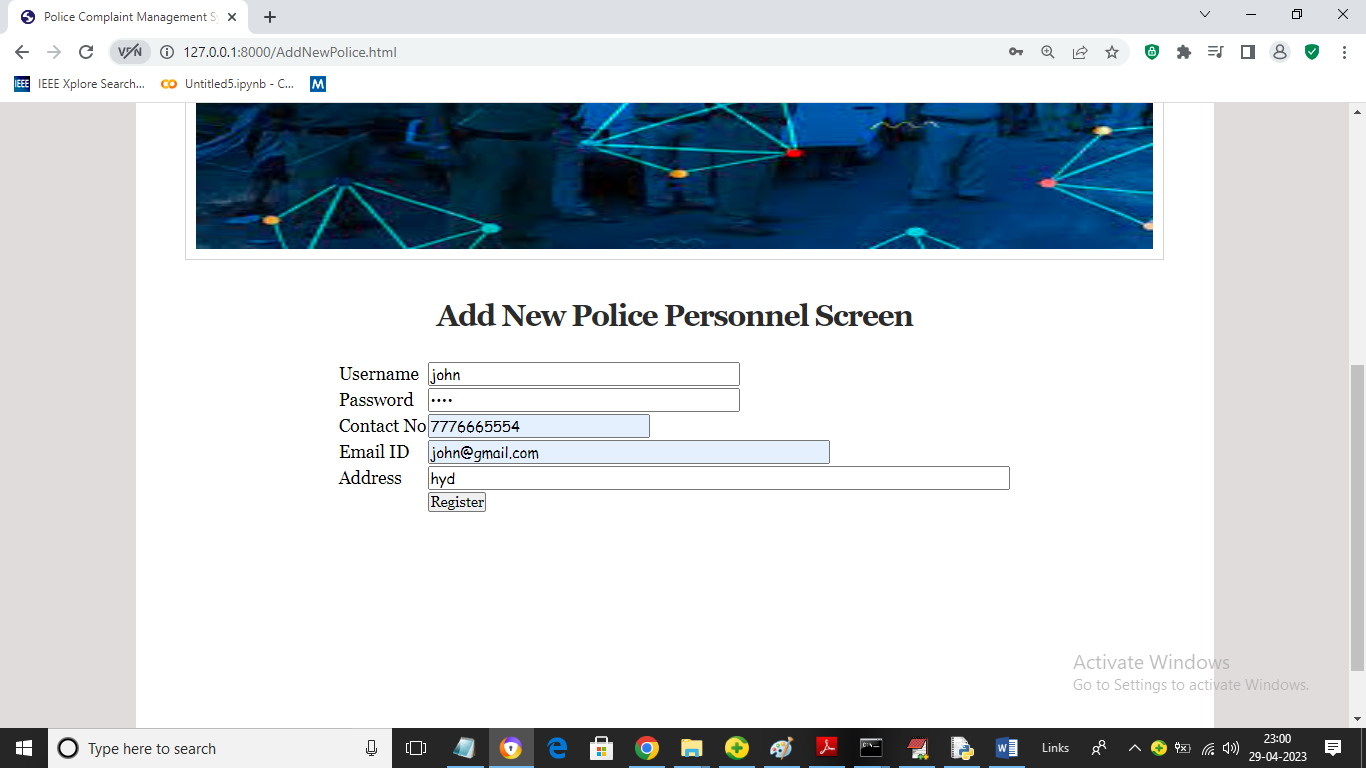
In above screen click on ‘Admin Login’ link to get below login screen



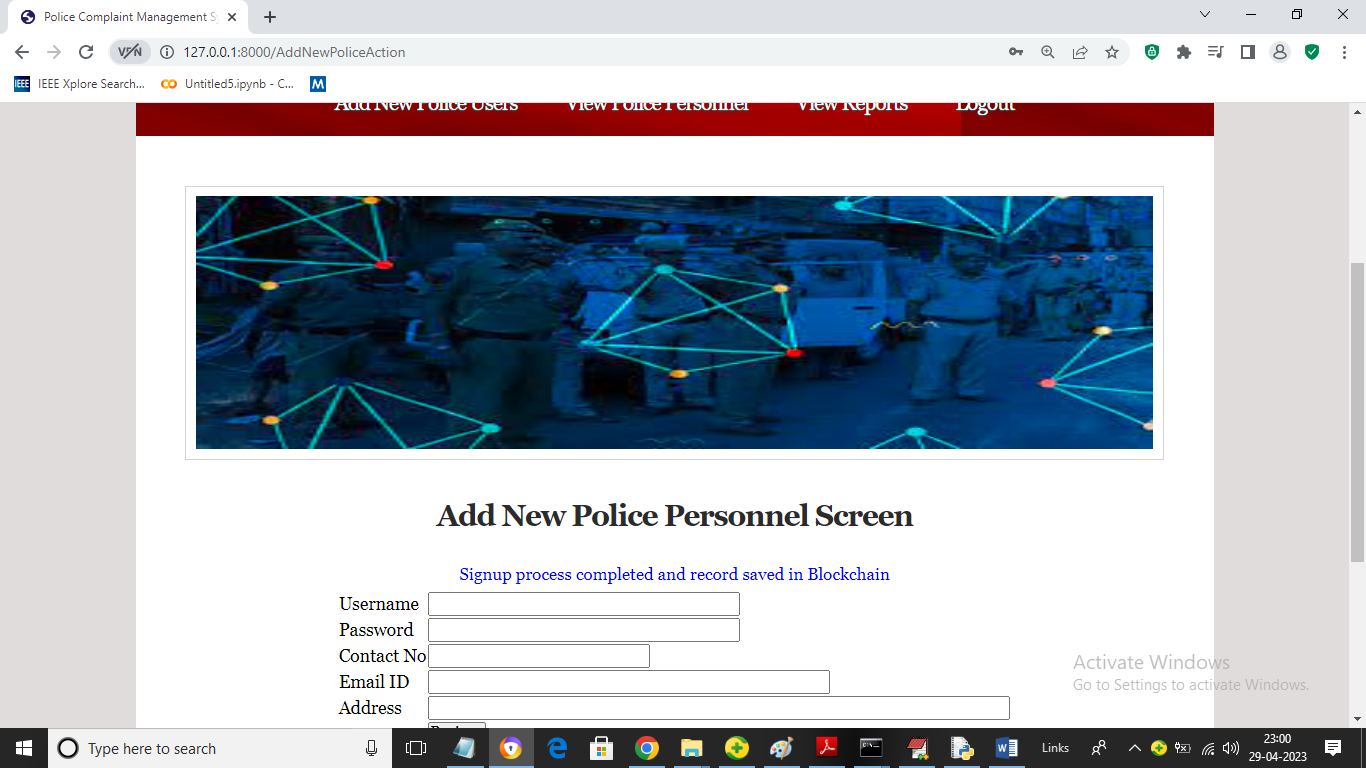
In above screen admin is login and after login will get below page



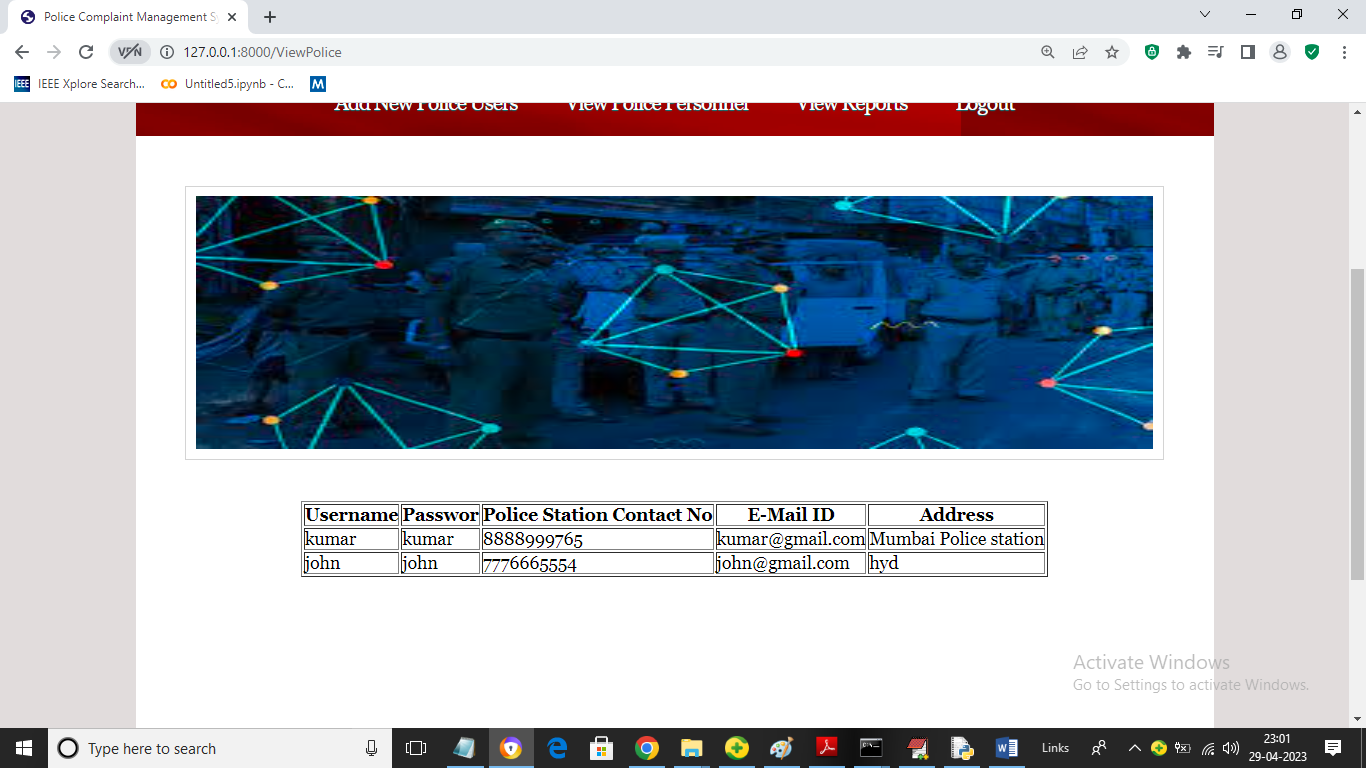
In above screen admin can click on ‘Add New Police Users’ link to get below page



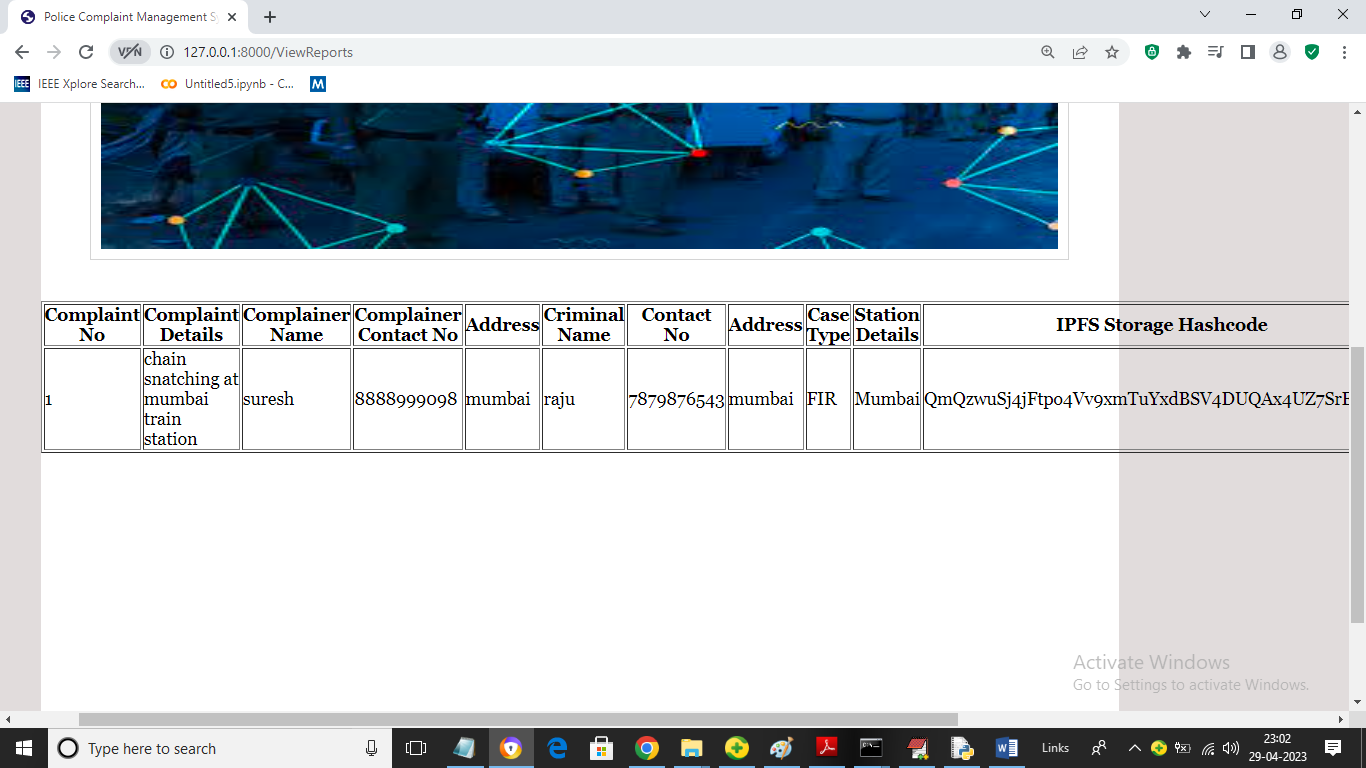
In above screen admin will add new Police personnel details and then give this login details to police personnel to add FIR details

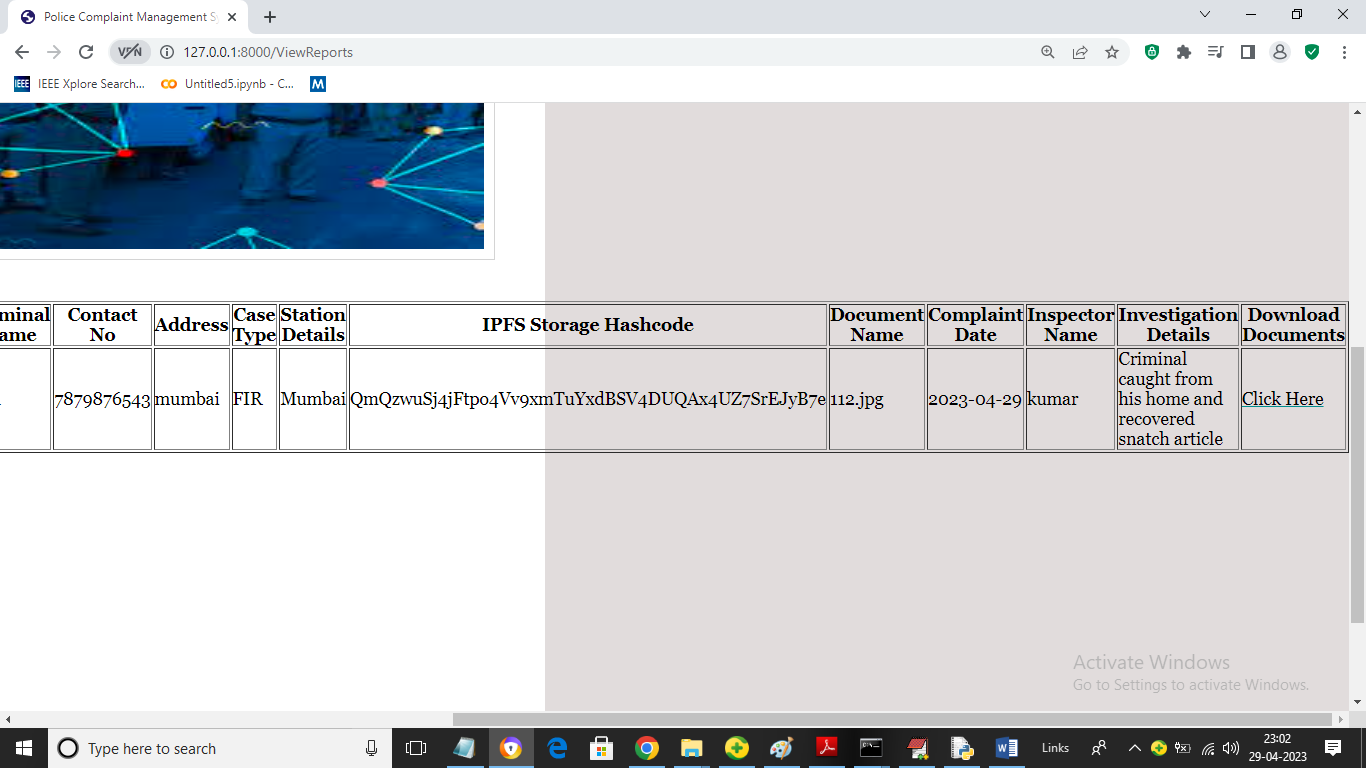


In above screen police details added in Blockchain and now click on ‘View Police Personnel’ link to view all available police men details

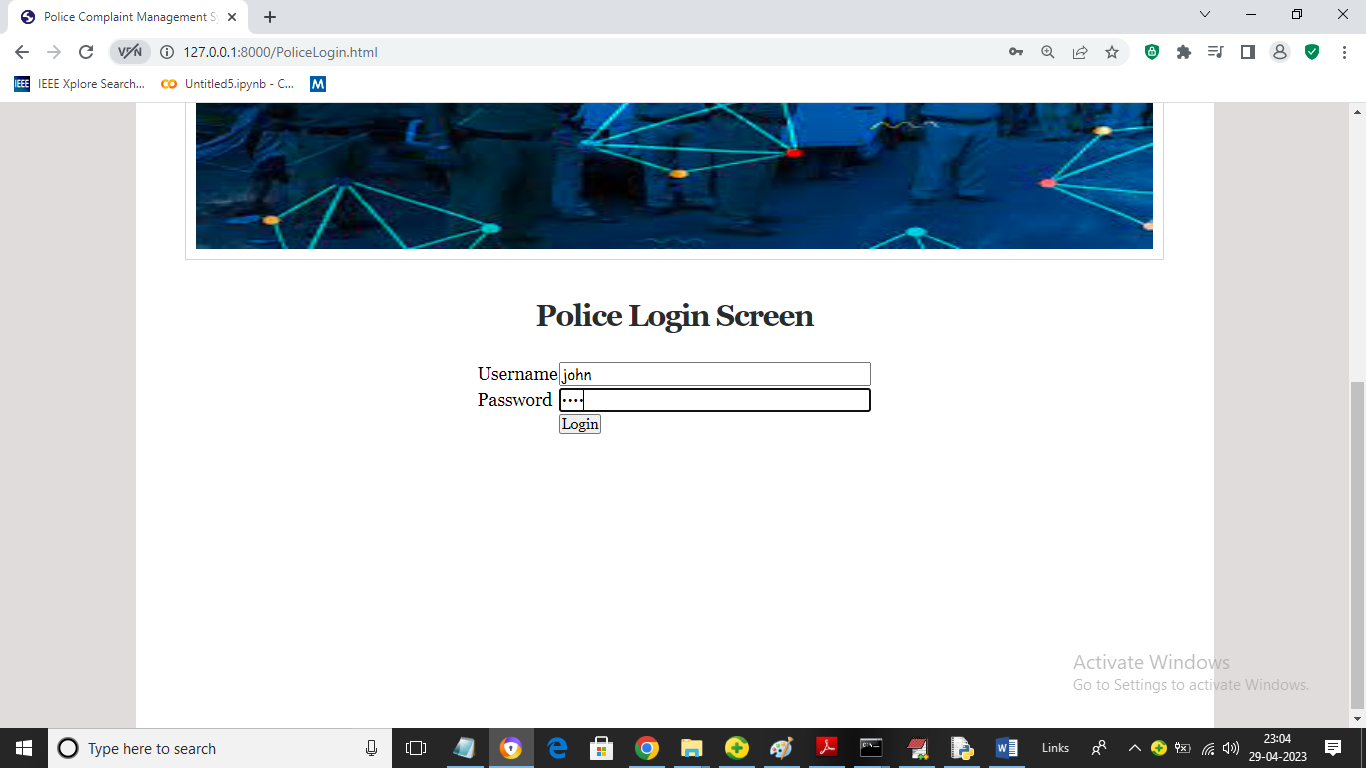


In above screen admin can view all police details and now click on ‘View Reports’ link to view all existing FIR reports and investigations

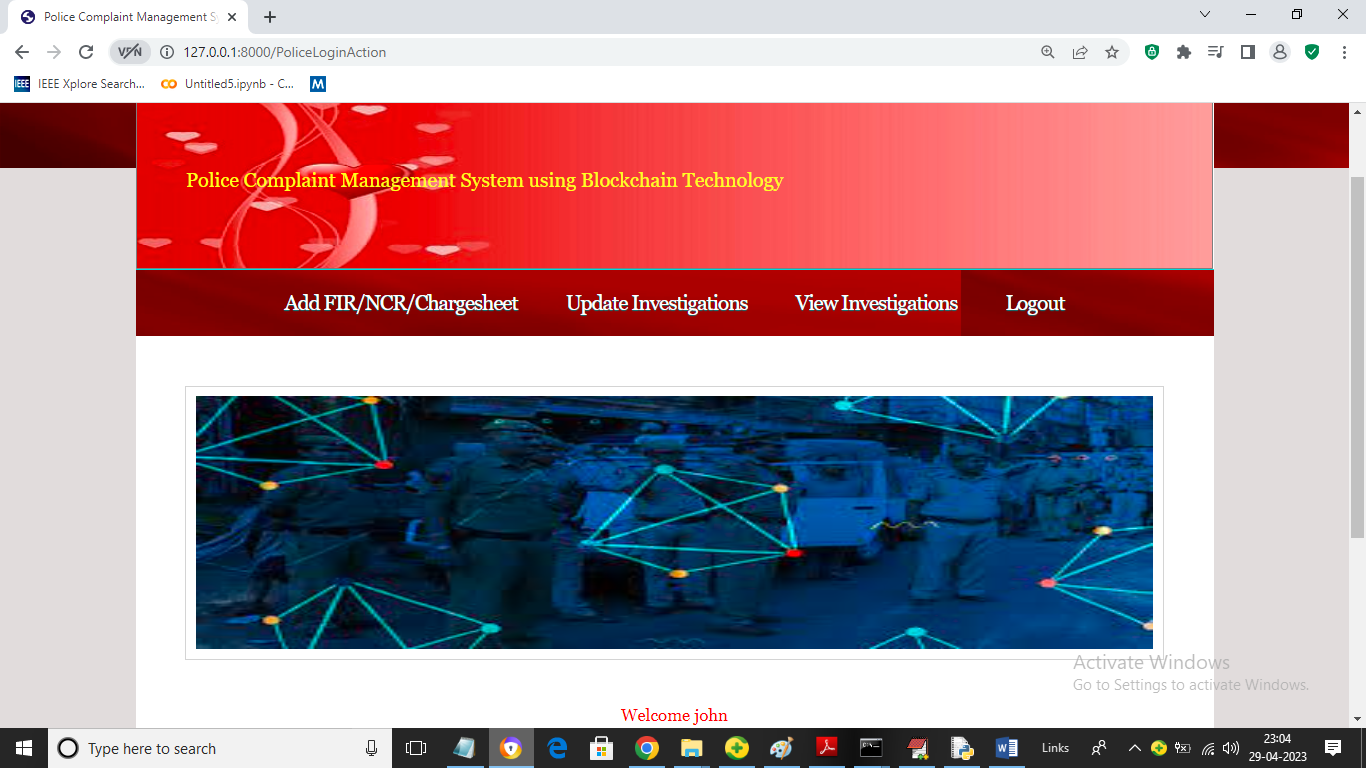




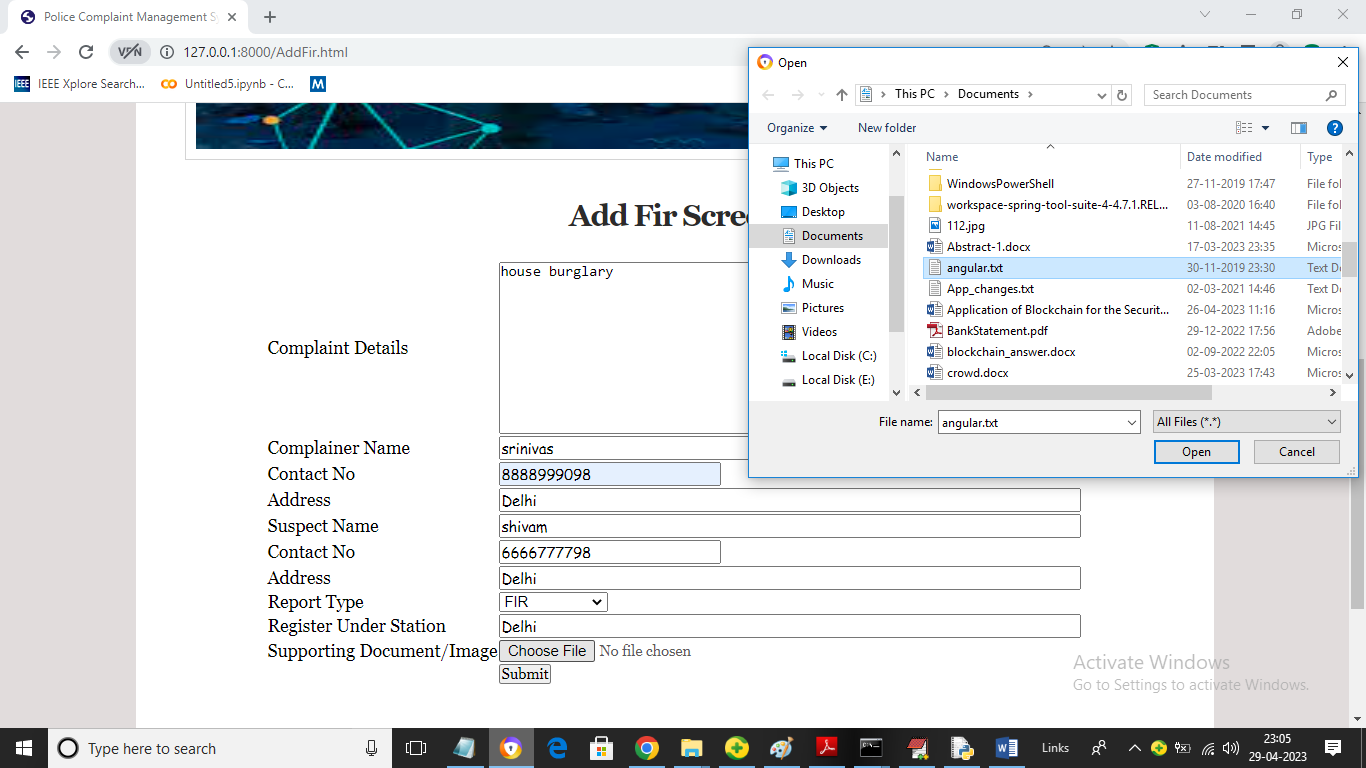
In above two screen admin can view all previously added FIR reports and can click on ‘Click Here’ link to download all crime related report and this report will saved in Blockchain in encrypted format. Now logout and login as Police personnel



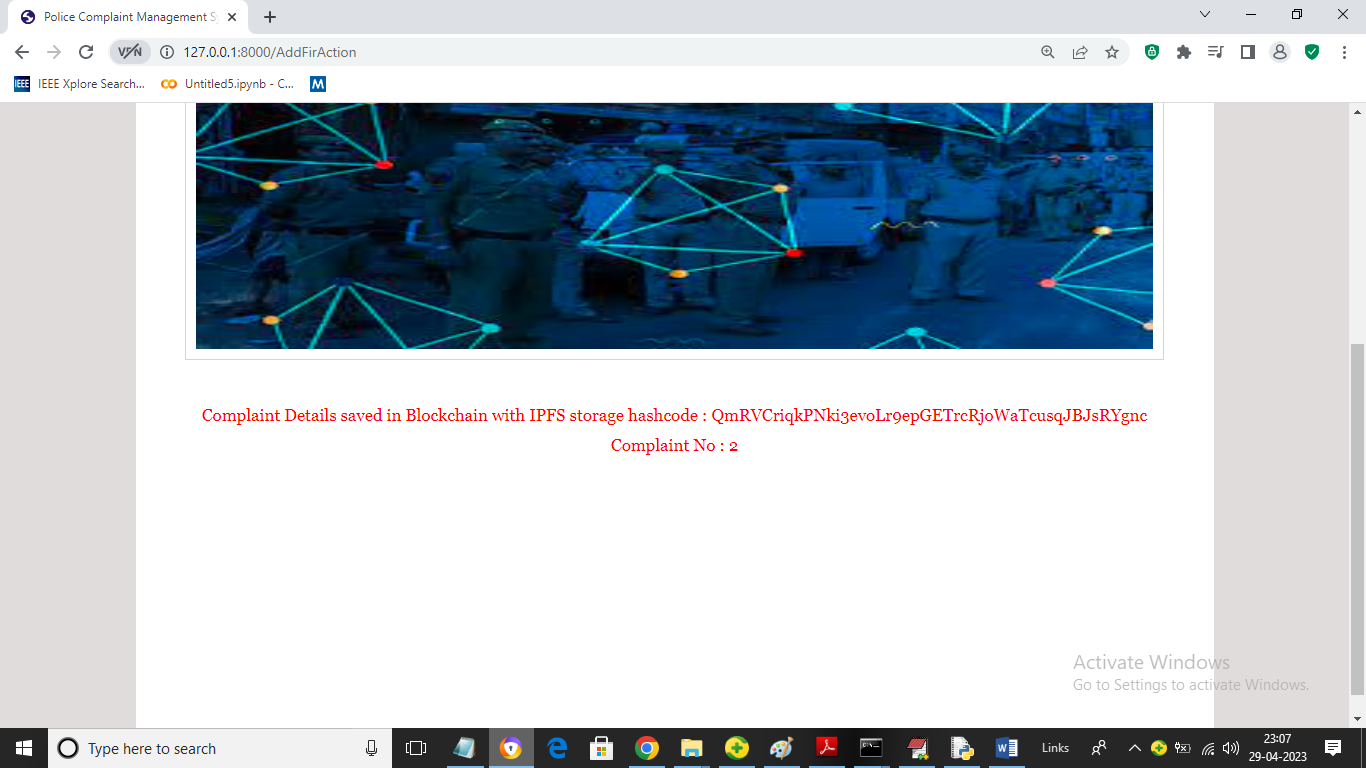
In above screen police personnel is login and after login will get below page



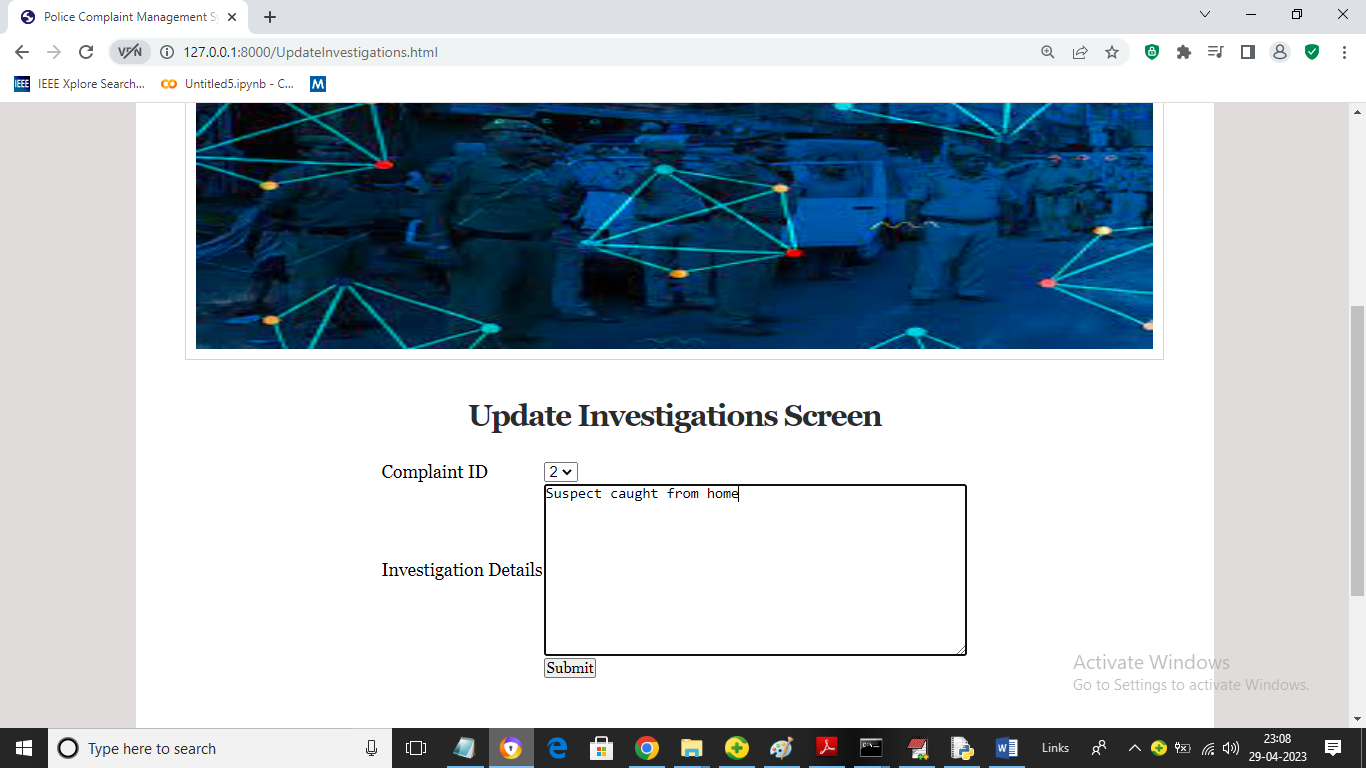
In above screen police can click on ‘Add FIR/NCR/Charge sheet’ link to add FIR



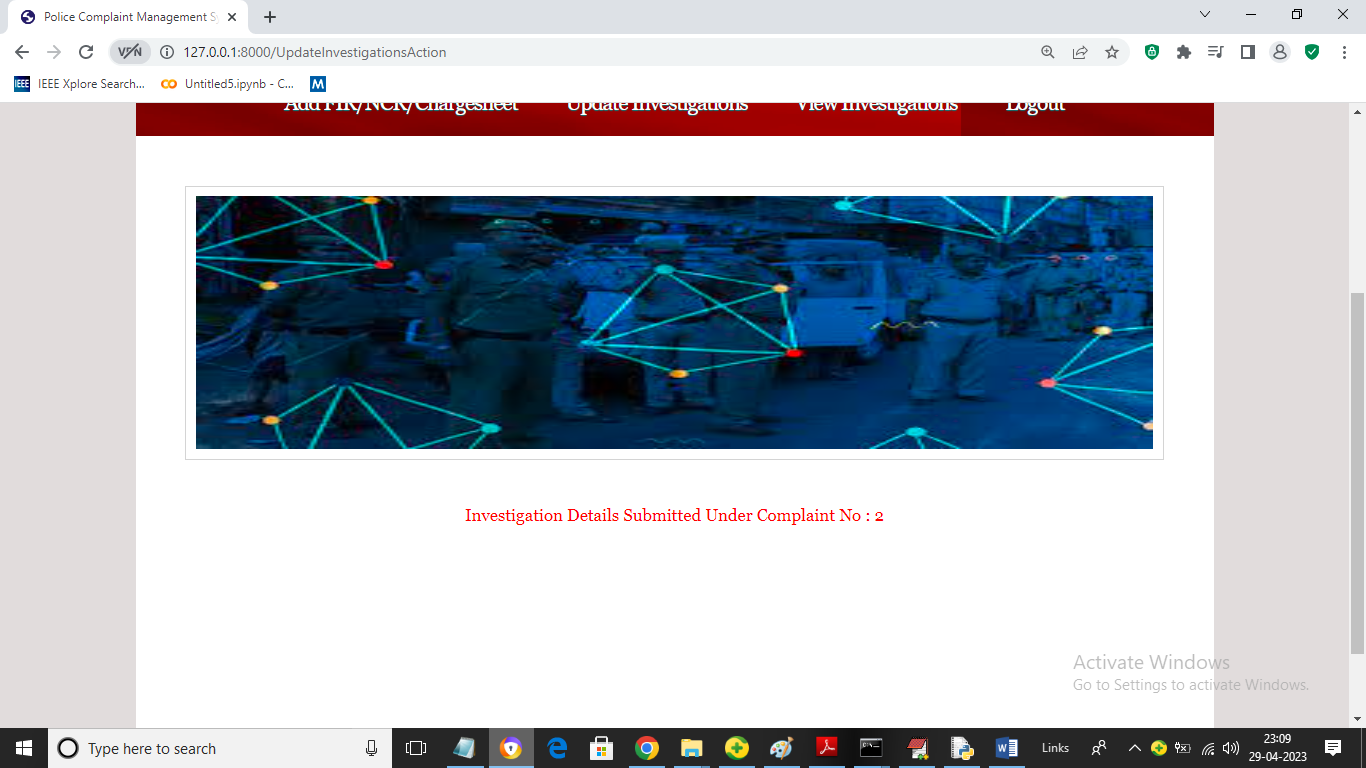
In above screen police user can add all Crime details as FIR and then upload crime related file and then click on ‘Open’ and ‘Submit’ button to get below output



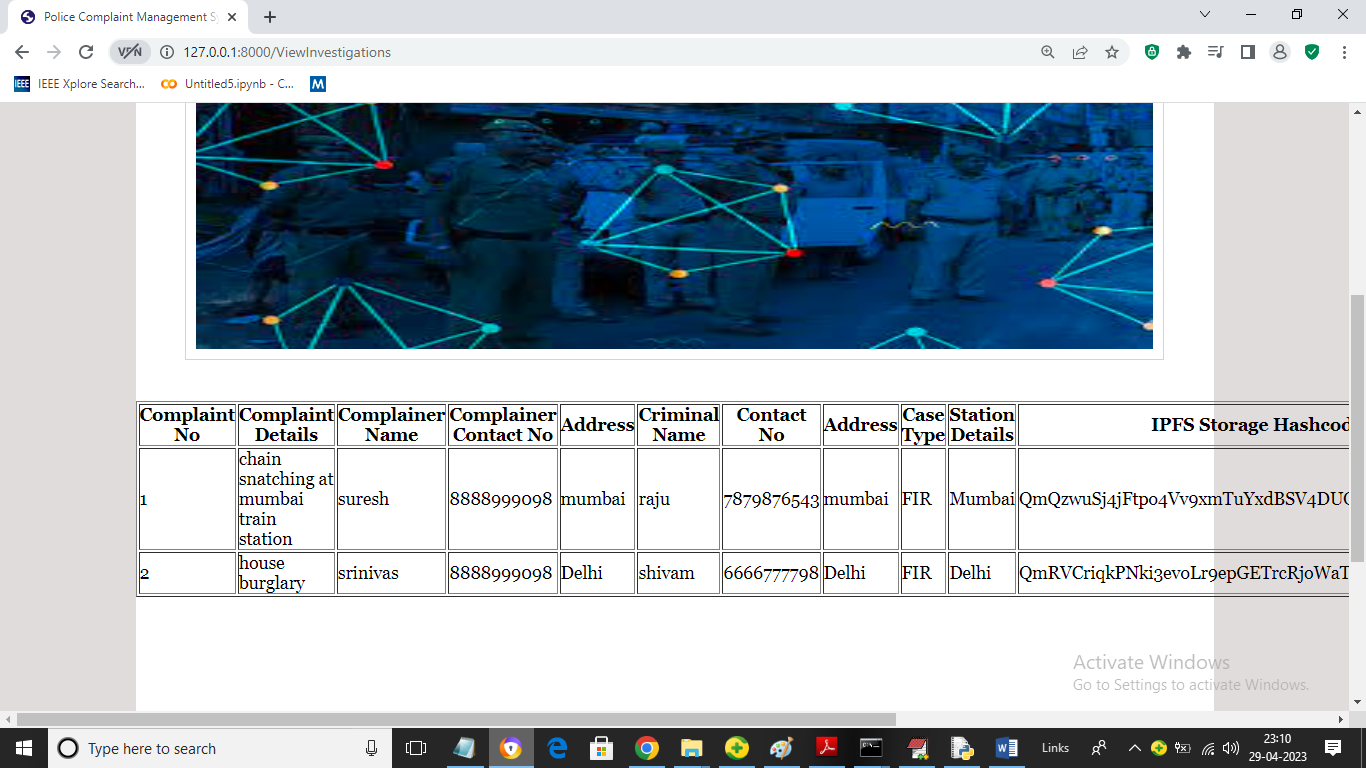
In above screen complaints details added to Blockchain and file get saved in IPFS and now police can click on ‘Update Investigation’ link to add future investigation data to current FIR like below page

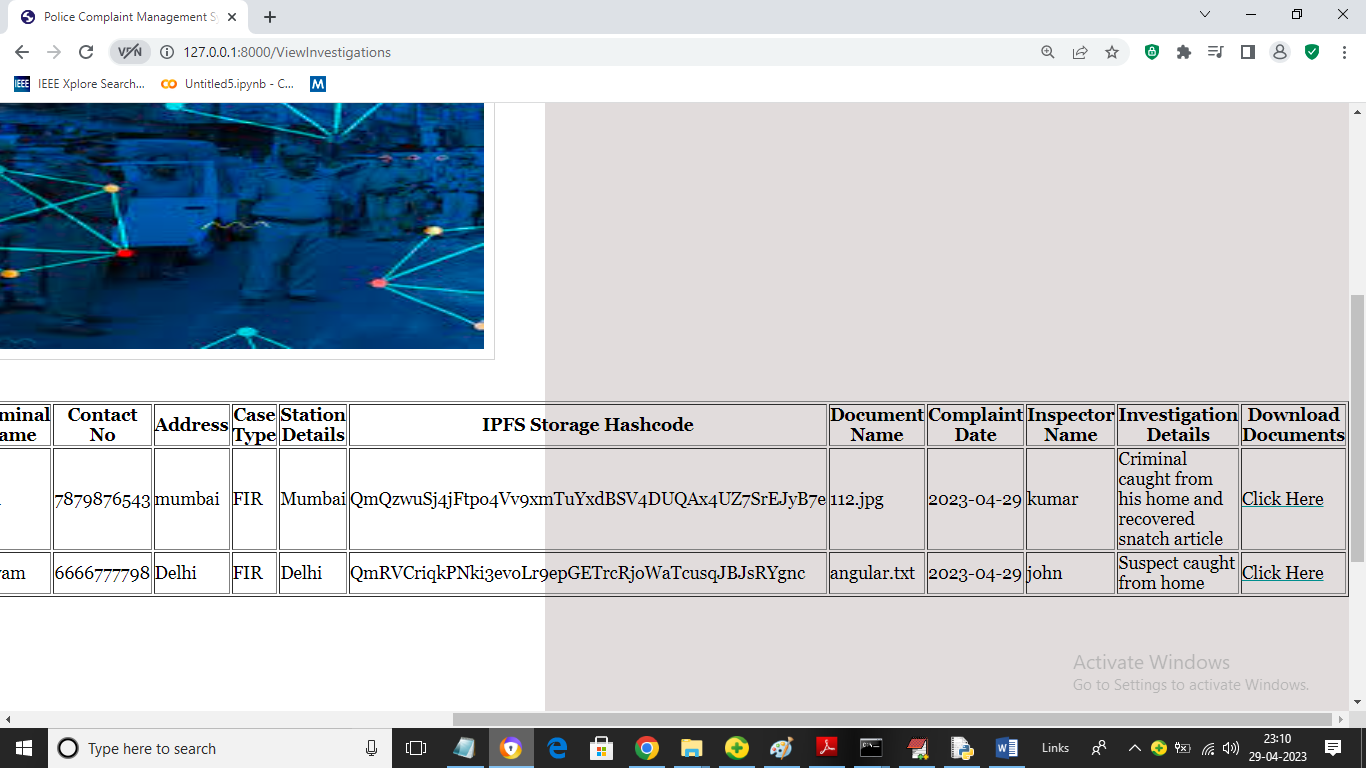


In above screen police will select complaint number and then add investigation data and then press button to save data in Blockchain and then will get below page



In above screen we can see investigation details updated and now click on ‘View Investigation’ link to get below investigated and complaint details from Blockchain





In above screen police can view all crime investigation from Blockchain from anywhere.

Similarly by following above screens you can manage all crime related data in Blockchain

**8. CONCLUSION:**

Indian Police Services are a crucial part of our country. On a yearly basis, more than 50 lakh complaints against cognizable crimes are filed. Even after having systems to manage complaints online, there is still a burden on the police officers for filing handwritten reports and fear of filing a complaint in the mind of society. Managing Police Complaints in an efficient and secure way is very crucial because it contains sensitive data. The proposed system will provide transparency while also ensuring the confidentiality of the data stored. It will also motivate people to come forward and file their complaints knowing that it cannot be ignored by the police. It will also benefit the police officers by simplifying the tedious work of filing reports such as FIRs. The decentralized network does not rely on the trust factor of the stakeholders. We proposed a system that will protect against corrupt police activities and provide justice at the very beginning.

**9. REFERENCES:**

[1]https://ncrb.gov.in/sites/default/files/Crime%20in%20India%202018%20 %20Volume%201.pdf

[2] https://www.tatatrusts.org/upload/pdf/spir-2018-common-cause.pdf

[3] http://epgp.inflibnet.ac.in/epgpdata/uploads/epgpcontent/S001608/P001746/M022182/ET/154501511Module-21-Q1.pdf

[4] http://www.thanepolice.gov.in/faq14.php

[5] <https://bitcoin.org/bitcoin.pdf>

[6] <https://indiankanoon.org/doc/760919/>

[7] Gupta, Antra and D. V´ılchez Jose. “A Method to Secure FIR System using Blockchain.”.International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019

[8] K. Tabassum, H. Shaiba, S. Shamrani and S. Otaibi, ”e-Cops: An Online Crime Reporting and Management System for Riyadh City,” 2018 1st International Conference on Computer Applications Information Security (ICCAIS), Riyadh, 2018, pp. 1-8, doi: 10.1109/CAIS.2018.8441987.

[9] Iyer A, Kathale P, Gathoo S and Surpam N 2016 E-Police SystemFIR Registration and Tracking through Android Application International Research Journal of Engineering and Technology 3(2) 1176-1179

[10] P. A. K. S. Y. K. S. , Shivaganesh Pillai, “Online Fir Registration and Sos System”, int. jour. eng. com. sci, vol. 5, no. 4, Dec. 2017. Omoregbe, Nicholas Misra, Sanjay Maskeliunas, Rytis Damasevicius, Robertas Adesola, Falade Adewumi, Adewole. (2019).

[11] Design and Implementation of an E-Policing System to Report Crimes in Nigeria. 10.1007/978-981-13-6351-1 21.

[12] Mollah, Muhammad Islam, Sikder Aman Ullah, Engr. Mohammad. (2012). Proposed e-police system for enhancement of e-government services of Bangladesh. 881-886. 10.1109/ICIEV.2012.6317444.

[13] P. Kormpho, P. Liawsomboon, N. Phongoen and S. Pongpaichet, ”Smart Complaint Management System,” 2018 Seventh ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, 2018, pp. 1-6, doi: 10.1109/ICT-ISPC.2018.85239

[14] Mollah, Muhammad Baqer Islam, Kazi Islam, Sikder. (2012). EPolice System for Improved E-Government Services of Developing Countries. Canadian Conference on Electrical and Computer Engineering. 10.1109/CCECE.2012.6335057.

[15] Onuiri, Ernest Oludele, Awodele A, Olaore O, Sowunmi A., UgoEzeaba. (2015). A REAL-TIME CRIME RECORDS MANAGEMENT SYSTEM FOR NATIONAL SECURITY AGENCIES. European Journal of Computer Science and Information Technology.

[16] Tasnim, Maisha Omar, Abdullah Rahman, Shahriar Bhuiyan, Md. (2018). CRAB: Blockchain Based Criminal Record Management System. 294-303. 10.1007/978-3-030-05345-1 25.

[17] A. T. Dini, E. Gabriel Abete, M. Colombo, J. Guevara, B. S. Menchon Hoffmann and M. Claudia Abeledo, ”Analysis of implementing ´ blockchain technology to the argentinian criminal records information system,” 2018 Congreso Argentino de Ciencias de la Informatica y ´ Desarrollos de Investigacion (CACIDI), Buenos Aires, 2018, pp. 1-3, doi: ´ 10.1109/CACIDI.2018.8584365.

[18] Swathi Sharma,Ramya Naik,“CRIME MANAGEMENT SYSTEM (CMS),” International Journal OF Current Engineering and Scientific Rese (IJCESR)

[19] Pratibha Mishra,Ghousiya Bee. N2, Mohsina S3, Mubashshira Sultana, Surbhi Singh, “Online Criminal Record Management System,” International Journal of Engineering Science and Computing, vol. 9,no. 5, May 2019.

[20] ArchanaM, Durga S, Saveetha K, “Online Crime Reporting System,” Int. Jnl. Of Advanced Networking Applications (IJANA), https://www.ijana.in/papers/82.pdf.