User Authentication Using Keystroke Dynamics

Main Project Report

Submitted by

Harshika T

Reg No : FIT20MCA-2057

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

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)
JULY 2022

DECLARATION

I, **Harshika T** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date :

Place: Angamaly

Harshika T

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
(An ISO 9001:2015 Certified, NAAC ('A' Grade) Accredited Institution with NBA Accredited Programmes
(Approved by AICTE – Affiliated to APJ Abdul Kalam Technological University, Kerala)
Owned and Managed by Federal Bank Officers' Association Educational Society
HORMIS NAGAR, MOOKKANNOOR P.O., ANGAMALY - 683 577, ERNAKULAM DT., KERALA, S. INDIA.
Tel: (O) 0484-2725272 Fax: 0484 – 2725250 E-mail: mail@fisat.ac.in Website: www.fisat.ac.in

11TH July 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. HARSHIKA. T (Reg. No. FIT20MCA-2057) has successfully completed his/her Main Project with the title "User Authentication Using Keystroke Dynamics", in the Department of Computer Applications, FISAT, during the period from 30th March 2022 to 11th July 2022.

Dr DEEPA MARY MATHEWS
HEAD OF THE DEPARTMENT

FEDERAL INSTITUTE OF SCIENCE & TECHNOLOGY (FISAT) ANGAMALY
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

Focus on Excellence

CERTIFICATE

This is to certify that the project report titled **'User Authentication Using Keystroke Dynamics'** submitted by **Harshika T [Reg No: FIT20MCA-2057]** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

Project Guide

Dr.Deepa Mary Mathews

FEDERAL INSTITUTE OF SCIENCE & TECHNOLOGY (FISAT) ANGAMALY
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Head of the Department

Dr.Deepa Mary Mathews

Submitted for the viva-voice held on ..... at .....

Examiner :

ABSTRACT

Keystroke dynamics is the study of whether people can be distinguished by their typing rhythms, much like handwriting is used to identify the author of a written text. Many authentication systems are being used. In today's age, the use of computers has increased exponentially with a variety of online applications for public usage such as e-commerce, blogs, bank services, etc. All of these existing online platforms require an authentication process to ensure the individuality of a genuine user. Authentication systems like passwords, patterns, biometrics, etc. are used. But this system has some drawbacks. Password can be easy to guess if it is small or the user has written somewhere. The pattern can be guessed using shoulder surfing. A Biometric system is more secure than a password and pattern. Keystroke dynamics is a biometric technique to recognize and an analysis of his/her typing patterns. The unique typing patterns of users can be used as a signature to identify genuine users from imposters. The concept is known for its behavioral biometrics which makes use of typing patterns to analyze and gain insights into a user accessing the system.

In this system, we can manage the access of unknown users. Whenever a user tries to log in, this system will check whether the registered and login users are the same. Thus it is the same it will give access else deny it. Keystroke biometrics is Economical to implement and can be easily integrated into the existing computer security systems. This unique typing rhythm tends to become a natural choice for the security of computers and is commonly known as Keystroke Dynamics. The algorithm to be used is the Decision Tree algorithm. The keystroke data collected include the typing speed, time, and keypress events.

Contents

Chapter 1

INTRODUCTION

Nowadays, there is a major threat of misusing personal as well as official data. Data that is easily accessible from the storage device by an unauthenticated user is a major concern and needs to be avoided. An increase in the number of software and devices for hacking and cracking causes gains in unauthorized access which results in the exploitation of important data. Methods like user ID and password which are mostly used as security is now not reliable and secure due to the rapid increase in hackers. Also, this method no longer provides consistent safety measures because passwords are prone to shoulder surfing and passwords can also be hacked.

Most computer systems rely on usernames and passwords as a mechanism for access control and authentication of authorized users. These credential sets offer marginal protection to a broad scope of applications with differing levels of sensitivity. Traditional physiological biometric systems such as fingerprint, face, and iris recognition are not readily deployable in remote authentication schemes. The solution or the problem is keystroke dynamics. Keystroke dynamics provide the ability to combine the ease of use of a username or password scheme with the increased trustworthiness associated with biometrics. It verifies the individual person by its keystroke typing pattern.

To gain secure and efficient access either the user must change his password

frequently or the user should use a strong password. Users do not accept these conditions because they feel that they are restricted or strict and also difficult to be applied. The solution for above said problems is keystroke dynamics. Keystroke Dynamics is a behavioral biometric approach to improving the quality of computer access rights. It verifies the individual person by its keystroke typing pattern. It is based on the belief that each person has a unique typing pattern.

Keystroke dynamics is the study of whether people can be distinguished by their typing rhythms, much like handwriting is used to identify the author of a written text. Possible applications include acting as an electronic fingerprint, or in an access-control mechanism. A digital fingerprint would tie a person to a computer-based crime in the same manner that a physical fingerprint ties a person to the scene of a physical crime. Access control could incorporate keystroke dynamics both by requiring a legitimate user to type a password with the correct rhythm and by continually authenticating that user while they type on the keyboard. Keystroke dynamics is a behavioral biometric, this means that the biometric factor is 'something you do.

The unique typing patterns of users can be used as a signature to identify genuine users from imposters. Keystroke signature fuses the simplicity of passwords with increased reliability from biometrics. Keystroke biometrics is economical to implement and can be easily integrated into the existing computer security systems. The machine learning model is trained with the typing patterns of the subjects. Then it is provided with test data with patterns from the subject as well as from imposters posing as the subject. The model demonstrates the ability to discern genuine users from imposters based on the test pattern similar to the trained model.

Chapter 2

PROOF OF CONCEPT

Our dependence on computers and digital platforms has been increased to simplify our lives.The use of such automated information systems together has resulted in improved performance of the available services.With such efficient utilization the advances in technology have generated a collective interest in global access to such online platforms. However, at the same time there is a rise in the threats regarding to the security of computers.Hence there is a need to generate foolproof measures to prevent such unauthorized access is being worked upon. One such preventive method to give access to individuals by detecting their unique and behavioral pattern is an individual's typing rhythm. This unique typing rhythm tends to become a natural choice for security of computers and is commonly known as Keystroke Dynamics. It is observed that when a person types; the placement of his fingers, applied pressure on the keys and the regularly typed strings appears to be consistent for a specific individual. Hence this concept is used to differentiate between an intruder and a legit user as they will be typing on the keyboard anyway. Therefore, making such kind of typing rhythms easily accessible to track computer activities.

The physiological biometric samples like retina scans, fingerprints, iris scans, hand geometry, facial recognition, etc. However, implementing a system that uses such biometrics specifically requires added hardware to deal with the physiological samples. An alternative to this is using behavioral biometrics to identify a user. These biometrics do not need separate hardware for their implementation and largely depend on the habits and behavior of an individual.

Keystroke authentication is a biometric system that analyzes a user's typing rhythm as an identifier.With machine learning, keystroke authentication can predict the rhythm as the respective user or as an imposter.Keystroke dynamics features are usually extracted using the timing information of the key down/hold/up events.The hold time or dwell time of individual keys, and the latency between two keys, i.e., the time interval between the release of a key and the pressing of the next key are typically exploited.

As technology is advancing continuously, threat to security is increasing every day. Today every computerized system handles various users for different purposes. This handling is done using identification and authentication which serves the purpose of security and prevents misuse of any system or its underlying information. It becomes very easy to hack into system protected by a password by obtaining it unethically. Therefore, identification is also done either using physiological or by using behavioral characteristics of a user. These characteristics need to be unique in such a way that a particular feature should be specifically associated with one and only one individual, thus, making sure of his/her identity. An alternative to this is using behavioral biometrics to identify a user. They do not need separate hardware for their implementation and largely depend on the habits and behavior of an individual. Examples of behavioral biometrics are typing pattern, gait analysis, voice ID, signature analysis, mouse movements, etc. keystroke dynamics as a unique way to identify a user. Since two users don't usually type in the same style we can use this as a simple way to differentiate two users. [1]

Today, secondary biometric authentication systems can be used to mitigate these vulnerabilities. Biometrics are unique traits that distinguish every individual from each other. One biometric technique is keystroke dynamics which is an authentication method based on a user's typing rhythm on a keyboard. These rhythm patterns are based on digraphs or the timing between two successive key presses. The goal is to record the keystroke patterns of each user to determine if the users can be authenticated accurately by this method using machine learning. Here Two text sizes where also compared to compare each algorithm's prediction rate based on input size.[2]

computer systems rely on usernames and passwords as a mechanism for authentication and access control. These credential sets offer weak protection to a broad scope of applications with differing levels of sensitivity. Keystroke dynamics provide the ability to combine the ease of use of username / password schemes with the increased trustworthiness associated with biometrics.[3]

2.1 FINDINGS AND PROPOSALS

The proposed model in the paper 'Biometric Authentication Using Keystroke Dynamics' records the unique typing style and pattern of a user at the time when they register into the system by feature extraction and train the knowledge base with these feature data. Then, identify and authenticate the users at a large stage when they try to log into the system or access the system in any way. The model provides an advance- level security to critical systems where security is one of the major concerns. The model does not require any extra hardware like other physiological biometric systems do and hence, the overall cost is considerably negligible.

In the paper 'A Comparison of Machine Learning Algorithms in Keystroke Dynamics', first of all, there is a need to build a python program that collects

users' timings, collect these timings from the volunteering subjects, teach and create training models for each machine learning algorithm, and evaluate the results from each experiment. Start with a preliminary testing phase to ensure every component of the project was working correctly and to create a basis of results. After confirming the base results, experiment with the volunteering colleagues and college students. With the results, it can be concluded that the Random Forest algorithm is most effective at classifying subjects to their respective typing patterns and that a strong password is weaker in comparison to a sentence or phrase that is relevant to the user. Using a relevant phrase, such as the name or an answer to a personal question, would be more practical in a real-world situation because the user would more likely have a clearer pattern with something that they type normally.

In the paper'Evaluating Reliability of Credential Hardening through Keystroke Dynamics', a web-based application was developed to acquire a large database to be used to establish the viability of keystroke dynamics with username and password credential sets as a possible method of hardening a traditional authentication scheme. The study was novel in that shift key behavior was included in the feature matching process. There appeared to be a significant performance difference between the eight letter lowercase English passwords and twelve character randomly generated password which required shift-key behavior. The long password sequences performed better across numerous performance tests and comparisons providing support for the hypothesis that shift key behavior plays a significant role in classification, especially for short inputs such as usernames and passwords. Our results may be considered unique in that the scope of the data tested currently outnumbers what has been tested in previous studies. Performance results would most likely not be sufficient for a high security environment. That having been said, it seems our approach offers adequate improvement when taken as an unobtrusive holistic approach merging password-based authentication with a behavioral biometric.

One of the major issues or a problem the digital world face is with regards to passwords used. Although, a few organizations do exist that provide secured authentication for users when they login but on the other hand there are still most sites that skip the process of identification leading to major crimes across the globe. Hence the research work in this thesis aims to suggest better security systems that could be analysed using the typing behaviour of individuals using keystroke dynamics.

Many authentication systems are being used. Authentication systems like password, pattern, biometric etc. are used. But this system has some drawbacks. Password can be easily guessed if it is small or user has written it somewhere. Pattern can be guessed using shoulder surfing. Biometric system is more secured than password and pattern. But it is more threatening to user physically.So we are using biometric system which is least used i.e. keystroke dynamics.

Keystroke dynamics is observed to be an emerging field of interest in terms of security that is responsible to validate the authenticity of the system based on users typing rhythm. Keystroke dynamics is the study of whether people can be distinguished by their typing rhythms, much like handwriting is used to identify the author of a written text. Possible applications include acting as an electronic fingerprint, or in an access-control mechanism. A digital fingerprint would tie a person to a computer-based crime in the same manner that a physical fingerprint ties a person to the scene of a physical crime. Access control could incorporate keystroke dynamics both by requiring a legitimate user to type a password with the correct rhythm, and by continually authenticating that user while they type on the keyboard. Keystroke dynamics is a behavioral biometric, this means that the biometric factor is 'something you do'.

And coming to the proposed model,Keystroke Dynamics is a measure which authenticates the access to PCs by recognizing certain unique patterns in a user's typing rhythm and pattern while logging in. We feel that the use of keystroke dynamics is a natural and important choice for cybersecurity.It records the unique typing style and pattern of a user at the time when they register into

the system by feature extraction and trains the knowledge base with these feature data. Then, identify and authenticate the users at a large stage when they try to log into the system or access the system in any way.And provided the picture upload for more security.And in a time period it helps to update the values we uploaded. The model provides advanced-level security to critical systems where security is one of the major concerns. The overall cost is considerably negligible.

2.2 OBJECTIVES

The objective is to reduce the unauthenticated intervention of users and to give more security

1.To ensure security to the users for the passwords by behavioral biometric.

Keystroke dynamics is a behavioural biometric.The unique typing patterns of users can be used as a signature to identify genuine users from imposters.

2.To distinguish people by their typing rhythm

This confirm the identity of an individual based on the manner and the rhythm of typing on a keyboard.

3.To ensure correct identification and verification of respective user.

This technique is emerging as an effective way to strengthen user authentication. Keystroke dynamics is a detailed description of the timing of key-down and key-up events when users enter usernames, passwords, or any other string of characters.

Chapter 3

IMPLEMENTATION

Keystroke Dynamics is the study of the typing patterns of people to distinguish them from one another, based on these patterns. Every user has a unique typing pattern that is very tough to replicate.The proposed system provides a model to record the unique typing speed of the user and extract the feature then use it for further authentication time. Also, in the proposed system introduced the method to upload the picture of the user. As the model identify the user it also goes through the process of detecting the person, then it will give access or deny it.The keystroke value obtained at the time of login is checked with those collected of the time of registration.If it is a match the user is logged in.The value in the database are obtained at the registration phase.During the registration,the user is given by a phrase to collect the values.This values are used to building the model.When each user register into system,the model is built using value in database.Keystroke dynamics are an effective behavioral biometric, which captures the habitual patterns or rhythms an individual exhibits while typing on a keyboard input device. These rhythms and patterns of tapping are idiosyncratic, in the same way as hand writings or signatures, due to their similar governing neuro physiological mechanisms.With this implementation we were able to create a user verification software that relied on both the face and keystroke recognition. The prime step of developing this model is to design a machine learning model for user authentication.The

proposed model uses Decision tree for classification.

This model is used to authenticate users based on his/her typing patterns. Keystroke dynamics is a behavioral biometric, this means that the biometric factor is 'something you do'. It is the study of whether people can be distinguished by their typing rhythms. The project is developed using the decision tree Classifier Algorithm. The user provides basic details and types in the given text during registration. The keystroke values are successfully collected and stored in the database. The model is trained using this data. Also, the images are uploaded at the time of registration After successful registration, the user can login to the system using correct login credentials and keystroke values. And it also recognise the person. The user is logged in after providing correct details and not logged in if there is any mismatch in the credentials.In the model, it is designed the user interface which has 2 forms registration and login. Then it is connected the user interface to the model. Also provided the feature of updating the keystroke value in a particular time period. Thus it helps for the smooth working. The model is very effective.

3.1 ARCHITECTURE

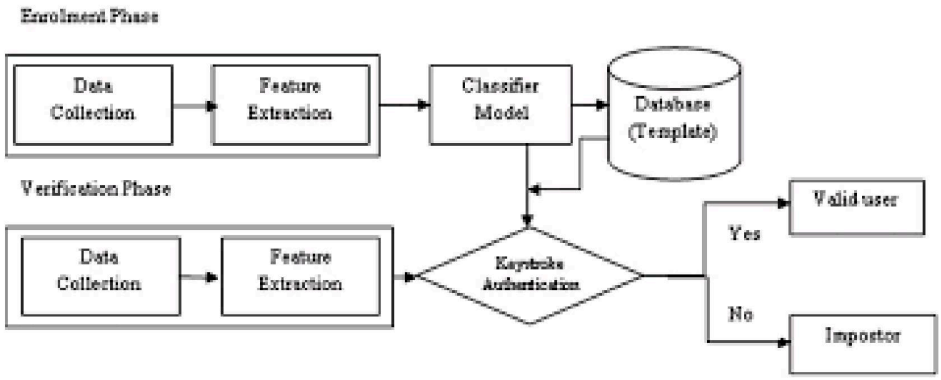


Figure 3.1: Block diagram of model

The project is basically using python,MySQL for the backend, interfacing with a database. The front end is built using HTML, CSS, and JavaScript. The backend of the system is written in python to allow for high scalability, even when used in systems with high throughput. The database is used, which further improves the performance of the system, allowing for easy extensibility and providing high availability.This architecture depicts the working of the system.The data extracted from the user is stored into the database.The collected data are used for the training and testing.The classifier compare the data stored with the new given data.Once both the data are checked and the accuracy matches, then it will provide access otherwise it will deny the access.

3.2 DATASET

The dataset for the implementation consists of keystroke dynamics based on the information accumulated from users at the time of registration; wherein each user was given a phrase to type at the time of registration, yielding an array of

keystroke values.

The keystroke values are obtained during the time of registration. These values are stored in the database. That is, at the time of registration, the user is asked to give a username and a password. In addition to this, the user is also given a specific sentence or phrase which the user is asked to type. In this way, we get the keystroke values.

The database contains two tables, user table and login table. The fields of user table are user id, login id, first name, last name, age, email and phone. The fields of login table are login id, username and features. The features are the keystroke values obtained. This includes the event of the key pressed, typing speed and time taken for typing.

The values for the database is collected at the time of registration. It includes name, age, phone number, email, username, password and a phrase from which keystroke values are collected. The figure shows values inserted into database after registration.

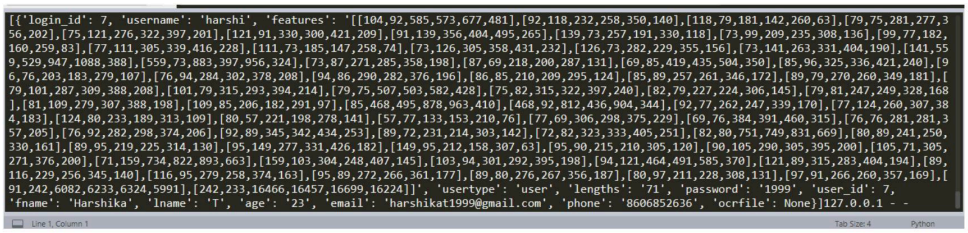


Figure 3.2: Values inserted to database

3.3 MODULES

3.3.1 STRATEGY

In the first phase of Machine Learning project realization, we mostly outline strategic goals. They assume a solution to a problem, define a scope of work, and plan the development.So here first we are collecting the data from the user.Later the data pre processing followed by model training was done.As it collect datas it check the collected data and stored data are similar.
Each of these phases can be split into several steps.

3.3.2 DATASET PREPARATION AND PREPROCESSING

Data is the most important for any machine learning project. The second stage of project implementation is complex and involves data collection, selection, preprocessing, and transformation.In this collecting the data of user like typing speed,time taken to type it.Data is collected at the registration phase.The data which is used in this system is acquired from various inputs.Also collect the image of the user.

Data collection

Data collection is the process of gathering and measuring information on targeted variables in an proposed system, which then enables one to answer relevant questions and evaluate outcomes.

Data selection

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data.The selected data includes attributes that need to be considered when building a model.

Attribute	Type	Description
Subject	Text	Subject Value
DD	Numerical	Duration of pressing a key and pressing another key
UD	Numerical	Duration of releasing a key and pressing a key
UU	Numerical	Duration of releasing a key and releasing another key
H	Numerical	Duration of pressing and releasing a key
DD.space	Numerical	Duration of pressing a key and pressing space key
UD.space	Numerical	Duration of releasing a key and pressing space key

Figure 3.3: Feature variables

Data preprocessing

Data preprocessing involves conversion of data from the given number values.Here the values are the number of keys usedtime taken etc are considered. The proposed method deals with authentication method using classifier.Decision tree is used here to classify the data collected by the user.And check wheather it is able to give access or not.

3.3.3 MODELLING

During this stage,it trains numerous models to define which one of them provides the most accurate predictions.The values in the database are obtained at the registration phase. During registration, typing the given phrase collects keystroke values. This keystroke value is used for building the model. When each user registers into the system, the model is built using the values available in the database. Decision tree algorithm is used.

3.3.4 MODEL DEPLOYMENT

A pretrained model was deployed to train this system to give appropriate predictions.During the implementation we have to save the model. Then we have to design the user interface which has 2 forms for registration and login. Then we have to connect the user interface to the model.Once we chosen a reliable model and specified its performance requirements, we deploy the model to production.The User Authentication Using Keystroke Dynamics system is implemented successfully. The system logs in the user if the keystroke values are matched. Else not. It is very effective and user friendly.

3.4 ALGORITHM

The proposed model identify keystroke dynamics. The machine learning approach is used for keystroke identification. Machine learning (ML) is defined as the study of computer programs that leverage algorithms and statistical models to learn through inference and patterns without being explicitly programmed. ML algorithms learn over experience and improve automatically. It finds techniques, trains models, and uses the learned approach to determine the output automatically. Machine learning systems can also adjust themselves to a changing environment.

Various standard machine learning algorithms are used to keystroke identification. Among the selected algorithms, the decision tree provided the best accuracy. Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the

above figure. This process is then repeated for the subtree rooted at the new node.

Decision trees have three main parts: a root node, leaf nodes and branches. The root node is the starting point of the tree, and both root and leaf nodes contain questions or criteria to be answered. Branches are arrows connecting nodes, showing the flow from question to answer. Each node typically has two or more nodes extending from it.

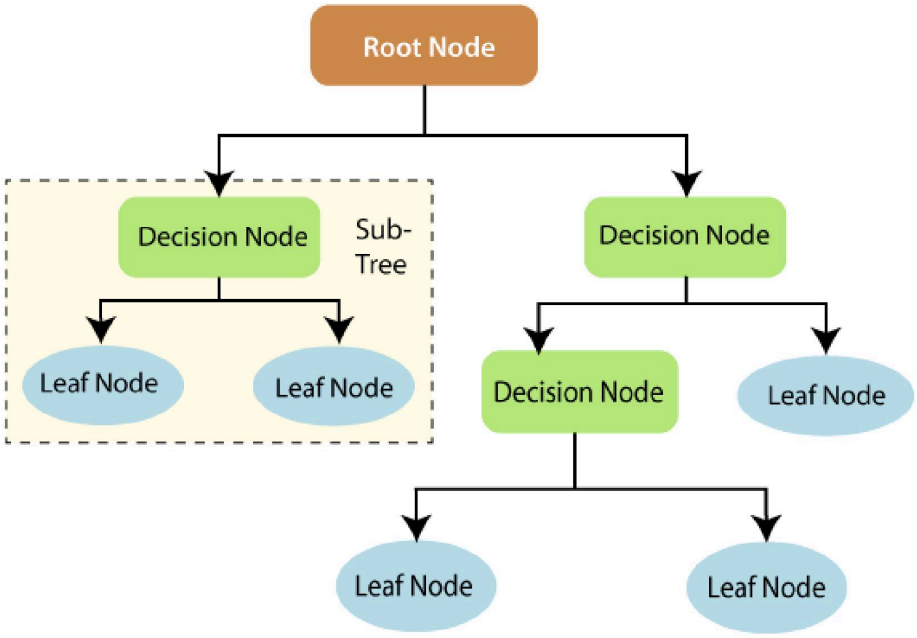


Figure 3.4: Decision tree

3.4.1 WORKING

The Decision Tree working can be explained on the basis of the below algorithm:

• Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
• Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
• Step-3: Divide the S into subsets that contains possible values for the best attributes.
• Step-4: Generate the decision tree node, which contains the best attribute.
• Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3.

Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

3.4.2 FLOWCHART

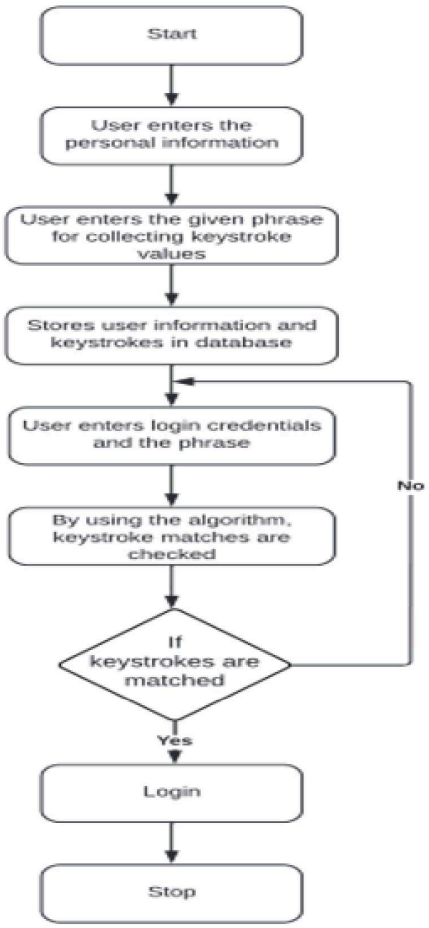


Figure 3.5: System framework

3.4.3 FEATURES AND CHARACTERISTICS OF THE ALGORITHM

• Decision trees are able to generate understandable rules.
• Decision trees perform classification without requiring much computation.
• Decision trees are able to handle both continuous and categorical variables.
• Decision trees provide a clear indication of which fields are most important for prediction or classification.

In this project, decision tree is used to train the data. Training is done using the keystroke values of different users obtained at the time of registration. Suppose there are x users in the system. The features of these users obtained at the time of registration is given to the decision tree and the model is built. When we give an input value, it is checked with the trained value.

3.5 PROJECT PIPELINE

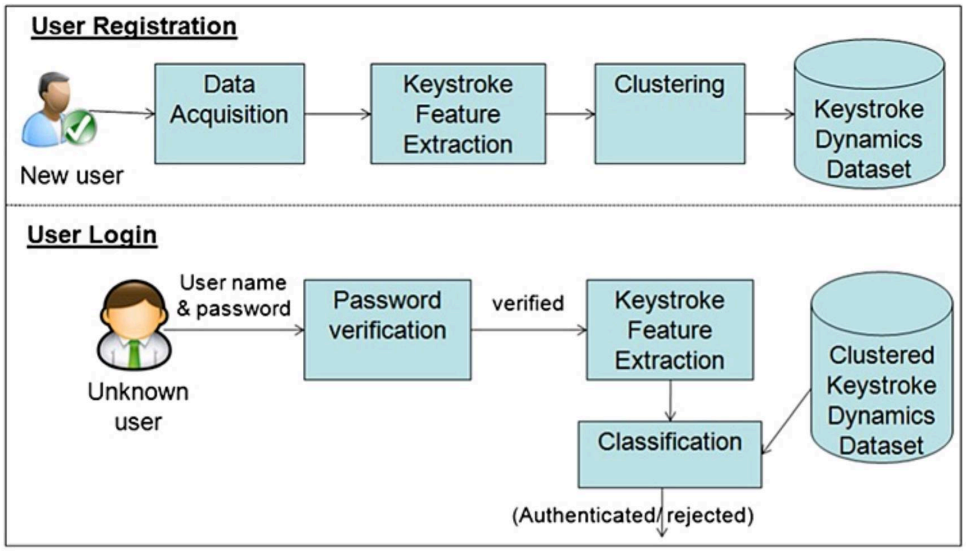


Figure 3.6: working of model

The first step in the project is to collect the keystroke features. These details are collected in the registration phase. The user is asked to type in the basic details, username, password and a given phrase. Typing this phrase collects the keystroke values. Decision tree algorithm is used. The model is built using the keystroke values. All these values and information are stored in the database. At the login phase, the user is asked to type in the username, password and the phrase which was asked during the time of registration. Once the details are typed in and login button is clicked, the password and the keystroke values are validated. Only if the keystroke values at the time of registration matches with that given at login, the user gets logged in. Else the user cannot login.

Chapter 4

RESULT ANALYSIS

The result of the proposed project 'User authentication using keystroke dynamics' lies in developing a web page that can be used for login into the page for valid user.It will not allow the entry of non user.Here User Authentication Using Keystroke Dynamics is used to authenticate users based on his/her typing patterns. Keystroke dynamics is a behavioural biometric, this means that the biometric factor is 'something you do'. It is the study of whether people can be distinguished by their typing rhythms, much like handwriting is used to identify the author of a written text. The project is developed using Decision Tree Classifier Algorithm. The user provides basic details and types in the given text during registration. The keystroke values are successfully collected and stored in the database. Model is trained using this data. After successful registration, the user can login to the system using correct login credentials and keystroke values. The user is logged in after providing correct details and not logged in if there is any mismatch in the credentials.The data collected are stored and checked when the time of login.When comparing the features extracted at time of registration and login are compared if both are having accuracy more 75,it will be given access else denied.The web application uses php language which is user friendly and efficient along with MySQL. the proposed system requires very less time factors The cost needed for development of the application is optimum.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In this project, we address the practical importance of using keystroke dynamics as a biometric for authenticating. Keystroke dynamics is the process of analysing the way users type by monitoring keyboard inputs and authenticating them based on habitual patterns in their typing rhythm. We review the current state of keystroke dynamics and present some techniques to validate the user depending on his typing pattern Verification of user's identity is the most important goal of authentication. Passwords are adopted as the most widely used security mechanism. It is obvious that, passwords (something the user knows) are not a very secure mechanism for authenticating users because of its limitation, as well as, it is difficult to confirm that the demand is from the rightful owner. Strong method of authentication should cover one or several factors of identification to improve security. Keystroke dynamics is the process of analyzing and measuring the style in which any user types the elements of his/her secret information by the keyboard. This process is either done at either during a complete session, which is called continuous keystroke dynamics or during login time or after a pre-

determined period of time which is called static keystroke dynamics. This work is focused on identifying a person based on his/her typing behaviour. I collected the keystroke data from different users typing a given phrase during the registration phase and applied decision tree algorithm to classify the data. In this work, keystroke dynamics is used for user authentication. This method is more accurate than the other system when it comes to person authentication. This system is still secure even when password is compromised. And it will analyse the person by their keystroke and only authenticated user will able to access the account.

5.2 FUTURE SCOPE

The future of biometric technologies is promising. Biometric devices and applications continue to grow worldwide. Several factors will push the growth of biometric technologies. A major inhibitor of the growth of biometrics has been the cost to implement them. Moreover, increased accuracy rates will play a big part in the acceptance of biometric technologies. Keyboard Dynamics, being one of the cheapest forms of biometric, has great scope.In the future there should be a study on whether mistakes can be used as a supplementary tool to improve accuracy, instead of being included directly in the model, especially for users that have a high mistake rate and the correctly typed data is scarce. In this project, only the classification of different people is done. However, for the method to work as an identification, consistency is important. There should be another study on the same person whether they can produce the similar keystrokes throughout a long-term period. In the future, investigation can be done on how well the performance for one handed typing and the scalability can be improved.

The keystroke and mouse features can also be combined which may provide a high level of security. The future scope also includes the identification of other parameters and the use of other algorithms to improvise the level of security and implementation of keystroke on touch screen system and use of pressure keyboard.

Chapter 6

APPENDIX

6.1 CODING

6.1.1 core.py

```
Code for coverting the data and finding accuracy,
Preprocessing ,training etc.

from database import *
import demjson
import numpy as np
from model_manager import Model
import pickle
import math
import cv2
import uuid
import face_recognition
import argparse
```

```
from imutils import paths
import os
import requests
import io
import json

def get_max_login_id():
        q = "select max(login_id) as  max from login"
        res = select(q)
        print(res[0]['max'])
        if res:
                return res[0]['max']
        else:
                return 0

def create_matrix():
        max_id = get_max_login_id()
        matrix = []
        for i in range(0,max_id+1):
                row = []
                for j in range(0,max_id+1):
                        m = Model(i,j)
                        row.append(m)
                matrix.append(row)
        for i in range(0,max_id+1):
                for j in range(0,max_id+1):
                        matrix[j][i] = matrix[i][j]
        return matrix

def pre_process_features(features):
```

```
        # print(features)
        temp = []
        for f in features:

                if len(f) == 6 and None not in f:
                        temp.append(f)

        if temp:
                temp = temp / np.max(temp)
                temp = np.asarray(temp)
        return np.asarray(features)
def train_matrix(matrix,user1,user2):
        user_1_id = user1['login_id']
        user_2_id = user2['login_id']
        # print((user1['features']))
        # print((user2['features']))

user_1_features = pre_process_features(demjson.decode(user1
['features']))
user_2_features = pre_process_features(demjson.decode(user2
['features']))

user_1_op = np.asarray([user_1_id] * user_1_features.shape[0])
user_2_op = np.asarray([user_2_id] * user_2_features.shape[0])
# X_train = np.append(user_1_features,user_2_features,axis=0)
# Y_train = np.concatenate((user_1_op,user_2_op),axis=0)
matrix[user_1_id][user_2_id].train(user_1_features,
user_2_features,user_1_op,user_2_op)
matrix[user_2_id][user_1_id].train(user_1_features,
user_2_features,user_1_op,user_2_op)
```

```
def train():
        matrix = create_matrix()
        q = "select * from login"
        res = select(q)
        for i in range((len(res))):
                for j in range((len(res))):
                        user1 = res[i]
                        user2 = res[j]
                        train_matrix(matrix,user1,user2)
        file = open("model.pickle","wb")
        pickle.dump(matrix,file)
        file.close()
def predict(matrix,id1,id2,features):
        # print(features)
        if id1 > -1 and id2 > -1:
                res = matrix[id2][id1].predict(features)
                print(matrix[id2][id1])
        elif id1 > -1:
                res = id1
        elif id2 > -1:
                res = id2
        else:
                res = -1
        # print(res)
        # prob = matrix[id2][id1].predict_proba(features)
        # print(prob)
        return res
def predict_from_array(matrix,array,features):
        print(array)
        new_layer = []
```

```
        if len(array) > 1:
                for i in range((len(array) - 1)):
                        user1 = array[i]
                        user2 = array[i+1]
new_layer.append(predict(matrix,user1,user2,features))
                if len(new_layer) == 1:
                        return new_layer[0]
        else:
                user1 = array[0]
                user2 = array[0]
                # print(features)
return predict(matrix,user1,user2,features)
return predict_from_array(matrix,new_layer,features)


def get_login_id(features):
file = open("model.pickle","rb")
matrix = pickle.load(file)
file.close()
features = pre_process_features(demjson.decode(features))
q = "select * from login"
res = select(q)
layer = []
for row in res:
layer.append(row['login_id'])
id = predict_from_array(matrix,layer,features)
        return id

q="select * from login"
res=select(q)
```

```
 cnt=0
 for row in res:
 ss=get_login_id(row['features'])print("ssss",ss)
 print("lid",row['login_id'])
 if int(row['login_id'])
 ==int(ss) or int(row['login_id'])+1==
 int(ss) or
 int(-1)==int(ss):
 cnt=cnt+1

         print("lll",len(res))
         l=len(res)

         s=cnt/l
         ss=s*100
         print("Accuracy is "+str(ss))
camera for image testing
def val(vals):
         size = 4
classifier = cv2.CascadeClassifier
('haarcascade_frontalface_alt.xml')
webcam = cv2.VideoCapture(0)
#Using default WebCam connected to the PC.
(rval, im) = webcam.read()
while True:
         (rval, im) = webcam.read()
im=cv2.flip(im,1,0) #Flip to act as a mirror

 detect MultiScale / faces
faces = classifier.detectMultiScale(mini)
```

```
 Draw rectangles around each face
flag=0
for f in faces:
(x, y, w, h) = [v * size for v in f]
#Scale the shapesize backup
        cv2.rectangle(im, (x,y),
        (x+w,y+h), (0,255,0), 4)
FaceFileName = "static/test.jpg"
#Saving the current image from the webcam for testing.
cv2.imwrite(FaceFileName, sub_face)
# break
val=rec_face_image(FaceFileName)
print(val)
print("user",vals)
str1=""
for ele in val:
        str1 = ele
        print(str1)
        val=str1.replace("'","")
        if int(vals) == int(val):
                flag=0
                else:
                flag=1
        print(FFf",flag)
        if flag==0:
                return "NA"
                        break
                if flag==1:
                        break
```

```
        Show the image
                cv2.imshow('Capture',    im)
                key = cv2.waitKey(10)
if Esc key is press then break out of the loop
                if key == 13: #The Esc key
                        return "failed"
                        break
         if flag == 1: #The Esc key
                        return "failed"
                break
#recognize face image
def rec_face_image(imagepath):
    print(imagepath)

data = pickle.loads(open('faces.pickles', "rb").read())

# load the input image and convert it from BGR to RGB
image = cv2.imread(imagepath)
#print(image)
h,w,ch=image.shape
print(ch)
  rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

def enf(path): imagePaths = path
#initialize the list of known encodings and known names
knownEncodings = []
knownNames = []
for  fname in os.listdir(imagePaths):
facedir=os.path.join(imagePaths,fname)
for  imagePt in os.listdir(facedir):
```

```
img=os.path.join(facedir,imagePt)
# extract the person name from the image path
print("[INFO] processing image {}/{}"
.format(fname,len(imagePt)))
print("imagepath-------",imagePaths)
print(magePath.split(os.path.sep))
 name = fname
# load the input image and convert it
from RGB (OpenCV ordering)
 to dlib ordering (RGB)
image = cv2.imread(img)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# detect the (x, y)-coordnates
of the bounding boxes
face_recognition.face_locations(rgb,model='hog')
# compute the facial embedding for the face
encodings = face_recognition.face_encodings(rgb, boxes)
# loop over the encodings
for encoding in encodings:
# add each encoding + name to our set of known names and
 encodings
knownEncodings.append(encoding)
knownNames.append(name)
dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open('faces.pickles', "wb")
f.write(pickle.dumps(data))
 f.close()
#OCR
```

```
def ocrgenerate(image):
print(image)
img = cv2.imread(image)
height, width, _ = img.shape
Cutting image
roi = img[0: height, 400: width]
roi = img
# Ocr
url_api = "https://api.ocr.space/parse/image"
_, compressedimage = cv2.imencode(".jpg", roi, [1, 90])
file_bytes = io.BytesIO(compressedimage)
result = requests.post(url_api,
files = {"screenshot.jpg": file_bytes},
data = {"apikey": "f2c63a5eb288957",
"language": "eng"})

result = result.content.decode()
result = json.loads(result)
print(result)
parsed_results = result.get("ParsedResults")[0]
text_detected = parsed_results.get("ParsedText")
print(text_detected)

cv2.imshow("roi", roi)
cv2.imshow("Img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
return text_detected
```

6.1.2 main.py

```
from flask import Flask,render_template,request,session,
redirect,url_for,flash
from database import *
import demjson
import urllib
import numpy
import sklearn
from core import *
import uuid
from capture import *

app=Flask(__name__)
app.secret_key="val"

@app.route('/',methods=['get','post'])
def home():
 enf("static/uploads/")
  val(1)

return render_template('index.html')

@app.route('/updatefeature',methods=['get','post'])
def updatefeature():

return render_template('updatefeature.html')
@app.route('/enterpassword',methods=['get','post'])
def enterpassword():
val()
data={}
```

```
data['uname']=session['username']
if "login" in request.form:
uname=request.form['username']
passs=request.form['password']
q="select * from login inner join
user using(login_id) where username='%s'
and password='%s'" %(uname,passs)
res=select(q)
if res:
session['login_id'] = res[0]['login_id']
session['user_id'] = res[0]['user_id']
return redirect(url_for('user_home'))
return render_template('enterpassword.html',data=data)
@app.route('/register',methods=['get','post'])
def register():
return render_template('register.html')
@app.route('/login',methods=['get','post'])
def login():
q="insert into "
return render_template('login.html')
@app.route('/user_home',methods=['get','post'])
def user_home():
q="delete from checkk"
delete(q)
gets=val(session['user_id'])
print("fhG"+gets)
 if gets=="failed":
return redirect(url_for('home'))
return render_template('user_home.html')
@app.route('/validatingidcard',methods=['get','post'])
```

```
def validatingidcard():
if 'submit' in request.form:
image=request.files['image']
path="static/checking/testocrfiles/fileocr.jpg"
image.save(path)
content=ocrgenerate(path)
if content=="c":
flash("Not Recognized")
else:
print("gh",content)
lcontent=session['ocrfile']
print("ghsg",lcontent)
if lcontent==content:
return redirect(url_for('user_home'))
else:
return redirect(url_for('home'))

return render_template('validatingidcard.html')
@app.route('/userfaceegister',methods=['get','post'])
def userfaceegister():
if 'submit' in request.form:
q="select max(user_id) as id from user"
res=select(q)
print(res)
val=res[0]['id']
print(val)
image1=request.files['image1']
image2=request.files['image2']
image3=request.files['image3']
 path = 'static/uploads/'
```

```
path=""
# Check whether the specified path is
# an existing file
isFile = os.path.isdir("static/uploads/"+str(val))
print(isFile)
if(isFile==False):
os.mkdir('static\\uploads\\'+str(val))
path="static/uploads/"+str(val)+
"/"+str(uuid.uuid4())+image1.filename
image1.save(path)
path="static/uploads/"+str(val)+"/"
+str(uuid.uuid4())+image2.filename
image2.save(path)
path="static/uploads/"+str(val)+"/"
+str(uuid.uuid4())+image3.filename
image3.save(path)
enf("static/uploads/")

return redirect(url_for('login'))

return render_template('userfaceegister.html')

@app.route('/login_action/',methods=['get','post'])
def login_action():
data = {}
if 'login' in request.form:
uname=request.form['username']
features=request.form['features']
feature_text=request.form['feature_text']
length=request.form['length']
```

```
check=request.form['check']
length=request.form['length']
print(length)
session['uname']=uname
print(feature_text)
print(check)
if check==feature_text:
print(features)
s="select * from login inner join `user`
using(login_id) where username='%s'
and lengths='%s'"%(uname,length)
sel=select(s)
print('kkkkkkkkkkkkkkkkkk')
print(sel)
if len(sel)>0:
print(len(sel))
bool = get_login_id(features)
print("dsf",bool)
if bool != 1:

session['login_id'] = sel[0]['login_id']
session['user_id'] = sel[0]['user_id']
session['ocrfile'] = sel[0]['ocrfile']
data['status'] = 'success'
data['data'] = sel
else:
q="insert into checkk values(null,'%s')" %(uname)
q=insert(q)
q="select count(check_id)as
cc from checkk where
```

```
username='%s'" %(uname)
res=select(q)
print(res)
if int(res[0]['cc'])>=3:
session['username']=uname
data['status']="password"
return redirect(ul_for('enterpassword'))
else:
data['status'] = 'failed'
data['reason'] = 'Time difference'
else:
q="insert into checkk values(null,'%s')" %(uname)
insert(q)
q="select count(check_id)as
cc from checkk where
username='%s'" %(uname)
res=select(q)
print(res)
if int(res[0]['cc'])>=3:
session['username']=uname
data['status']="password"
 return redirect(url_for('enterpassword'))
else:
data['status'] = 'failed'
data['reason'] = 'Username  is incorrect
or entry difference'
else:
flash("Please enter the value in same
format given above")
return demjson.encode(data)
```

```
@app.route('/register_action/'
,methods=['get','post'])
def register_action():
if 'register' in request.form:
print("haiii")
fnam=request.form['fname']
lnam=request.form['lname']
ag=request.form['age']
eml=request.form['email']
phn=request.form['phone']
user=request.form['user']
pwd=request.form['pwd']
features=request.form['features']
feature_text=request.form['feature_text']
length=request.form['length']
check=request.form['check']
cpwd=request.form['cpwd']
if pwd==cpwd:
if check==feature_text:
pass
lo="insert into login(username,features,
usertype,lengths,password)
values('%s','%s','user','%s','%s')
"%(user,features,length,pwd)
log=insert(lo)
q="insert into `user`(login_id,fname,lname,
age,email,phone)
values('%s','%s','%s','%s','%s','%s')"
%(log,fnam,lnam,ag,eml,phn)
```

```
res=insert(q)
train()
else:
flash("Please enter the value in same format given above")
else:
flash("Please enter the same password and confirm password")
return "ok"

@app.route('/update_features/',methods=['get','post'])
def update_features():
if 'register' in request.form:
print("haiii")
features=request.form['features']
feature_text=request.form['feature_text']
length=request.form['length']
check=request.form['check']
if check==feature_text:
pass
lo="update login set features='%s',
lengths='%s' where login_id='%s'"%(features,length,
session['login_id'])
update(lo)
train()
else:
flash("Please enter the value in same format given above")

return "ok"
 @app.route('/startvideo/',methods=['get','post'])
 def startvideo():
        return render_template('startvideo.html')
```

```
app.run(debug=True, port=5035)
```

6.1.3 register.html

```
 feature extraction at the registration phase

 {% include 'publicheader.html' %}
<div class="w3ls-section contact" id="contact">
<div class="container">
<div class="w3-heading-all contact-head">

$(document).ready(function(){

key_timing = []

enable = true

$("#register_via_keystroke
input[type='button']")
.click(function(){
features = []

if (key_timing.length > 1){
feature = []
j = 0
alert(key_timing.length)
for(i=0;i<key_timing.length-1;i++){
// feature.push(key_timing[i][0])
// feature.push(key_timing[i+1][0])
feature.push(key_timing[i][1]-key_timing[i][0])
```

```
feature.push(key_timing[i+1][1]-key_timing[i+1][0])
feature.push(key_timing[i+1][0]-key_timing[i][0])
feature.push(key_timing[i+1][1]-key_timing[i][1])
feature.push(key_timing[i+1][1]-key_timing[i][0])
feature.push(key_timing[i+1][0]-key_timing[i][1])
j++;
if(j==1){

console.log(feature)
features.push(feature)

feature = []
j = 0
}
}

}
obj = $('#register_via_keystroke')
.serializeArray()
obj.push({name:$(this).
attr('name'),value:$(this).val()})
obj.push({name:'features',
value:JSON.stringify(features)})
obj.push({name:'length',
value:JSON.stringify(key_timing.length)})
$.post('/register_action/',obj,function(data){
window.location = '/userfaceegister'
})
key_timing = []
features = []
```

```
})

key_info = []

$("#register_via_keystroke textarea[name='feature_text']").
keydown(function(e){

if (key_info.length == 0){
key_info.push(Date.now())
}
});
$("#register_via_keystroke
textarea[name='feature_text']")
.keyup(function(e){
if (key_info.length == 1){
key_info.push(Date.now())
key_timing.push(key_info)
key_info = []
}
});
</script>
```

6.2 SCREENSHOTS

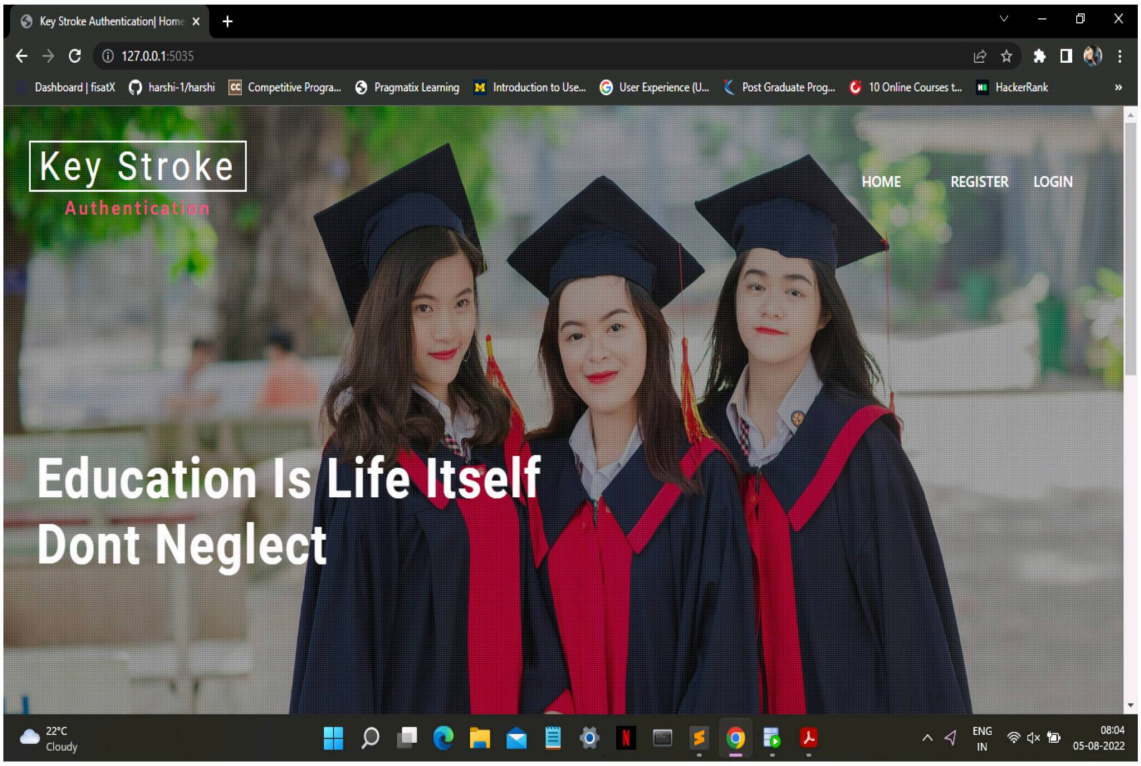
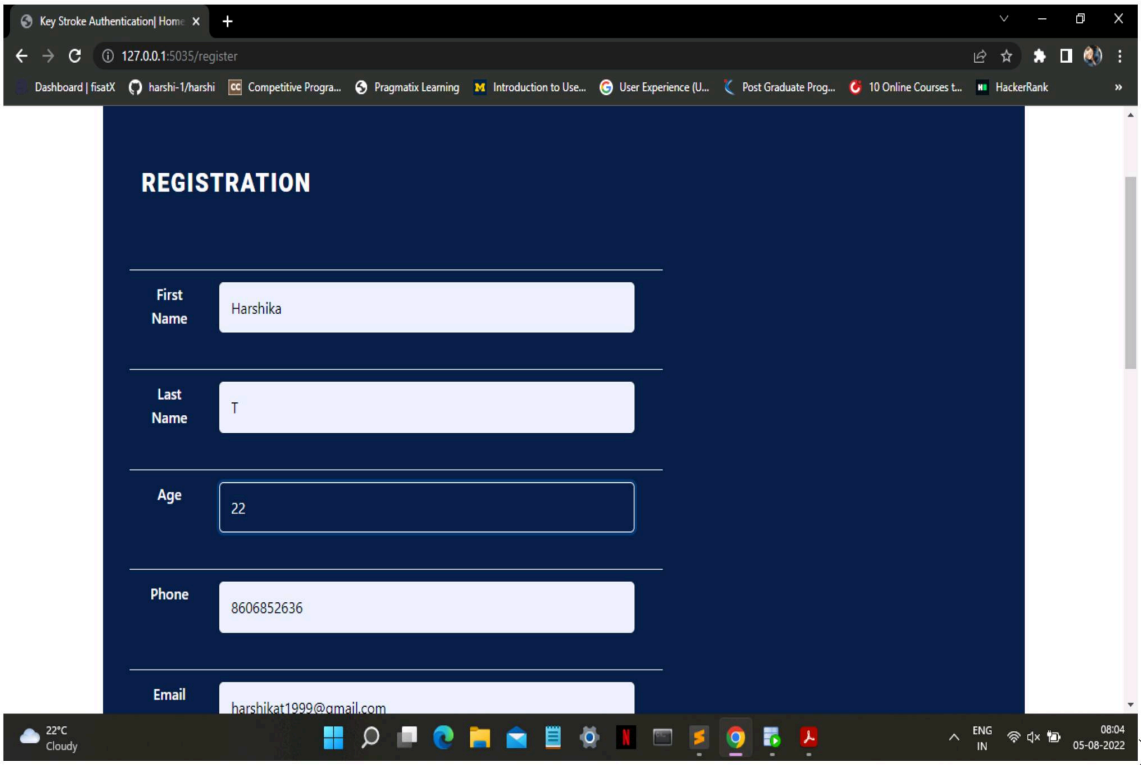


Figure 6.1: home page



).

Figure 6.2: Registration page 1

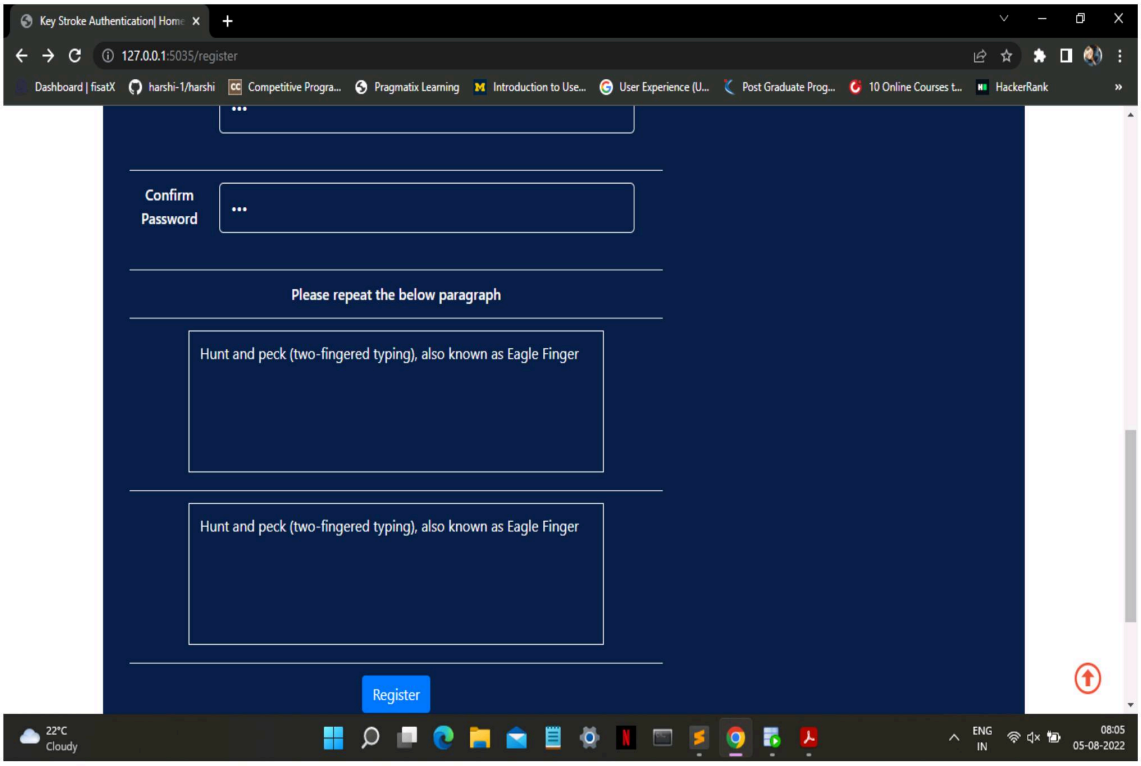


Figure 6.3: Registration page 2

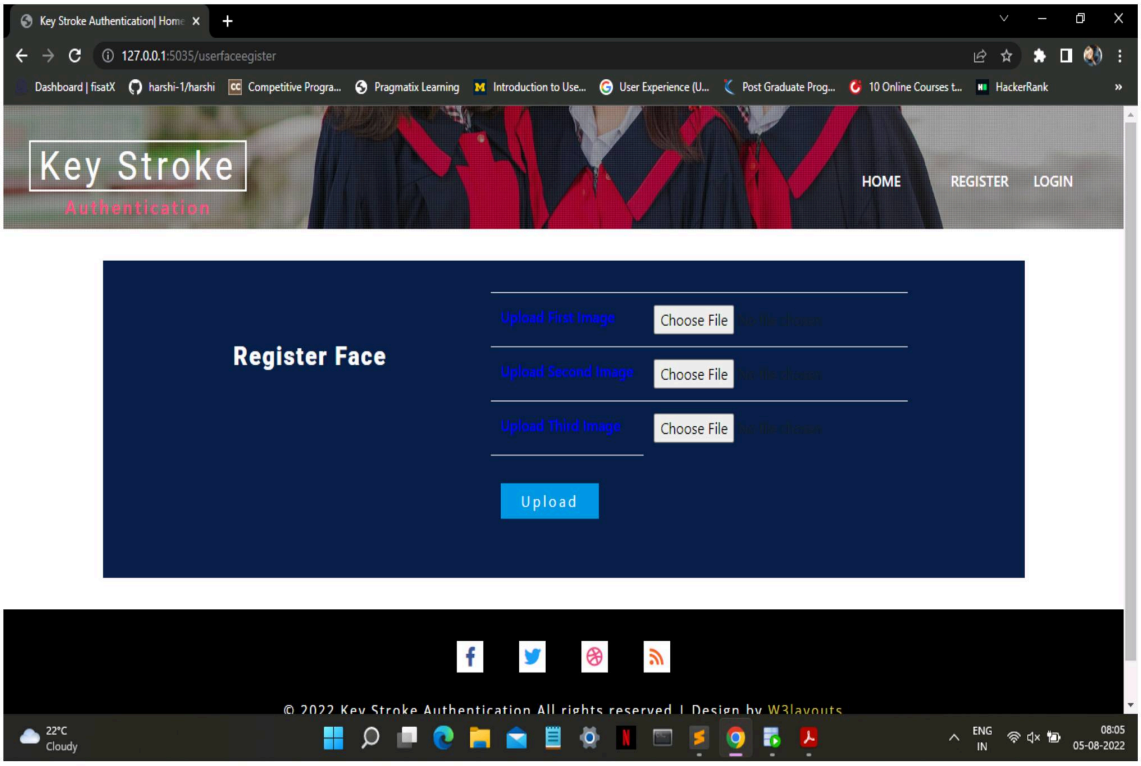


Figure 6.4: image upload page

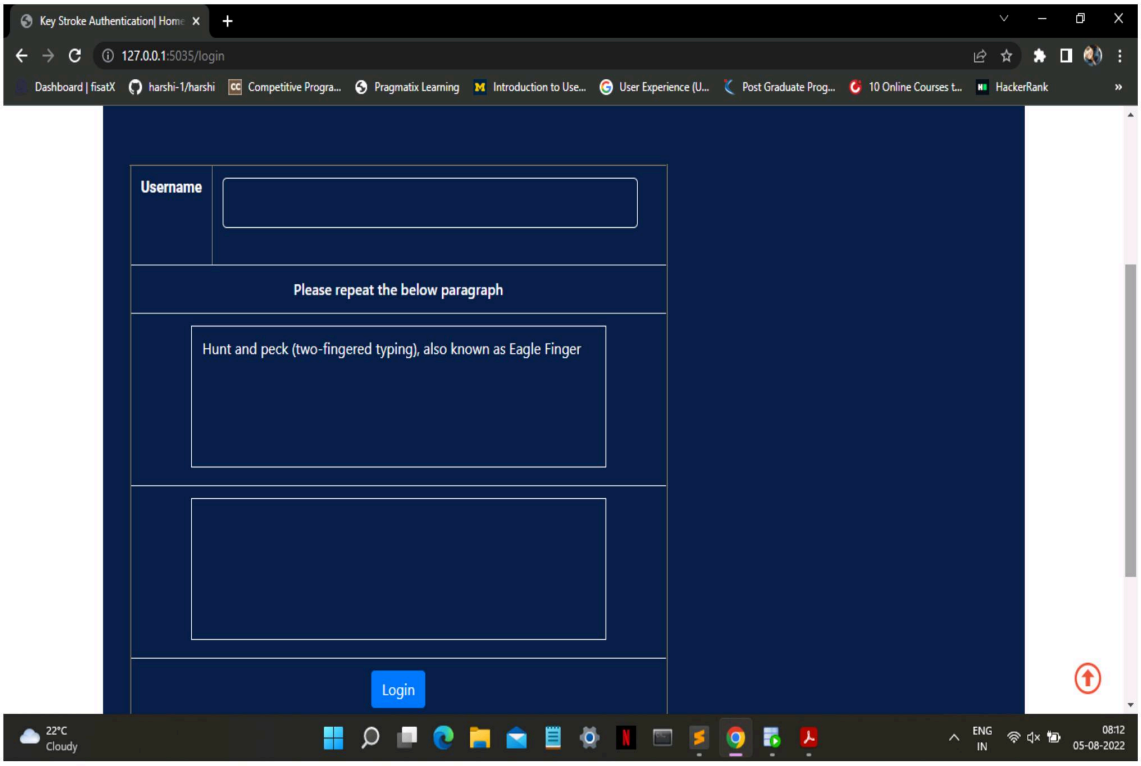


Figure 6.5: Login page

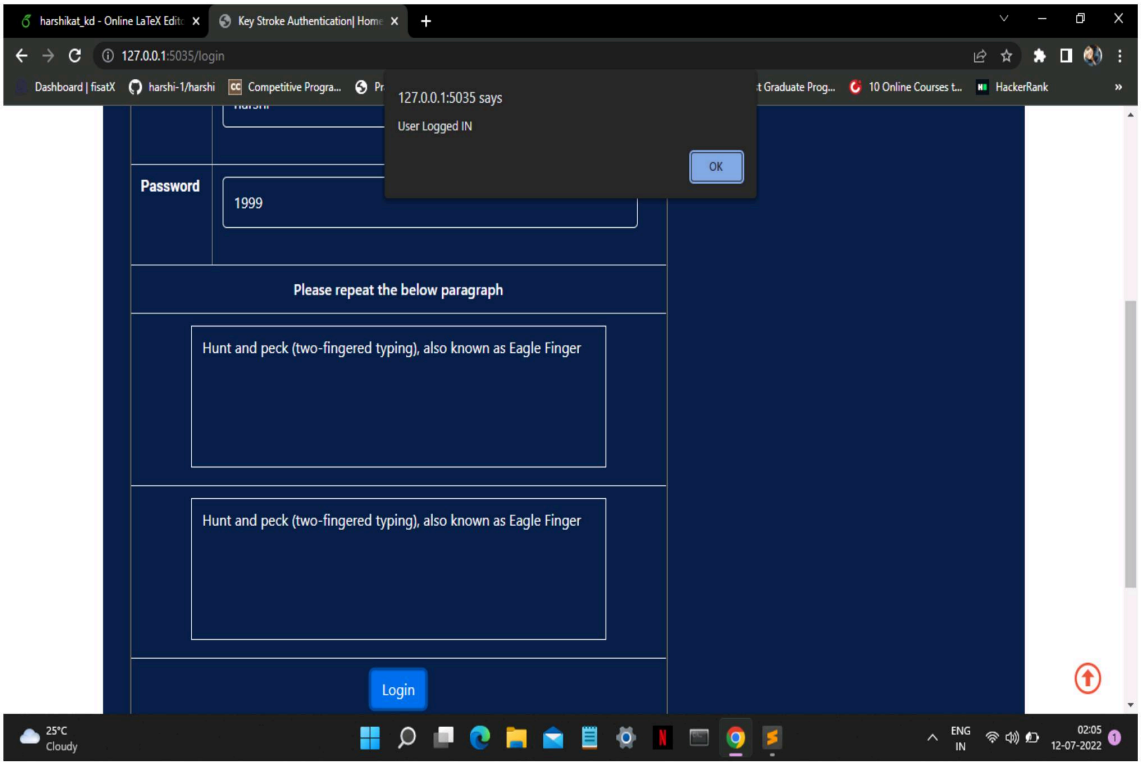


Figure 6.6: incorrect inputs

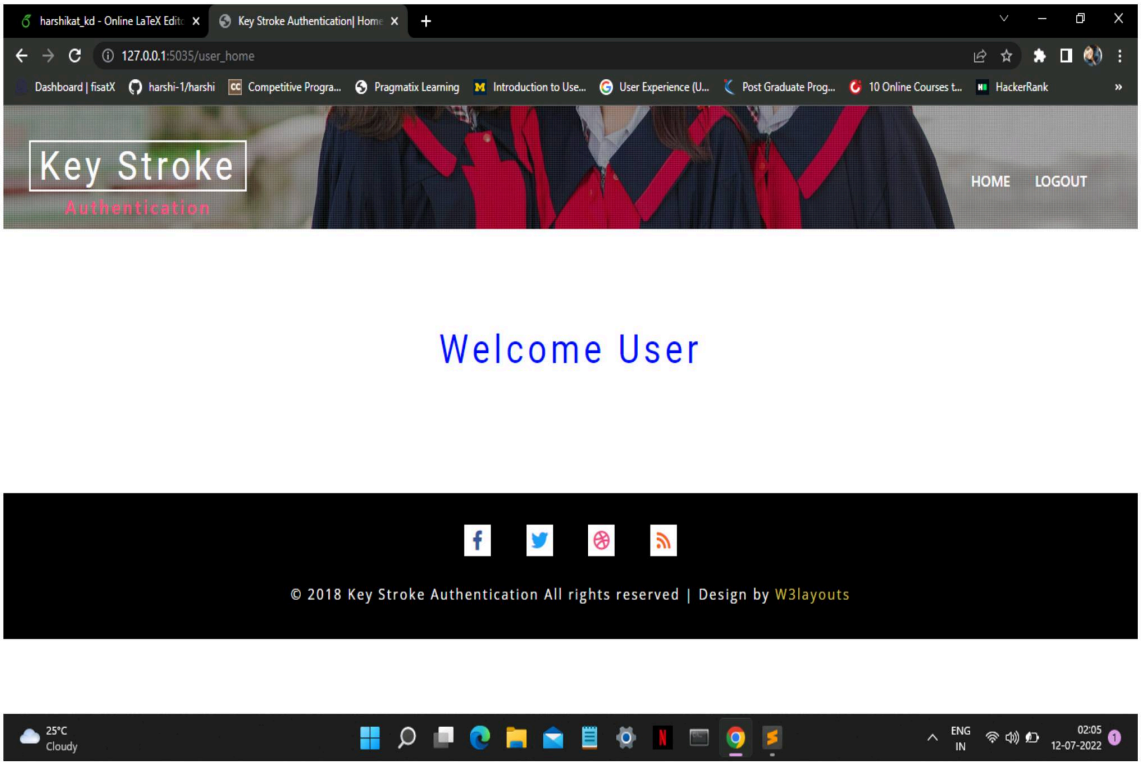


Figure 6.7: user home

Chapter 7

REFERENCES

1 .Chandralekha Jadhav, Siddhi Kulkarni, Sagar Shelar, Kaustubh Shinde, Nagaraj V. Dharwadkar. (2019). Biometric Authentication Using Keystroke Dynamics. International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) 870 (I-SMAC 2019)

2 .Elliot, Kwesi; Graham, Jonathan; Yassin, Yusef; Ward, Trenton; Caldwell, John; Attie, Tawab (2019). A Comparison of Machine Learning Algorithms in Keystroke Dynamics. IEEE 2019 International Conference on Computational Science and Computational Intelligence (CSCI)

3 .Nick Bartlow and Bojan Cukic (2017). Evaluating Reliability of Credential Hardening through Keystroke Dynamics. 17th International Symposium on Software Reliability Engineering (ISSRE'17) 0-7695-2684-5/17 IEEE

4 .KEYSTROKE DYNAMICS FOR USER AUTHENTICATION Priyanka Namnaik, Rajeshree Kurale, Sanyukta Mahindrakar