

A
PROJECT REPORT ON
FAKE PROFILE IDENTIFICATION USING
ARTIFICIAL NEURAL NETWORKS

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



MASTER OF COMPUTER APPLICATIONS

By

Mr. T. DILIP KUMAR

Regd. No: 235N1F0027

Under the Guidance of

Miss .V. NIRMALA,

Assistant Professor, APGCCS.

&

Mr. B. V. SAI KISHORE,

Project Leader,

Bits Computers, Kadapa.



DEPARTMENT OF COMPUTER APPLICATIONS

ANNAMACHARYA P.G COLLEGE OF COMPUTER STUDIES

NEW BOYANAPALLI-516126, RAJAMPET (A.P)

(Approved by A.I.C.T.E., New Delhi & Affiliated to J.N.T.U.A,

Anantapuramu, UGC (2f) Recognized Institution)

(2023 - 2025)

**ANNAMACHARYA P.G COLLEGE OF COMPUTER STUDIES
NEW BOYANAPALLI, RAJAMPETA, A.P, 516126.**



Affiliated to

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
ANANTAPUR, ANANTAPURAMU**

**DEPARTMENT OF
MASTER OF COMPUTER APPLICATIONS
CERTIFICATE**

This is to certify that the project work entitled "**FAKE PROFILE IDENTIFICATION USING ARTIFICIAL NEURAL NETWORKS**" is the Bonafide work carried out by **Mr. T. DILIP KUMAR**, Regd. No: **235N1F0027**, is submitted in partial fulfilment of the requirements for the award of the degree of **Master of Computer Applications** during the year **2023-2025**.

Project Guide

Principal

External Examiner

DECLARATION

I, **T. DILIP KUMAR** hereby declare that the project report entitled as "**FAKE PROFILE IDENTIFICATION USING ARTIFICIAL NEURAL NETWORKS**" done at Bits Computers, Kadapa is an original and independent record of work, submitted by me to JNTUA, Anantapuramu, under the guidance of **Miss .V. NIRMALA**, Assistant Professor of **ANNAMACHARYA P.G COLLEGE OF COMPUTER STUDIES**, Rajampet, for the award of the degree of **Master of Computer Applications** and has not been submitted either in part or in full for the award of any Degree or Diploma.

Place: Rajampet

T. DILIP KUMAR

Date:

Regd. No: 235N1F0027

ACKNOWLEDGEMENT

An endeavor over a long period can be successful only with the advice of many well-wishers. I take this opportunity to express my deep gratitude and appreciation for all those who encouraged me to successfully complete the project.

I wish to express my sincere gratitude to **Dr. D. J. SAMATHA NAIDU**, Principal of Annamacharya P.G College of Computer Studies, New Boyanapalli, Rajampet, for her consistent help and providing such facilities to complete this project.

I express my sincere thanks to my guide **Miss. V. NIRMALA**, Assistant Professor of Annamacharya P.G College of Computer Studies, New Boyanapalli, Rajampet, for her valuable guidance and suggestions in analysing and testing throughout the period of my project work.

I express my sincere thanks to my guide **Mr. B.V. SAI KISHORE**, Project leader, Bits Computers, Kadapa for his valuable guidance and suggestions in analysing and testing throughout the period of my project work.

Last but not least, I would like to thank my friends, teaching and nonteaching, one and all those who helped me to complete this project successfully.

T. DILIP KUMAR

Regd. No: 235N1F0027

ABSTRACT

Abstract**Fake Profile Identification Using
Artificial Neural Network****ABSTRACT**

Online Social Network (OSN) is a network hub where people with similar interests or real world relationships interact. Launch Fake profile is the creation of profile in the name of a person or a company which does not really exist in social media, to carry out malicious activities. Fake profiles are detected based on set of rules that can effectively classify fake and genuine profile. Using Artificial Neural Networks we are identifying whether given account details are from genuine or fake users. ANN algorithm will be trained with all previous users fake and genuine account data and then whenever we gave new test data then that ANN train model will be applied on new test data to identify whether given new account details are from genuine or fake users. We use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved. Furthermore, we discuss the sigmoid function and how the weights are determined and used. Finally, we consider the parameters of the social network page which are utmost important in the provided solution. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information.

CONTENTS

<u>Topics</u>	<u>Page No.</u>
1. INTRODUCTION	1-4
1.1. Purpose	2
1.2. Scope	2
1.3. Need for System	2
2. SOFTWARE REQUIREMENT	5-20
ANALYSIS ANDSPECIFICATION	
2.1. Related Work	5
2.1.1. Literature Survey	6
2.1.2. Existing Algorithms	9
2.2. Research Methodology	10
2.2.1. System Architecture	10
2.2.2. Proposed Algorithm	10
2.3. Proposed Modules	11
2.4. User constraints	11
2.5. Hardware Requirements	12
2.6. Software Requirements	12
2.7. Non-Functional Requirements	13
3. SYSTEM DESIGN	21-48
3.1. Database Design (ER-Model)	22
3.2. Data Dictionary	29
3.3. UML Design	37
4. TESTING	49-53
4.1. Testing Methodologies	49
4.2. Test Cases	53
5. IMPLEMENTATION	54-80
5.1. Working Model Installation Procedure	54
5.2. Sample Screens	63
6. CONCLUSION	
6.1. Future Enhancement	

BIBLIOGRAPHY

Appendix-A

- URL Listing
- Reference Books

Appendix – B

- Glossary

Appendix – C

- List of Figures
- List of Tables
- List of Screens

Appendix – D

- Coding
- Plagiarism Report

INTRODUCTION

1. INTRODUCTION

In Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Face book makes revenue via advertisements and data. More specifically, the average American user generates about \$26.76 per quarter. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Face book and Twitter. They can also target banks and other financial institutions. There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Face book had a data breach which affected about 50 million users. Face book provides a set of clearly defined provisions that explain what they do with the user's data.

The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Face book's built-in security features. The other the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media. This solution would come in the form of an algorithm. The language that we chose to use is Python. The algorithm would be able to determine if a current friend request that a user gets online is an actual person or if it is a bot or it is a fake friend request fishing for information. Our algorithm would work with the help of

the social media companies, as we would need a training dataset from them to train our model and later verify if the profiles are fake or not. The algorithm could even work as a traditional layer on the user's web browser as a browser plug-in..

1.1 PURPOSE

Fake profiles seem to slip through Facebook's built-in security features. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing..

1.2 SCOPE

To develop machine learning algorithms in predicting and detecting Fake profiles in Twitter. Fake profiles on set of rules that can effectively classify fake and genuine profiles. we use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic or not. Each equation at each neuron (node) is put through a Sigmoid function to keep the results between the interval of 0.0 and 1.0. At the output end, this could easily be multiplied by 100 to give us the possible percentage that it is a malicious request. Our solution would be only one deep neural network, meaning it only has a single hidden layer. Each input neuron would be a different, previously chosen feature of each profile converted into a numerical value (e.g., gender as a binary number, female 0 and male 1) and if needed, divided by an arbitrary number (e.g., age is always divided by 100) to minimize one feature having more influence on the result than the other.

1.3 NEED FOR SYSTEM

The neurons represent nodes. Each node would be responsible for exactly one decision-making process. Each object has a weight and bias that in turn would help in the decision-making process. The output would be the possibility in the percentage that the friend request is not from a real person. Figure 1 shows the used neural network. We would need a training data set which would be provided by Facebook or other social network sites or just web scraping given that we find enough fake profiles. This would allow our deep learning algorithm to learn the patterns of bot behavior by backpropagation, minimizing the final cost function and adjusting each

neuron's weight and bias, changing the equations. In this paper, we outline the classes and libraries involved.

EXISTING SYSTEM

In Existing system is similarity of attribute values from original and cloned profiles and the second method is based on the network relationships. A person who doubts that this profile has been cloned will be chosen as a victim. A classification method for detecting fake accounts on Twitter. They have collected some effective features for the detection process from different research and have filtered and weighted them in first stage. Malicious users create fake profiles to phish login information from unsuspecting users. A fake profile will send friend requests to many users with public profiles. These counterfeit profiles bait unsuspecting users with pictures of people that are considered attractive. Once the user accepts the request, the owner of the phony profile will spam friend requests to anyone this user is a friend. The fake profile's contents typically have links that lead to an external website where the damage happens. An unaware curious user clicking the bad link will damage their computer. The cost can be as simple as catching a virus to as bad as installing a rootkit turning the computer into a zombie. While Facebook has a rigorous screening to keep these fake accounts out, it only takes one fake profile to damage the computers of many.

DIS-ADVANTAGES

- In the existing work, the system doesn't calculate fake accounts due to lack of Attribute similarity finding.
- This system less effective due to absence of Attribute similarity which is not calculated based on the similarity of attribute values between the profiles.

PROPOSED SYSTEM

To develop machine learning algorithms in predicting and detecting Fake profiles in Twitter. Fake profiles on set of rules that can effectively classify fake and genuine profiles. we use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic or not. Each equation at each neuron (node) is put through a

Sigmoid function to keep the results between the interval of 0.0 and 1.0. At the output end, this could easily be multiplied by 100 to give us the possible percentage that it is a malicious request. Our solution would be only one deep neural network, meaning it only has a single hidden layer. Each input neuron would be a different, previously chosen feature of each profile converted into a numerical value (e.g., gender as a binary number, female 0 and male 1) and if needed, divided by an arbitrary number (e.g., age is always divided by 100) to minimize one feature having more influence on the result than the other. The neurons represent nodes. Each node would be responsible for exactly one decision-making process. Each object has a weight and bias that in turn would help in the decision-making process. The output would be the possibility in the percentage that the friend request is not from a real person. It is used neural network, We would need a training data set which would be provided by Facebook or other social network sites or just web scraping given that we find enough fake profiles. This would allow our deep learning algorithm to learn the patterns of bot behavior by backpropagation, minimizing the final cost function and adjusting each neuron's weight and bias, changing the equations. In this paper, we outline the classes and libraries involved.

ADVANTAGES

- Fake profiles have become a very serious social threat.
- As information like phone number, email id, school or college name, company name, location etc are readily exposed in social networks, hackers can easily hack this information to create fake or clone profiles.
- A detection method has been proposed which can detect both fake and clone profiles in order to make the social life of the users more secure.
- The architecture of proposed system is as shown in the proposed system.
- Accuracy which gives the ratio of number of correct results to the total number of inputs.
- Precision which gives the proportion of positive detection that was actually correct.
- Recall which gives the proportion of actual positives that was detected correctly.

**SOFTWARE REQUIREMENT
ANALYSIS AND SPECIFICATION**

2. SOFTWARE REQUIREMENT ANALYSIS AND SPECIFICATION

2.1. RELATED WORK

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data. More specifically, the average American user generates about \$26.76 per quarter. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions. There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users. Facebook provides a set of clearly defined provisions that explain what they do with the user's data. The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse, is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media. This

solution would come in the form of an algorithm. The language that we chose to use is Python. The algorithm would be able to determine if a current friend request that a user gets online is an actual person or if it is a bot or it is a fake friend request fishing for information. Our algorithm would work with the help of the social media companies, as we would need a training dataset from them to train our model and later verify if the profiles are fake or not. The algorithm could even work as a traditional layer on the user's web browser as a browser plug-in.

2.1.1 LITERATUER SURVEY

[1]. P. K. Jain, V. Saravanan, and R. Pamula, “A hybrid cnn-lstm: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 5, pp. 1–15, 2021.

With the fastest growth of information and communication technology (ICT), the availability of web content on social media platforms is increasing day by day. Sentiment analysis from online reviews drawing researchers' attention from various organizations such as academics, government, and private industries. Sentiment analysis has been a hot research topic in Machine Learning (ML) and Natural Language Processing (NLP). Currently, Deep Learning (DL) techniques are implemented in sentiment analysis to get excellent results. This study proposed a hybrid convolutional neural network-long short-term memory (CNN-LSTM) model for sentiment analysis. Our proposed model is being applied with dropout, max pooling, and batch normalization to get results. Experimental analysis carried out on Airlinequality and Twitter airline sentiment datasets. We employed the Keras word embedding approach, which converts texts into vectors of numeric values, where similar words have small vector distances between them. We calculated various parameters, such as accuracy, precision, recall, and F1-measure, to measure the model's performance. These parameters for the proposed model are better than the classical ML models in sentiment analysis. Our results analysis demonstrates that the proposed model outperforms with 91.3% accuracy in sentiment analysis.

[2] P. K. Jain, R. Pamula, and G. Srivastava, “A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews,” *Computer Science Review*, vol. 41, no. 1, 2021.

Consumer sentiment analysis is a recent fad for social media-related applications such as healthcare, crime, finance, travel, and in academia. Disentangling consumer perception to gain insight into the desired objective and reviews is significant. With the advancement of technology, a massive amount of social web data increasing in volume, subjectivity, and heterogeneity becomes challenging to process manually. Machine learning (ML) techniques have been utilized to handle this difficulty in real-life applications. We study to determine the usefulness, scope, and applicability of this alliance of ML techniques for consumer sentiment analysis (CSA) for online reviews in the domain of hospitality and tourism. We show a systematic literature review to compare, analyse, explore, and understand the attempts and directions to find research gaps in illustrating the future scope of this pairing. The primary objective is to read and analyse the use of ML techniques for consumer sentiment analysis on online reviews in the domain of hospitality and tourism. This research has significant implications for service providers in terms of developing managerial strategies for consumers in terms of selecting services that meet their needs. Furthermore, there is high impact for researchers in terms of prospective research directions.

[3] M. Fazil and M. Abulaish, “A hybrid approach for detecting automated spammers in twitter,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2707–2719, 2018.

Twitter is one of the most popular microblogging services, which is generally used to share news and update through short messages restricted to 280 characters. However, its open nature and large user base are frequently exploited by automated spammers, content polluters, and other ill-intended users to commit various cyber crimes, such as cyberbullying, trolling, rumor dissemination, and stalking. Accordingly, a number of approaches have been proposed by researchers to address these problems. However, most of these approaches are based on user

characterization and completely disregarding mutualinteractions. In this study, we present a hybrid approach fordetecting automated spammers by amalgamating community-based features with other feature categories, namely metadata-,content-, and interaction-based features. The novelty of theproposed approach lies in the characterization of users basedon their interactions with their followers given that a usercan evade features that are related to his/her own activities,but evading those based on the followers is difficult. Nineteendifferent features, including six newly defined features and tworedefined features, are identified for learning three classifiers,namely, random forest,decision tree, and Bayesian network,on a real dataset that comprises benign users and spammers.The discrimination power of different feature categories is alsoanalyzed, and interaction- and community-based features aredetermined to be the most effective for spam detection, whereasmetadata-based features are proven to be the least effective.

[4]. M. Abulaish and M. Fazil, “Socialbots: Impacts, threat-dimensions, and defense challenges,” IEEE Technology and Society Magazine, vol. 39, no. 3, pp. 52–61, 2020.

Online Social Networks (OSNs) are the modern communication media that are under threat by socialbots. The rise of socialbots and their role has posed diverse challenges in the contexts of political astroturfing, fake news, and spear phishing. This article presents a concise and multifaceted study of socialbots. It commences with a comparative analysis of social botnets with conventional web botnets and further presents an experimental analysis of socialbots’ impact on network infiltration. We also categorize the threat landscape of socialbots into four dimensions and present a detailed discussion of each threat-dimension and its impact on different OSN stakeholders. It presents different categories of defense challenges against the socialbots to understand the complications of the problem, which will help in devising future mitigation strategies. Finally, we present a brief overview of the current trends in the direction of socialbot research and their role in the context of COVID-19 pandemic

[5] M. Abulaish, A. Kamal, and M. J. Zaki, “A survey of figurative language and its computational detection in online social networks,” *ACM Transaction on the Web*, vol. 14, no. 1, pp. 1–52, Jan. 2020.

The frequent usage of figurative language on online social networks, especially on Twitter, has the potential to mislead traditional sentiment analysis and recommender systems. Due to the extensive use of slangs, bashes, flames, and non-literal texts, tweets are a great source of figurative language, such as sarcasm, irony, metaphor, simile, hyperbole, humor, and satire. Starting with a brief introduction of figurative language and its various categories, this article presents an in-depth survey of the state-of-the-art techniques for computational detection of seven different figurative language categories, mainly on Twitter. For each figurative language category, we present details about the characterizing features, datasets, and state-of-the-art computational detection approaches. Finally, we discuss open challenges and future directions of research for each figurative language category

2.1.2 EXISTING ALGORITHMS/TECHNIQUES

Due to the immense growth of e-commerce and increased online based payment possibilities, credit card fraud has become deeply relevant global issue. Recently, there has been major interest for applying machine learning algorithms as data mining technique for credit card fraud detection. However, number of challenges appear, such as lack of publicly available data sets, highly imbalanced class sizes, variant fraudulent behavior etc. we compare performance of three machine learning algorithms: Random Forest, Support Vector Machine and Logistic Regression in detecting fraud on real-life data containing credit card transactions. To mitigate imbalanced class sizes, we use SMOTE sampling method. The problem of ever-changing fraud patterns is considered with employing incremental learning of selected ML algorithms in experiments. The performance of the techniques is evaluated based on commonly accepted metric: precision and recall.

However, number of challenges appear, such as lack of publicly available data sets, highly imbalanced class sizes, variant fraudulent behavior etc. we compare performance of three machine learning algorithms: Random

Forest, Support Vector Machine and Logistic Regression in detecting fraud on real-life data containing credit card transactions.

2.2 RESEARCH METHODOLOGY

2.2.1 SYSTEM ARCHITECTURE

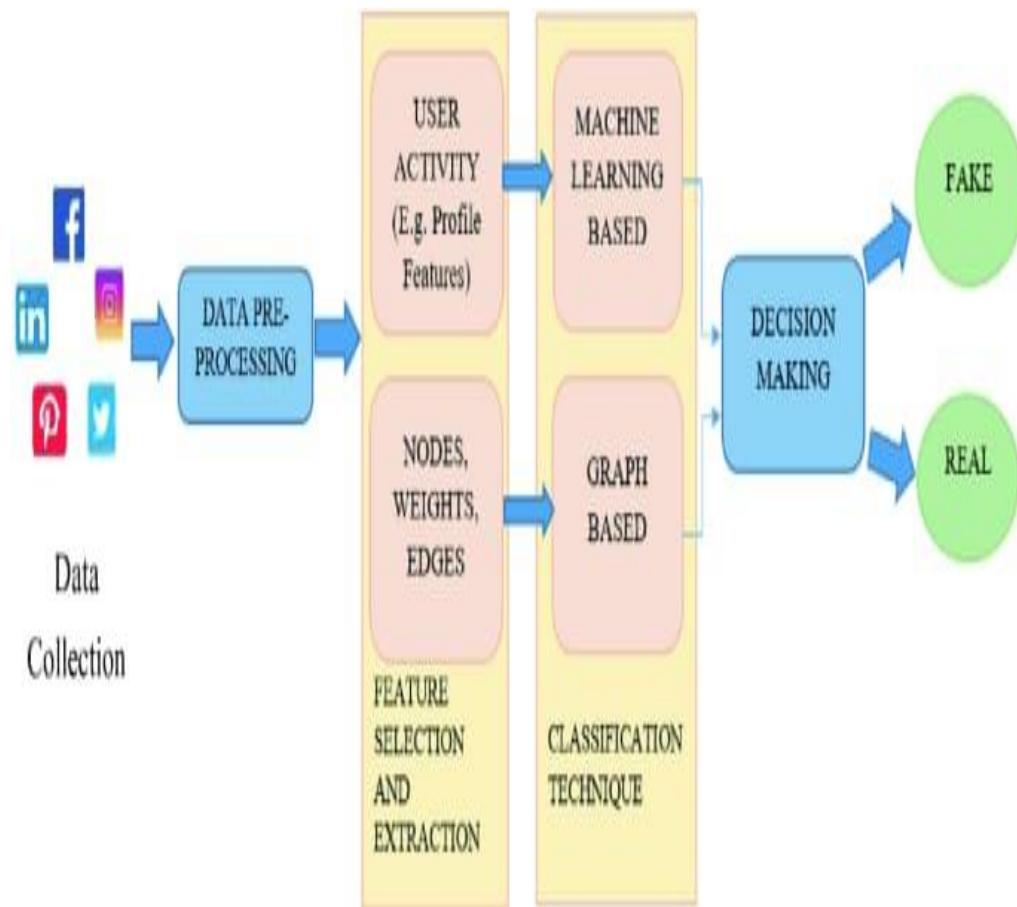


Fig.2.2.1.1: System Architecture

2.2.2 PROPOSED ALGORITHMS/TECHNIQUE

We can understand the working of the ANN algorithm with the help of following

steps:

Step 1 – First, start with the selection of ANN samples from a given dataset.

Step 2 – Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 – In this step, voting will be performed for every predicted result.

Step 4 – At last, select the most voted prediction result as the final prediction result

2.3 PROPOSED MODULE

SERVICE PROVIDER Module:

Admin will login to application by using username as ‘admin’ and password as ‘admin’ and then perform below actions.

Generate SVM & NAVI BAYES Train Model:

Admin will upload profile dataset to SVM & NAVI BAYES algorithm to build train model. This train model can be used to predict fake or genuine account by taking new account test data.

View SVM & NAVI BAYES Train Dataset:

Using this module admin can view all dataset used to train SVM & NAVI BAYES model.

User Module:

Any user can use this application and enter test data of new account and call SVM & NAVI BAYES algorithm. SVM & NAVI BAYES algorithm will take new test data and applied train model to predict whether given test data contains fake or genuine details.

2.4 USER CONSTRAINTS

User Constraints for project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL CONSTRAINTS
- TECHNICAL CONSTRAINTS
- SOCIAL CONSTRAINTS

ECONOMICAL CONSTRAINTS

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within

the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL CONSTRAINTS

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL CONSTRAINTS

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE REQUIREMENTS

Processor	:	I3 or higher
Speed	:	2.9 GHz
RAM	:	4 GB (min)
Hard Disk	:	160 GB

2.6 SOFTWARE REQUIREMENTS

- Operating system : Windows 7 Ultimate
- Coding Language : Python
- Back-End : Django-ORM
- Designing : Html, css, javascript
- Data Base : MySQL (WAMP Server)

Functional Requirements

Functional requirements describe what the system should do. The functional requirements can be further categorized as follows:

- What inputs the system should accept?

- What outputs the system should produce?
- What data the system must store?
- What are the computations to be done?

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and the steps are necessary to put transaction data in to a usable form for processing that can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

1. What data should be given as input?
2. How the data should be arranged or coded?
3. The dialog to guide the operating personnel in providing input.
4. Methods for preparing input validations and steps to follow when error occur.

2.7 Non-Functional Requirements

Non-functional requirements are the constraints that must be adhered during development. They limit what resources can be used and set bounds on aspects of the software's quality.

User Interfaces

The User Interface is a GUI developed using Python.

Software Interfaces

The main processing is done in Java and console application.

Manpower Requirements

5 members can complete the project in 2 – 4 months if they work fulltime on it.

Description of non-functional requirements is just as critical as functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

Non-functional requirements describe user-visible aspects of the system that are not directly related to functionality of the system. Non-functional requirements these are constraints on the services or functions offered by the System. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements". Qualities, that are non-functional requirements, can be divided into two main categories.

Execution qualities, such as security and usability, which are observable at run time. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

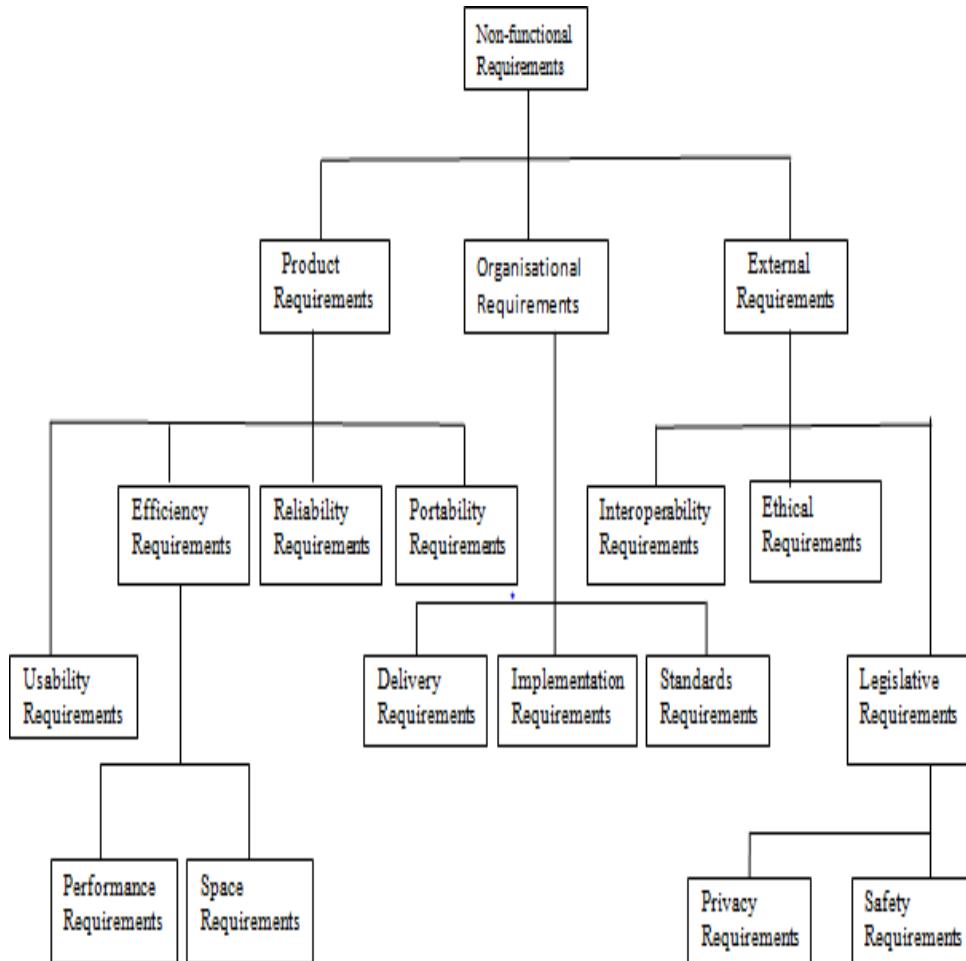


Fig.2.7.1: Non-Functional Requirements

Usability

- As it is an Internet Application, must have some usability Features. End users of this System are Unlimited and from Various Skilled groups, so that we can't restrict them. By providing some facilities we have to make them comfortable.
- Colors what we use in this Web Portal design must be attractive.
- Easy Navigations are Preferable to do any task.
- Home page Should be Centralized System (Screen/Window) to go to any feature and to get any result.
- The facility to return to Home page from any page are Should available.

Serviceability requirement

Serviceability requirements refer to the specific criteria or standards that a product or system must meet to ensure it can be maintained, repaired, or serviced effectively throughout its lifecycle. These requirements are important to ensure that the product can be kept in good working condition, address any issues or failures that may arise, and minimize downtime or disruptions.

Manageability requirement

Manageability requirements refer to the specific criteria or capabilities that a system or software application should possess to facilitate its efficient management, administration, and monitoring. These requirements are essential for ensuring that the system can be effectively controlled, configured, and maintained by administrators or operators.

Recoverability requirement

Recoverability requirements refer to the specific criteria or capabilities that a system or software application should possess to enable the recovery of data, services, or functionality in the event of a failure, error, or disaster. These requirements are essential for ensuring that the system can be restored to a functional state with minimal downtime and data loss.

Security requirement

The web server and database server should be protected from hacking, virus etc.

Data integrity requirement

Data integrity requirements refer to the criteria or measures that must be implemented to ensure the accuracy, consistency, and reliability of data within a system or database. These requirements are crucial for maintaining data quality, preventing unauthorized modifications or corruption, and enabling trust in the information stored or processed by the system.

Capacity requirement

Capacity requirements refer to the specific criteria or capabilities that a system or infrastructure must possess to handle anticipated workloads, data volumes, or user demands within acceptable.

performance parameters. These requirements are essential for ensuring that the system can effectively support the required scale and growth.

Availability requirement

The system is implemented based on the web browser and server. Using this web browser the user can access the data and store the data in the server; here we can use the web browser as Mozilla and server as Tomcat.

Scalability requirement

Scalability requirements refer to the specific criteria or capabilities that a system or infrastructure must possess to accommodate increasing workloads, data volumes, or user demands without experiencing significant degradation in performance or resource constraints. Scalability is essential for ensuring that the system can grow and adapt to changing requirements, allowing it to handle increased workloads or accommodate additional users without compromising performance or stability.

Interoperability requirement

Interoperability requirements refer to the specific criteria or capabilities that a system or software application must possess to interact, communicate, and exchange data effectively with other systems, software components, or external entities. Interoperability is essential for seamless integration and collaboration between different systems or components, allowing them to work together efficiently and share information without compatibility issues or data loss.

Reliability requirement

Reliability requirements refer to the specific criteria or characteristics that a system or software application must possess to consistently perform its intended functions without failure or interruption. Reliability is crucial for ensuring that the systems.

Maintainability requirement

The first tier is the GUI, which is said to be front-end and the second tier is the database, which uses MYSQL, which is the back-end. The front-end can be run on different systems (clients).

Regulatory requirement

Regulatory requirements refer to the specific criteria or obligations that a system, software application, or organization must comply with based on legal, industry, or government regulations. These requirements are designed to ensure that systems or organizations operate in accordance with applicable laws, standards, and guidelines to protect users, consumers, and stakeholders, as well as to maintain ethical and legal standards.

SDLC Methodologies

SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. The intent of a SDLC process it to help produce a product that is cost-efficient, effective, and of high quality. The SDLC methodology usually contains the following stages:

1. Requirement Gathering
2. System Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance
7. SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. The methodology within the SDLC process can vary across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems. SDLC consists of following activities:
The sequential phases in Waterfall model are:

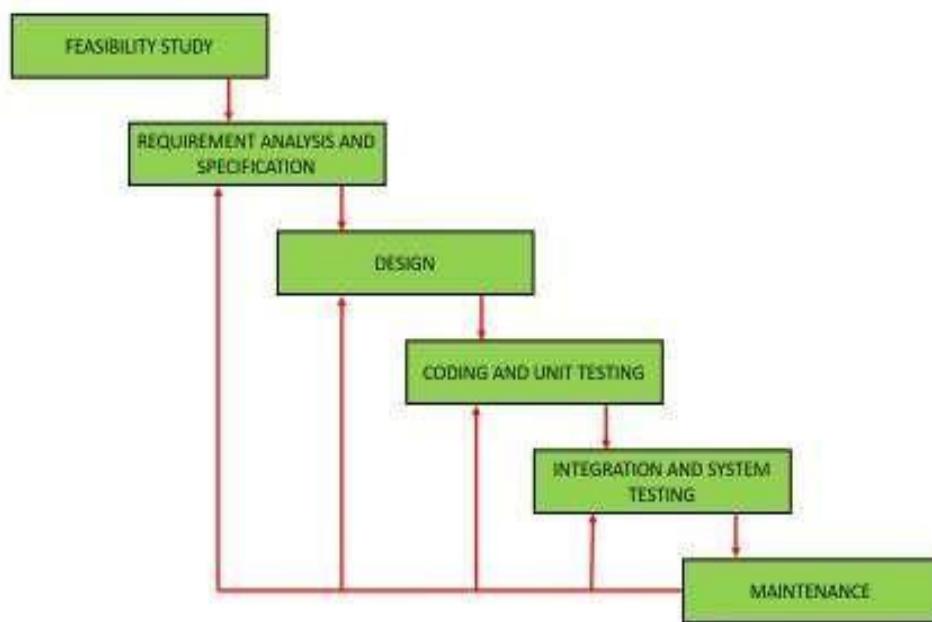


Fig. 2.7.2: Water Fall Model

Requirement Gathering and analysis:

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification.

System Design:

The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

Implementation:

With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

Integration and Testing:

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment of system:

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

Maintenance:

There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Advantages:

- The waterfall model is a simple model.
- It is easily understood as all the phases are done step by step.
- No complexity as the deliverables of each phase are well defined.

Disadvantages:

- This model cannot be used for the Project wherein the requirement is not clear or the requirement keeps on changing.
- A working model can only be available once the software reaches at last stage of the cycle.
- It is a time-consuming model.

SYSTEM DESIGN

3. SYSTEM DESIGN

System design defines a system's architecture, components, and data flow for scalability and reliability. It includes database design (SQL/NoSQL), caching (Redis), load balancing, message queues (Kafka), and security (OAuth, JWT, SSL). Monitoring tools (Prometheus, ELK) ensure performance, while redundancy and fault tolerance maintain system stability and efficiency.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During detailed design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module are usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented. In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. In other words, in system design the attention is on what components are needed, while in detailed design how the components can be implemented in software is the issue.

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

3.1 Database Design (E-R Diagram)

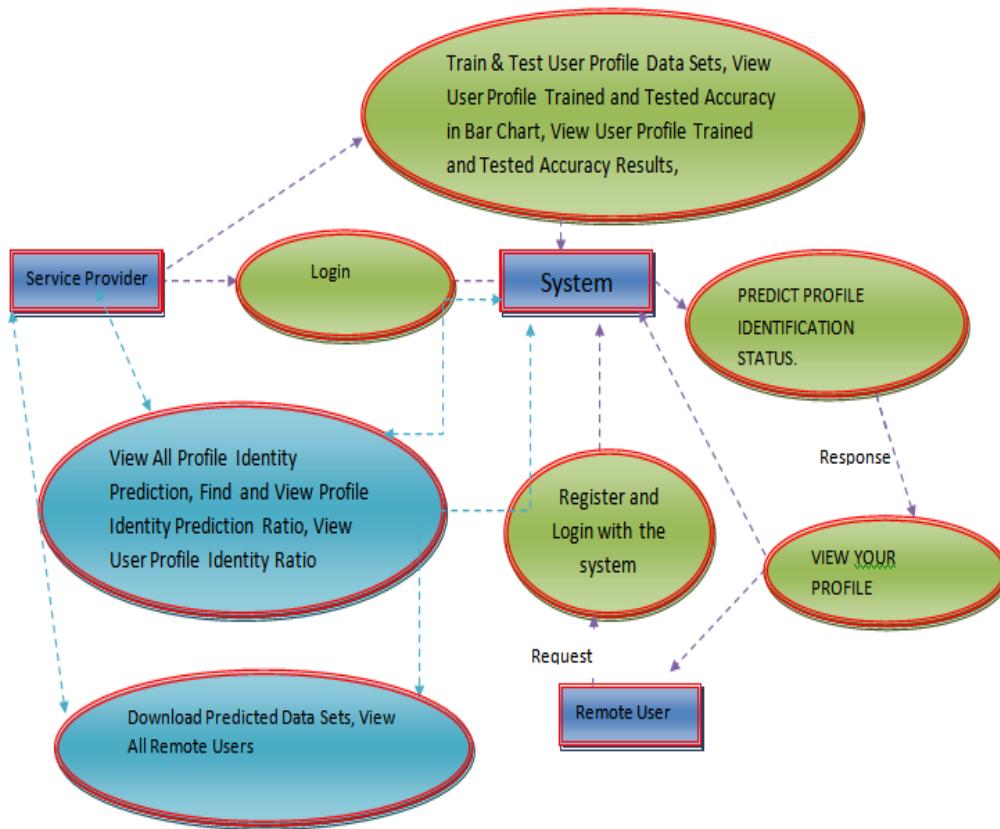
An Entity-Relationship (ER) model illustrates the structure of a database using a visual representation known as an Entity-Relationship Diagram (ER Diagram). This model serves as a blueprint for designing the database schema and capturing the relationships between different entities and attributes.

The ER model provides a systematic approach to organizing and conceptualizing the data within a database system. It represents entities as well as the relationships between them, helping to clarify how data elements are connected and organized.

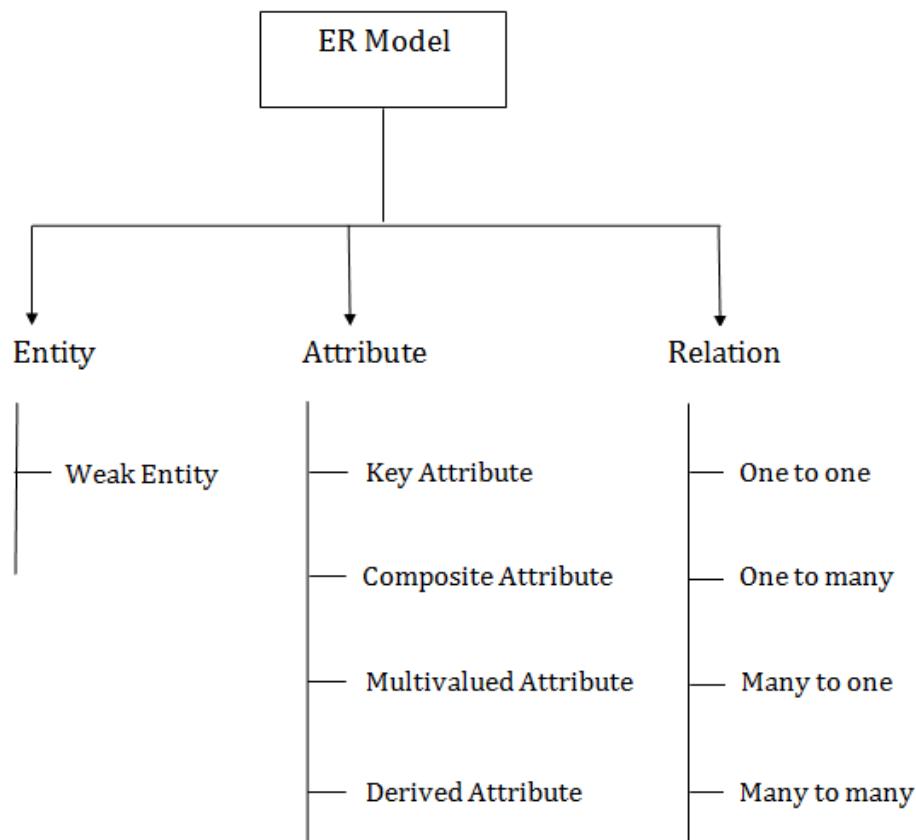
DATA FLOW DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

➤ Data Flow Diagram :

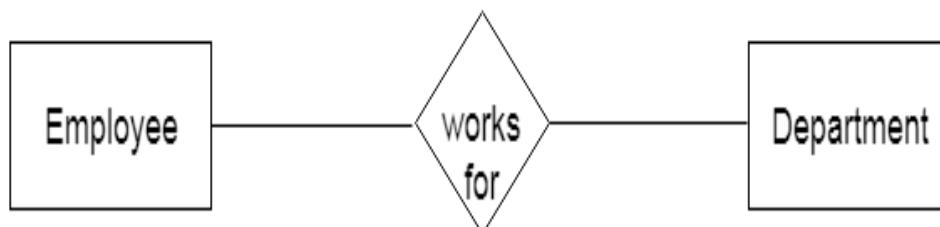
**ER model**

- 1.The Emergency Room model corresponds to an Entity-Relationship model, serving as a high-level representation of data structures. It is utilized to illustrate the data components and relationships within a defined system.
- 2.It establishes a structured framework for the database. Moreover, it provides a straightforward and easily understandable perspective on the data.
- 3.In Entity-Relationship modeling, the organizational database structure is depicted through a design known as an Entity-Relationship diagram.
- 4.For instance, consider designing a school database. An educational record could be represented as an entity with attributes such as name, ID, age, etc. Similarly, the address could be another entity with attributes like city, street name, zip code, etc., and there would be a relationship between them.

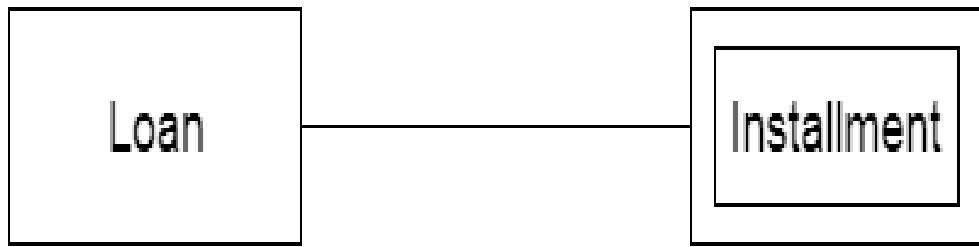
Component of ER Diagram**1. Entity:**

A substance may be anything, class, individual or spot. In the ER frame, a substance can be tended to as square shapes.

Think about a relationship as a delineation chief, thing, specialist, office, etc can be takeas a substance.

**1. Powerless Entity**

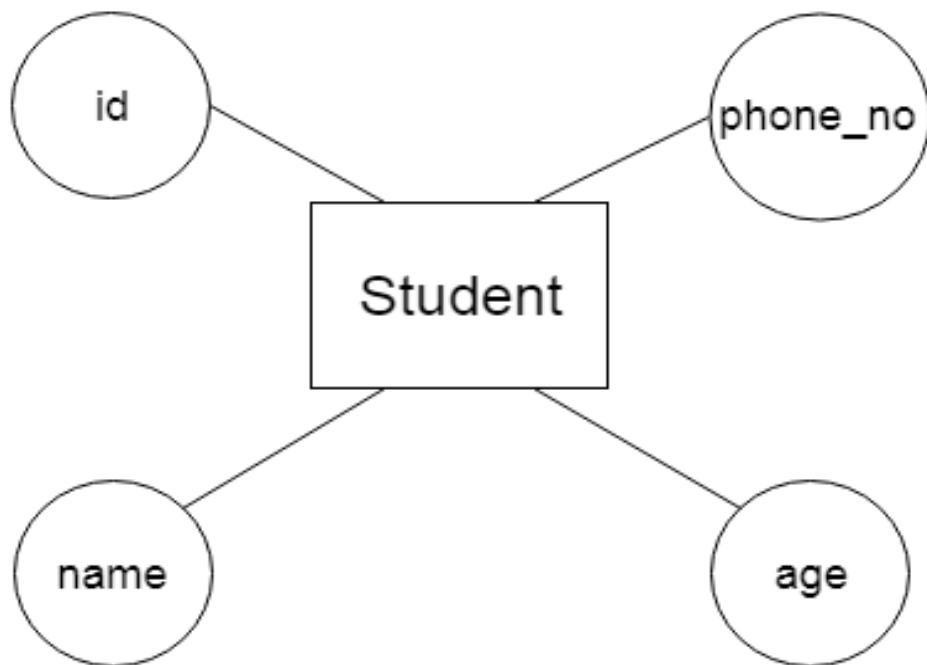
A substance that depends upon another component called afraiil substance. The frail element contains no critical trait of its own. The feeble substance is addressed by a twofold square shape.



2. Characteristic

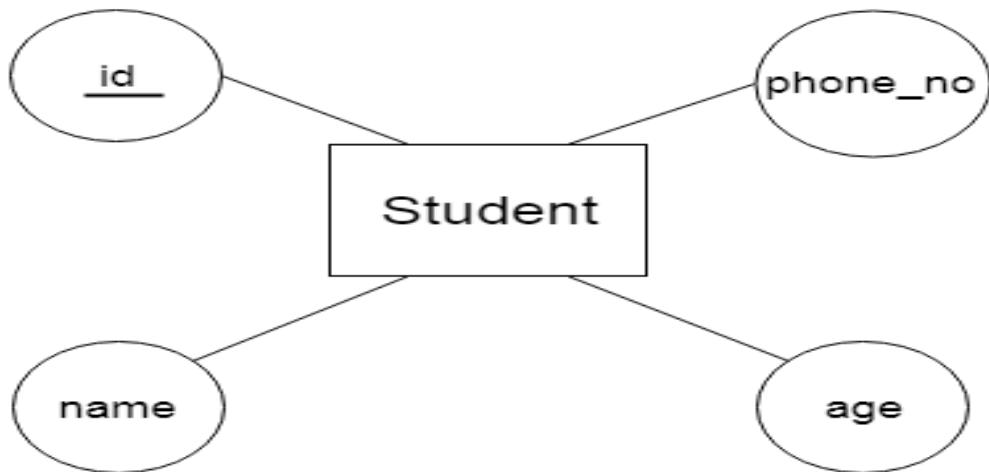
The quality is utilized to depict the property of a section. Obscure is utilized to address a quality.

For example, id, age, contact number, name, etc can be attributes of a student.

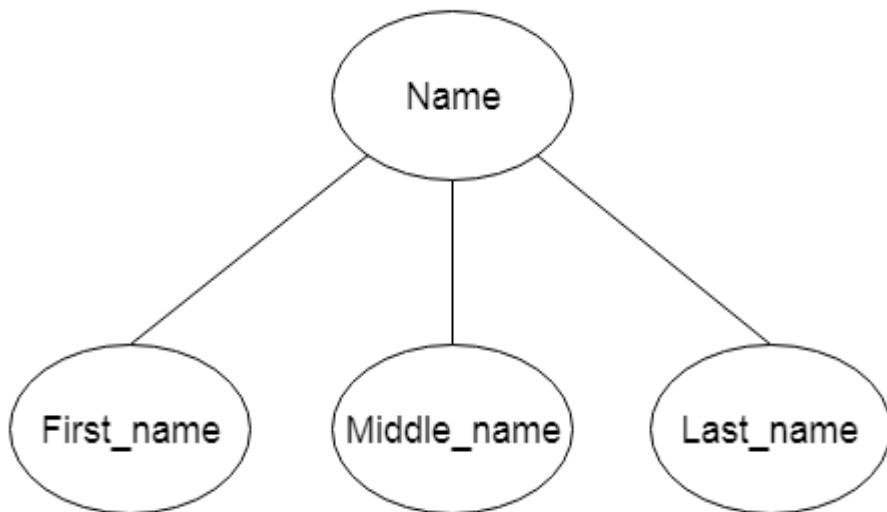


a. Key Attribute

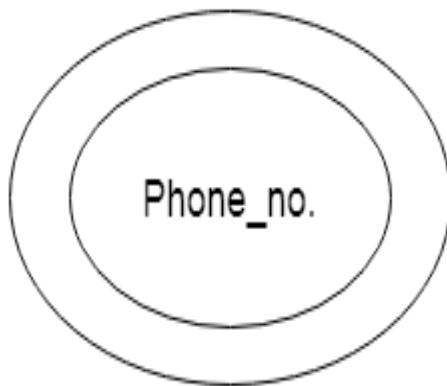
The key quality is used to address the essential ascribes of a substance. It tends to a fundamental key. The key property is tended to by a circle with the text underlined.

**b. Composite Attribute**

A property made from various attributes is known as a composite quality. The composite trademark is tended to by an oval, and those circles are related with a circle.

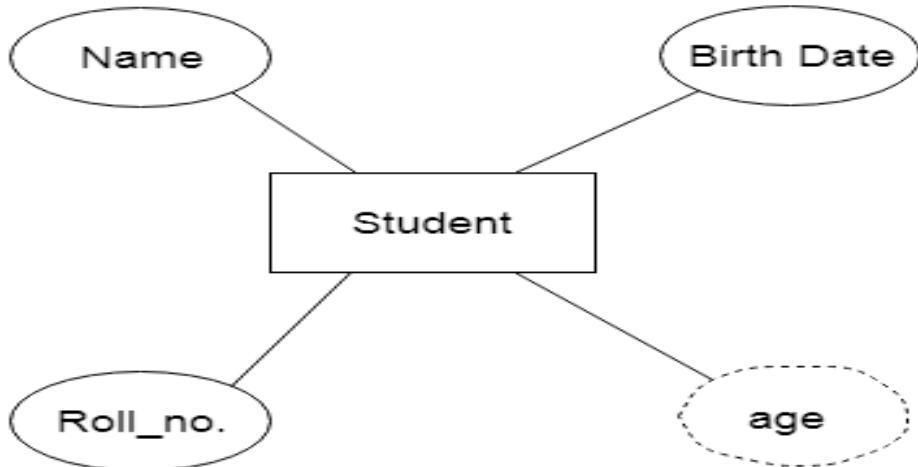
**c. Multivalued Attribute**

A quality can have more than one worth. These qualities are known as a multivalued property. The twofold oval is used to address multivalued property. For example, a student can have more than one phone number.

**d. Determined Attribute**

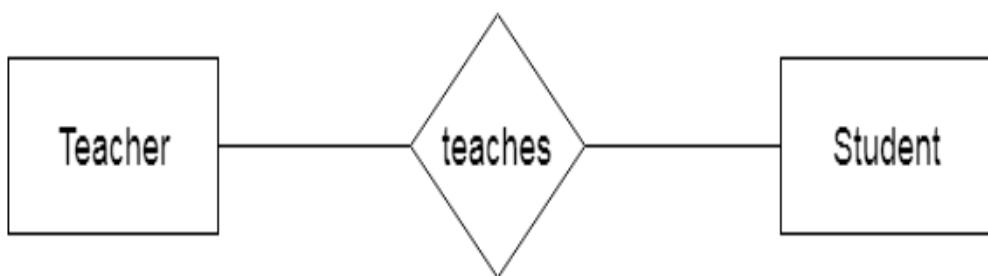
A property that can be gotten from another quality is known as a decided attribute. It will in general be tended to by a ran circle.

For example, a singular's age changes long term and can be gotten from one more quality like Date of birth.

**3. Relationship**

A relationship is used to depict the connection between substances.

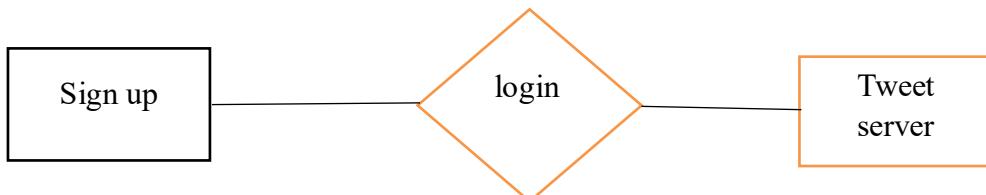
Important stone or rhombus is utilized to address the relationship.



Sorts of relationship are as per the following:

a. One-to-One relationship

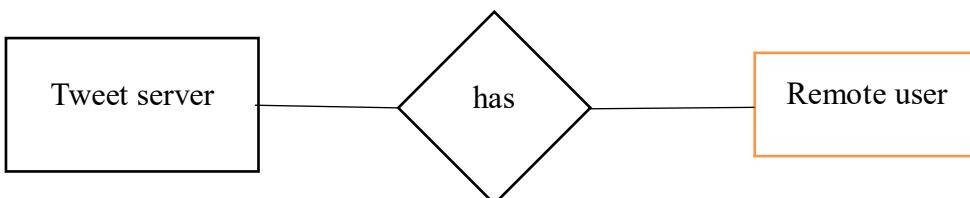
At the point when just a single instance of a component is connected with the relationship, then it is known as facilitated relationship .For instance, A female can wed to one male, and a male can wed to one female.



b. One-to-many relationship

Exactly when simply a solitary illustration of the substance on the left, and more than one event of a component on the right associates with the relationship then this is known as a one-to-various connections.

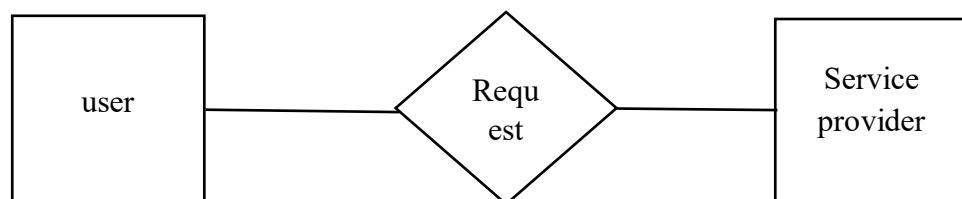
For example, Scientist can envision various manifestations, but the improvement is done by the really express analyst.



c. Many-to-one relationship

Exactly when more than one event of the component on the left, and simply a solitary event of a substance on the right associates with the relationship then it is known as a many-to-one relationship.

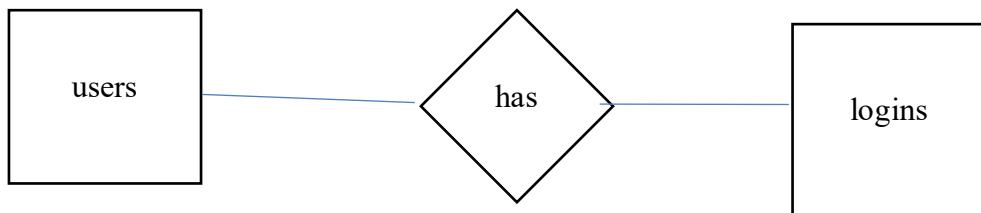
For example, Student enrols for only a solitary course, but a course can have various students.



d. Many-to-many relationship

At the point when more than one event of the substance on the left, and more than one event of a component on the right associates with the relationship then it is known as a many-to-various connections.

For example, Employee can allot by numerous exercises and project can have various specialists.



E-R Diagram

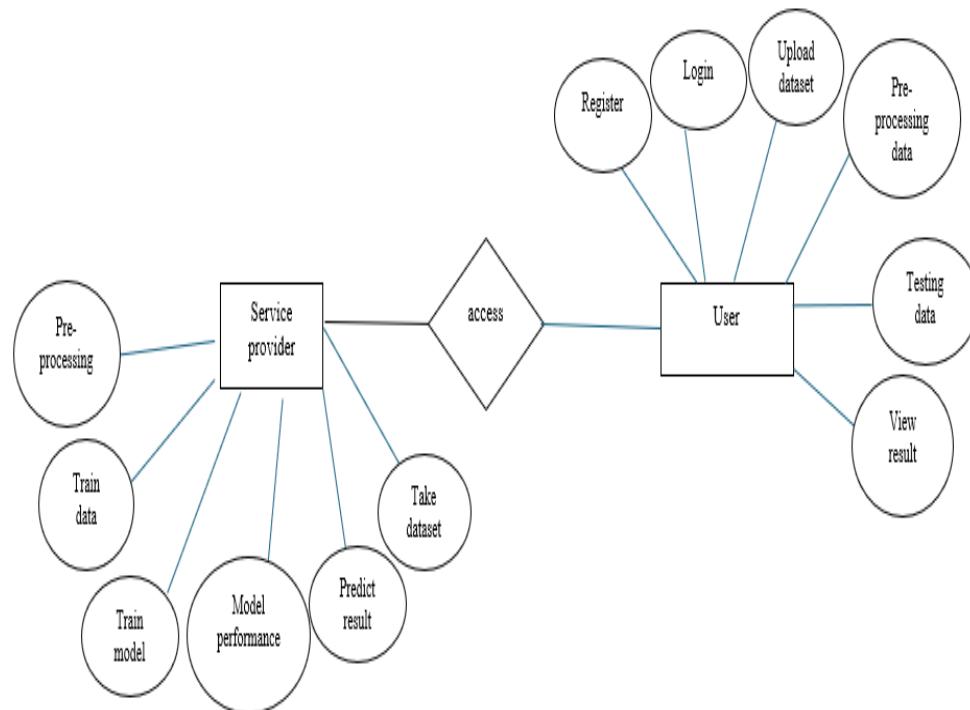


Fig:3.1.1.1 ER Diagram

3.2. Data Dictionary

A Data Dictionary compiles names, definitions, and attributes concerning data elements utilized or stored within a database, information system, or part of a research project. It delineates the meanings and functions of data elements within the context of a project and offers guidance on

understanding, recognizing meanings, and description. Additionally, a Data Dictionary offers metadata about data elements, aiding in defining the scope and attributes of data elements, as well as the guidelines for their usage and application.

Table Name: User

Description	Datatype	Constraint
Register	Varchar2 ()	Primary Key
Login	Number ()	NOT NULL
Upload Dataset	Varchar2 ()	NOT NULL
Pre-Processing Data	Varchar2 ()	NOT NULL
Testing The Data	Varchar2 ()	NOT NULL
Viewmodel Performance	Varchar2 ()	NOT NULL
View Result	Varchar2 ()	NOT NULL

Table no.3.2.1 User

Description: The above table shows all details about data owner.

Table Name: Service Provider

Description	Datatype	Constraint
Preprocessing The Data	Varchar2 ()	Primary Key
Train The Data	Number ()	NOT NULL
Train The Model	Varchar2 ()	NOT NULL
Model Performance	Varchar2 ()	NOT NULL
Predict The Result	Varchar2 ()	NOT NULL
Take Dataset	Varchar2 ()	NOT NULL

Table no 3.2.2 Service Provider

Description: The above table shows all details about cloud server

Table Name: Authenticator

3.3. Normalization

Normalization is the primary method for optimizing data in a database to fulfill two essential criteria:

Data dependencies are logical, ensuring that all related data items are stored together. Normalization is crucial for various reasons, primarily because it

enables databases to occupy minimal disk space, resulting in enhanced performance.

Normalization is also referred to as data standardization.

The three primary types of normalization are outlined below. Note: "NF" stands for "normal form."

First typical structure (1NF)

Tables in 1NF should comply with certain standards:

- 1.Every cell should contain just a solitary (nuclear) esteem.
- 2.Each part in the table ought to be astoundingly named.
- 3.All characteristics in a part ought to connect with a comparative region.

UserID	Username	Password
015	John	*****
016	Princess	*****
027	Tom	*****
028	Claire	*****
029	Robert	*****

Table No: 3.3.1 1NF

Second typical structure (2NF)

Tables in 2NF ought to be in 1NF and not have any most of the way dependence (e.g., each non-prime quality ought to be dependent upon the table's fundamental key).

System Design

Fake Profile Identification Using Artificial Neural Network

User Id	Received Data through IOT	pswd	Login
1	11	*****	Sign_up
2	12	*****	Sign_up
3	13	*****	Sign_up
4	14	*****	Sign_up
5	15	*****	Sign_up

Table no3.3.2 2NF

Third ordinary structure (3NF)

Tables in 3NF ought to be in 2NF and have no transitive reasonable circumstances on the fundamental key. The going with two NFs furthermore exists anyway are only here and there used:

USERDETAILS

ID	NAME	EMAIL	STATE	CITY	COUNTRY
11	Vijay	vijay@gmail.com	AP	RZP	INDIA
12	Vinod	vinod@gmail.com	AP	RZP	INDIA
13	Ramu	Ramu@gmail.com	AP	RZP	INDIA
14	Vishnu	vishnu@gmail.com	AP	RZP	INDIA

Table no 3.3.3 User Details

USER DETAILS

USER ID	PASSWORD	LOGIN
server	*****	Sign_up
Vijay	*****	Sign_up

Table no3.3.4 User details

Boyce-Codd Normal Form (BCNF)

Normalization is a critical process in database management aimed at organizing tables to minimize anomalies and ensure data integrity. It follows a series of stages known as normal forms. These normal forms help structure tables efficiently and reduce redundancy and inconsistency in data.

Unnormalized Form (UNF): The initial state of a table where data is not organized according to any specific rules.

First Normal Form (1NF): In 1NF, each column contains atomic values, and there are no repeating groups or arrays within a row.

Second Normal Form (2NF): 2NF requires that every non-key attribute be fully functionally dependent on the primary key.

Third Normal Form (3NF): In 3NF, no transitive dependencies should exist, meaning that non-key attributes should not depend on other non-key attributes.

Elementary Key Normal Form (EKNF): EKNF is a further refinement of 3NF, emphasizing the use of elementary keys.

Boyce-Codd Normal Form (BCNF): BCNF addresses anomalies that may arise when multiple candidate keys exist. It requires that for every non-trivial functional dependency ($X \rightarrow Y$), X must be a superkey.

Fourth Normal Form (4NF): To achieve 4NF, a table must be in BCNF and should not have multi-valued dependencies.

Essential Tuple Normal Form (ETNF): ETNF is a condition where each attribute in a tuple is essential to the understanding of the tuple itself.

Normal Form	Description
1NF	An alliance is in 1NF enduring it contains an atomic worth.

System Design

Fake Profile Identification Using Artificial Neural Network

<u>2NF</u>	An association will be in 2NF expecting it is in 1NF and all non-key credits are totally down to earth ward on the fundamental key.
<u>3NF</u>	An alliance will be in 3NF enduring it is in 2NF and no change dependence exists.
BCNF	A more grounded importance of 3NF is known as Boyce Codd's common design.
<u>4NF</u>	An association will be in 4NF expecting it is in Boyce Codd's commonplace construction and has no multi-regarded dependence.
<u>5NF</u>	An association is in 5NF. In case it is in 4NF and contains no join dependence, joining should be lossless.

Benefits of Normalization:

Reduction of data redundancy: Normalization helps eliminate redundant data by organizing it efficiently across tables. Improved overall database organization: By structuring data according to normalization rules, databases become more organized and easier to manage. Data consistency within the database:

Normalization ensures that data remains consistent across tables, reducing the risk of inconsistencies. More flexible database design:

Normalization allows for more flexibility in database design, making it easier to accommodate changes and updates. Upholds the principle of data integrity: Normalization promotes data integrity by minimizing anomalies and ensuring accurate representation of data relationships.

Disadvantages of Normalization:

Careless decomposition: If normalization is done without a clear understanding of user requirements, it can lead to excessive decomposition and unnecessary complexity in the database design.

Decreased performance: As tables are normalized to higher normal forms such as 4NF or 5NF, it may lead to decreased performance due to increased join operations and complexity in querying the database.

3.2. UML DIAGRAMS INTRODUCTION

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntax, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

3.2.1. Design of Code

Since information systems projects are designed with space, time and cost saving in mind, coding methods in which conditions, words, ideas or control errors and speed the entire process. The purpose of the code is to facilitate the identification and retrieval of the information. A code is an ordered collection of symbols designed to provide unique identification of an entity or an attribute.

3.2.2. Design of Input

Design of input involves the following decisions

- Input data

- Input medium
- The way data should be arranged or coded
- Validation needed to detect every step to follow when error occurs

The input controls provide ways to ensure that only authorized users access the system guarantee the valid transactions, validate the data for accuracy and determine whether any necessary data has been omitted. The primary input medium chosen is display. Screens have been developed for input of data using HTML. The validations for all important inputs are taken care of through various events using JSP control.

3.2.3. Design of Output

Design of output involves the following decisions

- Information to present
- Output medium
- Output layout

Output of this system is given in easily understandable, user-friendly manner, Layout of the output is decided through the discussions with the different users.

3.2.4 Design of Control

The system should offer the means of detecting and handling errors.

Input controls provides ways per

- Valid transactions are only acceptable
- Validates the accuracy of data
- Ensures that all mandatory data have been captured

All entities to the system will be validated. And updating of tables is allowed for only valid entries. Means have been provided to correct, if any by change incorrect entries have been entered into the system they can be edited.

3.3. UML DIAGRAMS

Why We Use UML in projects?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modeling Language (UML) was designed to respond to these needs. Simply, Systems design refers to the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

In the project four basic UML diagrams have been explained among the following list:

- Class Diagram
- Use Case Diagram
- Sequence Diagram
- Activity Diagram
- Collaboration Diagram
- Deployment Diagram
- State Chart Diagram
- Component Diagram

Class Diagram

A Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

This is one of the most important of the diagrams in development. The diagram breaks the class into three layers. One has the name, the second describes its attributes and the third its methods. A padlock to left of the name represents the private attributes. The relationships are drawn between the classes. Developers use the Class Diagram to develop the classes. Analyses use it to show the details of the system.

Architects look at class diagrams to see if any class has too many functions and see if they are required to be split.

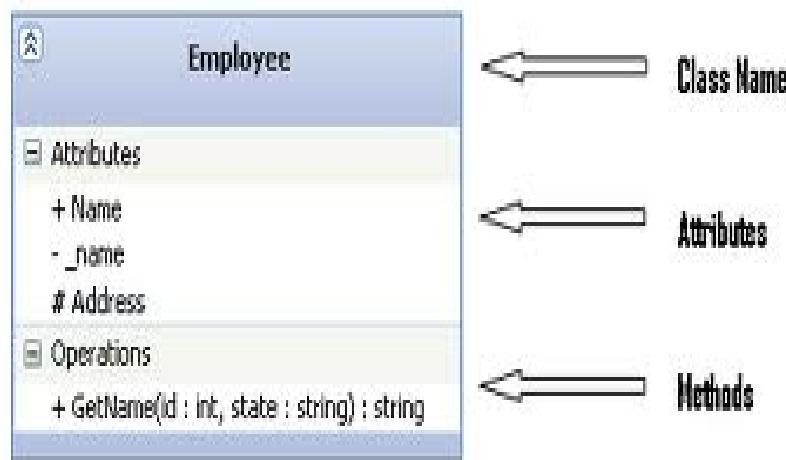


Fig.3.3.1: Class Diagram

Use Case Diagram

A Use Case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor.

System Design

Fake Profile Identification Using Artificial Neural Network

Roles of the actors in the system can be depicted. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from the external point of view.

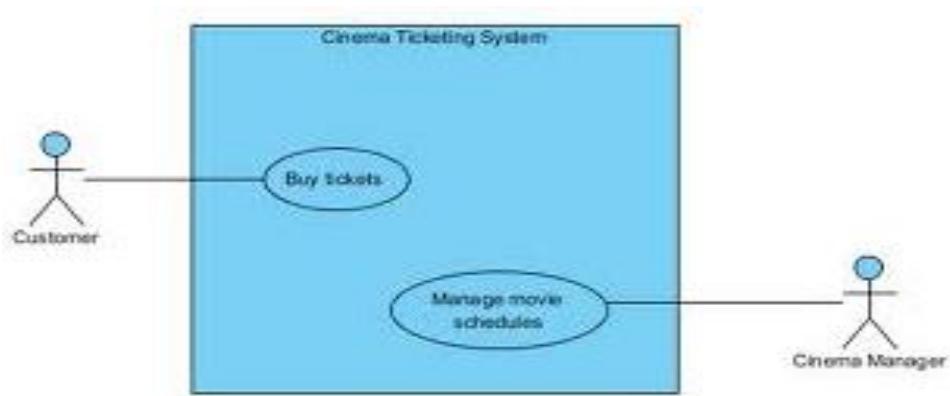


Fig.3.3.2: Use Case Diagram

Sequence Diagram

A Sequence diagram in Unified Modeling Language (UML) 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. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams

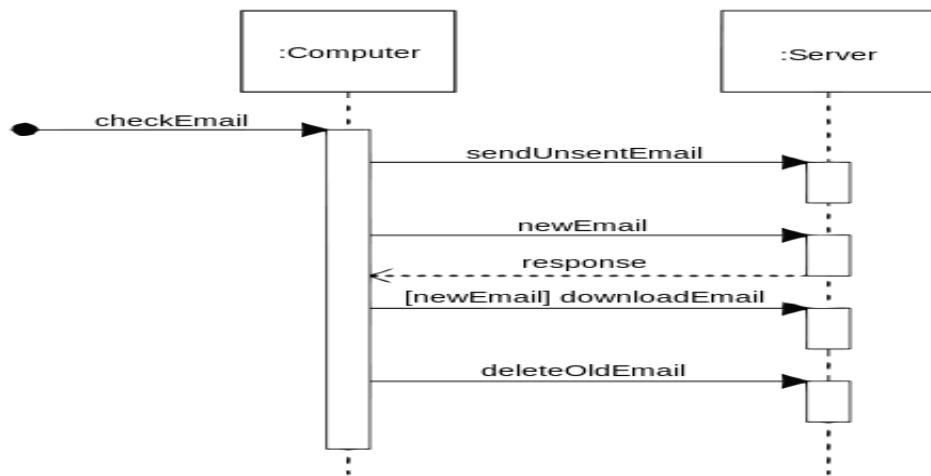


Fig.3.3.3: Sequence Diagram

Activity Diagram

Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

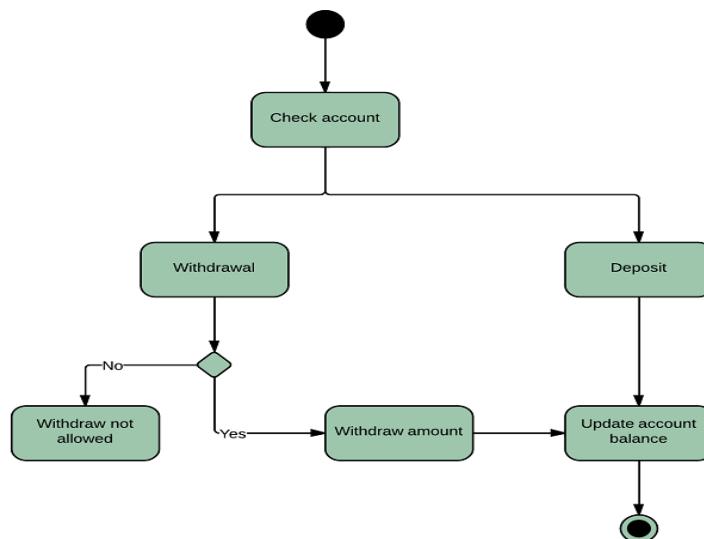


Fig.3.3.4: Activity Diagram

Collaboration Diagram

A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication 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.

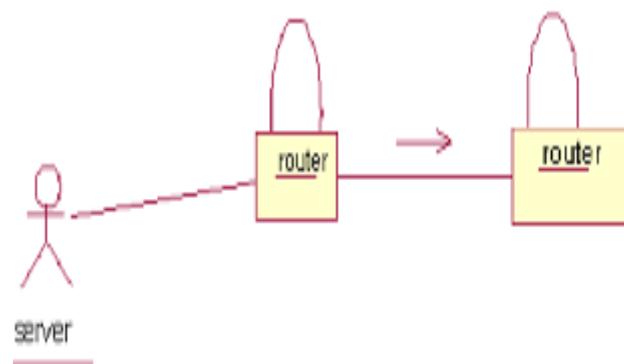


Fig.3.3.5: Collaboration Diagram

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

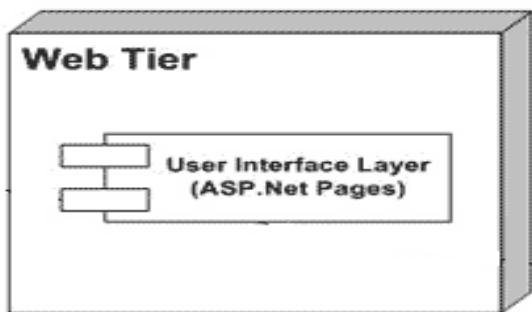


Fig.3.3.6: Deployment Diagram

State Chart Diagram

A State diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

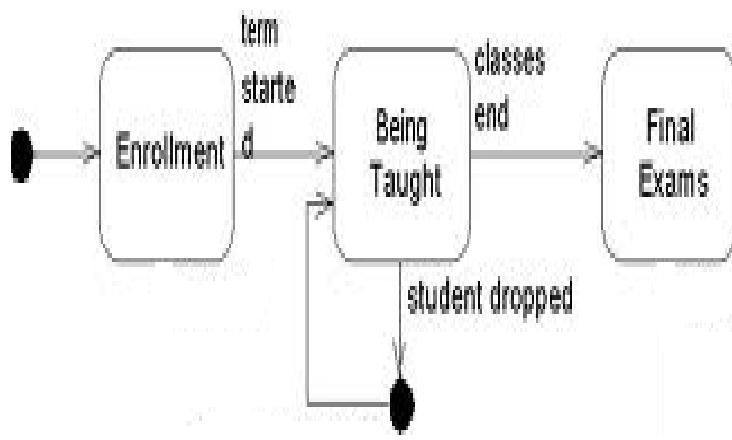


Fig.3.3.7: State Chart Diagram

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.

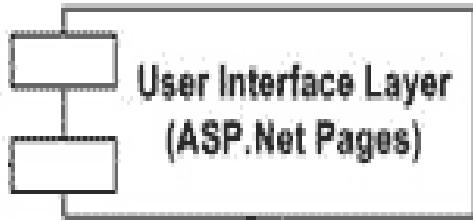


Fig.3.3.8: Component Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

3.3.1. USE CASE DIAGRAM

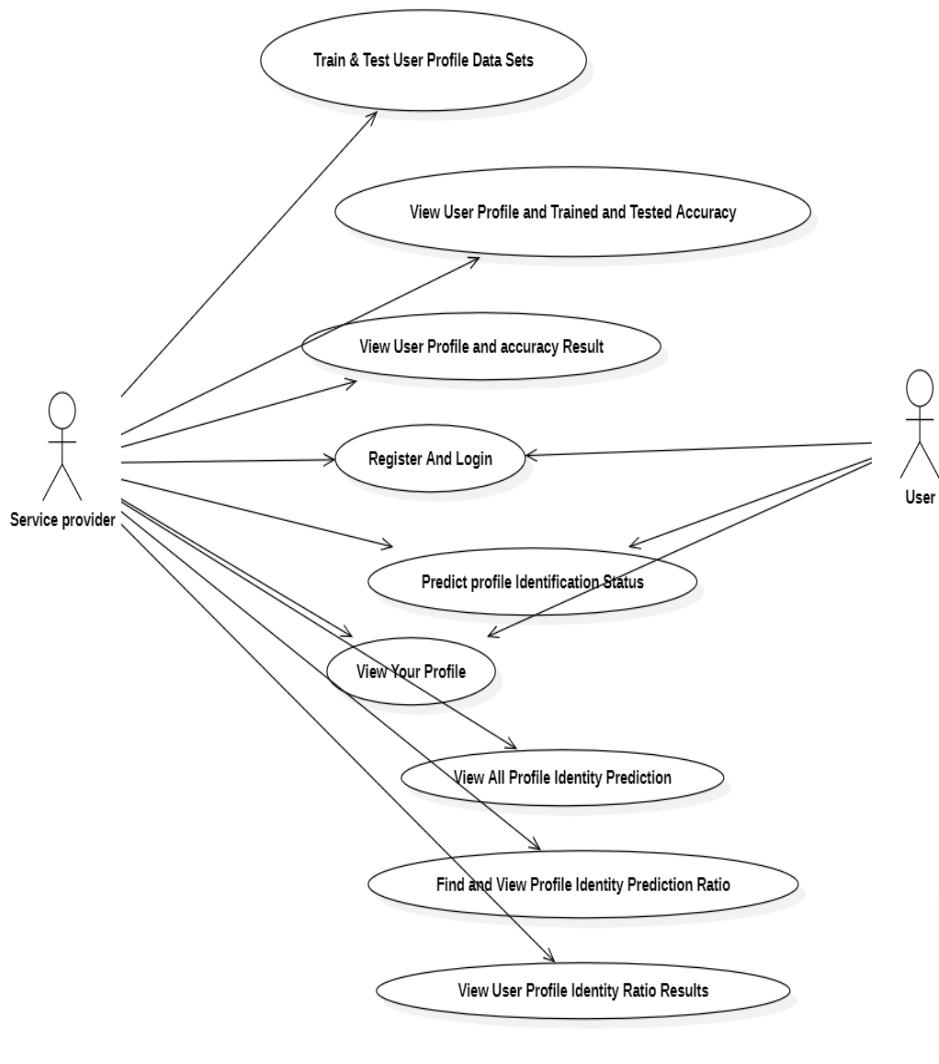


Fig.3.3.1.1: Use case Diagram for overall project

Description: In this use case diagram sender and receiver is an actor. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor.

3.3.2. CLASS DIAGRAM:



Fig.3.3.2.1: Class Diagram for Overall Project

Description: In this class diagram using two object source and destination. In the user object class the file name, id address source id as string. In the class diagram some operations and attributes. In the system class system id as integer, name, system-type as string. Some operations are involved in the class.

3.3.3. SEQUENCE DIAGRAM:

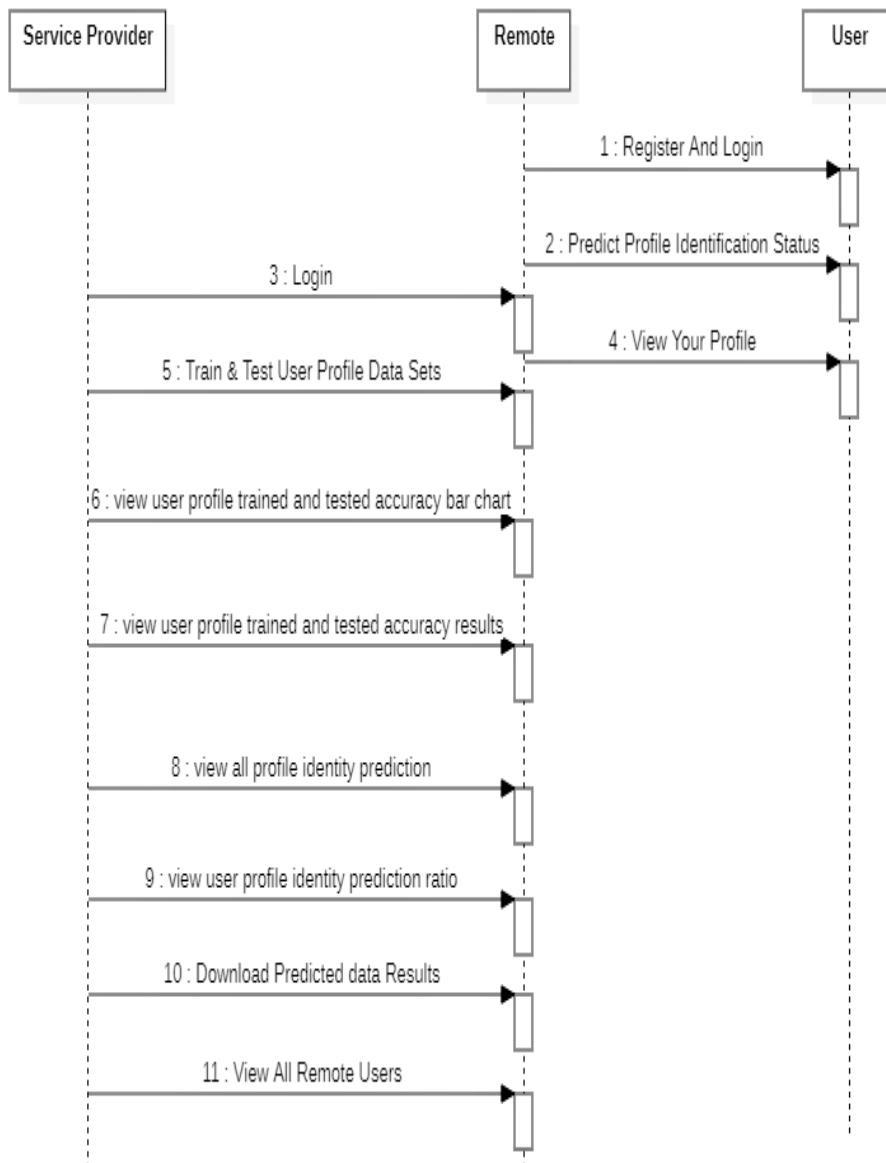


Fig.3.3.3.1: Sequence Diagram for Overall Project

Description: A Sequence diagram in Unified Modelling Language (UML) 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 as Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

3.3.4. ACTIVITY DIAGRAM:

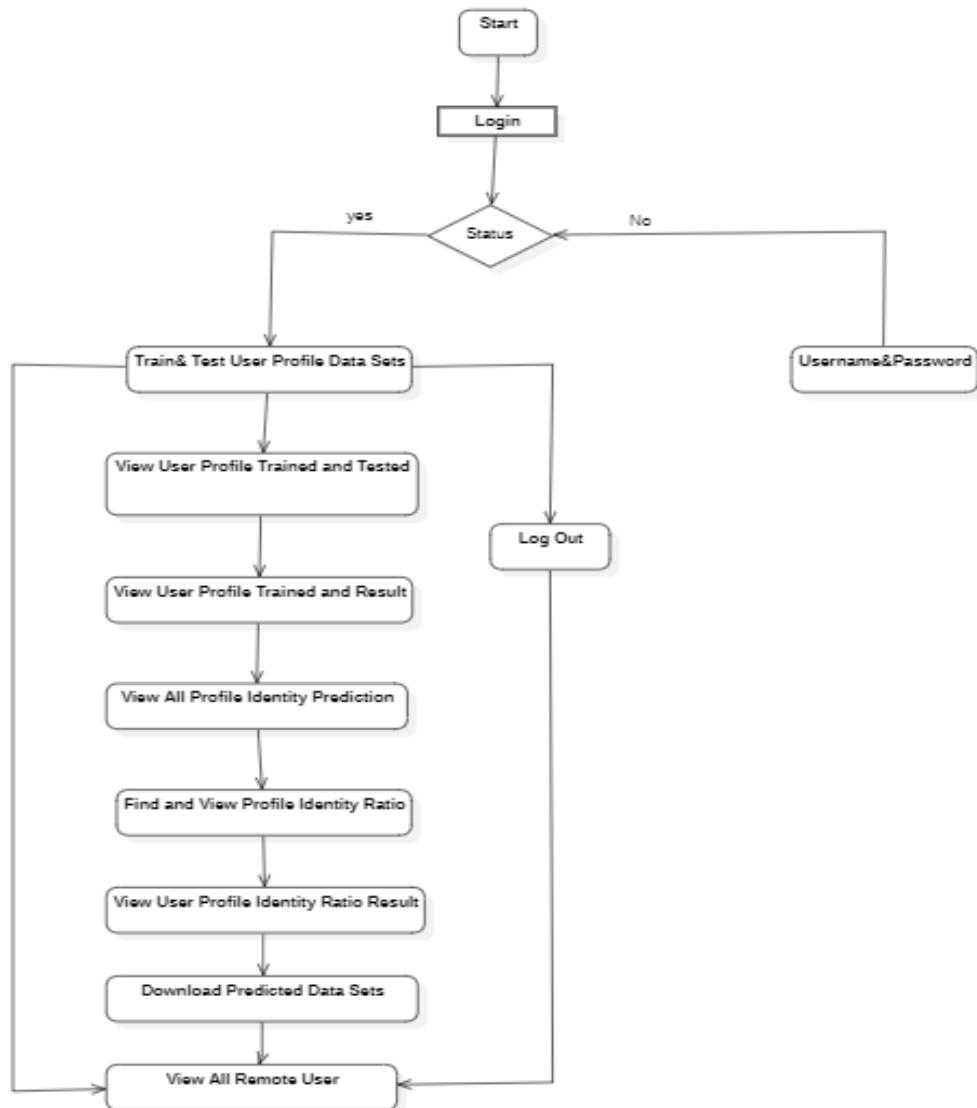


Fig.3.3.4.1: Activity Diagram

Description: Activity diagrams are graphical representations of workflows of a step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

3.3.5 Deployment Diagram



Fig.3.3.5 Deployment Diagram for overall project

Description: This deployment diagram provides a high-level overview of the architecture for network traffic analysis using machine learning, but actual implementations may vary depending on specific requirements, infrastructure, and available technologies. It typically involves multiple components distributed across various layers of the network infrastructure.

TESTING

4. TESTING

4.1 TESTING METHODOLOGIES

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

4.1.1. Testing Objectives

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a probability of finding an as yet un discovered error.
3. Un discovered error.
4. A successful test is one that uncovers an undiscovered error.
5. These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present

4.1.2. Test Case Design

Any engineering product can be tested in one of two ways:

White Box Testing

This testing is also called as glass box testing. In this testing, by knowing the specified function that a product has been designed to perform test can be conducted that demonstrates each function is fully operation at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases.

Basis path testing is a white box testing.

Basis Path Testing

- Flow graph notation
- Cyclomatic Complexity

Deriving test cases Control Structure Testing

- Condition testing
- Data flow testing
- Loop testing

Black Box Testing

In this testing by knowing the internal operation of a product, tests can be conducted to ensure that “all gears mesh”, that is the internal operation

performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing
- Graph matrices

4.1.3 SOFTWARE TESTING STRATEGIES

A Strategy for software testing integrates software test cases into a series of well planned steps that result in the successful construction of software. Software testing is a broader topic for what is referred to as Verification and Validation. Verification refers to the set of activities that ensure that the software correctly implements a specific function. Validation refers to the set of activities that ensure that the software that has been built is traceable to customer's requirements.

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

Integration Testing

Integration testing is a systematic technique for constructing the program structure, while conducting test to uncover errors associated with the interface. The objective is to take unit tested methods and build a program structure that has been dictated by design.

Top-Down Integration

Top down integrations is an incremental approach for construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control program. Modules

subordinate to the main program are incorporated in the structure either in the breath-first or depth-first manner.

Bottom-up Integration

This method as the name suggests, begins construction and testing with atomic modules i.e., modules at the lowest level. Because the modules are integrated in the bottom up manner the processing required for the modules subordinate to a given level is always available and the need for stubs is eliminated.

Regression Testing

In this context of an integration test strategy, regression testing is the re execution of some subset of test that have already been conducted to ensure that changes have not propagate unintended side effects.

Validation Testing

At the end of integration testing software is completely assembled as a package. Validation testing is the next stage, which can be defined as successful when the software functions in the manner reasonably expected by the customer. Reasonable expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

Reasonable expectation is defined in the software requirement specification – a document that describes all user-visible attributes of the software. The specification contains a section titled “Validation Criteria”. Information contained in that section forms the basis for a validation testing approach.

Validation Test Criteria

Software validation is achieved through a series of black-box tests that demonstrate conformity with requirement. A test plan outlines the classes of tests to be conducted, and a test procedure defines specific test cases that will be used in an attempt to uncover errors in conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all performance requirements are achieved, documentation is correct and human-engineered; and other requirements are met.

After each validation test case has been conducted, one of two possible conditions exists: (1) The function or performance characteristics conform

Testing

Fake Profile Identification Using Artificial Neural Network

to specification and are accepted, or (2) a deviation from specification is uncovered and a deficiency list is created. Deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is often necessary to negotiate with the customer to establish a method for resolving deficiencies.

Alpha and Beta Testing

It is virtually impossible for a software developer to foresee how the customer will really use a program. Instructions for use may be misinterpreted. Strange combination of data may be regularly used; and output that seemed clear to the tester may be unintelligible to a user in the field.

When custom software is built for one customer, a series of acceptance tests are conducted to enable the customer to validate all requirements. Conducted by the end user rather than the system developer, an acceptance test can range from an informal “test drive” to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of weeks or months, thereby uncovering cumulative errors that might degrade the system over time.

The beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a “live” application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta test, the software developer makes modification and then prepares for release of the software product to the entire customer base.

System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated to perform allocated functions.

Security Testing

Attempts to verify the protection mechanisms built into the system.

4.2. TEST CASES

S. No	Test Cases	Input	Expected Result	Actual Result	Status
1	User registration	Enter all fields	User gets registered	Registration is successful	pass
2	User registration	If user miss any fields	User not registered	Registration is unsuccessful	Fail
3	Admin login	Give the user name and password	Admin home page should be opened	Admin home page has been opened	Pass
4	Upload CAMPUS dataset	Test whether the CAMPUS dataset is uploaded or not into the system	If CAMPUS dataset is uploaded	We can do further operations	Pass
5	Preprocess dataset	Verify the dataset is pre processed or not	loading the dataset	We can preprocess data	Pass
6	Runcnn algorithm	Verify the algorithm will run or not	Without training model	We cannot run algorithm	Fail

Fig.no 4.2.1 Test cases

IMPLEMENTATION

5. IMPLEMENTATION

5.1 Working Model Installation Procedure

Python

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc. The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python:-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python; they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps

Implementation

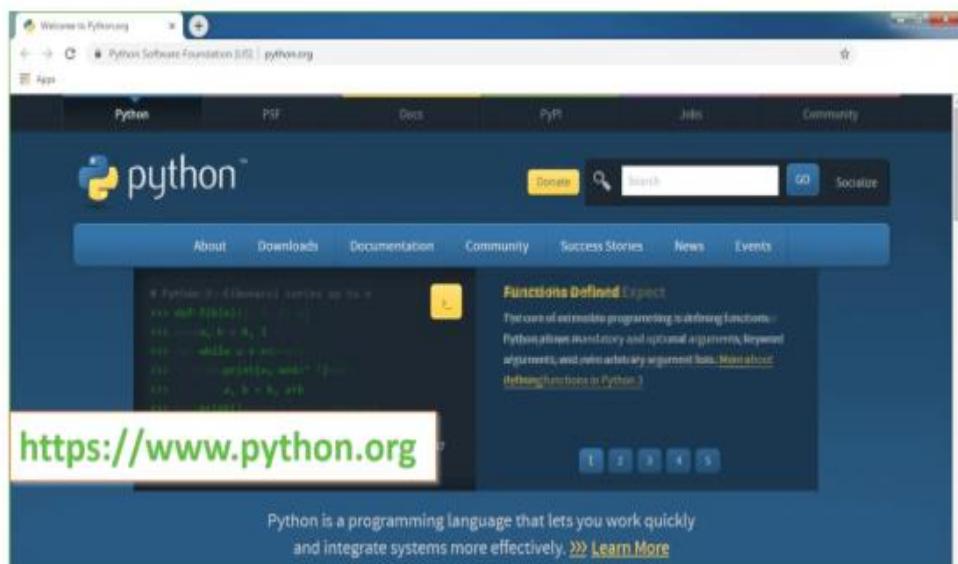
Fake Profile Identification Using Artificial Neural Network

below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here](#).The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

1. Installation of Python :

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser.

Click on the following link: <https://www.python.org>



Screen 5.1.1: Install Python

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Screen 5.1.2: Download Python

Implementation

Fake Profile Identification Using Artificial Neural Network

Step 3 : You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?			
Python releases by version number:			
Release version	Release date	Click for more	
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.18	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec 24, 2018	Download	Release Notes

Screen 5.1.3: Downloading Python

Step 4: Scroll down the page until you find the Files option.

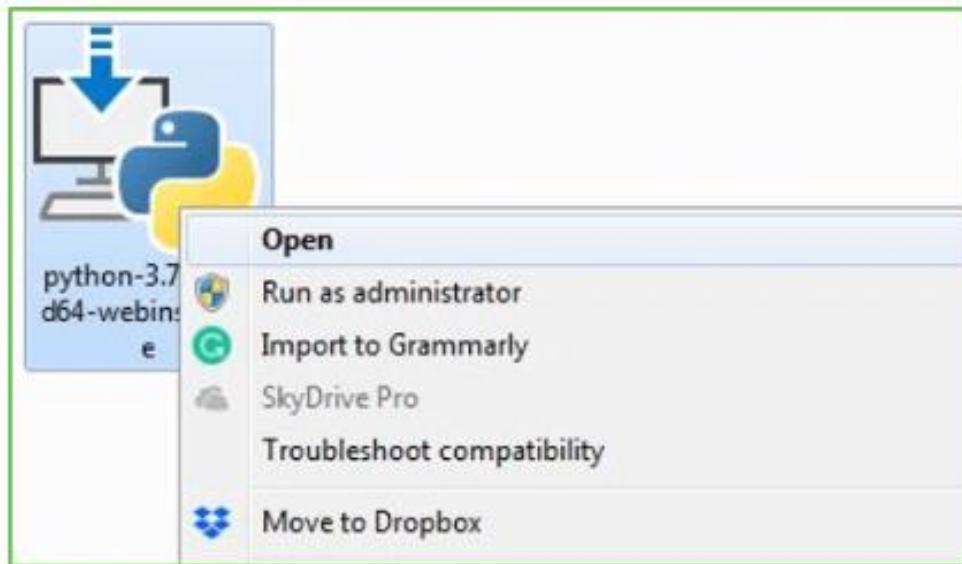
Step 5: Here you see a different version of python along with the operating

Files					
Version	Operating System	Description	MD5 Sum	File Size	GP6
Gzipped source tarball	Source release		6f111671e5b20b4aef7b9ab01bf0f9be	1301763	SG
XZ compressed source tarball	Source release		d33e4aae6097051c2eca5ee3004803	1713142	SG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428bd4fa7583da#f1a443cbacce0fe6	34896456	SG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd695c38217a45773b75ea936b341f	28082845	SG
Windows helpt file	Windows		d6399573a2c9b2ac56cad6b47cd2	8111761	SG
Windows x86-64 embeddable zip file	Windows	for AMD64 (18647)64	9b00c4cf8fbec2fb6a8318aa4072ba2	7504391	SG
Windows x86-64 executable installer	Windows	for AMD64 (18647)64	a102b4bca7fdehd03041a183e5a3408	26480318	SG
Windows x86-64 web-based installer	Windows	for AMD64 (18647)64	28cb1cf08fbef73ae553a3b2518-4bd2	1362904	SG
Windows x86 embeddable zip file	Windows		95a63b01fba41879fd04133574139d3	6741628	SG
Windows x86 executable installer	Windows		33cc02942a5446a20451470294789	25663848	SG
Windows x86 web-based installer	Windows		1b670cf5d317df02c3093ea37d87c	1324608	SG

Screen 5.1.4: Version Of Software

Installation of Python:

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

**Screen 5.1.5: Installing Python**

Step 2 : Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

**Screen 5.1.6: Installing Process**

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Screen 5.1.7: Successful Install

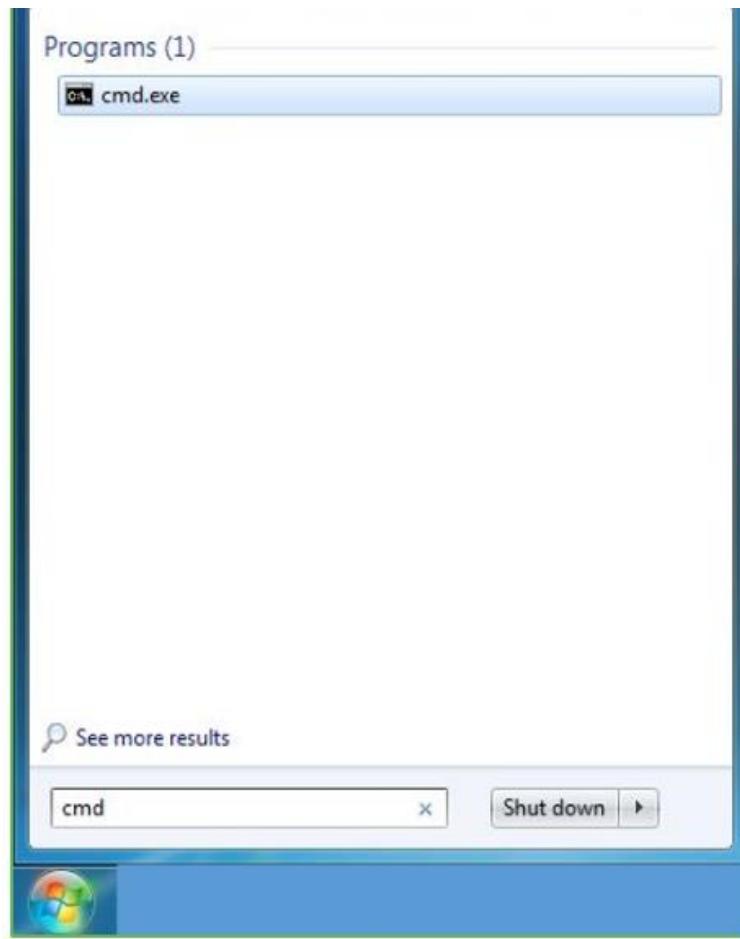
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation :

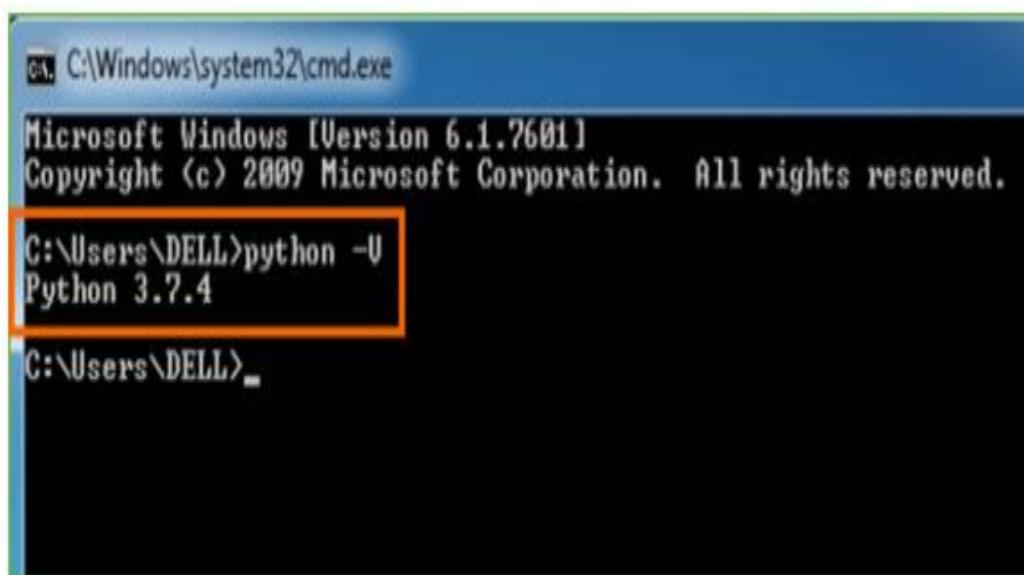
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

**Screen 5.1.8: Command Prompt**

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python – V and press Enter.



A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window displays the following text:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\DELL>python -V
Python 3.7.4
C:\Users\DELL>

Screen 5.1.9: Check Python Install

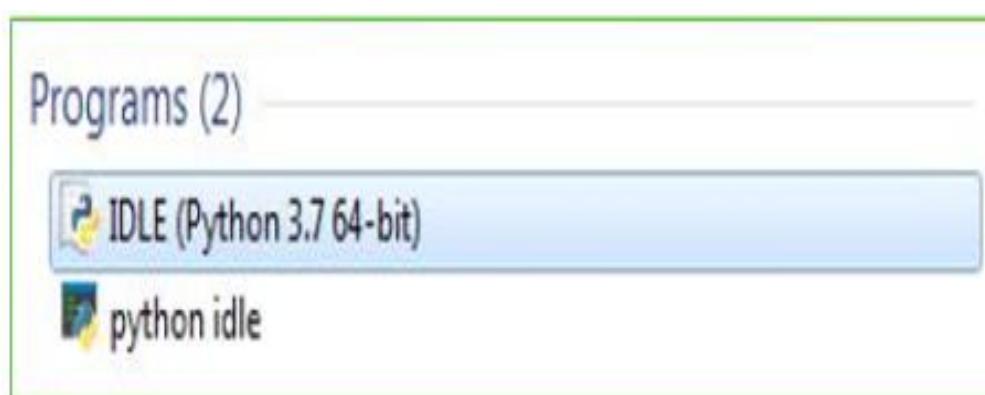
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

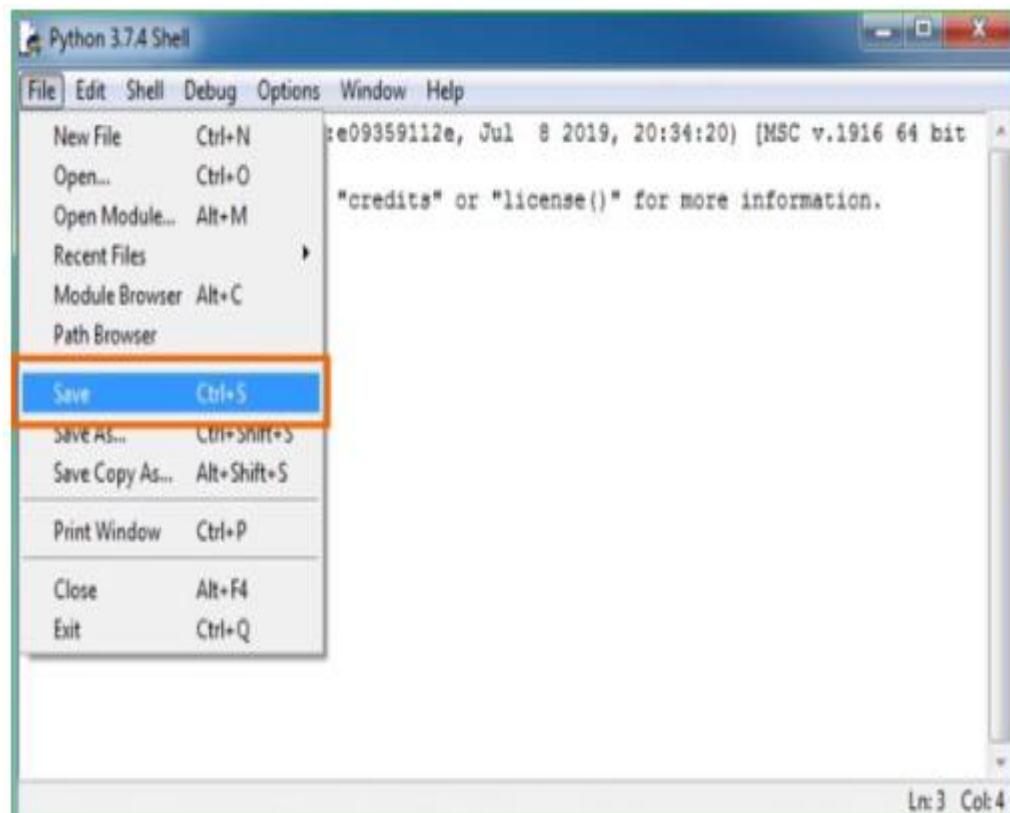
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

**Screen 5.1.10: Run Command**

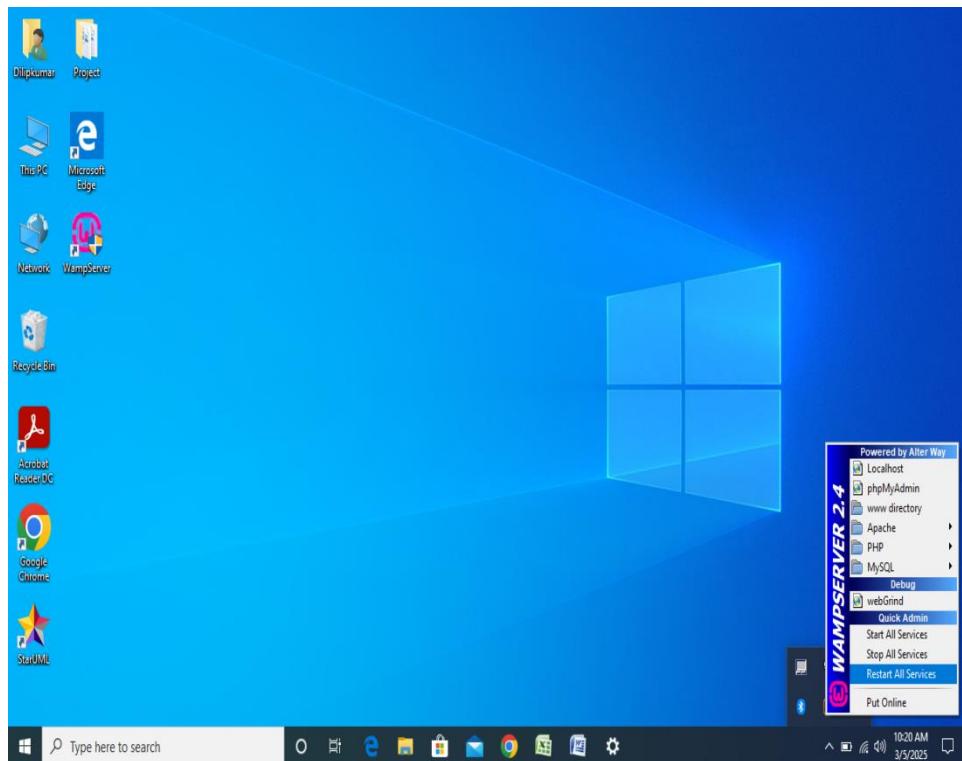
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save

**Screen 5.1.11: Working In Idle**

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print

5.2. Sample Screen**Screeen 5.2.1: Start wamp server**

Description: - To start our project activate the WampServer and select start all services as shown on the above screen.

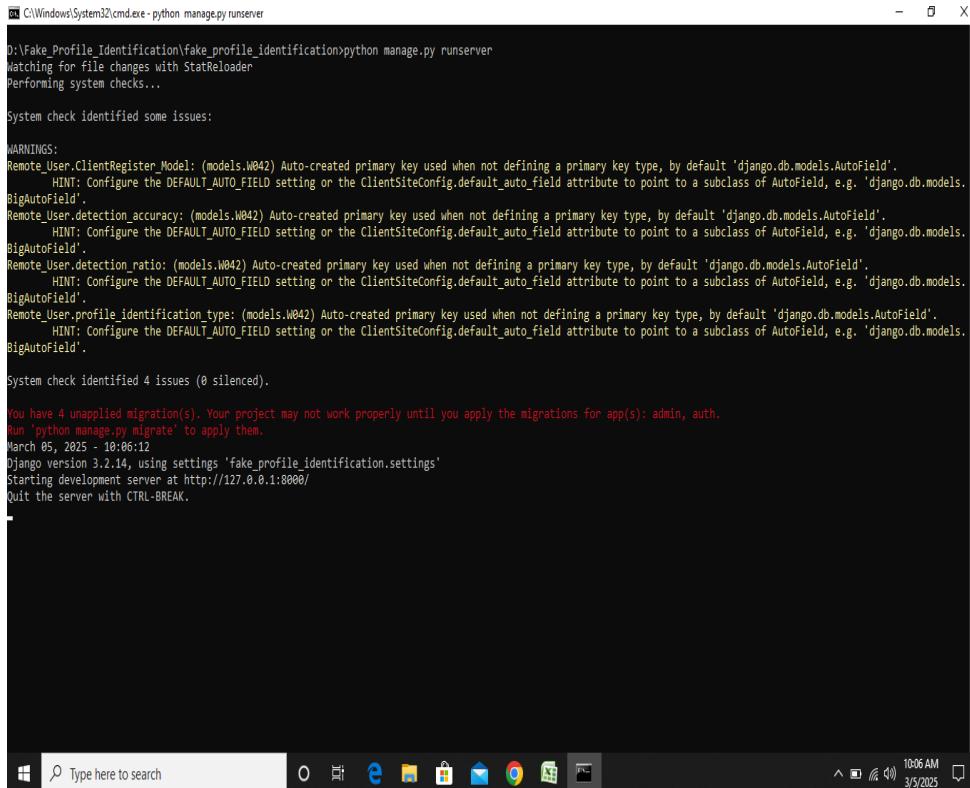
Implementation

Fake Profile Identification Using Artificial Neural Network

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	id	name	screen_name	fav_numbers	statuses_count	followers_count	friends_count	favourites_count	listed_in_location	created_at	default_profile_image	profile_beacon	profile_use_website	profile_text	profile_in_page	profile_sidebar	profile_status	profile_type	profile_in_picture	profile_silhouette	profile_beacon
2	3.7E+08	perfectmo	perfectmo	1	24	4	588	16	0	Thu Sep 08 13:20:35 +0000 2011	http://a0.NULL	https://tw	1	https://tw	333333	https://tw	FFFF	DDEEF			
3	37384589	SAK Nair	bsknair19	898	656	57	693	597	0	Sun May 0 Kuwait	http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
4	72110028	Deepak dediyen		4507	1234	15	104	1150	0	Sun Sep 0 India	http://a0.NULL		1	https://sil	333333	https://sil	EEEEEE	1	EFFEF		
5	82885728	Marcos Vi BravAlve		0	573	14	227	530	0	Fri Oct 16 Rio de Janeiro	http://a0.NULL		1	https://sil	1F1D1	https://sil	CODEED	DDEEF			
6	1.1E+08	Shri Kanti kanauijasi		2321	675	18	519	653	0	Sun Jan 31 lucknow	http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
7	1.34E+08	Shree visl shreeswai		5033	1333	73	1998	1262	1	Sun Aug 18 12:04:04 +0000 2010	http://a0.NULL		1	https://tw	333333	https://tw	EEEEEE	1	EFFEF		
8	1.96E+08	crystiane crystiane		14	99	26	1548	80	0	Mon Sep 27 21:53:12	http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
9	2.53E+08	shashank creativebvi		203	553	63	1930	497	0	Tue Feb 1 Pune	http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
10	2.9E+08	santosh n santosha		13	1576	8	501	1402	1	Sat Apr 30 Ranai	http://a0.NULL		1	https://sil	OC353	https://sil	F2E95	FF7CC			
11	3.04E+08	DATTARAIDATTARAID		136443	1378	48	1998	1108	0	Mon May 23 17:15:15	1 http://a0.NULL	https://sil	1	https://sil	333333	https://sil	CODEED	DDEEF			
12	3.49E+08	suraj jadh surajjadh		69395	1444	35	390	1283	0	Sat Aug 06 amravati ,maharashtra	http://a0.NULL		1	https://sil	333333	https://sil	CODEED	1	DDEEF		
13	4.76E+08	Nirmal smartnirml		1316	1351	7	328	1273	1	Fri Jan 27 11:24:28 +0	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
14	6.16E+08	Rochell Crochellcar		1320	43	17	641	0	0	Sat Jun 23 DIADEMA	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
15	6.16E+08	Thomaser thomasen		24	50	20	630	0	0	Sat Jun 23 In your ho	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
16	6.16E+08	Armetta W whitfield		1170	68	22	602	0	0	Sat Jun 23 Arizona	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
17	6.16E+08	Tonia Jacc tontonjac		30	60	14	592	0	0	Sat Jun 23 DÑ±D,D/0/1	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
18	6.16E+08	Kassandra kasandrac		23	52	27	620	0	0	Sat Jun 23 Rio Grand	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
19	6.16E+08	Stefania S searsfo		1044	67	32	639	0	0	Sat Jun 23 queens n	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
20	6.16E+08	Shamika shamikag		0	44	21	614	0	0	Sat Jun 23 pontianak	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
21	6.16E+08	Esperanza maxwelllr		79060	66	23	650	0	0	Sat Jun 23 malaysia	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
22	6.16E+08	Janina Casjaninahzrc		3939	56	20	643	0	0	Sat Jun 23 WORLDWI	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
23	6.16E+08	Myrtle Bo bowmanfr		337	65	23	641	0	0	Sat Jun 23 Fairburn, i	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
24	6.16E+08	Genevieve geneviev		56	79	20	779	0	0	Sat Jun 23 Arizona	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			
25	6.16E+08	Awilda Fo awildafagxv		11	59	16	622	0	0	Sat Jun 23 bangalore	1 http://a0.NULL		1	https://sil	333333	https://sil	CODEED	DDEEF			

Screen 5.2.2 Data Sets

Description: - To Predicate one type of Facebook Data to check the data to identify profile



```
C:\Windows\System32\cmd.exe - python manage.py runserver
D:\Fake_Profile_Identification>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
Remote_User.ClientRegister_Model: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
    HINT: Configure the DEFAULT_AUTO_FIELD setting or the ClientSiteConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Remote_User.detection_accuracy: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
    HINT: Configure the DEFAULT_AUTO_FIELD setting or the ClientSiteConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Remote_User.detection_ratio: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
    HINT: Configure the DEFAULT_AUTO_FIELD setting or the ClientSiteConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Remote_User.profile_identification_type: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
    HINT: Configure the DEFAULT_AUTO_FIELD setting or the ClientSiteConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 4 issues (0 silenced).

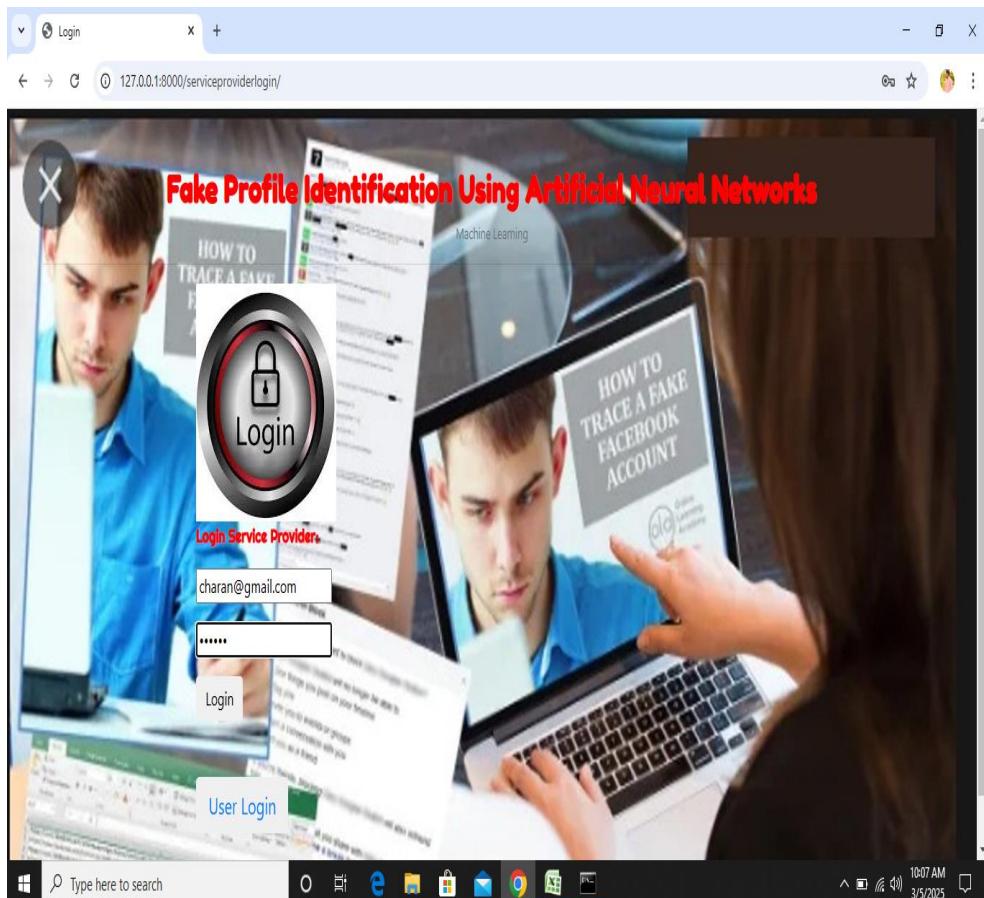
You have 4 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth.
Run 'python manage.py migrate' to apply them.
March 05, 2025 - 10:06:12
Django version 3.2.14, using settings 'fake_profile_identification.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Screeen 5.2.3: Start Project Execution

Description: - To start the server type python manage.py runserver on command prompt it provide a link, that link provide our project web page

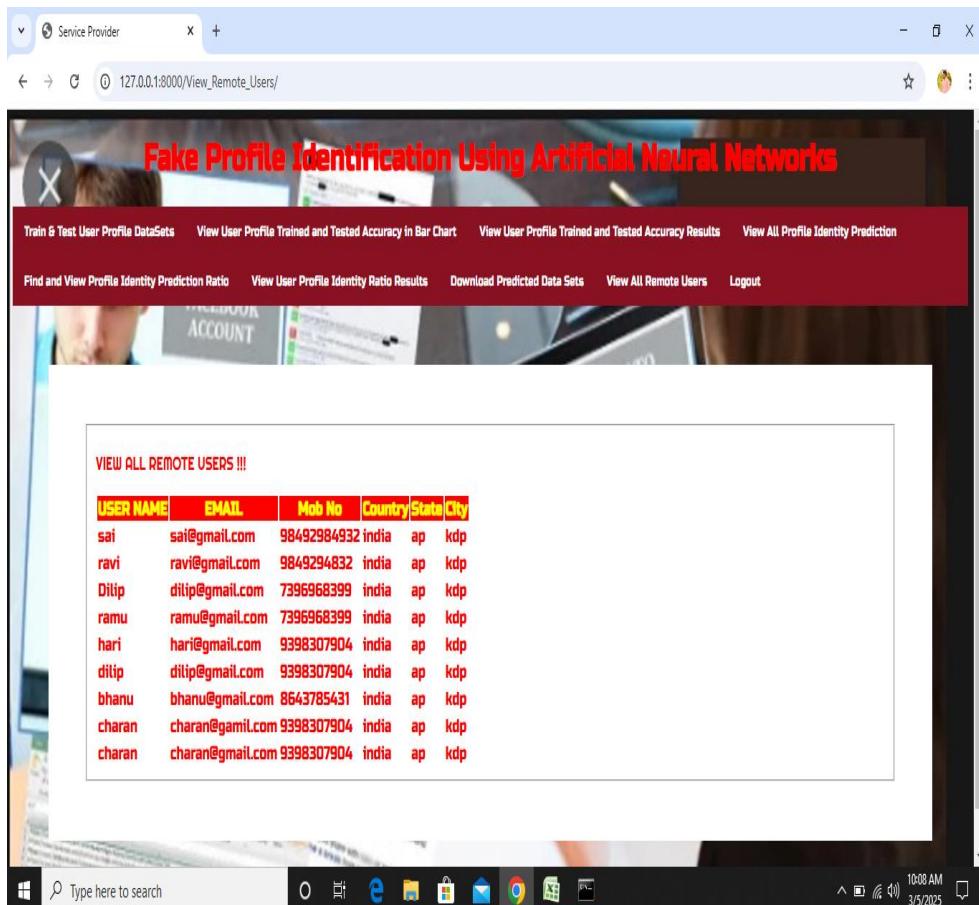
**Screen 5.2.4:User Login**

Description: - It shows the project web page, in this page to login an account

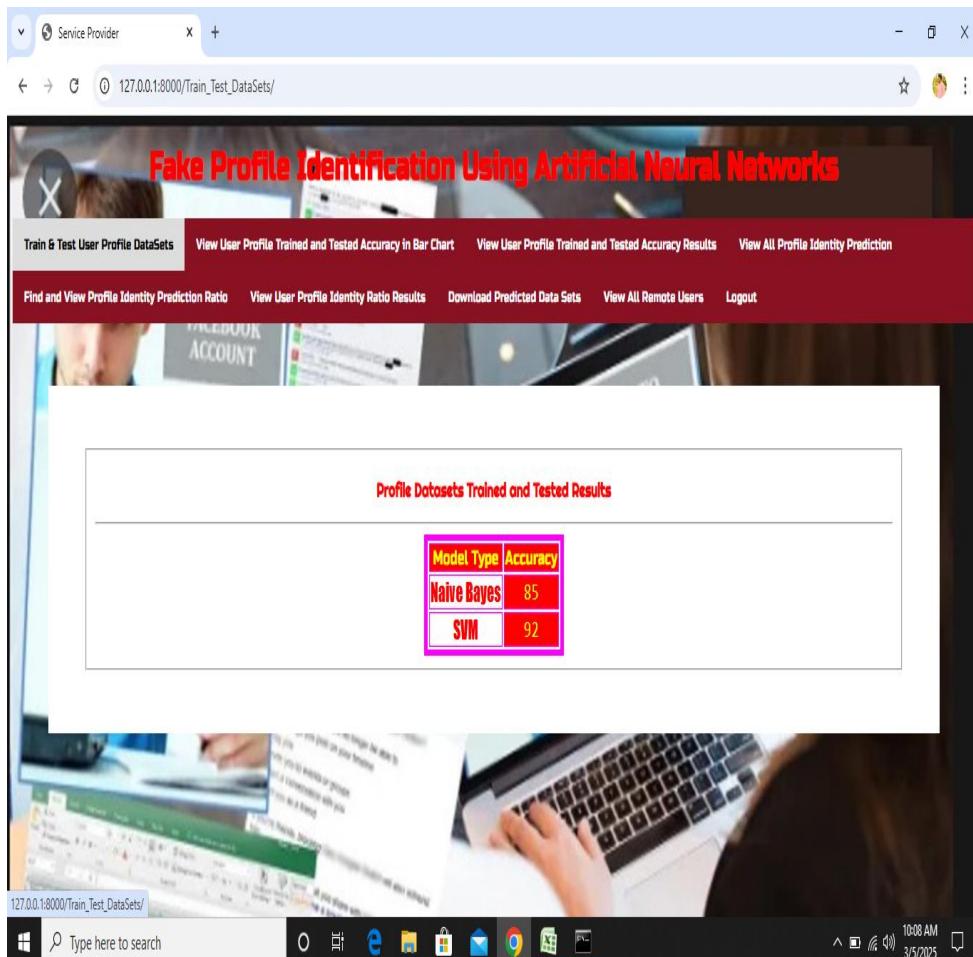


Screen 5.2.5: Service Provider Login

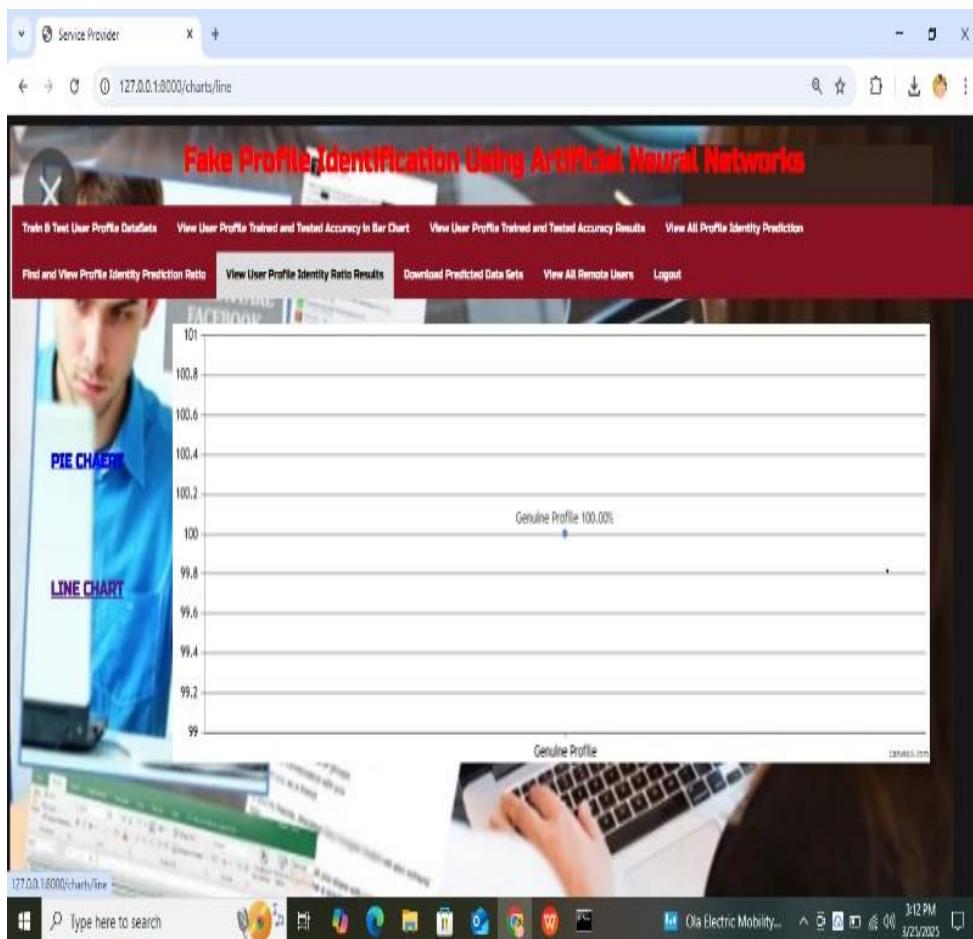
Description: - To Login the Service Provider Page

**Screen 5.2.6: Service Provider Menu**

Description: It shows the View all Service provider information

**Screen 5.2.7: Accuracy Ratio**

Description: It shows the View predication of Fake profile type ratio



Screen 5.2.8: Genuine profile

Description: Genuine profile 100.00%

Service Provider

127.0.0.1:8000/View_Profile_Identity_Prediction/

Fake Profile Identification Using Artificial Neural Networks

Train & Test User Profile Datasets View User Profile Trained and Tested Accuracy in Bar Chart View User Profile Trained and Tested Accuracy Results View All Profile Identity Prediction Find and View Profile Identity Prediction Ratio

View User Profile Identity Ratio Results Download Predicted Data Sets View All Remote Users Logout

View All Profile Status Prediction Type !!

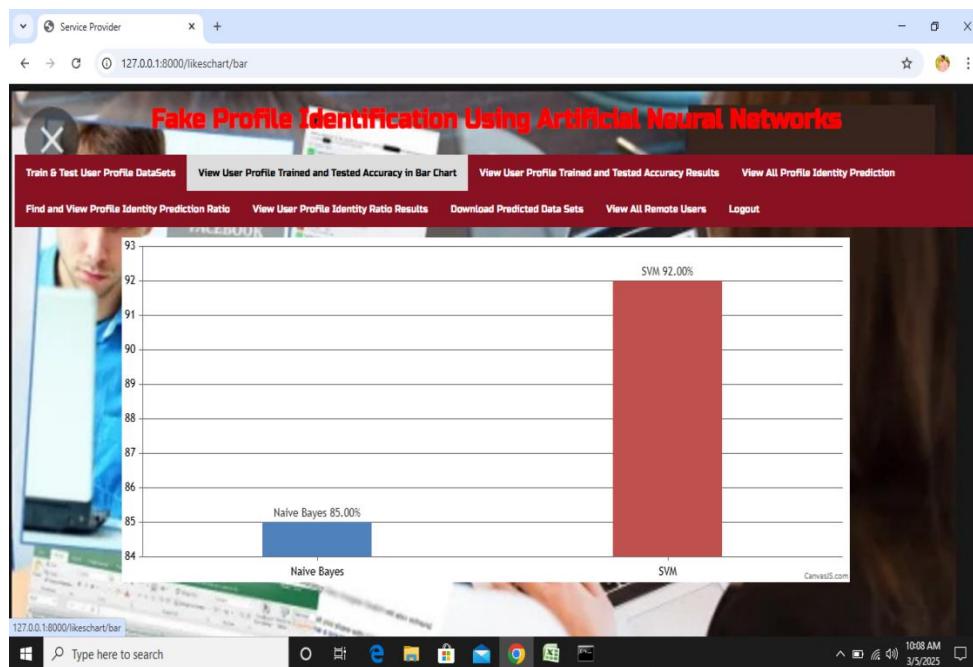
prof_id	Profile_Name	screen_name	statuses_count	followers_count	friends_count	created_at	location	default_profile	prediction
37384589	SAK Nair	bsknair1967	898	656	57				
303932719	DATTARAM PARAB	DATTARAMPARAB	1378	48	1998	Mon May 23 17:15:15 +0000 2011			
370098498	pirfectmoses	pirfectmoses	24	4	588	Thu Sep 08 13:20:35 +0000 2011			
37384589	SAK Nair	bsknair1967	898	656	57	Mon May 23 17:15:15 +0000 2011	kadapa	1	

Type here to search

Ola Electric Mobility... 3:13 PM 3/25/2025

Screen 5.2.9: All Profile status

Description: View all profile statuses prediction

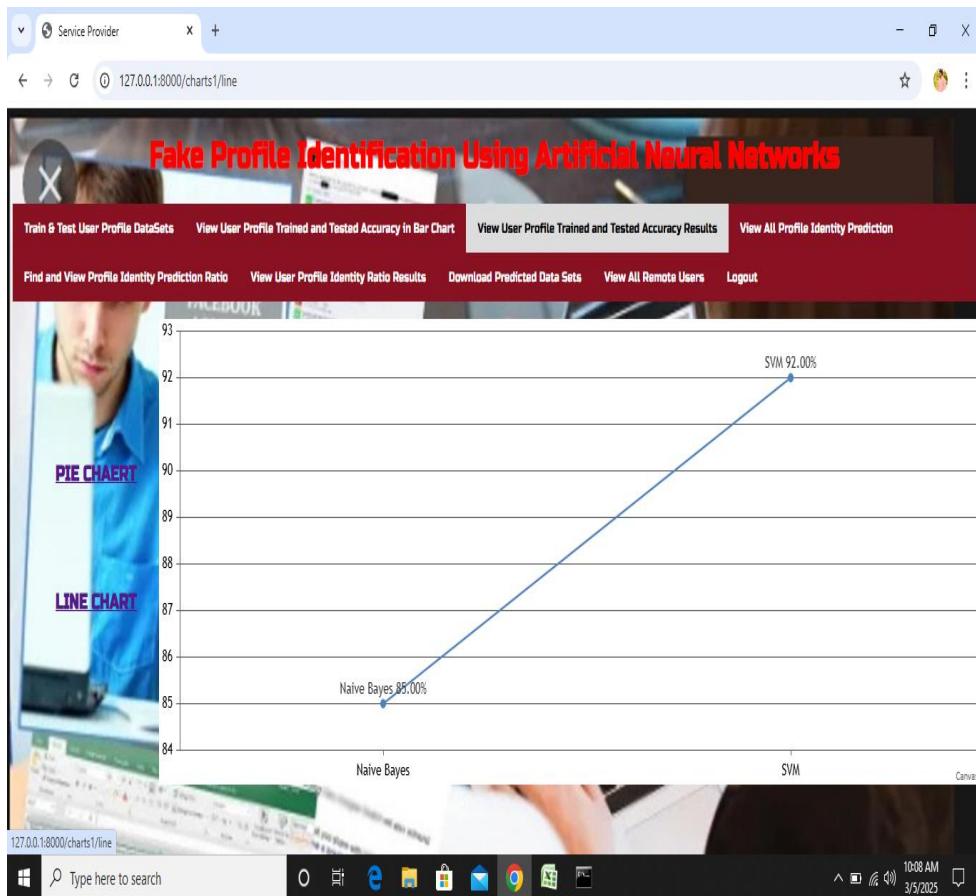


Screen 5.2.10: Accuracy In Bar Chart

Description: It shows the view trained and tested datasets accuracy in bar chart.

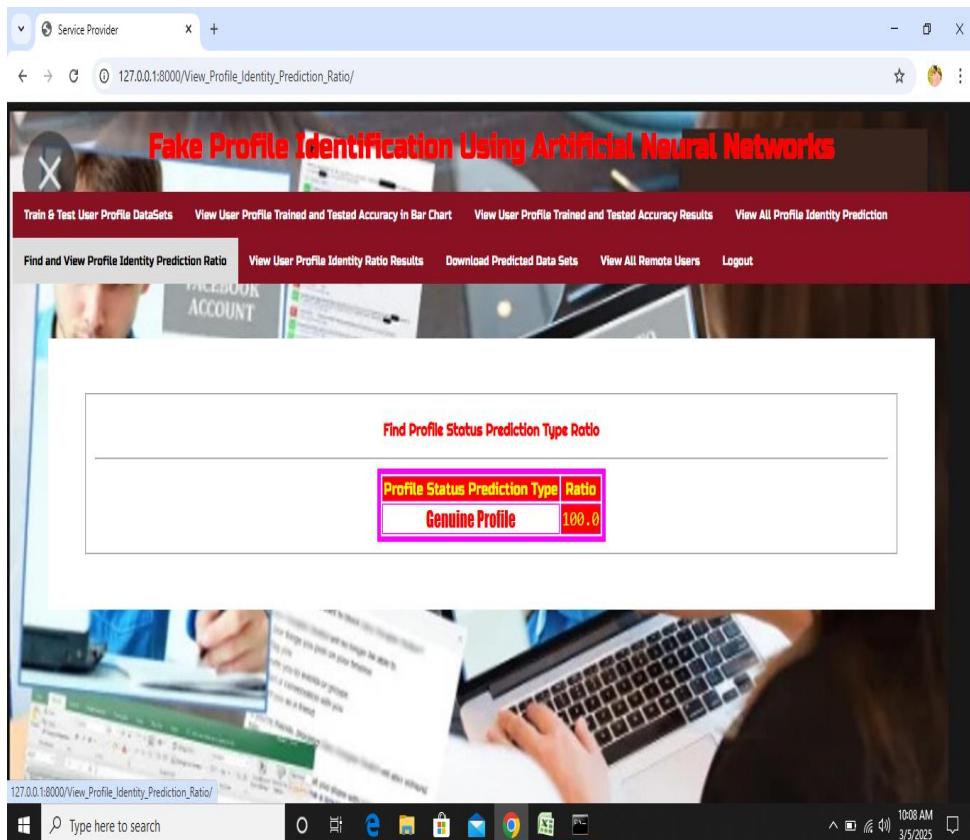
Implementation

Fake Profile Identification Using Artificial Neural Network



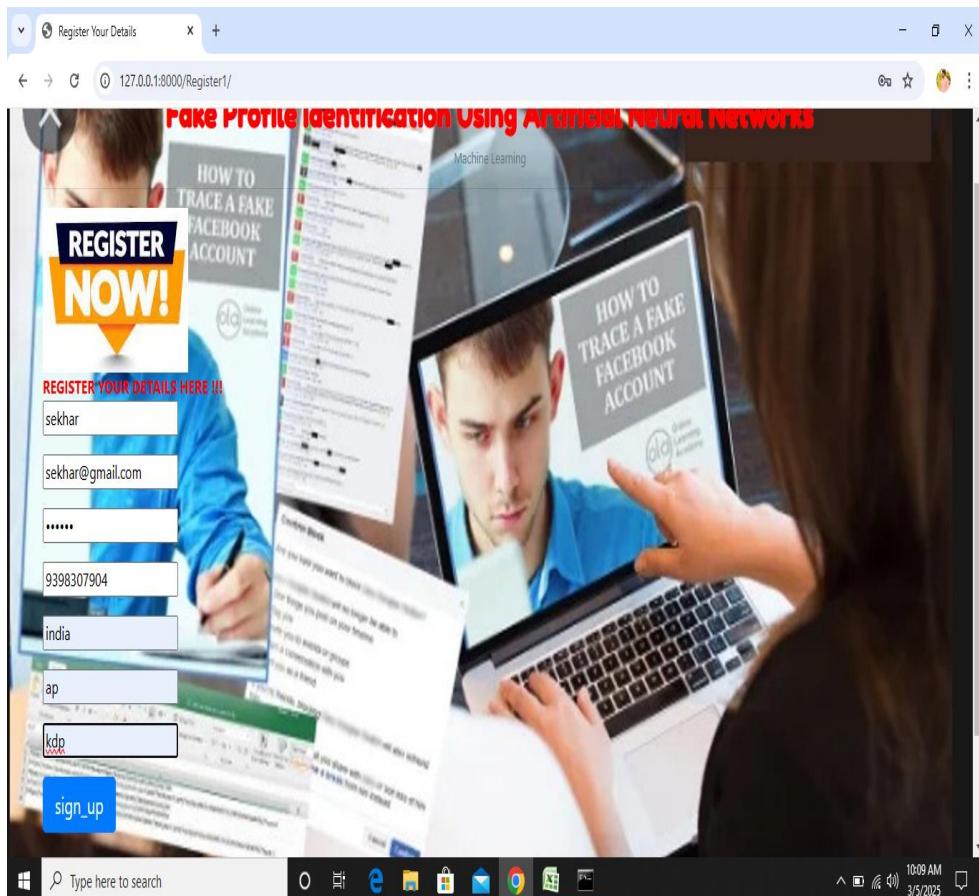
Screen 5.2.11: Accuracy In Line Chart

Description: It shows view trained and tested datasets accuracy result in form of Line Chart.



Screen 5.2.12: Ratio Of Dataset

Description: To find the Prediction Ratio 100 Percentage Profile.

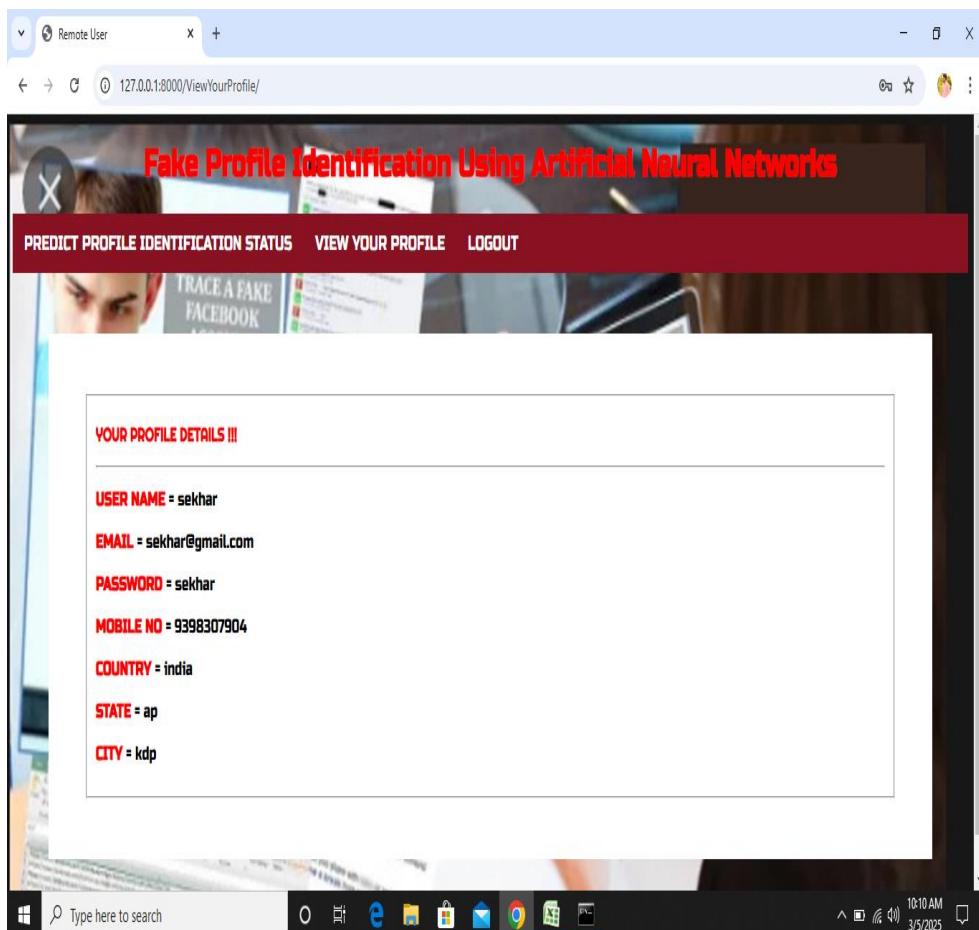


Screen 5.2.13: User Registration

Description: User Register Page.

**Screen 5.2.14: User Login**

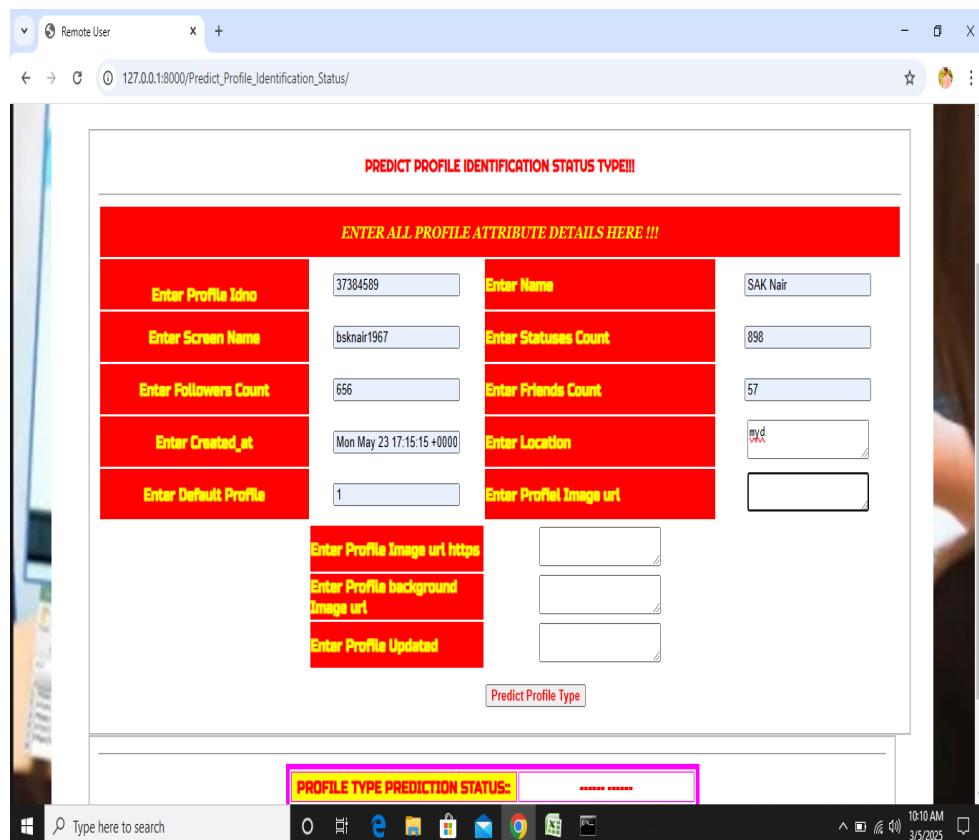
Description: User Register After Login page.

**Screen 5.2.15: User Profile**

Description: User Profile Data.

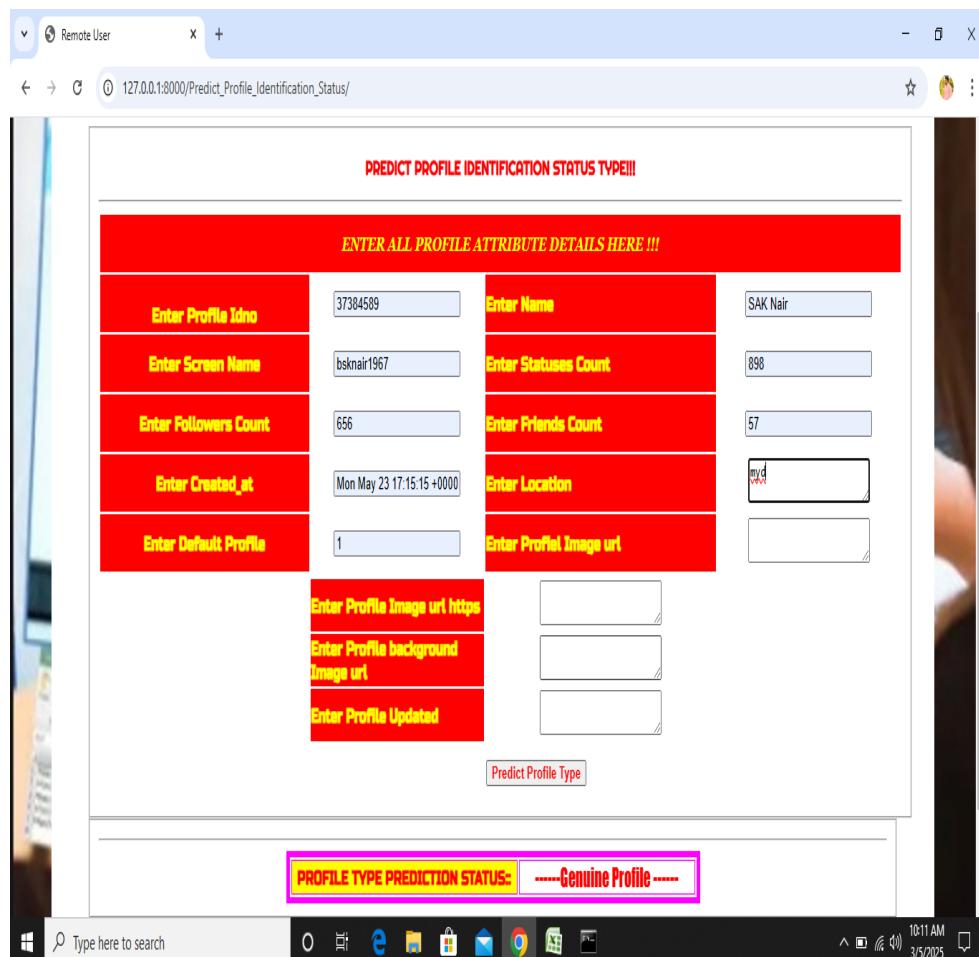
Implementation

Fake Profile Identification Using Artificial Neural Network



Screen 5.2.16: User Prediction Of Fake Profile

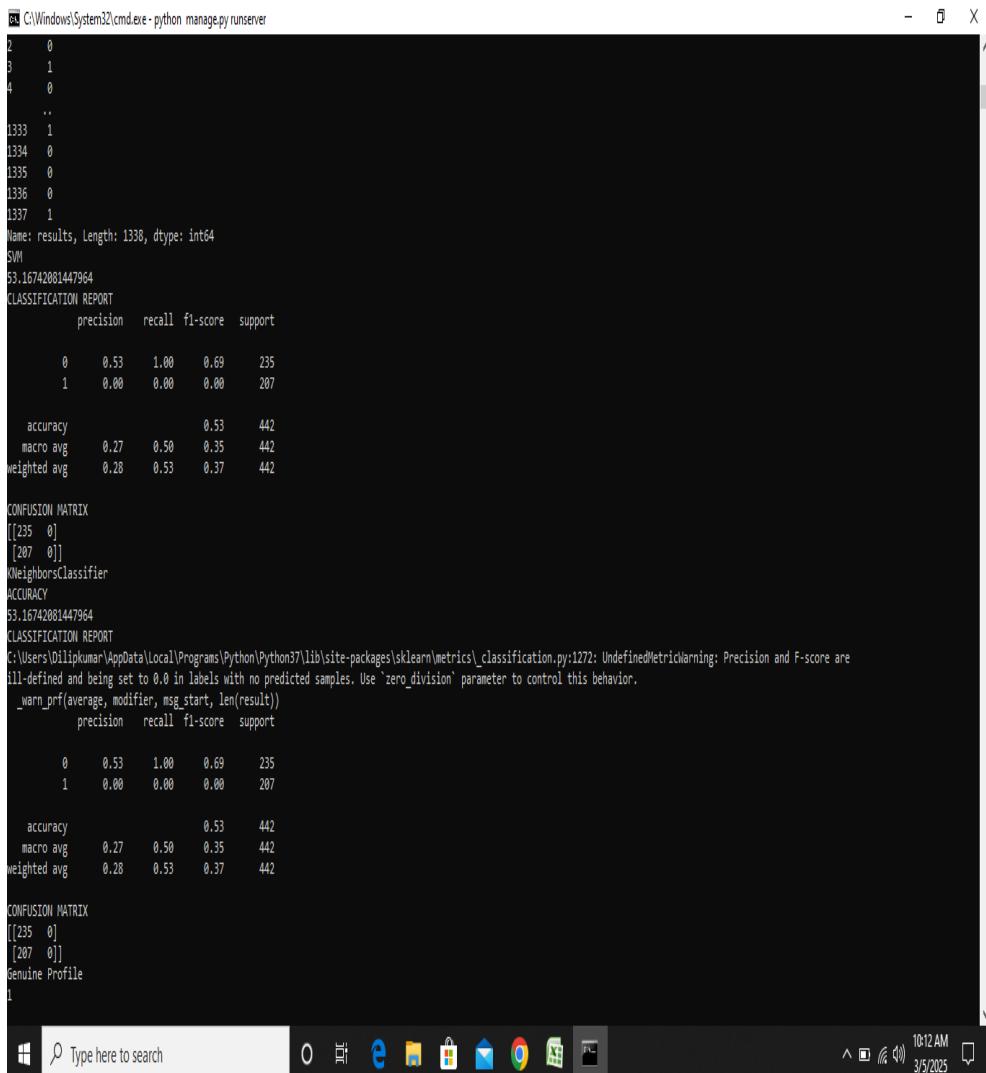
Description: It shows the Predicate Genuine Or Fake types.

**Screen 5.2.17: Result Of Prediction**

Description: It shows the Predicate Genuine Profile.

Implementation

Fake Profile Identification Using Artificial Neural Network



```
C:\Windows\System32\cmd.exe - python manage.py runserver
2    0
3    1
4    0
...
1333  1
1334  0
1335  0
1336  0
1337  1
Name: results, Length: 1338, dtype: int64
53.16742081447964
CLASSIFICATION REPORT
precision    recall   f1-score   support
          0       0.53      1.00      0.69      235
          1       0.00      0.00      0.00      207

accuracy                           0.53      442
macro avg       0.27      0.50      0.35      442
weighted avg    0.28      0.53      0.37      442

CONFUSION MATRIX
[[235  0]
 [207  0]]
KNeighborsClassifier
ACCURACY
53.16742081447964
CLASSIFICATION REPORT
C:\Users\Olipkumar\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\metrics\_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
precision    recall   f1-score   support
          0       0.53      1.00      0.69      235
          1       0.00      0.00      0.00      207

accuracy                           0.53      442
macro avg       0.27      0.50      0.35      442
weighted avg    0.28      0.53      0.37      442

CONFUSION MATRIX
[[235  0]
 [207  0]]
Genuine Profile
1
```

Screen 5.2.18: Accuracy Values

Description: To see the Genuine Profile In command prompt.

CONCLUSION

6. CONCLUSION

In conclusion, the project successfully demonstrates the effectiveness of artificial neural networks in identifying fake profiles. By leveraging data patterns such as profile activity, content authenticity, and behavioral traits, the model achieves accurate detection results. The system efficiently analyzes complex data features, improving identification accuracy compared to traditional methods. This approach enhances online security by minimizing the risks associated with fraudulent profiles. Future enhancements may involve refining the model with additional datasets and advanced algorithms.

6.1 FUTURE ENHANCEMENT

Each input neuron would be a different, previously chosen feature of each profile converted into a numerical value (e.g., gender as a binary number, female 0 and male 1) and if needed, divided by an arbitrary number (e.g., age is always divided by 100) to minimize one feature having more influence on the result than the other. The neurons represent nodes. Each node would be responsible for exactly one decision-making process

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] <https://www.statista.com/topics/1164/social-networks/>
- [2] <https://www.cnbc.com/2018/01/31/facebook-earnings-q4-2017-arpu.html>
- [3] <https://www.cnet.com/news/facebook-breach-affected-50-millionpeople/> [4] <https://www.facebook.com/policy.php>
- [5] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12). USENIX Association, Berkeley, CA, USA, 15-15.
- [6] Akshay J. Sarode and Arun Mishra. 2015. Audit and Analysis of Impostors: An experimental approach to detect fake profile in an online social network. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015 (ICCCT '15). ACM, New York, NY, USA, 1-8. DOI: <https://doi.org/10.1145/2818567.2818568>
- [7] Devakunchari Ramalingam, Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering, Volume 65, 2018, Pages 165-177, ISSN 0045-7906, <https://doi.org/10.1016/j.compeleceng.2017.05.020>.
- [8] <https://www.enigmasoftware.com/top-20-countries-the-most-cybercrime>
- [9] pages.cs.wisc.edu/~bolo/shipyard/neural/local.html
- [10] <https://stackoverflow.com/questions/40758562/can-anyone-explain-mestandardscaler>
- [11] Dr. S. Kannan, Vairaprakash Gurusamy, “Preprocessing Techniques for Text Mining”, 05 March 2015.
- [12] Shalinda Adikari and Kaushik Dutta, Identifying Fake Profiles in LinkedIn, PACIS 2014 Proceedings, AISeL
- [13] Z. Halim, M. Gul, N. ul Hassan, R. Baig, S. Rehman, and F. Naz, “Malicious users’ circle detection in social network based on

Bibliography

Fake Profile Identification Using Artificial Neural Network

- spatiotemporal co-occurrence,” in Computer Networks and Information Technology (ICCNIT),2011 International Conference on, July, pp. 35–390.
- [14] Liu Y, Gummadi K, Krishnamurthy B, Mislove A,” Analyzing Facebook privacy settings: User expectations vs. reality”, in: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference,ACM,pp.61–70.
- [15] Mahmood S, Desmedt Y,” Poster: preliminary analysis of google?’s privacy. In: Proceedings of the 18th ACM conference on computer and communications security”, ACM 2011, pp.809–812.
- [16] Stein T, Chen E, Mangla K,” Facebook immune system. In: Proceedings of the 4th workshop on social network systems”, ACM 2011, pp
- [17] Saeed Abu-Nimeh, T. M. Chen, and O. Alzubi, “Malicious and Spam Posts in Online Social Networks,” Computer, vol.44, no.9, IEEE2011, pp.23– 28.
- [18] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, B. Zhao, Understanding latent interactions in online social networks, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, ACM, 2010, pp. 369–382
- [19] Kazienko, P. and K. Musiał (2006). Social capital in online social networks. Knowledge-Based Intelligent Information and Engineering Systems, Springer
- [20] <https://pandas.pydata.org>
- [21] https://www.tutorialspoint.com/python_pandas/index.htm
- [22] <http://www.numpy.org>
- [23] <https://www.mathworks.com/products/matlab.html>
- [24] <http://www.deeplearning.net/software/theano/>
- [25] <https://scikit-learn.org/stable/>
- [26] <https://keras.io>
- [27] <https://www.tensorflow.org>

APPENDIX – A**URL LISTING**

- www.google.co.in
- www.python.org
- www.w3schools.com
- www.pythontutorial.com

REFERENCE BOOKS

- Python Crash Course 2nd Edition - this is a basic level book for beginners.
- Learning python 5th Edition - this book is a practical learning book for basic to advanced level.
- Python Cookbook - this book for advanced programmer interested in learning about modern python development tools.
- Automating Boring Stuff With Python - In this book you will learn to write programs in python.
- Head First Python - this book covered the fundamental of python.

Think Python - the basics of programming concepts and cover advanced topics like data structure and object-oriented design.

APPENDIX – B**GLOSSARY**

- GUI : Graphical User Interface
- UML : Unified Modeling Language
- API : Application Programming Interface
- HTML : Hyper Text Markup Language
- URL : Uniform Resource Locator
- ODBC : Open Database Connectivity

APPENDIX – C**List of Figures**

S. No.	Fig No.	Description	Page No.	Chapter
1	2.2.1.1	System Architecture	10	SRS
2	3.1.1	Data Flow Diagram	23	System Design
3	3.1.2	Er Diagram Components	24	System Design
4	3.1.3	Er Diagram	29	System Design
5	3.3.1	Class Diagram	38	System Design
6	3.3.2	Use Case Diagram	39	System Design
7	3.3.3	Sequence Diagram	39	System Design
8	3.3.4	Activity Diagram	40	System Design
9	3.3.5	Collaboration Diagram	40	System Design
10	3.3.6	Deployment Diagram	41	System Design
11	3.3.7	State Chart Diagram	41	System Design
12	3.3.8	Component Diagram	42	System Design
13	3.3.1.1	Use Case Diagram For Overall Diagram	44	System Design

Bibliography**Fake Profile Identification Using
Artificial Neural Network**

14	3.3.2.1	Class Diagram For Overall Diagram	45	System Design
15	3.3.3.1	Sequence Diagram For Overall Diagram	46	System Design
16	3.3.4.1	Activity Diagram For Overall Diagram	47	System Design
17	3.3.5.1	Deployment Diagram For Overall Diagram	48	System Design

List of Tables

S. No	Table No	Table Name	Page No.	Chapter
1	3.2.1	User	30	System design
2	3.2.2	Service Provider	30	System design
3	4.2.1	Test cases	53	System design

Bibliography

Fake Profile Identification Using Artificial Neural Network

List of Screen

S. No.	Screen No.	Screen Name	Page No.	Chapter
1	5.2.1	Start Wamp Server	63	Implementation
2	5.2.2	Data Sets	64	Implementation
3	5.2.3	Start Project Execution	65	Implementation
4	5.2.4	User Login	66	Implementation
5	5.2.5	Service Provider Login	67	Implementation
6	5.2.6	Service Provider Menu	68	Implementation
7	5.2.7	Accuracy Ratio	69	Implementation
8	5.2.8	Genuine profile	70	Implementation
9	5.2.9	All Profile status	71	Implementation
10	5.2.10	Accuracy In Bar Chart	72	Implementation
11	5.2.11	Accuracy In Line Chart	73	Implementation
12	5.2.12	Ratio Of Dataset	74	Implementation
13	5.2.13	User Registration	75	Implementation
14	5.2.14	User Login	76	Implementation

Bibliography**Fake Profile Identification Using
Artificial Neural Network**

15	5.2.15	User Profile	77	Implementation
16	5.2.16	User Prediction Of Fake Profile	78	Implementation
17	5.2.17	Result Of Prediction	79	Implementation
18	5.2.18	Accuracy Values	80	Implementation

CODING

APPENDIX – D**CODING**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>HOME PAGE</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/droid_sans_400-
droid_sans_700.font.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style2 {

color: #FF0000;

font-style: italic;

font-weight: bold;

}
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
.style3 { color: #FF0000}

.style4 {
    font-size: 36px;
    color: #FF0000;
    font-weight: bold;
}

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home Page</span></a></li>

<li><strong><a href="DataUser.jsp">Data User </a></strong></li>

<li><a href="CloudServer.jsp"><span>Cloud Server </span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="logo">

<h1 class="style4">A Verifiable Cloud Data Deduplication Scheme with Integrity and Duplication Proof</h1>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> <a href="#"> </a> <a href="#"> </a> <a href="#"> </a> </div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2 class="style3"></h2>

<p class="infopost"></p>

<div class="clr"></div>

<div class="img"></div>

<div class="post_content">

<p align="justify" class="style2"></p>

</div>

<div class="clr"></div>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
</div>

</div>

<div class="sidebar">

<div class="searchform"></div>

<div class="clr"></div>

<div class="gadget">

<li><strong><a href="index.html">Home</a></strong></li>

<li><strong><a href="DataUser.jsp">Data User
</a></strong></li>

<li><strong><a href="CloudServer.jsp">Cloud Server
</a></strong></li>

class="star"><span></span> </h2>

<div class="clr"></div>

<ul class="sb_menu">

</ul>

</div>

</div>

</div>

<div class="fbg"></div>

<div class="footer">

</html>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<%@ page language="java" contentType="text/html; charset=ISO-  
8859-1"  
  
    pageEncoding="ISO-8859-1"%>  
  
<%@page import="java.util.*"%>  
  
<%@ include file="connect.jsp"%>  
  
<%@page  
  
    import="java.util.*,java.security.Key,java.util.Random,javax.crypto.  
Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders  
.Base64"%>  
  
<%@ page  
  
    import="java.sql.* ,java.util.Random,java.io.PrintStream,java.io.File  
OutputStream,java.io.FileInputStream,java.security.DigestInputStream,j  
ava.math.BigInteger,java.security.MessageDigest,java.io.BufferedInput  
Stream"%>  
  
<%@ page  
  
    import="java.security.Key,java.security.KeyPair,java.security.KeyP  
airGenerator,javax.crypto.Cipher"%>  
  
<%@page  
  
    import="java.util.* ,java.text.SimpleDateFormat,java.util.Date,java.i  
o.FileInputStream,java.io.FileOutputStream,java.io.PrintStream,java.m  
ath.*"%>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>

<title>Upload1</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/droid_sans_400-
droid_sans_700.font.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style3 {

    color: #FF0000;
    font-weight: bold;
}

.style4 {color: #0000FF}

.style10 {

    font-size: 24px;
    color: #FF0000;
    font-weight: bold;
}

.style13 {color: #FFFF00; font-weight: bold; }

-->
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home
Page</span></a></li>

<li><a href="DataUser.jsp"><span>Data User </span></a></li>

<li><a href="CloudServer.jsp"><span>Cloud Server
</span></a></li>

<li><a href="EndUser.jsp"><span>End User </span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="logo">

<h1> </h1>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> <a href="#"> </a> <a href="#"></a><a href="#"></a></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2><span><span class="style10"> Data Replication of Fragmentations Has Found!!!</span> </h2>

<div class="clr"></div>

<div class="img"></div>

<div class="post_content">

<% try

{

String file=request.getParameter("tt");

String cont=request.getParameter("text");

String keys = "ef50a0ef2c3e3a5f";

String h[]=new String[5];
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
String filename="filename.txt";  
  
String filename1="filename1.txt";  
  
String filename2="filename2.txt";  
  
String filename3="filename3.txt";  
  
String filename4="filename4.txt";  
  
ResultSet  
  
rs=connection.createStatement().executeQuery("select * from  
cloudserver where fname='"+file+"' ");  
  
if(!rs.next())  
  
{  
  
int sourceFileSize=cont.length();  
  
int s=sourceFileSize%4;  
  
int s1=sourceFileSize/4;  
  
int a1=s1;  
  
int a2=s1+s1;  
  
int a3=s1+s1+s1;  
  
int a4=s1+s1+s1+s1+s;  
  
  
String  
  
encryptedValue3="",encryptedValue2="",encryptedValue1="",encrypte  
dValue4="";  
  
//      String keys = "ef50a0ef2c3e3a5f";  
  
  
byte[] keyValue = keys.getBytes();  
  
Key key = new SecretKeySpec(keyValue,  
"AES");
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
Cipher c = Cipher.getInstance("AES");
c.init(Cipher.ENCRYPT_MODE, key);

String encryptedValue = new
String(Base64.encode(cont.getBytes()));

encryptedValue1 = new String(Base64

.encode(cont.substring(0,a1).getBytes()));

encryptedValue2 = new String(Base64

.encode(cont.substring(a1+1,a2).getBytes()));

encryptedValue3 = new String(Base64

.encode(cont.substring(a2+1,a3).getBytes()));

encryptedValue4 = new String(Base64

.encode(cont.substring(a3+1,a4).getBytes()));

/*encryptedValue1=cont.substring(0,a1);
encryptedValue2=cont.substring(a1+1,a2);
encryptedValue3=cont.substring(a2+1,a3);
encryptedValue4=cont.substring(a3+1,a4);*/



//int NSplits=sourceFileSiz/5

%>

source<%=sourceFileSize %>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
a1<%=a1 %>
```

```
a2<%=a2 %>
```

```
a3<%=a3 %>
```

```
a4<%=a4 %>
```

```
split<%=s %>
```

```
<%
```

```
PrintStream p1 = new PrintStream(new  
FileOutputStream("E:/deypos/filename1.txt"));
```

```
p1.print(new String(encryptedValue1));
```

```
PrintStream p2 = new PrintStream(new  
FileOutputStream("E:/deypos/filename2.txt"));
```

```
p2.print(new String(encryptedValue2));
```

```
PrintStream p3 = new PrintStream(new  
FileOutputStream("E:/deypos/filename3.txt"));
```

```
p3.print(new String(encryptedValue3));
```

```
PrintStream p4 = new PrintStream(new  
FileOutputStream("E:/deypos/filename4.txt"));
```

```
p4.print(new String(encryptedValue4));
```

```
for(int i=1;i<5;i++)
```

```
{
```

```
MessageDigest md =  
MessageDigest.getInstance("SHA1");
```

```
FileInputStream fis11 = new  
FileInputStream("E:/deypos/filename"+i+".txt");
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
DigestInputStream dis1 = new  
DigestInputStream(fis1, md);  
  
BufferedInputStream bis1 = new  
BufferedInputStream(dis1);  
  
  
//Read the bis so SHA1 is auto calculated at dis  
  
while (true) {  
  
    int b1 = bis1.read();  
  
    if (b1 == -1)  
  
        break;  
  
}  
  
  
BigInteger bi1 = new BigInteger(md.digest());  
  
String spl1 = bi1.toString();  
  
h[i] = bi1.toString(16);  
  
}  
  
  
%>  
  
<div class="clr"></div>  
  
<form action="DDUploadBlocks2.jsp" method="post">  
  
<table width="598" border="1" align="center">
```

```
<tr>

    <td width="286" bgcolor="#FF0000"><span
class="style13">File Name :- </span></td>

    <td width="356"><input name="t42" type="text" id="t42"
size="50"

        value=<%= file %>" readonly="readonly" /></td>

</tr>

<tr>

    <td width="286" bgcolor="#FF0000"><span
class="style13">Fragment-1 </span></td>

    <td width="286"><span
class="style4">Size:<%=a1%>Bytes</span></td>

</tr>

<tr>

    <td bgcolor="#FF0000"> </td>

    <td><textarea name="text1" cols="50" rows="15"
readonly><%= encryptedValue1 %></textarea></td>

</tr>

<tr>

    <td bgcolor="#FF0000"><span class="style13">MAC-1:
</span></td>

    <td><input name="t1" type="text" id="t4" size="50"
value=<%= h[1] %>" readonly="readonly" /></td>

</tr>

<tr>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<td width="286" bgcolor="#FF0000"><span  
class="style13">Fragment-2 </span></td>  
  
<td width="286"><span class="style4">Size:<%=(a2-  
a1)%>Bytes</span></td>  
  
</tr>  
  
<tr>  
  
<td bgcolor="#FF0000"> </td>  
  
<td><textarea name="text2" cols="50" rows="15"  
readonly><%= encryptedValue2 %></textarea></td>  
  
</tr>  
  
<tr>  
  
<td bgcolor="#FF0000"><span class="style13">MAC-2:  
</span></td>  
  
<td><input name="t2" type="text" id="t4" size="50"  
value=<%= h[2] %>" readonly="readonly" /></td>  
  
</tr>  
  
<tr>  
  
<td width="286" bgcolor="#FF0000"><span  
class="style13">Fragmentk-3 </span></td>  
  
<td width="286"><span class="style4">Size:<%=(a3-  
a2)%>Bytes</span></td>  
  
</tr>  
  
<tr>  
  
<td bgcolor="#FF0000"> </td>  
  
<td><textarea name="text3" cols="50" rows="15"  
readonly><%= encryptedValue3 %></textarea></td>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
</tr>

<tr>

    <td bgcolor="#FF0000"><span class="style13">MAC-3:</span></td>

    <td><input name="t3" type="text" id="t4" size="50" value="<% = h[3] %>" readonly="readonly" /></td>

</tr>

<tr>

    <td width="286" bgcolor="#FF0000"><span class="style13">Fragment-4 </span></td>

    <td width="286"><span class="style4">Size:<% = (a4-a3)%>Bytes</span></td>

</tr>

<tr>

    <td bgcolor="#FF0000"> </td>

    <td><textarea name="text4" cols="50" rows="15" readonly><% = encryptedValue4 %></textarea></td>

</tr>

<tr>

    <td bgcolor="#FF0000"><span class="style13">Fragment-4: </span></td>

    <td><input name="t4" type="text" id="t4" size="50" value="<% = h[4] %>" readonly="readonly" /></td>

</tr>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<tr>

    <td> </td>

    <td> </td>

</tr>

<tr>

    <td>

        <div align="right"></div> </td>

    <td><input type="submit" name="Submit2" value="Find DeDuplication" /></td>

</tr>

</table>

</form>

<%
}

else

{

%>

<h1 class="style10">File name already exist</h1>

<a href="O_UploadFile.html">Back</a>

<%
}

connection.close();

}

catch(Exception e)
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
{  
}  
%>  
</div>  
  
<div class="clr">sadf dasfd asdfa</div>  
  
</div>  
  
</div>  
  
<div class="sidebar">  
  
<div class="searchform">  
  
<form id="formsearch" name="formsearch" method="post"  
action="#">  
  
<span>  
  
<input name="editbox_search" class="editbox_search"  
id="editbox_search" maxlength="80" value="Search our ste:"  
type="text" />  
  
</span>  
  
<input name="button_search" src="images/search.gif"  
class="button_search" type="image" />  
  
</form>  
  
</div>  
  
<div class="clr"></div>  
  
<div class="gadget">  
  
<h2 class="star"><span>Owner Main </span> Menu</h2>  
  
<div class="clr"></div>
```

```
<ul class="ex_menu style3"><li class="style4">
<li><a href="OMainPage.jsp">Home</a></li>
<li><a href="index.html">Logout</a></li>
<li></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg"></div>
<div class="footer">
</html>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.util.*"%>
<%@ include file="connect.jsp"%>
<%@page
import="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>
<%@ page
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
import="java.sql.* ,java.util.Random,java.io.PrintStream,java.io.File
OutputStream,java.io.FileInputStream,java.security.DigestInputStream,j
ava.math.BigInteger,java.security.MessageDigest,java.io.BufferedInput
Stream">%>

<%@ page

    import="java.security.Key,java.security.KeyPair,java.security.KeyP
airGenerator,javax.crypto.Cipher"%>

<%@page

    import="java.util.* ,java.text.SimpleDateFormat,java.util.Date,java.i
o.FileInputStream,java.io.FileOutputStream,java.io.PrintStream,java.m
ath.*"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Upload</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/droid_sans_400-
droid_sans_700.font.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>
```

```
<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style3 {

color: #FF0000;

font-weight: bold;

}

.style4 {color: #0000FF}

.style10 {

font-size: 24px;

color: #FF0000;

font-weight: bold;

}

.style13 {color: #FFFF00; font-weight: bold; }

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home</span></a></li>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<li><a href="DataUser.jsp"><span>Data User </span></a></li>

<li><a href="CloudServer.jsp"><span>Cloud Server
</span></a></li>

<li><a href="EndUser.jsp"><span>End User </span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="logo">

<h1> </h1>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> <a href="#"> </a> <a href="#"> </a> <a
href="#"> </a> </div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">
```

```
<div class="article">

<h2><span>Confirm Upload Fragments !!! </h2>

<div class="clr"></div>

<div class="img"></div>

<div class="post_content">

<% try

{

    String file=request.getParameter("tt");

    String cont=request.getParameter("text");

    String keys = "ef50a0ef2c3e3a5f";

    String h[]=new String[5];

    String filename="filename.txt";

    String filename1="filename1.txt";

    String filename2="filename2.txt";

    String filename3="filename3.txt";

    String filename4="filename4.txt";

    ResultSet

    rs=connection.createStatement().executeQuery("select * from

cloudserver where fname='"+file+"' ");

    if(!rs.next())

    {

        int sourceFileSize=cont.length();
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
int s=sourceFileSize%4;
```

```
int s1=sourceFileSize/4;
```

```
int a1=s1;
```

```
int a2=s1+s1;
```

```
int a3=s1+s1+s1;
```

```
int a4=s1+s1+s1+s1+s;
```

String

```
encryptedValue3 = "", encryptedValue2 = "", encryptedValue1 = "", encryptedValue4 = "";
```

```
//     String keys = "ef50a0ef2c3e3a5f";
```

```
byte[] keyValue = keys.getBytes();
```

```
Key key = new SecretKeySpec(keyValue,  
"AES");
```

```
Cipher c = Cipher.getInstance("AES");
```

```
c.init(Cipher.ENCRYPT_MODE, key);
```

```
String encryptedValue = new
```

```
String(Base64.encode(cont.getBytes()));
```

```
encryptedValue1 = new String(Base64
```

```
.encode(cont.substring(0,a1).getBytes()));
```

```
encryptedValue2 = new String(Base64
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
.encode(cont.substring(a1+1,a2).getBytes())));  
  
encryptedValue3 = new String(Base64  
.encode(cont.substring(a2+1,a3).getBytes()));  
  
encryptedValue4 = new String(Base64  
.encode(cont.substring(a3+1,a4).getBytes()));  
  
%>  
source<%=sourceFileSize %>  
a1<%=a1 %>  
a2<%=a2 %>  
a3<%=a3 %>  
a4<%=a4 %>  
split<%=s %>  
<%  
PrintStream p1 = new PrintStream(new  
FileOutputStream("filename1.txt"));  
p1.print(new String(encryptedValue1));  
PrintStream p2 = new PrintStream(new  
FileOutputStream("filename2.txt"));
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
p2.print(new String(encryptedValue2));  
  
PrintStream p3 = new PrintStream(new  
FileOutputStream("filename3.txt"));  
  
p3.print(new String(encryptedValue3));  
  
PrintStream p4 = new PrintStream(new  
FileOutputStream("filename4.txt"));  
  
p4.print(new String(encryptedValue4));
```

```
for(int i=1;i<5;i++)  
{  
  
    MessageDigest md =  
    MessageDigest.getInstance("SHA1");  
  
    FileInputStream fis11 = new  
    FileInputStream("filename"+i+".txt");  
  
    DigestInputStream dis1 = new  
    DigestInputStream(fis11, md);  
  
    BufferedInputStream bis1 = new  
    BufferedInputStream(dis1);  
  
    //Read the bis so SHA1 is auto calculated at dis  
  
    while (true) {  
  
        int b1 = bis1.read();  
  
        if (b1 == -1)  
  
            break;  
  
    }  
  
  
    BigInteger bi1 = new BigInteger(md.digest());
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
String spl1 = bi1.toString();  
  
h[i] = bi1.toString(16);  
  
}  
  
%>  
  
<div class="clr"></div>  
  
  
  
<form action="OUploadBlocks2.jsp" method="post">  
  
<table width="598" border="1" align="center">  
  
  
  
<tr>  
  
    <td width="286" bgcolor="#FF0000"><span  
class="style13">File Name :- </span></td>  
  
    <td width="356"><input name="t42" type="text" id="t42"  
size="50"  
  
        value="<% file %>" readonly="readonly" /></td>  
  
</tr>  
  
<tr>  
  
    <td width="286" bgcolor="#FF0000"><span  
class="style13">Fragment-1 </span></td>  
  
    <td width="286"><span  
class="style4">Size:<% a1 %>Bytes</span></td>  
  
</tr>  
  
<tr>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<td bgcolor="#FF0000"> </td>

<td><textarea name="text1" cols="50" rows="15"
readonly><%= encryptedValue1 %></textarea></td>

</tr>

<tr>

<td bgcolor="#FF0000"><span class="style13">MAC-1:
</span></td>

<td><input name="t1" type="text" id="t4" size="50"
value=<%= h[1] %>" readonly="readonly" /></td>

</tr>

<tr>

<td width="286" bgcolor="#FF0000"><span
class="style13">Fragment-2 </span></td>

<td width="286"><span class="style4">Size:<%=(a2-
a1)%>Bytes</span></td>

</tr>

<tr>

<td bgcolor="#FF0000"> </td>

<td><textarea name="text2" cols="50" rows="15"
readonly><%= encryptedValue2 %></textarea></td>

</tr>

<tr>

<td bgcolor="#FF0000"><span class="style13">MAC-2:
</span></td>

<td><input name="t2" type="text" id="t4" size="50"
value=<%= h[2] %>" readonly="readonly" /></td>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
</tr>

<tr>

    <td width="286" bgcolor="#FF0000"><span
class="style13">Fragment-3 </span></td>

    <td width="286"><span class="style4">Size:<%=(a3-
a2)%>Bytes</span></td>

</tr>

<tr>

    <td bgcolor="#FF0000"> </td>

    <td><textarea name="text3" cols="50" rows="15"
readonly><%= encryptedValue3 %></textarea></td>

</tr>

<tr>

    <td bgcolor="#FF0000"><span class="style13">MAC-3:
</span></td>

    <td><input name="t3" type="text" id="t4" size="50"
value=<%= h[3]%>" readonly="readonly" /></td>

</tr>

<tr>

    <td width="286" bgcolor="#FF0000"><span
class="style13">Fragment-4 </span></td>

    <td width="286"><span class="style4">Size:<%=(a4-
a3)%>Bytes</span></td>

</tr>

<tr>

    <td bgcolor="#FF0000"> </td>
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<td><textarea name="text4" cols="50" rows="15"
readonly><%= encryptedValue4 %></textarea></td>

</tr>

<tr>

    <td bgcolor="#FF0000"><span class="style13">MAC-4:
</span></td>

    <td><input name="t4" type="text" id="t4" size="50"
value=<%= h[4] %>" readonly="readonly" /></td>

</tr>

<tr>
    <td> </td>
    <td> </td>
</tr>

<tr>
    <td>
        <div align="right"></div> </td>
        <td><input type="submit" name="Submit2"
value="Upload" /></td>
    </td>
</tr>
</table>
</form>
<%
}
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
else
{
%>

<h1 class="style10">File name already exist</h1>
<a href="O_UploadFile.html">Back</a>
<%
}

connection.close();

}

catch(Exception e)
{
    out.println(e.getMessage());
}

%>

</div>
<div class="clr">sadf dasfd asdfa</div>
</div>
</div>
<div class="sidebar">
<div class="searchform"></div>
<div class="clr"></div>
<div class="gadget">
```

Coding

Fake Profile Identification Using Artificial Neural Network

```
<h2 class="star"><span>Owner Main </span> Menu</h2>

<div class="clr"></div>

<ul class="ex_menu style3"><li class="style4">

<li><a href="OMainPage.jsp">Home</a></li>

<li><a href="index.html">Logout</a></li>

<li></li>

</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="fbg"></div>

<div class="footer">

</html>
```

V R C

Fake Profile Identification Using Artificial NeuralNetwork

-  38
-  BioTech
-  Institut Seni Indonesia Surakarta

Document Details

Submission ID**trn:oid:::1:3188887465****Submission Date****Mar 20, 2025, 5:52 PM GMT+7****Download Date****Mar 20, 2025, 6:15 PM GMT+7****File Name****1.docx****File Size****3.1 MB****50 Pages****4,602 Words****29,508 Characters**

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- ▶ Bibliography
 - ▶ Quoted Text
-

Match Groups

-  **3** Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks
 -  **0** Missing Quotations 0%
Matches that are still very similar to source material
 -  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
 -  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks
-

Top Sources

- 5%  Internet sources
- 5%  Publications
- 5%  Submitted works (Student Papers)

Match Groups

-  7 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks
-  0 Missing Quotations 0%
Matches that are still very similar to source material
-  0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 5%  Internet sources
- 5%  Publications
- 5%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

Rank	Type	Source	Percentage
1	Internet	www.irejournals.com	5%
2	Internet	www.researchgate.net	8%
3	Internet	www.abulaish.com	6%
4	Internet	ijariie.com	6%
5	Internet	sureshotpost.com	5%
6	Internet	www.journal-iiie-india.com	4%
7	Internet	journals.pen2print.org	4%
8	Internet	ieeexplore.ieee.org	3%
9	Internet	www.cs.rpi.edu	3%
10	Publication	Gergo Hajdu, Yaclaudes Minoso, Rafael Lopez, Miguel Acosta, Abdelrahman Elleit...	2%

11 Internetnebula.wsimg.com **2%****12** Student papersUT, Dallas **2%****13** Internetijrpr.com **2%****14** Internetwww.ijraset.com **1%****15** Publication

Shakir Khan, Ashraf Kamal, Mohd Fazil, Mohammed Ali Alshara et al. "HCovBi-Cap... <1%

16 Internet

cupdf.com <1%

17 Internet

2021.jnic.es <1%

18 Internet

kar.kent.ac.uk <1%

19 Internet

www.hindawi.com <1%

20 Internet

www.jetir.org <1%

21 Internet

ijsrset.com <1%

22 Internet

ir.mallareddyecw.com <1%